

Introduction to **easysoft**[®] ODBC for CODA

Version 1.5

Topics covered

Module 1. Introduction to ODBC (Open Database Connectivity) 1-1

Background to ODBC _____	1-3
Architecture _____	1-5
Driver Manager _____	1-6
Drivers _____	1-6
Data Sources _____	1-11
Conformance and Standards _____	1-13
ODBC Review Questions _____	1-15
ODBC Review Answers _____	1-16
Bibliography _____	1-18

Module 2. Introduction to SQL (Structured Query Language) 2-1

Structured Query Language _____	2-3
Tabular Structure of Data _____	2-3
Indexes _____	2-6
Combining Information from Different Tables _____	2-7
Additional Notes _____	2-9
Manipulating Data using SQL _____	2-10
SELECT statement basics _____	2-10
SELECT Statement - Advanced _____	2-12
INSERT Statement _____	2-19
UPDATE Statement _____	2-20
DELETE Statement _____	2-21
Summary _____	2-22
SQL Review Questions _____	2-23
SQL Review Answers _____	2-24
Further Exercises _____	2-26
Suggested Answers to Exercises _____	2-28
Additional Notes _____	2-30
The Easysoft Catalog _____	2-30
Supported SQL _____	2-30
Reserved Words _____	2-31
Bibliography _____	2-33
Easysoft Query for Windows _____	2-34
Start and Connect _____	2-34
The Example Database _____	2-37
Exercises _____	2-39
Suggested Answers _____	2-41

Module 3. The Easysoft System	3-1
Easysoft Architecture _____	3-2
Overall Process of using Easysoft _____	3-3
Data Sources and Catalogs _____	3-4
Easysoft Catalog Structure _____	3-5
Summary _____	3-6
Review Questions _____	3-7
Review Answers _____	3-8
Module 4. Microsoft ODBC Administrator	4-1
Purpose _____	4-2
Administrator _____	4-3
Set up a Data Source _____	4-4
Settings _____	4-6
Additional Notes _____	4-7
Summary _____	4-9
Supplement: troubleshooting _____	4-10
Module 5. Easysoft Administration	5-1
CODA Utilities _____	5-2
Changing Catalog Passwords _____	5-3
Adding CODA Companies to Catalog _____	5-4
Changing CODA Usernames, Passwords and Privileges _____	5-6
Ordering BALANCE and BUDGET Data _____	5-8
Summary _____	5-9
Catalog Administration Exercises _____	5-10
Administration Review Questions _____	5-14
Administration Review Answers _____	5-15
Easysql _____	5-16
Module 6. CODA Tables, Indexes and Relationships	6-1
CODA-IAS and Table Correspondences _____	6-2
List of CODA Tables _____	6-4
CODA Table Relationships _____	6-5
Column Details Tables _____	6-6
Index Details Tables _____	6-8
Easysoft Optimisation _____	6-9
Changing Codaxref Batch Job Defaults _____	6-9
Summary _____	6-11
CODA Tables Review Questions _____	6-12
CODA Tables Review Answers _____	6-14
Module 7. Using ODBC Applications with CODA	7-1
Microsoft Excel _____	7-2
Download Entire Table _____	7-2
Add Criteria _____	7-5
View SQL _____	7-7
Return Data to Excel _____	7-8
Microsoft Access _____	7-10
Download Entire Table _____	7-10
Add Criteria _____	7-12

View SQL	7-13
View Indexes	7-14
Lotus 1-2-3	7-15
Configure Lotus 1-2-3 to work with Easysoft ODBC	7-15
Download Entire Table	7-16
Add Criteria	7-18
View SQL	7-18
Microsoft Word Mail Merge	7-19
Module 8. CODA Training Company	8-1
Easysoft Travel Company	8-2
Chart of Accounts	8-2
Nominals, Subaccounts and Level3s	8-5
Account Group Hierarchy	8-8
List of Currencies	8-11
Module 9. CODA Exercises	9-1
General Information	9-2
Tips and Reminders	9-2
1. Master File Maintenance	9-6
Task 1.1 : Produce a list of all Ledgers	9-6
Task 1.2 : List Ledgers and their Control Account	9-7
Task 1.3 : Produce a directory of all Nominals	9-8
Task 1.4 : Produce a directory of the Sales Nominal (2100)	9-9
Task 1.5 : Produce a directory of the Sales Analysis Nominals	9-10
Task 1.6 : List SUBACCOUNTS with a destination of Blackpool or Bermuda (Hamilton)	9-11
Task 1.7 : List Name and address Subaccounts and the LEDGER they belong to	9-13
Task 1.8 : List existing currencies and currency rates	9-15
Task 1.9 : Find out how many customers there are in the database	9-19
Task 1.10 : Produce a list of CODA Account Groups	9-21
Task 1.11 : Produce a list of the Group Hierarchy to one level	9-22
Task 1.12 : Produce a list of the Group Hierarchy to two levels	9-24
Task 1.13 : Repeat previous tasks using the Easysoft Excel macro for CODA9-27	
2. Budgets	9-28
Task 2.1 : Download budget information, change it and upload	9-28
Task 2.2 : Use the Macro to download and upload budgets	9-31
3. Reporting	9-32
Task 3.1 : Produce a Trial Balance	9-32
Task 3.2 : Produce a Trial Balance using the macro	9-34
Task 3.3 : Produce an Actual Budget Variance report using pivot tables	9-35
Task 3.4 : Produce an Actual Budget Variance report using the macro	9-38
Task 3.5 : Produce a Profit and Loss Account	9-39
Task 3.6 : Use the macro to produce a Profit and Loss Account	9-41
Task 3.7 : Produce a Balance Sheet for the year so far	9-42
Task 3.8 : Use the macro to produce a Balance Sheet	9-45
Task 3.9 : Produce an Ageing Summary	9-46
Task 3.10 : Use the macro to Produce an Ageing Summary	9-48

4. Input and Intraday	9-49
Task 4.1 : Invoice Upload	9-49
Task 4.2 : Invoice Download	9-51
Task 4.3 : Upload Salary Journal	9-52
Task 4.4 : Download Salary Journal	9-53

Module 10. Easysoft Excel Macro for CODA **10-1**

Overview of Macro	10-2
Macro Installation	10-3
Initialisation	10-5
Masters-related options	10-7
Download	10-7
Upload	10-9
Reports-related options	10-10
Budgets	10-10
Fixed Reports	10-12
Input and Intraday	10-18
Lookups	10-20
Reports	10-21
Information	10-24
About Easysoft ODBC for CODA	10-24
Contacting Easysoft	10-24
Company Structure	27
Using CODA	28
External Access to CODA Data	30
CODA-IAS Review Questions	31
CODA-IAS Review Answers	32
Starting the Driver Installation	34
Easysoft ODBC Logging	39
EASYSOFT.INI	40
Example Server Log File	41
Microsoft Trace Options	42
Version 3.0 Trace Options	42
SQL.LOG Described	43
How to Work Out What the Server is Doing	44
Find the Process Causing the Problem	44
View Log File	45
Stop the Process	46
Support Check List	48
The Easysoft FTP Site	49
Sending Data to Easysoft	50

Appendices **A-1**

1. Introduction to ODBC (Open Database Connectivity)

This module outlines the Open Database Connectivity (ODBC) functionality and architecture. These course notes address the background problem that ODBC deals with.

In this module you will learn

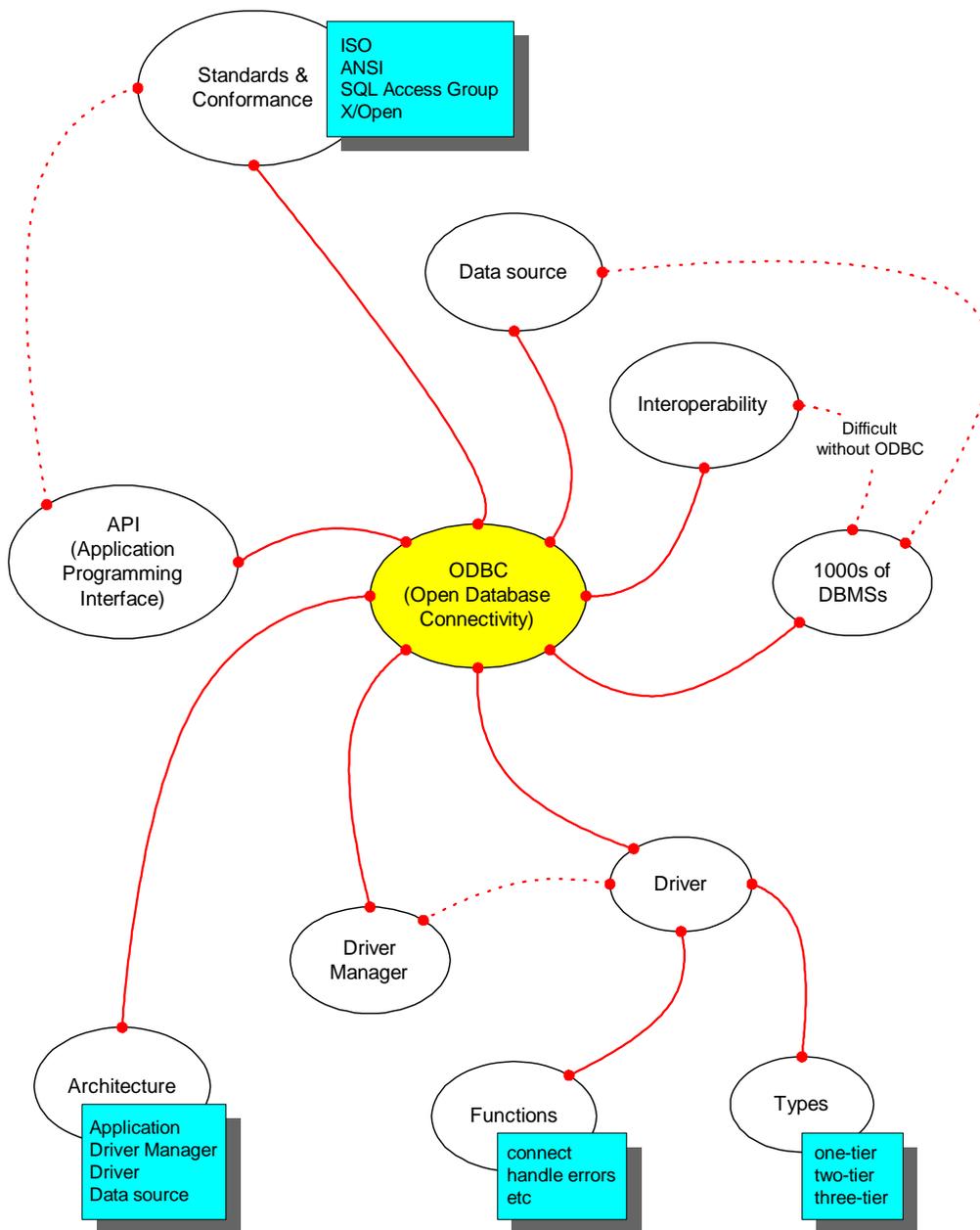
- ✦ why we need ODBC and what the alternatives are
- ✦ the architecture of ODBC
- ✦ the terminology used with ODBC

At the end of the module there are some review questions.

Contents

Background to ODBC _____	1-3
Architecture _____	1-5
Driver Manager _____	1-6
Drivers _____	1-6
Data Sources _____	1-11
Conformance and Standards _____	1-13
ODBC Review Questions _____	1-15
ODBC Review Answers _____	1-16
Bibliography _____	1-18

ODBC Mind Map



Background to ODBC

This section presents a brief background to ODBC, and then we discuss the overall architecture of ODBC, the Driver Manager, drivers and conformance.

Position

Hundreds of different software manufacturers produce *Database Management Systems* (DBMS) for storing, accessing and manipulating data. For various reasons, users may wish to access data that is stored using any number of different DBMSs via various PC applications.

Problem

Typically, each different DBMS uses storage and access methods for the *database* (i.e. collection of data files) that are incompatible with any other DBMS, thus complicating data access via PC applications.

Alternatives

1. Each PC application contains different versions of software to enable it to communicate with all the DBMSs on the market.

For example, consider Excel and three different Database Management Systems, such as Oracle, Ingres and Sybase.

It would take three different versions of Excel - for Oracle, Ingres and Sybase - and a vast amount of work from developers of these applications. Plus, users would have to buy and maintain three different versions of Excel.

2. Define a standard that uses common terms to describe the data and operations on that data.

For each different DBMS, the software vendor provides software which converts the common terms into the particular operations of the DBMS. Conversely, any outgoing data is presented in the defined format.

Any computer program, providing it conforms to the standard, can send a request to the DBMS, and the data will be returned in a standard format.

Solution

The latter option is the ODBC solution to the problem. ODBC was developed by Microsoft to give a single *API* (Application Programming Interface) which can access a variety of data sources. This means developers can write applications which are not targeted to any specific DBMS. End users connect applications to their databases by using add-in modules called *drivers*, which are available from database vendors and driver developers.

The aim of ODBC is to allow interoperability between different DBMSs and applications which run under the Microsoft Windows operating system. It works by supporting heterogeneous data access - in other words, applications access different data sources by using drivers which access the data. Applications can submit any SQL statement which is supported by a given driver.

SQL is a computer language that is used with a particular type of database called a relational database, and in general, when reference is made to databases, we mean relational databases. It is possible to use ODBC, in conjunction with additional software, to access non-relational data. This is exactly what we do at Easysoft.

GLOSSARY

API (Application Programming Interface) A set of standards that allows an application to connect to a DBMS.

Database A collection of data files.

Data Source (in ODBC terms) A combination of the data, the DBMS used to access that data, the operating system and the network.

DBMS (Database Management System) The software that manages access to a database.

Driver The software that implements ODBC functions.

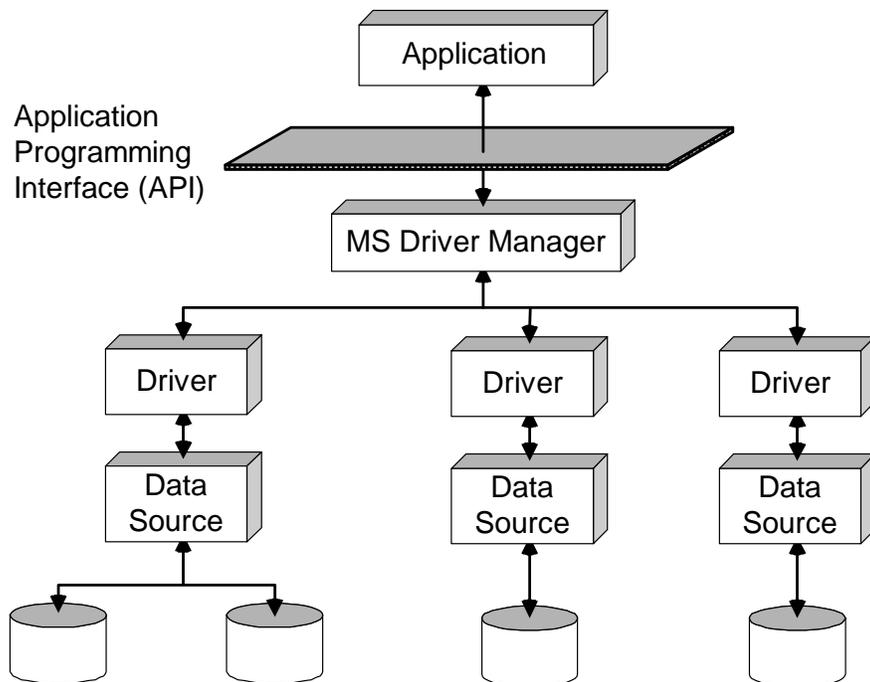
SQL (Structured Query Language) An official standard computer language for interacting with (relational) database systems.

Architecture

The four main components of the ODBC architecture are the *Application*, the *Driver Manager*, the *Driver* and the *Data source*. Their relationship is shown below, and their functions are briefly described. The following section then explain the components in more detail.

Note: in later modules you will use the Microsoft ODBC Data Sources Administrator. For all practical purposes, this is equivalent to the Driver Manager.

ODBC Architecture



The ODBC Interface, which essentially defines the API and SQL syntax, is not part of the architecture, and is described in the section entitled “Conformance” on page 1-13.

An application, such as Microsoft Access or Lotus 1-2-3, calls ODBC functions (i.e. requests the performance of ODBC operations) which are sent to the driver via the Driver Manager. As far as the application is concerned the driver and Driver Manager appear to be a single functional unit. A single application may wish to access data from a number of different sources and ODBC is designed to allow this.

The Driver Manager (provided by Microsoft) loads drivers for an application when an application calls certain SQL functions and, if requested, traces calls and keeps a log file of these.

The driver processes the function calls sent by the application, submits SQL calls to the data source and returns results to the application. It may change the syntax of the request to conform to the syntax used by the DBMS in the data source. Another function of the driver is to format errors into standard error codes and return these codes to the application.

In the context of ODBC a data source is more than just a set of data files. A data source consists of a set of data and its associated environment, which includes the operating system, Database Management System and networks (if any). Typically, the term data source is used in a loose fashion to describe any software component that is not one of the other three described above. An application may access more than one data source at any given time if it so requires.

Driver Manager

The Driver Manager sits between applications and drivers in the ODBC architecture and it manages the interactions between an application and a driver. Both multiple applications and multiple drivers can be managed simultaneously by the Driver Manager.

An application calls an ODBC function, and the Driver Manager sends that call to the appropriate driver. The first time that an application uses ODBC to connect to data, the required driver is determined by the Driver Manager and is then loaded into memory. Thereafter, the Driver Manager takes each incoming function call from the application and calls a function of the same name in the driver. (The only exception to this is when the function call is one that the Driver Manager is expected to process, such as when an application asks for the name of a driver). Finally, when the application calls the ODBC function which requests disconnection, the Driver Manager unloads the driver from memory (unless another application is also using that driver).

Additionally, elementary error checking is performed by the Driver Manager to ensure that functions are called in the correct order and that arguments contain valid values.

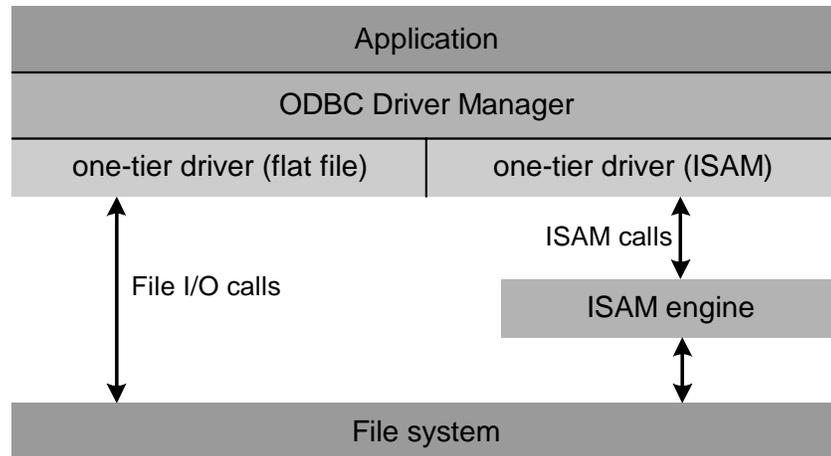
Drivers

First, we look at the architecture of drivers, and then the basic functions of a driver are presented.

Types of Driver

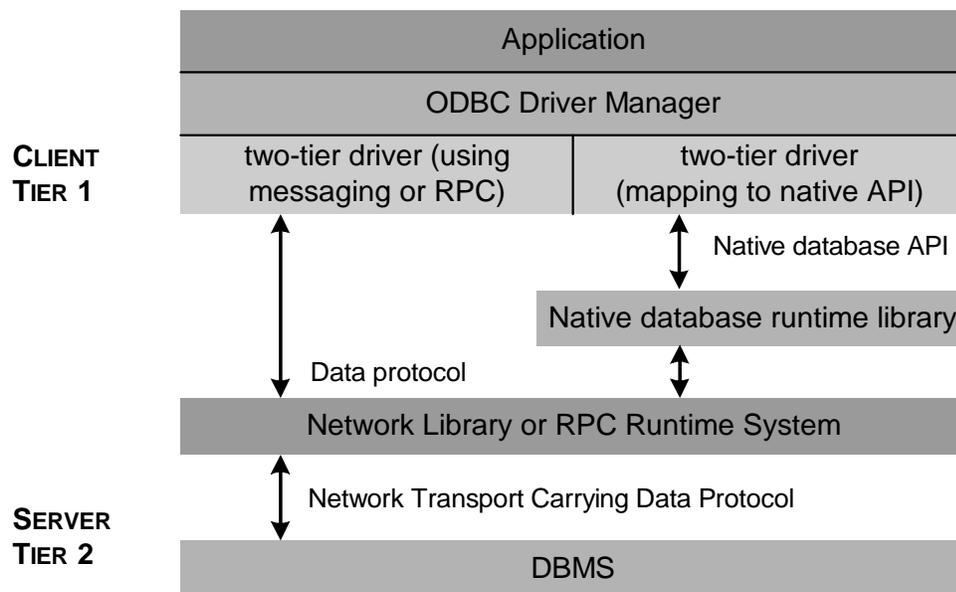
Two basic types of driver are defined by ODBC. A *single-tier driver* processes data directly (i.e. it processes both ODBC calls and resultant SQL statements) whereas a *multi-tier driver* sends SQL statements to the data source. Multi-tier drivers can be further described as two-tier drivers and three-tier drivers. This is shown in the figures below.

One-tier driver architecture



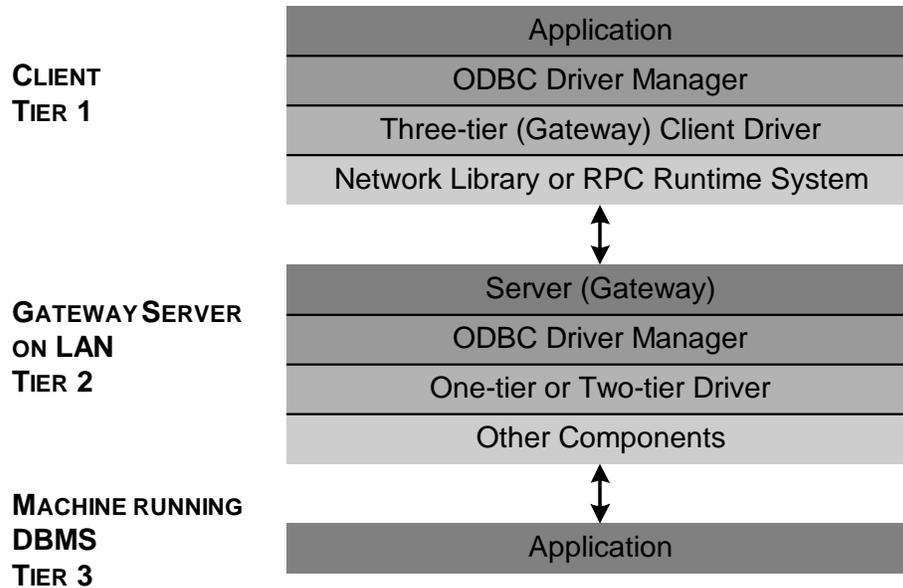
The term one-tier in ODBC refers to a driver that accesses a desktop database file or a flat file. The main distinguishing feature of one-tier drivers is that the driver itself does all of the SQL processing.

Two-tier driver architecture



Two-tier systems are the classic client/server systems. The client, i.e. the driver, sends and receives the Database Management System's data protocol or maps to the native database API. It doesn't access the data directly. The server, i.e. the DBMS, receives SQL requests from the client, executes them and then sends the results back to the client.

Three-tier driver architecture



From the perspective of a client, there is little difference between two-tier and three-tier drivers. The main difference is that in a three-tier system, the client connects to a server which acts as a gateway to the target Database Management System.

To avoid confusion it is important to note the distinction between driver architecture and hardware architecture. A single-tier driver can run on a physical client-server hardware architecture (i.e. the driver accesses data that is stored on a server, but the driver itself still sends SQL from the client to the server). Conversely, a multi-tier driver may reside on a single machine - the driver sends SQL requests to the data access software which processes these requests.

Typically, however, in a multi-tier configuration, the application, driver and Driver Manager reside on one machine (the client), and the database and data access software reside on a different system (the server). This has the advantage over a single-tier driver on client-server hardware that only the required data is sent over the network.

Driver Functionality

Once a driver has been loaded by the Driver Manager, there are a number of tasks it performs:

- Connect to machine containing data
- Deal with errors
- Transform SQL
- Catalog functions
- Information functions
- Data type conversions

Connect to Machine Containing Data

Once the Driver Manager has loaded a driver, the driver first connects to the machine on which the data resides. For one-tier drivers there is no network communication to deal with because the data is local. Two-tier and three-tier drivers must also deal with network communication.

Consider a two-tier driver. It has to establish a network connection to the DBMS. Typically, the driver is responsible for displaying a dialog box in which users type their names and passwords for logging into the DBMS. Then, after this security check, the driver loads the correct network library (unless it is already linked into the driver). The network library functions then connect to the server using the server name or network address information that was provided when the data source was set up.

Deal with Errors



To give applications a standard way of dealing with error conditions, ODBC requires drivers to provide not only DBMS-specific error codes, but also standard error codes. Although the overall types of errors that are returned by DBMSs are similar, the internals, such as error numbers, message content and the amount of information displayed will all be different. To overcome these differences ODBC provides:

- a return code mechanism which reports success or failure for each function
- a standard error function that applications can call every time an ODBC function fails or presents a warning
- standard error codes for over 85 error conditions
- all the error information that is supplied from the DBMS; native error codes and messages are returned to the application
- error information from the driver itself
- a tagging scheme that identifies the component that reported the error. This example shows the Easysoft driver returning a server login error message:



- the ability to return multiple errors from a single ODBC function call

From the perspective of an end user who is using an interactive application, the main advantage to the standard error handling is the common style. However, if an application itself makes use of the errors returned from a DBMS in order to perform some further action, then there is an important advantage to having a standard code. Instead of needing programming code to deal with all the possible DBMSs that might be used by the application, there only needs to be one check for the standard error that is returned.

Transform SQL

DBMSs do not always use standard SQL as defined by ISO/ANSI, and therefore, drivers need the capability of transforming one dialect of SQL into another. Some DBMSs do not provide all the functionality defined in the standard, and drivers need not need to add functionality. However, if a DBMS provides the same functionality as the standard, but if it does it differently from the standard, then the driver must translate the standard into a form that is understood by the DBMS.

ODBC provides extensions to standard SQL which provide essential interoperability functionality. For example, many DBMSs support the SQL data types DATE, TIME and TIMESTAMP. However, the literal formats used with SQL statements can vary between DBMSs. The ISO SQL standard representation for DATE data is *yyyy-mm-dd*, where *yyyy* represents four digits corresponding to the year, *mm* represents two digits corresponding to the month and *dd* represents two digits corresponding to the day. This is what ODBC uses; if a driver supports the DATE data type, then it must transform this into the syntax of the DBMS that it is accessing.

Catalog Functions

A major task of a driver is to provide information about the tables, columns and other objects in a DBMS in a uniform way so that users can interactively select the items of interest. ODBC provides a set of catalog functions which provides applications with the basic information that they need from the catalog in the DBMS. Each driver generates an SQL query that returns the required information to the application. The catalog functions defined in ODBC include the ability to return information about:

- tables that a user can access (*SQLTables* function)
- the columns in a table (*SQLColumns* function)
- information about a table, such as indexes, if there are any, that are defined on the table (*SQLStatistics* function)

Information Functions

A driver must be able to provide information about its capabilities and the capabilities of the DBMS it interacts with. A few important functions are shown here:

- *SQLGetInfo* informs the application of the conformance level, version and functionality capabilities of the driver

- *SQLGetTypeInfo* returns information about the data types a DBMS uses.
- *SQLDataSources*, *SQLDrivers* and *SQLGetFunctions* inform the application of all the available data sources, all the drivers and all the functions that a driver supports.

Data Type Conversions

One of the goals of ODBC is to provide data in a form that is most suitable to any given application.

Any data type conversion that makes sense is allowed by ODBC. The application specifies the conversion, and the driver must do this conversion.

Data Sources

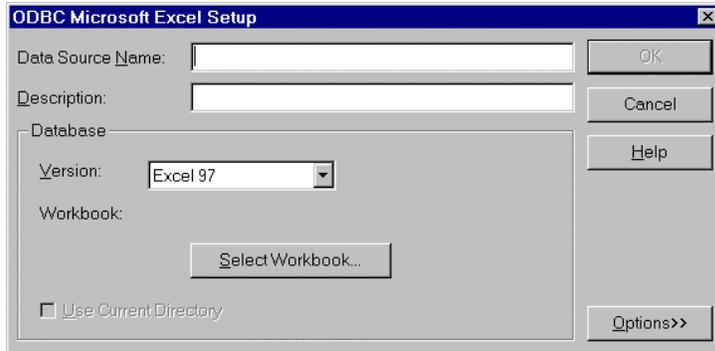
Ideally, users should not be aware of underlying communications software, the drivers, the DBMSs, server addresses etc. To hide these things from end users, ODBC uses a *data source name* (DSN). All the underlying software components of any given data source are mapped to one data source name. Typically, the data source name is chosen by the end user or a system administrator, and ideally, it should be a name that self-evidently describes the data that it represents (for example, "SALES DATA").

Within ODBC the term *data source* is used for two purposes:

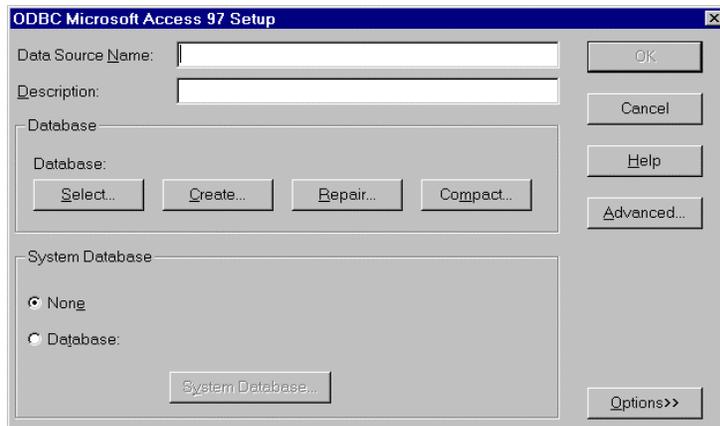
1. It defines the data that an end user wants to access. The data source includes the DBMS software that runs on a particular machine, the operating system, and networking software.
2. It is used to refer to the data source name (so instead of saying, for example, "connect to the data source which is defined by the 'SALES' data source name", we say "connect to the 'SALES' data source").

Every driver has its own particular set up, and a few examples are shown here.

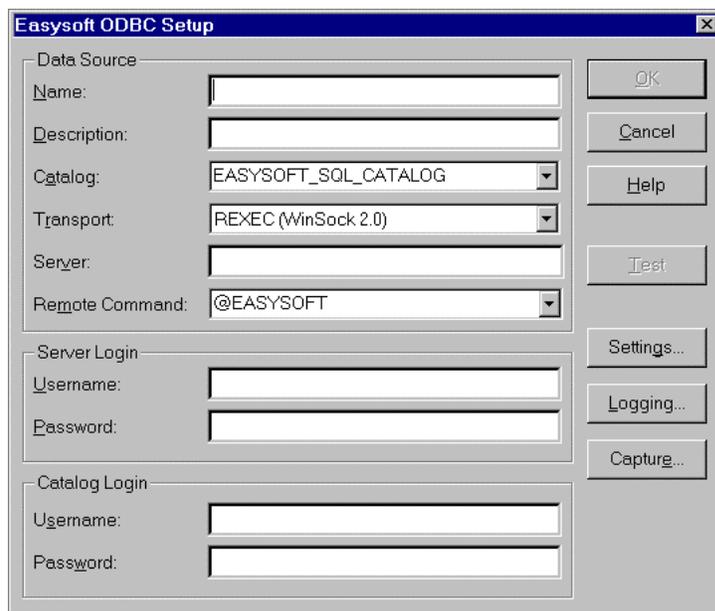
ODBC Microsoft Excel Setup



ODBC Microsoft Access 97 Setup



Easysoft ODBC Setup



Conformance and Standards

Successful operation of ODBC depends upon defining a set of standards common to all participating elements of a system.

ODBC is used to facilitate data communication between different applications. It allows the use of plug-in modules - in other words, drivers from many different vendors can be used. However, all DBMSs and applications provide different functionality and so for systems to communicate, it is necessary to define standards for functionality. ODBC defines conformance levels in two areas of functionality, the ODBC API and SQL grammar (including data types), and for both of these there are three levels of conformance, each one being more comprehensive than the level below.

The conformance levels for the API are named: core API, level 1 API, level 2 API.

The conformance levels for SQL are named: minimum SQL grammar, core SQL grammar, extended SQL grammar.

Selecting an appropriate level of conformance depends upon the needs of the application. Microsoft suggest that driver developers implement all level 1 ODBC API functions since many ODBC applications require this. Conforming to a given level does not mean that additional functionality cannot be provided, but if a driver is claimed to conform to a given level, then all the functionality of that level should be provided.

Conformance levels can be determined in a number of ways:

- from driver documentation
- call the *SQLGetFunctions* function from the Driver Manager
- if a driver supports the *SQLGetInfo* and *SQLGetTypeInfo* functions you can use these to return information on conformance. Although these are level 1 functions many core level drivers support them

Standards Groups

Various standards groups have influenced the development of ODBC. Notable ones are

ANSI (American National Standards Institute) This is a U.S. government agency which sets standards in commerce and industry. It works in collaboration with the ISO (below).

ISO (International Standards Organisation) This is a federation of national standards organisations.

SQL Access Group. This was formed in 1989 with the aim of accelerating the acceptance of formal standards which would be use to increase portability and interoperability for database applications. The SQL Access Group merged with X/Open (below) in 1994.

X/Open. This organisation deals with portability issues in a practical way. Wherever possible, X/Open follows ISO standards, but if such standards do not exist, then this group will define additions to existing standards in order to help developers create portable applications.

Summary

In this module you have learnt

- ✦ ODBC allows different applications to communicate by means of a common interface
- ✦ Applications use plug-in modules called drivers to communicate
- ✦ When an application wants data, it connects to a data source using a driver
- ✦ The Microsoft Driver Manager loads drivers for a calling application
- ✦ A data source consists of a set of data and its associated environment, which includes the operating system, Database Management System and networks (if any)
- ✦ Single-tier drivers process both ODBC calls and SQL statements (SQL is a language for relational databases)
- ✦ Two-tier and three-tier drivers send SQL to the data source
- ✦ Three conformance levels are defined for both ODBC and SQL

ODBC Review Questions

1. What does the acronym ODBC stand for?

2. What is ODBC used for?

3. The four main components of ODBC architecture are:
 - a.
 - b.
 - c.
 - d.

4. What is a database?

5. Name at least two functions of the Microsoft Driver Manager

6. What is the function of a driver?

7. Name the three basic types of driver architecture.

8. What's the difference between the three driver architectures?

9. What is Structured Query Language (SQL)?

10. What is an API (Application Programming Interface)?

11. What is the function of conformance levels?

ODBC Review Answers

1. What does the acronym ODBC stand for?

Open Database Connectivity

2. What is ODBC used for?

To connect different (and hence incompatible) databases

3. The four components of ODBC architecture are:

a. Application b. Driver Manager c. Driver d. data source

4. What is a database?

A collection of data files

5. Name at least two functions of the Microsoft Driver Manager.

a. loads drivers
b. traces and keeps a log of SQL function calls
c. list the driver that is used by each data source (information is taken from the ODBC.INI file)

6. What is the function of a driver?

a. Processes ODBC function calls which are sent by an application
b. Submits SQL calls to a data source
c. Returns results to an application
d. Format errors into standard forms, and return these errors to an application

7. Name the three basic types of driver architecture.

a. Single-tier
b. Two-tier
c. Three-tier

8. What's the difference between the three driver architectures?

A one-tier driver processes data directly (i.e. it processes both ODBC calls and the resultant SQL statements).

Two-tier and three-tier drivers send SQL statements to the data source for processing. The main difference between two-tier and three-tier drivers is that with three-tier drivers the client connects to a server which acts as a gateway to the DBMS.

9. What is Structured Query Language (SQL)?

An official standard language for interacting with relational database systems.

10. What is an API (Application Programming Interface)?

An API is a set of standards that allows an application to connect to a DBMS.

11. What is the function of conformance levels?

Since all DBMSs and applications provide different functionality, in order for systems to communicate, it is necessary to define standards for functionality.

Bibliography



This bibliography lists a few sources of information relevant to information in this module. Entries are listed alphabetically by title, except that initial articles (i.e. 'A' and 'The') are ignored for the purposes of ordering.

Inside ODBC, K. Geiger. Published by: Microsoft Press, 1995. ISBN 1-55615-815-7.

Microsoft ODBC 2.0 Programmer's Reference and SDK Guide, Microsoft. Published by Microsoft Press, 1994. ISBN 1-55615-658-8.

Microsoft TechNet CD. This CD contains technical information relevant to the Microsoft Corporation. The CD is updated frequently.

Teach yourself ODBC Programming in 21 days. B. Whiting, B. Morgan, J. Perkins. Sams Publishing, 1996. ISBN 0-672-30609-3.

The ODBC Solution - Open Database Connectivity in Distributed Environments, R. Signore, J. Creamer, M. O. Stegman. Published by McGraw-Hill, 1995. ISBN 0-07-911880-1.

Using ODBC 2, R. Gryphon, L. Charpentier, J. Oelschlaeger, A. Shoemaker, J. Cross, A. W. Lilley. QUE Corporation, 1995. ISBN 0-7897-0015-8.

2. Introduction to SQL

(Structured Query Language)

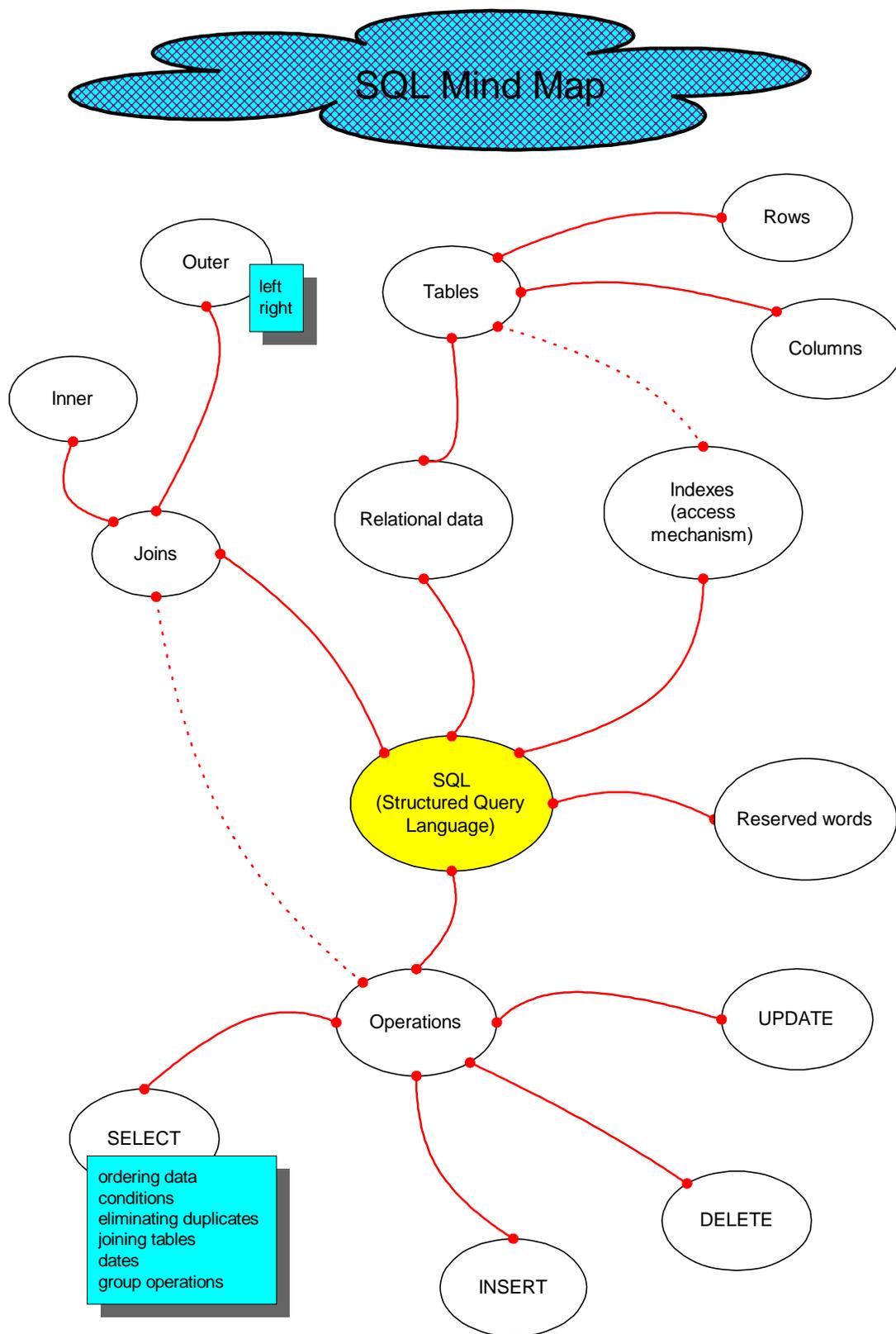
In this module you will learn

- ✧ what SQL is and the basic properties of SQL
- ✧ the tabular structure of data
- ✧ how to retrieve, enter, change and delete data using SQL
- ✧ how to set conditions to specify what data is returned in a query
- ✧ how to obtain data from more than one table
- ✧ how to perform aggregate functions (e.g. count the records, sum of values) on the data that is returned

At the end of the module there are some review questions.

Contents

Structured Query Language	2-3
Tabular Structure of Data	2-3
Indexes	2-6
Combining Information from Different Tables	2-7
Additional Notes	2-9
Manipulating Data using SQL	2-10
SELECT statement basics	2-10
SELECT Statement - Advanced	2-12
INSERT Statement	2-19
UPDATE Statement	2-20
DELETE Statement	2-21
Summary	2-22
SQL Review Questions	2-23
SQL Review Answers	2-24
Further Exercises	2-26
Suggested Answers to Exercises	2-28
Additional Notes	2-30
The Easysoft Catalog	2-30
Supported SQL	2-30
Reserved Words	2-31
Bibliography	2-33
Easysoft Query for Windows	2-34
Start and Connect	2-34
The Example Database	2-37
Exercises	2-39
Suggested Answers	2-41



Structured Query Language

SQL stands for *Structured Query Language*. It is a computing language for interacting with *relational* database systems. In simple terms, a relational database is one in which all the data is perceived as being contained in *tables*, and a table itself is composed of rows and columns of data.

SQL is used with ODBC because ODBC is designed for relational databases. The idea behind ODBC is that any relational database can communicate with any other relational database or application. The differences in physical implementation between different database systems are overcome because ODBC defines standards for communicating - any application that supports ODBC (i.e. an ODBC-compliant application) can send and receive data from any other application that conforms to ODBC standards. There are different levels of conformance to various ODBC standards and there are official standards for different versions of SQL. In addition, there are ODBC extensions to SQL. For the discussion here, these different standards are not important, but we do sometimes use the ODBC extensions (these are clearly indicated when they are used).

SQL began to be developed in the early 1970s by IBM and has grown considerably since then. By the late 1980s, both the American National Standards Institute (ANSI) and the International Standards Organisation (ISO) had produced definitions for SQL. In the business and industrial world there are many hundreds, probably thousands, of applications and systems which use SQL.

Tabular Structure of Data

All data is perceived to be stored in *tables*. These consist of *columns* and *rows*. A column is the vertical dimension of the table and a row is the horizontal dimension of the table (sometimes columns are also called fields and rows are called records). A *database* is a collection of tables.

Imagine that there is a database that is used to store information about both the products that are made by a company and the orders that customers send to the company. For now, consider only the products that are made. Say that we need information on only the product number, name and price. This information is shown in the PRODUCTS table below.

PRODUCTS		
<u>P_NUMBER</u>	<u>P_NAME</u>	<u>P_PRICE</u>
P1	widgit	13.50
P2	fergulator	19.60
P3	doofer	15.45
P4	dongle	8.05

In this table there are four rows corresponding to the four products made by the company. For each of these products, three items of information relating to that product are stored, namely the product number, the product name and the price.

Say that we want to keep information about orders despatched. For this, we will use two tables. First, here is the ORDERS table, which contains information about each order.

ORDERS		
O_NUMBER	CUSTOMER	O_DATE
O1	Smith	29-01-1997
O2	Jones	29-01-1997
O3	Black	03-02-1997
O4	Smith	04-02-1997

The format of date information in a DBMS (Database Management System) is application dependent.

For exemplary purposes we don't consider any further the relationships between other tables that would be required in a real-world case. Let us say that the CUSTOMER column sufficient to identify all the customers known to the database.

For each order, we need to know the product(s) requested, and the number required. There is another table which is used to keep track of the details of each order that are despatched by the company. This is called OLINES (for order lines).

OLINES			
O_NUMBER	P_NUMBER	QUANTITY	TOTAL_PRICE
O1	P1	1	13.50
O1	P4	2	16.10
O3	P4	1	8.05
O4	P2	1	19.60
O4	P4	1	8.05

To relate information in different tables, the tables must be "joined" (see "Combining Information from Different Tables", page 2-7 of this module).

GLOSSARY

Column The vertical dimension of a table.

Database A collection of tables.

Row The horizontal dimension of a table.

Table A table contains data which is arranged in rows and columns.

Indexes

To speed up the rate of data retrieval, indexes can be used. An index can be unique or non-unique. Indexes provide ordered access to rows of a table, and they are based on the values of one or more columns of a table. An additional feature is that the ordering may be ascending or descending.

Although indexes are useful when retrieving data, they slow down the input of data (because in addition to updating the table, the index must be maintained). It is not possible to state definitively when indexes should and should not be used, because this depends upon how the data is used. Typically, if the data does not change frequently, it will be more beneficial to use an index than if the data changes frequently.

As example of a unique index, consider the ORDERS table. This could have a unique index defined on the O_NUMBER column. If we are looking for a specific order, then this index would allow rapid retrieval of the data.

ORDERS		
O_NUMBER	CUSTOMER	O_DATE
O1	Smith	29-01-1997
O2	Jones	29-01-1997
O3	Black	03-02-1997
O4	Smith	04-02-1997

unique index: every data value in this column is different.

non-unique index: data values in this column can be the same.

A non-unique index could be defined on the CUSTOMER column, so that all the orders for a specific customer could be found rapidly.

Combining Information from Different Tables

Say we want to gather information for the order line details of a prototypical invoice. If we don't need to show part names, then we could just obtain data from the OLINES table. If we wanted to include part names, then we would have to combine information from both the OLINES and PRODUCTS tables, because product names are stored in the PRODUCTS table. This operation is called a join. The tables are joined on the P_NUMBER field, which is common to both tables.

Types of Join

There are essentially two basic types of join called *inner join* and *outer join*. The inner join selects records which match in both tables. The outer join operation selects all the records from one table, even if they do not have a corresponding match in the other table. Outer joins can be classified as left outer joins and right outer joins. A left outer join is one where all the records in the "left" table (i.e. the first table in the join condition) are shown. A right outer join is one where all the records in the "right" table (i.e. the second table in the join condition) are shown.

Inner and outer joins are exemplified using the OLINES and PRODUCTS tables. Say the join is OLINES joining PRODUCTS, and say that we just want to see just the P_NUMBER and P_NAME fields (purely so that this page is not cluttered with unnecessary text). For ease of reference, the relevant data is replicated here.

OLINES		PRODUCTS	
<u>O_NUMBER</u>	<u>P_NUMBER</u>	<u>P_NUMBER</u>	<u>P_NAME</u>
O1	P1	P1	widget
O1	P4	P2	fergulator
O3	P4	P3	doofer
O4	P2	P4	dongle
O4	P4		

On all the joins that will be shown, we will join P_NUMBER from the OLINES table with P_NUMBER in the PRODUCTS table. In reality, these P_NUMBERs are the same thing - the product number of a particular product. However, in our representation of reality, they are different, and we need some method for distinguishing them. The convention is to prefix the column name with the table name, so we have OLINES.P_NUMBER and PRODUCTS.P_NUMBER.

GLOSSARY

- Index** A data structure that behaves like an ordered list of pointers to the rows of a table.
- Inner Join** Returns records which match in the join column(s) in both tables.
- Join** Basically, a join is a query in which data is retrieved from more than one table.
- NULL** represents a case where a value is unknown.
- Outer Join** All records from one of the tables are returned, even if there are no matching values in the join column(s) of the other table.

Inner Join

The result of an inner join (just showing P_NUMBER and P_NAME, and with duplicates removed) is:

OLINES.P_NUMBER	PRODUCTS.P_NAME
P1	widget
P2	fergulator
P4	dongle

Of all the products available, only three products, namely P1, P2 and P4, appear in the OLINES table. Therefore these three products appear in the result of the inner join.

Left Outer Join

The 'left' table is OLINES, so all the product numbers which appear in this table will be matched with all the product numbers in the PRODUCTS table. The result is:

OLINES.P_NUMBER	PRODUCTS.P_NAME
P1	widget
P2	fergulator
P4	dongle

In our case, the result is the same as an inner join (because there cannot be any products in OLINES which do not occur in PRODUCTS).

Right Outer Join

The ‘right’ table is PRODUCTS, so all the product numbers which appear in this table will be matched with all the product numbers in the PRODUCTS table. The result is:

OLINES.P_NUMBER	PRODUCTS.P_NAME
P1	widget
P2	fergulator
NULL	doofer
P4	dongle

The product ‘doofer’ (P3) exists in the PRODUCTS table, so it appears (indirectly as P_NAME) in the join result. P3 does not exist in the O_LINES table, so it does not appear in the result. The term NULL in SQL refers to unknown or missing information.

Additional Notes

Typically, table names and the column names are typed in upper case. This is not mandatory, but is a convention - lowercase letters are allowed. Table and column names must start with an alphabetic letter and cannot contain spaces - the underscore character (_) is usually used as a separator.

SQL is not concerned with the actual storage of data.

There are many pictorial methods of describing the structure of a database. If just the table structure is being defined, then a useful method is to use a variant of a Bachman diagram; tables are shown as rectangles, and lines linking the rectangles show the relationships between tables. Our database would be shown thus:

Pictorial representation of database



The ‘crow’s foot’ () indicates a one-to-many relationship, the single line at PRODUCTS and ORDERS representing the ‘one’, and the triple lines (at ORDER_LINES) representing ‘many’. That is, one product can appear in many different order lines, but each order line can reference just a single product. Similarly, each order in ORDERS can have many (one or more) different lines, whereas any one order line belongs to just one order.

Manipulating Data using SQL

Although SQL is capable of far more than just manipulating data, as far as we are concerned, its purpose is to retrieve and store data. The four operations that are available for this are the SELECT, INSERT, UPDATE and DELETE statements. These do exactly what their names suggest, and we will look at each of them in turn, starting with the most complex (in terms of its richness of structure), namely SELECT. Each of the sections will start with a brief overview of the structure of the statement, followed by one or more examples from the example database described above. All the possible options for the statements are not covered - refer to one of the many books on SQL for a comprehensive reference. In the real world, SQL is often used to deal with millions of records; the principles can be explained best when there are only a few rows in each of the tables.

SELECT statement basics

The SELECT statement is used to extract data from one or more tables in the database. Its basic structure is

```
SELECT    <columns>
FROM      <tables>
WHERE     <search conditions>
```

<columns> can be one or more columns from one or more of the tables specified in the following FROM clause.

<tables> is a list of one or more tables which contain the columns listed in the SELECT line.

<search conditions> is a list of conditions that can be used to limit or refine the selection of data that is returned.

Example 1

List all the information available about all the products that are made.

```
SELECT    *
FROM      PRODUCTS
```

The asterisk (*) in the SELECT statement indicates that all the columns in the tables listed in the following FROM clause should be returned in the order that they are defined in the PRODUCTS table. There is no necessity to specify the WHERE clause, because no conditions for selection are being made.

All the data in the PRODUCTS table is returned.

Example 2

Show the price and name (in that order) of all the products that are made.

```
SELECT P_PRICE, P_NAME
FROM PRODUCTS
```

The result is:	P_PRICE	P_NAME
	13.50	widget
	19.60	fergulator
	15.45	doofer
	8.05	dongle

Example 3

List the name and product number for each of the products that costs more than £10.00.

```
SELECT P_NAME, P_NUMBER
FROM PRODUCTS
WHERE P_PRICE > 10.0
```

The result is:	P_NAME	P_NUMBER
	widget	P1
	doofer	P3
	fergulator	P2

This example shows the use of a simple search condition that results in only rows with a P_PRICE value greater than 10 being returned from the database. Although P_PRICE is used in the query, it does not appear in the result. Often, the unit of value of numerical columns is implicit in the data. We know that the units refer to pounds sterling, because that was the way the database was designed. In an American context, the units would probably refer to U.S. dollars.

The “>” operator is one of the six scalar comparison operators which are used to compare data values:

operator	means
=	equal to
<	less than
>	greater than
<=	less than or equal to
>=	greater than or equal to
<>	not equal to

GLOSSARY

Comparison operator is used to compare two values.

Search condition is used to limit the data that is returned by a query.

SELECT Statement - Advanced

This section deals briefly with some of the more advanced features of the SELECT statement.

Complex Conditions

Search conditions can be grouped using the *logical operators* AND, OR and NOT.

Example 4

Find the name and price of all the products with a price between £10 and £15.

```
SELECT    P_NAME, P_PRICE
FROM      PRODUCTS
WHERE     P_PRICE >= 10.0 AND P_PRICE <= 15.0
```

The result is:	P_NUMBER	P_NAME	P_PRICE
	P1	widget	13.50

Although the query above is correct, the use of parentheses in the WHERE clause is good practice. This is because an implicit order is applied to the clause when it is processed, and the order in which a clause is processed can affect the result. In some cases, the implicit order may not be what is wanted; parentheses will always over-ride the implicit ordering.

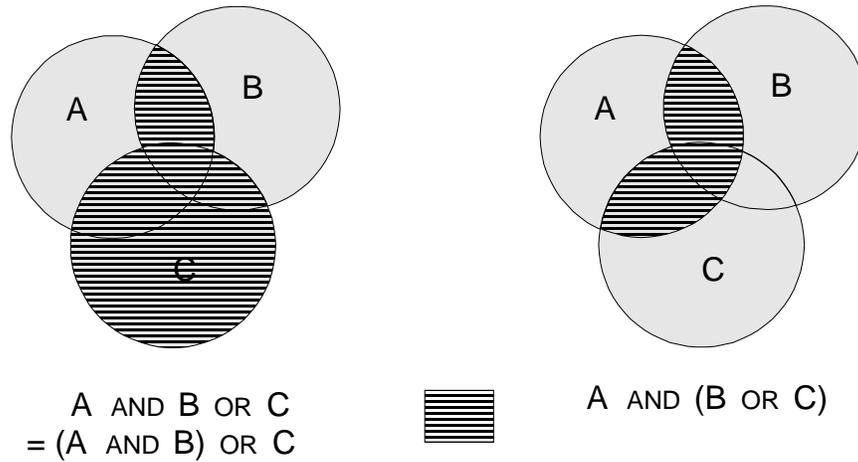
The WHERE clause would better be written:
 WHERE (P_PRICE >= 10.0) AND (P_PRICE <= 15.0)

Precedence Order

There is an implicit order in which logical operations are carried out. This is called the *precedence order*. The order is

```
NOT (highest)
AND
OR (lowest)
```

For example, the condition
 a OR b AND c
 means
 a OR (b AND c)



Where there is a combination of operators, parentheses (some people call these brackets) should be used to group conditions for two reasons.

- To over-ride the order of precedence, you must use parentheses.
- To aid clarity. Even if the logic defined by the precedence rules is what you want, using parentheses makes the logic easy to see.

Example 5

This example uses the ORDERS table, shown below.

ORDERS		
O_NUMBER	CUSTOMER	O_DATE
O1	Smith	29-01-1997
O2	Jones	29-01-1997
O3	Black	03-02-1997
O4	Smith	04-02-1997

Find any orders for the customer named 'Smith' dated the 3rd or 4th of February 1997.

```

SELECT      *
FROM        ORDERS
WHERE       CUSTOMER = 'Smith' AND
           (O_DATE = {d '1997-02-04'} OR O_DATE = {d '1997-02-03'})

```

Note: At this stage, do not worry about the date format in the WHERE clause; you will learn about this later.

The result is:

O_NUMBER	CUSTOMER	O_DATE
O4	Smith	04-02-1997

Without the use of parentheses in the WHERE clause, that is:

```
WHERE    CUSTOMER = 'Smith' AND
         O_DATE = {d '1997-02-04'} OR O_DATE = {d '1997-02-03' }
```

the returned data would be:

O_NUMBER	CUSTOMER	O_DATE
O3	Black	03-02-1997
O4	Smith	04-02-1997

If the dates had been swapped in the WHERE clause without parentheses, that is:

```
WHERE    CUSTOMER = 'Smith' AND
         O_DATE = {d '1997-02-03'} OR O_DATE = {d '1997-02-04' }
```

then in this case the result would contain order O4 only. Clearly, it is wise to use parentheses to ensure that the query uses the logic you want.

Precedence order also applies to arithmetic operators; again, use parentheses to explicitly state the order of evaluation.

GLOSSARY

Logical operator is used to group conditions.

Precedence order is the implicit order in which operations are carried out

Other Conditional Operators

SQL has other conditional operators, such as BETWEEN, LIKE and NULL. A few examples are presented here.

Example 6

Recall Example 4, “Find the name and price of all the products with a price between £10 and £15.” This could be written using the BETWEEN operator:

```
SELECT    P_NAME, P_PRICE
FROM      PRODUCTS
WHERE     P_PRICE BETWEEN 10.0 AND 15.0
```

This has exactly the same result as using the AND operator in Example 4, but for most people, it is easier to understand.

Example 7

Character data can be queried on the basis of its content.

List all the products which start with the letter “d” and display their price.

```
SELECT    P_NAME, P_PRICE
FROM      PRODUCTS
WHERE     P_NAME LIKE 'd%'
```

When used with the LIKE operator, the % character represents any string of zero or more characters.

In this example, we’ve introduced the character data type. Data types are used to classify data into groups. One of the reasons for doing this is to help prevent errors. If we know that a data item should be numeric, then if a user enters a character, the program can alert the user. In SQL, these are always enclosed in single quote marks, unlike numeric data types. The single quotes are not part of the data itself, so they do not appear in the output. Supported data types are listed in the appendix.

Joining tables

The abstract discussion of SQL showed how tables could be joined. Here we show how this is done using SQL.

Example 8

Generate (partial) data for an invoice. Each invoice for a single order (for example, O4) should contain the order number, product number, product name, quantity and total price.

```
SELECT    O_NUMBER, P_NUMBER, P_NAME, QUANTITY,
TOTAL_PRICE
FROM      PRODUCTS, OLINES
WHERE     (O_NUMBER = 'O4') AND
          (OLINES.P_NUMBER = PRODUCTS.P_NUMBER)
```

The result is:

O_NUMBER	P_NUMBER	P_NAME	QUANTITY	TOTAL_PRICE
O4	P2	fergulator	1	19.60
O4	P4	dongle	1	8.05

Eliminating Duplicates

Recall that two or more rows of a result set may be identical. These duplicate rows can be removed.

Example 9

Which orders contain either fergulators (P2) or dongles (P4), or both

```
SELECT DISTINCT O_NUMBER
FROM           OLINES
WHERE          (P_NUMBER='P2') OR (P_NUMBER='P4')
```

The result is:

O_NUMBER
O1
O3
O4

If the SELECT statement had not contained the DISTINCT keyword, the following data would have been returned:

O_NUMBER
O1
O3
O4
O4

Ordering Data

Although data in tables is not ordered, the result of a query can be ordered using the ORDER BY statement. The default ordering is ascending, but if you want to order the data in descending order, you can, as shown in this example.

Example 10

Find the names and price of all products and arrange them in descending order of price.

```
SELECT      P_NAME, P_PRICE
FROM        PRODUCTS
ORDER BY    P_PRICE DESC
```

The result is:	P_NAME	P_PRICE
	fergulator	19.60
	doofer	15.45
	widget	13.50
	dongle	8.05

DESC in the ORDER BY clause means descending. If you do not use it, then the default order is ascending (if desired, this can be specifically stated using ASC).

Aggregate Functions

There are a number of operations that can be performed, such as finding the average value of data, on a collection of values in one column of a table. These are called *aggregate functions*, and they are listed below.

function	meaning
COUNT DISTINCT	count the number of distinct values in the column
COUNT (*)	count the number of rows in a table
SUM	sum of the values in the column (numeric values only)
AVG	average of the values in the column (numeric values only)
MAX	largest value in the column
MIN	smallest value in the column

Example 11

How many products are there?

```
SELECT    COUNT (*)
FROM      PRODUCTS
```

The result is:

4

The table that is returned contains a single column with just a single row. The column does not have a name.

Example 12

What is the price of the cheapest product.

```
SELECT    MIN (P_PRICE)
FROM      PRODUCTS
```

The result is:

8.05

Dealing with dates

When specifying dates with ODBC, we need to use SQL with ODBC extensions. Previously, we stated that the format of dates in a database is application dependent. ODBC must convert this into some standard form. A date is specified in ODBC as follows:

{d 'yyyy-mm-dd' }

yyyy represents four digits corresponding to the year

mm represents two digits corresponding to the month

dd represents two digits corresponding to the day of the month

For example, 17th January 1995 would be represented as {d '1995-01-17' }

Example 13

Write a query to list all the orders for 4th February 1997.

```
SELECT      *
FROM        ORDERS
WHERE       O_DATE = {d '1997-02-04' }
```

The result is:

O_NUMBER	CUSTOMER	O_DATE
O4	Smith	04-02-1997

INSERT Statement

The INSERT statement is used to add a row to a table. Its basic structure is

```
INSERT INTO <table> [<column list>]
VALUES <list of values>
```

Example 14

The company has just started to produce a new product, a symmetrical screw, with a price of £0.27. It has been allocated a part number of 6.

```
INSERT INTO PRODUCTS
VALUES ('P6', 'symmetrical screw', 0.27)
```

The PRODUCTS table now contains:

PRODUCTS		
P_NUMBER	P_NAME	P_PRICE
P1	widget	13.50
P2	fergulator	19.60
P3	doofer	15.45
P4	dongle	8.05
P6	symmetrical screw	0.27

In this case, it is not necessary to specify the column names into which data is entered because the order in the VALUES clause matches the column order in the table. However, column names can be specified. The advantage of this is that there is then a partial check that the data is entered in the correct columns. If column names are specified, then the order of the data entered need not match that of the table (but it must match the order specified in the INSERT clause).

```
INSERT INTO PRODUCTS (P_NAME, P_NUMBER, P_PRICE)
VALUES ('symmetrical screw', 'P6', 0.27)
```

The result of this query would be exactly the same as that for the previous query.

UPDATE Statement

The UPDATE statement is used to modify existing rows in a table. Its basic structure is

```
UPDATE    <table>
SET       <column> = <value>
WHERE    <conditions>
```

Every row whose column(s) match the conditions is updated.

Example 15

Say that widgets have been reduced in price to £12.20. Update the PRODUCTS table to reflect this.

```
UPDATE    PRODUCTS
SET       P_PRICE = 12.20
WHERE    P_NAME = 'widget'
```

This table now contains

PRODUCTS		
P_NUMBER	P_NAME	P_PRICE
P1	widget	12.20
P2	fergulator	19.60
P3	doofer	15.45
P4	dongle	8.05
P6	symmetrical screw	0.27

Example 16

Say that all product prices are raised by 10%. Updating the table could be done like this.

```
UPDATE    PRODUCTS
SET       P_PRICE = P_PRICE * 0.1
```

This table now contains

PRODUCTS		
P_NUMBER	P_NAME	P_PRICE
P1	widget	13.42
P2	fergulator	8.86
P3	doofer	17.00
P4	dongle	21.56
P6	symmetrical screw	0.30

This result shows the effects of rounding. In our example, we rounded up. Exactly what happens is application dependent.

DELETE Statement

The DELETE statement is used to delete one or more rows from a table. Its basic structure is

```
DELETE
FROM   <table>
WHERE  <conditions>
```

Example 17

Say that doofers are no longer made - delete that product for the PRODUCTS table

```
DELETE
FROM   PRODUCTS
WHERE  P_NAME = 'doofer'
```

An alternative WHERE clause is: WHERE P_NUMBER=P3. This has exactly the same effect, since there is a 1:1 correspondence between P_NUMBER and P_NAME. The result in both cases is:

PRODUCTS		
P_NUMBER	P_NAME	P_PRICE
P1	widget	13.42
P2	fergulator	8.86
P4	dongle	21.56
P6	symmetrical screw	0.30

Just a single row has been deleted. The next example shows the deletion of many rows.

Example 18

Say that order number O4 had been entered by mistake. Delete this order from the OLINES table (in practice, an application running the SQL would probably have a method for dealing with incorrect entries. It's not a good idea to allow users to delete data at will).

```
DELETE
FROM   OLINES
WHERE  O_NUMBER = 4
```

This table now contains

OLINES			
O_NUMBER	P_NUMBER	QUANTITY	TOTAL_PRICE
O1	P1	1	13.50
O1	P4	2	16.10
O3	P4	1	8.05

Omitting the WHERE clause would have resulted in all the rows being deleted.

Summary

In this module you have learnt that

- ★ SQL (Structured Query Language) is a language used for relational databases
- ★ data is perceived to be stored in tables
- ★ tables are composed of rows and columns
- ★ indexes can be used to speed up data retrieval
- ★ an index can be unique or non-unique
- ★ the basic data manipulation statements are: SELECT, UPDATE, INSERT, DELETE
- ★ data can be combined from different tables using joins
 - Inner joins select records that have matching values in both tables
 - Outer joins select all records from one table; the other table may not have matching values
- ★ NULL represents a case where a value is unknown
- ★ there are six scalar comparison operators (<, >, <>, <=, >=, =) which are used to compare data values
- ★ there are three logical operators (AND, OR, NOT) which are used to combine conditions
- ★ parentheses
 - over-ride precedence order
 - allow you to see clearly the logic of a condition
- ★ aggregate functions are group operations, such as MAX, AVG, which work on collections of data values

SQL Review Questions

1. What type of databases is SQL designed for?
2. What is the purpose of a join?
3. What is the difference between an inner join and an outer join?
4. What do the terms 'left' and 'right' mean in the context of outer joins?
5. What three operators are used to combine conditions in SQL?
6. Is it always necessary to use parentheses when using an AND or OR operator in SQL?
7. The result of an SQL query may contain duplicate values. How can these duplicates be removed?
8. How many comparison operators are there? List them.

SQL Review Answers

1. What type of databases is SQL designed for?

Relational databases.

2. What is the purpose of a join?

To obtain information from more than one table

3. What is the difference between an inner join and an outer join?

An inner join returns data for the join column only if it occurs in both tables
An outer join returns data from the join column whether or not it appears in both tables.

4. What do the terms 'left' and 'right' mean in the context of outer joins?

They refer to the relative position of the table in the join condition. The first table is the left table, and the second table is the right table.

5. What three operators are used to combine conditions in SQL?

AND, OR, NOT.

6. Is it always necessary to use parentheses when using an AND or OR operator in SQL?

No, but to ensure that the result is based on the condition you thought you had defined, and for clarity, it's a good idea always to use them.

7. The result of an SQL query may contain duplicate values. How can these duplicates be removed?

Use SELECT DISTINCT.

8. How many scalar comparison operators are there? List them.

There are 6:

= (equal), < (less than), > (greater than), <= (less than or equal),
>= (greater than or equal), <> (not equal).

Further Exercises

For your convenience when answering the questions which require the writing of SQL code, the three tables are replicated here.

PRODUCTS		
P_NUMBER	P_NAME	P_PRICE
P1	widget	13.50
P2	fergulator	19.60
P3	doofer	15.45
P4	dongle	8.05

ORDERS		
O_NUMBER	CUSTOMER	O_DATE
O1	Smith	29-01-1997
O2	Jones	29-01-1997
O3	Black	03-02-1997
O4	Smith	04-02-1997

OLINES			
O_NUMBER	P_NUMBER	QUANTITY	TOTAL_PRICE
O1	P1	1	13.50
O1	P4	2	16.10
O3	P4	1	8.05
O4	P2	1	19.60
O4	P4	1	8.05

1. Write a query to show all the orders.
2. Write a query to show all the order numbers and the customer who sent the order.
3. Write a query to list all the orders for 1st January 1997 - show the order number and the customer only.

4. Using SQL, what is the easiest way to find out how many orders there are?

5. Use SQL to calculate the total value of all the orders.

6. Find the average price of all the products by writing a simple query.

7. Write a query to list the name and product number of all the products despatched on 29th January 1997.

8. Write a query to list all the customers in alphabetic order.

9. Write a query to show the name of all products and their price; list this information in descending price order.

Suggested Answers to Exercises

1. Write a query to show all the orders.

```
SELECT  *
FROM    ORDERS
```

2. Write a query to show all the order numbers and the customer who sent the order.

```
SELECT  O_NUMBER, CUSTOMER
FROM    ORDERS
```

3. Write a query to list all the orders for 1st January 1997 - show the order number and the customer only.

```
SELECT  O_NUMBER, CUSTOMER
FROM    ORDERS
WHERE   O_DATE = {d '1997-01-01' }
```

Here, we must use an ODBC extension to SQL to specify the date.

4. Using SQL, what is the easiest way to find out how many orders there are?

```
SELECT  COUNT (*)
FROM    ORDERS
```

5. Use SQL to calculate the total value of all the orders.

```
SELECT  SUM (TOTAL_PRICE)
FROM    OLINES
```

6. Find the average price of all the products by writing a simple query.

```
SELECT  AVG (P_PRICE)
FROM    PRODUCTS
```

7. Write a query to list the name and product number of all the products despatched on 29th January 1997.

```
SELECT    P_NAME, P_NUMBER
FROM      OLINES, PRODUCTS
WHERE     O_DATE = {d '1997-01-29'} AND
          OLINES.P_NUMBER = PRODUCTS.P_NUMBER
```

Here, we must use an ODBC extension to SQL to specify the date.

8. Write a query to list all the customers in alphabetic order.

```
SELECT DISTINCT CUSTOMER
FROM      ORDERS
ORDER BY  CUSTOMER
```

Note: in practice, this query would be posed against a table dedicated to customer information.

9. Write a query to show the name of all products and their price; list this information in descending price order.

```
SELECT    P_NAME, P_PRICE
FROM      PRODUCTS
ORDER BY  P_PRICE DESC
```

Additional Notes

The Easysoft Catalog

The Easysoft Catalog can be considered as a set of tables which themselves contain the information needed to describe the non-relational server data in relational terms, so that it may be viewed by ODBC-compliant applications (which interact with data using SQL). These tables can be queried by SQL just as any other table can.

Supported SQL

The following SQL statements are supported by the Easysoft software. Words enclosed in angle brackets (<>) should be replaced by any valid SQL syntax.

INSERT

```
INSERT INTO <table> [( <column identifiers>)] VALUES (<values>)
```

SELECT

```
SELECT [ALL | DISTINCT] <select-list>  
FROM <table>  
[WHERE <search condition>]  
[GROUP BY <column names>]  
[HAVING <search condition>]  
[ORDER BY <sort specification>]
```

DELETE SEARCHED

```
DELETE FROM <table> [WHERE <search condition>]
```

UPDATE SEARCHED

```
UPDATE <table>  
SET <column identifiers> = <expression>  
[WHERE <search condition>]
```

Supported Set (Aggregate) Functions

COUNT(*), COUNT, MAX, MIN, SUM, AVG

Reserved Words

SQL contains reserved words (a word that is part of the language itself), which should not be used when writing SQL commands unless they are expressed as character literals (i.e. enclosed between single quote marks). Examples of reserved words are SELECT, UPDATE, BETWEEN and SET. These words are defined by the standards organisations that define the SQL language, and are listed below.

ABSOLUTE	COLLATION_NAME	DICTIONARY
ACTION	COLLATION_SCHEMA	DISCONNECT
ADA	COLUMN	DISTINCT
ADD	COLUMN_NAME	DOMAIN
AFTER	COMMAND_FUNCTION	DOUBLE
ALIAS	COMMIT	DROP
ALL	COMMITTED	DYNAMIC_FUNCTION
ALLOCATE	COMPLETION	EACH
ALTER	CONDITION_NUMBER	ELSE
AND	CONNECT	ELSEIF
ANY	CONNECTION	END
ARE	CONNECTION_NAME	END-EXEC
AS	CONSTRAINT	EQUALS
ASC	CONSTRAINT_CATALOG	ESCAPE
ASSERTION	CONSTRAINT_NAME	EXCEPT
ASYN	CONSTRAINT_SCHEMA	EXCEPTION
AT	CONSTRAINTS	EXEC
AUTHORIZATION	CONTINUE	EXECUTE
AVG	CONVERT	EXISTS
BEFORE	CORRESPONDING	EXTERNAL
BEGIN	COUNT	EXTRACT
BETWEEN	CREATE	FALSE
BIT	CROSS	FETCH
BIT_LENGTH	CURRENT	FIRST
BOOLEAN	CURRENT_DATE	FLOAT
BOTH	CURRENT_TIME	FOR
BREADTH	CURRENT_TIMESTAMP	FOREIGN
BY	CURRENT_USER	FORTRAN
C	CURSOR	FOUND
CALL	CURSOR_NAME	FROM
CASCADE	CYCLE	FULL
CASCADE	DATA	GENERAL
CASE	DATE	GET
CAST	DATETIME_INTERVAL_CODE	GLOBAL
CATALOG	DATETIME_INTERVAL_PRECISIO	GO
CATALOG_NAME	N	GOTO
CHAR	DAY	GRANT
CHAR_LENGTH	DEALLOCATE	GROUP
CHARACTER	DEC	HAVING
CHARACTER_LENGTH	DECIMAL	HO
CHARACTER_SET_NAME	DECLARE	IDENTITY
CHARACTER_SET_SCHEMA	DEFAULT	IF
CHECK	DEFERRABLE	IGNORE
CLASS_ORIGIN	DEFERRED	IMMEDIATE
CLOSE	DELETE	IN
COALESCE	DEPTH	INCLUDE
COBOL	DESC	INDEX
COLLATE	DESCRIBE	INDICATOR
COLLATION	DESCRIPTOR	INITIALLY
COLLATION_CATALOG	DIAGNOSTICS	INNER

INPUT	ORDER	SMALLINT
INSENSITIVE	OTHERS	SOME
INSERT	OUTER	SPACE
INT	OUTPUT	SQL
INTEGER	OVERLAPS	SQLCA
INTERSECT	PAD	SQLCODE
INTERVAL	PARAMETERS	SQLERROR
INTO	PARTIAL	SQLEXCEPTION
IS	PASCAL	SQLSTATE
ISOLATION	PENDANT	SQLWARNING
JOIN	PLI	STRUCTURE
KEY	POSITION	SUBCLASS_ORIGIN
LANGUAGE	PRECISION	SUBSTRING
LAST	PREORDER	SUM
LEADING	PREPARE	SYSTEM
LEAVE	PRESERVE	SYSTEM_USER
LEFT	PRIMARY	TABLE
LENGTH	PRIOR	TABLE_NAME
LESS	PRIVATE	TEMPORARY
LEVEL	PRIVILEGES	TEST
LIKE	PROCEDURE	THEN
LIMIT	PROTECTED	THERE
LOCAL	PUBLIC	TIME
LOOP	READ	TIMESTAMP
LOWER	REAL	TIMEZONE_HOUR
MATCH	RECURSIVE	TIMEZONE_MINUTE
MAX	REF	TO
MESSAGE_LENGTH	REFERENCES	TRACEPOINT
MESSAGE_OCTET_LENGTH	REFERENCING	TRAILING
MESSAGE_TEXT	RELATIVE	TRANSACTION
MIN	REPEATABLE	TRANSLATE
MINUTE	REPLACE	TRANSLATION
MODIFY	RESIGNAL	TRIGGER
MODULE	RESTRICT	TRIM
MONTH	RETURN	TRUE
MORE	RETURNED_LENGTH	TYPE
MUMPS	RETURNED_OCTET_LENGTH	UNCOMMITTED
NAME	RETURNED_SQLSTATE	UNDER
NAMES	RETURNS	UNION
NATIONAL	REVOKE	UNIQUE
NATURAL	RIGHT	UNKNOWN
NCHAR	ROLE	UNNAMED
NEW	ROLLBACK	UPDATE
NEXT	ROUTINE	UPPER
NO	ROW	USAGE
NONE	ROW_COUNT	USER
NOT	ROWS	USING
NOTRACEPOINT	SAVEPOINT	VALUE
NULL	SCALE	VALUES
NULLABLE	SCHEMA	VARCHAR
NULLIF	SCHEMA_NAME	VARIABLE
NUMBER	SCROLL	VARYING
NUMERIC	SEARCH	VIEW
OBJECT	SECOND	VIRTUAL
OCTET_LENGTH	SECTION	VISIBLE
OF	SELECT	WAIT
OFF	SENSITIVE	WHEN
OID	SEQUENCE	WHENEVER
OLD	SERIALIZABLE	WHERE
ON	SERVER_NAME	WHILE
ONLY	SESSION	WITH
OPEN	SESSION_USER	WITHOUT
OPERATION	SET	WORK
OPERATORS	SIGNAL	WRITE
OPTION	SIMILAR	YEAR
OR	SIZE	ZONE

Bibliography



This bibliography lists a few sources of information relevant to information in this module. Entries are listed alphabetically by title, except that initial articles (i.e. 'A' and 'The') are ignored for the purposes of ordering.

Database Language SQL, Document ANSI X3.135-1986, American National Standards Institute, 1986.

This is the original SQL standard definition. It is a heavy-duty reference.

Database Language SQL with Integrity Enhancement, Document ISO/IEC 9075: 1989(E), International Organisation for Standardisation, 1989.

This document defines additions to the original SQL standard which deal with the integrity of data. It is a heavy-duty reference.

Data Management: Structured Query Language (SQL), Version 2. X/Open CAE Specification. X/Open Document Number:C449. IISBN: 1-85912-151-9.

Database Systems Engineering, R. P. Whittington, Clarendon Press, Oxford, 1988. ISBN 0-19-859666-9

Fundamentals of Database Systems, R. Elmasri & S. B. Navathe, Benjamin/Cummings Publishing Co. Inc., 1989. ISBN 0-201053090-2

An Introduction to Database Systems, volume 1, C. J. Date, Addison-Wesley Publishing Company, Inc., 1990. ISBN 0-201-52878-9

One of the most popular books on databases and SQL (and it covers more than just relational systems). Although it calls itself an introduction, at nearly 1000 pages it's not entirely a trivial read. Nevertheless, it starts simply, and contains numerous examples and exercises (and answers).

Introduction to SQL, R. F. van der Lans, Addison-Wesley Publishing Company, Inc., 1988. ISBN 0-201-17521-5

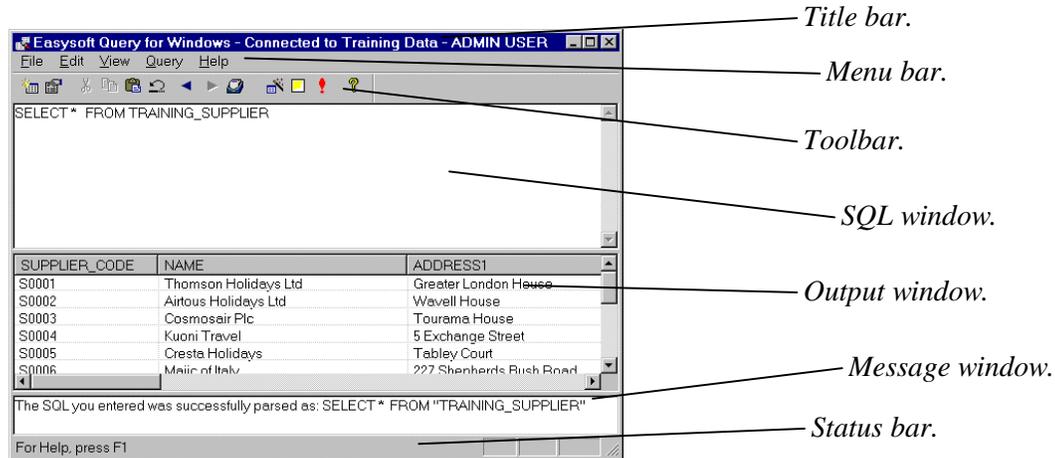
Rather an old book, so some of the material is dated. However, the principles remain sound. Packed with worked examples, exercises (and answers).

A Relational Model of Data for Large Shared Data Banks, E. F. Codd, Communications of the ACM, volume 13, number 6 (June 1970).

A technical paper, not suitable for the beginner.

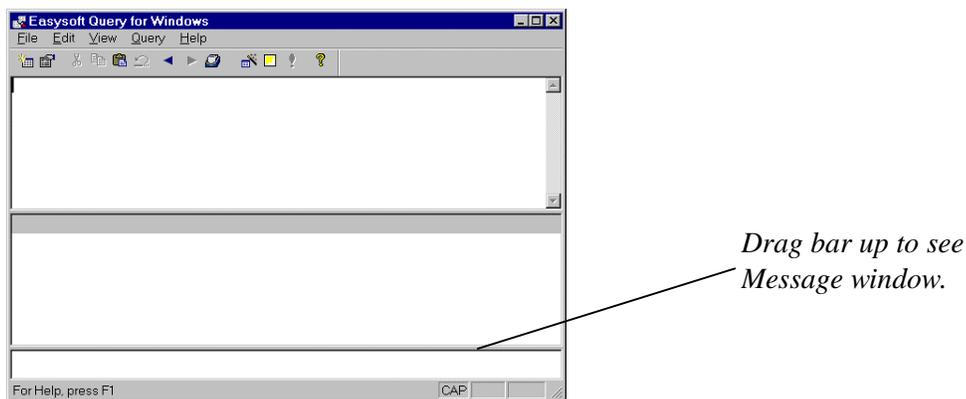
Easysoft Query for Windows

The figure below shows the Easysoft Query for Windows main dialog box.

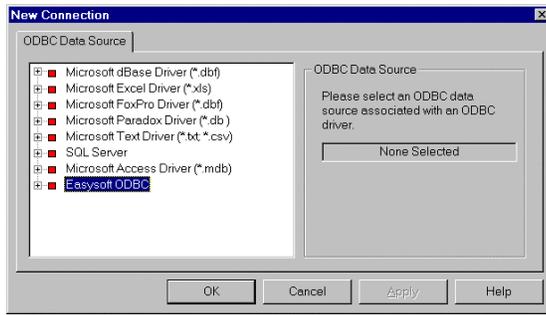


Start and Connect

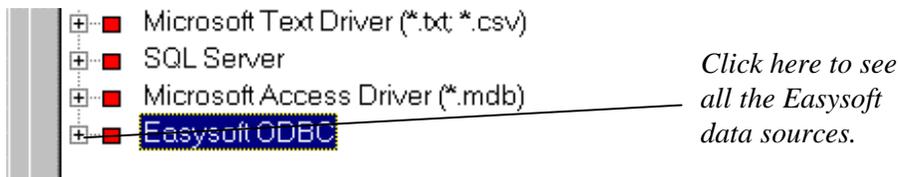
1. Start Easysoft Query for Windows by clicking on the shortcut that has been created:



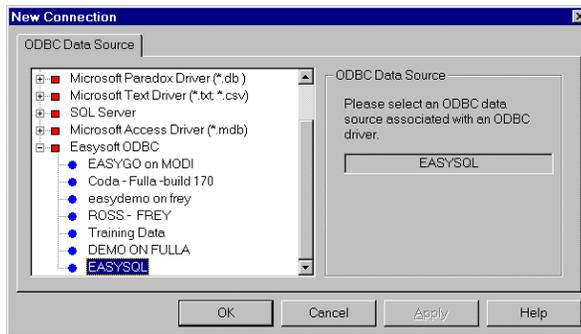
2. Connect to a data source by selecting the **File, New Connection** menu sequence. The New Connection dialog box appears.



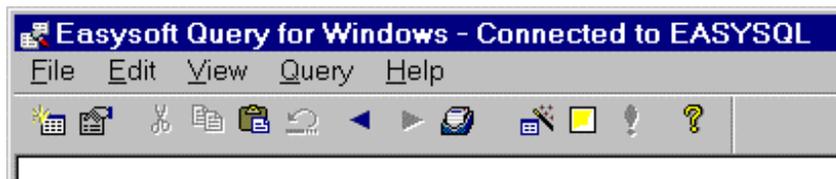
- The data sources are organised by driver type. To see all the data sources that use the Easysoft driver, click on the plus sign at the left of Easysoft ODBC.



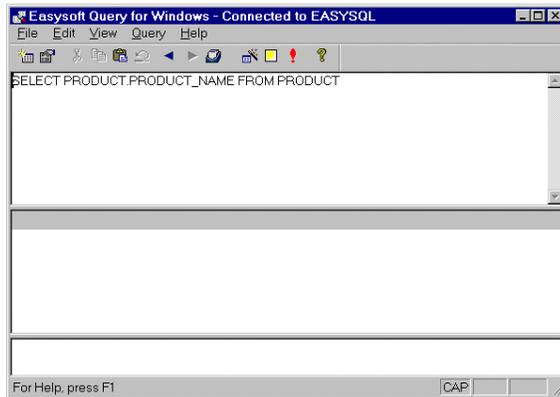
- A list of available data sources is displayed. Highlight EASYSQL and click **OK**.



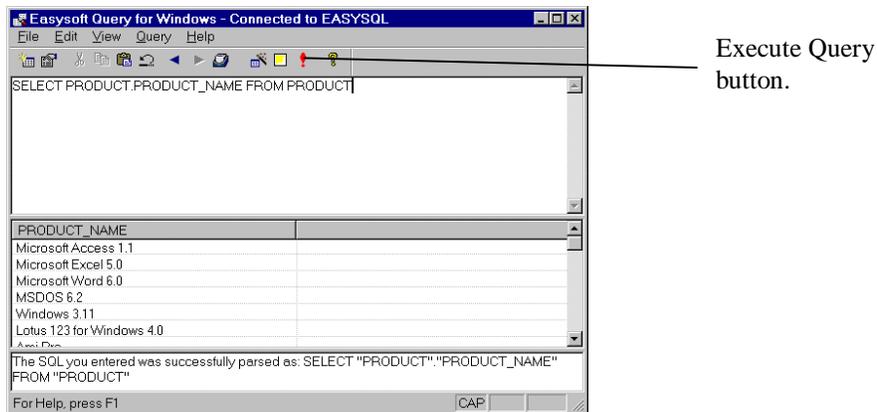
- The title bar now shows which data source you are connected to.



6. Now that you have connected to a data source, you can send SQL to it. Enter this test query in the SQL window.



7.  Click the Execute Query button (or select **Query, Execute** from the menu options or press F5) to run the query. The query is cleared from the window in preparation for the following query, and the result appears in the output window.

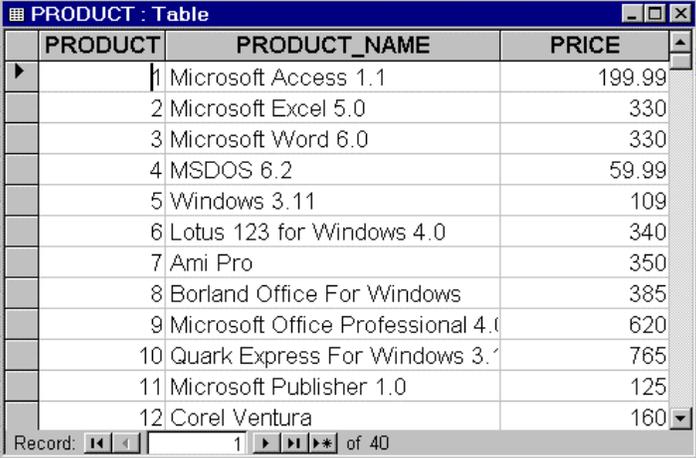


8. To close Easysoft Query for Windows select **Exit** from the **File** menu. (Do not close Easysoft Query for Windows at this stage).

The Example Database

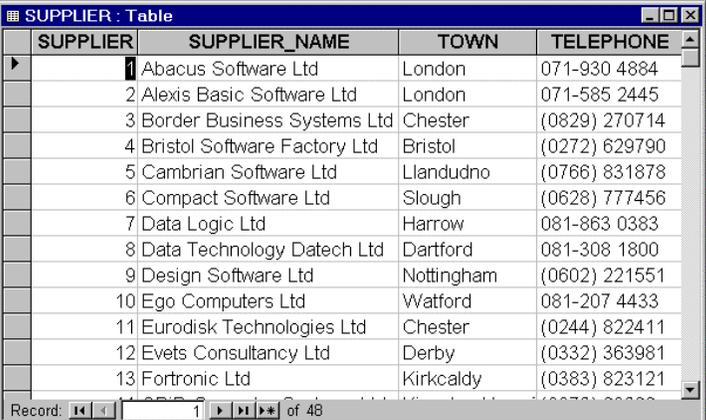
There are three tables in this database, PRODUCT, SUPPLIER and SUPPLIER_PRODUCT. The columns in the tables, and some of the data for each of the tables is shown in the screen shots below.

PRODUCT table



PRODUCT	PRODUCT_NAME	PRICE
1	Microsoft Access 1.1	199.99
2	Microsoft Excel 5.0	330
3	Microsoft Word 6.0	330
4	MSDOS 6.2	59.99
5	Windows 3.11	109
6	Lotus 123 for Windows 4.0	340
7	Ami Pro	350
8	Borland Office For Windows	385
9	Microsoft Office Professional 4.0	620
10	Quark Express For Windows 3.1	765
11	Microsoft Publisher 1.0	125
12	Corel Ventura	160

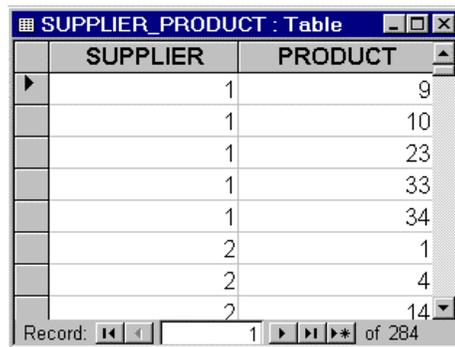
SUPPLIER table



SUPPLIER	SUPPLIER_NAME	TOWN	TELEPHONE
1	Abacus Software Ltd	London	071-930 4884
2	Alexis Basic Software Ltd	London	071-585 2445
3	Border Business Systems Ltd	Chester	(0829) 270714
4	Bristol Software Factory Ltd	Bristol	(0272) 629790
5	Cambrian Software Ltd	Llandudno	(0766) 831878
6	Compact Software Ltd	Slough	(0628) 777456
7	Data Logic Ltd	Harrow	081-863 0383
8	Data Technology Datech Ltd	Dartford	081-308 1800
9	Design Software Ltd	Nottingham	(0602) 221551
10	Ego Computers Ltd	Watford	081-207 4433
11	Eurodisk Technologies Ltd	Chester	(0244) 822411
12	Evets Consultancy Ltd	Derby	(0332) 363981
13	Fortronic Ltd	Kirkcaldy	(0383) 823121

SUPPLIER_PRODUCT table

This table links Products to Suppliers.



SUPPLIER	PRODUCT
1	9
1	10
1	23
1	33
1	34
2	1
2	4
2	4
2	14

Exercises

Write the SQL in the space below the questions, then use Easysoft Query for Windows to send the query to the server. The database is described in the previous section.

1. Write a query to show all the products.
2. Write a query to show all the suppliers and their telephone numbers.
3. List all products costing more than £300.00. Also include the price of the product.
4. How many products are there?
5. How many products cost over £300.00?
6. List all Microsoft products.

Suggested Answers

1. Write a query to show all the products.

```
SELECT *  
FROM PRODUCT
```

2. Write a query to show all the suppliers and their telephone numbers.

```
SELECT SUPPLIER_NAME, TELEPHONE  
FROM SUPPLIER
```

3. List all products costing more than £300.00. Also include the price of the product.

```
SELECT PRODUCT_NAME, PRICE  
FROM PRODUCT  
WHERE PRICE > 300
```

4. How many products are there?

```
SELECT COUNT(*)  
FROM PRODUCT
```

5. How many products cost over £300.00?

```
SELECT COUNT(PRODUCT_NAME)  
FROM PRODUCT  
WHERE PRICE > 300
```

6. List all Microsoft products.

```
SELECT PRODUCT_NAME  
FROM PRODUCT  
WHERE PRODUCT_NAME like 'Microsoft%'
```

7. For each supplier, list the products that are supplied.

```
SELECT SUPPLIER_NAME, PRODUCT_NAME
FROM SUPPLIER, SUPPLIER_PRODUCT, PRODUCT
WHERE SUPPLIER.SUPPLIER = SUPPLIER_PRODUCT.SUPPLIER AND
SUPPLIER_PRODUCT.PRODUCT = PRODUCT.PRODUCT
```

8. List all the products supplied by the supplier named "Compact Software Ltd". Include the supplier's name in the result.

```
SELECT SUPPLIER_NAME, PRODUCT_NAME
FROM SUPPLIER, SUPPLIER_PRODUCT, PRODUCT
WHERE SUPPLIER.SUPPLIER_NAME = 'Compact Software Ltd' AND
SUPPLIER.SUPPLIER = SUPPLIER_PRODUCT.SUPPLIER AND
SUPPLIER_PRODUCT.PRODUCT = PRODUCT.PRODUCT
```

9. List all suppliers of Microsoft Word. (Hint: use PRODUCT table first)

```
SELECT PRODUCT_NAME, SUPPLIER_NAME
FROM PRODUCT, SUPPLIER_PRODUCT, SUPPLIER
WHERE PRODUCT.PRODUCT_NAME like 'Microsoft Word%' AND
SUPPLIER_PRODUCT.PRODUCT = PRODUCT.PRODUCT AND
SUPPLIER.SUPPLIER = SUPPLIER_PRODUCT.SUPPLIER
```

3. The Easysoft System

This module presents the global architecture of Easysoft ODBC for CODA.

In this module you will learn about

- ✦ the architecture of Easysoft ODBC for CODA
- ✦ the relationship between data sources, catalogs, databases and physical files
- ✦ the mapping between files and tables
- ✦ the overall procedure for using Easysoft software

At the end of the module there are some review questions.

Contents

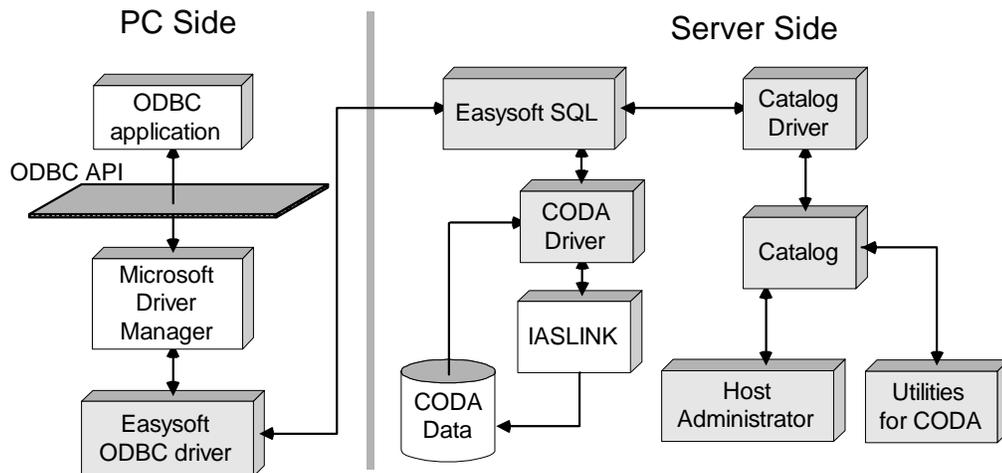
Easysoft Architecture _____	3-2
Overall Process of using Easysoft _____	3-3
Data Sources and Catalogs _____	3-4
Easysoft Catalog Structure _____	3-5
Summary _____	3-6
Review Questions _____	3-7
Review Answers _____	3-8

Easysoft Architecture

Easysoft ODBC for CODA is used to connect ODBC-compliant applications (such as Lotus 1-2-3 and Microsoft Excel) to CODA data files which reside on remote file Servers (on the OpenVMS platform). Since CODA data is not relational, one of the functions of the software, in addition to its ODBC functionality, is to make CODA data appear relational, so that ODBC-compliant applications can read from and write to CODA data.

Easysoft ODBC has a two-tier architecture in that the data access software resides on the Server and the driver passes function calls to this. The figure below shows the logical architecture of the components of the system and their relationship within the overall ODBC architecture.

Easysoft software architecture for CODA



Easysoft Client component. This resides on the PC and contains the Microsoft ODBC Administrator / Driver Manager and the Easysoft ODBC driver. The Microsoft ODBC Administrator is used to install the Easysoft ODBC driver on the PC and to configure data sources.

Easysoft Server component. This contains a number of sub-components:

Easysoft SQL - deals with processing of SQL statements.

Host Administrator - runs on the host machine and is used to manage catalogs at a high level - primarily their creation. It is also used to administer licensing functions.

Catalog - used to store information about the Server files.

Catalog Driver - the mechanism for obtaining information on the structure of the data.

CODA Driver - used to access CODA data on the Server (this should not be confused with the Easysoft ODBC driver which resides on the PC).

CODA Utilities - used as an aid to accessing data. They are:

Codacat - populates the catalog

Codouser - informs the catalog about CODA users

Catuser - informs the catalog of the catalog users

Codaxref - used to return data in N/S/L3 order

These are described in more detail in module 5, "Easysoft Administration".

IASLINK. This is not part of the Easysoft software. It is part of the CODA system. It is an API, the purpose of which is to allow user programs to access CODA data without using the IAS interface. Easysoft ODBC for CODA uses IASLINK when updating CODA data, and thus there is no possibility of inadvertently corrupting the data. If IASLINK permits an operation, then that is a valid operation for the user to perform in CODA. If IASLINK does not permit an operation, then the user would not be able to perform the operation within CODA-IAS.

Overall Process of using Easysoft

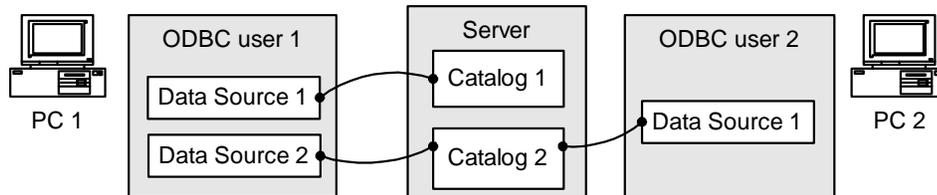
How exactly is the Easysoft system told of the existence and structure of the data files on the Server, and what do you do to enable an ODBC-compliant application to access this data? The recommended procedure is:

1. Install the Easysoft Server component on the Server. Part of this includes the CODA setup routine which inserts information into the catalog.
2. License the software.
3. Install the Easysoft Client Component on the PC and then define one or more data sources associated with a catalog.
4. Use an ODBC-compliant application to access the data.

Data Sources and Catalogs

The figure below indicates the relationship between data sources and catalogs.

Relationships between data sources and catalogs

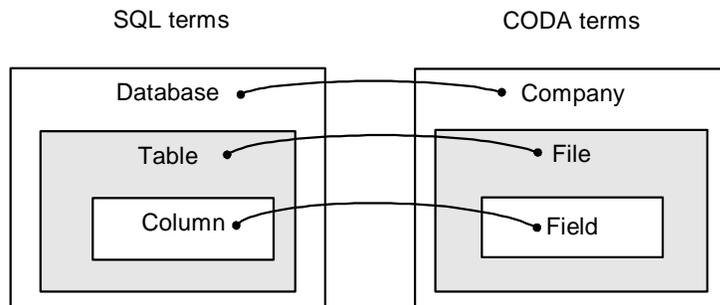


Data sources are defined at the PC level using the Microsoft ODBC Administrator. Each data source on a PC has a unique name. The data source defines which Server is to be accessed, the ODBC driver to use when accessing the Server data, the catalog associated with the data, network transport information and the user of the Server.

To ensure ease of upgrading it is advisable to keep data source names on different PCs identical if they are to be used to access the same catalog.

The correspondences between SQL and CODA are shown in the diagram below.

Correspondences between SQL terms and CODA terms



SQL tables map to files and SQL columns map to fields. A database is a uniquely named collection of SQL tables, and before the mapping between files and tables can be made, there must be at least one defined database. Associated with tables and databases are privileges which allow access rights to be defined. The function of the catalog is to hold all this mapping information.

All the mappings that are needed so that CODA data can be accessed in terms of SQL are carried out entirely automatically by Easysoft ODBC for CODA. Each CODA company that you tell the catalog about corresponds to an SQL database, which is given the same name as the company.

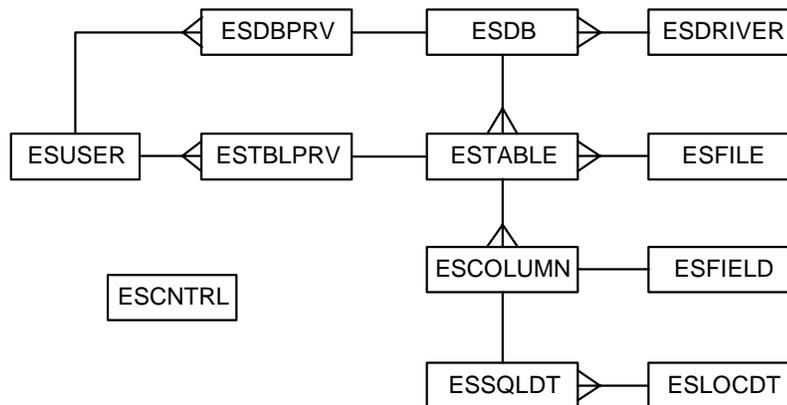
Easysoft Catalog Structure

Easysoft SQL needs a set of tables in order to manipulate the CODA data which resides on the Server. Collectively, these files are known as the Easysoft Catalog. Data associated with a data source is defined in these Easysoft Catalog files, which hold, for example information on the location and structure of data files. A catalog can contain information on many different data sources.

You can have as many catalogs as you wish, but each catalog must reside in a separate directory. In most cases it is usual to have just one catalog to deal with all the data.

The diagram below shows the Easysoft catalog tables. (A crow's foot indicates a one-to-many relationship).

Easysoft Catalog



Information about users and their privileges is contained in these tables: ESUSER, ESDBPRV, ESTBLPRV.

Information about data access is contained in these tables: ESDB, ESTABLE, ESCOLUMN.

Mapping information is contained in these tables: ESDRIVER, ESFILE, ESFIELD, ESLOCDT.

Summary

In this module you have learnt

- ✦ the Easysoft Client component contains the ODBC driver and resides on the PC
- ✦ a single PC can have many different data sources defined on it
- ✦ the Easysoft Server component consists of
 - Easysoft SQL, which processes SQL statements
 - the catalog, which contains a description of the CODA data
 - CODA utilities, an aid to setting up and using the software
 - the Host Administrator, used mainly for licensing and catalogs
 - the CODA driver, which accesses CODA data
- ✦ updates to CODA data go through IASLINK, thus ensuring data is not corrupted

Review Questions

1. Easysoft ODBC for CODA is
 - a) single tier
 - b) two-tier
 - c) three-tier

2. The data access software resides on
 - a) the PC
 - b) the server

3. Name the functions of the Easysoft Host Administrator

4. What is the function of the Easysoft catalog?

5. Where is the catalog stored?

6. Data sources are defined on
 - a) the PC
 - b) the server
 - c) either PC or server
 - d) both PC and server

7. Name the SQL equivalents of these CODA terms:
 - a) File
 - b) Field
 - c) Company

8. What is the overall process of installing and using Easysoft ODBC for CODA?

Review Answers

1. Easysoft ODBC for CODA is

Two-tier

2. The data access software resides on

The server

3. Name the functions of the Easysoft Host Administrator

Create catalog, manage catalog, deal with licensing

4. What is the function of the Easysoft catalog?

Store information about CODA files

5. Where is the catalog stored?

On the server

6. Data sources are defined on

The PC

7. Name the SQL equivalents of these CODA terms:

File	Table
Field	Column
Company	Database

8. What is the overall process of installing and using Easysoft ODBC for CODA?

Install on the server
License the software
Install on the PC
Define a data source
Access the data using ODBC-compliant application

4. Microsoft ODBC Administrator

Before you can use an ODBC-compliant application to connect to data on the Server, you must set up a data source. Later, when you use the application, you will connect to the server data using this data source. To set up a data source, use the Microsoft ODBC Administrator.

In this module you will learn

- ✦ how to create a data source using the Easysoft ODBC Setup dialog box

Contents

Purpose _____	4-2
Administrator _____	4-3
Set up a Data Source _____	4-4
Settings _____	4-6
Additional Notes _____	4-7
Summary _____	4-9
Supplement: troubleshooting _____	4-10

Purpose

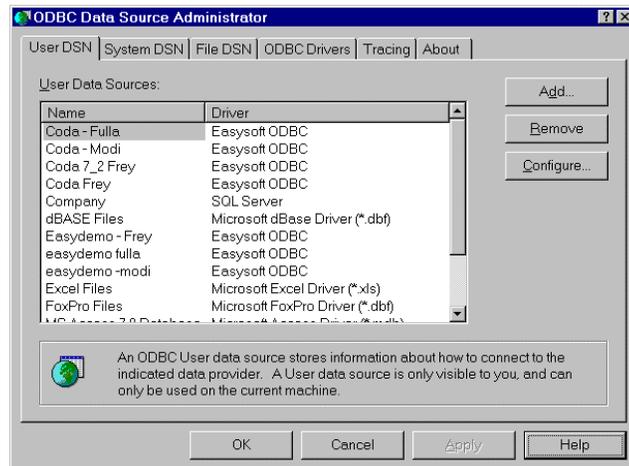
The Microsoft ODBC Administrator is used for the creation, deletion and modification of data sources in conjunction with driver-specific dialog boxes which are provided by the creators of the drivers. The creation of a data source is described in this module. The Administrator is also used for diagnostic purposes, namely the tracing of ODBC function calls (discussed in the appendix entitled, “Troubleshooting”).

This module gives step-by-step instructions for setting up (System) data sources, first using Microsoft ODBC Administrator.

Administrator

Version 3.0 is used with 32 bit drivers only. The layout of previous versions is significantly different, although the functionality is similar to that described here, namely to add, delete and configure data sources, and to set trace options.

Microsoft ODBC Data Sources Administrator



The following tabs are available on the Microsoft ODBC Data Source Administrator, version 3.0:

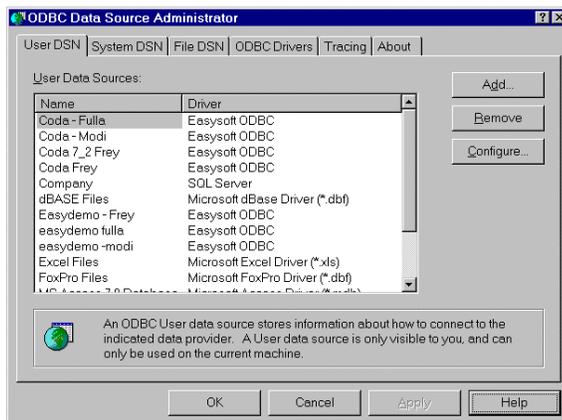
Tab	Function
User DSN	Add, remove and configure User data sources. A User data source is available to a specific user on a machine (compare System DSN).
System DSN	Add, remove and configure System data sources. A System data source is available to all users on a machine (compare User DSN).
File DSN	These are file-based data sources that can be shared between all users that have the same drivers installed. This option is not applicable to Easysoft ODBC and was not available in previous versions of the Microsoft ODBC Administrator.
ODBC Drivers	Displays information about the installed drivers (see "Drivers Tab").
Tracing	Set trace options for the ODBC Driver Manager. Described in Appendix G, "Troubleshooting".

About Information about the core components of the Microsoft ODBC. This About tab has no relationship to the About dialog box in version 2.5 of the Administrator, which described information about the installed drivers. For information about drivers, look under the ODBC Drivers tab.

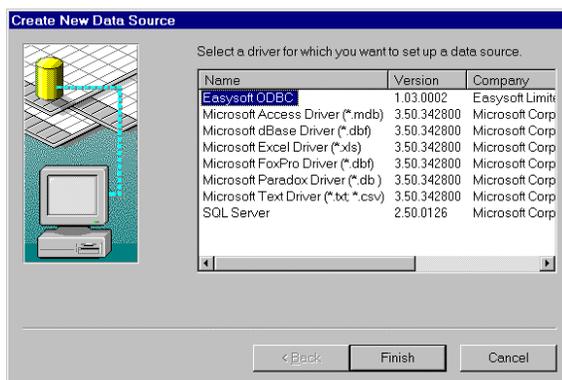
Set up a Data Source

The operation of the User data source dialog box and the System data source dialog box is identical. A System data source is created in this example.

1. Start the Microsoft ODBC Administrator version 3.0 by clicking on the ODBC icon (found in the Control Panel).
2. Select the **System DSN** tab.



3. Click the **Add...** button. The Create New Data Source dialog box is displayed.



4. Highlight **Easysoft ODBC** and click **Finish**. The Easysoft ODBC Setup dialog box appears. Complete this using the information below.

Enter this information

Reason / explanation

Name

TRAINING DATA

The name of a data source. This name appears in the ODBC connection dialog box when you connect to an ODBC source.

Naming restrictions:

- minimum number of characters = 1
- maximum number of characters = 32
- first letter must be alphabetic
- not valid: [] { } () ? * = ! @ , ;

Description

Type any description.

A more descriptive name for the data source (optional).

Catalog

This will be given on the day of the course.

The directory where the Easysoft Catalog resides on the Server.

Transport

Select **REXEC**.

The network transport to be used. You are presented with the options available to your particular PC.

Server

MODI

The name of the Server on which the source data resides.

Remote Service /Command/ Object

This will be given on the day of the course.

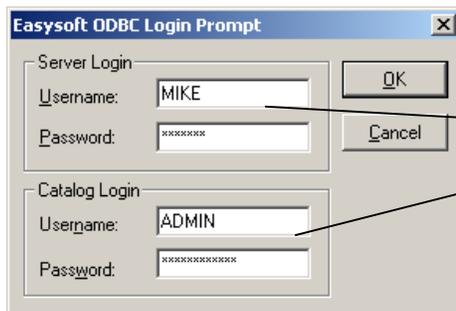
Select the name of the service to connect to on the Server. Remote Service and Remote Object map to a command to run the software. Remote Command is the command to run in order to access the software.

Usernames, Passwords

Leave blank.

If no user and password information is stored with a data source, each user has to enter this information in the Easysoft ODBC Login Prompt (see next figure) each time they connect.

5. For the purposes of this exercise, the **Settings**, **Logging** and **Capture** buttons are not needed. **Settings** is discussed in “Settings” on page 4-6, **Logging** is discussed in the appendix entitled “Troubleshooting”. (**Capture** is currently disabled).
6. Press the **Test** button. This validates the inserted information. The Easysoft ODBC Login Prompt appears, and the entry fields are initially blank. Information on how to switch off the login prompt is in the “Settings” section.



Server and catalog usernames and passwords can be different.

7. Enter the username and password information (you will be given these on the day of the course) and click the **OK** button. If the test is successful, a dialog box appears stating this; click **OK** to continue. If the test is unsuccessful a message is generated. After the test is successful, click on **OK** to save the new data source and return to the Data Sources dialog box. Leave the Data Sources dialog box by selecting **Close**.

Server Login Username The user name used to connect to the Server. Case is ignored.

Server Login Password The password associated with the Server login username. Case is ignored.

Catalog Login Username The user name that provides access to the catalog. Case is ignored.

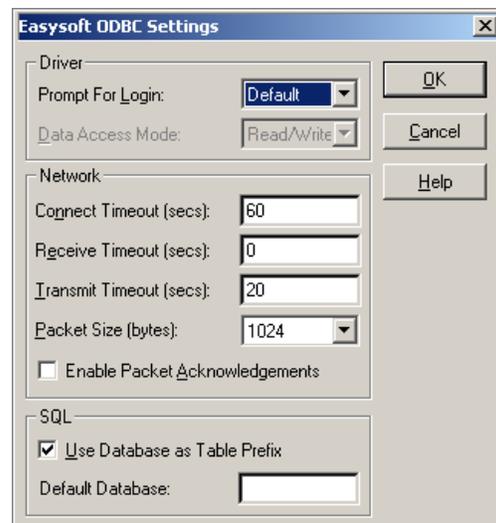
Catalog Login Password The password associated with the Catalog login user name. Case is ignored.

Now that a data source has been created, you can connect to data on the server using any ODBC-compliant application.

Settings

The Easysoft ODBC Settings dialog box is accessed by selecting **Settings...** on the Easysoft ODBC Setup dialog box. It is used to set driver, network and SQL options.

Easysoft ODBC Settings dialog box



Information required for Easysoft ODBC Settings

Prompt For Login Controls the display of the Easysoft ODBC Login Prompt when a connection is made to a data source:
Default: display is defined by the application.
Never: dialog box never displayed. To ensure successful operation, ensure that login information is supplied in the Easysoft ODBC Setup dialog box.
Always: dialog box always displayed.

Data Access Mode (not yet supported) Allows data to be either read/write or read only.

Connect Timeout The attempt to connect is cancelled if it is not successful within the specified time.

Receive Timeout The connection is cancelled if a reply is not received within the specified time. **Note:** the value zero means wait forever.

Transmit Timeout The connection is cancelled if the network layer cannot send the packet in the specified time.

Note: the timeouts for each option are identical for both Client and Server.

Packet Size Size of the data packets used to transmit data over the network. Valid options are 128, 256, 512, 1024 (default), 2048, 4096, 8192, 16384, 32768.

Enable Packet Acknowledgements This option checks to see if data has been received. The default is OFF. If network problems are experienced it can be turned ON. Using this options slows down the transmission of the data.

Use Database as Table Prefix In the case where a catalog contains information on two or more databases, and where those databases contain tables with the same name, this option allows us to differentiate the tables. e.g. If there are two SALES tables, one in a database called MINE and one in a database called THEIRS, then we would be able to connect to both of these using the data source. We would see the tables as MINE_SALES and THEIRS_SALES.

Default Database If the Use Database as Table Prefix option is not selected, then a default database must be specified, even if only a single database exists. If there is more than one database in the catalog, only the one specified with this option will be accessible.

Additional Notes

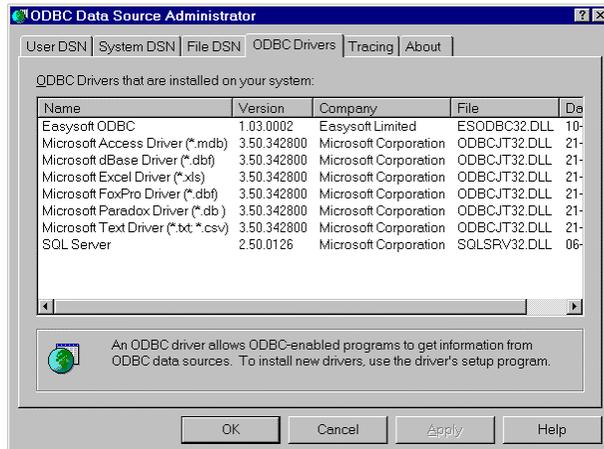
To remove a data source, highlight it and select **Remove**. A dialog box appears asking for confirmation. Choose **Yes** to delete the data source, **No** to cancel the operation.

To modify a data source, highlight it and select **Configure...**

Drivers Tab

The ODBC Drivers tab gives information about drivers.

ODBC Drivers tab



Each line in the list box shows the driver name, followed by the version, company, name of the driver file and the date it was produced.

Summary

In this module you learnt

- ✦ the Easysoft ODBC Setup dialog box is used to add and modify Easysoft data sources

Supplement: troubleshooting

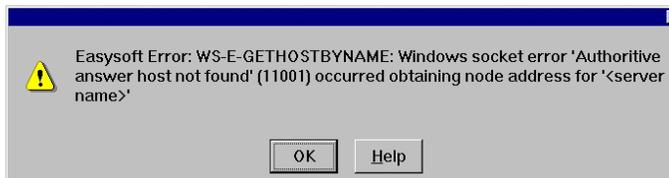
This section lists some of the common error message that can arise when connecting to server data. In cases where the answer is not obvious, a solution is described.



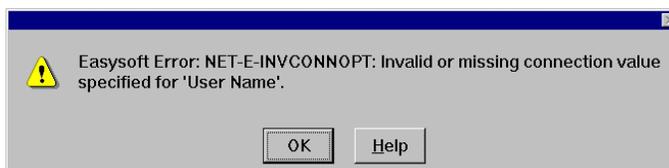
Cause: Incorrect server login name and/or password specified in the Easysoft ODBC Setup dialog box or the Easysoft ODBC Login Prompt.



Cause: Incorrect catalog login name and/or password specified in the Easysoft ODBC Setup dialog box or the Easysoft ODBC Login Prompt.



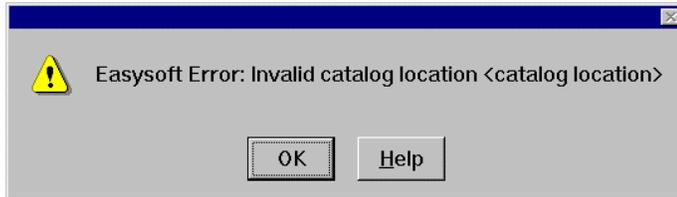
Cause: Incorrect server name specified in the Easysoft ODBC Setup dialog box or the Easysoft ODBC Login Prompt.



Cause: A blank user name was specified in the Easysoft ODBC Setup dialog box or the Easysoft ODBC Login Prompt.



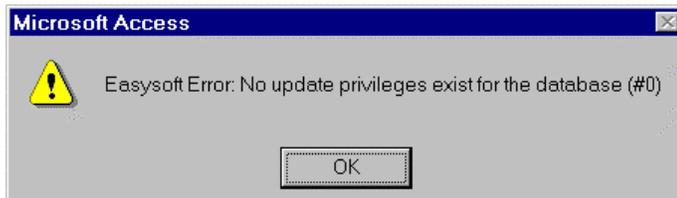
Cause: A blank password was specified in the Easysoft ODBC Setup dialog box or the Easysoft ODBC Login Prompt.



Cause: non-existent catalog location specified in the Easysoft ODBC Setup dialog box.



Cause: different versions of Easysoft on the server and the PC.



Cause: An attempt was made to update a record. However, the user does not have update privileges. Similar messages occur for other operations (select, insert and delete) if the user does not have the required privileges.

5. Easysoft Administration

This module deals with the CODA utilities on the server.

In this module you will learn how to

- ✦ make CODA companies available to the Easysoft system
- ✦ change CODA names and passwords in the Easysoft Catalog
- ✦ restrict access to CODA data
- ✦ change catalog passwords

Contents

CODA Utilities _____	5-2
Changing Catalog Passwords _____	5-3
Adding CODA Companies to Catalog _____	5-4
Changing CODA Usernames, Passwords and Privileges _____	5-6
Ordering BALANCE and BUDGET Data _____	5-8
Summary _____	5-9
Catalog Administration Exercises _____	5-10
Administration Review Questions _____	5-14
Administration Review Answers _____	5-15
Easysql _____	5-16

CODA Utilities

The Easysoft catalog contains information about the structure of CODA data and CODA users. This information is used by the software when it accesses CODA data and returns it to an ODBC application. This information can change, for example, users may be deleted, or a new company may be created. Therefore, when changes are made, the catalog needs to be updated. The CODA utilities aid this update process. (Note that it is not a change to the data that is important to the catalog, but the structure of the data). The left column of the following table lists the changes in CODA that would require the catalog to be updated. The right column indicates which of the utilities should be used to notify the catalog.

CODA change	Notify catalog using
Add or delete company	Codacat
Add or delete CODA user	Codacat (refresh users)
Change CODA user name or password	Codauser
Add or delete Nominals, Subaccounts or Level3s	Codaxref (also see module 6)
Change VMS password (not strictly CODA)	Catalog does not need to know, but must change Server Login in Easysoft ODBC Setup dialog box

In addition to CODA changes which need to be registered in the catalog, the catalog supports a number of supplementary roles.

Catalog change	Notify catalog using
Change password for catalog access (note: cannot change username for catalog)	Catuser routine
<i>Additional</i> restrictions on CODA access rights (cannot over-ride CODA restrictions)	Codauser

Note: CODA passwords are case sensitive. Catalog passwords are not case sensitive.

The four utilities which are used in conjunction with Easysoft ODBC for CODA, namely Catuser, Codacat, Codauser and Codaxref, are run automatically during the installation on the server, but they may also be run individually when and if required.

Catuser informs the catalog of catalog users. It is used to change the password of existing catalog users. Any catalog user can use this routine to change their own password, and there is also a special option for the catalog administrator who can change the passwords for all users.

Codacat is used to populate the catalog with CODA companies and associated information. This command does not create a company for CODA; it adds an existing CODA company to the Easysoft catalog. The catalog must already exist (one is created during the CODA setup process).

Codauser informs the catalog of CODA users. It is used to change the names and/or passwords of existing CODA users. It can also be used to set additional restriction on access privileges to CODA companies (these changes are within the Easysoft Catalog only; no changes are made to CODA data).

Codaxref is used to arrange data in N/S/L3 order when viewed using the BALANCE and BUDGET tables. In general, if Nominals, Subaccounts or Level3s are added or deleted, the routine should be re-run (but see "Ordering BALANCE and BUDGET Data", page 5-8 of this module for an exception). An option in the Easysoft ODBC for CODA installation on the server allows you to set up a batch job to run the Codaxref routine automatically every night (you can change when the routine is run if the default time is not convenient - see module 6).

Changing Catalog Passwords

The Catuser routine is used to change the password of existing Catalog users. It resides in the directory pointed to by the EASYSOFT_SQL_SYSTEM logical. Any catalog user can run this routine, but there is a special option for the catalog administrator. First, the general case is described, followed by the catalog administrator user options.

```
$ RUN EASYSOFT_SQL_SYSTEM:CATUSER
<informational message>
Catalog Directory : EASYSOFT_SQL_CODA_CATALOG
Catalog User      : MIKE
Catalog Password  :
New Password      :
Verify Password   :
Catalog password successfully updated for user MIKE
```

Catalog Directory refers to the directory in which the catalog resides. Either type the name of the directory, or type a logical that points to the directory.

Catalog User refers to the user whose password you wish to change. If the specified user does not exist, an error message is generated. If the catalog administrator user is entered here, special conditions apply (see below). Catalog User equates to Catalog Login Username in the Easysoft ODBC Setup dialog box.

Type the current Catalog Password. If the password is incorrect, an error message is generated. Catalog Password equates to Catalog Login Password in the Easysoft ODBC Setup dialog box.

Enter the New Password. You must verify the new password. If the names do not match, the following message appears: `Passwords do not match`. You must re-enter both the new password and the verification.

You can quit the routine at any stage except the last one by entering a blank line. All the existing values are retained. However, if you get as far as entering a new password, then you must complete the process by entering the verification. If you enter a blank line at this stage, the verify password prompt re-appears.

Catalog Administrator Options

The catalog administrator has the option of changing passwords for all users.

```
Catalog Directory : EASYSOFT_SQL_CODA_CATALOG
Catalog User      : ADMIN
Catalog Password  :
Administrator Options:
  1) Change Administrator Password
  2) Change User Password
  0) Exit
Enter option number [1, 2 or 0] - 2
Catalog User      : DEMO
New Password      :
Verify Password   :
Catalog password successfully updated for user DEMO
Catalog User      :
```

Catalog administrator's name (always "ADMIN") and password.

Name and new catalog password of non-administrator user.

If, at the Catalog User and Password prompts, the catalog administrator details are entered, three options are presented.

Option 1 behaves exactly as described in the previous section.

Option 2 (shown in the example) allows the catalog administrator to change catalog passwords for any user defined in the catalog. Press the **Enter** key at the Catalog User prompt to quit the routine.

Option 3 (labelled 0) is used to quit the routine at this point.

Adding CODA Companies to Catalog

The Codacat routine is used to add CODA companies and associated information to the Easysoft Catalog. This command does not *create* a company for CODA; it *adds* an existing CODA company to the Easysoft Catalog. The catalog must already exist - a catalog is created using the CODA setup routine.

```
$ RUN_EASYSOFT_SQL_SYSTEM:CODACAT
<informational messages>
```

```
Catalog Directory : EASYSOFT_SQL_CODA_CATALOG
Admin Password   :
```

Catalog Directory refers to the directory in which the catalog resides. Either type the name of the directory, or type a logical that points to the directory.

Only catalog administrators can use this routine; enter the Admin Password. If the specified password does not exist, an error message is generated. Admin Password equates to the Catalog Login Password (for the ADMIN user) in the Easysoft ODBC Setup dialog box.

Following a message indicating the CODA version, a list of CODA companies is presented.

```
CODA version : <version number>
Companies within group IAS$GROUP :
  DEMO      *NEWDEM      OLDDEMO      TEST      * (ALL)
Select Company : NEWDEM
Updating Catalog
  Writing : NEWDEM .....
Select Company :
$
```

Quit the routine at any prompt by entering a blank line.

Enter one or more companies at the Select Company: prompt. If you enter a list of companies, they must be separated either by a comma and/or a space. To select all the companies use the asterisk character (*).

The asterisk character (*) can be used as a wildcard, representing one or more characters; it can be placed either at the start or the end of a word, and there can only be one wildcard in a company name. For example, entering ***DEMO** at the Select Company prompt would result in DEMO and OLDDEMO being added to the catalog.

The asterisk character (*) preceding a company (e.g. ***NEWDEM**) indicates that this company already exists in the catalog. If you try to add this to the company, you are presented with the following prompt:

```
Select Company : NEWDEM
NEWDEM exists in catalog. Overwrite, Delete or Refresh users? (O/D/R) :
```

The Overwrite option clears all existing definitions for the company, and enters new definitions into the catalog.

The Delete option removes all information about the specified company from the catalog.

The Refresh users option results in the addition of new CODA users to the catalog (i.e. if a CODA user is not already in the catalog, then the user is added) and the deletion from the catalog of users who are no longer valid CODA users. If there are any existing users already in the catalog, their names and passwords are not added to the catalog (see next section for information about names and passwords in the catalog).

If a new company is added to the catalog, and if a user of the company already exists within the catalog the catalog password will not be changed.

Users and Passwords

The Codacat routine does not store CODA passwords in the catalog. Instead, for each CODA user, the password is set to the username. To make the password which is stored in the catalog the same as the CODA password, use the Codauser routine described in the next section.

Additionally, each user is given a Catalog Login Username and Password which is used to control access to the catalog. These are both set to the CODA username. To change the password of the catalog user, use the Catuser routine (see “Changing Catalog Passwords”, page 5-3).

For example, say there is a CODA user called MIKE with a password MIKEPWD. The Codacat routine stores the following information in the catalog:

CODA user:	MIKE	—————	<i>Change using the Codauser routine.</i>
CODA password:	MIKE	—————	
Catalog (Login) user:	MIKE	—————	
Catalog (Login) password:	MIKE	—————	<i>Change using the Catuser routine.</i>

To access CODA data, either the CODA password stored in the catalog must be changed to MIKEPWD using the Codauser routine (see next section) or the CODA password itself must be changed (to MIKE) using CODA.

Note that once a Catalog user has been added to the catalog, the catalog username cannot be changed.

Changing CODA Usernames, Passwords and Privileges

The Codauser routine is used to change the names and/or passwords of existing CODA users and to change access privileges to CODA companies (these changes are within the Easysoft Catalog only; no changes are made to CODA data). This latter function is only available to the catalog administrator and is explained in the next section. The Codauser routine resides in the directory pointed to by the EASYSOFT_SQL_SYSTEM logical.

```
$ RUN EASYSOFT_SQL_SYSTEM:CODAUSER
<informational message>
Catalog Directory      : EASYSOFT_SQL_CODA_CATALOG
Catalog User          : STEVE
Catalog Password      :
```

Catalog Directory refers to the directory in which the catalog resides. Either type the name of the directory, or type a logical that points to the directory.

Catalog User refers to the user whose CODA name or password you wish to change. If the specified user does not exist in the catalog, an error message is generated: `Catalog user <name> does not exist`. Catalog User equates to Catalog Login Username in the Easysoft ODBC Setup dialog box.

Type the Catalog Password for the user. If the password is incorrect, an error message is generated: `Catalog Password for catalog user <name> is invalid`. Catalog Password equates to Catalog Login Password in the Easysoft ODBC Setup dialog box.

Since each CODA user may have access to many different companies, a list of available companies is presented. You can change the password either for all the companies at once, in which case all the passwords will be identical, or you can change the password for one or more companies separately, in which case the password for the user can be different for different companies.

Companies available to catalog user STEVE :

NEWDEM * (ALL)

Company : **NEWDEM**

Coda User : **<enter CODA user here>**

New Password :

Verify Password :

Update successful for catalog user STEVE with company NEWDEM
Company :

Enter the Coda User. If the user does not already exist, then a user with the new name is created and linked to the catalog user.

Enter the new password. You must verify the new password. If the names do not match, the following message appears: `Passwords do not match`. You must re-enter both the new password and the verification.

You can quit the routine at any stage except the last one by entering a blank line. All the existing values are retained. However, if you get as far as entering a new password, then you must complete the process by entering the verification. If you enter a blank line at this stage, the verify password re-appears.

Changing CODA Privileges

The catalog administrator can change CODA access privileges for all users.

Catalog Directory	: EASYSOFT_SQL_CODA_CATALOG	<i>Catalog administrator's</i>
Catalog User	: ADMIN	<i>name (always "ADMIN")</i>
Catalog Password	: <administrator password>	<i>and password.</i>

Administrator Options:

- 1) Change Coda Password
- 2) Change Coda company privileges
- 0) Exit

Enter option number [1, 2 or 0] - **2**

Catalog User (*=ALL) : **STEVE**

Companies available to catalog user STEVE :

DEMO NEWDEM OLDDEMO * (ALL)

Company : **NEWDEM**

Privilege Options:

- 1) Read Only
- 2) Full Access
- 0) Exit

Enter option number [1, 2 or 0] - 1

Privileges changed for catalog user STEVE within company NEWDEM

If, at the Catalog User and Password prompts, the catalog administrator details are entered, three options are presented.

Option 1 behaves exactly as described in the previous section.

Option 2 (shown in the example) allows the catalog administrator to change access privileges to CODA companies. The options are:

- Read Only 1
- Full Access 2
- Exit 0

Option 3 (labelled 0) is used to quit the routine at this point.

Ordering BALANCE and BUDGET Data

The BUDGET and BALANCE tables return data in Nominal/Subaccount/Level3 order by using a special table called ACCOUNT which is generated by running the Codaxref routine. This can take from a few minutes to a few hours. The time taken depends not only on the processor speed of the Server, and the priority allocated to the process, but also on the size of the data files. This is purely an optimisation process for reading data, and does not in any way affect the CODA data.

Take the following steps to run Codaxref. This must be done for each company in the CODA group for which you wish to use the Easysoft software.

```
$ RUN EASYSOFT_SQL_SYSTEM:CODAXREF
```

```
<copyright message>
```

```
Catalog Directory : EASYSOFT_SQL_CODA_CATALOG
Catalog User      : STEVE
Catalog Password  :
```

Enter the location of the catalog for which you want to generate the cross-reference information. Catalog user should be one that is also a CODA user who has privileges on all the companies for which you are generating the cross-reference information.

```
Companies available to user STEVE :
EASYSOFT * (ALL)
Company                            :
```

```
Processing files in directory <device><directory>
<informational messages appear here>
Company                            :
```

In general, whenever Nominals, Subaccounts or Level3s are added or deleted, Codaxref should be run. This ensures that the file which it generates is always up-to-date. A default batch job to run Codaxref is set up during the installation of Easysoft ODBC for CODA on the server platform.

If a new N/S/L3 combination is inserted through the BALANCE or BUDGET table, that combination is automatically written to the ACCOUNT table. If *only* Easysoft ODBC for CODA is used to insert Budget data using these two tables, the Codaxref routine need not be run, apart from the initial installation. However, if you use any other tables to change Budget data, or if you use CODA itself to make the changes, then Codaxref should be run as usual.

Summary

In this module you learnt how to

- ✦ tell the catalog about CODA companies using the Codacat routine
- ✦ change catalog passwords using the Catuser routine
- ✦ change names and passwords of CODA users (stored in the catalog) using the Codauser routine
- ✦ set additional access restrictions on CODA users using the Codauser routine

Catalog Administration Exercises

Please complete the following steps of the exercise:

1. Log onto CODA and change your password
2. Update the Catalog with your new CODA password
3. Update the Catalog with your new Catalog password
4. Update your Data source with your new Catalog password
5. Test the connection to CODA data

Note 1: Steps 3 and 4 are optional. You may leave your catalog password the same as your username to enable you to easily recall it.

Note 2: In the following examples, the original CODA password is TRAINEE1 (replace this with your CODA password), and it is changed to EASYSOFT (select a password of your choice).

1. Log onto CODA and change your password

Step One	Open a Reflections session (double click on the icon)
Step Two	Use your assigned user name and password to gain access to CODA
Step Three	Select the menu options ME (Me) PA (Password)
Step Four	Input a new password, press the Tab key; enter your old password, press the Enter key to return to the ME menu

ME Menu type	GOLD	HELP	INS/OVER	EXIT
PA Password	Menu type	Password	Company	Feature Test
CO Company	Expert mode	Change printer	Keypad Behaviour	Enable logging
FT Feature Test	Change user	User defaults	Deferred postings	Enter
EM Expert mode		Change batch Q		
CP Change printer	New password	EASYSOFT	Old password	TRAINEE1
KB Keypad Behaviour				
LG Enable logging				
CU Change user				
UD User defaults				
DP Deferred postings				
CQ Change batch Q				

2. Update the Catalog with your new CODA password

Step One	Go to the IAS> prompt by pressing Gold \$ (Num Lock and Shift 4 (\$))
Step Two	Run the Codauser routine: IAS> RUN EASYSOFT_SQL_SYSTEM:CODAUSER
Step Three	Follow the steps provided - see the worked example below

```
To return simply type LOGOFF to the prompt string
IAS> RUN EASYSOFT_SQL_SYSTEM:CODAUSER
Easysoft Coda Catalog Password Changer Version <version>
Copyright (c) 1993-1997 by Easysoft Limited, Leeds, United
Kingdom.
All Rights Reserved.
Catalog Directory      : EASYSOFT_SQL_CODA_CATALOG
Catalog User          : TRAINEE1
Catalog Password      : TRAINEE1
Companies available to catalog user TRAINEE1 :
  EASYG01      * (ALL)
Company        : *
Coda User      : TRAINEE1
New Password   : EASYSOFT
Verify Password : EASYSOFT
Update successful for catalog user TRAINEE1 with company EASYG01
Company       :
```

Note: to return to the IAS> prompt press the **Enter** key. To return to CODA from the IAS> prompt type **LOGOFF** followed by the **Enter** key.

3. Update the Catalog with your new Catalog password

Step One	Go to the IAS> prompt by pressing Gold \$ (Num Lock and Shift 4 (\$))
Step Two	Run the Catuser routine: IAS> RUN EASYSOFT_SQL_SYSTEM:CATUSER
Step Three	Follow the steps provided - see the worked example below

```

IAS> RUN EASYSOFT_SQL_SYSTEM:CATUSER
Easysoft Catalog Password Changer Version <version>
Copyright (c) 1993-1997 by Easysoft Limited, Leeds, United
Kingdom.
All Rights Reserved.
Catalog Directory : EASYSOFT_SQL_CODA_CATALOG
Catalog User      : TRAINEE1
Catalog Password  : TRAINEE1
New Password      : EASYSOFT
Verify Password   : EASYSOFT
Catalog password successfully updated for user TRAINEE1

```

Note: to return to the IAS> prompt press the **Enter** key. To return to CODA from the IAS> prompt type **LOGOFF** followed by the **Enter** key.

4. Update your Data source with your new Catalog password

Modify the Data source you set up in the previous module.

Change the Catalog Login Password.

*New password
is EASYSOFT*

Make sure you save your changes by selecting **OK**.

5. Test the connection to CODA data

Press the **Test** button on the Easysoft ODBC Setup dialog box. Assuming everything is correct, you should see:



If you have input your Catalog password incorrectly then you will get a message:



Administration Review Questions

1. What CODA utilities are there, and what is the basic function of each one of them?

2. Imagine there is a new CODA user called “JOHN”, with a password of “JOHNPWD”. When the Easysoft catalog is told about this user, what are the following items in the catalog set to?

CODA username:

CODA password:

catalog username:

catalog password:

3. If a CODA user has read-only access privileges in CODA, is it possible to over-ride these using Easysoft, and give the user write privileges?

4. When should Codaxref be run and when is it usually run by default?

5. If the OpenVMS password is changed, does the Easysoft system need to be told of this?

Administration Review

Answers

1. What CODA utilities are there, and what is the basic function of each one of them?

Codacat, Codauser, Catuser, Codaxref

- Codacat - add information about CODA companies to the Easysoft catalog
- Codauser - change name and/or password of CODA users and change access privileges (in the Easysoft catalog)
- Catuser - change password of existing catalog users
- Codaxref - pre-requisite to using the BALANCE and BUDGET tables. Allows data to be returned in N/S/L3 order

2. Imagine there is a new CODA user called "JOHN", with a password of "JOHNPWD". When the Easysoft catalog is told about this user, what are the following items in the catalog set to?

CODA username: JOHN

CODA password: JOHN

catalog username: JOHN

catalog password: JOHN

3. If a CODA user has read-only access privileges in CODA, is it possible to over-ride these using Easysoft, and give the user write privileges?

No.

4. When should Codaxref be run and when is it usually run by default?

In general, it should be run after a Nominal, Subaccount or Level3 has been added to or deleted from a CODA company.

By default, it is run every night at midnight. This ensures that any changes are promptly inserted into the catalog.

5. If the OpenVMS password is changed, does the Easysoft system need to be told of this?

Yes. The Server Login Password in the Easysoft ODBC Setup dialog box (accessed using Microsoft ODBC Administrator on the PC) needs to be changed. (Note, this is not a CODA utility, but whilst discussing passwords, it is a useful point to mention).

Easysql

Easysql is a debug tool which forms part of the Easysoft Server Component. Its function is to send SQL to the server data and output the results. If problems are encountered, check whether the query that is sent from an application returns the same data as Easysql.

In these exercises, you will use Easysql to send two queries to the server. These are identical to the two queries you have used in the previous exercises, namely, downloading the entire BUDGET data, and then downloading just the Actuals data.

1. Log on to the server using Reflection.
2. Set up a symbol to point to Easysql. At the command prompt type:

```
$ SQL:==$EASYSOFT_SQL_EASYSQL
```
3. Start Easysql. At the command prompt type:

```
$ SQL EASYSOFT_SQL_CODA_CATALOG <CODA user name> <CODA password>
```

Easysql takes three parameters

- 1) catalog directory i.e. EASYSOFT_SQL_CODA_CATALOG
- 2) a CODA user name
- 3) a password for the CODA user

4. After a copyright message you will see the EASYSQL> prompt. At this type:

```
EASYSQL> SELECT * FROM EASYGO_BUDGET;  
<BUDGET data appears here>
```
5. To select just the Actuals, type:

```
EASYSQL> SELECT * FROM EASYGO_BUDGET WHERE BUDGET_CODE = 'A';  
<BUDGET Actuals data appears here>
```
6. To exit from Easysql, press the **Enter** key at the EASYSQL> prompt.

6. CODA Tables, Indexes and Relationships

In this module you will learn about

- ✦ the structure of data tables you see when you use Easysoft ODBC for CODA
- ✦ the correspondences between what you see in an application and the CODA-IAS interface
- ✦ Easysoft optimisation

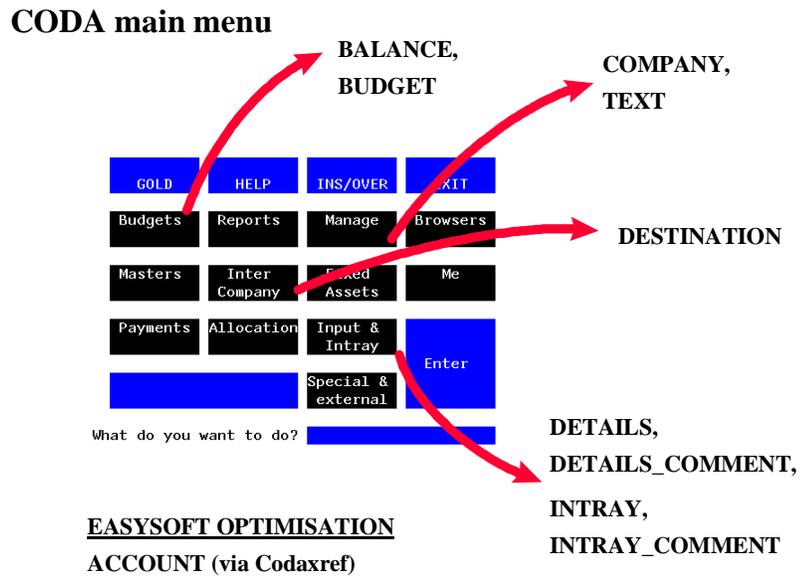
At the end of the module there are some review questions.

Contents

CODA-IAS and Table Correspondences	6-2
List of CODA Tables	6-4
CODA Table Relationships	6-5
Column Details Tables	6-6
Index Details Tables	6-8
Easysoft Optimisation	6-9
Changing Codaxref Batch Job Defaults	6-9
Summary	6-11
CODA Tables Review Questions	6-12
CODA Tables Review Answers	6-14

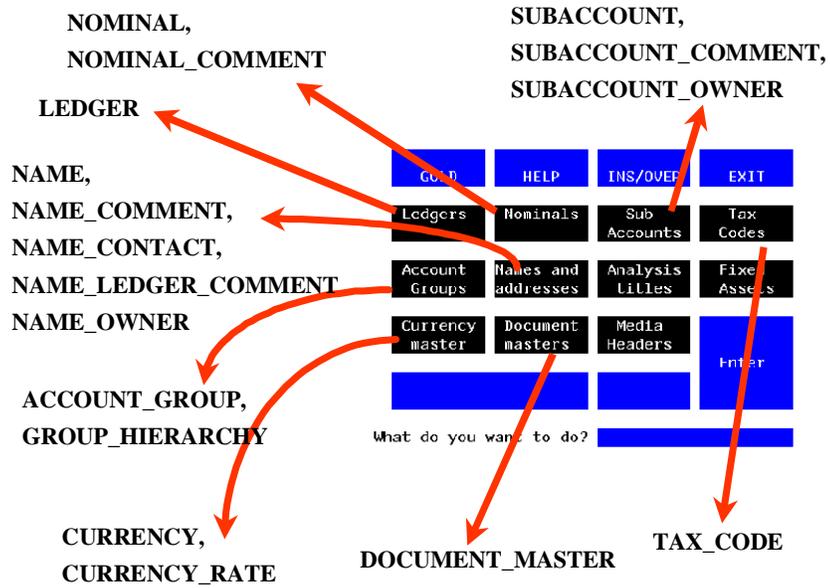
CODA-IAS and Table Correspondences

The tables that users see when accessing CODA data using Easysoft ODBC are called *CODA tables*. The correspondence between these and the CODA-IAS interface is shown graphically in the next two diagrams. Refer to “List of CODA Tables” on page 6-4 for a view of these correspondences in tabular form.



NB: Easysoft optimisation is discussed on page 6-9.

CODA Masters Menu



Although the exact appearance of these CODA tables depends upon the application you are using, the structure will be the same whatever application you use - there will be columns containing data values. Here is an example using BUDGET data. The correspondences between column names and CODA menus are described in the next section, “**Error! Not a valid link.**”.

NOMINAL	SUBACCOUNT	LEVEL3	BUDGET_YEAR	BUDGET_COD	ACCOUNT_BA	ADJUSTMEN	FINAL_VALUE	OPENING_VAL	TOTAL_VALUE	VALUE_0001
	ENCARTA		95 A	N		0	0	0	-2135	0
	QUATTROPRO		95 A	N		0	0	0	-5618	0
	FOREST&TREE		95 A	N		0	0	0	-5040	0
	WINDOWS		95 A	N		0	0	0	-32940	0
	MPROMPTU		95 A	N		0	0	0	-959.94	0
	MPROMPTU		95 A	N		0	0	0	-2999.9	0
	CORELDRAW		95 A	N		0	0	0	-29640	0
	POWERBUILD		95 A	N		0	0	0	-6499.67	0
	WORDPERFEC		95 A	N		0	0	0	0	0
	VISUALBASIC		95 A	N		0	0	0	0	0
	VISUALC++		95 A	N		0	0	0	0	0
	MYSUB		96 F	N		4	0	0	0	9.85
	MYSUB		96 G	N		0	0	0	100	9.85
	MYSUB		96 H	N		0	0	0	200	6.15
	MYSUB		96 A	N		0	0	0	-149075	-60000
PURCHA	FRONTLINE		95 A	N		0	0	0	-149075	-60000
PURCHA	FRONTLINE		95 B	Y		0	0	0	-534.49	-50
PURCHA	FRONTLINE		96 B	Y		0	0	0	-163982.5	-66000
PURCHA	FRONTLINE		97 B	Y		0	0	0	-180360.75	-72600
PURCHA	FRONTLINE		98 B	Y		0	0	0	-198418.83	-79860
PURCHA	FRONTLINE		99 B	Y		0	0	0	-218260.71	-87846
SALES			95 A	N		0	0	0	360600.24	6135.14

Annotations in the image:

- COLUMN NAMES**: A red oval highlights the header row of the table.
- DATA VALUES IN COLUMN**: A red oval highlights a column of data values.

List of CODA Tables

The table below lists the CODA tables, gives an explanation of the content, indicates the (primary) CODA menu which relates to the information contained in the CODA table, and then lists the operations which are allowed on the table. All the data can be read (selected) by ODBC applications, but some operations are not allowed on some tables.

The fields that can be updated in a record, and the records that can be inserted and deleted depend upon the particular version of CODA and the version of IASLINK that your system uses. These features are outside the control of Easysoft. As a guideline, the operations that are *disallowed* for CODA version 6.7 are shown in the table below. The disallowed operations are indicated using the “×” character (I = INSERT, U = UPDATE, D = DELETE).

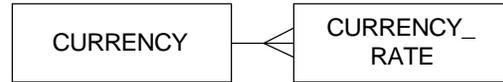
CODA table name	Contains	CODA menu	I	U	D
ACCOUNT	Easysoft optimisation	-	×	×	×
ACCOUNT_GROUP	Account Group information	MM AG			×
BALANCE	Budgets	BU			
BUDGET	Budgets	BU			
COMPANY	CODA Company information	MA GM CO	×	×	×
CURRENCY	Currencies	MM CM		×	×
CURRENCY_RATE	Currency rates	MM CM			×
DESTINATION	Intercompany Destination	IC MD			
DETAILS	Input to books	IN IN		×	×
DETAILS_COMMENT	Details comments	IN IN		×	
DOCUMENT_MASTER	Document Masters	MM DO	×		×
GROUP_HIERARCHY	Account Group hierarchy	MM AG		×	
INTRAY	Input to Intray	IN AD			×
INTRAY_COMMENT	Intray comments	IN AD		×	
LEDGER	Ledgers	MM LE			×
NAME	Name and Address	MM NA			
NAME_COMMENT	Name and Address account comments	MM NA		×	
NAME_CONTACT	Name and Address contact list	MM NA		×	
NAME_LEDGER_COMMENT	Name and Address ledger comments	MM NA		×	
NAME_OWNER	Name and Address financial details	MM NA			
NOMINAL	Nominal accounts	MM NO			
NOMINAL_COMMENT	Nominal accounts comments list	MM NO			×
SUBACCOUNT	Subaccounts	MM SA			
SUBACCOUNT_COMMENT	Subaccount comments	MM SA		×	
SUBACCOUNT_OWNER	Subaccount Nominal/ledger list	MM SA			
TAX_CODE	Tax Codes	MM TX			
TEXT	Field labels, messages in CODA	MA GM TE	×	×	×

These tables are described in the appendix.

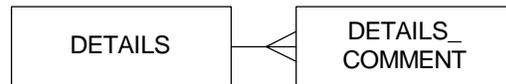
CODA Table Relationships

This section shows the relationships between the most commonly used CODA tables. The “crows foot” (—<) indicates a one:many relationship.

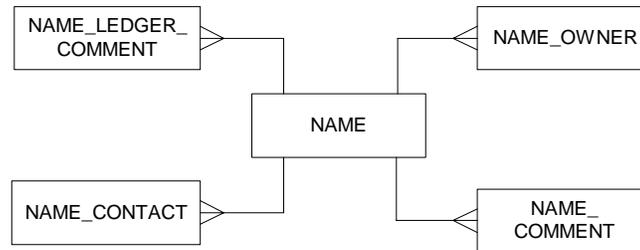
CURRENCY



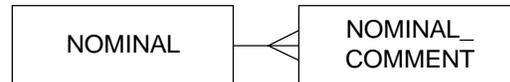
DETAILS



NAME



NOMINAL



SUBACCOUNT



Column Details Tables

Users need to know the correspondences between the column names in a table and the CODA-IAS menu names. Usually, the correspondences are obvious, but this is not always the case.

Additionally, some users may need to know information about the structure of the CODA tables that they see. For example, the exact data type of some column, or the size of a column.

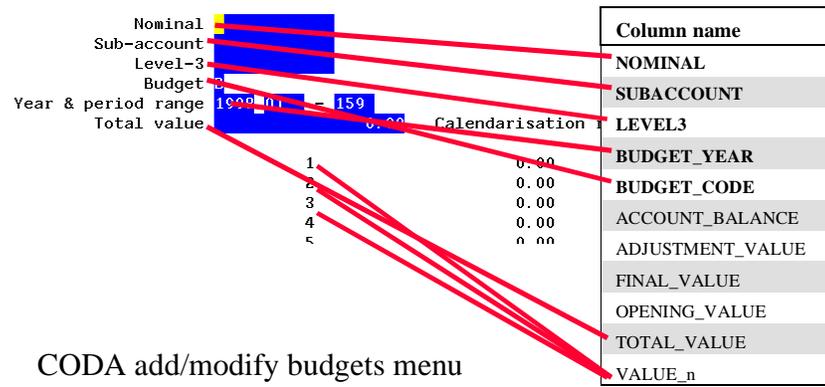
These two requirements are fulfilled by the *column details* tables which are shown in the appendix. Each of the CODA tables can be described in detail using these. Here there is just one example using the BUDGET column details table.

BUDGET column details					
Column name	Data type	Size	Def	CODA name	IASLINK name
NOMINAL	VARCHAR	12		Nominal	LB-BUDGET-NOM
SUBACCOUNT	VARCHAR	12		Sub-account	LB-BUDGET-SUB
LEVEL3	VARCHAR	12		Level-3	LB-BUDGET-LV3
BUDGET_YEAR	SMALLINT	2		Year & period range	LB-BUDGET-YEAR
BUDGET_CODE	VARCHAR	1	B	Budget	LB-BUDGET-BUD
ACCOUNT_BALANCE	VARCHAR	1		-	-
ADJUSTMENT_VALUE	DOUBLE	8		Value	LB-BUDGET-ADJUSTMENTS
FINAL_VALUE	DOUBLE	8		Value	LB-BUDGET-FINAL
OPENING_VALUE	DOUBLE	8		Value	LB-BUDGET-OPENING
TOTAL_VALUE	DOUBLE	8		Total value	-
VALUE_n	DOUBLE	8		[number]	LB-BUDGET-VALUE(n)

Column name is the name of the column as it appears to users of ODBC applications. **PRIMARY KEYS** are indicated by **BOLD CAPITAL** text (e.g. **NOMINAL**, **SUBACCOUNT**, etc.)

CODA name is the corresponding name of the column as it appears on CODA menus.

The relationship between column names in a CODA table and CODA (menu) names is exemplified in the figure below which shows a CODA Budget Maintenance screen and its equivalents in the BUDGET column details table.



Some of the columns that are available in the BUDGET table are not seen on the CODA screen. This is because the exact appearance of the CODA-IAS interface depends upon options that are selected. In the case above, Adjustment value, Final value and Opening value were not requested, so they do not appear.

Other cases in which there is no correspondence between a CODA screen and a table is with the columns CREATED_USER, MODIFIED_USER, CREATED_DATE and MODIFIED_DATE where the information is taken internally from the CODA system.

Data type is the SQL data type of the column. (Data types are briefly discussed in module two, and a technical specification of the available data types is given in appendix D of the manual).

Size is the number of bytes needed to store the data. If the data type is *varchar*, then *size* is the maximum number of bytes needed to store the data. The format for columns of datatype DATE depends upon the Windows settings.

If a default value exists for a particular CODA column, this is shown in the column labelled *Def*. If there is no default, the column is blank.

IASLINK name is the corresponding name of the column in IASLINK. Replication of the IASLINK name is indicated by the suffix (*n*) e.g. LB-BUDGET-VALUE(*n*).

Index Details Tables

To aid accessing the data, there are indexes on CODA tables. If an index exists on the CODA table, it is described in an *index details* table. A full listing can be found in the manual, here we just explain the concepts. Again, the example is based on the BUDGET table.

BUDGET index details				
Index name	No	Dup	Coll	Fields in index
BUDGET_001	1	No	A	NOMINAL, SUBACCOUNT, LEVEL3, BUDGET_YEAR, BUDGET_CODE
BUDGET_002	2	Yes	A	NOMINAL
BUDGET_003	3	Yes	A	SUBACCOUNT
BUDGET_004	4	Yes	A	LEVEL3

No refers to the index number. Each index is given a number for identification purposes.

Dup refers to duplicates. Options are Yes or No.
Yes means that duplicate values are allowed in the index.
No means that duplicate values are not allowed in the index.

Coll refers to the collation (ordering) of the index, which may be Ascending (A) or Descending (D). This collation sequence operates on all the columns in the index.

Using indexes to aid retrieval of data has no effect on the data retrieved - they just speed up the accessing of data. This increase in speed is often huge, and therefore, whenever possible, indexes should be used. Although users cannot explicitly specify that an index should be used in a query, they can write queries such that if an index is available, it will be used.

To use an index, add criteria on the columns that are in the index. The more information that can be supplied related to the columns in the index, the faster the query will run.

The columns that are retrieved have no effect on the index usage. What is important is the columns in the criteria (the WHERE clause of a query).

Easysoft Optimisation

The BALANCE and BUDGET tables return data in N/S/L3 order. To do this, they require the ACCOUNT table, which references internal CODA ID numbers. The content of the ACCOUNT table is generated by means of the Codaxref routine which is run on the server. In general, if Nominals, Subaccounts or Level3s are added or deleted, the routine should be re-run. An option in the Easysoft ODBC for CODA installation on the server allows you to set up a batch job to run the Codaxref routine automatically every night just after midnight (you can change when the routine is run if the default time is not convenient).

Changing Codaxref Batch Job Defaults

The EASYSOFT_SQL_TEMP:CODAXREF.COM file controls the operation of the batch job which runs the Codaxref routine on the Server. You can change any of the lines to suit your particular requirements. The default options are shown immediately below, and following these, each line is explained in detail.

```
$ SUBMIT/AFTER=TOMORROW/KEEP/NOPRINT/LOG=EASYSOFT_SQL_TEMP:CODAXREF.LOG
  /NAME=CODAXREF EASYSOFT_SQL_TEMP:CODAXREF.COM
$ SHOW TIME
$ DEFINE IAS$GROUP <device><directory>
$ CODAXREF:=$EASYSOFT_SQL_SYSTEM:CODAXREF.EXE
$ CODAXREF EASYSOFT_SQL_CODA_CATALOG ADMIN ADMIN *
$ SHOW TIME
$ CURRENT_DIRECTORY = F$ENVIRONMENT("DEFAULT")
$ SET DEF IAS$GROUP
$ PURGE [...]EASY*.DAT
$ SET PROTECTION=(WORLD:R) [...]EASY*.DAT
$ SET DEF 'CURRENT_DIRECTORY'
$ PURGE/KEEP=5 EASYSOFT_SQL_TEMP:CODAXREF.LOG$
```

Description of Lines

```
$ SUBMIT/AFTER=TOMORROW/KEEP/NOPRINT
  /LOG=EASYSOFT_SQL_TEMP:CODAXREF.LOG
  /NAME=CODAXREF EASYSOFT_SQL_TEMP:CODAXREF.COM
```

This is the line that specifies when the batch job should run. Each of the options is explained below.

```
/AFTER=TOMORROW
```

States that the batch job should be run on the following day (i.e. just after midnight).

```
/KEEP
```

States that the log file should be kept.

/NOPRINT

Do not print a paper copy of the log file. This is a short form of NOPRINTER.

/LOG=EASYSOFT_SQL_TEMP:CODAXREF.LOG

Specifies the location and name of the log file that is generated when the batch job is run.

/NAME=CODAXREF

This is the job name - an identifier for the queue of batch jobs.

EASYSOFT_SQL_TEMP:CODAXREF.COM

The name of the command file used to run the batch job (i.e. this file which defines the batch job).

\$ SHOW TIME

This tells the program to record the current time.

\$ DEFINE IAS\$GROUP <device><directory>

This defines the logical name called IAS\$GROUP. <device> is the name of the machine on which the CODA data resides, and <directory> is the full path of the top-level directory in which the CODA data is stored. Data for each CODA company is stored in a separate sub-directory within IAS\$GROUP.

\$ CODAXREF:=\$EASYSOFT_SQL_SYSTEM:CODAXREF.EXE

This defines the symbol called CODAXREF to point to the Codaxref routine.

\$ CODAXREF EASYSOFT_SQL_CODA_CATALOG ADMIN ADMIN *

This runs the Codaxref routine. The routine accepts three arguments

1. ADMIN (this is the catalog administrator's name and cannot be changed)
2. the catalog administrator's password (by default, this is also ADMIN. Ideally, you should have changed it).
3. the CODA companies for which Codaxref should operate. You can either run Codaxref on a single named company or on all the companies known to the Easysoft system (i.e. all the companies for which details are recorded in the Easysoft Catalog). The asterisk character (*) represents all the companies known to the catalog.

By default, the EASYSOFT_SQL_CODA_CATALOG logical points to the directory [EASYSOFT.SQL.CODA.CATALOG]. If you want to use a catalog in a different location, then redefine the logical and re-run the CODA setup routine.

\$ DEFINE EASYSOFT_SQL_CODA_CATALOG <DEVICE><DIRECTORY>

\$ @EASYSOFT_SQL_SYSTEM:SETUP_CODA

\$ CURRENT_DIRECTORY = F\$ENVIRONMENT("DEFAULT")

Save the name of the current directory in a symbol called CURRENT_DIRECTORY.

\$ SET DEF IAS\$GROUP

Set the current directory to be the one in which the CODA company data resides.

\$ PURGE [...]EASY*.DAT

The Codaxref routine generates a file called EASYF001.DAT for each of the CODA companies. Each time that Codaxref is run, a new version of this file is created. The PURGE command removes all but the latest version of this file from all the sub-directories (each sub-directory contains data for a CODA company).

\$ SET PROTECTION=(WORLD:R) [...]EASY*.DAT

Allow all users to read, but not modify, Easysoft data files.

\$ SET DEF 'CURRENT_DIRECTORY'

Reset the current directory to be what it was when the Codaxref routine was started.

\$ PURGE/KEEP=5 EASYSOFT_SQL_TEMP:CODAXREF.LOG

CODAXREF.LOG is the log file that is generated when the Codaxref routine is run. Each time it is run, a new copy of this log file is made (unless meanwhile the name of the log file is changed). The PURGE command is used to prevent the server becoming overcrowded with log files.

/KEEP=<*n*> means that all but the last *n* log files will be removed. The default value for *n* is 5. The minimum allowed value for *n* is 1. If you do not include the /KEEP qualifier, all but the highest numbered version of the log file is deleted from the directory.

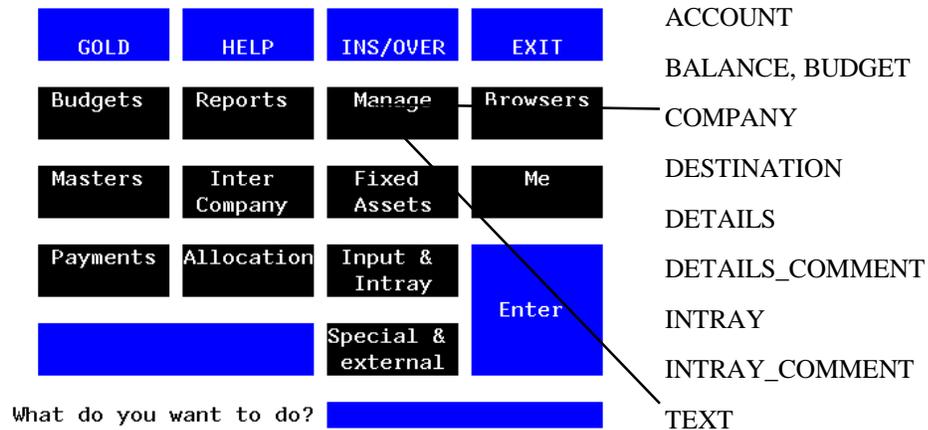
Summary

In this module you learnt about

- ✦ viewing CODA data in tables
- ✦ the correspondences between CODA-IAS and CODA tables
- ✦ the relationships between CODA tables
- ✦ describing the structure of a CODA table using a Column Details table
- ✦ determining what indexes are available using an Index table
- ✦ Easysoft optimisation. Run Codaxref to return BUDGET data in N/S/L3 order
- ✦ changing options in the Codaxref batch job using the codaxref.com file

CODA Tables Review Questions

- For the CODA main menu, draw the relationships between the submenus and the CODA tables. For example, data obtained using the Manage menu can be found in the tables COMPANY and TEXT.



- One table shown in the list above has no correspondence to the CODA-IAS screen. What is it, and what is it used for.

3. For the CODA Masters menu, draw the relationships between the submenus and the CODA tables.



ACCOUNT_GROUP
 CURRENCY
 CURRENCY_RATE
 DOCUMENT_MASTER
 GROUP_HIERARCHY
 LEDGER
 NAME
 NAME_COMMENT
 NAME_CONTACT
 NAME_LEDGER_COMMENT
 NAME_OWNER
 NOMINAL
 NOMINAL_COMMENT
 SUBACCOUNT
 SUBACCOUNT_COMMENT
 SUBACCOUNT_OWNER
 TAX_CODE

3. In addition to the tables that users see when accessing CODA data (i.e. CODA tables), Easysoft documentation lists two other types of table. What are they called, and what is their purpose?

4. List some situations in which there is not an exact 1:1 correspondence with field names on the CODA screen and column names in the CODA tables.

5. What is the purpose of running the Codaxref batch job?

6. When should Codaxref be run?

7. By default, when is the Codaxref batch job run?

CODA Tables Review Answers

1. For the CODA main menu, draw the relationships between the submenus and the CODA tables. For example, data obtained using the Manage menu can be found in the tables COMPANY and TEXT.

Check your answer against the first diagram in this module, and the list of CODA tables.

2. One table shown in the list above has no correspondence to the CODA-IAS screen. What is it, and what is it used for.

ACCOUNT - references internal CODA identification numbers and links N/S/L3

3. For the CODA Masters menu, draw the relationships between the submenus and the CODA tables.

Check your answer against the second diagram in this module, and the list of CODA tables.

4. In addition to the tables that users see when accessing CODA data (i.e. CODA tables), Easysoft documentation lists two other types of table. What are they called, and what is their purpose?

Column details tables, Index details tables

- Column details tables - describe the structure of the CODA tables.
- Index details tables - describe the indexes that are available to aid the accessing of CODA data.

5. List some situations in which there is not an exact 1:1 correspondence with field names on the CODA screen and column names in the CODA tables.

Field not named on the CODA screen
No field on the CODA screen
Set of fields given a single name in CODA

6. What is the purpose of running the Codaxref batch job?

Allows BALANCE and BUDGET data to be returned in N/S/L3 order

7. When should Codaxref be run?

After the addition or deletion of Nominals, Subaccounts or Level3s

8. By default, when is the Codaxref batch job run?

Each evening, just after midnight

7. Using ODBC Applications with CODA

This module shows you how to read CODA data using popular ODBC-compliant applications. You will also see how to access the data directly from the server.

In this module you will learn how to

- ✦ download data using Microsoft Query from Excel, Microsoft Access, Lotus 1-2-3, Microsoft Word Mail Merge
- ✦ add criteria to a query in each of the applications
- ✦ view the SQL statement that was sent in a query
- ✦ view available indexes using Microsoft Access

Contents

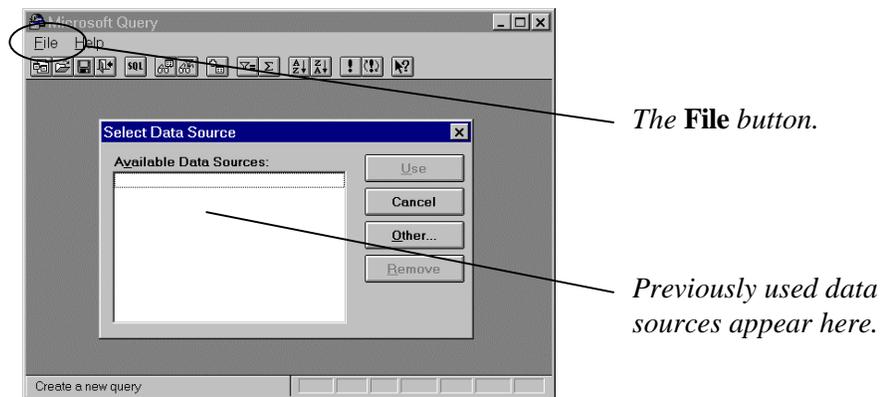
Microsoft Excel	7-2
Download Entire Table	7-2
Add Criteria	7-5
View SQL	7-7
Return Data to Excel	7-8
Microsoft Access	7-10
Download Entire Table	7-10
Add Criteria	7-12
View SQL	7-13
View Indexes	7-14
Lotus 1-2-3	7-15
Configure Lotus 1-2-3 to work with Easysoft ODBC	7-15
Download Entire Table	7-16
Add Criteria	7-18
View SQL	7-18
Microsoft Word Mail Merge	7-19

Microsoft Excel

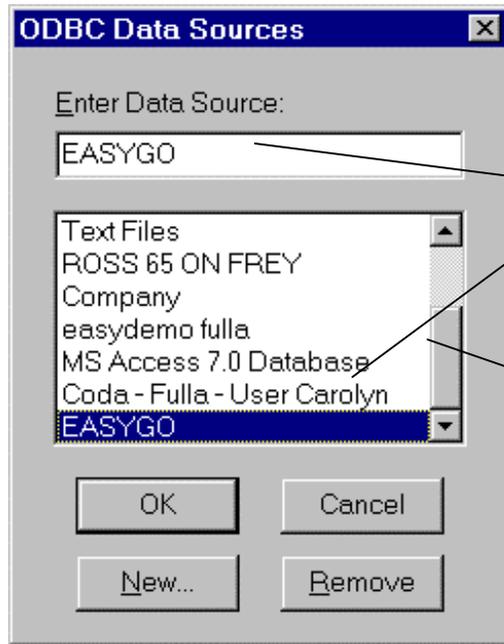
These examples are based on Microsoft Excel, version 7.0, for Windows 95 and Microsoft Query, version 2.0. In these exercises you will download the entire contents of the BUDGET table. Then you will set a criterion so that you see just the Actuals. Finally, you will see how to view the SQL that is sent to the server.

Download Entire Table

1. Start Excel and select an empty worksheet.
2. Select **Get External Data...** from the **Data** menu. Microsoft Query appears with the **Select Data Source** dialog box.



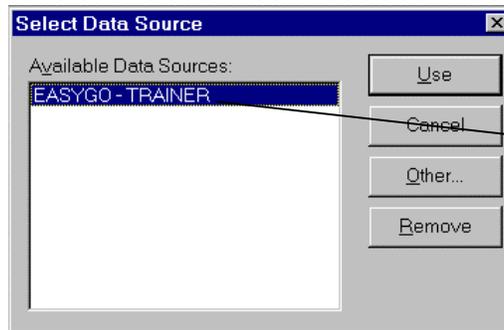
3. Since the data source has not been used before with Microsoft Query, it does not appear in the Available Data Sources list box, and so you have to make it available for use. Click **Other...**. The ODBC Data Sources dialog box appears. (In the general case, if the data source were to appear in the list, you would proceed directly to step 5).



Highlighting a data source causes it to appear in the top box.

Data sources are not in alphabetical order. Use the scroll bar to see all available options.

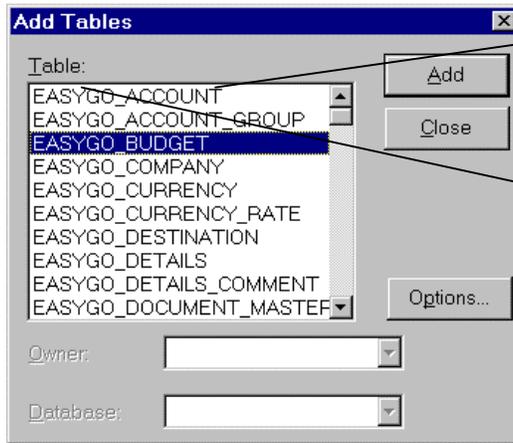
4. Select EASYGO from the Enter Data Source list box and click **OK** when you have done this. Control passes back to the Select Data Source dialog box.
5. Highlight the EASYGO data source.



The name of the data source may be amended automatically - here it is suffixed with the user name.

6. Click the **Use** button, and the Add Tables dialog box appears - this may take a few seconds. (In the background you are able to see changes to the Microsoft Query dialog box).

If you accidentally close the Select Data Source dialog box before selecting a data source, you can open it again by clicking **New Query** from the **File** menu.



The name of the table in the database.

The name of the database. This can be hidden by using the SQL options in the Easysoft ODBC Settings dialog box (refer to "Settings", module 4).

7. Highlight the BUDGET table and click **Add** - the table is added to the Microsoft Query Table pane (see next illustration).
8. Click the **Close** button on the Add Tables dialog box.

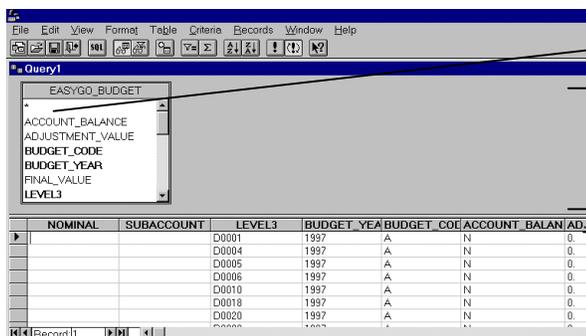


If you cannot see the Table pane (see next illustration) at all, ensure that the **Tables** option under the **View** menu is selected (i.e. there is a tick by it). If isn't selected, click on it to select it. The Table pane should now be visible, and it should contain the BUDGET table.

If you can see a third pane between the Table pane and the Data pane, don't worry. This is the Criteria pane, which is explained later. If you want to hide it for now, de-select it from the **View** menu by clicking to remove the tick.

If you added the wrong table by mistake, then delete it by highlighting one of the fields in the table and then selecting **Delete** from the **Edit** menu.

9. To see the contents of a field, highlight the field using the left mouse button, and drag it into the data pane (or double-click on the field). To select all the columns, drag the asterisk into the data pane. Microsoft Query now looks like this.



Drag the asterisk into the Data pane to see all the data.

The Table pane.

The Data pane.

Add Criteria

At this stage, we've got all the Balance data. Modify the query to view data for just the Actuals.

1. Select the **View** menu, and toggle the **Criteria** option so that there is a tick by it - the Criteria pane should then become visible between the Table and Data panes.



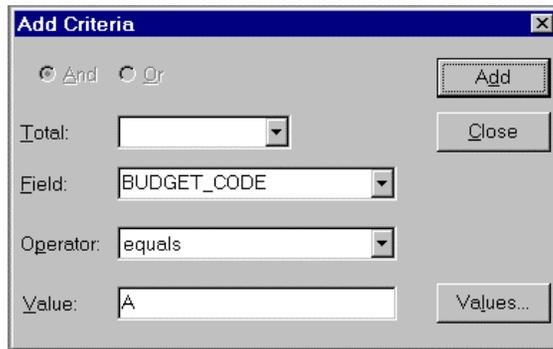
Criteria should be selected so that the Criteria pane is visible.

If this is selected, and the pane is not visible, then place the mouse over the division line between panes, and drag the bar downwards.



Alternatively, you can click the Show/Hide Criteria button to toggle the Criteria pane on and off.

2. Select **Add Criteria...** from the **Criteria** menu option. The Add Criteria dialog box appears (here shown with the first criterion entered).



Set the following criteria using the Add Criteria dialog box:

Field	Operator	Value
(= Column name)		
BUDGET_CODE	equals	'A'

Ignore the Total entry box.

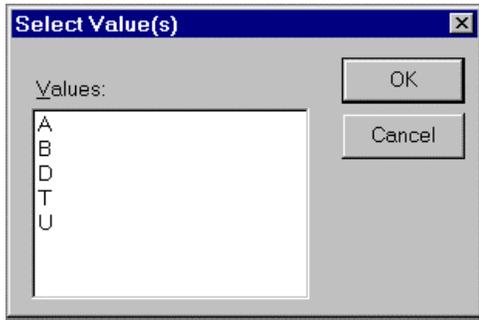
Field in the dialog box is equivalent to the Column name in the CODA table.

Criterion is the criterion we want in the SQL that is generated. To generate this, enter the value shown for Operator. The "equals" operator will be converted to the "=" symbol in SQL.

3. When you have entered the criterion for a field, click **Add**. The criterion that you entered becomes visible in the Criteria pane (see next diagram), and the Add Criteria dialog box is cleared.

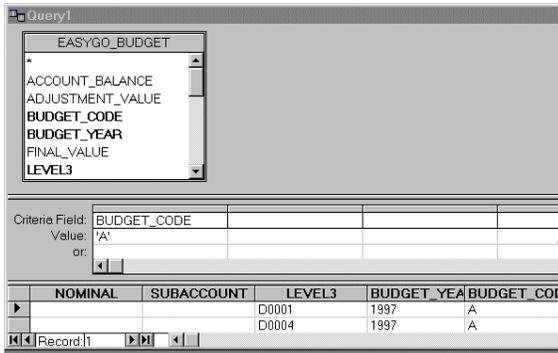


When it is possible to enter a value, you can either enter values directly in the Value field, or you can use the Select Value(s) dialog box.



In this example, we see that there are just five possible years to choose from. You needn't use this method of selecting a criterion value, but the advantage is that you can be sure that the value you choose exists in the database. For example, if you had entered a value of Z in the Add Criteria dialog box, no errors would have been generated (there are none), but you wouldn't have got any data back, because there are no budgets with a code of Z. Another advantage is that you do not need to know whether data is character data or numeric - if it is character data, the required quote marks are automatically added.

4. When you have set all the criteria, press the **Close** button on the **Add Criteria** dialog box. The data which matches the criteria is downloaded.

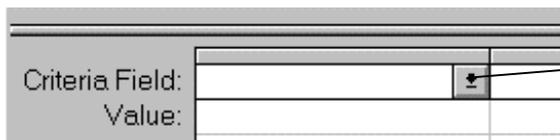


The Criteria pane.

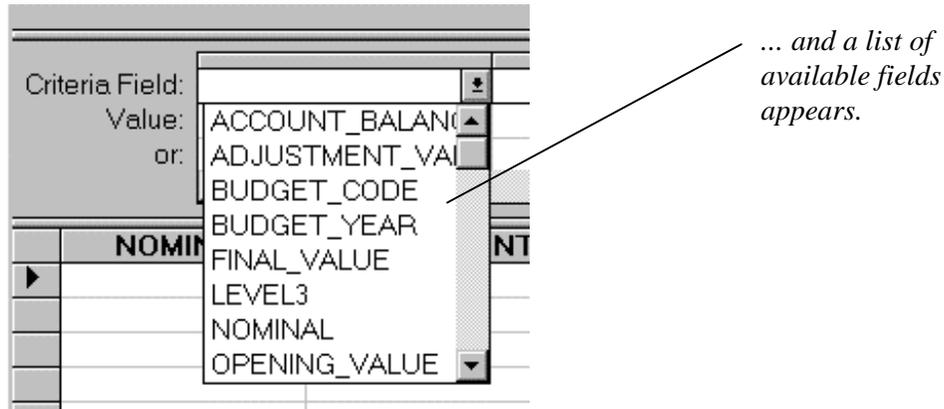
Alternative Methods of Creating Criteria



1. Another method is to place the cursor in the Criteria field. A drop-down button appears. Click on this button to display a list of fields.



Click here...

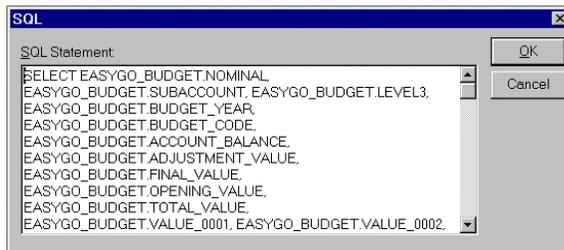


Click on the field for which you want to create a selection criterion (BUDGET_CODE in our case). The field name appears in the Criteria Field. Then type a value in the Value field; to see the Actuals, type 'A' and then move the cursor to another field.

- Another method of criteria selection is to highlight a data value that you want to set as a criterion and then click the Criteria Equals button (). (Obviously, there must be data in the data pane for you to do this).

View SQL

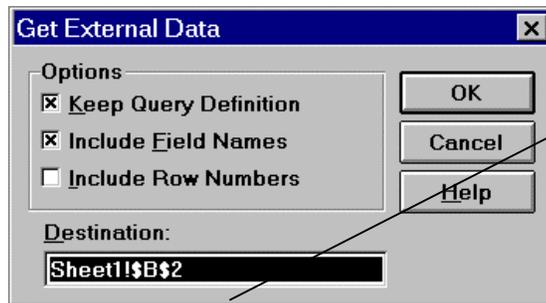
- Look at the SQL that is sent to the database by the query. Press the SQL () button (or select **SQL...** from the **View** menu). The SQL dialog box appears showing the SQL statement that was just used in the query.



- It is possible to modify the SQL directly and then re-send the query. For now, just press the **Cancel** button.

Return Data to Excel

1. To return data to Excel from Microsoft Query, select **Return Data to Microsoft Excel** from the **File** menu. The Get External Data dialog box appears.



Here the Destination list box contains the location of the first column in the first row on the data sheet in which the returned data will be placed.

2. The Keep Query Definition option and the Include Field Names option should be selected (i.e. cross in box).
3. The Destination value is the first cell of the first row in the spreadsheet where the data will be returned. Check that the value it contains is one that you want (if it doesn't, type it in the entry box).
4. Click **OK**. The data is returned to Excel.

	A	B	C	D	E	F	G
1	NOMINAL	SUBACCOUNT	LEVEL3	BUDGET_YEAR	BUDGET_CODE	ACCOUNT_BALANCE	ADJUSTMEN
2			D0001	1997	A	N	
3			D0004	1997	A	N	
4			D0005	1997	A	N	
5			D0006	1997	A	N	
6			D0010	1997	A	N	
7			D0018	1997	A	N	
8			D0020	1997	A	N	

You can now use Excel to create charts, reports etc.

Get External Data options

The following paragraphs explain the Options which are available on dialog box.

Keep Query Definition

This allows you to save the query definition in the *worksheet* (not the same as saving the query in a .QRY file) so that you can update the result set (i.e. the data that is returned to Excel) directly from Excel.

Include Field Names

If you want column headings to be shown, then select this option. The column headings are the field names that are used in the file from which the data is taken.

Include Row Numbers

Row numbers appear in the first column of the returned data; they are not part of the original data.

Destination:

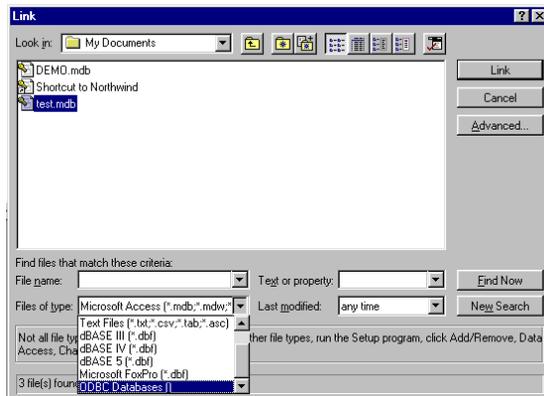
If the Destination: list box contains a range, then the data is returned to those cells specified by the range (hence not all of it may be displayed). If the list box references just one cell, then that cell is the location of the first column of the first row of the returned data. The default value of the list box is taken directly from the Excel worksheet from which the Get External Data dialog box was called. If the Destination is not what you want, change the values in the list box.

Microsoft Access

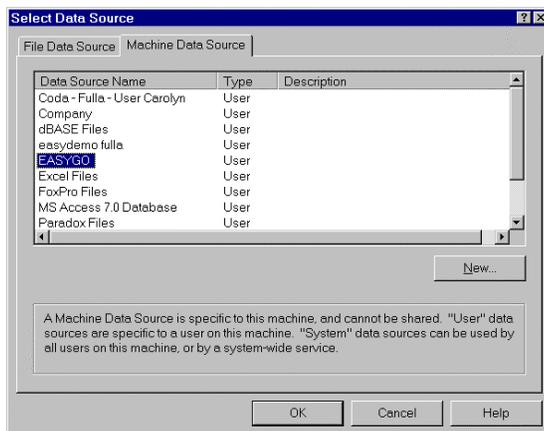
These Access examples are based on Microsoft Access version 7.0, for Windows 95. In these exercises you will download the entire contents of the BUDGET table. Then you will set a criterion so that you see just the Actuals. Next, you will see how to view the SQL that is sent to the server. Finally, you will see how to view the indexes that are available on a table.

Download Entire Table

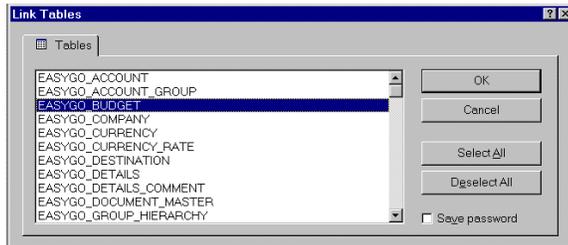
1. Start Microsoft Access and either open an existing database or create a new one.
2. From the **File** menu select **Get External Data** followed by **Link Tables...** The Link dialog box appears.



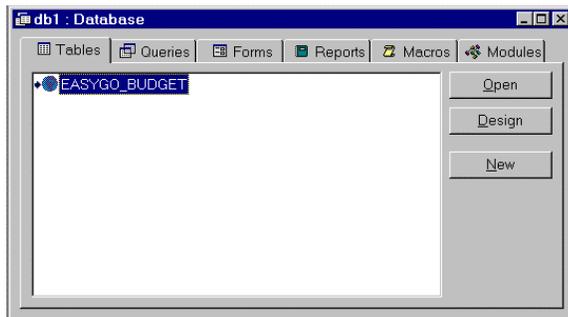
3. Select the database to use, then from the Files of type list box select **ODBC Databases()**. Click the **Link** button. The Select Data Source dialog box appears if you have ODBC version 3.0 installed (with the Microsoft ODBC version 2.5 the SQL Data Sources dialog box appears).



4. Select **EASYGO** as the data source and click **OK** . (If the Easysoft ODBC Login Prompt dialog box appears, click **OK**). The Link Tables dialog box appears.



5. Select the **BUDGET** table and click **OK**. When the operation is complete the Database dialog box contains a list of linked tables - just one in this case.



6. Double clicking on a table displays the data. You can select subsets of the data and add and delete records. The changes you make are updated to the file on the Server which relates to the table you are changing. Double click the **BUDGET** table to see CODA data in Microsoft Access.

NOMINAL	SUBACCOUNT	LEVEL3	BUDGET_YEAR	BUDGET_COD	ACCOUNT_BA	ADJUSTMENT	FINAL_VALUE	OPENII
		D0001	1997	A	N	0	0	
		D0004	1997	A	N	0	0	
		D0005	1997	A	N	0	0	
		D0006	1997	A	N	0	0	
		D0010	1997	A	N	0	0	
		D0018	1997	A	N	0	0	
		D0020	1997	A	N	0	0	
		D0022	1997	A	N	0	0	
		D0023	1997	A	N	0	0	
		D0024	1997	A	N	0	0	
		D0029	1997	A	N	0	0	
		D0030	1997	A	N	0	0	
		D0032	1997	A	N	0	0	
		D0033	1997	A	N	0	0	

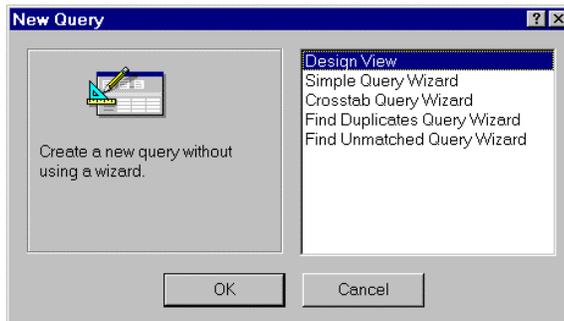
Add Criteria

In this exercise you will select records for a single year from the BUDGET table.

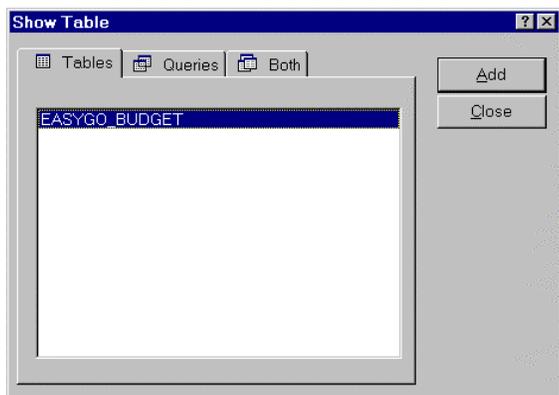
1. Click the **Queries** tab on the Database dialog box.



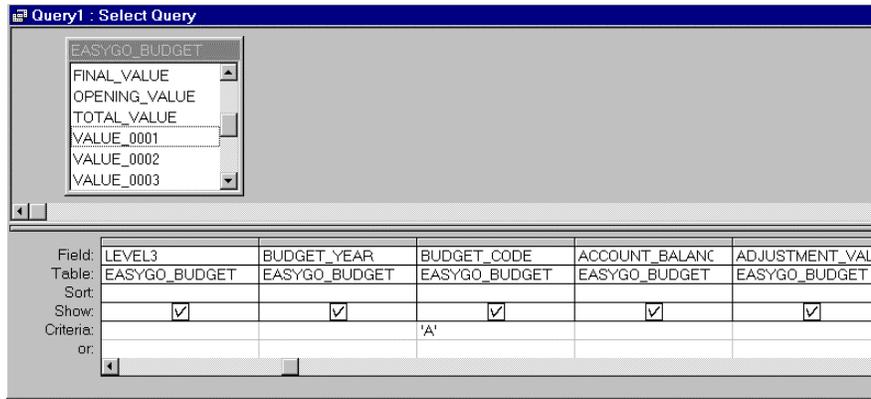
2. Click the **New** button. The New Query dialog box appears.



3. Ensure **Design View** is selected, then click the **OK** button. In the background you can see Microsoft Query, and in the foreground the Show Table dialog box appears. Ensure the **BUDGET** table is highlighted in the Tables tab, and click the **Add** button.



4. The **BUDGET** table appears in the Table pane of Microsoft Query. Click the **Close** button on the Show table dialog box. The Select Query dialog box is now visible.

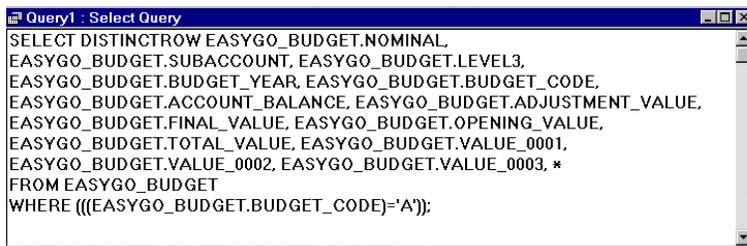


5. Select all the columns from the BUDGET table in the Data pane, and drag them into the Query pane.
6. To select the Actuals, enter the value 'A' in the Criteria row for the BUDGET_CODE field.
7. Select **Run** from the Query menu on the Access menu bar, or click the Run icon () on the toolbar.

The BUDGET data for the Actuals is downloaded.

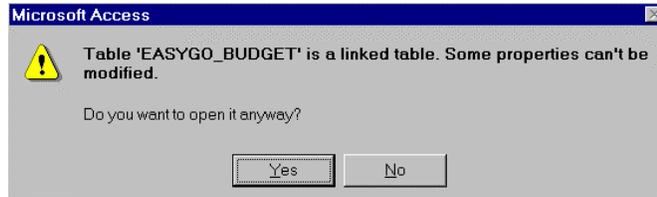
View SQL

1. To see the SQL that is sent, select SQL from the View menu.

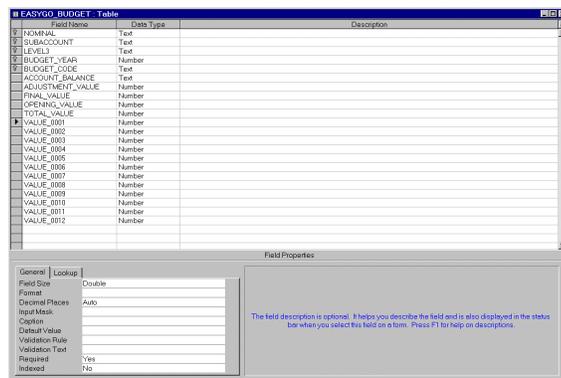


View Indexes

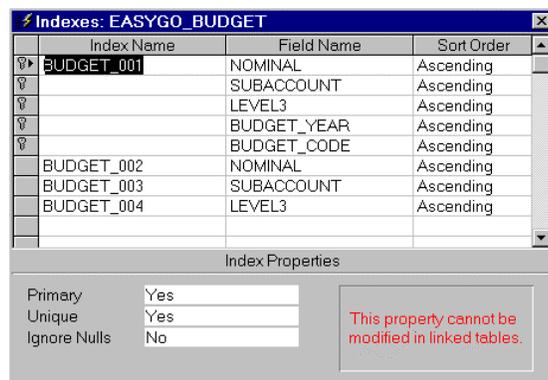
- To view the indexes that are available on a table, select Design mode for the table (e.g. **Tables, Design or View, Table Design**). You will see the following message:



- Click the **Yes** button; the table is displayed in Design mode.



- Select **Indexes** from the **View** menu. The details of the available indexes for the table are displayed.



Lotus 1-2-3

These exercises are based on Lotus 1-2-3 release 5. In these exercises you will first set up Lotus 1-2-3 prior to downloading the entire contents of the BUDGET table. Then you will set a criterion so that you see the Actuals data. Finally, you will see how to view the SQL that is sent to the server.

Lotus 1-2-3 works with 16-bit ODBC only. Therefore, before doing these exercises, set up a 16-bit data source.

Configure Lotus 1-2-3 to work with Easysoft ODBC

Before you can use Easysoft ODBC for CODA with Lotus 1-2-3, you must add a line to the `lotus.bcf` file. Use a text editor, or if you use a word-processor, ensure that you save the document as a text file.

1. Edit the **lotus.bcf** file, which will probably be in the default directory `\windows\lotusapp\dataens`.

If the file is not in this directory, search for its location using the Windows Explorer by selecting the following sequence: **Tools, Find, Files or Folders**.

2. Add a line to the bottom of the **lotus.bcf** file containing the following:
`DN="ODBC" DD="ODBC" DL="DLODBC" ;`

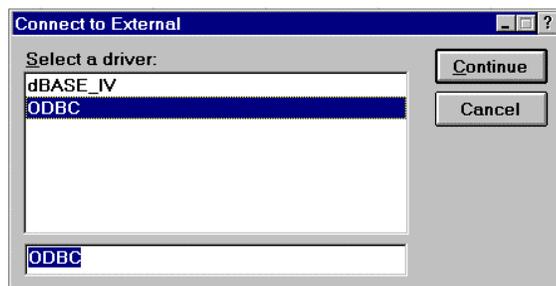
If this line already exists in the file, there is no need to repeat it.

Download Entire Table

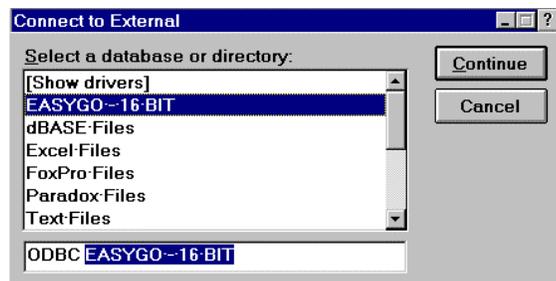
1. Start Lotus and create a new worksheet if necessary.
2. Select the **Tools** menu, and from this select **Database**, followed by **New Query...** The New Query Assistant dialog box appears.



3. Click **External...** The Connect to External dialog box appears.

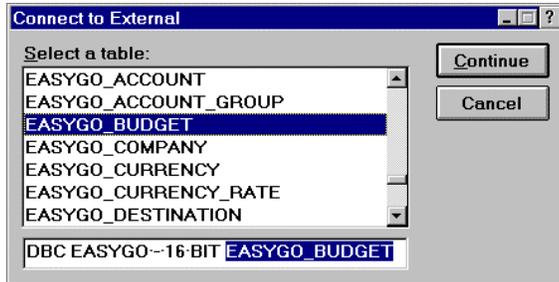


4. Highlight ODBC, then click **Continue**. A list of data sources appears.



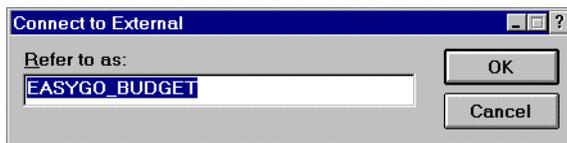
Highlight the data source you want to use, and then click the **Continue** button.

- A list of tables for each of the available databases appears.

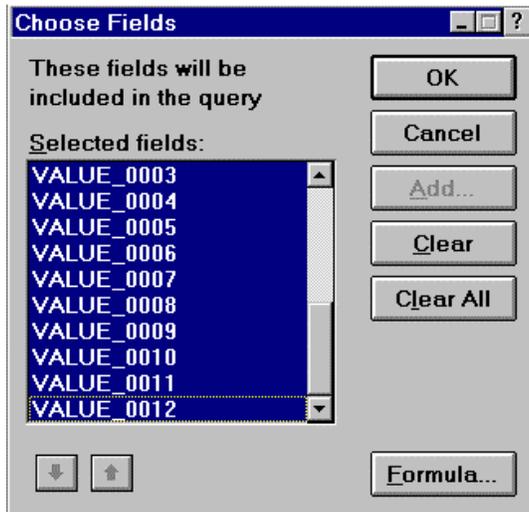


Highlight the BUDGET table, and click the **Continue** button.

- The Connect to External - Refer to as: option appears. Click **OK**.



- The New Query Assistant re-appears. Click **Choose Fields...**
- The Choose Fields dialog box appears.



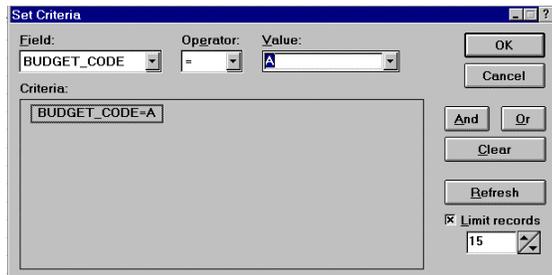
Highlight all the fields and click **OK**.

- The New Query Assistant re-appears. Click **OK** to download the data.

	A	B	C	D	E	F	G	H	I
1	NOMINAL	SUBACCOLLEVEL3	BUDGET_Y	BUDGET_C	ACCOUNT	ADJUSTME	FINAL_VAL	OPENING_Y	
2			D0001	1997	A	N	0	0	0
3			D0004	1997	A	N	0	0	0
4			D0005	1997	A	N	0	0	0
5			D0006	1997	A	N	0	0	0

Add Criteria

1. Place the cursor in the downloaded data.
2. The Query menu appears on the toolbar. From this, select **Set Criteria...** The Set Criteria dialog box appears.

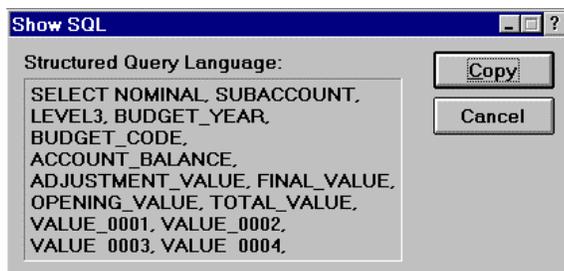


3. Click the **Clear** button to remove any default criteria.
4. Select the field, comparison operator and value from the three list boxes. The criteria you set appears in the Criteria window. Click **OK**.
5. The data in the spreadsheet changes to reflect the new criteria.

View SQL

This option is limited. You cannot change the SQL, and you may not even be able to view it in its entirety in Lotus 1-2-3.

1. Place the cursor in the downloaded data.
2. The Query menu appears on the toolbar. From this, select **Show SQL...** The Show SQL dialog box appears.



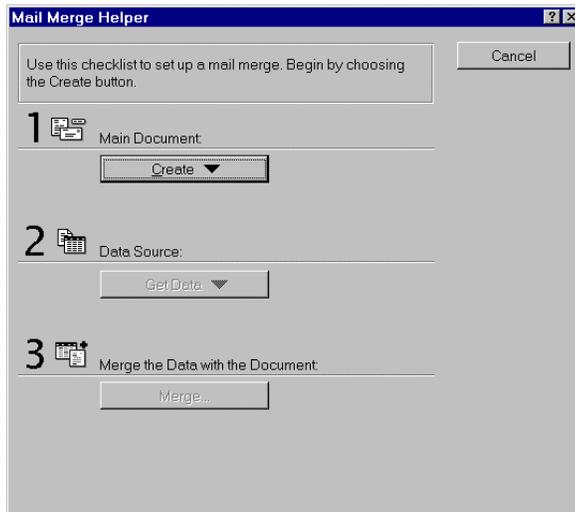
3. There are no scroll bars in this dialog box, so you may not be able to see all the SQL. However, you can copy it to the clipboard by clicking the **Copy** button. Then paste it into a text editor or word processor to view it. When you click either the **Copy** button or the **Cancel** button, the dialog box closes.

Microsoft Word Mail Merge

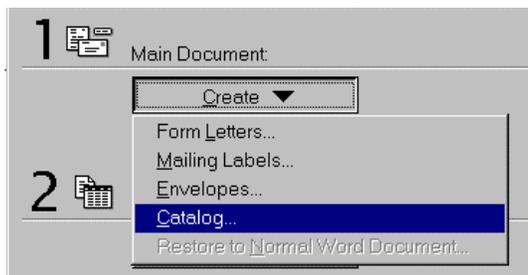
This example is based on Microsoft Word 7.0 for Windows 95.

Mail merge allows you to create documents which contain information imported from a data source. This example uses the currency data; you will see how to create a list of all currency rates.

1. Open a new Word document.
2. Select **Tools, Mail Merge...** The Mail Merge Helper appears.



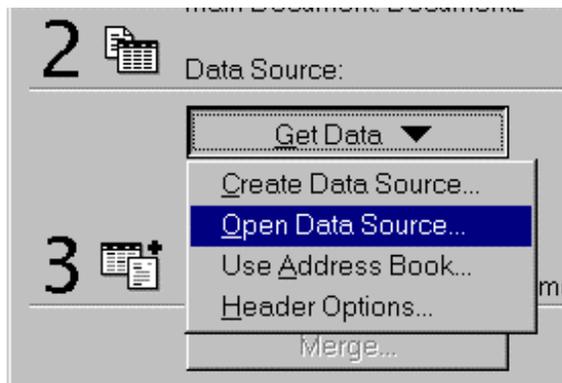
3. Create a document for the merged data - click the **Create** button and select the Catalog option



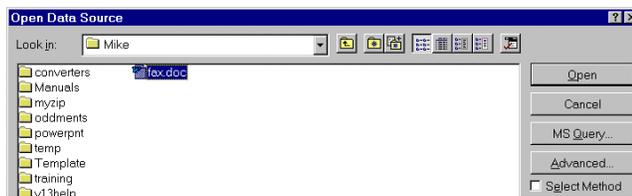
4. You are asked whether you want to use the active window or a new one. Select **Active Window** (you already created a new document in step 1 of this exercise).



5. Step two of the Mail Merge Helper is now available for use. Click the **Get Data** button

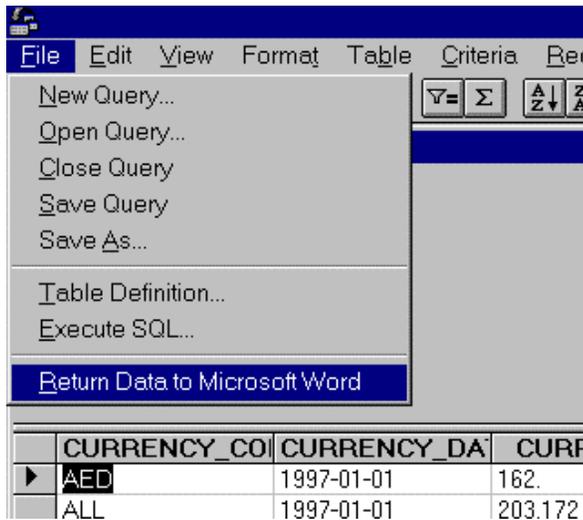


6. Select the **Open Data Source...** option. The Open Data Source dialog box appears.

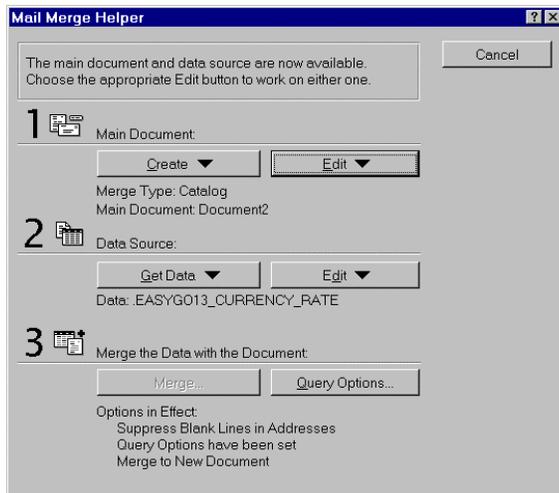


7. Click the **MS Query...** button. This starts Microsoft Query. You have already seen how to use this, so the details are not repeated. If you need to refresh your memory, look at Microsoft Excel at the start of this module.
8. Select the EASYGO data source.
9. Add the CURRENCY_RATE table to the query.
10. Insert all the fields into the Data Pane.

11. To return the data to Word, select the **Return Data to Microsoft Word** option from the **File** menu.



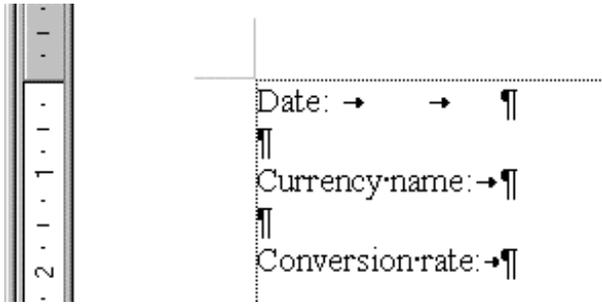
12. You will notice that the Merge option of Step 3 is not available. Click the **Close** button on the Mail Merge Helper.



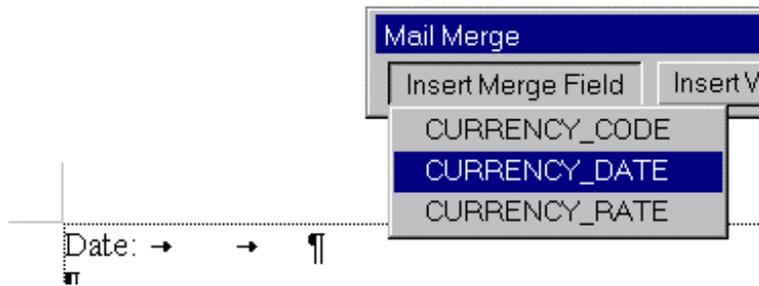
13. The Mail Merge toolbar is now available.



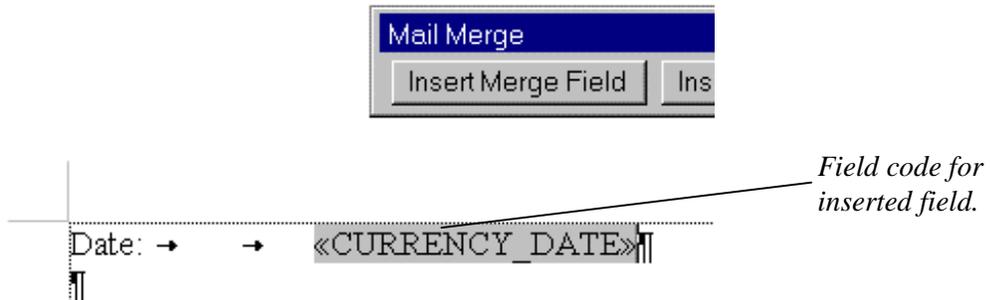
14. Type some text in the word document. Place the cursor where you want the merged data to appear. For example:



15. Click the **Insert Merge Field** option on the Mail Merge toolbar and select the field you want to insert.

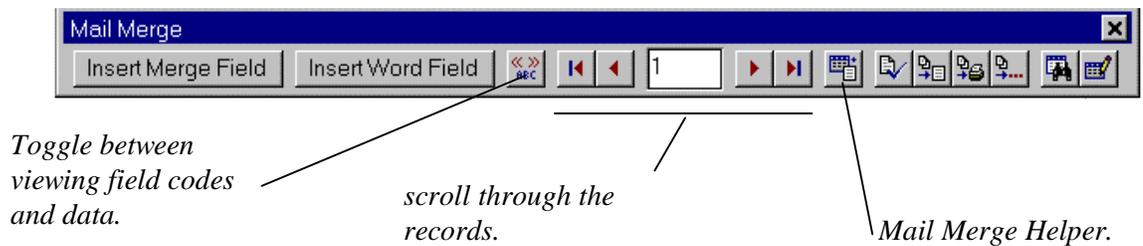


16. The field is inserted into the document.



17. Repeat this process for as many of the fields as you require.

18. Use the Mail Merge toolbar to view the data.



Note: to set criteria and view the SQL, use the Query Options button on the Mail Merge Helper to gain access to Microsoft Query.

8. CODA Training Company

This module describes the CODA Training Company. There is nothing to learn; the contents of this module are for reference purposes when you do the practical exercises.

In this module you will see

- ✦ what the training company is
- ✦ its overall structure (with the chart of accounts)

Contents

Easysoft Travel Company	8-2
Chart of Accounts	8-2
Nominals, Subaccounts and Level3s	8-5
Account Group Hierarchy	8-8
List of Currencies	8-11

Easysoft Travel Company

The example company that we use is a small independent travel agency which is based in the city of Leeds. It has a total of four members of staff: one manager, one full-time YTS employee and two part-time travel consultants.

Holidays are purchased from various tour operators, and these are then sold to the public. When a sale is made, a commission of 10% is received. For exemplary purposes, VAT is not paid on holidays.

Fixed assets were calculated in the opening period of 1997, and the company commenced trading with £50,000 capital on 1st January 1997.

The fixed assets are desks, chairs, VDUs, telephones, a fax machine, a large brochure rack and a company car (a Ford Fiesta).

Salaries are paid to the employees on the last day of each month. Rent and utilities (telephone, electricity, etc.) are paid quarterly.

Budgeting has been performed on the nominal holidays in Sales Analysis at subaccount (brochure) level for the first three periods of 1997.

The chart of accounts for Easysoft Travel Company is shown below.

Chart of Accounts

1000	Fixed Assets
1100	Fixtures & Fittings
1110	Cost
1120	Depreciation
1200	Buildings
1210	Cost
1220	Depreciation
1300	Computer Equipment
1310	Cost
1320	Depreciation
1400	Motor Vehicles
1410	Cost
1420	Depreciation
1500	Office Equipment
1510	Cost
1520	Depreciation

2000	Current Assets		
		2100	Sales
		2200	Bank
			2210 Current Account
			2220 Deposit Account
		2300	Cash
3000	Capital		
		3100	Capital
		3200	Retained Earnings
4000	Long Term Liabilities		
		4100	Loans
		4200	Suspense
5000	Current Liabilities		
		5100	Purchases
		5200	Payroll
		5300	VAT
			V0001 Standard Rate
			V0002 Zero Rated
			V0003 Exempt
6000	Sales Analysis		
		6100	Holidays
		6200	Insurance
		6300	Travellers Cheques
		6400	Foreign Currency
7000	Purchase Analysis		
		7100	Holidays
		7200	Insurance
		7300	Travellers Cheques
		7400	Foreign Currency
8000	Expenses		
		8100	Office Rental
			8110 Rent
			8120 Service Charge
		8200	Payroll
		8300	Motor

- 8400 Utilities
 - 8410 Communication
 - 8411 Telephone
 - 8412 Fax
 - 8420 Electricity
 - 8430 Water
 - 8440 Gas

- 8500 Depreciation
 - 8510 Fixtures & Fittings
 - 8520 Buildings
 - 8530 Computer
 - 8540 Motor Vehicles
 - 8550 Office Equipment

- 8600 Professional Fees
 - 8610 Accountant

- 8700 Finance
 - 8710 Bank Charges

- 8800 Office Supplies

- 8900 Miscellaneous

NOTES

Accounts 1000 to 5999 represent balance sheet items, and accounts 6000 to 8999 represent trading items.

2000 Current Assets. Customers are input as Names and Addresses, with a code of Sn , where n represents a four character number and S stands for Sales.

5000 Current Liabilities. Suppliers are input as Names and Addresses, with a code of Hn , where n represents a four character number and H stands for the Holiday company.

5300 VAT. VAT rates are input as subaccounts.

6100 Holidays. Brochures are input as subaccounts, with a code of Bn , where n represents a four character number and B stands for Brochure. Destinations are input as Level3s, with a code of Dn , where n represents a four character number and D stands for Destination.

6200 Insurance. The type of insurance is input as a subaccount.

7100 Holidays. Brochures are input as subaccounts, with a code of Bn , where n represents a four character number and B stands for Brochure. Destinations are input as Level3s, with a code of Dn , where n represents a four character number and D stands for Destination.

7200 Insurance. The type of insurance is input as a subaccount.

Nominals, Subaccounts and Level3s

This is the data that is contained in the Nominals, Subaccounts and Level3s of the training company.

7-Mar-97	Easysoft Travel Company		CODA-Financials
v7.20.057			
11:13:50	Nominal details maintenance - Directory		Carolyn Richardson
C002			
1100	Fixtures & Fittings	1110	Costs
1120	Depreciation	1200	Buildings
1210	Buildings Costs	1220	Depreciation
1300	Computer Equipment	1310	Comp Equip. costs
1320	Comp Equip. Deprn.	1400	Motor Vehicles
1410	Motor Vehicles Costs	1420	Motor Vehicles Deprn
1500	Office Equipment	1510	Office Equip. Costs
1520	Office Equip. Depr.	2100	Sales
2210	Current Account	2220	Deposit Account
2300	Cash	3100	Capital
3200	Retained Earnings	4100	Loans
4200	Suspense	5100	Purchases
5200	Payroll	5300	VAT Control Account
6000	Sales Analysis	6100	Holiday Sales
6200	Insurance Sales	6300	Travellers Cheques
6400	Foreign Currency	7000	Purchase Analysis
7100	Holiday Purchases	7200	Insurance Purchases
7300	Travellers Cheques	7400	Foreign Currency
8000	Expenses	8110	Office Rent
8120	Service Charge	8200	Salaries
8300	Motor Vehicles	8410	Communications
8411	Telephone	8412	Fax
8420	Electricity	8430	Water
8440	Gas	8510	Fixtures & Fittings
8520	Buildings	8530	Computer Equipment
8540	Motor Vehicles	8550	Office Equipment
8610	Accountant	8710	Bank Charges
8800	Office Supplies	8900	Miscellaneous

-Mar-97	Easysoft Travel Company		CODA-Financials
v7.20.057			
1:15:24	Sub-account maintenance - Directory		Carolyn Richardson
C002			
B0001	Faraway Shores	B0002	Ski-ing
B0003	Young At Heart	B0004	Price Breakers
B0005	City Breaks	B0006	World Wide
B0007	Flight Only	B0008	A La Carte
B0009	Weddings In Paradise	B0010	All Inclusive
B0011	Summer Sun	B0012	Winter Sun
B0013	Airtours Summer Sun	B0014	Airtours Winter Sun
B0015	Airtours Far & Away	B0016	Weddings of Fantasy
B0017	All Inclusive	B0018	Airtours Flight Only
B0019	Airtours Florida	B0020	Airtours Cruises
B0021	Golden Years	B0022	Airtours Fly-Drive
B0023	Distant Dreams	B0024	Dream Weddings
B0025	Cosmos Sandals	B0026	Cosmos Summer Sun
B0027	Cosmos Winter Sun	B0028	Kuoni World Wide

B0029	Australia&Newzealand	B0030	Caribbean & All Inc
B0031	Kuoni Switzerland	B0032	Kuoni Cruise & Stay
B0033	Kuoni Safaris	B0034	Malidive & SriLanka
B0035	Kuoni Weddings	B0036	Kuoni South Africa
B0037	Kuoni Three	B0038	America & Canada
B0039	Ski Switzerland	B0040	Cresta Cities
B0041	Cresta Rome	B0042	Cresta Italy
B0043	Cresta Ireland	B0044	Cresta France
B0045	Disneyland Paris	B0046	Cresta Eurostar
B0047	Golf in France	B0048	M.of Italy Beaches
B0049	M. Italy Cities	B0050	Magic of Italy Lakes
B0051	Italy Countryside	B0052	Italy Islands
B0053	Getaway Breaks	B0054	Christmas & New Year
B0055	Footloose	B0056	Mexico & USA W/sun
B0057	Trek America	B0059	Live The Dream
B0060	Hawaii	B0061	Las Vegas
B0062	Tennessee	B0063	Dest. New England
B0064	A.A. Fly-Drive	B0065	A.A. Coach Tours
B0066	A.A.Motor Homes	B0067	A.A. Hotels & Apts
B0068	Bales Egypt	B0069	Bales Worldwide
B0070	Florida & Caribbean	B0071	Weddings
B0072	Flight Only	B0073	Virgin Fly-Drive
B0074	P & O Ferries	B0075	P & O Hovercraft
B0076	P & O Cruises		
D0001	London	D0003	Edinburgh
D0002	Leeds	D0005	Paris
D0004	Blackpool	D0007	Calais
D0006	Cannes	D0009	Chamonix - The Alps
D0008	Bordeaux	D0011	San Remo - Italy
D0010	Rome - Italy	D0013	Sorrento - Italy
D0012	Venice - Italy	D0015	Lake Maggiore
D0014	Lake Como - Italy	D0018	Barcelona - Spain
D0017	Benidorm	D0020	Torremolinos
D0019	Benalmedina	D0022	Magaluf - Majorca
D0021	Andorra	D0024	Las Palmas
D0023	Playa De Las America	D0027	Tunis - Tunisia
D0026	Gibraltar	D0029	Amsterdam
D0028	Bangul - The Gambia	D0031	Berlin - Germany
D0030	Athens - Greece	D0033	Bruges - Belgium
D0032	Boston - USA	D0035	Budapest - Hungary
D0034	Brussels - Belgium	D0037	Cologne - Germany
D0036	Chicago - USA	D0039	Cork - Ireland
D0038	Copenhagen - Denmark	D0041	Dublin - Ireland
D0040	Dubai	D0043	Granda - Spain
D0042	Geneva - Switzerland	D0045	Helsinki - Finland
D0044	Guernsey	D0047	Lisbon - Portugal
D0046	Istanbul - Turkey	D0049	Lyon - France
D0048	Lucerne	D0051	Madrid - Spain
D0050	Madeira	D0053	Marrakech - Morrocco
D0052	Malta	D0055	Monte Carlo - France
D0054	Milan - Italy	D0057	Munich - Germany
D0056	Montreal - Canada	D0059	Nice - France
D0058	New York - USA	D0061	Prague
D0060	Oslo - Norway	D0063	Rio De Janeiro
D0062	Reykjavik - Iceland	D0065	Salzburg - Austria
D0064	Salamanca - Spain	D0067	Stockholm - Sweden
D0066	Seville - Spain	D0069	Toulouse - France
D0068	Toronto - Canada	D0071	Warsaw - Poland
D0070	Vienna - Austria	D0073	Daytona
D0072	Washington D.C.	D0075	Tampa - Florida
D0074	Orlando - Florida	D0077	Miami - Florida
D0076	Clearwater Bay	D0079	Los Angeles - USA
D0078	Las Vegas - USA	D0081	Siera Nevada - USA
D0080	Denver - Colerado	D0083	Cancun - Mexico
D0082	New Orleans - USA		

D0084	Playa Dorada	D0085	Montego Bay
D0086	Bridge Town	D0087	Castries - St. Lucia
D0088	Eng. Harbour Antigua	D0089	Honolulu - Hawaii
D0090	Luxor - Egypt	D0091	Ari Atol - Maldives
D0092	Phuket - Thailand	D0093	Goa - India
D0094	Columbo - Sri Lanka	D0095	Cairnes - Australia
D0096	Sydney - Australia	D0097	Auckland
D0098	Beijing - China	D0099	Hong Kong
D0100	Grand Baie	D0101	Mombasa - Kenya
D0102	Cape Town	D0103	Trinidad - CUBA
D0104	Montego Bay	D0105	Hamilton - Bermuda
D0106	Paradise Island		
V0001	VAT - Standard Rate		
V0002	VAT - Zero Rated	V0003	VAT - Exempt

-Mar-97 Easysoft Travel Company

v7.20.057

1:30:25 Name & address maintenance - Directory C002

CODA-Financials

Carolyn Richardson

H0001	Thomson Holidays Ltd	H0002	Airtours
H0003	COSMOS	H0004	Kuoni Travel
H0005	Cresta Holidays	H0006	Magic of Italy
H0007	Best Western	H0008	Trek America
H0009	N American Travel	H0010	American Airlines
H0011	Bales Tours Ltd	H0012	Virgin Holidays Ltd
H0013	P&O		
P0001	DTZ Debenham Thorpe		
P0002	BT British Telecom	P0003	Brixton Motors
P0004	Mr A Cornwell		
S0001	Carolyn Richardson		
S0002	John Kos	S0003	Jason Crummack
S0004	Caroline Welburn	S0005	Guy Smithson
S0006	Karl Berlin	S0007	Keeley Gothard
S0008	Ian Naylor	S0009	Nick Gorman
S0010	Mike Umwalla	S0011	Martin Greaves
S0012	Micheal Carter	S0014	Tony Chan
S0015	Richard Spencer	S0016	John Beeby
S0017	Mark Morley		

Account Group Hierarchy

<u>Group</u>	<u>Group Title</u>	<u>Group Type</u>
Relations		
ALL	All Season Euro Destinations	Subaccount
Parents		
EUHOLS	European Holidays	(Group)
Children		
D0005	Paris - France	
(Subaccount)		
D0017	Benidorm - Costa Brava	
(Subaccount)		
D0018	Barcelona - Spain	
(Subaccount)		
D0020	Torremolinos - Costa Del Sol	
(Subaccount)		
D0022	Magaluf - Majorca	
(Subaccount)		
D0024	Las Palmas - Gran Canaria	
(Subaccount)		
D0027	Tunis - Tunisia	
(Subaccount)		
D0030	Athens - Greece	
(Subaccount)		
D0052	Malta	
(Subaccount)		

<u>Group</u>	<u>Group Title</u>	<u>Group Type</u>
Relations		
EUHOLS	European Holidays	Subaccount
Children		
WINTER	Winter Holidays	(Group)
SUMMER	Summer European Destinations	(Group)
ALL	All Season Euro Destinations	(Group)

<u>Group</u>	<u>Group Title</u>	<u>Group Type</u>
Relations		
SUMMER	Summer European Destinations	Subaccount
Parents		
EUHOLS	European Holidays	(Group)
Children		
D0004	Blackpool - United Kingdom	
(Subaccount)		
D0006	Cannes - France	
(Subaccount)		
D0013	Sorrento - Italy	
(Subaccount)		
D0026	Gibraltar	
(Subaccount)		
D0051	Madrid - Spain	
(Subaccount)		

D0054	Milan - Italy
(Subaccount)	
D0055	Monte Carlo - France
(Subaccount)	
D0059	Nice - France
(Subaccount)	

<u>Group</u>	<u>Group Title</u>	<u>Group Type</u>
Relations		
WALL	All Season World Wide Dest's	Subaccount
Parents		
WWHOLS	World Wide Holiday Destination	(Group)
Children		
D0073	Daytona Beach - Florida	
(Subaccount)		
D0077	Miami - Florida	
(Subaccount)		
D0078	Las Vegas - USA	
(Subaccount)		
D0085	Montego Bay - Jamaica	
(Subaccount)		

<u>Group</u>	<u>Group Title</u>	<u>Group Type</u>
Relations		
WINTER	Winter Holidays	Subaccount
Parents		
EUHOLS	European Holidays	(Group)
Children		
D0009	Chamonix - French Alps	
(Subaccount)		
D0021	Andorra - The Pyrenees	
(Subaccount)		
D0042	Geneva - Switzerland	
(Subaccount)		

<u>Group</u>	<u>Group Title</u>	<u>Group Type</u>
Relations		
WSUMMER	Summer World Wide Destinations	Subaccount
Parents		
WWHOLS	World Wide Holiday Destination	(Group)
Children		
D0058	New York - USA	
(Subaccount)		
D0063	Rio De Janeiro - Brazil	
(Subaccount)		
D0079	Los Angeles - USA	
(Subaccount)		

<u>Group</u>	<u>Group Title</u>	<u>Group Type</u>
Relations		

WWHOLS	World Wide Holiday Destination	Subaccount
Children		
WWINTER	Winter World Wide Destinations	(Group)
WSUMMER	Summer World Wide Destinations	(Group)

<u>Group</u>	<u>Group Title</u>	<u>Group Type</u>
Relations		
WWINTER	Winter World Wide Destinations	Subaccount
Parents		
WWHOLS	World Wide Holiday Destination	(Group)
Children		
D0081	Siera Nevada - USA	
(Subaccount)		
D0092	Phuket - Thailand	
(Subaccount)		
D0093	Goa - India	
(Subaccount)		
D0096	Sydney - Australia	
(Subaccount)		

List of Currencies

CURRENCY CODE	TITLE	CURRENCY CODE	TITLE
AED	United Arab Emirate Dirham	LBP	Lebanese Pound
ALL	Albanian Lek	LKR	Sri Lanka Rupee
ATS	Austrian Schilling	LSL	Lesotho Loti
AUD	Australian Dollar	LTL	Lithuanian Lita
BDT	Bangladeshi Taka	LVL	Latvian Lat
BEF	Belgian Franc	MAD	Moroccan Dirham
BGL	Bulgarian Lev	MRO	Mauritanian Ouguiya
BHD	Bahraini Dinarling	MTL	Maltese Lira
BIF	Burundi Franc	MUR	Mauritius Rupee
BRL	Brazilian Real	MWK	Malawi Kwacha
BWP	Botswana Pula	MXP	Mexican Peso
CAD	Canadian Dollar	MYR	Malaysian Ringgit
CHF	Swiss Franc	MZM	Mozambique Metical
CLP	Chilean Peso	NAD	Namibian Dollar
CNY	Yuan (Chinese) Renminbi	NGN	Nigerian Naira
COP	Colombian Peso	NLG	Dutch Guilder
CSK	Czech Koruna	NOK	Norwegian Kroner
CVE	Cape Verde Escudo	NZD	New Zealand Dollar
CYP	Cyprus Pound	OMR	Omani Rial
DEM	German Mark	PEN	Peruvian Nuevo Sol
DKK	Danish Krone	PHP	Philippine Peso
DZD	Algerian Dinar	PLZ	Polish Zloty
ECS	Ecuador Sucre	PTE	Portugese Escudo
EEK	Estonian Krone	PYG	Paraguay Guarani
EGP	Egyptian Pound	QAR	Qatari Rial
ESP	Spanish Peseta	ROL	Romanian Leu
FIM	Finnish Markka	SAR	Saudi Riyal
FJD	Fiji Dollar	SBD	Solomon Islands Dollar
FRF	French Franc	SCR	Seychelles Rupee
GBP	British Pound	SEK	Swedish Krona
GHC	Ghanaian Cedi	SGD	Singapore Dollar
GMD	Gambian Dalasi	SIT	Slovenian Tolar
GRD	Greek Drachma	SKK	Slovak Koruna
HKD	Hong Kong Dollar	SUR	Russian Rouble
HRK	Croatian Kuna	SZL	Swaziland Lilangeni
HUF	Hungarian Forint	THB	Thai Baht
IDR	Indonesian Rupiah	TRL	Turkish Lire
IEP	Irish Punt	TWD	Taiwan Dollar
ILS	Israeli New Shekel	UAK	Ukraine Karbovanets
INR	Indian Rupee	UGS	Uganda Shilling
IRR	Iranian Rial	USD	US Dollar
ISK	Iceland Krona	VEB	Venezuelan Bolivar
ITL	Italian Lire	VUV	Vanuatu Vatu
JOD	Jordian Dinar	XAF	CFA Franc BEAC
JPY	Japanese Yen	XEU	ECU
KES	Kenyan Shilling	XPD	Palladium
KRW	Korean Won	XQU	Gold (oz.)
KZT	Kazakhstan Tenge	ZAR	South African Rand
		ZWD	Zimbabwe Dollar

9. CODA Exercises

In this module you will

- ✦ use Easysoft ODBC for CODA to download and upload CODA data
- ✦ use the Easysoft Excel macro for CODA to automate the tasks you perform

Contents

General Information	9-2
Tips and Reminders	9-2
1. Master File Maintenance	9-6
Task 1.1 : Produce a list of all Ledgers	9-6
Task 1.2 : List Ledgers and their Control Account	9-7
Task 1.3 : Produce a directory of all Nominals	9-8
Task 1.4 : Produce a directory of the Sales Nominal (2100)	9-9
Task 1.5 : Produce a directory of the Sales Analysis Nominals	9-10
Task 1.6 : List SUBACCOUNTS with a destination of Blackpool or Bermuda (Hamilton)	9-11
Task 1.7 : List Name and address Subaccounts and the LEDGER they belong to	9-13
Task 1.8 : List existing currencies and currency rates	9-15
Task 1.9 : Find out how many customers there are in the database	9-19
Task 1.10 : Produce a list of CODA Account Groups	9-21
Task 1.11 : Produce a list of the Group Hierarchy to one level	9-22
Task 1.12 : Produce a list of the Group Hierarchy to two levels	9-24
Task 1.13 : Repeat previous tasks using the Easysoft Excel macro for CODA	9-27
2. Budgets	9-28
Task 2.1 : Download budget information, change it and upload	9-28
Task 2.2 : Use the Macro to download and upload budgets	9-31
3. Reporting	9-33
Task 3.1 : Produce a Trial Balance	9-33
Task 3.2 : Produce a Trial Balance using the macro	9-35
Task 3.3 : Produce an Actual Budget Variance report using pivot tables	9-36
Task 3.4 : Produce an Actual Budget Variance report using the macro	9-39
Task 3.5 : Produce a Profit and Loss Account	9-40
Task 3.6 : Use the macro to produce a Profit and Loss Account	9-42
Task 3.7 : Produce a Balance Sheet for the year so far	9-43
Task 3.8 : Use the macro to produce a Balance Sheet	9-46
Task 3.9 : Produce an Ageing Summary	9-47
Task 3.10 : Use the macro to Produce an Ageing Summary	9-49
4. Input and Intraday	9-50
Task 4.1 : Invoice Upload	9-50
Task 4.2 : Invoice Download	9-52
Task 4.3 : Upload Salary Journal	9-53
Task 4.4 : Download Salary Journal	9-54

General Information

The purpose of the exercises is to show you an easy way to retrieve and update CODA data. We do not concern ourselves with the manipulation and formatting of data. In these exercises, you will use Microsoft Excel to connect to the data source which you previously set up.

All of the exercises follow the same format; they have been designed to introduce you quickly to the concepts of using Easysoft ODBC for CODA. There are four sections, Master File Maintenance, Budgets, Reporting and Input and Inquery. Each section contains several tasks each of which is followed by a series of steps and the results expected. In cases where many rows are returned, then only a fragment of the result is shown. As you progress further you should try to use the steps less and less. Also, where appropriate, there are additional exercises which can be used to take the tasks further.

Tips and Reminders

The best approach to solving the exercises is to keep everything as simple as possible.

You cannot update data directly using Excel (it is possible with the use of a macro). However, since Excel uses Microsoft Query to access the data, you can upload data from Query, provided you have not returned to Excel.

In Microsoft Query, to add a column to the Data Pane from a table which is in the Table Pane, you can either highlight the column and drag it into the Data Pane, or you can double-click on the column.

Each time that you return data to Excel from Query or each time you change a query, Excel reconnects to the data source.

To update data, you should enable the editing option in Microsoft Query. To do this, select **Records, Allow Editing**.

The “%” character can be used as a wildcard in queries. It represents zero or more characters.

To make a report more comprehensible, you can change the names of columns that are returned to the spreadsheet when they are not clear. To do this, double click on the column heading and type in the new column name.

If a record is inserted, deleted or updated, then Query re-reads the entire file, which can be a slow process.

Connecting to the Datasource

These steps are the same for all exercises. This section briefly lists the steps - for full details with screen shots, refer to module 7.

1. To start Excel, click **Start** then select **Excel** from the **Programs** option (or double click on the Excel icon).
2. Select **Data, Get External Data...** The Select Data Source dialog box appears.
3. a) Highlight your data source and click **Use**.
 b) If the data source does not appear as a choice, click **Other**, select the data source you require and then click **OK** to return to the Select Data Source dialog box. Click **Use**. Next time you use the Select Data Source dialog box, this data source will appear in the list box.
4. The Add Tables dialog box appears. You are presented with a list of available tables. Highlight the table you want and click **Add**. The table is added to the Table Pane.
5. Repeat the previous step if necessary, then press **Close**.

Adding Criteria (method 1)

The steps for adding criteria are the same for all exercises. Here the steps are briefly listed - for full details with screen shots, refer to module 7.

1. Select the **View** menu, and toggle the **Criteria** option so that there is a tick by it - the Criteria pane should then become visible between the Table and Data panes.



Alternatively, you can click the Show/Hide Criteria button to toggle the Criteria pane on and off.

2. Select **Add Criteria...** from the **Criteria** menu option. The Add Criteria dialog box appears. Set the criteria you require.
3. When you have entered the criterion for a field, click **Add**. The criterion that you entered becomes visible in the Criteria pane, and the Add Criteria dialog box is cleared.
4. When you have set all the criteria, press the **Close** button on the **Add Criteria** dialog box.

Adding Criteria (method 2)

1. Place the cursor in the Criteria field. A drop-down button appears. Click on this button to display a list of fields.
2. Click on the field for which you want to create a selection criterion. The field name appears in the Criteria Field. Then type a value in the Value field.

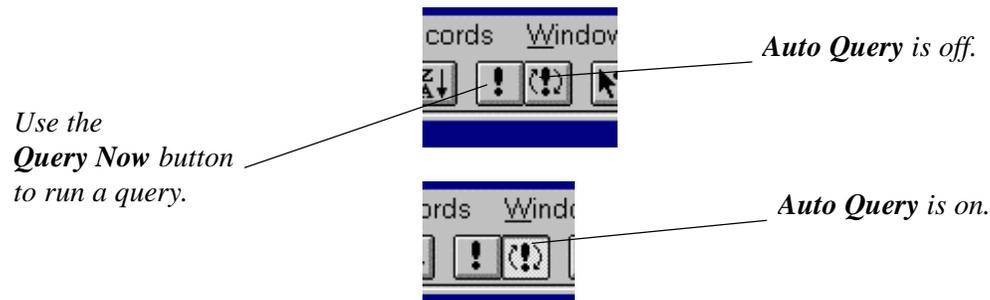
Adding Criteria (method 3)

1. Highlight a data value that you want to set as a criterion and then click the Criteria Equals button (). There must be data in the data pane for you to do this.

Turn off Auto Query

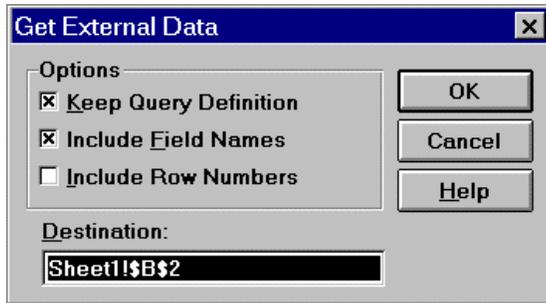
Turn this off so that a query runs when you tell it to, not after every change to criteria. Running a query automatically after each change can seriously impede your work rate!

To run a query, use the **Query Now** button.



Return Data to Excel

1. To return data to Excel from Microsoft Query, select **Return Data to Microsoft Excel** from the **File** menu. The Get External Data dialog box appears.



2. The Keep Query Definition option and the Include Field Names option should be selected (i.e. cross in box).
3. The Destination value is the first cell of the first row in the spreadsheet where the data will be returned.
4. Click **OK**. The data is returned to Excel.

Additional details can be found in Module 7.

1. Master File Maintenance

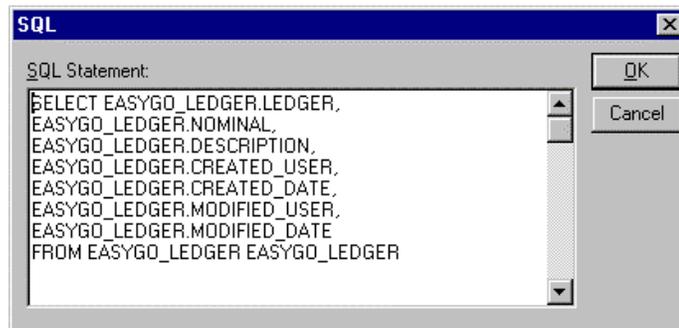
Task 1.1 : Produce a list of all Ledgers

Step One	Add the LEDGER table to the query
Step Two	To select all the columns in the table, highlight the asterisk (*) and drag it into the Data Pane (or double-click on the column)

Result Set

	LEDGE	NOMINAL	DESCRIPTION	CREATED_USER	CREATED_DA	MODIFIED_US
▶	E0001		VAT Ledger	CAROLYN	1997-03-04	TRAINER
	E0002		Fixtures & Fittings	CAROLYN	1997-03-04	TRAINER
	E0003		Holidays	CAROLYN	1997-03-04	TRAINER
	E0004		Brochures	CAROLYN	1997-03-04	TRAINER
	E0005		Destination	CAROLYN	1997-03-04	TRAINER
	P0001	5100	Purchase Ledger	CAROLYN	1997-03-04	TRAINER
	S0001	2100	Sales Ledger	CAROLYN	1997-03-04	TRAINER

SQL



Notice how Excel has listed all of the columns.

The SQL could also be written as follows:

SELECT * FROM EASYGO_LEDGER

Task 1.2 : List Ledgers and their Control Account

Note:

The result should contain the columns LEDGER, NOMINAL, DESCRIPTION from the LEDGER table

LEDGER in the LEDGER table corresponds to Ledger identifier in CODA.

NOMINAL in the LEDGER table corresponds to Control Account in CODA.

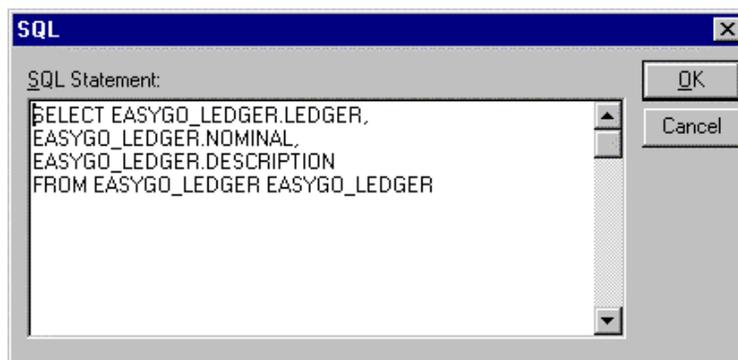
DESCRIPTION in the LEDGER table corresponds to Ledger name in CODA.

Step One	Add the LEDGER table to the query
Step Two	Turn off Auto Query
Step Three	Add the LEDGER column into the query
Step Four	Add the NOMINAL column into the query
Step Five	Add the DESCRIPTION column into the query
Step Six	Click Run Query

Result Set

	LEDGER	NOMINAL	DESCRIPTION	
	E0001		VAT Ledger	
	E0002		Fixtures & Fittings	
	E0003		Holidays	
	E0004		Brochures	
	E0005		Destination	
	P0001	5100	Purchase Ledger	
▶	S0001	2100	Sales Ledger	

SQL



Task 1.3 : Produce a directory of all Nominals

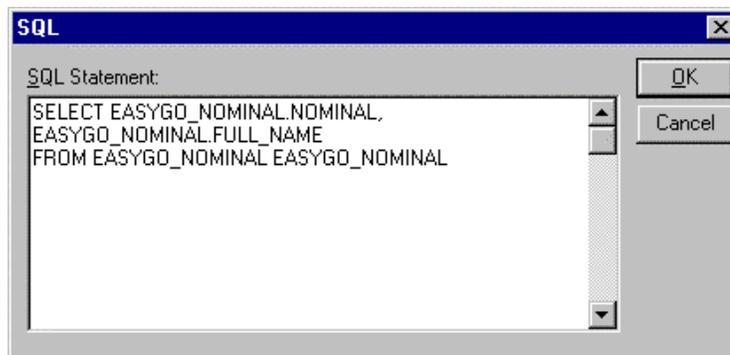
Note: A directory includes NOMINAL and FULL_NAME.

Step One	Add the NOMINAL table to the query
Step Two	Turn off Auto Query
Step Three	Add the NOMINAL column into the query
Step Four	Add the FULL_NAME column into the query
Step Five	Click Run Query

Result Set

NOMINAL	FULL_NAME
1100	Fixtures & Fittings
1110	Fixtures & Fitting Costs
1120	Fixtures & Fitting Depreciation
1200	Buildings
1210	Buildings Costs
1220	Buildings Depreciation
1300	Computer Equipment
1310	Computer Equipment Costs
1320	Computer Equipment Depreciation
1400	Motor Vehicles
1410	Motor Vehicles Costs

SQL



Task 1.4 : Produce a directory of the Sales Nominal (2100)

Step One	Add the NOMINAL table to the query
Step Two	Turn off Auto Query
Step Three	Add the NOMINAL column into the query
Step Four	Add the FULL_NAME column into the query
Step Five	Use the Add Criteria dialog box to select data where NOMINAL equals 2100 (see below)



Step Six	Click Run Query
----------	------------------------

Result Set

	NOMINAL	FULL_NAME
▶	2100	Sales



Task 1.5 : Produce a directory of the Sales Analysis Nominals

Note: All the Sales Nominals begin with the number 6.

Step One	Add the NOMINAL table to the query
Step Two	Turn off Auto Query
Step Three	Add the NOMINAL column into the query
Step Four	Add the FULL_NAME column into the query
Step Five	Add a criterion to select only data where NOMINAL begins with 6. Use LIKE with a wildcard.

Criteria Field: NOMINAL
Value: Like '6%'
or:

Step Six	Click Run Query
----------	------------------------

Result Set

	NOMINAL	FULL_NAME
▶	6000	Sales Analysis
	6100	Sales Analysis - Holidays
	6200	Sales Analysis - Insurance
	6300	Sales Analysis - Travellers Cheques
	6400	Sales Analysis - Foreign Currency

SQL

SQL Statement:

```
SELECT EASYGO_NOMINAL.NOMINAL,
EASYGO_NOMINAL.FULL_NAME
FROM EASYGO_NOMINAL EASYGO_NOMINAL
WHERE (EASYGO_NOMINAL.NOMINAL Like '6%')
```

OK Cancel

Task 1.6 : List SUBACCOUNTS with a destination of Blackpool or Bermuda (Hamilton)

Note:

Blackpool = D0004

Bermuda = D0105

In the listing show SUBACCOUNT, TITLE and LEDGER.

Step One	Add the SUBACCOUNT table to the query
Step Two	Turn off Auto Query
Step Three	Add the SUBACCOUNT column into the query
Step Four	Add the TITLE column into the query
Step Five	Add the LEDGER column into the query
Step Six	Add a criterion to select only data where SUBACCOUNT equals D0105

The screenshot shows a dialog box titled "Add Criteria". At the top, there are two radio buttons: "And" (unselected) and "Or" (selected). To the right of these are "Add" and "Close" buttons. Below this, there are four rows of input fields:

- "Total:" followed by a dropdown menu.
- "Field:" followed by a dropdown menu containing the text "SUBACCOUNT".
- "Operator:" followed by a dropdown menu containing the text "equals".
- "Value:" followed by a text input field containing "D0105".

 To the right of the "Value:" field is a "Values..." button.

Step Seven	Add a criterion to select only data where SUBACCOUNT equals D0004
------------	---

Step Eight	Run the query
------------	---------------

Result Set

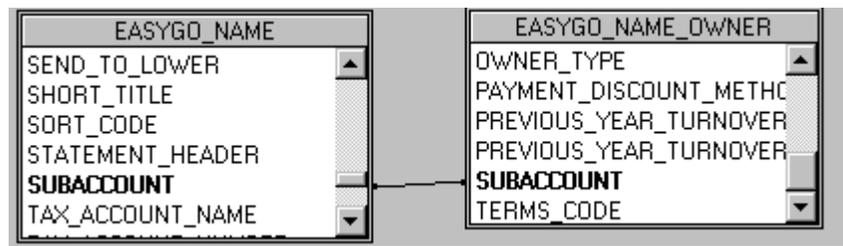
	SUBACCOUNT	TITLE	LEDGER
▶	D0105	Hamilton - Bermuda	0000
	D0004	Blackpool - United Kingdom	0000

SQL

Notice how brackets have been inserted around the two criteria. The bracketing shows how the operators are grouped, and makes for easier reading of the SQL. In this case, even without brackets, there is no possibility of ambiguity, but where there are many levels of AND and OR, it becomes difficult to understand the logic unless brackets are used.

Task 1.7 : List Name and address Subaccounts and the LEDGER they belong to

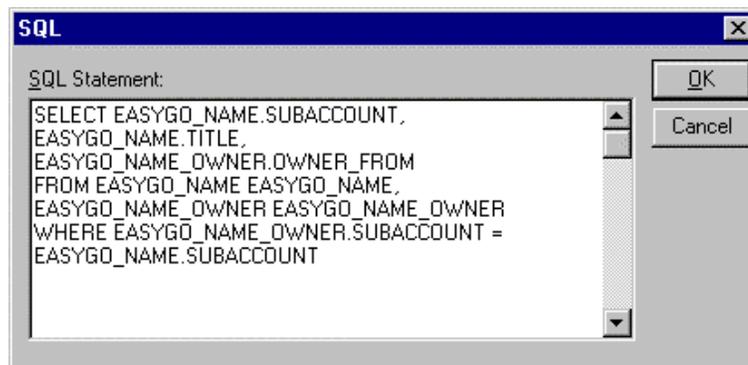
Step One	Add the NAME table to the query
Step Two	Turn off Auto Query
Step Three	Add the SUBACCOUNT column into the query
Step Four	Add the TITLE column into the query
Step Five	Add the NAME_OWNER table into the query
Step Six	Ensure that the tables join on the correct fields (SUBACCOUNT in this case)



Step Seven	Double click on OWNER_FROM to add the column to the query. This column specifies which Ledger the Subaccount belongs to
Step Eight	Run the query

Result Set

	SUBACCOUNT	TITLE	OWNER_FROM
▶	H0001	Thomson Holidays Ltd	P0001
	H0002	Airtours Holidays Ltd	P0001
	H0003	Cosmosair Plc	P0001
	H0004	Kuoni Travel	P0001
	H0005	Cresta Holidays	P0001
	H0006	Magic of Italy	P0001
	H0007	Best Western	P0001
	H0008	Trek America	P0001
	H0009	North American Travel Service	P0001
	H0010	American Airlines Holidays Ltd	P0001
	H0011	Bales Tours Ltd	P0001
	H0012	Virgin Holidays Ltd	P0001
	H0013	P&O	P0001

SQL

```
SQL Statement:
SELECT EASYGO_NAME.SUBACCOUNT,
EASYGO_NAME.TITLE,
EASYGO_NAME_OWNER.OWNER_FROM
FROM EASYGO_NAME EASYGO_NAME,
EASYGO_NAME_OWNER EASYGO_NAME_OWNER
WHERE EASYGO_NAME_OWNER.SUBACCOUNT =
EASYGO_NAME.SUBACCOUNT
```

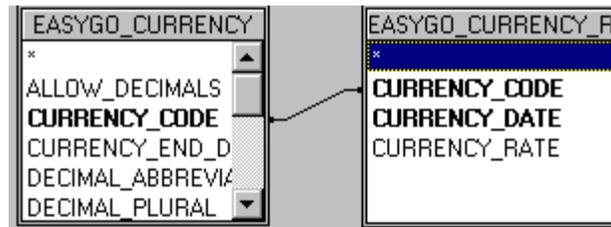
Task 1.8 : List existing currencies and currency rates

Note:

Details of currencies are in the CURRENCY table

Currency rates are in the CURRENCY_RATE table

Step One	Add the CURRENCY table to the query
Step Two	Turn off Auto Query
Step Three	Add the CURRENCY_CODE column into the query
Step Four	Add the TITLE column into the query
Step Five	Add the CURRENCY_RATE table to the query

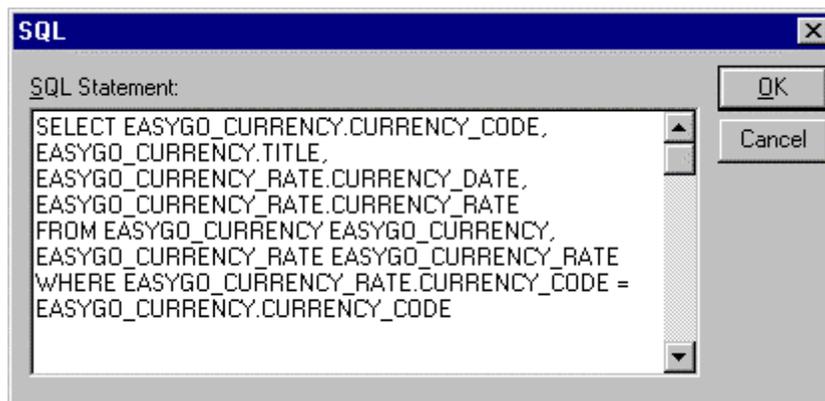


Step Six	Add the CURRENCY_DATE column into the query
Step Seven	Add the CURRENCY_RATE column into the query
Step Eight	Run the query

Result Set

	CURRENCY_CODE	TITLE	CURRENCY_D.	CURRENCY_RATE
▶	AED	United Arab Emirate Dirh	1997-01-01	3.2
	ALL	Albanian Lek	1997-03-14	207.12
	ALL	Albanian Lek	1997-01-01	169.31
	ATS	Austrian Schilling	1997-01-01	18.55
	AUD	Australian Dollar	1997-01-01	2.15
	BDT	Bangladeshi Taka	1997-01-01	72.5
	BEF	Belgian Franc	1997-01-01	54.38
	BGL	Bulgarian Lev	1997-01-01	807.82
	BHD	Bahraini Dinarling	1997-01-01	0.64
	BIF	Burundi Franc	1997-01-01	520.09
	BRL	Brazilian Real	1997-03-14	1.69
	BRL	Brazilian Real	1997-01-01	1.78

SQL



```

SQL Statement:
SELECT EASYGO_CURRENCY.CURRENCY_CODE,
EASYGO_CURRENCY.TITLE,
EASYGO_CURRENCY_RATE.CURRENCY_DATE,
EASYGO_CURRENCY_RATE.CURRENCY_RATE
FROM EASYGO_CURRENCY EASYGO_CURRENCY,
EASYGO_CURRENCY_RATE EASYGO_CURRENCY_RATE
WHERE EASYGO_CURRENCY_RATE.CURRENCY_CODE =
EASYGO_CURRENCY.CURRENCY_CODE
  
```

SUGGESTED EXERCISE

Translate currency rates from GBP to a currency of your choice, and update this information in CODA.

The remainder of this section gives step-by-step instructions for getting currency rates from the internet and for uploading the data to CODA.

1. Start a web browser of your choice. Here we use Microsoft Internet Explorer. To start this, double-click the icon or select the option from the **Start** menu.



2. Go to <http://www.oanda.com/cgi-bin/ncc>, the Olsen and Associates Currency Converter page. (A shortcut has been set up for you under **Favourites**).



3. Select the currencies to use and the date (the default date is today) and click the “Convert Now” button. (Full instructions taken from the Olsen web page are shown in the next step).



4. Here are the online instructions:

Follow these steps to convert currencies.

1. Click the "Currency Converters" folder in the SiteSeeing frame, then choose the language you want or click 164 Currencies Converter in the OANDA homepage. See [To find the 164 Currencies Converter in the SiteSeeing frame](#).
2. Choose a currency or precious metal you want to convert **from** in the left scrolling list. Click the scrolling arrows to see more currency selections.
3. Choose a currency or precious metal you want to convert **to** in the right scrolling list. Click the scrolling arrows to see more currency selections.
4. If you want to change the amount of the currency to convert **from**, click the text box next to "Convert Amount" and type the amount of the currency you want to convert. The default is "1."
5. Click the "Convert Now!" button.

Important: All currencies are listed with the name of their country first. For example, the "Peso" for Mexico is listed as "Mexican Peso."

5. An example currency conversion is shown here:

On Wednesday, July 9, 1997

1 British Pound = 0.9623 Latvian Lats
(1 Latvian Lats = 1.0392 British Pound)

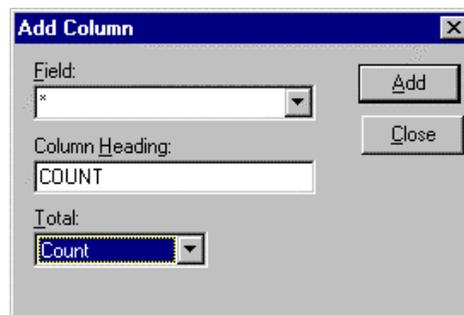
Median price was 0.9623 / 0.9638 (bid/ask).
Estimated price based on daily US dollar rates.

6. Now you have the new rate, you can update the CODA data. Since you cannot update to more than one table using Microsoft Query, remove the CURRENCY table from the download query you had been working on and refresh the data (or start a new query).
7. Switch on the option in Microsoft Query which allows editing (**Records, Allow Editing**).
8. Enter the currency code, date and rate for the currency you want to update.
9. Move the cursor off the row; the new data will be uploaded to CODA.

Task 1.9 : Find out how many customers there are in the database

Note: This can be achieved by counting the rows of the NAME table.

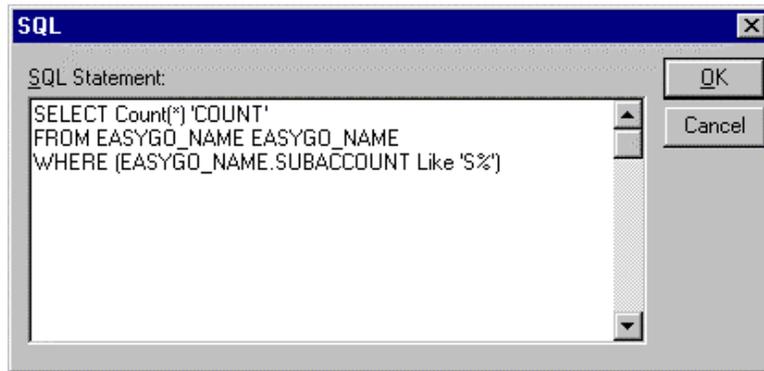
Step One	Add the NAME table to the query
Step Two	Add a criterion to select Subaccounts beginning with S
Step Three	Select Records from the tool bar and chose Add Column . Fill in the dialog box with the necessary information



Step Four	Click on Add . Then run the query.
-----------	---

Result Set

	COUNT
	16

SQL

Task 1.10 : Produce a list of CODA Account Groups

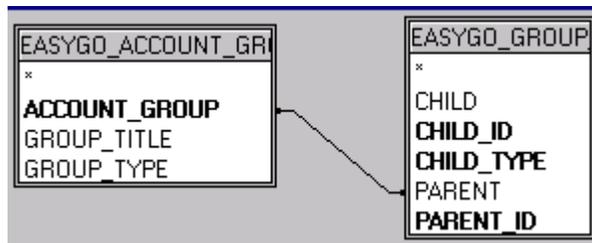
Step One	Add the ACCOUNT_GROUP table to the query
Step Two	Turn off Auto Query
Step Three	Add the ACCOUNT_GROUP column into the query
Step Four	Add the GROUP_TITLE column into the query
Step Five	Run the query

Result Set

	ACCOUNT_GROUP	GROUP_TITLE
▶	ALL	All Season Euro Destinations
	EUHOLS	European Holidays
	SUMMER	Summer European Destinations
	WALL	All Season World Wide Dest's
	WINTER	Winter Holidays
	WSUMMER	Summer World Wide Destinations
	WWHOLS	World Wide Holiday Destination
	WWINTER	Winter World Wide Destinations

Task 1.11 : Produce a list of the Group Hierarchy to one level

Step One	Add the ACCOUNT_GROUP table to the query
Step Two	Turn off Auto Query
Step Three	Add the ACCOUNT_GROUP and GROUP_TITLE columns into the query
Step Four	Add the GROUP_HIERARCHY table to the query
Step Five	Add a join between the tables, by dragging the mouse from ACCOUNT_GROUP to GROUP_HIERARCHY linking together the column ACCOUNT_GROUP to the column PARENT (see below)

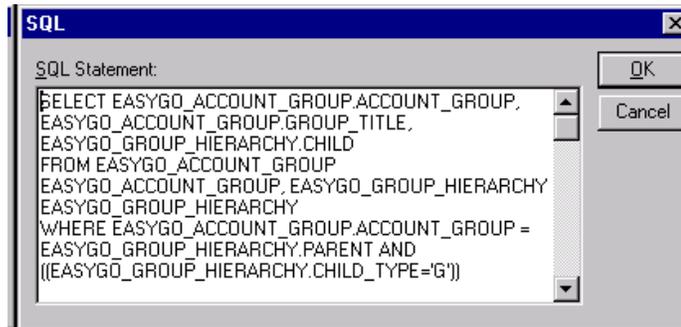


Step Six	Add criteria to select only the CHILD_TYPE equal to G (Group within CODA)
Step Seven	Add the CHILD column into the query
Step Eight	Run the Query to display the hierarchy to one level

Result Set

	ACCOUNT_GROUP	GROUP_TITLE	CHILD
▶	EUHOLS	European Holidays	ALL
	EUHOLS	European Holidays	SUMMER
	EUHOLS	European Holidays	WINTER
	WWHOLS	World Wide Holiday Destination	WALL
	WWHOLS	World Wide Holiday Destination	WSUMMER
	WWHOLS	World Wide Holiday Destination	WWINTER

SQL



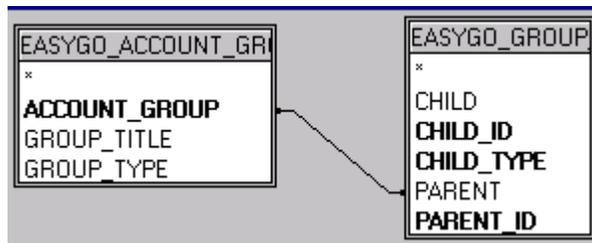
The image shows a screenshot of an SQL dialog box. The dialog box has a title bar with the text "SQL" and a close button (X). Below the title bar, there is a label "SQL Statement:" followed by a text area containing the following SQL query:

```
SELECT EASYGO_ACCOUNT_GROUP.ACCOUNT_GROUP,  
EASYGO_ACCOUNT_GROUP.GROUP_TITLE,  
EASYGO_GROUP_HIERARCHY.CHILD  
FROM EASYGO_ACCOUNT_GROUP  
EASYGO_ACCOUNT_GROUP, EASYGO_GROUP_HIERARCHY  
EASYGO_GROUP_HIERARCHY  
WHERE EASYGO_ACCOUNT_GROUP.ACCOUNT_GROUP =  
EASYGO_GROUP_HIERARCHY.PARENT AND  
((EASYGO_GROUP_HIERARCHY.CHILD_TYPE='G'))
```

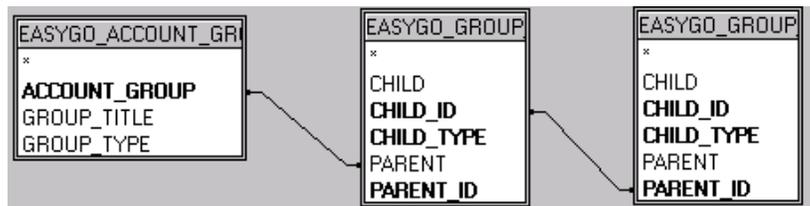
To the right of the text area, there are two buttons: "OK" and "Cancel".

Task 1.12 : Produce a list of the Group Hierarchy to two levels

Step One	Add the ACCOUNT_GROUP table to the query
Step Two	Turn off Auto Query
Step Three	Add the ACCOUNT_GROUP and GROUP_TITLE columns into the query
Step Four	Add the GROUP_HIERARCHY table to the query
Step Five	Add a join between the tables, by dragging the mouse from ACCOUNT_GROUP to GROUP_HIERARCHY linking together the column ACCOUNT_GROUP to the column PARENT



Step Six	Add criteria to select only the CHILD_TYPE equal to G (Group within CODA)
Step Seven	Add the CHILD column into the query
Step Eight	Add the GROUP_HIERARCHY table into the query
Step Nine	Link CHILD_ID to PARENT_ID. Note: ID Numbers are used here for optimisation as these are part of the key.



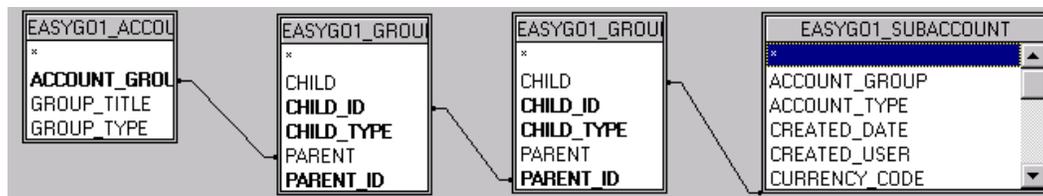
Step Ten	Add CHILD from GROUP_HIIERARCHY_1 into the query
Step Eleven	Click Run Query

Result Set

	ACCOUNT_GROUP	GROUP_TITLE	CHILD	CHILD
	EUHOLS	European Holidays	ALL	D0005
	EUHOLS	European Holidays	ALL	D0017
▶	EUHOLS	European Holidays	ALL	D0018
	EUHOLS	European Holidays	ALL	D0020
	EUHOLS	European Holidays	ALL	D0022
	EUHOLS	European Holidays	ALL	D0024
	EUHOLS	European Holidays	ALL	D0027
	EUHOLS	European Holidays	ALL	D0030
	EUHOLS	European Holidays	ALL	D0052
	EUHOLS	European Holidays	SUMMER	D0004
	EUHOLS	European Holidays	SUMMER	D0006
	EUHOLS	European Holidays	SUMMER	D0013
	EUHOLS	European Holidays	SUMMER	D0026
	EUHOLS	European Holidays	SUMMER	D0051
	EUHOLS	European Holidays	SUMMER	D0054
	EUHOLS	European Holidays	SUMMER	D0055
	EUHOLS	European Holidays	SUMMER	D0059
	EUHOLS	European Holidays	WINTER	D0009
	EUHOLS	European Holidays	WINTER	D0021
	EUHOLS	European Holidays	WINTER	D0042
	EUHOLS	European Holidays	WINTER	D0043
	EUHOLS	European Holidays	WINTER	D0065
	EUHOLS	European Holidays	WINTER	D0070

SUGGESTED EXERCISE

Add the SUBACCOUNT table to get the title of the Child (see example below)



Result Set

	GROUP_TITLE	CHILD	CHILD	TITLE
	European Holidays	WINTER	D0009	Chamonix - French Alps
	European Holidays	WINTER	D0021	Andorra - The Pyrenees
	European Holidays	WINTER	D0042	Geneva - Switzerland
	European Holidays	WINTER	D0043	Granda - Spain
	European Holidays	WINTER	D0065	Salzburg - Austria
	European Holidays	WINTER	D0070	Vienna - Austria
	European Holidays	SUMMER	D0004	Blackpool - United Kingdom
	European Holidays	SUMMER	D0006	Cannes - France
	European Holidays	SUMMER	D0013	Sorrento - Italy
	European Holidays	SUMMER	D0026	Gibraltar
	European Holidays	SUMMER	D0051	Madrid - Spain
	European Holidays	SUMMER	D0054	Milan - Italy
	European Holidays	SUMMER	D0055	Monte Carlo - France
	European Holidays	SUMMER	D0059	Nice - France
	European Holidays	ALL	D0005	Paris - France
	European Holidays	ALL	D0017	Benidorm - Costa Brava
	European Holidays	ALL	D0018	Barcelona - Spain
	European Holidays	ALL	D0020	Torremolinos - Costa Del Sol
	European Holidays	ALL	D0022	Malta - Malta

SUGGESTED EXERCISES

- i.** Add criteria to look at any one ACCOUNT_GROUP and then open the DETAILS table to get the corresponding transaction amounts from the Books.

REMEMBER DESTINATION is a LEVEL3.

- ii.** Add a new group, UK Holiday Destination Group, to the account groups. Start by adding the group to the ACCOUNT_GROUP Table and then add the rest of the hierarchy using GROUP_HIERARCHY.

REMEMBER CODA assigns its own Identification numbers and therefore these should be left blank.

Task 1.13 : Repeat previous tasks using the Easysoft Excel macro for CODA

Note: The macro is described in Module 10.

The macro can be used to simplify the tasks that you have performed. In this exercise, you will repeat some of those tasks.

- Subtask 1. Produce a list of all Ledgers (previously done manually as task 1).
- Subtask 2: List Ledger (identifier and name) and Control Account from the Ledgers Master File (previously done manually as task 2).
- Subtask 3: Produce a directory of the Sales Nominal (2100) (previously done manually as task 4).
- Subtask 4: Produce a directory of the Sales Analysis Nominals (previously done manually as task 5).
- Subtask 5: List SUBACCOUNTS with a destination of Blackpool or Bermuda (previously done manually as task 6).
- Subtask 6 : Download existing currency rates, add a new rate and upload (previously done manually as a further exercise in task 8).

You may have noticed that the queries involving joins are not included here. The reason for this is that the main use of the Masters-related options on the macro is the download and upload of bulk information.

If you want more specific information, then you can use the Report Writer (under the **Reports** menu option) to join tables.

2. Budgets

Task 2.1 : Download budget information, change it and upload

Download budget information for a particular budget year and budget code. Change some of the budget values and upload them into CODA. In this case we will modify all Budgets (BUDGET_CODE=B) and upload them with the changes.

Note: ACCOUNT_BALANCE (used later in the query) is a flag which is internal to the CODA system. It cannot be seen on CODA-IAS. There are two allowed values for this column, namely Y and N. Y indicates that the value is at the lowest level of N, S or L3. N indicates that the value is a total value of all other N/S/L3 accounts under the given account.

Step One	Add the BUDGET table to the query
Step Two	Turn off Auto Query
Step Three	Add Criteria where BUDGET_CODE equals B
Step Four	Add Criteria where BUDGET_YEAR equals 1997
Step Five	Add Criteria where ACCOUNT_BALANCE is equal to N
Step Six	Select NOMINAL, SUBACCOUNT, LEVEL3 and VALUE_0001, VALUE_0002, VALUE_0003, etc.
Step Seven	Run the query

Result Set

	NOMINAL	SUBACCOUNT	LEVEL3	VALUE_0001	VALUE_0002	VAI
	6100	B0006		-900.	-1350.	-2250.
	6100	B0013		-400.	-600.	-1000.
	6100	B0024		-1400.	-2100.	-3500.
	6100	B0033		-20000.	-30000.	-50000.
	6100	B0040		-80.	-120.	-200.
	6100	B0049		-80.	-120.	-200.
	6100	B0053		-160.	-240.	-400.
	6100	B0057		-1400.	-2100.	-3500.
	6100	B0063		-700.	-1050.	-1750.
	6100	B0067		-1000.	-1500.	-2500.
	6100	B0068		-500.	-750.	-1250.
	6100	B0069		-800.	-1200.	-2000.
	6100	B0073		1000.	1500.	2500.

Step Eight	Click on Records from the Toolbar and select Allow Editing . (If it is already selected it a tick will appear next to the selection). This adds a blank row onto the bottom of the table.
Step Nine	Select the row you want to update. You can add a selection criteria for a particular NOMINAL and SUBACCOUNT

Example

	SUBACCOUNT	LEVEL3	VALUE_0001	VALUE_0002	VALUE_0003	VAI
	B0033		-20000.	-30000.	-50000.	0.
▶						

Step Nine	Edit one or more of the VALUE columns (include the sign rule +/-).
-----------	--

Example

	NOMINAL	SUBACCOUNT	LEVEL3	VALUE_0001	VALUE_0002	VAI
	6100	B0033		-2000.	-3000.	-50000.
▶						

Step Ten	Move off the row to upload the Budget data into CODA
----------	--



It is not possible to see the SQL produced from within Query. The SQL for the update only would be:

```
UPDATE EASYGO_BUDGET
SET EASYGO_BUDGET.VALUE_0001 = '-20000',
    EASYGO_BUDGET.VALUE_0002 = '-30000'
WHERE (EASYGO_BUDGET.BUDGET_CODE='B') AND
      (EASYGO_BUDGET.BUDGET_YEAR=1997) AND
      (EASYGO_BUDGET.SUBACCOUNT='B0033') AND
      (EASYGO_BUDGET.ACCOUNT_BALANCE='N')
```

Task 2.2 : Use the Macro to download and upload budgets

In this exercise, you will download Actuals and upload these as Budgets.

Step One	Download Actual information for Sales Analysis of holiday company Thomson City Breaks (Coda, Budgets, Download)
----------	--

This is equivalent to having no criteria.

Use a lookup to find which Nominal code is used for Sales Analysis (Holidays).

Use a lookup to find which Subaccount code is used for Thomson City Breaks.

Step Two	Upload these Actuals as Budgets (Coda, Budgets, Upload)
----------	--

Step Three	Download budget information for one period and upload it to a different period
------------	--

3. Reporting

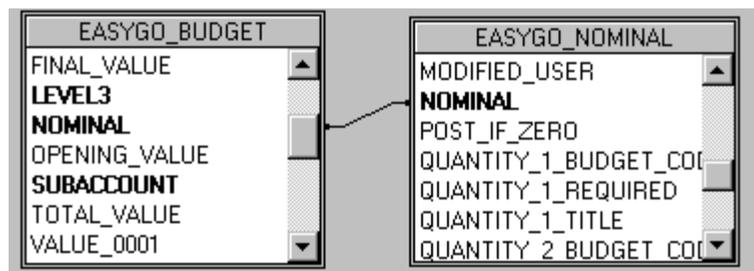
Task 3.1 : Produce a Trial Balance

Note:

The report should show NOMINAL, FULL_NAME and TOTAL_VALUE based on the Actuals (Budget code A).

Try to complete the task without using the steps.

Step One	Add the BUDGET table to the query
Step Two	Turn off Auto Query
Step Three	Add Criteria where BUDGET_CODE is equal to A
Step Four	Add Criteria where BUDGET_YEAR is 1997
Step Five	Add Criteria where SUBACCOUNT IS NULL
Step Six	Add Criteria where LEVEL3 IS NULL
Step Seven	Add the columns NOMINAL and TOTAL_VALUE to the query
Step Eight	Add the NOMINAL table to the query and check the joins are correct



Step Nine	Add the column FULL_NAME to the query
Step Ten	Run the query

	NOMINAL	FULL_NAME	TOTAL_VALUE
	1110	Fixtures & Fitting Costs	2000.
▶	1120	Fixtures & Fitting Depreciation	-200.
	1310	Computer Equipment Costs	3500.
	1320	Computer Equipment Depreciation	-600.
	1410	Motor Vehicles Costs	9980.
	1420	Motor Vehicles Depreciation	-2000.
	1510	Office Equipment Costs	1700.
	1520	Office Equipment Depreciation	-600.
	2100	Sales	31891.
	2210	Bank - Current Account	32040.
	3100	Capital	-50000.

Step Eleven	Click on File then Return Data to Microsoft Excel
Step Twelve	Format the text and numbers (highlight the column, right click and select Format Cells – you will find a few interesting options to try)

Example

	TRIAL BALANCE	
1110	Fixtures & Fitting Costs	2000.00
1120	Fixtures & Fitting Depreciation	200.00
1310	Computer Equipment Costs	3500.00
1320	Computer Equipment Depreciation	600.00
1410	Motor Vehicles Costs	9980.00
1420	Motor Vehicles Depreciation	2000.00
1510	Office Equipment Costs	1700.00
1520	Office Equipment Depreciation	600.00
2100	Sales	31891.00
2210	Bank - Current Account	32040.00
3100	Capital	50000.00
5100	Purchases	30052.12
5200	Payroll	1500.00
5300	VAT Control Account	167.12
6100	Sales Analysis - Holidays	31891.00

Step Thirteen	Add a cell on your worksheet to sum the TOTAL_VALUES displayed. Check the balance is zero
---------------	---

Example

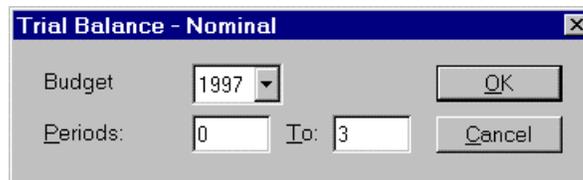
8200	Salaries	5680.00
8300	Motor Vehicles	135.00
8610	Accountant	820.00
	TOTAL	0.00

Task 3.2 : Produce a Trial Balance using the macro

Note:

The report should be based on the Actuals (budget code A). Try to complete the task without using the steps.

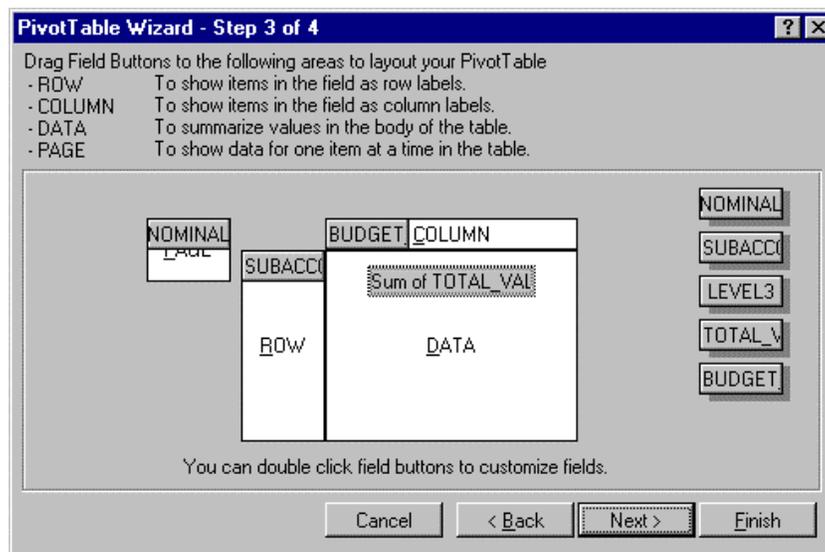
Step One	Download the BUDGET data (Coda, Fixed Reports, Trial Balance)
----------	---



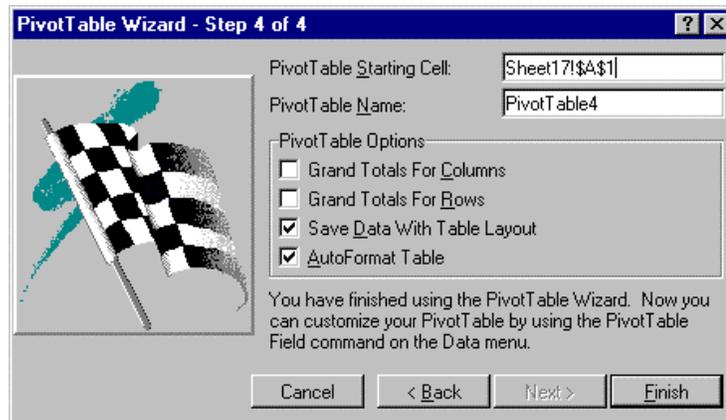
Step Two	Also look at different periods
Step Three	When data has been returned to the worksheet, double-click on an amount to see the transactions for the corresponding account. (To return to the report, re-select the worksheet on which the report was generated)
Step Four	Double-click on a Nominal or Description to see a breakdown of the Nominal
Step Five	From the breakdown of the Nominal, double-click on Amount to see the transactions

Task 3.3 : Produce an Actual Budget Variance report using pivot tables

Step One	Add the BUDGET table to the query
Step Two	Turn off Auto Query
Step Three	Add Criteria where BUDGET_CODE=A OR BUDGET_CODE=B
Step Four	Add Criteria where BUDGET_YEAR=1997
Step Five	Add Criteria where SUBACCOUNT IS NOT NULL
Step Six	Select the columns NOMINAL, SUBACCOUNT, LEVEL3, TOTAL_VALUE and BUDGET_CODE
Step Seven	Run the query
Step Eight	Return the data to Excel
Step Nine	Highlight the data on the spreadsheet. Select Pivot Table from the Data menu. Choose Microsoft Excel List or Database and click Next



Step Ten	Drag NOMINAL onto PAGE , SUBACCOUNT onto ROW , BUDGET_CODE onto COLUMN , TOTAL_VALUE onto DATA then select Next
----------	--



Step Eleven	Uncheck Grand Totals for Columns and Grand Totals for Rows
Step Twelve	Click on Pivot Table Starting Cell and then on an unused cell on the worksheet to select a starting cell
Step Thirteen	Click on Finish to see the Pivot Table

Result Set

NOMINAL	6100	
Sum of TOTAL_VALUE	SUBACCOUNT	BUDGET_CODE
	A	B
	B0005	-2563 -500
	B0006	-9570 -4500
	B0013	-2684 -2000
	B0024	-16940 -7000
	B0033	-3388 -100000
	B0040	-2244 -400
	B0049	-1661 -400
	B0053	-1322 -800
	B0057	-6710 -7000

SUGGESTED EXERCISES		
i. Flick through the Nominals to see Subaccounts for each one	ii. Work out the variance on a Nominal that has both Actual and Budget data	iii. Double-click on the Subaccount to see Level3 information if that exists

Result Set for exercise ii.

NOMINAL	6100			
Sum of TOTAL_VALUE	BUDGET CODE			
SUBACCOUNT	A	B	VARIANCE	
B0005	-2563	-500	-2063	
B0006	-9570	-4500	-5070	
B0013	-2684	-2000	-684	
B0024	-16940	-7000	-9940	
B0033	-3388	-100000	96612	
B0040	-2244	-400	-1844	
B0049	-1661	-400	-1261	
B0053	-1322	-800	-522	
B0057	-6710	-7000	290	
B0063	2000	2500	4700	

Task 3.4 : Produce an Actual Budget Variance report using the macro

Step One	Select Coda, Fixed Reports, Actual Budget Variance
Step Two	Set the criteria as shown in the screen shot

Budget Variance Selection

Nominal: Like 6% OK

Subaccount: <> Cancel

Level3: Like % Defaults

Budget: 1997

Periods: 0 To: 3

Main Code: A

Compariso: B

Wildcard Characters are percent % meaning a group of characters and underscore _ meaning a single character

Step Three	Double-click on Amount to determine the transactions that make up the Actual
------------	--

Task 3.5 : Produce a Profit and Loss Account

Notes:

All the Sales nominals begin with 6.

All the Purchase nominals begin with 7.

All the Expense nominals begin with 8.

Gross Profit = Sales – Cost of Goods sold (purchases – closing stock)

Net Profit = Gross Profit - Expenses

This report is easily broken down into separate queries and then exported back onto an excel worksheet to be formatted.

Try to follow the initial steps to produce the first query and then continue without help.

Step One	Add the BUDGET table to the query
Step Two	Turn off Auto Query
Step Three	Add Criteria where BUDGET_CODE is equal to A
Step Four	Add Criteria where BUDGET_YEAR=1997
Step Five	Add Criteria where SUBACCOUNT IS NULL
Step Six	Add Criteria where NOMINAL begins with 6 (Sales Analysis)
Step Seven	Select the column TOTAL_VALUE, highlight it and sum it



Sum of TOTAL_VAL
-31891.

Step Eight	Run the query and then save it – call it Sales
Step Nine	Return the data to Excel
Step Ten	Move off the result set, select Data, Get External Data to return to Query

Step Eleven	Press Close on the Tables dialog box and open your Sales query
Step Twelve	Modify the query to select Purchases (NOMINAL begins with 7). Run the query and save it as Purchases
Step Thirteen	Repeat Steps Nine, Ten and Eleven
Step Fourteen	Modify the query to select Expenses (NOMINAL begins with 8). Run the query and save it as Expenses
Step Fifteen	Return the data to Excel
Step Sixteen	Format the worksheet headings
Step Seventeen	Perform the calculations (-sales+purchases+expenses)

A	B	C	D
SALES	PURCHASES	EXPENSES	NET PROFIT
-31891	28930	6635	3674

SUGGESTED EXERCISES	
i. Add a Column for Gross Profit	ii. Format the numbering on the columns

Example for exercise i.

SALES	PURCHASES	GROSS PROFIT	EXPENSES	PROFIT OR LOSS
-31891	28930	-2961	6635	3674

Task 3.6 : Use the macro to produce a Profit and Loss Account

Step One	Select Coda, Fixed Reports, Profit & Loss
Step Two	Set the criteria

The screenshot shows a dialog box titled "Profit & Loss Selection". It has the following fields and options:

- Financial Year:** A dropdown menu set to "1997".
- Period Range:** Two input boxes, the first containing "0" and the second containing "3".
- Categories:** A list box containing "Sales", "Purchases", and "Expenses". "Sales" is currently selected.
- Selection:** A table with three rows. The first row has "NOMINAL" in a dropdown, "Like" in a dropdown, and "6%" in a text box. To the right of each row are two radio buttons labeled "And" and "Or".
- Buttons:** "OK", "Cancel", and "Reset" are located on the right side of the dialog.

*Similarly,
Purchases: Nominal Like 7%
Expenses: Nominal Like 8%*

Step Three	Click OK to return the data to Excel
------------	---

Task 3.7 : Produce a Balance Sheet for the year so far

Notes:

This exercise should be done using Microsoft Query.

Fixed Assets start with 1.

Current Assets start with 2.

Capital starts with 3, or is greater than or equal to 6 (explanation later).

Long Term Liabilities start with 4.

Current Liabilities start with 5.

Assets = Capital- Liabilities.

Use Headings for NOMINAL (FULL_NAME).

As in the previous task, this report is broken down into separate queries and then exported back onto an Excel worksheet to be formatted.

Try to follow the initial steps to produce the first query and then continue without help.

Step One	Add the BUDGET table to the query
Step Two	Turn off Auto Query
Step Three	Add Criteria where BUDGET_CODE is equal to A
Step Four	Add Criteria where BUDGET_YEAR=1997
Step Five	Add Criteria where SUBACCOUNT IS NULL
Step Six	Add Criteria where NOMINAL begins with 1 (Fixed Assets)
Step Seven	Add the NOMINAL table and select FULL_NAME
Step Eight	Select the column TOTAL_VALUE, highlight it and sum it



Sum of TOTAL_VALUE	
13780.	

Step Nine	Run the query and save the Query – call it Fixed Assets
-----------	---

Step Ten	Return the Result Set to Excel, total up the results on the worksheet
Step Eleven	Move off the result set, select Data, Get External Data to return to Query
Step Twelve	Press Close on the Tables dialog box and open your Fixed Assets query
Step Thirteen	Modify the query to select Current Assets (NOMINAL begins with 2). Run and save as Current Assets
Step Fourteen	Return the Result Set to Excel. Total up the Current Assets on the worksheet
Step Fifteen	On the worksheet, add the Sum of Fixed Assets and the Sum of Current Assets together to get the Value of Total Assets

FIXED ASSETS		
NOMINAL		
1110	Fixtures & Fitting Costs	2000
1120	Fixtures & Fitting Depreciation	-200
1310	Computer Equipment Costs	3500
1320	Computer Equipment Depreciation	-600
1410	Motor Vehicles Costs	9980
1420	Motor Vehicles Depreciation	-2000
1510	Office Equipment Costs	1700
1520	Office Equipment Depreciation	-600
	TOTAL FIXED ASSETS	13780
CURRENT ASSETS		
NOMINAL		
2100	Sales	31891
2210	Bank - Current Account	32040
	TOTAL CURRENT ASSETS	63931
	TOTAL ASSETS	77711

Step Sixteen	Repeat Steps Ten and Eleven
Step Seventeen	Modify to select Capital (NOMINAL begins with 3, or is equal to or greater than 6). Run and save as Capital
	The second criteria is needed here because a year-end has not been performed and therefore retained earnings have not been calculated. For this reason, the trading aspects are used
Step Eighteen	Repeat steps Nine, Ten and Eleven

Step Nineteen	Modify to select Current Liabilities (NOMINAL begins with 5). Run and save as Current Liabilities
Step Twenty	Repeat steps Nine, Ten and Eleven
Step Twenty-one	Modify to select Long Term Liabilities (NOMINAL begins with 4). Run and save as Long Term Liabilities
Step Twenty-two	Return the result set to Excel
Step Twenty-three	On the worksheet, add the Sum of Current Liabilities, Long Term Liabilities and the Sum of Capital together to get the Value of Total Liabilities

CAPITAL		
3100	Capital	-50000
6100	Sales Analysis - Holidays	-31891
7100	Purchase Analysis - Holidays	28930
8200	Salaries	5680
8300	Motor Vehicles	135
8610	Accountant	820
	TOTAL CAPITAL	-46326
CURRENT LIABILITIES		
5100	Purchases	-30052.12
5200	Payroll	-1500
5300	VAT Control Account	167.12
	TOTAL CURRENT LIABILITIES	-31385
LONG TERM LIABILITIES		
	TOTAL LONG TERM LIABILITIES	0
	TOTAL LIABILITIES	-77711

SUGGESTED EXERCISES

i. Format the worksheet headings

ii. Format the numbering on the columns

Task 3.8 : Use the macro to produce a Balance Sheet

Step One	Select Coda, Fixed Reports, Balance Sheet
Step Two	Set the criteria as shown in the screen shot

Balance Sheet Selection Criteria

Financial Year: 1997

Period Range: 0 To: 3

Categories:

- Fixed Assets
- Current Assets
- Capital**
- Long Term Liabilities
- Current Liabilities

Selection

NOMINAL	Like	3%	<input type="radio"/> And	<input checked="" type="radio"/> Or
NOMINAL	>=	6	<input type="radio"/> And	<input type="radio"/> Or
			<input type="radio"/> And	<input type="radio"/> Or

Budget Year = 1997

for Fixed Assets: Nominal Like 1%

for Current Assets: Nominal Like 2%

for Capital: Nominal Like 3% or Nominal >= 6 (shown above)

for Long Term Liabilities: Nominal Like 4%

for Current Liabilities: Nominal Like 5%

Step Three	Return the data to Excel
------------	--------------------------

Task 3.9 : Produce an Ageing Summary

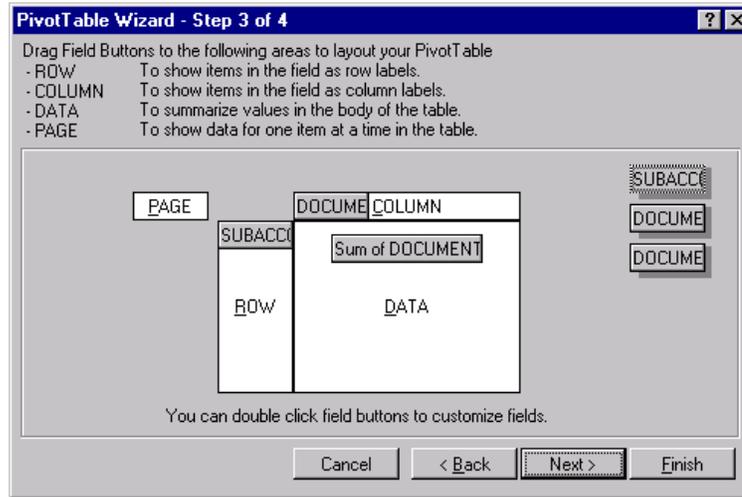
Notes:

An Ageing Summary is list of outstanding debts in relation to time.

Use a Pivot Table to perform this task.

The summary should be for the first three periods in 1997.

Step One	Add the DETAILS table to the query
Step Two	Turn off Auto Query
Step Three	Add Criteria where LINE_NUMBER is equal to 1
Step Four	Add Criteria where NOMINAL is not null
Step Five	Add Criteria to select the customers (SUBACCOUNT begins with S)
Step Six	Add Criteria to select all non-paid documents (i.e. DETAIL_STATUS not equal to P)
Step Seven	Select SUBACCOUNT and add the column to the query
Step Eight	Select DOCUMENT_PERIOD and add the column to the query
Step Nine	Select DOCUMENT_VALUE and add the column to the query
Step Ten	Run the query
Step Eleven	Export the result set back into Excel
Step Twelve	Highlight the Data on the spreadsheet. Select Data, Pivot Table... Choose Microsoft Excel List or Database and click Next
Step Thirteen	Drag SUBACCOUNT into the Row area, DOCUMENT_PERIOD to the columns area and DOCUMENT_VALUE to the Data area



Step Fourteen	Click Finish
---------------	---------------------

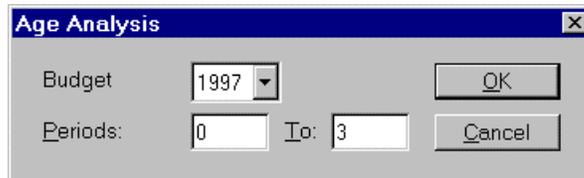
Result Set

Sum of DOCUMENT VALUE	DOCUMENT PERIOD			Grand Total
SUBACCOUNT	1	2	3	
S0001	3151.5	770	0	3921.5
S0002	2860	0	0	2860
S0003	6215	0	0	6215
S0004	1089	440	0	1529
S0005	187	330	0	517
S0006	995.5	0	0	995.5
S0007	331	0	0	331
S0008	2750	770	0	3520
S0009	1540	0	770	2310
S0010	2300	1925	0	4225
S0011	1100	242	0	1342
S0012	990	0	220	1210
S0014	1650	0	0	1650
S0015	0	0	220	220
S0016	0	0	440	440
S0017	0	0	605	605
Grand Total	25159	4477	2255	31891

SUGGESTED EXERCISE
Move the columns around so that you can page on SUBACCOUNT

Task 3.10 : Use the macro to Produce an Ageing Summary

Step One	Select Coda, Fixed Reports, Age Analysis
Step Two	Set the criteria as shown in the screen shot



Step Three	Return the data to Excel
------------	--------------------------

SUGGESTED EXERCISES	
i. Move the columns around so that you can page by SUBACCOUNT	ii. Double-click on Subaccounts to see Level3s

4. Input and Intray

In all of these exercises you will use the Easysoft Excel macro for CODA.

Task 4.1 : Invoice Upload

In this exercise you will upload an invoice to CODA for Mr. A. Cornwell (accountant).

Step One	Select Coda, Input & Intray, Document Setup
Step Two	Remove all the pre-selected columns by clicking Deselect All
Step Three	Select the following columns, then click OK

LINE_NUMBER
DOCUMENT_VALUE
NOMINAL
SUBACCOUNT
LEVEL3
DESCRIPTION

Step Four	Enter the following values in the worksheet
-----------	---

	A	B	C	D	E	F
1	LINE_NUMBER	DOCUMENT_VALUE	NOMINAL	SUBACCOUNT	LEVEL3	DESCRIPTION
2	1	500.00	5100	P0004		PO
3	2	412.50	8610			
4	3	87.50	5300	V0001		VAT
5						

Step Five	Select Coda, Input & Intray, Post Documents
Step Six	Enter the following values in the dialog box

Invoice Upload

Post to
 Books Intray

Header
 Type: PINV Number: <num> Currency: GBP
 Date: <date> Year: 1997 Period: 3

OK Cancel

The values in the dialog box are defaults, and are only used if any

required details are missing from the worksheet.

Number refers to the document number. The value of Number depends upon the sequence rule specified in the document master (SEQUENCING_ACTION in the DOCUMENT_MASTER table). If it is not set to automatic (A), then it must be entered here. If it is set to A (as in these exercises), then the value here is irrelevant - it will be ignored.

Step Seven	Click the OK button
------------	----------------------------

Task 4.2 : Invoice Download

In this exercise you will download the invoice you just uploaded, just to check the information you entered.

Step One	Select Coda, Input & Intray, Details Download
Step Two	Use the default columns which are highlighted, and in addition, select the DOCUMENT_VALUE column
Step Three	Add criteria: DOCUMENT_TYPE = PINV
Step Four	Click OK to download the data

Result Set

	A	B	C	D	E	F	G	H	I	J	K
1	DOCUMENT_TYP	DOCUMENT_L	LINE_NUM	DOCUMENT_DATE	DOCUMENT_VALUE	NOMINAL	SUBACCO	LEV	DOCUMENT_YE	DOCUMENT_PE	INPUT_DATE
2	PINV	3	1	1-Feb-1997	-158.62	5100	P0003		1997	2	6-Mar-1997
3	PINV	3	2	1-Feb-1997	135.00	8300			1997	2	6-Mar-1997
4	PINV	3	3	1-Feb-1997	23.62	5300	V0001		1997	2	6-Mar-1997
5	PINV	4	1	1-Feb-1997	-963.50	5100	P0004		1997	2	6-Mar-1997
6	PINV	4	2	1-Feb-1997	820.00	8610			1997	2	6-Mar-1997
7	PINV	4	3	1-Feb-1997	143.50	5300	V0001		1997	2	6-Mar-1997

Task 4.3 : Upload Salary Journal

Step One	Select Coda, Input & Intray, Document Setup
Step Two	Remove all the pre-selected columns by clicking Deselect All
Step Three	Select the following columns, then click OK

LINE_NUMBER
DOCUMENT_VALUE
NOMINAL
SUBACCOUNT
LEVEL3
DESCRIPTION

Step Four	Enter the following values in the worksheet
-----------	---

	LINE_NUMBER	DOCUMENT_VALUE	NOMINAL	SUBACCOUNT	LEVEL3	DESCRIPTION
1						
2	1	0.00	2210			
3	2	-2,090.00	2210			
4	3	-750.00	5200			
5	4	2,840.00	8200			

Step Five	Select Coda, Input & Intray, Post Documents
Step Six	Enter the following values in the dialog box

Invoice Upload

Post to
 Books Intray

Header
 Type: PJNL Number: <num> Currency: GBP
 Date: <date> Year: 1997 Period: 3

OK Cancel

Step Seven	Click the OK button
------------	----------------------------

Task 4.4 : Download Salary Journal

In this exercise you will download the journal you just uploaded, just to check the information you entered.

Step One	Select Coda, Input & Intray, Details Download
Step Two	Use the default columns which are highlighted, and in addition, select the DOCUMENT_VALUE column
Step Three	Add criteria: DOCUMENT_TYPE = PJNL
Step Four	Click OK to download the data

Result Set

DOCUMENT_TYPE	DOCU LINE	NUM	DOCUMENT_DATE	DOCUMENT_VALUE	NOMINAL	SUBA LEVI	DOCUMENT_YE	DOCUMENT_PER	INPUT_DATE
PJNL	1	1	31-Jan-1997	-2,090.00	2210		1997	1	6-Mar-1997
PJNL	1	2	31-Jan-1997	-750.00	5200		1997	1	6-Mar-1997
PJNL	1	3	31-Jan-1997	2,840.00	8200		1997	1	6-Mar-1997
PJNL	2	1	28-Feb-1997	-2,090.00	2210		1997	2	7-Mar-1997
PJNL	2	2	28-Feb-1997	-750.00	5200		1997	2	7-Mar-1997
PJNL	2	3	28-Feb-1997	2,840.00	8200		1997	2	7-Mar-1997

10. Easysoft Excel Macro for CODA

This module describes the macro add-in for Microsoft Excel.

In this module you will learn

- ✦ what the macro is
- ✦ how to install it
- ✦ how to use it

Contents

Overview of Macro _____	10-2
Macro Installation _____	10-3
Initialisation _____	10-5
Masters-related options _____	10-7
Download _____	10-7
Upload _____	10-9
Reports-related options _____	10-10
Budgets _____	10-10
Fixed Reports _____	10-12
Input and Intraday _____	10-18
Lookups _____	10-20
Reports _____	10-21
Information _____	10-24
About Easysoft ODBC for CODA _____	10-24
Contacting Easysoft _____	10-24

Overview of Macro

The Easysoft Excel Macro for CODA (also known as the “CODA macro”) allows you to create reports and upload data to CODA with the minimum of effort.

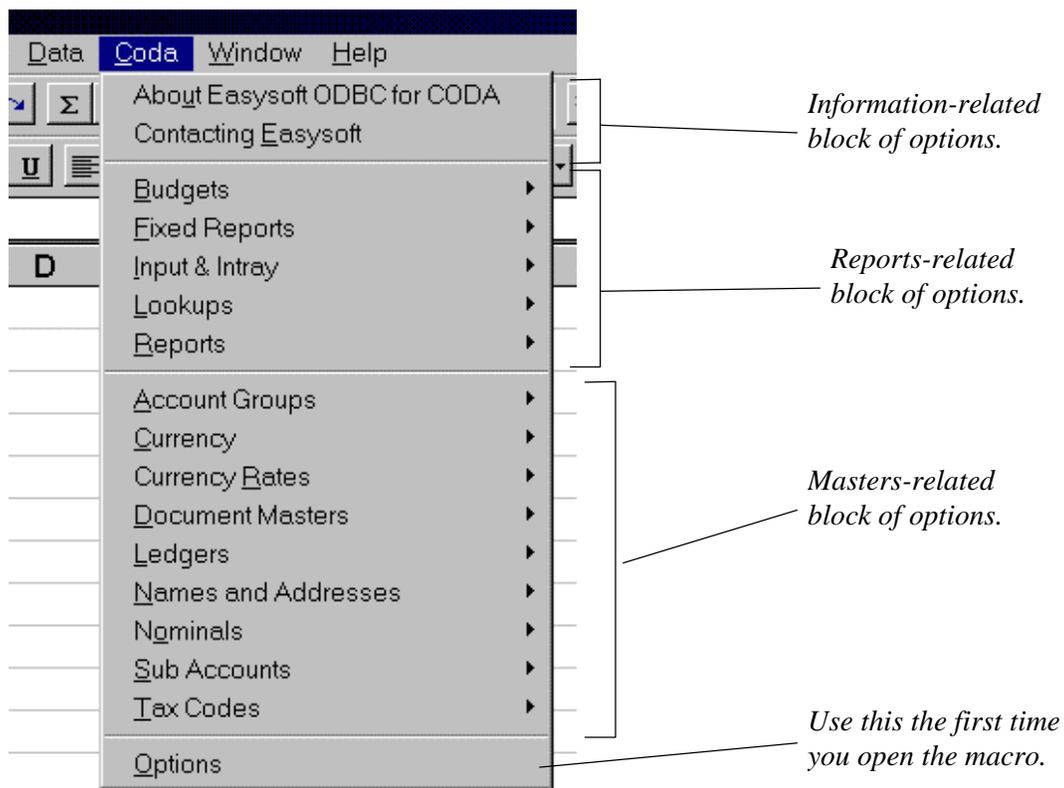
There are two versions of the Easysoft Excel Macro for CODA.

16 bit CODA16.XLA for Microsoft Excel version 5.x only
 32 bit CODA32.XLA for Microsoft Excel 95 and Excel 97

The software applies to PC versions of Excel only.

The Easysoft Excel Macro for CODA requires Easysoft ODBC for CODA version 1.3 build 170 and above on both the client and the server.

The **Coda** menu option on the menu bar gives access to the top level pulldown shown here. There are four groups of options, related to information about the macro, reports, masters and initialisation. They will be discussed later in the following order: initialisation, masters-related options, reports-related options, information.



Before the macro can be used, it must be installed, and this is described in the next section.

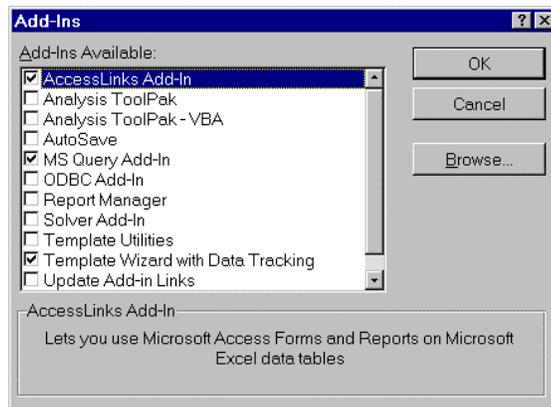
Macro Installation

To install the macro, you must add the software as an Excel add-in. Take the following steps:

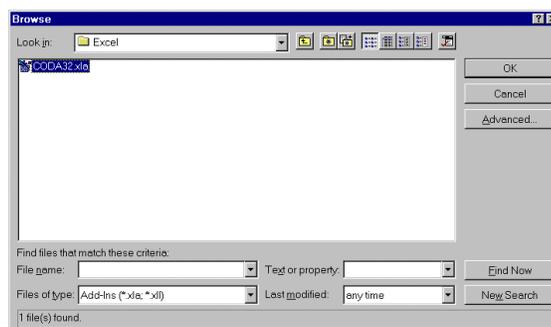
1. Copy the macro to a convenient location (the recommended location is `c:\easysoft\excel`).

If you obtain the software from the Easysoft web site your web browser should allow you to download and save the file in the newly created directory. If you obtain the software on floppy disk, simply copy it to the directory.

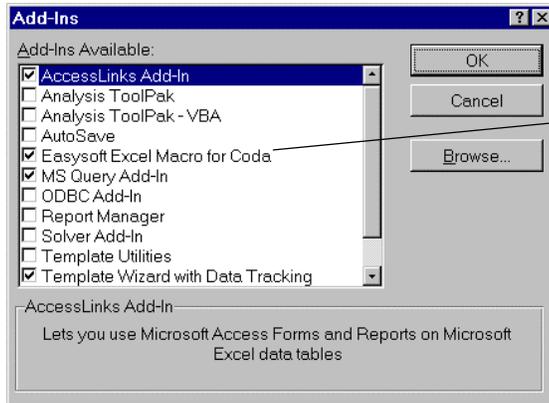
2. Start Microsoft Excel. Select **Add-Ins...** from the **Tools** menu. The Add-Ins dialog box appears.



3. Click the **Browse...** button, navigate to the `c:\easysoft\excel` directory, and highlight the `.XLA` file. Click the **OK** button.



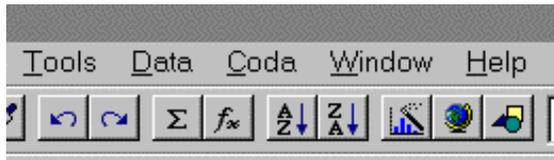
4. The Add-in now appears in the Add-Ins Available list box.



The Easysoft macro is now accessible.

Select the CODA add-in, and click the **OK** button.

5. The CODA macro is now available for use by clicking the newly added **Coda** menu which appears on the Excel menu bar.



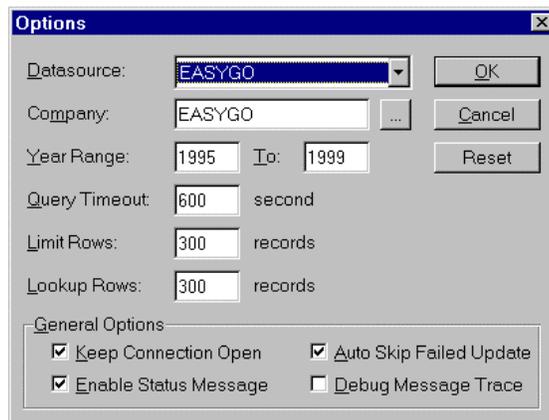
Upgrading the Macro

To upgrade the macro, overwrite the existing *.XLA file(s) for the macro with the new one(s).

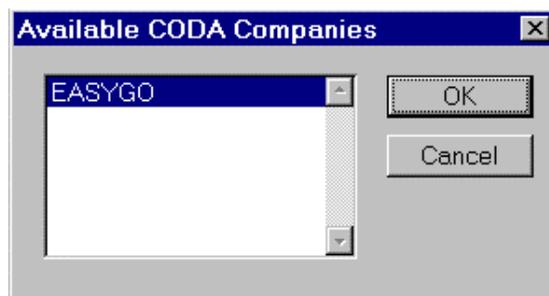
Initialisation

The first time the macro is used on a PC this initialisation step must be completed. It only has to be done once (unless you want to change the options); the information is saved, and when you, or another user, uses the macro in future, the information is already available for use.

1. To set the initialisation options, select **Options** from the **Coda** menu. The Options dialog box appears. The various settings you can choose are described in the remainder of this section.

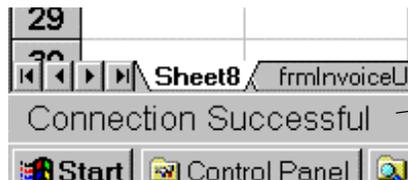


2. Select the required data source from the dropdown list box (in these examples we use EASYGO). Only data sources which use Easysoft drivers are shown in the dropdown list.
3. Enter the CODA company on which you want to operate. You cannot enter data directly into the entry box. Click the  button (lookup button) to see the available companies in a new dialog box.
4. In this example, there is just one CODA company in the database, and it happens to have the same name as the data source. Highlight the company you want and click the **OK** button.



The company will be entered into the list box on the Options dialog box.

5. The year range can be changed by typing in the upper and lower bounds of the years you want to see. By default, the lower value is the current year minus two, and the upper value is the current year plus two e.g. if the current year is 1997, then the range is 1995 to 1999.
6. Query Timeout refers to the maximum length of (clock) time that the query is allowed to take up. If the query is not completed within this time, it will be cancelled. The default value of 600 seconds can be changed if required.
7. Limit Rows is used to limit the number of rows initially returned by the query. If the query result contains more rows than the number specified, you will be asked whether or not you wish to see these. The default value of 300 rows can be changed if required.
8. Lookup Rows is used to limit the number of rows initially returned by lookup queries ("Lookups", page 10-20).
9. Keep Connection Open is used to keep the connection to the data source in an open state. This can speed up the accessing of data, but the trade-off is that if many users want to access the CODA data, there may not be enough licence slots available. By default, this option is selected.
10. Enable Status Message is used to show messages at the bottom of the worksheet e.g.



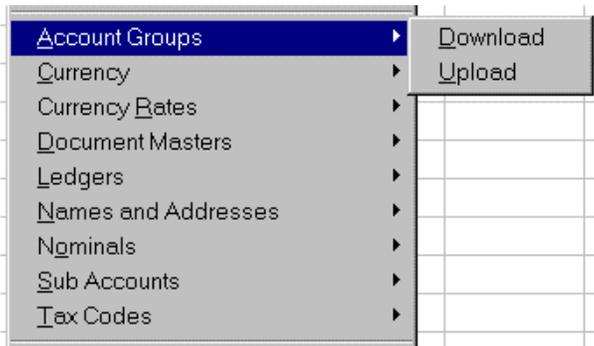
This is where you will see messages.

By default, this option is selected.

11. Auto Skip Failed update is used to prevent error messages which result from attempts at invalid updates being shown on screen. However, at the end of the update operation, all rows which the user wanted to update, but which were not updated will be flagged. By default, this option is selected.
12. Debug Message Trace. If problems are being experienced, this option should be selected. It results in messages being displayed on screen. By default, this option is not selected.
13. When all the information has been entered, press the **OK** button to accept it. Control passes back to Microsoft Excel, and you can now use the macro functions, which are described in the following sections. (Press the **Cancel** button to return to Excel without saving any changes you may have made. If you have not previously initialised the macro, you will not be able to use it).

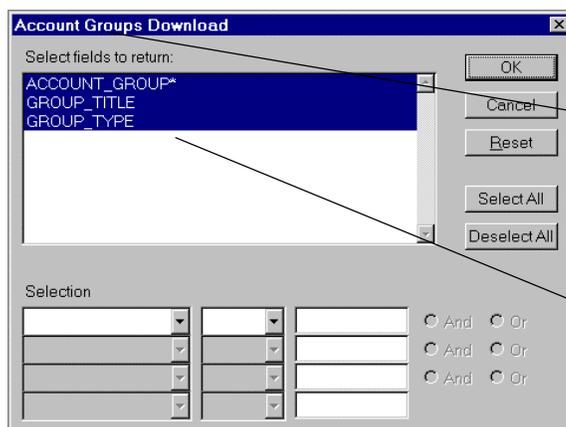
Masters-related options

All the menu options contained in this group follow a similar pattern. You can either download the data in the workbook, or you can upload it. These options are obtained by selecting the appropriate option as demonstrated below.



Download

If you download data, you will be presented with the Table Download dialog box. The operation of this dialog box is identical for all the masters, the only differences being the name of table and the columns that are available in the list box. For example, when you download account groups, the title of the dialog box is Account Groups Download. In the list box you will see the columns that are in the ACCOUNT_GROUP table.



This changes depending on the table to download.

The columns you see here are those available in the table.

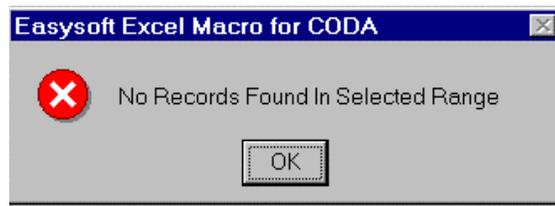
1. Highlight the column(s) that you want to download.

The **Select All** button can be used to select all the columns, and the **Deselect All** button is used to clear the selection.

An asterisk after the column name indicates that the column is part of the primary key.

2. Click the **OK** button to proceed with the download, or **Cancel** to abandon the operation.
3. The data will be returned to the worksheet.

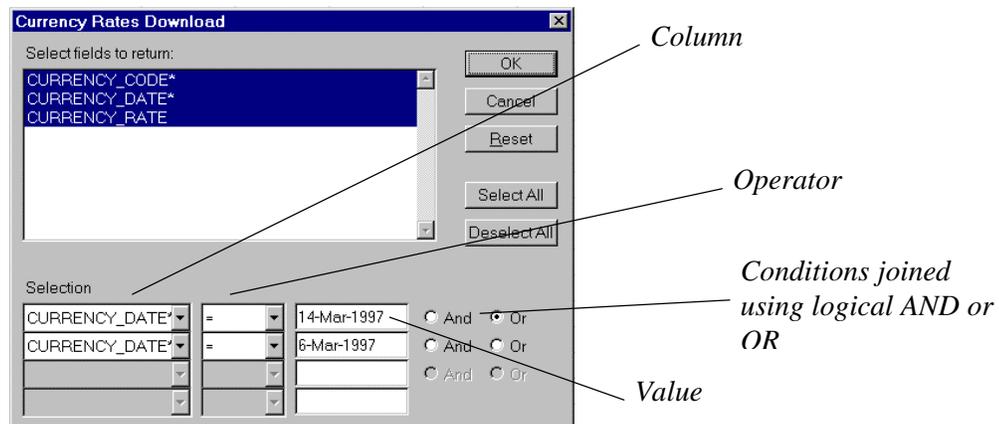
If the master does not contain any data, you will see the following message (click **OK** to return to the worksheet).



Selection criteria

You can set selection criteria to restrict the rows which are returned. Up to four simple condition clauses can be combined using logical AND and OR. This example uses Currency Rates: we will download all the fields for all currency rates dated either 6th March 1997 or 14th March 1997.

1. Select Coda, Currency Rates, Download to display the Currency Rates Download dialog box.



2. Select the fields you want to download.
3. Enter the following for the first selection criterion:

Column: **CURRENCY_DATE** (select using dropdown)
 Operator: **=** (select using dropdown)
 Value: **6-Mar-1997** (type required value)

4. Select **OR** to combine with the second criterion.

5. Enter the following for the second selection criterion:

Column: **CURRENCY_DATE** (select using dropdown)
 Operator: **=** (select using dropdown)
 Value: **14-Mar-1997** (type required value)

6. Click the **OK** button to download the data

Upload

After the upload has been performed, the colour of the primary key fields of the uploaded rows will change:

row colour	indicates
red	update failed
blue	row inserted successfully
green	row updated successfully

This example is for an upload of currency rates.

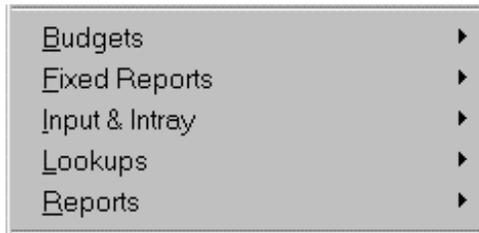
	A	B	C	D	E	F	G	H
1	CURRENCY_CODE	CURRENCY_DATE	CURRENCY_RATE					
2	USD	24-Mar-1997	246.24					
3	USD	14-Mar-1997	192.00					
4	USD	1-Jan-1997	205.20					
5	USD	25-Jan-2020	999.00					
6	UDS	2-Jan-1997	203.30					Easysoft Error: IASLINK, I'm sorry, that currency does not exist (7011)
7								

The rates for USD were first downloaded (**Coda, Currency_Rates, Download**), which resulted in the first three rows on the worksheet. Then two additional rows were created - the second additional row contains a typographical error. After the update was performed, the colours on the worksheet changed. The first three rows are green, indicating successful update. The fourth row is blue, indicating successful insert, the final row is red, indicating the attempted update was not successful. Additionally, the IASLINK error message is shown.

The rules for updates are as follows:

- First attempt an update (green)
- If the update is unsuccessful, then attempt to insert the row (blue)
- If the attempted insertion is unsuccessful, then indicate failure (red)

Reports-related options



The following options are available: Budgets, Fixed Reports, Input and Intraday, Lookups and Reports. Within each of these there are a number of options. All of these are described here.

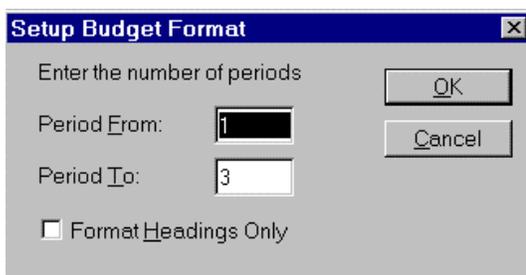
Budgets

There are three options under the Budgets menu, namely Setup, Download, and Upload.



Setup

This option is used to set up the upload options.



Period From and Period To limit the range of the periods that you want to upload.

The Format Headings Only check box can be used when a change is made to the period range. It prevents existing data in the worksheet from being deleted, since only the column heading is changed.

Download

The columns on which you can base selection criteria are shown in the dialog box.

Budget Download

Nominal: Like % ... OK

Subaccount: Like % ... Cancel

Level3: Like % ... Defaults

Budget: A

Budget: 1995

Periods: 1 To: 3

Wildcard Characters are percent % meaning a group of characters and underscore _ meaning a single character

The **Defaults** option resets the selection criteria to the default values.

Upload

Set the upload options, then click **OK** to proceed, or **Cancel** to quit.

Budget Upload

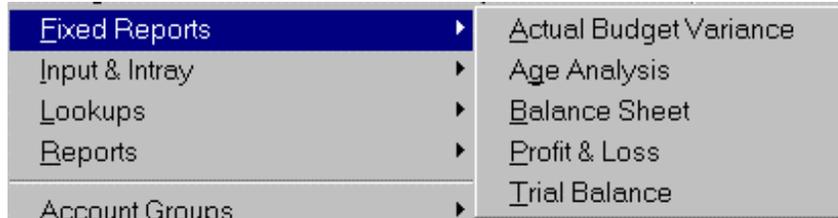
Budget: B OK

Budget: 1997 Cancel

Periods: 1 To: 3

Fixed Reports

There are five fixed reports: Actual Budget Variance, Age Analysis, Balance Sheet, Profit & Loss and Trial Balance.



Actual Budget Variance (ABV)

The ABV Report is used to compare the variation between Actuals and Budgets. Set any selection criteria that you require, then click the **OK** button.

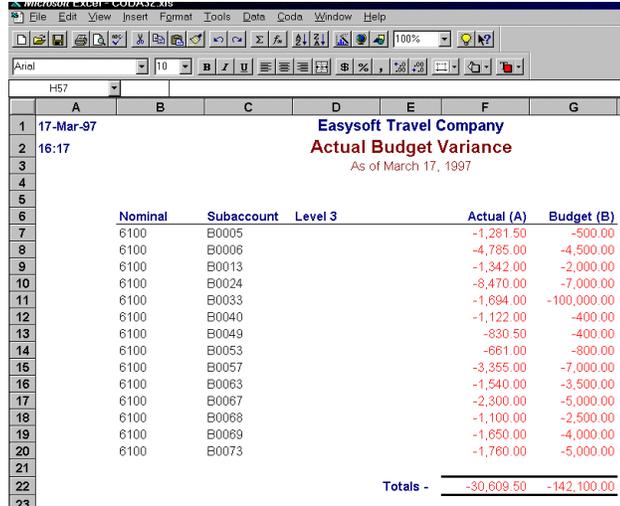
 A screenshot of a dialog box titled 'Budget Variance Selection'. It contains several input fields and buttons:

- Nominal: Like (dropdown), 6% (text), ... (button), OK (button)
- Subaccount: <> (dropdown), (empty text), ... (button), Cancel (button)
- Level3: = (dropdown), (empty text), ... (button), Defaults (button)
- Budget: 1997 (dropdown)
- Periods: 0 (text), To: 3 (text)
- Main Code: A (dropdown)
- Compariso: B (dropdown)

 A note on the right side of the dialog box reads: 'Wildcard Characters are percent % meaning a group of characters and underscore _ meaning a single character'.

The data is downloaded, and will appear similar to the report shown here.

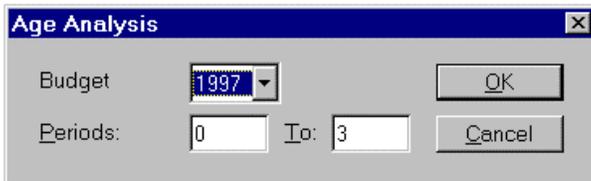
Example ABV Report



	A	B	C	D	E	F	G
1	17-Mar-97	Easysoft Travel Company					
2	16:17	Actual Budget Variance					
3		As of March 17, 1997					
4							
5							
6		<u>Nominal</u>	<u>Subaccount</u>	<u>Level 3</u>		<u>Actual (A)</u>	<u>Budget (B)</u>
7		6100	B0005			-1,281.50	-500.00
8		6100	B0006			-4,785.00	-4,500.00
9		6100	B0013			-1,342.00	-2,000.00
10		6100	B0024			-8,470.00	-7,000.00
11		6100	B0033			-1,694.00	-100,000.00
12		6100	B0040			-1,122.00	-400.00
13		6100	B0049			-830.50	-400.00
14		6100	B0053			-861.00	-800.00
15		6100	B0057			-3,355.00	-7,000.00
16		6100	B0063			-1,540.00	-3,500.00
17		6100	B0067			-2,300.00	-5,000.00
18		6100	B0068			-1,100.00	-2,500.00
19		6100	B0069			-1,650.00	-4,000.00
20		6100	B0073			-1,760.00	-5,000.00
21							
22						Totals -	-30,609.50 -142,100.00

Age Analysis

This displays a list of customers who have not paid. Set any selection criteria that you require, then click the **OK** button.



Age Analysis

Budget: 1997

Periods: 0 To: 3

OK Cancel

Two sheets are created; a worksheet and a pivot table. The report defaults to the pivot table sheet, and you will see something similar to this.

Example Age Analysis Report

	A	B	C	D	E	F
1	17-Mar-97			Easysoft Travel Company		
2	16:28			Age Analysis		
3				As of March 17, 1997		
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						
17						
18						
19						
20						

Sum of VALUE	PERIOD			Grand Total
SUBACCOUNT	1	2	3	
H0001	-1100	-2250	-215	-3565
H0002	-2000	-820	-400	-3220
H0003	-7700	0	0	-7700
H0004	-1540	0	0	-1540
H0005	-320	-700	0	-1020
H0006	-355	0	-400	-755
H0007	-210	-300	0	-510
H0008	-2500	0	-550	-3050
H0009	-1400	0	0	-1400
H0010	-2070	0	0	-2070
H0011	-2500	0	0	-2500
H0012	0	0	0	0

Balance Sheet

Each category has its own selection criteria.

Balance Sheet Selection Criteria

Financial Year: 1997

Period Range: 0 To: 3

Categories:

- Fixed Assets
- Current Assets
- Capital
- Long Term Liabilities
- Current Liabilities

Selection

NOMINAL Like 1%

And Or

And Or

And Or

The data is downloaded, and will appear similar to the report shown here.

Example Balance Sheet Report

	A	B	C	D	E	F	G
1	18-Mar-97			Easysoft Travel Company			
2	10:33			Balance Sheet			
3				As of March 18, 1997			
4							
5							
6				ASSETS			
7				Fixed Assets			
8				1110 Fixtures & Fitting Costs			2,000.00
9				1120 Fixtures & Fitting Deprecia			-200.00
10				1310 Computer Equipment Costs			3,500.00
11				1320 Computer Equipment Depr			-600.00
12				1410 Motor Vehicles Costs			9,980.00
13				1420 Motor Vehicles Depreciatic			-2,000.00
14				1510 Office Equipment Costs			1,700.00
15				1520 Office Equipment Deprecia			-600.00
16							
17				Total Fixed Assets			13,780.00
18							
19				Current Assets			
20				1110 Fixtures & Fitting Costs			2,000.00
21				1120 Fixtures & Fitting Deprecia			-200.00
22				1310 Computer Equipment Costs			3,500.00
23				1320 Computer Equipment Depr			-600.00
24				1410 Motor Vehicles Costs			9,980.00
25				1420 Motor Vehicles Depreciatic			-2,000.00
26				1510 Office Equipment Costs			1,700.00
27				1520 Office Equipment Deprecia			-600.00
28				2100 Sales			31,891.00
29				2210 Bank Current Account			22,000.00

Profit and Loss

Use this report to see the profits and losses. Enter the required criteria, if any, then click the **OK** button.

The screenshot shows a dialog box titled "Profit & Loss Selection". It contains the following elements:

- Financial Year:** A dropdown menu set to "1997".
- Period Range:** Two input boxes, the first containing "0" and the second containing "3".
- Categories:** A list box containing "Sales", "Purchases", and "Expenses". "Sales" is currently selected.
- Buttons:** "OK", "Cancel", and "Reset" buttons are located on the right side.
- Selection:** A section at the bottom with three rows of dropdown menus and radio buttons. Each row has two radio buttons labeled "And" and "Or".

The example report below was generated using the following criteria:

Category Selection Operator Value

Sales	NOMINAL	like	6%
Purchases	NOMINAL	like	7%
Expenses	NOMINAL	like	8%

Example Profit and Loss Report

	A	B	C	D	E	F	G	H
1	18-Mar-97							
2	10:45			Easysoft Travel Company				
3				Profit & Loss				
4				As of March 18, 1997				
5								
6								
7				Sales				
8				6100 Sales Analysis - Holidays				-31,891.00
9								
10				Total Sales				<u>-31,891.00</u>
11								
12				Purchases				
13				7100 Purchase Analysis - Holidays				28,930.00
14								
15				Total Purchases				<u>28,930.00</u>
16								
17				Expenses				
18				8200 Salaries				5,680.00
19				8300 Motor Vehicles				135.00
20				8610 Accountant				820.00
21								
22				Total Expenses				<u>6,635.00</u>
23								
24								TOTAL:- <u>3,674.00</u>

Trial Balance

This displays a trial balance for the selected year and period range. Set the selection criteria that you require, then click the **OK** button.

The dialog box titled "Trial Balance - Nominal" contains the following elements:

- Budget:** A dropdown menu currently showing "1997".
- Periods:** An input field containing the number "0".
- To:** An input field containing the number "3".
- Buttons:** "OK" and "Cancel" buttons.

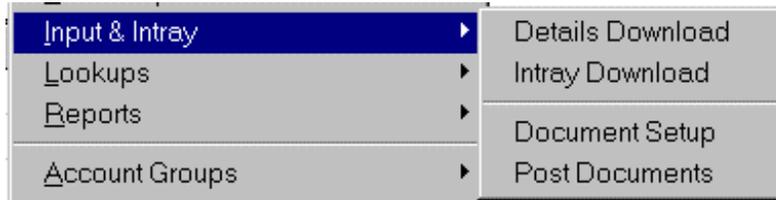
The data is downloaded, and will appear similar to the report shown here.

Example Trial Balance

	A	B	C	D	E	F	G
1	17-Mar-97		Easysoft Travel Company				
2	16:58		Trial Balance				
3			As of March 17, 1997				
4							
5			Standard Trial Balance for Year:- 1997 Start Period:- 0 End Period:- 3				
6							
7			Nominal	Description		Debit	Credit
8			1110	Fixtures & Fitting Costs		2,000.00	
9			1120	Fixtures & Fitting Depreciation			200.00
10			1310	Computer Equipment Costs		3,500.00	
11			1320	Computer Equipment Depreciation			600.00
12			1410	Motor Vehicles Costs		9,980.00	
13			1420	Motor Vehicles Depreciation			2,000.00
14			1510	Office Equipment Costs		1,700.00	
15			1520	Office Equipment Depreciation			600.00
16			2100	Sales		31,891.00	
17			2210	Bank - Current Account		32,040.00	
18			3100	Capital			50,000.00
19			5100	Purchases			30,062.12
20			5200	Payroll			1,500.00
21			5300	VAT Control Account		167.12	
22			6100	Sales Analysis - Holidays			31,891.00
23			7100	Purchase Analysis - Holidays		28,930.00	
24			8200	Salaries		5,680.00	
25			8300	Motor Vehicles		135.00	
26			8610	Accountant		820.00	
27							
28						Totals -	
						116,843.12	116,843.12

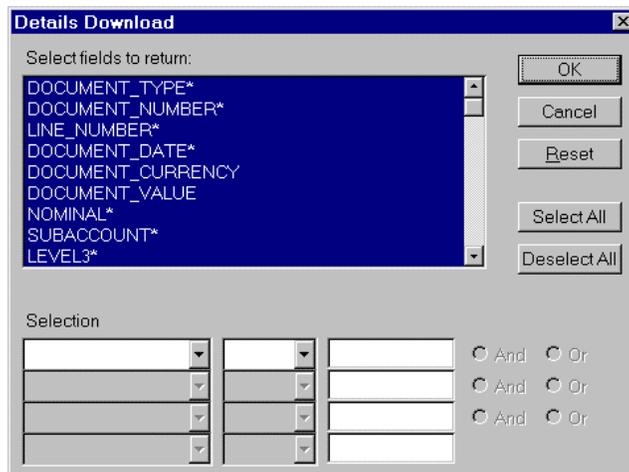
Input and Intray

There are four options available: Details Download, Intray Download, Invoice Setup and Post Documents. The operation of the Details Download and Intray Download options is similar, so only the first is shown.



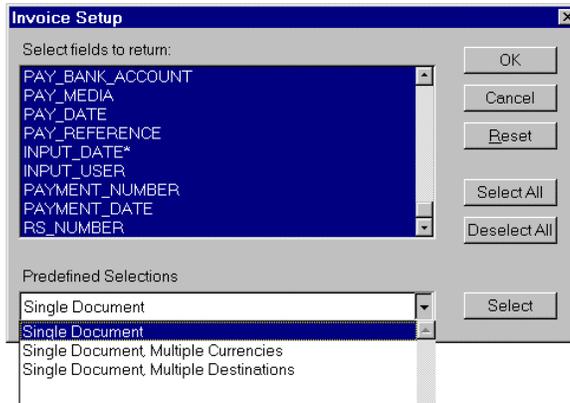
Details Download

Use this option to download details of the documents. Set any selection criteria you want, then click the **OK** button.



Document Setup

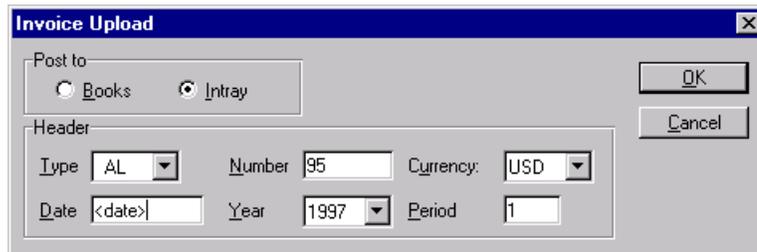
This is a preparatory step, needed before an upload can take place. Select the fields you want to update in CODA.



The Predefined Selections option can be used to select fields rapidly. Choose the option you require, then click the **Select** button. The fields are automatically selected in the Select fields to return list box.

Post Documents

When you select this option, the Invoice Upload dialog box appears. Select the options you require, then click the **OK** button.



The values in the dialog box are defaults, and are only used if any required details are missing from the worksheet.

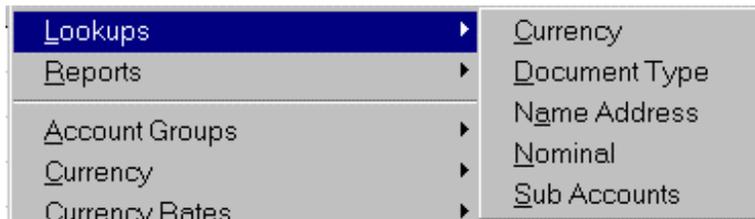
Number refers to the document number. It is automatically entered if autosequence rules are used in CODA, otherwise, it must be entered manually.

If `LINE_NUMBER` was not selected in the Invoice Setup, then line numbers are automatically formatted such that if there is a change in the value of a `DOCUMENT_TYPE` or `DOCUMENT_NUMBER` column in the worksheet, then the `LINE_NUMBER` value will be reset to correspond to these changes.

If the document format in CODA is JN, then the first line of the worksheet data should be left blank.

Lookups

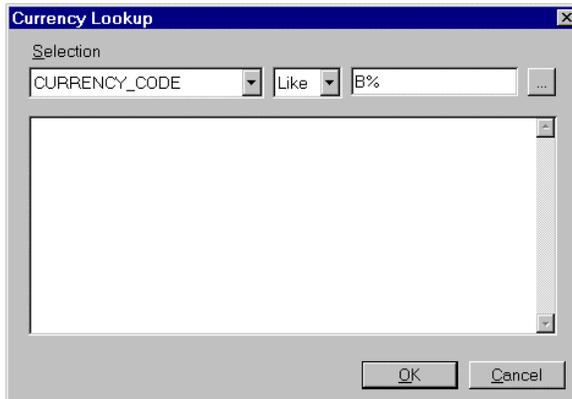
Lookups allows you to see all the values for a given category. This lets you ascertain, for example, whether a given value you want to enter is valid.



There are five lookup options: Currency, Document Type, Name Address, Nominal and Sub Accounts. The operation of these is identical, and so only one example, using Currency, is shown.

In the Selection dropdown enter the column on which you want to search. From the central dropdown, select the comparison operator. In the rightmost entry field type the characters you want to use in the search. You may use the percent character (%) as a wildcard to represent zero or more characters, and the underscore character (_) to represent one character.

Press the  button (lookup button) button to display the required values.



Press **OK** (or **Cancel**) to exit.

Reports

This option allows you to create reports tailored to your exact needs. There are two options, Setup New Report, which defines the report, and Run Report.



Setup New Report

	A	B	C	D	E	F	G	H
1	Report Name	Author	Company					
2	simple	Mike	Easygo					
3								
4	Worksheet	Row	Column	Headings	Format	SQL	Last Rows	Last Time
5	Sheet1	1	1	YES	YES	SELECT * FROM EASYGO_LEDGER		
6								

1. Enter values for Report Name, Author and Company at the top of the report. These are optional, and do not affect the output.
2. Worksheet Row. This specifies the location of the output. It must be a worksheet that currently exists, and it must not be the worksheet on which the Setup Report resides. Remember to enter the name of the worksheet exactly as it appears on the tabs at the bottom of the screen.
3. Row and Column specify the first row and column respectively on which the data is to be returned.
4. Headings. If **YES** is specified, then column headings will be included in the report. This option may be left blank (i.e. headings not shown).
5. Format. If **YES** is specified, then the width of each column is adjusted to fit the data returned. This option may be left blank (i.e. no formatting).
6. SQL. This is the SQL that is sent to the server. Multiple rows of SQL are allowed.
7. Save the worksheet if you will want to use the report again. The report is now ready to run.

Note: Last Rows and Last Time are filled in automatically when the report is run. They specify the number of rows returned and the finishing time of the report.

Run Report

1. Select the worksheet that contains the report you wish to run.
2. Run the report (**Coda, Reports, Run Report**). Example output:

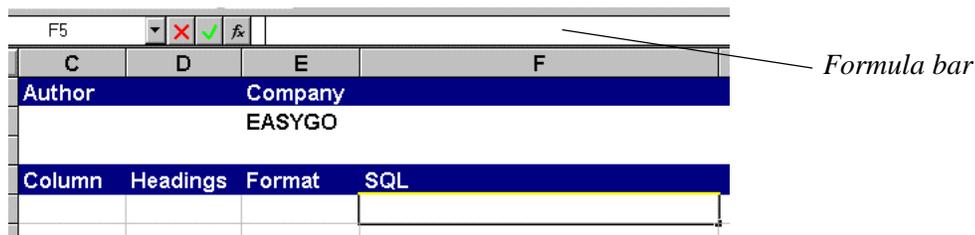
	A	B	C	D	E	F	G
1	LEDGER	NOMINAL	DESCRIPTION	CREATED_USER	CREATED_DATE	MODIFIED_USER	MODIFIED_DATE
2	E0001		VAT Ledger	CAROLYN	4-Mar-1997	TRAINER	5-Mar-1997
3	E0002		Fixtures & Fittings	CAROLYN	4-Mar-1997	TRAINER	5-Mar-1997
4	E0003		Holidays	CAROLYN	4-Mar-1997	TRAINER	5-Mar-1997
5	E0004		Brochures	CAROLYN	4-Mar-1997	TRAINER	5-Mar-1997
6	E0005		Destination	CAROLYN	4-Mar-1997	TRAINER	5-Mar-1997
7	P0001	5100	Purchase Ledger	CAROLYN	4-Mar-1997	TRAINER	5-Mar-1997
8	S0001	2100	Sales Ledger	CAROLYN	4-Mar-1997	TRAINER	5-Mar-1997

Advanced Techniques

To make the report easily transferable for any company, you can use a cell reference to the name of the company. If you change the company name, then all the references would change, so you would only need to make one global change to the worksheet. This is useful if there is much SQL.

A simple example follows (for clarity, only the relevant information is shown on the screen shots). Say we want this SQL query: `SELECT * FROM <company>_LEDGER`, where <company> is the name of the company that is displayed in the Company cell. When the company name is changed, the name will change in the SQL.

1. Enter the name of the company in the Company cell.
2. Select the cell in which you want to write the SQL.
3. Move the cursor to the Formula bar.

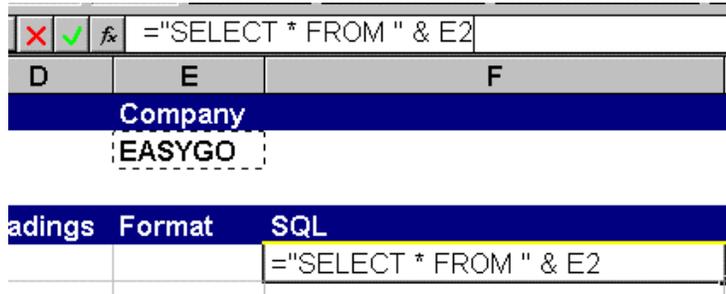


4. Press the equals (=) key (this indicates that a formula is to follow).
5. Type: `"SELECT * FROM "`

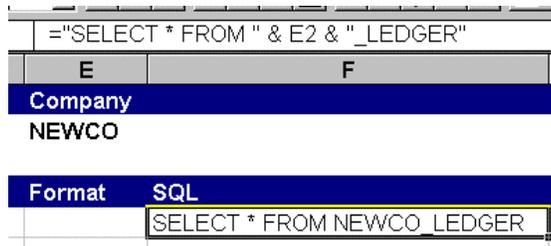
Ensure you type the quote marks, and leave a space before the closing quote. Notice how the text also appears in the SQL cell (see next screen shot).

6. Press the space bar.

7. Press the ampersand (&) key.
8. Press the space bar.
9. Select the cell (E2) containing the company name.
10. Notice how the value E2 is added to the formula (you could have typed it directly).



11. Press the space bar, the ampersand (&) key, and the space bar.
12. Type the remainder of the SQL: `\"_LEDGER\"`
13. Press the Enter key. The SQL is ready to run.
14. Notice how the formula contains a reference to cell E2 which contains the company name. If you change the name, then this change will automatically be reflected in the SQL.

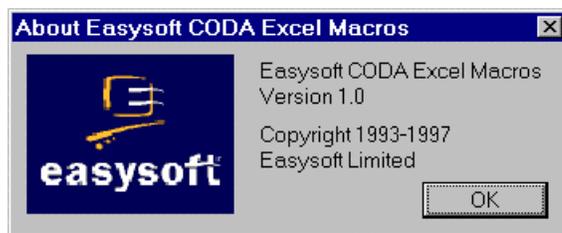


Information

Two menu options are available. One gives information about the macro, and the other displays contact information.

About Easysoft ODBC for CODA

Use this option to obtain version information about the macro.



Contacting Easysoft

Use this menu option to list the support contact options.

Appendices

Contents

Appendix A. Global Glossary	2
Appendix B. CODA Table Descriptions	7
Appendix C. CODA IAS Overview	27
Company Structure	27
Using CODA	28
External Access to CODA Data	30
CODA-IAS Review Questions	31
CODA-IAS Review Answers	32
Appendix D. Easysoft ODBC PC Installation	34
Starting the Driver Installation	34
Appendix E. Troubleshooting	38
Easysoft ODBC Logging	39
EASYSOFT.INI	40
Example Server Log File	41
Microsoft Trace Options	42
Version 3.0 Trace Options	42
SQL.LOG Described	43
How to Work Out What the Server is Doing	44
Find the Process Causing the Problem	44
View Log File	45
Stop the Process	46
Appendix F. Using Easysoft Support Services	47
Support Check List	48
The Easysoft FTP Site	49
Sending Data to Easysoft	50
Global Index	51

APPENDIX A

Global Glossary

\$ DCL prompt.

! DCL comment separator.

Account (accounting term) Details of transactions of a given type (i.e. items in a journal). Each account has a name (Account title) and a number (Account number).

ANSI (American National Standards Institute) The primary formal standards-making body in the United States.

API (Application Programming Interface). A set of related functions that a computer programmer uses to obtain some kind of service from another piece of software.

Application (In ODBC terms) a program that processes data and which runs under Microsoft Windows.

Block (In general terms) the smallest unit of space into which a mass storage device can be divided.

(In OpenVMS terms) The smallest logically addressable unit of data that a device can transfer in an I/O (Input/Output) operation. Typically, 512 contiguous bytes.

Boolean Boolean logic is the two-valued logic of true and false.

Catalog (In SQL terms) a named collection of one or more tables grouped together. The catalog contains definitions that describe SQL features of application databases, such as which columns belong to which tables, user privileges, etc.

Chart of accounts A list of account titles and account numbers of the accounts in a general ledger i.e. a description of the structure of a general ledger.

CLI (Call Level Interface) A set of function calls that a computer programmer uses to obtain some kind of service from another piece of software. CLI is used with reference to SQL standards to describe an interface that is not embedded SQL.

Column the vertical dimension of a table (cf. row). The field at the intersection of a row and a column holds the value in accordance with the data type specified.

Comma Separated Text see CSV.

Command An instruction (usually an English word) that specifies an operation to be performed.

Conformance level (API) Refers to the set of ODBC functions that an ODBC driver supports.

Conformance level (SQL) Refers to the set of SQL functions that an ODBC driver supports.

Control account (accounting term) An account which is used to check the accuracy of a ledger. It summarises the information in the accounts contained in a given ledger.

CST (Comma Separated Text). see CSV.

CSV (Comma Separated Values). CSV is text that defines data values. The values are separated by commas (same as CST).

Data source A set of database files plus (if appropriate) the associated operating system, DBMS and network.

Data type A data type is the specification of permitted values. A data type limits the values which are allowed to be used.

Database A collection of data files.

DBMS (Database Management System). Software that handles access to a database.

DECNET A proprietary communication protocol produced by Digital Equipment Corporation (DEC).

Default A value automatically inserted into a table in the case where a user has not specified a value to be used.

DCL Digital Command Language. The standard common interface to Digital's major operating systems.

DCL prompt By default this is the dollar sign (\$). It indicates that DCL is ready to accept a command.

DLL (Dynamic Link Library). A shared library of code that is loaded in and out of memory as and when it is needed.

Download(ing) In non-technical terms downloading is taking data from the Server and storing it on the Client PC (cf. uploading).

Driver See ODBC driver.

Driver Manager A dynamic link library (DLL) provided by Microsoft, the main function of which is to load ODBC drivers.

Entry (accounting term) An item recorded in a journal.

Field A segment of a data record.

Financial statement (accounting term) A summary of the financial information contained in the accounts.

FTP (File Transfer Protocol). A standard method (rfc 959) of transferring files between different machines.

Function (SQL) A function in SQL takes a *scalar value* or a set of scalar values and returns a scalar value.

General ledger (accounting term) A ledger that contains all accounts needed to prepare financial statements. Contains all accounts except those for customers and suppliers. Also known as a nominal ledger.

Interoperability An application that is interoperable is one that can access many different databases.

ISO (International Standards Organisation) A world-wide federation of national standards bodies that sets standards for a variety of technologies.

Journal (accounting term) A record of accounting information in chronological order.

Ledger (accounting term) (also see general, purchase and sales ledgers) A group (also known as book) of related accounts.

Ledger nominal (subaccount) (CODA) The ledger specified in the nominal (subaccount) master record.

Literal A literal is a way of representing a value. Each value has a *data type*, and for each data type there is a corresponding literal specification.

Local Data type A local data type represents the storage format of the data within the data files on the Server. This local data type is mapped onto an SQL data type within the Easysoft SQL Engine.

Logical see Logical name.

Logical name In the OpenVMS and VMS operating systems, an alias for a file specification (can include directories, subdirectories, input/output devices). It is a named variable which is replaced by a value when it is used.

Multi-tier driver (also see single-tier driver) In a multi-tier configuration the ODBC driver sends SQL requests to a server that processes those requests. Typically, multiple-tier systems are divided across platforms.

Nominal (CODA) The account number of the nominal as defined in the nominal master record.

Nominal Ledger (accounting) Another name for the general ledger.

ODBC (Open DataBase Connectivity). An industry standard defined by Microsoft which is by software that allows communication between different database systems.

ODBC driver The software that implements ODBC function calls. Each driver is specific to an application.

OpenVMS An operating system that runs on Alpha and VAX machines (see VMS).

Operating system (1) An integrated collection of programs that controls the execution of computer programs and performs system functions. (2) Software that organises a central processor and peripheral devices into an active unit for the development and execution of programs.

Packet In communications technology, a packet is the smallest unit of information that can be transmitted over a network.

Posting (accounting term) The transfer of information from journal entries to ledger accounts.

Predicate A predicate is a simple statement which can be true, false or sometimes unknown.

Privilege A privilege is the right to perform a particular action (e.g. INSERT) on an object (e.g. table).

Purchase ledger (accounting) A ledger for suppliers' accounts.

Record A collection of related data items treated as a unit. A record contains one or more fields.

RMS (Record Management Services) A DIGITAL-specific term. A set of routines that is used to manipulate data files. The OpenVMS high-level file system.

Routine A set of computer instructions that perform an operation.

Row A row is the horizontal dimension of a table (cf. column). A row in its most basic roll equates to a record within a file.

Sales ledger (accounting) A ledger for customers' accounts.

Scalar value A scalar value is in principle any value that can be assigned to a column.

Search condition A search condition is a statement or group of statements joined together by boolean operators which results in a true, false or unknown condition.

Server A server can be considered as an engine providing some service. A client calls the server, resulting in the service being performed.

Set function (SQL) An operation on a set of values in a column of a table or all values from a column in a group of rows in a table.

Single-tier driver (also see multi-tier driver) An ODBC implementation in which the data is processed directly by the ODBC driver.

SQL (Structured Query Language). A standard language for interacting with relational database systems.

SQL statement SQL statements can be categorised as data manipulation, data definition or SQL control statements.

Syntax The structure of a language, command or statement.

System data source A data source which can be accessed by any user on a given computer cf. user data source.

Table A table consists of column definitions and is the view a user sees of data definitions they have defined. You can consider a table as a rectangular sheet containing *columns* and *rows*. Each intersection (cell) can contain a value.

TCP/IP (Transmission Control Protocol/Internet Protocol). A standard method (rfc 793) of accessing data on different machines.

UIC (in VAX/OpenVMS terms) The abbreviation for user identification code. A 32 bit value assigned to users and files that specifies the allowed operations.

Upload(ing) In non-technical terms uploading is taking data from the Client PC and storing it on the Server (cf. downloading).

User data source A data source which can be accessed by a specific user only cf. system data source.

User Identification Code see UIC.

VAX (Virtual Address eXtension). A type of computer produced by Digital Equipment Corporation (DEC) which can use the VMS and OpenVMS operating systems.

VMS (Virtual Management System). An operating system that runs on VAX and Alpha machines. Superseded by OpenVMS.

X/Open SQL Access Group An industry consortium of DBMS vendors whose objective is to enable SQL-based products from multiple vendors to work together.

APPENDIX B

CODA Table Descriptions

CODA table name	Contains	CODA menu	I	U	D
ACCOUNT	Easysoft optimisation	-	×	×	×
ACCOUNT_GROUP	Account Group information	MM AG			×
BALANCE	Budgets	BU			
BUDGET	Budgets	BU			
COMPANY	CODA Company information	MA GM CO	×	×	×
CURRENCY	Currencies	MM CM		×	×
CURRENCY_RATE	Currency rates	MM CM			×
DESTINATION	Intercompany Destination	IC MD			
DETAILS	Input to books	IN IN		×	×
DETAILS_COMMENT	Details comments	IN IN		×	
DOCUMENT_MASTER	Document Masters	MM DO	×		×
GROUP_HIERARCHY	Account Group hierarchy	MM AG		×	
INTRAY	Input to Intraday	IN AD			×
INTRAY_COMMENT	Intraday comments	IN AD		×	
LEDGER	Ledgers	MM LE			×
NAME	Name and Address	MM NA			
NAME_COMMENT	Name and Address account comments	MM NA		×	
NAME_CONTACT	Name and Address contact list	MM NA		×	
NAME_LEDGER_COMMENT	Name and Address ledger comments	MM NA		×	
NAME_OWNER	Name and Address financial details	MM NA			
NOMINAL	Nominal accounts	MM NO			
NOMINAL_COMMENT	Nominal accounts comments list	MM NO			×
SUBACCOUNT	Subaccounts	MM SA			
SUBACCOUNT_COMMENT	Subaccount comments	MM SA		×	
SUBACCOUNT_OWNER	Subaccount Nominal/ledger list	MM SA			
TAX_CODE	Tax Codes	MM TX			
TEXT	Field labels, messages in CODA	MA GM TE	×	×	×

The operations that are *disallowed* for CODA version 6.7 are shown above. The disallowed operations are indicated using the “×” character (I = INSERT, U = UPDATE, D = DELETE).

Column name	Data type	Size
NOMINAL	VARCHAR	12
SUBACCOUNT	VARCHAR	12
LEVEL3	VARCHAR	12
NOMINAL_ID	INTEGER	4
SUBACCOUNT_ID	INTEGER	4
LEVEL3_ID	INTEGER	4

ACCOUNT is an Easysoft file which is used to reference internal CODA identification numbers. It is generated when the Codaxref routine is run (see Module 6). It is shown here, as it appears alongside the CODA tables when an application connects to a CODA data source. It adds keys which are not available in CODA.

Index name	No	Dup	Coll	Fields in index
ACCOUNT_001	1	No	A	NOMINAL, SUBACCOUNT, LEVEL3
ACCOUNT_002	2	Yes	A	NOMINAL
ACCOUNT_003	3	Yes	A	SUBACCOUNT
ACCOUNT_004	4	Yes	A	LEVEL3

Column name	Data type	Size	Def	CODA name	IASLINK name
ACCOUNT_GROUP	VARCHAR	12		Account group	LB-GROUP-CODE
DESCRIPTION	VARCHAR	30		Group title	LB-GROUP-TITLE
GROUP_TYPE	VARCHAR	1	N	Group type	LB-GROUP-TYPE

The ACCOUNT_GROUP table holds information on hierarchical account groups. The details are kept in the GROUP_HIERARCHY table.

Index name	No	Dup	Coll	Fields in index
ACCOUNT_GROUP_001	1	No	A	ACCOUNT_GROUP

Column name	Data type	Size	Def	CODA name	IASLINK name
NOMINAL	VARCHAR	12		Nominal	LB-BALANCE-NOM
SUBACCOUNT	VARCHAR	12		Subaccount	LB-BALANCE-SUB
LEVEL3	VARCHAR	12		Level3	LB-BALANCE-LV3
BUDGET_YEAR	SMALLINT	2		Year	LB-BALANCE-YEAR
BUDGET_CODE	VARCHAR	1	B	Budget code	LB-BALANCE-BUD
PERIOD [note 1]	SMALLINT	2		Period	LB-BALANCE-PERIOD
BALANCE	DOUBLE	8		Value	LB-BALANCE-VALUE
ACCOUNT_BALANCE [note 2]	VARCHAR	1		-	-

Note 1: Valid range for the PERIOD field is between 1 and the number of periods in the accounting year. Additionally, there are three other allowed values:

-2, indicates an adjustment period

-1, indicates a final period

0, indicates an opening period

Note 2: ACCOUNT_BALANCE is an internal flag (allowed values: Y, N) which is not seen on the ODA screen. Y indicates that the values shown in the record have been entered as data to the system. This is the normal balance of the account. N indicates that the values shown exist only as calculated data i.e. the total of N/S/L3 or a combination of N/S/L3.

Index name	No	Dup	Coll	Fields in index
BALANCE_001	1	No	A	NOMINAL, SUBACCOUNT, LEVEL3, BUDGET_YEAR, BUDGET_CODE, PERIOD
BALANCE_002	2	Yes	A	NOMINAL
BALANCE_003	3	Yes	A	SUBACCOUNT
BALANCE_004	4	Yes	A	LEVEL3

Column name	Data type	Size	Def	CODA name	IASLINK name
NOMINAL	VARCHAR	12		Nominal	LB-BUDGET-NOM
SUBACCOUNT	VARCHAR	12		Sub-account	LB-BUDGET-SUB
LEVEL3	VARCHAR	12		Level-3	LB-BUDGET-LV3
BUDGET_YEAR	SMALLINT	2		Year & period range	LB-BUDGET-YEAR
BUDGET_CODE	VARCHAR	1	B	Budget	LB-BUDGET-BUD
ACCOUNT_BALANCE [note 1]	VARCHAR	1		-	-
ADJUSTMENT_VALUE	DOUBLE	8		Value	LB-BUDGET-ADJUSTMENTS
FINAL_VALUE	DOUBLE	8		Value	LB-BUDGET-FINAL
OPENING_VALUE	DOUBLE	8		Value	LB-BUDGET-OPENING
TOTAL_VALUE	DOUBLE	8		Total value	[note 2]
VALUE_n [note 3]	DOUBLE	8		[number]	LB-BUDGET-VALUE(<i>n</i>)

Note 1: ACCOUNT_BALANCE is an internal flag (allowed values: Y, N) which is not seen on the CODA screen. Y indicates that the values shown in the record have been entered as data to the system. This is the normal balance of the account. N indicates that the values shown exist only as calculated data i.e. the total of N/S/L3 or a combination of N/S/L3.

Note 2: TOTAL_VALUE contains a computed value which is the sum (for any given record) of the data in all the value columns except FINAL_VALUE.

Note 3: The BUDGET table contains a variable number of VALUE columns depending upon the number of budget periods. The column VALUE_n shown here represents one or more columns in the CODA BUDGET table which have sequential numbers starting from 0001 and finishing at 9999. Thus a maximum of 9999 budget periods are supported (in addition to the opening, final and adjustment periods).

Index name	No	Dup	Coll	Fields in index
BUDGET_001	1	No	A	NOMINAL, SUBACCOUNT, LEVEL3, BUDGET_YEAR, BUDGET_CODE
BUDGET_002	2	Yes	A	NOMINAL
BUDGET_003	3	Yes	A	SUBACCOUNT
BUDGET_004	4	Yes	A	LEVEL3

Table 10. COMPANY column details					
Column name	Data type	Size	Def	CODA name	IASLINK name
COMPANY_NAME	VARCHAR	40		Name	LB-COMPANY-NAME
TELEPHONE	VARCHAR	20		Telephone	LB-COMPANY-PHONE
FAX	VARCHAR	20		Fax	LB-COMPANY-TELEX
COMPANY_YEAR	SMALLINT	2		Year	LB-COMPANY-YEAR
CURRENT_PERIOD	SMALLINT	2		Period	LB-COMPANY-PERIOD
POST_QUANTITIES_TO_ALL_LEVELS	VARCHAR	1		Balance qnts at NS3	LB-COMPANY-QTYNS3
AUTOMATIC_BATCH_NUMBERS	VARCHAR	1		Automatic batch nos	LB-COMPANY-AUTO-BATCH
LOGS_CHANGES_TO_MASTER_FILES	VARCHAR	1		Master file logging	LB-COMPANY-MFLOG
ALLOWS_DEFERRED_POSTING	VARCHAR	1		Deferred postings	LB-COMPANY-DEFINP
DECIMAL_POINT_CHARACTER	VARCHAR	1		Decimal point	LB-COMPANY-DPT-CHAR
PERIODS_IN_CURRENT_YEAR	SMALLINT	2		Prds/Year	LB-COMPANY-PRDS-THIS-YEAR
ACCUMULATES_FOREIGN_BALANCES	VARCHAR	1		Foreign balances	LB-COMPANY-FOREIGN-BAL
COMPANY_LANGUAGE	VARCHAR	2		Language	LB-COMPANY-LANG
DATE_FORMAT	VARCHAR	1		Date Format	LB-COMPANY-DATE-TYPE
DATE_SEPARATOR_CHARACTER	VARCHAR	1		Separator	LB-COMPANY-DATE-SEP
COMMITMENT_ACCOUNTING	VARCHAR	1		Commitment accounting	LB-COMPANY-COMACC
CURRENCY_LOCATION	VARCHAR	1		Company or group	LB-COMPANY-CURRLOCN
CURRENCY_RULE	VARCHAR	1		rate rule	LB-COMPANY-CURRRULE
CURRENCY	VARCHAR	4		code	LB-COMPANY-CURRENCY
BANK_NAME	VARCHAR	12		Default bank account	LB-COMPANY-BANK-NAME
EARLIEST_YEAR_WITH_DATA	SMALLINT	2		Earliest year with data	LB-COMPANY-BASE-YEAR
EARLIEST_YEAR_TO_POST_TO	SMALLINT	2		Earliest year to post to	LB-COMPANY-MIN-YEAR
FIXED_ASSETS	VARCHAR	1		[CODA internal]	LB_COMPANY_INC_FA
POST_DEBITS	VARCHAR	1		Are your debits +ve	LB-COMPANY-POSDEBITS
SPECIAL_TAX_CODE	VARCHAR	2		Special tax code	LB-COMPANY-SPEC-TAX-CODE
ADDRESS_FORMAT	VARCHAR	2		Address format	LB-COMPANY-AFMT
BANK_FORMAT	VARCHAR	2		Bank Format	LB-COMPANY-BFMT
SORT_CODE	VARCHAR	12		Sort Code	LB-COMPANY-SORTCD
ACCOUNT_NUMBER	VARCHAR	20		Account Number	LB-COMPANY-ACTNUM
ACCOUNT_NAME	VARCHAR	40		Account Name	LB-COMPANY-ACTNAM
ACCOUNT_REFERENCE	VARCHAR	20		Account Reference	LB-COMPANY-ACTREF
TAX_FORMAT	VARCHAR	2		Tax format	LB-COMPANY-TFMT
TAX_ID_NUMBER	VARCHAR	25		Tax id Number	LB-COMPANY-TAXID
PREVIOUS_TAX_ID_NUMBER	VARCHAR	25		Previous tax id no	LB-COMPANY-PREVID
MEMBER_STATE_ID	VARCHAR	2		Member state id	LB-COMPANY-MEMID
ADDRESS_LINE_1	VARCHAR	35		Address	LB-COMPANY-ADDR1
ADDRESS_LINE_2	VARCHAR	35		Address	LB-COMPANY-ADDR2
ADDRESS_LINE_3	VARCHAR	35		Address	LB-COMPANY-ADDR3
ADDRESS_LINE_4	VARCHAR	35		Address	LB-COMPANY-ADDR4
TOWN	VARCHAR	35		Post town	LB-COMPANY-TOWN
COUNTY	VARCHAR	35		County	LB-COMPANY-COUNTY
POST_CODE	VARCHAR	10		Post code	LB-COMPANY-POSTCODE
COUNTRY	VARCHAR	35		Country	LB-COMPANY-COUNTRY
Version 6.6 and above					
COMPANY_NUMBER	VARCHAR	25		Company number	LB-COMPANY-NUM
THOUSANDS_SEPARATOR	VARCHAR	1		Thousands Separator	LB-COMPANY-THOUSANDS
Version 6.7 and above					
MULTI_LEDGER	VARCHAR	1		Multi ledgers	LB-COMPANY-MULTI-LEDGER
RESERVE_STATUS	VARCHAR	1		Reserved status	LB-COMPANY-RESERVE-STATUS
Version 7.0 and above					
REG_DATE	DATE	6		Closed reg. date	LB-COMPANY-REG-DATE
Version 7.2					
Column name	Data type	Size	Def	CODA name	IASLINK name
CREDIT_LEVEL3	VARCHAR	1		Credit to level3	LB-COMPANY-CREDIT-LV3
TURNOVER_LEVEL3	VARCHAR	1		Turnover to level3	LB-COMPANY-TRNOVR-LV3
ALLOCATION_DATE	VARCHAR	3		Alloc. date method	LB-COMPANY-ALDATE

Table 11. COMPANY index details				
Index name	No	Dup	Coll	Fields in index
COMPANY_001	1	No	A	COMPANY_NAME

Table 12. CURRENCY column details					
Column name	Data type	Size	Def	CODA name	IASLINK name
CURRENCY_CODE	VARCHAR	4		Currency code	LB-CURNCY-CODE
TITLE	VARCHAR	30		[not shown on screen]	LB-CURDET-TITLE
UNIT_SINGULAR	VARCHAR	12		Names of units/ Singular	LB-CURDET-US
UNIT_PLURAL	VARCHAR	12		Names of units/ Plural	LB-CURDET-UP
UNIT_ABBREVIATION	VARCHAR	4		Names of units/ Abbrev	LB-CURDET-UA
DECIMAL_SINGULAR	VARCHAR	12		Names of decimals/ Singular	LB-CURDET-DS
DECIMAL_PLURAL	VARCHAR	12		Names of decimals/ Plural	LB-CURDET-DP
DECIMAL_ABBREVIATION	VARCHAR	4		Names of decimals/ Abbrev	LB-CURDET-DA
ALLOW_DECIMALS	VARCHAR	1	Y	Allow decimals	LB-CURDET-DPS
DECIMAL_POINT	VARCHAR	1	“fs”	Decimal point symbol	LB-CURDET-DECPT
Version 6.5 and above					
UNIT_INVERT	VARCHAR	1	H	Home/foreign rates	LB-CURDET-INVERT
UNIT_SCALE	SMALLINT	2	1	The scaling factor	LB-CURDET-SCALE
GENDER_OF_UNITS	VARCHAR	1	N	Names of units/ Gender	LB-CURDET-GENDER-U
GENDER_OF_DECIMALS	VARCHAR	1	N	Names of decimals/ Gender	LB-CURDET-GENDER-D
Version 7.2					
Column name	Data type	Size	Def	CODA name	IASLINK name
CURRENCY_END_DATE	DATE	6	31-DEC-3000	Currency end date	LB-CURNCYD-ENDDATE

“fs” means the fullstop character (.).

Table 13. CURRENCY index details				
Index name	No	Dup	Coll	Fields in index
CURRENCY_001	1	No	A	CURRENCY_CODE

Table 14. CURRENCY_RATE column details					
Column name	Data type	Size	Def	CODA name	IASLINK name
CURRENCY_CODE	VARCHAR	4		Currency code	LB-CURNCY-CODE
CURRENCY_DATE	DATE	6		Effective date	LB-CURNCY-DATE
CURRENCY_RATE	DOUBLE	8		Exchange rate	LB-CURNCY-RATE

Table 15. CURRENCY_RATE index details				
Index name	No	Dup	Coll	Fields in index
CURRENCY_RATE_001	1	No	A	CURRENCY_CODE, CURRENCY_DATE

Table 16. DESTINATION column details					
Column name	Data type	Size	Def	CODA name	IASLINK name
Version 6.5 and above					
DESTINATION_NAME	VARCHAR	12		Destination name	LB-DESTINATION-NAME
TITLE	VARCHAR	40		Destination title	LB-DESTINATION-TITLE
DESCRIPTION	VARCHAR	18		Description	LB-DESTINATION-DESCR
DESTINATION_TYPE	VARCHAR	3		Destination type	LB-DESTINATION-TYPE
COMPANY	VARCHAR	9		Company name	LB-DESTINATION-COMPANY
ACCESS_MEDIUM	VARCHAR	3		Transfer medium	LB-DESTINATION-MEDIUM
ACCESS_METHOD	VARCHAR	3		Transfer method	LB-DESTINATION-METHOD
DECNET_NODE	VARCHAR	6		DECnet node name	LB-DESTINATION-NODE
DECNET_OBJECT	VARCHAR	12		IAS object name	LB-DESTINATION-OBJECT
OUR_NAME	VARCHAR	12		Our destination name	LB-DESTINATION-OUR-NAME
FORWARD	VARCHAR	12		Forward destination	LB-DESTINATION-FORWARD
HIS_NOMINAL	VARCHAR	12		Destination accounts: Nominal	LB-DESTINATION-HIS-NOM
HIS_SUBACCOUNT	VARCHAR	12		Destination accounts: Sub-account	LB-DESTINATION-HIS-SUB
HIS_LEVEL3	VARCHAR	12		Destination accounts: Level3	LB-DESTINATION-HIS-LV3
MY_NOMINAL	VARCHAR	12		Own accounts: Nominal	LB-DESTINATION-MY-NOM
MY_SUBACCOUNT	VARCHAR	12		Own accounts: Sub- account	LB-DESTINATION-MY-SUB
MY_LEVEL3	VARCHAR	12		Own accounts: Level3	LB-DESTINATION-MY-LV3
CURRENCY	VARCHAR	4		Transfer currency	LB-DESTINATION-CURNCY
DOCUMENT_TYPE	VARCHAR	4		Document type	LB-DESTINATION-DOCTYPE
CURRENCY_CODE	VARCHAR	1		Document currency	LB-DESTINATION-DOCCURR
VALIDATION	VARCHAR	1	N	A/C validat'n method	LB-DESTINATION-VALID
RULE	VARCHAR	12		Structure name	LB-DESTINATION-RULE
FAILURE	VARCHAR	1	E	Failure action	LB-DESTINATION-FAIL
SECURITY_LEVEL	VARCHAR	1	3	Security level	LB-DESTINATION-SECLVL
Version 6.6 and above					
HIS_COMMENT	VARCHAR	1	N	Destination accounts: Comment on I/C acc.	LB-DESTINATION-HIS-COMM
MY_COMMENT	VARCHAR	1	N	Own accounts: Comment on I/C acc.	LB-DESTINATION-MY-COMM

Table 17. DESTINATION index details				
Index name	No	Dup	Coll	Fields in index
DESTINATION_001	1	No	A	DESTINATION_NAME

Table 18. DETAILS column details

Column name	Data type	Size	Def	CODA name	IASLINK name
DOCUMENT_TYPE	VARCHAR	4		Document type	LB-DET-DOC-TYPE
DOCUMENT_NUMBER	INTEGER	4		Document number	LB-DET-DOC-NUM
LINE_NUMBER	SMALLINT	2		Line number	LB-DET-LINE
DOCUMENT_DATE	DATE	6		Document date	LB-DET-DOC-DATE
DOCUMENT_CURRENCY	VARCHAR	4		Currency	LB-DET-DOC-CURR
DOCUMENT_VALUE	DOUBLE	8		Value	LB-DET-VALUE
NOMINAL	VARCHAR	12		Nominal	LB-DET-NOMINAL
SUBACCOUNT	VARCHAR	12		Subaccount	LB-DET-SUBACC
LEVEL3	VARCHAR	12		Level3	LB-DET-LEVEL3
POST_FLAG	VARCHAR	1	N	Post missing document	LB-DET-POST
DOCUMENT_FLAG	VARCHAR	1		Document control flag	LB-DET-DOC-FLAG
HOME_VALUE	DOUBLE	8		Document value	LB-DET-DOC-HOME
DOCUMENT_YEAR	SMALLINT	2		Year and period	LB-DET-YEAR
DOCUMENT_PERIOD	SMALLINT	2		Year and period	LB-DET-PERIOD
DESTINATION	VARCHAR	12		Destination	LB-DET-DESTINATION
DST_FLAG	VARCHAR	1		Destination control flag	LB-DET-DST-FLAG
DST_VALUE	DOUBLE	8		Destination foreign value	LB-DET-DST-FVAL
NOMINAL_FLAG	VARCHAR	1		Nominal control flag	LB-DET-NOM-FLAG
NOMINAL_VALUE	DOUBLE	8		Nominal foreign value	LB-DET-NOM-FVAL
SUBACCOUNT_FLAG	VARCHAR	1		Subaccount control flag	LB-DET-SUB-FLAG
SUBACCOUNT_VALUE	DOUBLE	8		Subaccount foreign value	LB-DET-SUB-FVAL
LEVEL3_FLAG	VARCHAR	1		Level3 control flag	LB-DET-LV3-FLAG
LEVEL3_VALUE	DOUBLE	8		Level3 foreign value	LB-DET-LV3-FVAL
DESCRIPTION	VARCHAR	30		Description	LB-DET-DESCR
QUANTITY_1	DOUBLE	8		Quantity 1	LB-DET-QTY1
QUANTITY_2	DOUBLE	8		Quantity 2	LB-DET-QTY2
DUE_DATE	DATE	6		Due date	LB-DET-DUE-DATE
EXTERNAL_TEXT	VARCHAR	8		External reference	LB-DET-EXT-TEXT
EXTERNAL_NUMBER	INTEGER	4		External reference	LB-DET-EXT-NUM
DOCUMENT_REFERENCE	VARCHAR	24		Document reference	LB-DET-DOC-REF
AUTHOR	VARCHAR	4		Authorising user	LB-DET-AUTHOR
IF_VALUE [note 1]	DOUBLE	8		If value	LB-DET-IF-VALUE
IF_DATE [note 2]	DATE	6		If date	LB-DET-IF-DATE
BATCH	INTEGER	4		Batch	LB-DET-BATCH
DOCUMENT_COUNT	SMALLINT	2		[field not named]	LB-DET-COUNT
DOCUMENT_STATUS	VARCHAR	1		Status	LB-DET-DOCSTS
DETAIL_STATUS	VARCHAR	1		[field not named]	LB-DET-DETSTS
RECONCILIATION_STATUS	VARCHAR	1		Reconciliation status	LB-DET-RECSTS
JR_DATE	DATE	6		Reversing Document date	LB-DET-JR-DATE
JR_YEAR	SMALLINT	2		Reversing Year	LB-DET-JR-YEAR
JR_PERIOD	SMALLINT	2		Reversing Period	LB-DET-JR-PERIOD
REFERENCE_DATE_1	DATE	6		Date 1 Reference	LB-DET-DATE-REF1
REFERENCE_DATE_2	DATE	6		Date 2 Reference	LB-DET-DATE-REF2
TAX_CODE_1	VARCHAR	4		Special Tax code	LB-DET-SPEC-TAX-CODE
TAX_VALUE_1	DOUBLE	8		Special Tax value	LB-DET-SPEC-TAX-VALUE
TAX_CODE_2	VARCHAR	4		Special Tax code	LB-DET-SPEC-TAX-CODE
TAX_VALUE_2	DOUBLE	8		Special Tax value	LB-DET-SPEC-TAX-VALUE
TAX_CODE_3	VARCHAR	4		Special Tax code	LB-DET-SPEC-TAX-CODE
TAX_VALUE_3	DOUBLE	8		Special Tax value	LB-DET-SPEC-TAX-VALUE
TAX_CODE_4	VARCHAR	4		Special Tax code	LB-DET-SPEC-TAX-CODE
TAX_VALUE_4	DOUBLE	8		Special Tax value	LB-DET-SPEC-TAX-VALUE
ACCOUNT_NAME	VARCHAR	40		Account name	LB-DET-ACTNAM
ADDRESS_1	VARCHAR	35		Address	LB-DET-ADRS
ADDRESS_2	VARCHAR	35		[Address]	LB-DET-ADRS
ADDRESS_3	VARCHAR	35		[Address]	LB-DET-ADRS
ADDRESS_4	VARCHAR	35		[Address]	LB-DET-ADRS
ADDRESS_5	VARCHAR	35		[Address]	LB-DET-ADRS
ADDRESS_6	VARCHAR	35		[Address]	LB-DET-ADRS
POST_CODE	VARCHAR	10		Code	LB-DET-CODE
COUNTRY	VARCHAR	35		Country	LB-DET-COUNTRY
PAY_APPLY	VARCHAR	1		Pay Apply	LB-DET-PAY-APPLY
PAY_BANK_INDEX	VARCHAR	1		Pay Bank Index	LB-DET-PAY-BANK-IDX
PAY_ADDRESS_INDEX	VARCHAR	1		Pay Address Index	LB-DET-PAY-ADRS-IDX
PAY_BANK_ACCOUNT	VARCHAR	12		Pay Bank Account	LB-DET-PAY-BANK-ACC
PAY_MEDIA	VARCHAR	12		Pay Media	LB-DET-PAY-MEDIA
PAY_DATE	DATE	6		Pay Date	LB-DET-PAY-DATE

Table continued on next page

DETAILS column details (continued)					
Column name	Data type	Size	Def	CODA name	IASLINK name
PAY_REFERENCE	INTEGER	4		Pay Reference	LB-DET-PAY-REF
INPUT_DATE	DATE	6		Entered on	LB-RDET-INP-DATE
INPUT_USER	VARCHAR	20		by	LB-RDET-USER
PAYMENT_NUMBER [note 3]	SMALLINT	2		Payment number	LB-RDET-PAY-NUM
PAYMENT_DATE [note 3]	DATE	4		Date paid	LB-RDET-DATE-PAID
RS_NUMBER [note 3]	SMALLINT	2		Remittance/statement number	-

Note 1. Users call this column DISCOUNT_VALUE.

Note 2. Users call this column DISCOUNT_DATE.

Note 3. This column is read-only.

The structure of the DETAILS table is identical to that of INTRAY. It is used to post a document directly to the books. On the other hand, INTRAY uses the CODA Intray facility - values are entered into the system, but the document does not need to balance, and so can be partially completed. Furthermore, INTRAY can be updated, whereas DETAILS cannot.

Table 19. DETAILS index details				
Index name	No	Dup	Coll	Fields in index
DETAILS_001	1	No	A	DOCUMENT_TYPE, DOCUMENT_NUMBER, LINE_NUMBER
DETAILS_002	2	Yes	A	NOMINAL, DOCUMENT_DATE
DETAILS_003	3	Yes	A	SUBACCOUNT, DOCUMENT_DATE
DETAILS_004	4	Yes	A	LEVEL3, DOCUMENT_DATE
DETAILS_005	5	Yes	A	NOMINAL, DOCUMENT_YEAR, DOCUMENT_PERIOD
DETAILS_006	6	Yes	A	SUBACCOUNT, DOCUMENT_YEAR, DOCUMENT_PERIOD
DETAILS_007	7	Yes	A	LEVEL3, DOCUMENT_YEAR, DOCUMENT_PERIOD
DETAILS_008	8	Yes	A	DOCUMENT_YEAR, DOCUMENT_PERIOD
DETAILS_009	9	Yes	A	INPUT_DATE

Table 20. DETAILS_COMMENT column details					
Column name	Data type	Size	Def	CODA name	IASLINK name
DOCUMENT_TYPE	VARCHAR	4		Document type	LB-COMMENT-DOC-TYPE
DOCUMENT_NUMBER	INTEGER	4		Document number	LB-COMMENT-DOC-NUM
DOCUMENT_LINE	SMALLINT	2		Line number	LB-COMMENT-DOC-LINE
COMMENT_TEXT	VARCHAR	78		[field not named]	LB-COMMENT-TEXT
COMMENT_NUMBER	INTEGER	4		-	-

Table 21. DETAILS_COMMENT index details				
Index name	No	Dup	Coll	Fields in index
DETAILS_COMMENT_001	1	No	A	DOCUMENT_TYPE, DOCUMENT_NUMBER, DOCUMENT_LINE, COMMENT_NUMBER

Table 22. DOCUMENT_MASTER column details					
Column name	Data type	Size	Def	CODA name	IASLINK name
DOCUMENT_TYPE	VARCHAR	4		Document type	LB-DOCM-TYPE
FORMAT	VARCHAR	2		Document format	LB-DOCM-FMT
MNEMONIC	VARCHAR	3		Document mnemonic	LB-DOCM-MNEM
BANK_RECONCILIATION_CODE_1	VARCHAR	4		Bank type code	LB-DOCM-BANK1
BANK_RECONCILIATION_CODE_2	VARCHAR	4		Bank type code	LB-DOCM-BANK2
DESCRIPTION	VARCHAR	30		Description	LB-DOCM-DESCR
DUE_DATE_REQUIRED	VARCHAR	1	N	Due date required	LB-DOCM-DUEDATE
DEFAULT_STATUS	VARCHAR	1		Default posting sts	LB-DOCM-DEFSTS
TURNOVER_REQUIRED	VARCHAR	1	N	Adds to turnover	LB-DOCM-TRNOVR
POST_TO_INTRAY	VARCHAR	1	N	Must place on intray	LB-DOCM-INTRAY
INTER_COMPANY_ALLOWED	VARCHAR	1	N	Inter-company doc	LB-DOCM-INTERCOMP
POST_TO_DETAILS	VARCHAR	1	N	Autopost to details	LB-DOCM-AUTO
CHANGE_DOCUMENT_TYPE	VARCHAR	1	N	Change doc type	LB-DOCM-CHANGE
CHECK_EXTERNAL_REFERENCES	VARCHAR	1	*	Duplicate ext ref	LB-DOCM-CHK-EXT
DUPLICATE_EXTERNAL_REFERENCE	VARCHAR	1	N	Duplicate ext ref	LB-DOCM-DUPL-EXT
REQUIRE_AUTHORISING_USER	VARCHAR	1	N	Requires auth. user	LB-DOCM-AUTHOR
SIGN_RULE	VARCHAR	1	N	Sign rule	LB-DOCM-SIGN
SEQUENCING_ACTION	VARCHAR	1	U	Sequence rule	LB-DOCM-SEQ
START_DOCUMENT_NUMBER_1	INTEGER	4		Sequence start	LB-DOCM-START
START_DOCUMENT_NUMBER_2	INTEGER	4		Sequence start	LB-DOCM-START
START_DOCUMENT_NUMBER_3	INTEGER	4		Sequence start	LB-DOCM-START
START_DOCUMENT_NUMBER_4	INTEGER	4		Sequence start	LB-DOCM-START
START_DOCUMENT_NUMBER_5	INTEGER	4		Sequence start	LB-DOCM-START
START_DOCUMENT_NUMBER_6	INTEGER	4		Sequence start	LB-DOCM-START
END_DOCUMENT_NUMBER_1	INTEGER	4		Sequence end	LB-DOCM-END
END_DOCUMENT_NUMBER_2	INTEGER	4		Sequence end	LB-DOCM-END
END_DOCUMENT_NUMBER_3	INTEGER	4		Sequence end	LB-DOCM-END
END_DOCUMENT_NUMBER_4	INTEGER	4		Sequence end	LB-DOCM-END
END_DOCUMENT_NUMBER_5	INTEGER	4		Sequence end	LB-DOCM-END
END_DOCUMENT_NUMBER_6	INTEGER	4		Sequence end	LB-DOCM-END
NUMBER_RANGE_OPEN	VARCHAR	6		Open, in use	LB-DOCM-OPEN
SEQUENCE_IN_USE	VARCHAR	6	NNNNN	Open, in use	LB-DOCM-SEQ-INUSE
CURRENCY	VARCHAR	4		Document currency	LB-DOCM-CURRENCY
CHANGE_CURRENCY	VARCHAR	1	Y	Change currency	LB-DOCM-CURR-CHNG
GENERATE_TAX_LINES	VARCHAR	1	N	Generates tax lines	LB-DOCM-TAX
CHECK_INTRAY_DOC_BALANCES	VARCHAR	1	N	Balance intray docs	LB-DOCM-INTRAY-BAL
SELF_PROPORTIONING	VARCHAR	1	N	Self proportioning	LB-DOCM-PROPN
REVERSE_QUANTITY_SIGNS	VARCHAR	1	Y	Reverse quantities	LB-DOCM-REV-QTY
DEFAULT_NOMINAL	VARCHAR	12		Account	LB-DOCM-NOM
DEFAULT_SUBACCOUNT	VARCHAR	12		Account	LB-DOCM-SUB
DEFAULT_LEVEL3	VARCHAR	12		Account	LB-DOCM-LV3
PRESERVE_PERIOD	VARCHAR	1	N	Preserve I/C period	LB-DOCM-PRSV-PRD
PRESERVE_EXTERNAL_REFERENCES	VARCHAR	1	Y	Preserve I/C ext. ref	LB-DOCM-PRSV-EXT
PRESERVE_BATCH_NUMBER	VARCHAR	1	N	Retain batch number	LB-DOCM-PRSV-BAT
PRESERVE_SOURCE_EXT_REF	VARCHAR	1	Y	Preserve i/c ext. ref	LB-DOCM-PRSV-SRC
DEFAULT_EXTERNAL_TEXT	VARCHAR	8		Ext. ref	LB-DOCM-EXT-TXT
DEFAULT_EXTERNAL_NUMBER	INTEGER	4		Ext. ref	LB-DOCM-EXT-NUM
DOCUMENT_REFERENCE	VARCHAR	24		Doc. ref	LB-DOCM-DOC-REF
CHECK_EXTERNAL_DOCUMENT	VARCHAR	1	E	Ext / doc ref check	LB-DOCM-EXT-DOC
CHECK_DOCUMENT	VARCHAR	1		Duplicate doc ref	LB-DOCM-CHK-DOC
DUPLICATE_DOCUMENT	VARCHAR	1	N	Duplicate doc ref	LB-DOCM-DUPL-DOC
ACCOUNT_MASKS	VARCHAR	1	N	Use account masks	LB-DOCM-USEMSK
ACCOUNT_EXTERNAL	VARCHAR	1	Y	Use references	LB-DOCM-USEEXT
USE_DOCUMENT_REFERENCE	VARCHAR	1	Y	Use references	LB-DOCM-USEDOC
USE_DATE_REFERENCE_1	VARCHAR	1	N	Use references	LB-DOCM-USEDT1
USE_DATE_REFERENCE_2	VARCHAR	1	N	Use references	LB-DOCM-USEDT2
COMPANY_MASK	VARCHAR	12		Company mask	LB-DOCM-CMP-MASK
NOMINAL_MASK	VARCHAR	12		Account masks	LB-DOCM-NOM-MASK
SUBACCOUNT_MASK	VARCHAR	12		Account masks	LB-DOCM-SUB-MASK
LEVEL3_MASK	VARCHAR	12		Account masks	LB-DOCM-LV3-MASK
DATE_REFERENCE_1	VARCHAR	11		Date 1 ref	LB-DOCM-DATE1
DATE_REFERENCE_2	VARCHAR	11		Date 2 ref	LB-DOCM-DATE2
SPECIAL_TAX	VARCHAR	1	N	Enter special tax	LB-DOCM-SPEC-TAX

Table continued on next page

DOCUMENT_MASTER column details (continued)					
Column name	Data type	Size	Def	CODA name	IASLINK name
Version 6.6 and above					
CLEAR_CURRENCY	VARCHAR	1	N	Reprompt currency	LB-DOCM-CLEAR-CUR
Version 6.7 and above					
PROPAGATE_PAY	VARCHAR	1	N	Propagate pay number	LB-DOCM-PROP-PAY
Version 7.2					
GROSS_EXPENSE	VARCHAR	1	N	Gross Expense	LB-DOCM-GROSS-NETT

Table 23. DOCUMENT_MASTER index details				
Index name	No	Dup	Coll	Fields in index
DOCUMENT_001	1	No	A	DOCUMENT_TYPE

Table 24. GROUP_HIERARCHY column details					
Column name	Data type	Size	Def	CODA name	IASLINK name
PARENT	VARCHAR	12		Parents	LB-GRPXREF-PARENT
CHILD	VARCHAR	12		Children	LB-GRPXREF-CHILD
CHILD_TYPE [note 1]	VARCHAR	1		-	LB-GRPXREF-CHILD-TYPE
PARENT_ID	INTEGER	4		-	-
CHILD_ID	INTEGER	4		-	-

See also ACCOUNT_GROUP table.

Note 1: CHILD_TYPE is a flag which may refer to an account group, a nominal, a subaccount or a level 3. For any given account which is to be included in more than one account group, the GROUP_HIERARCHY table should be used; there should be one record for each account in the account group (PARENT), and the CHILD should be the account that is in the group.

Table 25. GROUP_HIERARCHY index details				
Index name	No	Dup	Coll	Fields in index
GROUP_HIERARCHY_001	1	No	A	PARENT_ID, CHILD_ID, CHILD_TYPE
GROUP_HIERARCHY_002	2	Yes	A	PARENT

Table 26. INTRAY column details					
Column name	Data type	Size	Def	CODA name	IASLINK name
DOCUMENT_TYPE	VARCHAR	4		Document type	LB-DET-DOC-TYPE
DOCUMENT_NUMBER	INTEGER	4		Initial seq number	LB-DET-DOC-NUM
LINE_NUMBER [note 1]	SMALLINT	2		Line number	LB-DET-LINE
DOCUMENT_DATE	DATE	6		Document date	LB-DET-DOC-DATE
DOCUMENT_CURRENCY	VARCHAR	4		Currency	LB-DET-DOC-CURR
DOCUMENT_VALUE	DOUBLE	8		Value	LB-DET-VALUE
NOMINAL	VARCHAR	12		Nominal	LB-DET-NOMINAL
SUBACCOUNT	VARCHAR	12		Subaccount	LB-DET-SUBACC
LEVEL3	VARCHAR	12		Level3	LB-DET-LEVEL3
POST_FLAG	VARCHAR	1	N	Post missing document	LB-DET-POST
DOCUMENT_FLAG	VARCHAR	1		Document control flag	LB-DET-DOC-FLAG
HOME_VALUE	DOUBLE	8		Document value	LB-DET-DOC-HOME
DOCUMENT_YEAR	SMALLINT	2		Year and period	LB-DET-YEAR
DOCUMENT_PERIOD	SMALLINT	2		Year and period	LB-DET-PERIOD
DESTINATION	VARCHAR	12		Destination	LB-DET-DESTINATION
DST_FLAG	VARCHAR	1		Destination control flag	LB-DET-DST-FLAG
DST_VALUE	DOUBLE	8		Destination foreign value	LB-DET-DST-FVAL
NOMINAL_FLAG	VARCHAR	1		Nominal control flag	LB-DET-NOM-FLAG
NOMINAL_VALUE	DOUBLE	8		Nominal foreign value	LB-DET-NOM-FVAL
SUBACCOUNT_FLAG	VARCHAR	1		Subaccount control flag	LB-DET-SUB-FLAG
SUBACCOUNT_VALUE	DOUBLE	8		Subaccount foreign value	LB-DET-SUB-FVAL
LEVEL3_FLAG	VARCHAR	1		Level3 control flag	LB-DET-LV3-FLAG
LEVEL3_VALUE	DOUBLE	8		Level3 foreign value	LB-DET-LV3-FVAL
DESCRIPTION	VARCHAR	30		Description	LB-DET-DESCR
QUANTITY_1	DOUBLE	8		Quantity 1	LB-DET-QTY1
QUANTITY_2	DOUBLE	8		Quantity 2	LB-DET-QTY2
DUE_DATE	DATE	6		Due date	LB-DET-DUE-DATE
EXTERNAL_TEXT	VARCHAR	8		External reference	LB-DET-EXT-TEXT
EXTERNAL_NUMBER	INTEGER	4		External reference	LB-DET-EXT-NUM
DOCUMENT_REFERENCE	VARCHAR	24		Document reference	LB-DET-DOC-REF
AUTHOR	VARCHAR	4		Author	LB-DET-AUTHOR
IF_VALUE [note 2]	DOUBLE	8		Discount	LB-DET-IF-VALUE
IF_DATE [note 3]	DATE	6		If paid by	LB-DET-IF-DATE
BATCH	INTEGER	4		Batch	LB-DET-BATCH
DOCUMENT_COUNT	SMALLINT	2		[field not named]	LB-DET-COUNT
DOCUMENT_STATUS	VARCHAR	1		Status	LB-DET-DOCSTS
DETAIL_STATUS	VARCHAR	1		[field not named]	LB-DET-DETSTS
RECONCILIATION_STATUS	VARCHAR	1		Reconciliation status	LB-DET-RECSTS
JR_DATE	DATE	6		Reversing Document date	LB-DET-JR-DATE
JR_YEAR	SMALLINT	2		Reversing Year	LB-DET-JR-YEAR
JR_PERIOD	SMALLINT	2		Reversing Period	LB-DET-JR-PERIOD
REFERENCE_DATE_1	DATE	6		Date 1 Reference	LB-DET-DATE-REF1
REFERENCE_DATE_2	DATE	6		Date 2 Reference	LB-DET-DATE-REF2
TAX_CODE_1	VARCHAR	4		Special Tax code	LB-DET-SPEC-TAX-CODE
TAX_VALUE_1	DOUBLE	8		Special Tax value	LB-DET-SPEC-TAX-VALUE
TAX_CODE_2	VARCHAR	4		Special Tax code	LB-DET-SPEC-TAX-CODE
TAX_VALUE_2	DOUBLE	8		Special Tax value	LB-DET-SPEC-TAX-VALUE
TAX_CODE_3	VARCHAR	4		Special Tax code	LB-DET-SPEC-TAX-CODE
TAX_VALUE_3	DOUBLE	8		Special Tax value	LB-DET-SPEC-TAX-VALUE
TAX_CODE_4	VARCHAR	4		Special Tax code	LB-DET-SPEC-TAX-CODE
TAX_VALUE_4	DOUBLE	8		Special Tax value	LB-DET-SPEC-TAX-VALUE
ACCOUNT_NAME	VARCHAR	40		Account name	LB-DET-ACTNAM
ADDRESS_1	VARCHAR	35		Address	LB-DET-ADRS
ADDRESS_2	VARCHAR	35		[field not named]	LB-DET-ADRS
ADDRESS_3	VARCHAR	35		[field not named]	LB-DET-ADRS
ADDRESS_4	VARCHAR	35		[field not named]	LB-DET-ADRS
ADDRESS_5	VARCHAR	35		[field not named]	LB-DET-ADRS
ADDRESS_6	VARCHAR	35		[field not named]	LB-DET-ADRS
POST_CODE	VARCHAR	10		Code	LB-DET-CODE
COUNTRY	VARCHAR	35		Country	LB-DET-COUNTRY
PAY_APPLY	VARCHAR	1		Pay Apply	LB-DET-PAY-APPLY
PAY_BANK_INDEX	VARCHAR	1		Pay Bank Index	LB-DET-PAY-BANK-IDX
PAY_ADDRESS_INDEX	VARCHAR	1		Pay Address Index	LB-DET-PAY-ADRS-IDX
PAY_BANK_ACCOUNT	VARCHAR	12		Pay Bank Account	LB-DET-PAY-BANK-ACC

Table continued on next page

INTRAY column details (continued)					
Column name	Data type	Size	Def	CODA name	IASLINK name
PAY_MEDIA	VARCHAR	12		Pay Media	LB-DET-PAY-MEDIA
PAY_DATE	DATE	6		Pay Date	LB-DET-PAY-DATE
PAY_REFERENCE	INTEGER	4		Pay Reference	LB-DET-PAY-REF
INPUT_DATE	DATE	6		Entered on	LB-RDET-INP-DATE
INPUT_USER	VARCHAR	20		by	LB-RDET-USER

Note 1. The allowed values that this can take depend upon the sequencing rule (SEQUENCING_ACTION) that is specified in the DOCUMENT_MASTER table.

Note 2. Users call this column DISCOUNT_VALUE.

Note 3. Users call this column DISCOUNT_DATE.

Also see the DETAILS table.

Table 27. INTRAY index details				
Index name	No	Dup	Coll	Fields in index
INTRAY_001	1	No	A	DOCUMENT_TYPE, DOCUMENT_NUMBER, LINE_NUMBER
INTRAY_002	2	Yes	A	NOMINAL, DOCUMENT_DATE
INTRAY_003	3	Yes	A	SUBACCOUNT, DOCUMENT_DATE
INTRAY_004	4	Yes	A	LEVEL3, DOCUMENT_DATE
INTRAY_005	4	Yes	A	INPUT_DATE

Table 28. INTRAY_COMMENT column details					
Column name	Data type	Size	Def	CODA name	IASLINK name
DOCUMENT_TYPE	VARCHAR	4		Doc type	LB-COMMENT-DOC-TYPE
DOCUMENT_NUMBER	INTEGER	4		Initial Seq number	LB-COMMENT-DOC-NUM
DOCUMENT_LINE	SMALLINT	2		Line number	LB-COMMENT-DOC-LINE
COMMENT_TEXT	VARCHAR	78		[field not named]	LB-COMMENT-TEXT
COMMENT_NUMBER	INTEGER	4		-	-

Table 29. INTRAY_COMMENT index details				
Index name	No	Dup	Coll	Fields in index
INTRAY_COMMENT_001	1	No	A	DOCUMENT_TYPE, DOCUMENT_NUMBER, DOCUMENT_LINE, COMMENT_NUMBER

Table 30. LEDGER column details					
Column name	Data type	Size	Def	CODA name	IASLINK name
LEDGER	VARCHAR	5		Ledger ident.	LB-LEDGER-CODE
NOMINAL	VARCHAR	12		Control account	LB-LEDGER-NOMINAL
DESCRIPTION	VARCHAR	30		Ledger name	LB-LEDGER-TITLE
CREATED_USER	VARCHAR	20		[CODA internal]	-
CREATED_DATE	DATE	6		[CODA internal]	-
MODIFIED_USER	VARCHAR	20		[CODA internal]	-
MODIFIED_DATE	DATE	6		[CODA internal]	-

Table 31. LEDGER index details

Index name	No	Dup	Coll	Fields in index
LEDGER_001	1	No	A	LEDGER

Table 32. NAME column details

Column name	Data type	Size	Def	CODA name	IASLINK name
SUBACCOUNT	VARCHAR	12		Sub-account code	LB-SUB-CODE
LEDGER [note 1]	VARCHAR	5	0000	Subanalyse as	LB-SUB-LEDGER
ADDRESS_ID	SMALLINT	2		Address no (Address data)	LB-SUB-ADDID
TITLE	VARCHAR	40		Sub-account name	LB-SUB-NAME
SHORT_TITLE	VARCHAR	20		Short name	LB-SUB-SHORT
SECURITY_LEVEL	VARCHAR	1	3	Security level	LB-SUB-SECLVL
TAX_CODE	VARCHAR	4		Tax code	LB-SUB-TAX
EXPIRY_DATE	DATE	6		Expiry date	LB-SUB-EXPIRY
ACCOUNT_TYPE	VARCHAR	1	B	A/c type	LB-SUB-TYPE
CURRENCY_CODE	VARCHAR	4		Currency	LB-SUB-CURRENCY
RECONCILABLE	VARCHAR	1	N	Reconcilable	LB-SUB-RECON
ONE_OFF_SUPPLIERS	VARCHAR	1	N	One off supplier	LB-SUB-EXTADR
ADDRESS_FORMAT	VARCHAR	2		Address format type	LB-SUB-AFMT
BANK_FORMAT	VARCHAR	2		Format code	LB-SUB-BFMT
PRIORITY	VARCHAR	1	1	Account priority	LB-SUB-PRIORITY
SEND_TO_HIGHER	VARCHAR	1		Higher	LB-SUB-HIGHER
SEND_TO_LOWER	VARCHAR	1		Lower	LB-SUB-LOWER
DEFAULT_POSTING_STATUS	VARCHAR	1	A	Default posting sts	LB-SUB-DFLT
CROSS_REFERENCE_ACCOUNT	VARCHAR	12		Cross reference	LB-SUB-XREF
DRAFT	VARCHAR	1	N	Draft	LB-SUB-DRAFT
CHEQUE	VARCHAR	1	Y	Cheque	LB-SUB-CHEQUE
BACS	VARCHAR	1	N	Bacs	LB-SUB-BACS
CASH	VARCHAR	1	Y	Cash	LB-SUB-CASH
SORT_CODE	VARCHAR	12		Sort code	LB-SUB-SRT-CODE
ACCOUNT_NUMBER	VARCHAR	20		Bank account number	LB-SUB-ACT-NUM
ACCOUNT_NAME	VARCHAR	40		Account name	LB-SUB-ACT-NAM
ACCOUNT_REFERENCE	VARCHAR	20		Account reference	LB-SUB-ACT-REF
CREDIT_LIMIT	DOUBLE	8		Credit lim	LB-SUB-CLIM
TRADING_LIMIT	DOUBLE	8		Trade lim	LB-SUB-TLIM
INDIRECT_ACCOUNT	VARCHAR	12		Indirect	LB-SUB-INDIRECT
MNEMONIC_1	VARCHAR	12		Account mnemonics	LB-SUB-MNEM
MNEMONIC_2	VARCHAR	12		Account mnemonics	LB-SUB-MNEM
MNEMONIC_3	VARCHAR	12		Account mnemonics	LB-SUB-MNEM
MNEMONIC_4	VARCHAR	12		Account mnemonics	LB-SUB-MNEM
MNEMONIC_5	VARCHAR	12		Account mnemonics	LB-SUB-MNEM
MNEMONIC_6	VARCHAR	12		Account mnemonics	LB-SUB-MNEM
ACCOUNT_GROUP [note 2]	VARCHAR	12		Account groups	LB-SUB-GROUP
LEVEL3_ACCOUNT_GROUP	VARCHAR	12		Group as level3	LB-SUB-GROUP-LV3
ONE_CHEQUE_PER_TRANSACTION	VARCHAR	1	N	Single cheque / trans	LB-SUB-SINGCHQ
USE_SPECIAL_TAX	VARCHAR	1	N	Supply special tax	LB-SUB-TAX-FLAG
TAX_SORT_CODE	VARCHAR	12		Sort code	LB-SUB-TAX-SORT
TAX_ACCOUNT_NUMBER	VARCHAR	20		Account Number	LB-SUB-TAX-ACTNUM
TAX_ACCOUNT_NAME	VARCHAR	40		Account Name	LB-SUB-TAX-ACTNAM
TAX_ACCOUNT_REFERENCE	VARCHAR	20		Account Reference	LB-SUB-TAX-ACTREF
TAX_FORMAT	VARCHAR	2		General tax type	LB-SUB-TFMT
TYPE_OF_TAX_ID_NUMBER	VARCHAR	1		Type of TIN	LB-SUB-GENTAX-TYPE
TAX_ID_NUMBER	VARCHAR	25	00000 0000	Tax id number	LB-SUB-GENTAX-TIN

Table continued on next page

NAME column details (continued)					
Column name	Data type	Size	Def	CODA name	IASLINK name
Version 6.5 and above					
FOR_THE_ATTENTION_OF	VARCHAR	40		For atn of	LB-SUB-FAO
ADDRESS_LINE_1	VARCHAR	35		Address	LB-SUB-ADDR
ADDRESS_LINE_2	VARCHAR	35		Address	LB-SUB-ADDR
ADDRESS_LINE_3	VARCHAR	35		Address	LB-SUB-ADDR
ADDRESS_LINE_4	VARCHAR	35		Address	LB-SUB-ADDR
TOWN	VARCHAR	35		Town	LB-SUB-TOWN
COUNTY	VARCHAR	35		County	LB-SUB-COUNTY
POST_CODE	VARCHAR	10		Post code	LB-SUB-POST
COUNTRY	VARCHAR	35		Country	LB-SUB-COUNTRY
STATEMENT_HEADER	VARCHAR	30		Stmt header	LB-SUB-HEADER
TELEPHONE_NUMBER	VARCHAR	20		Telephone	LB-SUB-PHONE
FAX_NUMBER	VARCHAR	20		Fax	LB-SUB-TELEX
ACCOUNT_LANGUAGE	VARCHAR	2		Account language	LB-SUB-ACC-LANG
Version 6.6 and above					
COMPANY_NUMBER	VARCHAR	25		Company number	LB-SUB-COMPANY-NUM
Version 6.7 and above					
PREFERRED_MEDIA	VARCHAR	12		Pref media	LB-SUB-MEDIA
BANK_ID	SMALLINT	2		Address no (Bank Address data)	LB-SUB-BADDID
BANK_ADDRESS_LINE_1	VARCHAR	35		Bank address	LB-SUB-BADDR
BANK_ADDRESS_LINE_2	VARCHAR	35		Bank address	LB-SUB-BADDR
BANK_ADDRESS_LINE_3	VARCHAR	35		Bank address	LB-SUB-BADDR
BANK_ADDRESS_LINE_4	VARCHAR	35		Bank address	LB-SUB-BADDR
BANK_TOWN	VARCHAR	35		Town	LB-SUB-BTOWN
BANK_COUNTY	VARCHAR	35		County	LB-SUB-BCOUNTY
BANK_POST_CODE	VARCHAR	10		Post code	LB-SUB-BPOST
BANK_COUNTRY	VARCHAR	35		Country	LB-SUB-BCOUNTRY
CREATED_USER	VARCHAR	20		[CODA internal]	-
CREATED_DATE	DATE	6		[CODA internal]	-
MODIFIED_USER	VARCHAR	20		[CODA internal]	LB-SUB-MUSER
MODIFIED_DATE	DATE	6		[CODA internal]	LB-SUB-MDATE
Version 7.2					
INTEREST_RATE	DOUBLE	8			LB-SUB-INTCALC-PCENT

Note 1. The default for the ledger column is the space character followed by four zeros.

Note 2. If this account is associated with more than one account group, this column will contain the value “** many **”. If you want to insert account group data, and if there is more than one account group for the account, then you must use the GROUP_HIERARCHY table to insert the data.

Table 33. NAME index details				
Index name	No	Dup	Coll	Fields in index
NAME_001	1	No	A	SUBACCOUNT
NAME_002	2	Yes	A	LEDGER

Table 34. NAME_COMMENT column details					
Column name	Data type	Size	Def	CODA name	IASLINK name
SUBACCOUNT	VARCHAR	12		Sub-account code	LB-COMM-CODE
COMMENT	VARCHAR	78		Account comments	LB-COMM-TEXT
COMMENT_NUMBER	INTEGER	4		[field not named]	-

Table 35. NAME_COMMENT index details

Index name	No	Dup	Coll	Fields in index
NAME_COMMENT_001	1	No	A	SUBACCOUNT, COMMENT_NUMBER

Table 36. NAME_CONTACT column details

Column name	Data type	Size	Def	CODA name	IASLINK name
SUBACCOUNT	VARCHAR	12		Sub-account code	LB-COMM-CODE
COMMENT	VARCHAR	78		Account comments	LB-COMM-TEXT
COMMENT_NUMBER	INTEGER	4		[field not named]	-

Table 37. NAME_CONTACT index details

Index name	No	Dup	Coll	Fields in index
NAME_CONTACT_001	1	No	A	SUBACCOUNT, COMMENT_NUMBER

Table 38. NAME_LEDGER_COMMENT column details

Column name	Data type	Size	Def	CODA name	IASLINK name
SUBACCOUNT	VARCHAR	12		Sub-account code	LB-COMM-CODE
COMMENT	VARCHAR	78		Account comments	LB-COMM-TEXT
COMMENT_NUMBER	INTEGER	4		[field not named]	-

Table 39. NAME_LEDGER_COMMENT index details

Index name	No	Dup	Coll	Fields in index
NAME_LEDGER_COMMENT_001	1	No	A	SUBACCOUNT, COMMENT_NUMBER

Table 40. NAME_OWNER column details

Column name	Data type	Size	Def	CODA name	IASLINK name
SUBACCOUNT	VARCHAR	12		Sub-account code	LB-OWN-CODE
NOT_OR_ONLY	VARCHAR	1		Not (Nominal/ledger list)	LB-OWN-NOT
OWNER_TYPE	VARCHAR	1		Type (Nominal/ledger list)	LB-OWN-TYPE
OWNER_FROM	VARCHAR	12		From (Nominal/ledger list)	LB-OWN-FROM
OWNER_TO	VARCHAR	12		To (Nominal/ledger list)	LB-OWN-TO
LAST_YEAR_TURNOVER_HOME	DOUBLE	8		Turnover Last year	LB-OWN-LAST-HOME
PREVIOUS_YEAR_TURNOVER_HOME	DOUBLE	8		Turnover Previous year	LB-OWN-PREV-HOME
LAST_YEAR_TURNOVER_ACCOUNT	DOUBLE	8		Turnover Last year	LB-OWN-LAST-ACNT
PREVIOUS_YEAR_TURNOVER_ACCOUNT	DOUBLE	8		Turnover Previous year	LB-OWN-PREV-ACNT
TERMS_CODE	VARCHAR	1	R	Terms code	LB-OWN-TERMS
PAYMENT_DISCOUNT_METHOD	VARCHAR	1	N	Discount	LB-OWN-DISC-TYPE
EARLY_TERMS_DATE	VARCHAR	4		Early terms	LB-OWN-EDATE
EARLY_TERMS_DISCOUNT	INTEGER	4		Early terms	LB-OWN-EDISC
NORMAL_TERMS_DATE	VARCHAR	4	0100	Normal terms	LB-OWN-NDATE
NORMAL_TERMS_DISCOUNT	INTEGER	4		Normal terms	LB-OWN-NDISC
LIST_NUMBER	INTEGER	4		[field not named]	-

Table 41. NAME_OWNER index details				
Index name	No	Dup	Coll	Fields in index
NAME_OWNER_001	1	No	A	SUBACCOUNT, LIST_NUMBER

Table 42. NOMINAL column details					
Column name	Data type	Size	Def	CODA name	IASLINK name
NOMINAL	VARCHAR	12		Identification code	LB-NOM-CODE
LEDGER [note 1]	VARCHAR	5	0000	Ledger code	LB-NOM-LEDGER
FULL_NAME	VARCHAR	40		Nominal name	LB-NOM-NAME
SHORT_NAME	VARCHAR	20		Short name	LB-NOM-SHORT
SECURITY_LEVEL	VARCHAR	1	3	Security level	LB-NOM-SECLVL
TAX_CODE	VARCHAR	4		Tax code	LB-NOM-TAX
EXPIRY_DATE	DATE	6		Expiry date	LB-NOM-EXPIRY
ACCOUNT_TYPE	VARCHAR	1	T	Account type	LB-NOM-TYPE
CURRENCY	VARCHAR	4		Currency	LB-NOM-CURRENCY
DEFAULT_STATUS	VARCHAR	1	X	Default posting sts	LB-NOM-DEFSTS
POST_IF_ZERO	VARCHAR	1	N	Post details if zero	LB-NOM-PDIZ
DESCRIPTION_REQUIRED	VARCHAR	1	N	Description required	LB-NOM-DESCR
RECONCILABLE	VARCHAR	1	N	Reconcilable	LB-NOM-RECON
CHECK_CREDIT_LIMIT	VARCHAR	1	N	Check credit limits	LB-NOM-CHKCLIM
CHECK_TRADING_LIMIT	VARCHAR	1	N	Check credit limits	LB-NOM-CHKTLIM
QUANTITY_1_TITLE	VARCHAR	20		Quantity titles	LB-NOM-Q1TITL
QUANTITY_1_REQUIRED	VARCHAR	1	N	Quantity required	LB-NOM-Q1REQD
QUANTITY_1_BUDGET_CODE	VARCHAR	1		Quantity budg codes	LB-NOM-Q1BUDG
QUANTITY_2_TITLE	VARCHAR	20		Quantity titles	LB-NOM-Q2TITL
QUANTITY_2_REQUIRED	VARCHAR	1	N	Quantity required	LB-NOM-Q2REQD
QUANTITY_2_BUDGET_CODE	VARCHAR	1		Quantity budg codes	LB-NOM-Q2BUDG
ACCOUNT_GROUP [note 2]	VARCHAR	12		Account groups	LB-NOM-GROUP
CREATED_USER	VARCHAR	20		[CODA internal]	-
CREATED_DATE	DATE	6		[CODA internal]	-
MODIFIED_USER	VARCHAR	20		[CODA internal]	LB-NOM-MUSER
MODIFIED_DATE	DATE	6		[CODA internal]	LB-NOM-MDATE

Note 1. The default for the ledger column is the space character followed by four zeros.

Note 2: If this account is associated with more than one account group, this column will contain the value "*** many ***". If you want to insert account group data, and if there is more than one account group for the account, then you must use the GROUP_HIERARCHY table to insert the data.

Table 43. NOMINAL index details				
Index name	No	Dup	Coll	Fields in index
NOMINAL_001	1	No	A	NOMINAL
NOMINAL_002	2	Yes	A	LEDGER

Table 44. NOMINAL_COMMENT column details					
Column name	Data type	Size	Def	CODA name	IASLINK name
NOMINAL	VARCHAR	12		Identification code	LB-COMM-CODE
COMMENT	VARCHAR	78		Account comments	LB-COMM-TEXT
COMMENT_NUMBER	INTEGER	4		[field not named]	-

Table 45. NOMINAL_COMMENT index details				
Index name	No	Dup	Coll	Fields in index
NOMINAL_COMMENT_001	1	No	A	NOMINAL, COMMENT_NUMBER

Column name	Data type	Size	Def	CODA name	IASLINK name
SUBACCOUNT	VARCHAR	12		Sub-account code	LB-SUB-CODE
TITLE	VARCHAR	40		Title	LN-SUB-NAME
LEDGER	VARCHAR	5		Subanalyse as	LB-SUB-LEDGER
SUBACCOUNT_TYPE	VARCHAR	1		[CODA internal]	
SHORT_TITLE	VARCHAR	20		Short name	LB-SUB-SHORT
SECURITY_LEVEL	VARCHAR	1	3	Security level	LB-SUB-SECLVL
TAX_CODE	VARCHAR	4		Tax code	LB-SUB-TAX
EXPIRY_DATE	DATE	6		Expiry date	LB-SUB-EXPIRY
ACCOUNT_TYPE	VARCHAR	1	T	A/c type	LB-SUB-TYPE
CURRENCY_CODE	VARCHAR	4		Currency	LB-SUB-CURNCY
RECONCILABLE	VARCHAR	1	N	Reconcilable	LB-SUB-RECON
ONE_OFF_SUPPLIERS	VARCHAR	1	N	One off Suppliers	LB-SUB-EXTADR
ACCOUNT_GROUP [note 1]	VARCHAR	12		Account groups	LB-SUB-GROUP
LEVEL3_ACCOUNT_GROUP	VARCHAR	12		Account groups Lvl3	LB-SUB-GROUP-LV3
CREATED_USER	VARCHAR	20		[CODA internal]	-
CREATED_DATE	DATE	6		[CODA internal]	-
MODIFIED_USER	VARCHAR	20		[CODA internal]	-
MODIFIED_DATE	DATE	6		[CODA internal]	-

Note 1: If this account is associated with more than one account group, this column will contain the value "*** many ***". If you want to insert account group data, and if there is more than one account group for the account, then you must use the GROUP_HIERARCHY table to insert the data.

Index name	No	Dup	Coll	Fields in index
SUBACCOUNT_001	1	No	A	SUBACCOUNT
SUBACCOUNT_002	2	Yes	A	LEDGER

Column name	Data type	Size	Def	CODA name	IASLINK name
SUBACCOUNT	VARCHAR	12		Sub-account code	LB-COMM-CODE
COMMENT	VARCHAR	78		Account comments	LB-COMM-TEXT
COMMENT_NUMBER	INTEGER	4		[field not named]	-

Index name	No	Dup	Coll	Fields in index
SUBACCOUNT_COMMENT_001	1	No	A	SUBACCOUNT, COMMENT_NUMBER

Column name	Data type	Size	Def	CODA name	IASLINK name
SUBACCOUNT	VARCHAR	12		Sub-account code	LB-OWN-CODE
LIST_NUMBER	INTEGER	4		[field not named]	-
NOT_OR_ONLY	VARCHAR	1		Not (Nominal/ledger list)	LB-OWN-NOT
OWNER_TYPE	VARCHAR	1		Type (Nominal/ledger list)	LB-OWN-TYPE
OWNER_FROM	VARCHAR	12		From (Nominal/ledger list)	LB-OWN-FROM
OWNER_TO	VARCHAR	12		To (Nominal/ledger list)	LB-OWN-TO

Table 51. SUBACCOUNT_OWNER index details

Index name	No	Dup	Coll	Fields in index
SUBACCOUNT_OWNER_001	1	No	A	SUBACCOUNT, LIST_NUMBER

Table 52. TAX_CODE column details**a. CODA versions 6.4 to 7.1 only**

Column name	Data type	Size	Def	CODA name	IASLINK name
TAX_CODE	VARCHAR	4		Identification code	LB-TAXCODE-CODE
TITLE	VARCHAR	30		Tax name	LB-TAXCODE-TITLE
TAXRATE	DOUBLE	8		Tax rate	LB-TAXCODE-RATE
NOMINAL	VARCHAR	12		Associated account	LB-TAXCODE-NOM
SUBACCOUNT	VARCHAR	12		Associated account	LB-TAXCODE-SUB
LEVEL3	VARCHAR	12		Associated account	LB-TAXCODE-LV3
SECURITY_LEVEL	VARCHAR	1		Security level	LB-TAXCODE-SECLVL

Table continued on next page

TAX_CODE column details (continued)					
Column name	Data type	Size	Def	CODA name	IASLINK name
TAX_TYPE	VARCHAR	1	N	Tax type	LB-TAXCODE-TYPE
TAX_CODE	VARCHAR	4		Identification code	LB-TAXCODE-CODE
TITLE	VARCHAR	30		Tax name	LB-TAXCODE-TITLE
SECURITY_LEVEL	VARCHAR	1	3	Security level	LB-TAXCODE-SECLVL
REVERSE	VARCHAR	1	N	Reversible	LB-TAXCODE-REVERSE
NOMINAL_1	VARCHAR	12		[note 1]	LB-TAXCODE-NOM
SUBACCOUNT_1	VARCHAR	12		[note 1]	LB-TAXCODE-SUB
LEVEL3_1	VARCHAR	12		[note 1]	LB-TAXCODE-LV3
NOMINAL_2	VARCHAR	12		[note 1]	LB-TAXCODE-NOM
SUBACCOUNT_2	VARCHAR	12		[note 1]	LB-TAXCODE-SUB
LEVEL3_2	VARCHAR	12		[note 1]	LB-TAXCODE-LV3
NOMINAL_3	VARCHAR	12		[note 1]	LB-TAXCODE-NOM
SUBACCOUNT_3	VARCHAR	12		[note 1]	LB-TAXCODE-SUB
LEVEL3_3	VARCHAR	12		[note 1]	LB-TAXCODE-LV3
ACCOUNT_DESC_1	VARCHAR	30		[note 1]	LB-TAXCODE-ACCOUNT-DESC
ACCOUNT_DESC_2	VARCHAR	30		[note 1]	LB-TAXCODE-ACCOUNT-DESC
ACCOUNT_DESC_3	VARCHAR	30		[note 1]	LB-TAXCODE-ACCOUNT-DESC
TAXRATE_1	DOUBLE	8		Tax Rate	LB-TAXCODE-RATE
TAXPERCENT_1	DOUBLE	8		Non-recoverable %	LB-TAXCODE-PERCENT
TAX_EFF_DATE_1	DATE	6		Effective date	LB-TAXCODE-EFF-DATE
TAXRATE_2	DOUBLE	8		Tax Rate	LB-TAXCODE-RATE
TAXPERCENT_2	DOUBLE	8		Non-recoverable %	LB-TAXCODE-PERCENT
TAX_EFF_DATE_2	DATE	6		Effective date	LB-TAXCODE-EFF-DATE
TAXRATE_3	DOUBLE	8		Tax Rate	LB-TAXCODE-RATE
TAXPERCENT_3	DOUBLE	8		Non-recoverable %	LB-TAXCODE-PERCENT
TAX_EFF_DATE_3	DATE	6		Effective date	LB-TAXCODE-EFF-DATE
TAXRATE_4	DOUBLE	8		Tax Rate	LB-TAXCODE-RATE
TAXPERCENT_4	DOUBLE	8		Non-recoverable %	LB-TAXCODE-PERCENT
TAX_EFF_DATE_4	DATE	6		Effective date	LB-TAXCODE-EFF-DATE
VALUE_ROUND_TYPE	VARCHAR	1		Value rounding type	LB-TAXCODE-VAL-ROUND-TYPE
VALUE_ROUND_DIGIT	VARCHAR	1	2	Value rounding digit	LB-TAXCODE-VAL-ROUND-DIGIT
RATE_ROUND_TYPE	VARCHAR	1		Gross rounding type	LB-TAXCODE-RATE-ROUND-TYPE
RATE_ROUND_DIGIT	VARCHAR	1	9	Gross rounding digit	LB-TAXCODE-RATE-ROUND-DIGIT

Note 1. There are three account description columns (ACCOUNT_DESC_n). Related to each of these is a set of N/S/L3 columns. The ACCOUNT_DESC column contains the field label that CODA uses to identify a N/S/L3 combination. These field labels vary depending upon the context. For example, ACCOUNT_DESC_1 can contain “Associated account” or “VAT account”. From the CODA Tax maintenance - List screen, these fields are labelled “Account descriptions”.

Table 53. TAX_CODE index details				
Index name	No	Dup	Coll	Fields in index
TAXCODE_001	1	No	A	TAX_CODE

Table 54. TEXT column details					
Column name	Data type	Size	Def	CODA name	IASLINK name
TEXT_KEY	INTEGER	4		Number, To	LB-TEXT-KEY
LANG	VARCHAR	2		Language	LB-TEXT-LANG
TEXT	VARCHAR	80		[field not named]	LB-TEXT-TEXT

Table 55. TEXT index details				
Index name	No	Dup	Coll	Fields in index
TEXT_001	1	No	A	TEXT_KEY, LANG

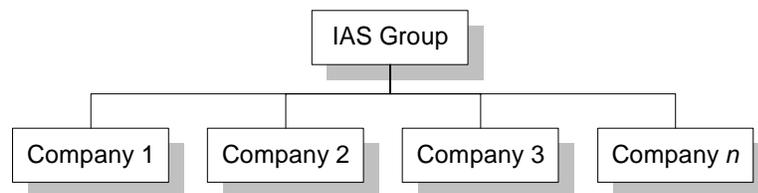
APPENDIX C

CODA IAS Overview

This appendix outlines the structure of data used in the CODA system. It introduces the concepts that are necessary to enable Easysoft to be used, and should not be taken as a full description. It is intended to be used by people who are not familiar with CODA.

In CODA terms a **company** is a separate accounting entity which may be a real company or a part of a company, for example, a division of a corporation. CODA also define a company as “a set of accounts that balance to zero”. A company equates to the general (or nominal) ledger in accounting terms. One or more companies can exist within an **IAS group**. This means a company may share, if appropriate, characteristics of another company within that group. The first company that is set up after installation is the master from which shared details are taken.

CODA Group Architecture



Other structures are possible, for example, the companies (i.e. accounting entities) could be separated into two or more IAS groups. The appropriate organisation depends on how the data is to be used; this issue is beyond the scope of this manual.

Easysoft ODBC for CODA is applicable to the Company level and below.

Company Structure

CODA defines four types of ledger which are subdivisions of the general (nominal) ledger: sales, purchase, expense and fixed asset (not always available). Every ledger has a **Ledger name** which is a 30 character field used to describe the ledger. The **Ledger identifier** is a five character field; the first character is a letter (S, P, E or F) which identifies the ledger type, and the remaining four characters are numbers.

Sales type ledger (S)

Groups accounts that are related to customers. There may be more than one sales ledger, but any customer is associated with only one sales ledger.

Purchase type ledger (P)

Groups accounts that are related to suppliers. There may be more than one purchase type ledger, but any supplier is associated with only one purchase ledger.

Expense type ledger (E)

Groups accounts that are related to income and expenditure.

Fixed Asset type ledger (F)

If this is available, it is used to subanalyse nominal accounts (i.e. top level of the hierarchy) to subaccount level.

Note: for convenience we drop the term “type” when referring to ledger types, and simply refer to them as ledgers. The context should make clear whether we are referring to a ledger type or a specific named instance of a ledger type.

Each CODA company has a chart of accounts which defines the accounting structure for that company. This is a three level hierarchy, unlike a chart of accounts in general accounting terms which has no theoretical limit to the subdivisions of the general ledger. The highest level of the CODA chart of accounts is called the **nominal** level, and accounts defined at this level are called **Nominals**.

It is possible, but not necessary, to have sub-accounts defined below the nominal level. An account at the second level is called a **Subaccount**, and an account at the third level is called a **Level3**. Accounts at these three levels are linked by **Ledger links**. Ledger links define which accounts each Nominal, Subaccount and Level3 can access. If a Subaccount or a Level3 exists, then it must have a Ledger Link which points back up the chart of accounts.

There are restrictions to the structure of the chart of accounts:

- Not all ledgers can be decomposed to the third level; sales and purchase ledgers can exist only at the nominal level, and subaccounts must be of a type called **names and addresses** (these hold information necessary for the operation of sales and purchase accounts).
- Every sales and purchase ledger requires a control account.

In addition to using a chart of accounts, it is possible to group unrelated entities at any given level by means of **Hierarchical Account Groups** (also known as **Account Groups**) This grouping is only used when data is to be read, not when it should be updated.

To use CODA other information not directly associated with a chart of accounts is also needed, and this is discussed next.

Using CODA

Before CODA can be used, it must be installed and initialised. The initialisation process includes the creation of a company, and immediately following this, two users are given different access rights to the company. One is a consultant user with a CODA security

level of 9 (the maximum), and the other is a normal user, with a security level between 1 and 8. The specific rights associated with the numbers 1 to 8 are defined by the organisation that uses CODA. The System Manager / Consultant user has total privileges for the system. The accounting structure is then defined, along with the static data which is needed to run the system. Finally, the system is licensed.

The data contained in CODA can be split into three areas: **Masters**, Transactions and Period Balances. Masters are the static master file records within which the system operates. They are briefly described here.

Account Groups

These are used as an additional level of analysis in the CODA accounting structure. An account can belong to one or more account groups.

Analysis Titles

A method of analysing Name and Address information. Used with IASLINK and Credit Management.

Currency Masters

Contain details of foreign currency and exchange rates.

Document masters

Used to set up the structure of the documents which are used to post (i.e. enter) transactions into CODA. For example, sales invoices. All CODA documents have a **document header**, known as “**line 1**”, and one or more lines (**detail lines**, also called **underneath lines**) containing other information.

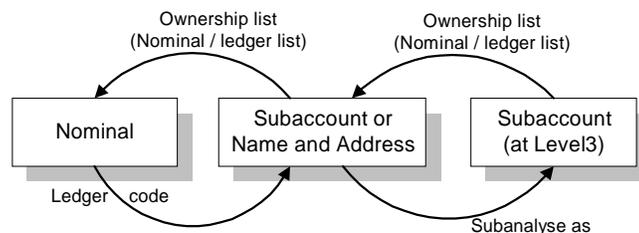
Fixed assets

Used for the accounting of Fixed Assets. This is an optional feature.

Ledgers (links)

Ledgers are used to group accounts. This is shown in the next figure. The Nominal/ledger list is also known as an ownership list. This is used to relate an account to an account or ledger in the level above. If a ledger code is specified in a Nominal, this means that the nominal can be subanalysed. Similarly, if a Subaccount or Name and Address has a values specified for the Subanalyse field, then it can be subanalysed.

Ledger links



Names and Addresses

These are a special type of subaccount for Sales and Purchase ledgers. In addition to information needed purely for the financial aspects of accounting, these accounts also hold information needed in the use of sales and purchase ledger accounts.

Media Headers

For each possible method of payment (e.g. cheque, cash, BACS), a media header contains information about the method of payment.

Nominals

The highest account level in the chart of accounts in CODA. Linked to subaccounts by ledger links.

Subaccounts

There are two types of subaccount in the CODA chart of accounts, namely subaccount and level3. Subaccount is the second level in the chart of accounts, and level3 is the third (bottom) level. Very often, the terms subaccount and level3 are used synonymously in CODA documentation.

Tax codes

Tax codes are used to hold rates of tax which are accessed by CODA using an identification code.

External Access to CODA Data

IASLINK, which is an Application Programming Interface (API), is part of the CODA software. Its purpose is to allow user programs (linked with the IASLINK library) to access CODA account data without using the IAS interface. Furthermore, it is not necessary to know the underlying IAS file structure.

This is achieved by defining the format of data blocks and fields within each block in the data files. There are two defining files: LNKARGBLK.TXT can be included directly in COBOL programs, and LNKARGBLK.H can be included directly in C programs. They are both located in the directory pointed to by the IAS\$TEXT logical on the Server.

Note that the files which are defined are logical - in practice a single file name may be associated with several physical files. Users of IASLINK are completely separated from the physical file structure, and hence do not need to know of any changes which CODA may make.

CODA-IAS Review Questions

1. What defines the accounting structure in CODA?
2. How many levels are there in the accounting hierarchy in CODA?
3. What's the highest level called?
4. What are the two types of subaccount called?
5. What's the function of a ledger?
6. Name the four ledgers in CODA; what are they used for?
7. List two restrictions on the structure of the Chart of Accounts.
8. What 3 categories are used to describe the organisation of data in the CODA system?
9. In addition to the Chart of Accounts, what structure can be used to group data in CODA?
10. What is IASLINK?

CODA-IAS Review Answers

1. What defines the accounting structure in CODA?

Chart of Accounts

2. How many levels are there in the accounting hierarchy in CODA?

Three

3. What's the highest level called?

Nominal

4. What are the two types of subaccount called?

Subaccount (S), Level 3 (L3)

5. What's the function of a ledger?

Used to group related accounts

6. Name the four ledgers in CODA; what are they used for?

Sales, Purchase, Expense, Fixed Asset

- Sales - groups accounts related to sales/customers
- Purchase - groups accounts related to suppliers/purchases
- Expense - groups accounts related to income and expenditure
- Fixed Asset - (if available), for plant and equipment

7. List two restrictions on the structure of the Chart of Accounts.

- a) Sales and Purchase ledgers at Nominal level only
- b) Subaccounts must be Names and Addresses
- c) Sales and Purchase ledgers need a Control Account

8. What 3 categories are used to describe the organisation of data in the CODA system?

Masters, Transactions, Period Balances.

- 9.** In addition to the Chart of Accounts, what structure can be used to group data in CODA?

Hierarchical Account Group.

- 10.** What is IASLINK?

An API which is used to gain access to CODA data without going through the CODA interface.

APPENDIX D

Easysoft ODBC PC Installation

This chapter describes the installation of the Easysoft ODBC Driver.

Easysoft ODBC requires Windows 95 or above and unless there are differences which need to be considered, these platforms are collectively referred to as “Windows”.

Easysoft ODBC storage requirement: 5 megabytes

Easysoft ODBC memory requirement: 4 megabytes

To upgrade from a previous Easysoft version, follow the procedure outlined for installing the software.

Note: When you upgrade the ODBC driver, you must also upgrade the Easysoft Server Component on the server and the version number of these two products must be identical.

Starting the Driver Installation

There are three ways to obtain the Easysoft ODBC Driver software:

- The Easysoft web site is available 24 hours a day at <http://www.easysoft.com> for downloads of definitive releases and documentation. Select Download from the Easysoft ODBC-RMS Driver section of the web site. Log in. (If you have not yet done so, you need to register first. On the registration form, an asterisk (*) indicates that a field is mandatory.) From the download page, choose the client platform release that you require.
- The Easysoft FTP server is available 24 hours a day at <ftp://ftp.easysoft.com>, containing free patches, upgrades, documentation and beta releases of Easysoft products, as well as definitive releases. Change to the pub/download/client/ directory and then choose the platform release that you require.
- You can order Easysoft software on CD by email, telephone or post (see Contacting Easysoft).

The name of the Easysoft ODBC Driver distribution file is of the form:

- `odbc-x.y.z-windows-x86.exe`

where "x" is the major version number, "y" is the minor version number and "z" is the build index.

Driver Installation Steps

Execute the file distribution that you obtained from one of the sources described earlier in this chapter. There will be a short delay while Setup prepares the wizard to guide you through the installation procedure. Then the Easysoft ODBC (Welcome) dialog box appears.



Click Next to continue. The Software License Agreement dialog box is displayed.



After you have read the License Agreement, click Yes to confirm your acceptance of its terms. If you do not accept the terms of the agreement, click No and then click Exit Setup to exit the installation.

The User Information dialog box is displayed.

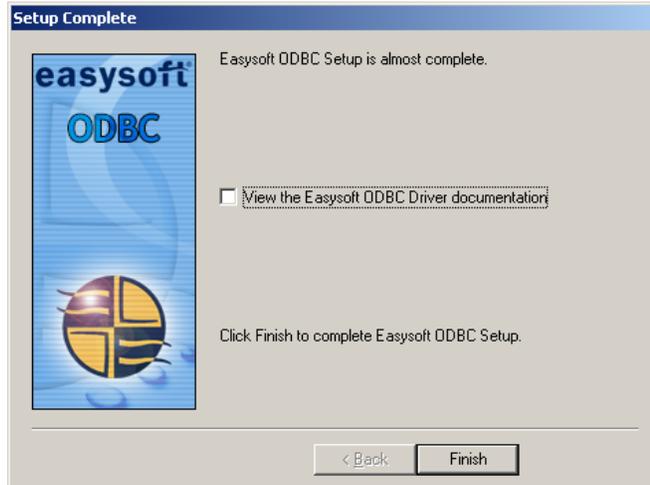


Type your name and company in the spaces provided. Click Next.

In the Start Copying Files dialog box, click Next.



In the Setup Complete dialog box, click Finish to return to Windows. To display the Easysoft ODBC Driver online Help, click the check box.



The installation successfully terminates, and the default Microsoft ODBC Administrator icon is created. Typically, the ODBC Administrator icon can be found in the Control Panel, or under the Start menu.

APPENDIX E

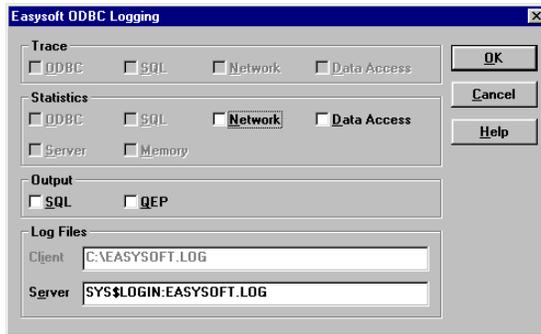
Troubleshooting

This appendix shows you how to investigate problems that may occur, and what to do to resolve them. You will see how to:

- determine what ODBC function calls are being sent to the server
- check what is happening to the network
- see what physical files are accessed by a query and the number of I/Os for that query
- view the SQL that is sent to the server
- view the Query Execution Plan for a query
- use the Microsoft ODBC Administrator to trace ODBC function calls
- work out what is happening at the server (processes running and how to stop them)

Easysoft ODBC Logging

The Easysoft ODBC Logging dialog box is used to specify what is logged and where the log information is saved. It is accessed by selecting **Logging...** on the Easysoft ODBC Setup dialog box.



Easysoft ODBC Logging Information

(Only available options are described)

Trace ODBC	The ODBC functions that are used are recorded in the log file. The option can only be enabled using the EASYSOFT.INI file, - see the next section.
Trace Network	Network information is recorded in the log file. The option can only be enabled using the EASYSOFT.INI file (described below this table).
Network (Statistics)	Statistics about data transmitted and received.
Data Access (Statistics)	Lists the files accessed for a given query and the number of I/Os to files for that query.
SQL (Output)	Lists the SQL query which is passed to the Server.
QEP (Output)	Lists the Query Execution Plan.
Log file (Server)	Path and name of server file to which logging information is sent. Default can be changed.

EASYSOFT.INI

This file controls Easysoft ODBC. An example is shown below, and following this, settings and options pertaining to Easysoft ODBC for CODA are described. If you have an Easysoft product which uses the Easysoft Administrator (Easysoft ODBC for CODA does not), then this file already exists in the Windows directory. If it does not exist, and if you want to use any of the options described here, then create the file using any text editor of your choice. (If you use a word processor, ensure that you save the file as a text file).

```
[Options]
SystemDB=C:\EASYSOFT\SQL\SYSTEM\essystem.sec
[Administrator]
Path=C:\EASYSOFT\SQL\SYSTEM
Language=ENG
Username=Mike
Company=Easysoft Limited

[LOGGING]
ODBC=1
NET=1
```

Explanation of Sections

The order of the sections in the EASYSOFT.INI file is not important.

Comments can be included by prefixing a line with a semicolon character (;).

[LOGGING]

ODBC=1 To turn on the **Trace ODBC** option in the Easysoft ODBC Logging dialog box, set this value to 1. The log file that is generated is called \ESODBC.LOG.

NET=1 To turn on the **Trace Network** option in the Easysoft ODBC Logging dialog box), set this value to 1. The log file that is generated is called \ESNET.LOG.

[NETWORK]

IgnoreWinsock=1 If Pathworks is being used, setting this flag to 1 disables Winsock.

[SETTINGS]

messages=0 To turn off informational messages generated by the Server.

[Options]

AllowReservedWords=1 This option lets you use words reserved for SQL (refer to “SQL Reserved Words” in the manual for a list of these). The default is 0 (i.e. the line can be omitted).

Example Server Log File

The example log file shown here was generated during the use of Microsoft Access. The TAX_CODE table had previously been linked to an Access database and subsequently the SQL query was sent to the table.

```

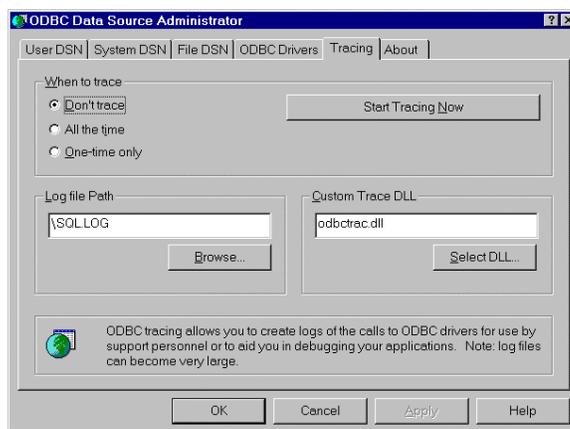
=====
EXECUTE STATEMENT
  SELECT "EASYSOFT_TAX_CODE"."TAX_CODE" FROM "EASYSOFT_TAX_CODE"
=====
QUERY EXECUTION PLAN
Qualifier 0
  Table      : EASYSOFT_TAX_CODE (EASYSOFT_TAX_CODE)
  Index      : 0          Start Mode : 0 (First)
Summary:  Tables [1]   Index Sort [No]   Quals [1]
=====
DATA ACCESS STATISTICS
  Table      : EASYSOFT_TAX_CODE
  Access     : Read Write
  Starts     : 1  Reads : 7  Writes : 0  Deletes : 0
=====
EXECUTE STATEMENT
  SELECT "TAX_CODE", "TITLE", "TAXRATE", "NOMINAL", "SUBACCOUNT", "LEVEL3",
  "SECURITY_LEVEL" FROM "EASYSOFT_TAX_CODE"
=====
QUERY EXECUTION PLAN
Qualifier 0
  Table      : EASYSOFT_TAX_CODE (EASYSOFT_TAX_CODE)
  Index      : 0          Start Mode : 0 (First)
Summary:  Tables [1]   Index Sort [No]   Quals [1]
=====
DATA ACCESS STATISTICS
  Table      : EASYSOFT_TAX_CODE
  Access     : Read Write
  Starts     : 1  Reads : 7  Writes : 0  Deletes : 0

```

Microsoft Trace Options

Version 3.0 Trace Options

The Tracing tab is used to specify how the ODBC Driver Manager traces calls to ODBC functions. The information that is generated using the Tracing tab is not particularly useful to the average user, but if there are problems with your system we may request that you send us a log file.



When to trace

These options can only be set when there is no connection.

Don't trace: disables tracing.

All the time: tracing is performed for all connections at all times.

One-time only: tracing is performed for the next connection, then disabled.

Start Tracing Now

Enables dynamic tracing, which is performed as long as the dialog box is open. Dynamic tracing can be enabled whether or not a connection is open. When this option is selected, the button is replaced with Stop Tracing Now. When Stop Tracing Now is selected, or when the ODBC Administrator dialog box is closed, dynamic tracing is disabled.

Log file Path

Displays the path and file where the log information will be stored. You can change the path and file name by editing the entry box, or by using the **Browse** button.

Custom Trace DLL

If you prefer to use your own custom DLL to perform the tracing operation, replace the default file with the name of your file.

Click the **Apply** button to accept changes without closing the dialog box, or click **OK** to make changes and close the dialog box.

SQL.LOG Described

```
SQLAllocEnv(phenv0006E740);  
SQLAllocConnect(henv0006E740, phdbc0007B658);  
SQLDriverConnect(hdbc0007B658, hwnd00010456, "DSN=Coda - Fulla - User  
Carolyn;UID=MIKEU", 41, szConnStrOut, 256, pcbConnStrOut, 1);  
SQLGetInfo(hdbc0007B658, 2, rgbInfoValue, 256, pcbInfoValue);  
SQLAllocStmt(hdbc0007B658, phstmt000A3AB0);  
SQLGetInfo(hdbc0007B658, 1, rgbInfoValue, 2, pcbInfoValue);
```

The log file that is generated lists the ODBC functions that were called by the query. This file can become huge. For example, during a test performed when this course was developed, the SQL.LOG grew to over 14 Mbytes.

How to Work Out What the Server is Doing

Imagine this scenario. You've started a query from some application on the PC, and the hourglass symbol appears. You wait. And wait... You decide you don't want to wait any longer. But what can you do?

In this section, you will see how to determine what processes are running on the server and how to stop the process that is causing the long wait at the PC. The overall steps are:

1. log on to server
2. determine what process is causing the problem
 - 2.a) monitor the process (optional)
3. view log file (optional)
4. stop the process

Step 1 requires no further explanation.

For steps 2,3 and 5 you will need the appropriate privileges.

The name and structure of the log files that are generated depend upon the network transport that was specified when the Server Component was installed. Here, the examples are based on DEC UCX. The transport will be one of the following:

UCX	(DEC UCX)
DECNET	(DEC PATHWORKS)
TCPWARE	(Process TCPWARE)
MULTINET	(TGC Multinet)
PATHWAY	(Wollongong Pathway)

Someone in your organisation should know which one was used (the default is DECNET).

Find the Process Causing the Problem

- To see what network processes are running type:

```
$ SHOW SYSTEM/NETWORK
```

You will see a list of processes. Here is an example for DEC UCX:

```
OpenVMS V6.2 on node FULLA 27-FEB-1997 11:48:14.86 Uptime 3 01:06:40
  Pid      Process Name      State Pri  I/O      CPU      Page flts  Pages
20200094  EVL                 HIB   6   39  0 00:00:00.37  766      78
N
20200241  UCX$BOOTP_BG3      LEF   10  74  0 00:00:00.94  621      675
N
```

```
2020024F UCX$REXEC_BG298 HIB      5 406 0 00:00:06.27      2446      2021
N
```

- You must make a guess at which of these processes might be the offender. Look at the details of the process, to verify that it is the one causing problems. To do this you would type:

```
$ SHOW PROC/ID=<process id>/CONTINUOUS
```

<process id> is the process identifier, and is found in the first column of the output above, under the Pid heading. /CONTINUOUS instructs the output to be shown on screen.

Continuing the example, and assuming that you thought that the last process in the list was the likely offender, you would type:

```
$ SHOW PROC/ID=2020024F/CONTINUOUS
```

```

                                Process UCX$REXEC_BG298                                11:49:19
State                           HIB                                           Working set                                2021
Cur/base priority               6/4                                           Virtual pages                             7124
Current PC                       7FFEDF8A                                       CPU time                                000:00:00:06.27
Current PSL                      03C00000                                       Direct I/O                               105
Current user SP                  7FEC75D4                                       Buffered I/O                             301
PID                              2020024F                                       Page faults                              2446
UIC                              [MIKEU]                                         Event flags                              60000001
                                                                80000000
DKA300:[VERSION.RMS12B350.SYSTEM]SQLSRV.EXE;12
```

- If the user with a UIC (user identification code - the login name) of MIKEU was the user who sent the query from the PC, and assuming that this user does not have any other processes running on the server, then we know that this is the offending process.

Transport		Process name
UCX	(DEC UCX)	UCX\$REXEC
DECNET	(DEC PATHWORKS)	UCX\$REXEC
TCPWARE	(Process TCPWARE)	with REXEC: REXECCD_<characters> with TCP/IP: EASYSOFT_<digit>
MULTINET	(TGC Multinet)	
PATHWAY	(Wollongong Pathway)	

View Log File

- To view the log file that is generated, type:

```
$ TYPE <log file name>
```

A list of the log file names for each of the transports is shown at the end of this section.

- Continuing with our example, to see the log file for UCX-REXEC type:

```

$ TYPE SYS$LOGIN:UCX$REXECD_STARTUP.LOG

-SYSTEM-F-LINKDISCON, network partner disconnected logical link
MIKEU          job terminated at 27-FEB-1997 11:41:37.99

Accounting information:
Buffered I/O count:          159          Peak working set size:
1326
Direct I/O count:           20          Peak page file size:
6410
Page faults:                 1684        Mounted volumes:
0
Charged CPU time:           0 00:00:02.53  Elapsed time:      0
00:00:14.60
$
    
```

Transport		Log file name
UCX- REXEC	(DEC UCX)	SYS\$LOGIN:UCX\$REXECD_STARTUP.LOG
UCX - TCP/IP	(DEC UCX)	EASYSOFT_SQL_LOG:RUN_SERVER_UCX.LOG
DECNET	(DEC PATHWORKS)	SYS\$LOGIN:NETSERVER.LOG
TCPWARE	(Process TCPWARE)	
MULTINET	(TGC Multinet)	
PATHWAY	(Wollongong Pathway)	

Stop the Process

Once you know which process is causing the problem, you can stop it. To do this, type:

```
$ STOP PROCESS/IDENTIFIER=<process id>
```

When you look at the PC Application, you will see a message. What you see depends upon the application. This example is from Microsoft Excel.



APPENDIX F

Using Easysoft Support Services

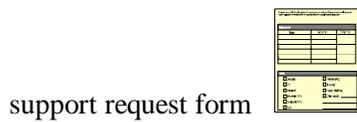
These addresses/numbers are for Support. Other services are also available on different numbers.



support@easysoft.com



01937 860001



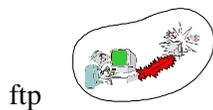
<http://www.easysoft.com/>



01937 860000



Thorp Arch Grange, Thorp Arch,
Wetherby LS23 7BA, United Kingdom



<ftp://ftp.easysoft.com>

Support Check List

Before contacting Easysoft, use the following check list to look for information relevant to the problem you wish to resolve. Only if you can't find anything, or if there is not enough information, contact our support team.

- Manuals (use the Index, Glossary, Contents, Lists of Figures and Tables, Troubleshooting chapters and Appendices).
- Easysoft FAQs (web site and FTP site).
- Third-party documentation where appropriate.
- Easysoft Web site - Technical Support (particularly Fixes), Help, Search.
- If you need to send data to Easysoft, refer to the next but one section, "Sending Data to Easysoft", for information on how to do this.

The Easysoft FTP Site

You've already seen the structure of the FTP site - recall the web tour - so we'll dive straight into accessing data. Let's say you want to access a FAQ sheet on Microsoft Access. There are many different implementations of FTP - here, we'll use Microsoft FTP.



1. Start a DOS session - click the MS DOS Command Prompt () in the Program Manager. The DOS prompt appears. Note the name of the current directory - this is where the file you download will be put.

2. Start an FTP session and connect to the Easysoft FTP site

```
\users\default> ftp ftp.easyssoft.com
```

3. An introductory message appears, after which you must enter a user name and password. The username to use is "anonymous" and the password is your full email address

```
User (lodur.easyssoft.com:(none)): anonymous
331 Guest login ok, send your complete e-mail address as password.
Password: <enter your email address here>
<additional messages may appear here>
230 Guest login ok, access restrictions apply.
```

1. Change directory to /pub/faq/windows and list the files there

```
ftp> cd /pub/faq/windows
```

```
ftp> dir
```

```
PORT command successful.
150 Opening ASCII mode data connection for /bin/ls. total 19288
<directory listing>
<informational messages>
```

2. ftp> bin
200 Type set to I.

The files should be downloaded in binary mode.

3. Download the Access 2.0 FAQ into the current directory.

```
ftp> get access2.doc (goes in directory from which FTP was called)
```

Note that filenames are case sensitive.

4. Just to check that you have what you want, start Microsoft Word, and look at this file.

Sending Data to Easysoft

To send data to Easysoft do the following:

1. Preparation

If you are requested to send your data to Easysoft, you need to make sure that you send both the Easysoft Catalog and the data. Before you back up the data make sure that no one is using the software by using the Host Administrator SHOW USERS command:

```
$ RUN EASYSOFT_SQL_ADMIN
```

```
ADMIN> SHOW USERS
```

2. Sending data

- On tape

Back up the Catalog and the data. You can back up to TK50 or DAT and send it in the post. e.g.

```
$ BACKUP/REWIND/VERIFY EASYSOFT_SQL_CATALOG:*.*,DATA:*. * -
MKA700:EASYSOFT/LOG
```

- On floppy

Back up the Catalog and the data to a disk file. e.g.

```
$ BACKUP/REWIND/VERIFY EASYSOFT_SQL_CATALOG:*.*,DATA:*. * -
EASYSOFT.BCK/SAVE/LOG
```

Create a Zip file using ZIP. e.g.

```
$ ZIP:==$EASYSOFT_SQL_SYSTEM:ZIP.EXE
$ ZIP "-v" EASYSOFT.ZIP EASYSOFT.BCK
```

Send the zipped files on a floppy disk to Easysoft.

- Electronically

Create a .ZIP file (see above) and then use one of the following methods:

Internet Mail: support@easysoft.com

Internet FTP: ftp.easysoft.com

Global Index

The page numbering shows the module number followed by the page number within the module.

%

% character
in SQL LIKE operator, 2-12

*

* (in SQL statement), 2-9

A

About dialog box. *See under* CODA macro
Account Group, A-16, A-28
Account Groups, A-8, A-29
restrictions on inserting data, A-20, A-22, A-23
Account Groups Download dialog box. *See under* CODA macro
ACCOUNT table, 5-8, 6-9
generating, A-8
ACCOUNT_BALANCE, 9-27
Actual Budget Variance, 9-34
using macro, 9-37
Actual Budget Variance report. *See under* CODA macro
Add Criteria dialog box, 7-5
Add criteria in application. *See under specific application*
Add Data Source dialog box, 4-3
Add rows to table. *See* INSERT statement (SQL)
Add Tables dialog box, 7-4
Adjustment period, A-8
Age Analysis dialog box. *See under* CODA macro
Age Analysis report. *See under* CODA macro
Ageing Summary, 9-45
using macro, 9-47
Aggregate functions, 2-14
Allow editing in MS Query. *See* Microsoft Query
American National Standards Institute. *See* ANSI
Analysis Titles, A-29
AND operator (SQL), 2-11
ANSI, 1-13
API, 1-4
conformance. *See* ODBC conformance
Application
using ODBC, 1-5
Application Programming Interface. *See* API
Architecture
Easysoft. *See* Easysoft architecture

ODBC. *See under* ODBC

Available CODA Companies dialog box. *See under* CODA macro

Available Data Sources list box, 7-2

AVG (SQL), 2-14

B

Balance Sheet, 9-41
using macro, 9-44
Balance Sheet report. *See under* CODA macro
Balance Sheet Selection Criteria dialog box. *See under* CODA macro
BALANCE table, 5-3, 5-8, 6-9
Batch job for Codaxref. *See under* Codaxref routine
BETWEEN operator (SQL), 2-11
Brackets in SQL. *See under* SQL, parentheses
Budget Download dialog box. *See under* CODA macro
Budget periods
number supported, A-9
BUDGET table, 5-3, 5-8, 6-9, 9-27
Budget Upload dialog box. *See under* CODA macro
Budget Variance Selection dialog box. *See under* CODA macro
Budgets
downloading and uploading, 9-27

C

Catalog. *See* Easysoft catalog functions, 1-10
Catalog Driver, 3-2
Catalog Login
changing password, 5-3
password, 5-6
username, 5-6
Catalog password
changing, 5-2, 5-3
Catuser routine, 5-3
example, 5-12
Cell reference, 10-22
Change rows in table. *See* UPDATE statement (SQL)
Chart of accounts
CODA, A-28
CODA restrictions, A-28
Choose Fields dialog box, 7-17
CODA

- access rights, A-29
- company. *See* Company
- identification number, referencing, A-8
- menu name
 - correspondence to column name, 6-6
- security level. *See* CODA access rights
- tables, defined, 6-2
- CODA data
 - allowed operations, 3-3
 - making relational, 3-2
 - retrieving - expected results, 9-2
 - retrieving and updating, 9-1-9-52
 - updating using MS Excel, 9-2
 - updating using MS Query, 9-2
- CODA Driver, 3-3
- CODA macro
 - About dialog box, 10-24
 - Account Groups Download dialog box, 10-7
 - Actual Budget Variance report, 10-12
 - Age Analysis dialog box, 10-13
 - Age Analysis report, 10-13
 - Available CODA Companies dialog box, 10-5
 - Balance Sheet report, 10-14
 - Balance Sheet Selection Criteria dialog box, 10-14
 - Budget Download dialog box, 10-11
 - Budget Upload dialog box, 10-11
 - Budget Variance Selection dialog box, 10-12
 - Budgets, 10-10
 - changing period range, 10-10
 - prevent deletion of worksheet data, 10-10
 - contact information, 10-24
 - Currency Lookup dialog box, 10-20
 - Currency Rates Download dialog box, 10-8
 - customised reports, 10-21
 - creating, 10-21
 - generalising the report, 10-22
 - running, 10-22
 - debug options, 10-6
 - define Budget upload format, 10-10
 - Details Download dialog box, 10-18
 - document details download, 10-18
 - download Budgets, 10-11
 - failed update colour, 10-9
 - filename, 10-2
 - Fixed reports, 10-12
 - font colour after upload, 10-9
 - IASLINK errors, 10-9
 - initialisation, 10-5
 - Input and Inray, 10-18
 - installation, 10-3
 - Invoice Setup dialog box, 10-19
 - keeping connection open, 10-6
 - location of add-in software, 10-3
 - lookups, 10-20
 - Options dialog box, 10-5
 - overview, 10-2
 - post documents, 10-19
 - prevent display of error messages, 10-6
 - Profit & Loss Selection dialog box, 10-15
 - Profit and Loss report, 10-15
 - restricting number of rows, 10-6
 - selecting CODA company, 10-5
 - selecting year range, 10-6
 - selection criteria, 10-8
 - setting timeout value, 10-6
 - Setup Budget Format dialog box, 10-10
 - status message, 10-6
 - successful insert colour, 10-9
 - successful update colour, 10-9
 - Table Download dialog box, 10-7
 - Technical Support dialog box, 10-24
 - Trial Balance dialog box, 10-16
 - Trial Balance report, 10-16
 - update sequence rules, 10-9
 - upgrading, 10-4
 - upload, 10-9
 - upload Budgets, 10-11
 - Upload dialog box, 10-19
 - upload invoice. *See* CODA macro, post documents
 - preparation, 10-19
 - upload status, 10-9
 - version information, 10-24
- CODA Main Menu, 6-2
- CODA Masters Menu, 6-3
- Coda menu option, 10-2
- CODA password
 - changing, 5-10
- CODA tables
 - relationships between, 6-5
- CODA user, 5-2, A-28
 - adding to catalog, 5-5
 - deleting from catalog, 5-5
- CODA Utilities, 3-3
 - purpose of, 5-2
- Codacat routine, 5-2, 5-3, 5-4-5-6
- Codauser routine, 5-2, 5-3
 - example, 5-11
 - function of, 5-6
 - location of, 5-6
- Codaxref routine, 5-2, 5-3, 5-8, 6-9
 - modifying batch job options, 6-9
 - when to re-run, 5-9
- Column
 - defined, 2-3
- Column name
 - allowed characters, 2-8
- Combining conditions in SQL, 2-11
- Combining data from different tables. *See* Join Company, 5-2
 - adding to catalog, 5-4
 - defined, A-27
 - example for training, 8-2
 - structure, A-28
- Conditions
 - Combining in SQL. *See* Combining conditions
- Connect Timeout. *See* Timeouts
- Connect to External dialog box, 7-16
- Control account
 - listing with ledger, 9-7
- Converting data types. *See under* Data type
- COUNT (*) (SQL), 2-14
- COUNT DISTINCT (SQL), 2-14

Create New Data Source dialog box, 4-9
 Criteria
 brackets, 9-12
 Criteria pane, 7-4
 Currency conversion, 9-16
 Currency Lookup dialog box. *See under* CODA macro
 Currency Masters, A-29
 Currency rates
 obtaining from Internet, 9-16
 Currency Rates Download dialog box. *See under* CODA macro
 CURRENCY table, 9-15
 CURRENCY_RATE table, 9-15
 Customised reports. *See under* CODA macro

D

Data
 sending to Easysoft, A-49, A-51
 Data access
 information in Easysoft catalog. *See under* Easysoft catalog
 Data source, 1-4, 1-11
 configuring, 4-10
 defined, 1-6
 method of definition, 3-4
 naming restrictions, 4-4
 removing, 4-10
 setting up, 4-2, 4-9
 settings, 4-6
 validation of, 4-5
 Data source name. *See* DSN
 Data Sources dialog box, 4-3
 Data Sources dialog box (16 bit), A-37
 Data type
 conversion, 1-11
 purpose of, 2-12
 Database, 1-4
 defined, 2-3, 3-4
 specifying in data source, 4-7
 Database dialog box, 7-11, 7-12
 Database management system. *See* DBMS
 Database structure
 pictorial representation, 2-8
 DATE data type, 1-10
 Date format
 default, 6-7
 DBMS, 1-4
 DELETE statement (SQL), 2-17
 Detail lines, A-29
 Details Download dialog box. *See under* CODA macro
 DETAILS table
 difference between INTRAY, A-14
 Disable Winsock. *See under* Winsock
 Document header, A-29
 Document master, A-29
 sequence rule, 9-49, 9-49
 Downloading budgets. *See* Budgets
 Driver. *See* ODBC driver

Driver Manager, 1-5, 1-6
 DSN, 1-11
 Duplicate rows
 eliminating, 2-13
 Dynamic tracing. *See* Function calls, tracing

E

EASYF001.DAT, 6-10
 Easysoft architecture, 3-2
 Easysoft Catalog, 2-25, 3-2
 adding CODA company, 5-3
 CODA changes affecting, 5-2
 data access information, 3-5
 mapping information, 3-5
 number of, 3-5
 purpose of, 3-5
 specifying location of, 4-4
 user information, 3-5
 Easysoft Client component, 3-2
 Easysoft Excel Macro for CODA. *See* CODA macro
 Easysoft ODBC
 cancelling installation, A-37
 driver installation, A-34
 driver versions, A-34
 installation
 from floppy disk, A-35
 from hard disk, A-35
 memory requirement, A-34
 storage requirement, A-34
 system requirement, A-34
 upgrading, A-34
 Easysoft ODBC Logging dialog box, A-39
 Easysoft ODBC login prompt
 hiding and showing, 4-6
 Easysoft ODBC Settings dialog box, 4-6
 Easysoft ODBC Setup dialog box, 4-4, 5-12, A-37
 Easysoft Server component, 3-2
 Easysoft SQL, 3-2
 Easysoft Support
 check list, A-49
 contact options, A-48
 EASYSOFT.INI, A-40
 EASYSOFT_SQL_CODA_CATALOG logical, 6-10
 Easysql
 using, 5-16
 Eliminating duplicate rows. *See under* Duplicate rows
 Errors
 dealing with in ODBC. *See under* ODBC
 ESNET.LOG, A-40
 ESODBC.LOG, A-40

F

File data source, 4-8
 Files
 accessed by a query, A-39

Final period, A-8
 Fixed assets, A-29
 Formula bar, 10-22
 Function call, 1-6
 Function calls
 tracing, A-42

G

Get External Data dialog box, 7-8

H

Hierarchical Account Group. *See* Account Group
 Host Administrator, 3-2

I

I/Os per query
 number of, A-39
 IAS group, A-27
 IAS\$GROUP
 define, 6-10
 IAS\$TEXT directory, A-30
 IASLINK, 3-3, A-30
 Index
 ascending, 2-5
 descending, 2-5
 disadvantages, 2-5
 non-unique, 2-5
 purpose, 2-5
 unique, 2-5
 Inner join. *See under* Join
 Input and Inray. *See under* CODA macro
 INSERT statement (SQL), 2-15
 Install Drivers dialog box, A-36
 International Standards Organisation. *See* ISO
 Inray, A-14
 INTRAY table
 difference between DETAILS. *See under*
 DETAILS table
 Invoice
 download, 9-50
 upload, 9-48
 Invoice Setup dialog box. *See under* CODA
 macro
 Invoice Upload dialog box. *See under* CODA
 macro
 ISO, 1-13

J

Join, 2-6
 in SQL, 2-12
 inner, 2-6
 outer, 2-6
 types, 2-6

L

Ledger

link, A-29
 listing, 9-6
 listing with control account, 9-7
 Ledger identifier, A-27
 Ledger link, A-28
 Ledger name, A-27
 Ledger type
 Expense, A-28
 Fixed asset, A-28
 Purchase, A-28
 Sales, A-27
 Left outer join. *See under* Join
 Level3, A-30
 add or delete, 5-2
 defined, A-28
 relationship to Codaxref routine, 5-9
 LIKE operator (SQL), 2-11, 2-12
 Line 1, A-29
 Link dialog box, 7-10
 Link Tables dialog box, 7-11
 LNKARGBLK.H, A-30
 LNKARGBLK.TXT, A-30
 Lotus 1-2-3
 add criteria, 7-18
 using, 7-15
 view SQL, 7-18
 lotus.bcf file, 7-15

M

Mail Merge, 7-19
 Mail Merge Helper dialog box, 7-19
 Mail Merge toolbar, 7-21
 Mappings
 information in Easysoft catalog. *See under*
 Easysoft catalog
 Masters, A-29. *See also under individual names*
 MAX (SQL), 2-14
 Media Headers, A-30
 Messages
 hiding, A-40
 Microsoft Access
 add criteria, 7-12
 design mode of table, 7-14
 using, 7-10
 view indexes, 7-14
 view SQL, 7-13
 Microsoft Excel
 Formula bar. *See* Formula bar
 reconnection to data source, 9-2
 show column headings, 7-9
 show row numbers, 7-9
 specify download location, 7-9
 using, 7-2
 Microsoft Internet Explorer, 9-16
 Microsoft ODBC Administrator, 4-1, A-37
 apparent version anomaly, A-34
 purpose of, 4-2
 version 2.5 (16 bit), 4-2
 version 2.5 (32 bit), 4-2
 version 3.0, 4-2, 4-8
 version shipped, A-34

- version shipped with Easysoft, 4-2
- Microsoft ODBC Setup
 - running, A-35
- Microsoft Query
 - add criteria, 7-5, 7-6
 - adding criteria, 9-3
 - allow editing, 9-2
 - modify SQL in query, 7-7
 - reading file, 9-2
 - reconnection to data source, 9-2
 - switch off automatic querying, 9-4
 - using, 7-2
 - view SQL, 7-7
 - wildcard character, 9-2
- Microsoft Word Mail Merge. *See* Mail Merge
- MIN (SQL), 2-14
- Modify rows in table. *See* UPDATE statement (SQL)

N

- Names and addresses, A-28, A-30
- Naming restrictions in data source. *See under* Data source
- Network connection
 - defining packet size. *See* Packet size
 - packet acknowledgement. *See* Packet acknowledgement
- Network statistics, A-39
- Network transport
 - PC options, 4-4
- New Query Assistant dialog box appears, 7-16
- New Query dialog box, 7-12
- Nominal, A-30
 - add or delete, 5-2
 - relationship to Codaxref routine, 5-9
- Nominal account. *See* Nominal
- Nominal/ledger list, A-29
- Non-unique index. *See under* Index
- NOT operator (SQL), 2-11
- NULL operator (SQL), 2-11

O

- ODBC
 - architecture, 1-5
 - components
 - application, 1-5
 - data source, 1-5
 - determining details of, 4-10
 - driver, 1-5
 - driver manager, 1-5
 - conformance, 1-13
 - error checking, 1-6
 - purpose of, 1-4
 - ODBC Administrator icon, A-37
 - ODBC Data Sources dialog box, 7-3
 - ODBC date format, 2-14
 - ODBC driver
 - 16 and 32 bit. *See under* Easysoft ODBC driver

- functions of, 1-8
- information about, 4-10
- listing, 4-10
- multi-tier. *See* two-, and three-tier
- single-tier, 1-6
- three-tier, 1-6
- two-tier, 1-6
- ODBC Drivers tab, 4-10
- ODBC icons, A-37
- ODBC Options dialog box, A-42
- ODBC Setup (welcome) dialog box, A-36
- Olsen and Associates Currency Converter, 9-16
- One-to-many relationship, 2-8
- Open Data Source dialog box, 7-20
- Opening period, A-8
- Options dialog box. *See under* CODA macro
- OR operator (SQL), 2-11
- ORDER BY statement (SQL), 2-13
- Order data
 - by N/S/L3 (CODA), 5-3, 5-8, 6-9
 - relational. *See* Relational data
- Ordered retrieval of data. *See under* Relational data
- Outer join. *See under* Join
- Ownership list, A-29

P

- Packet acknowledgement
 - enabling, 4-7
- Packet size
 - defining, 4-7
- Parentheses
 - using in SQL, 2-11
- Passwords and usernames
 - for Easysoft Catalog, 4-5
 - Server, 4-5
- PERIOD field
 - valid range, A-8
- posting
 - using DETAILS table, A-14
- Primary key
 - indicated by font, 6-6
- Profit & Loss Selection dialog box. *See under* CODA macro
- Profit and Loss Account, 9-38
 - using macro, 9-40
- Profit and Loss report. *See under* CODA macro

Q

- Query Execution Plan, A-39

R

- Receive Timeout. *See* Timeouts
- Relational data
 - date format, 2-4
 - ordered retrieval, 2-5, 2-13
 - structure, 2-3
- Remote Command

- specifying name of, 4-4
- Remote Object
 - specifying name of, 4-4
- Remote Service
 - specifying name of, 4-4
- Remove rows from table. *See* DELETE statement (SQL)
- Removing duplicate rows. *See under* Duplicate rows, eliminating
- Reserved words (SQL), 2-26
- retrieving CODA data. *See* CODA data
- Right outer join. *See under* Join
- Rounding in SQL, 2-17
- Row
 - defined, 2-3

S

- Salary Journal
 - download, 9-52
 - upload, 9-51
- Scalar comparison operator (SQL), 2-10
- Select Data Source dialog box, 7-3, 7-10
- SELECT DISTINCT (SQL), 2-13
- Select ODBC Trace File dialog box, A-42
- SELECT statement (SQL), 2-9
- Select Value(s) dialog box, 7-6
- Sequence rule. *See under* Document master. *See under* Document master
- Sequencing rule dependencies, A-18
- Server process
 - determining, A-45
 - stopping, A-47
- Set Criteria dialog box, 7-18
- Setup Budget Format dialog box. *See under* CODA macro
- SHARE.EXE, A-34
- Show SQL dialog box, 7-18
- Show Table dialog box, 7-12
- SQL, 1-4
 - conversion in ODBC, 1-10
 - defined, 2-3
 - ODBC extensions, 1-10, 2-3
 - purpose of, 2-3
 - using parentheses, 2-11
- SQL Access Group, 1-13
- SQL dialog box, 7-7
- SQL grammar conformance. *See* ODBC conformance
- SQL query, A-39
- SQL reserved words. *See* Reserved words
- SQL statements
 - supported by Easysoft SQL, 2-25
- SQLColumns function, 1-10
- SQLDataSources function, 1-11
- SQLDrivers function, 1-11
- SQLGetFunctions function, 1-11, 1-13
- SQLGetInfo function, 1-11, 1-13
- SQLGetTypeInfo function, 1-11, 1-13
- SQLStatistics function, 1-10
- SQLTables function, 1-10

- Standards for SQL. *See under* SQL
- Structured Query Language. *See* SQL. *See* SQL
- Subaccount, A-30
 - add or delete, 5-2
 - defined, A-28
 - relationship to Codaxref routine, 5-9
- SUM (SQL), 2-14
- System data source, 4-8
- System Data Source dialog box, 4-3
- System DSN tab, 4-9

T

- Table
 - defined, 2-3
 - differentiating between databases, 4-7
- Table Download dialog box. *See under* CODA macro
- Table name
 - allowed characters, 2-8
- Table pane
 - viewing, 7-4
- Tax code, A-30
- Technical Support dialog box. *See under* CODA macro
- Timeouts, 4-6
- Trace of Easysoft network protocols, A-40
- Trace of ODBC calls, A-40
- Tracing function calls. *See under* Function calls
- Tracing tab, A-43
- Training company. *See under* Company
- Transmit Timeout. *See* Timeouts
- Trial Balance, 9-31
 - with macro, 9-33
- Trial Balance dialog box. *See under* CODA macro
- Trial Balance report. *See under* CODA macro

U

- Underneath lines. *See* Detail lines
- Unique index. *See under* Index
- Update rows in table. *See* UPDATE statement (SQL)
- UPDATE statement (SQL), 2-16
- Updating CODA data. *See* CODA data
- Uploading budgets. *See* Budgets
- User
 - information in Easysoft catalog. *See under* Easysoft catalog
- User data source, 4-8

V

- View SQL in application. *See under specific application*

W

- Wildcard character
 - Codacat routine, 5-5

in Microsoft Query. *See under* Microsoft
Query
SQL, 2-12
Winsock
disable, A-40

X

X/Open, 1-14

