

BOOK 2

COMM1E - Database Systems

ORACLE

An Introduction To Forms

Part I

© Sue Stirk

School of Computing & Technology

University of Sunderland, 2002

Version 1 - Created 17/07/2002

Contents

1. INTRODUCTION TO FORMS	4
1.1 <u>INTRODUCTION</u>	4
1.2 <u>OBJECTIVES</u>	4
1.3 <u>FORM BUILDER MAIN SCREEN</u>	4
1.4 <u>FORMS OBJECT NAVIGATOR</u>	5
1.5 <u>CONNECTING TO THE DATABASE</u>	6
1.6 <u>USING FORMS</u>	7
2. CREATING A DEFAULT FORM	8
2.1 <u>INTRODUCTION</u>	8
2.2 <u>OBJECTIVES</u>	8
2.3 <u>CREATING A DEFAULT FORM</u>	8
2.3 <u>THE LAYOUT EDITOR</u>	18
2.4 <u>SAVING AND RUNNING FORMS</u>	20
2.4.1 <u>Saving a Form</u>	20
2.4.2 <u>Compiling a Form</u>	20
2.5 <u>BASIC CONCEPTS</u>	23
2.5.1 <u>Items</u>	23
2.5.2 <u>Canvases</u>	23
2.5.3 <u>Windows</u>	23
2.5.4 <u>Blocks</u>	23
3. MANIPULATING DATA USING ORACLE FORMS	25
3.1 <u>INTRODUCTION</u>	25
3.2 <u>OBJECTIVES</u>	25
3.3 <u>QUERY BY EXAMPLE - SIMPLE QUERIES</u>	25
3.4 <u>QUERY BY EXAMPLE - COMPLEX QUERIES</u>	26
3.4.1 <u>Queries using OR</u>	26
3.4.2 <u>Queries using ORDER BY</u>	29
3.4 <u>INSERTING RECORDS</u>	30
3.5 <u>UPDATING RECORDS</u>	31
3.6 <u>DELETING RECORDS</u>	32
4. CREATING MASTER/DETAIL FORMS	33
4.1 <u>INTRODUCTION</u>	33
4.2 <u>OBJECTIVES</u>	34
4.3 <u>A MASTER/DETAIL FORM</u>	34
5. CUSTOMISING FORMS - THE BASICS	37
5.1 <u>INTRODUCTION</u>	37
5.2 <u>OBJECTIVES</u>	37
5.3 <u>THE LAYOUT EDITOR</u>	38
5.3.1 <u>Changing and Adding Text</u>	38
5.3.2 <u>Moving Objects</u>	39
5.3.3 <u>Changing Font and Text Size</u>	39
5.3.4 <u>Changing Colours</u>	39
5.4 <u>THE PROPERTIES PALETTE</u>	40
5.4.1 <u>Changing properties</u>	42
5.4.2 <u>Viewing more than one Properties Palette at once</u>	42
5.4.3 <u>Adding a List Item</u>	43
5.4.4 <u>Adding Hint text</u>	44
5.4.5 <u>Adding Tool tips</u>	45
5.4.6 <u>Adding Radio Groups</u>	45
5.4.7 <u>Adding New Items</u>	47

5.4.8	<i>Calculating the value of an item.....</i>	<i>49</i>
5.4.9	<i>Creating a List of Values.....</i>	<i>50</i>
CASE STUDY.....		55

1. Introduction to Forms

1.1 Introduction

Oracle Developer 6i is an integrated development environment that allows you to build database applications for both the Internet and client/server systems. Oracle Developer consists of a number of tools such as Forms, Graphics and Reports. This Unit will introduce you to some of the basic building blocks of one component of Developer, namely, Forms Builder. This Unit will use the Property Case Study that you were introduced to in Unit 1. Where necessary, other Case Studies will be introduced so that you can apply what you learn using different tables.

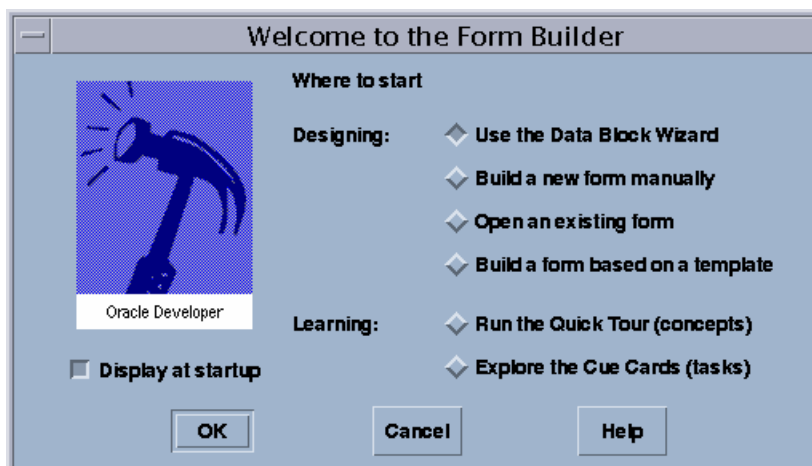
1.2 Objectives

When you have completed this section you will be able to:

- Start up Forms
- Connect to the database from the Forms Builder main screen

1.3 Form Builder Main Screen

The first thing you have to do is to start up Forms. You will be given separate instructions from your tutor on how to do this. When you start Forms Builder you will see the Welcome Screen:

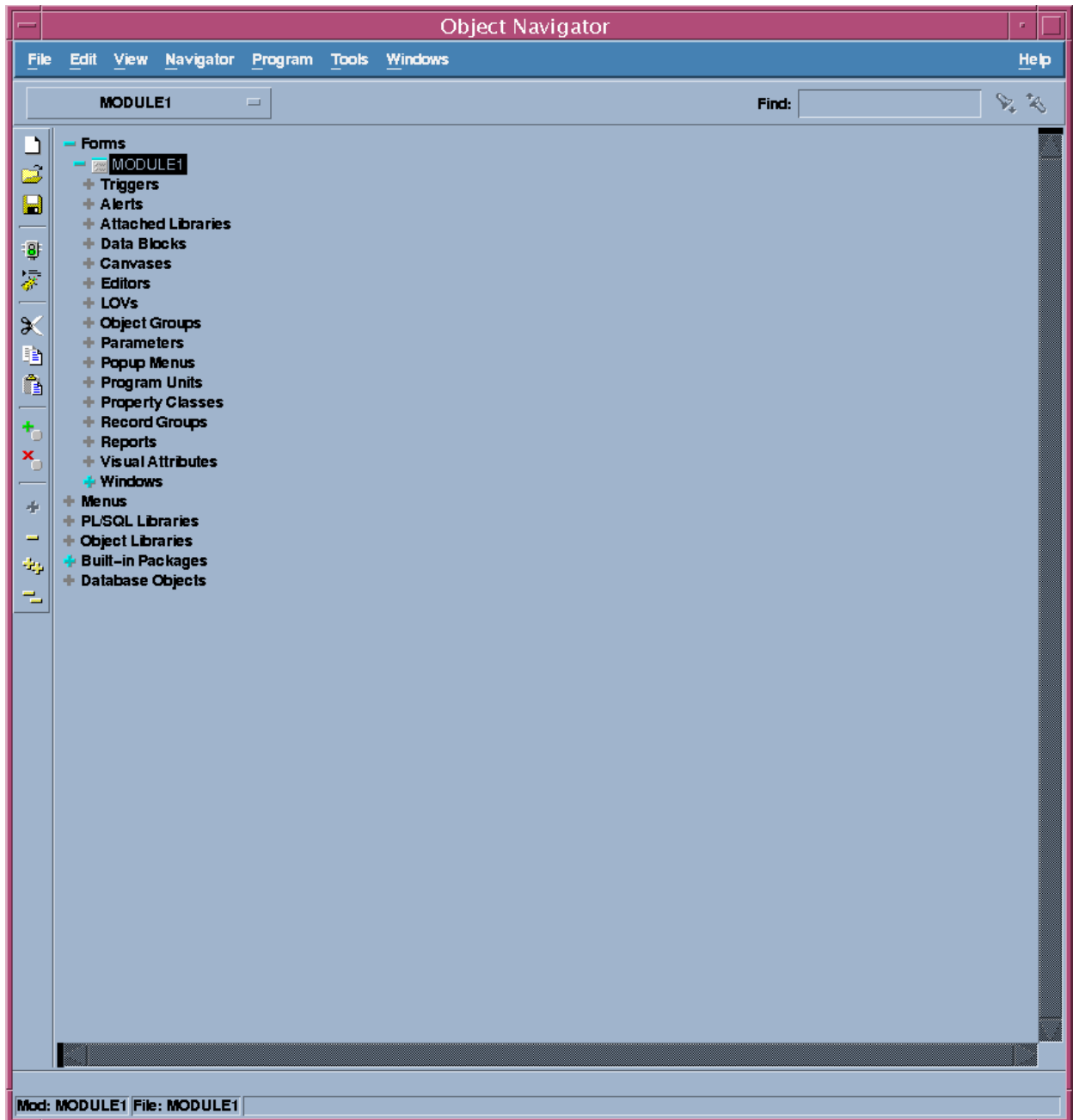


This screen gives you a quick way of developing forms, but we will not use this initially so that you can see one of the most important Forms Builder screens first.

Click on the cancel button for now – we will return to this later.

1.4 Forms Object Navigator

You should now see the main Forms screen. This is the Object Navigator:



The Object Navigator screen displays all of the objects that make up a form. Everything in the Object Navigator is displayed in the form of a tree. Every object in a form will have either a plus (+) or minus (-) sign at the side of it. Clicking on a + will expand an object to show more details. Clicking on a - will collapse the object so that the detail is not displayed. It works like the Windows Explorer screen.

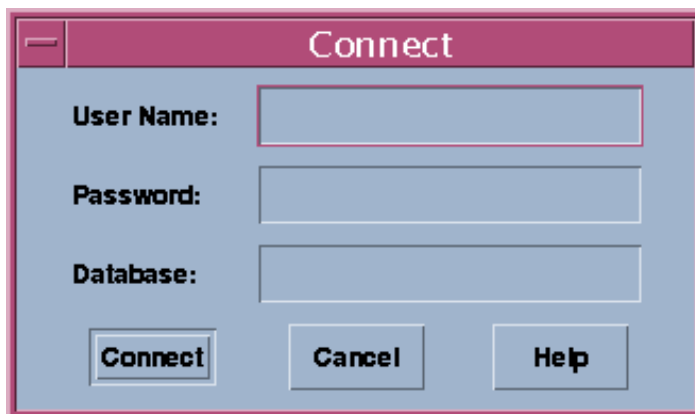
Just like all other Windows applications, there is a menu bar across the top of the screen.

Drop down some of the menu's so that you can see what they contain.

You will also see that at the side of this screen there is a button bar with icons on it. Most of the buttons do the same thing as some of the menu items. For example the 'open' icon at the top of the button bar is the same as pulling down the File menu and then selecting open. The buttons therefore are shortcuts because they involve less clicking of the mouse. If you want to know what the buttons do then place the mouse over the button and a short piece of text will describe the button.

1.5 Connecting to the Database

Before you can create any forms you need to connect to the database. To connect to the database select **[File/Connect]** from the Main Menu. You will see a connect window similar to the one below. Fill in your Oracle username, password and database details and then press the Connect button.



The image shows a 'Connect' dialog box with a red title bar. Inside, there are three text input fields labeled 'User Name:', 'Password:', and 'Database:'. At the bottom, there are three buttons: 'Connect', 'Cancel', and 'Help'.

1.6 Using Forms

The Case Study you are using contains several tables that make up the database. Forms are used to make manipulating the data in those tables easier. Until now the interaction you have had with ORACLE is via SQL. You have typed in commands at the SQL> prompt, or in a text file. However, most users would not have the knowledge of SQL syntax to enable them to type in such commands. To get round this we use Oracle Forms to allow non-technical users to perform similar types of action. In the next section we will look at how to create basic forms.

EXERCISE 1

1. List the items that are available from the File menu

2. What does the traffic lights icon on the side menu bar do?

3. What is the equivalent menu command?

2. Creating a Default Form

2.1 Introduction

Every form created in Oracle Forms Builder has the capability to perform several actions by default, without the application developer having to undertake any programming. The default functionality that exists is the ability to retrieve data, update existing data, insert new data, and to delete data. This section will examine how to create, save and run a simple form.

2.2 Objectives

When you have completed this section you will be able to:

- Create a default form
- Save a form
- Compile a form so that it will run
- State what different file extensions represent

2.3 Creating a Default Form

There are several types of layout that you can use when creating a form but in this section we will create the simplest type of form and then test its default functionality. The simplest kind of form you can create is a form that is based on one table from the database. A **Data Block** on a form usually corresponds to a table in the database. The table that the form is based on is called the **base table**. Our first steps are to give our new form a name and to create the Data Block. Data Blocks can be created quickly and easily using the Data Block Wizard provided with Forms Builder. First though we need to give our new form a name.

In the Object navigator on your screen you will see at the top the forms object name *Module 1*. This is the default name that Forms Builder gives to a new form. If you created a second form it would be called *Module 2* and so on. We want our form to be called Property and we will base it on the prop_for_rent table.

EXERCISE 2

1. Create a default form called **property.fmb** following the steps below

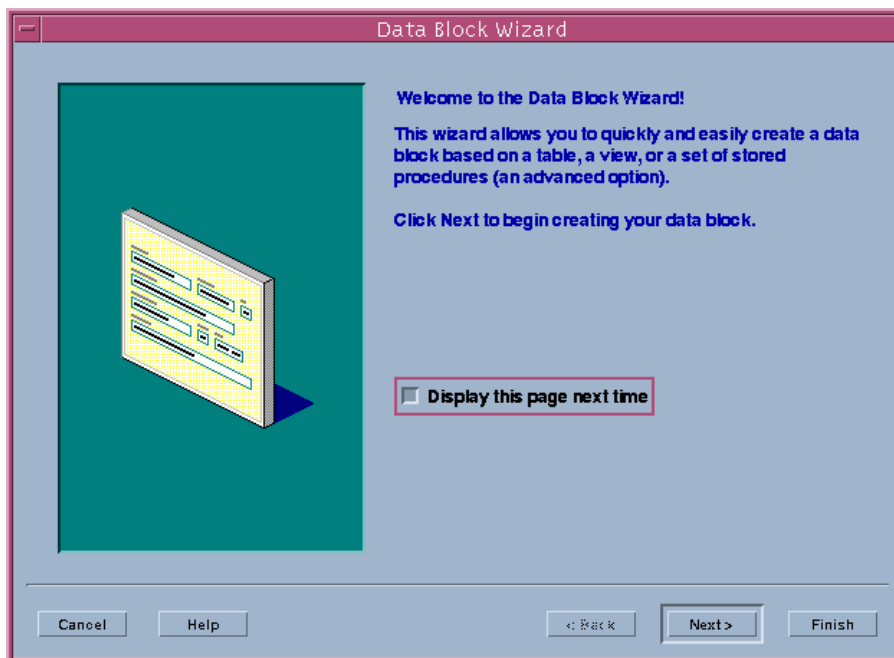
STEP 1 - Name the form

Click twice on the Form's name - Module 1

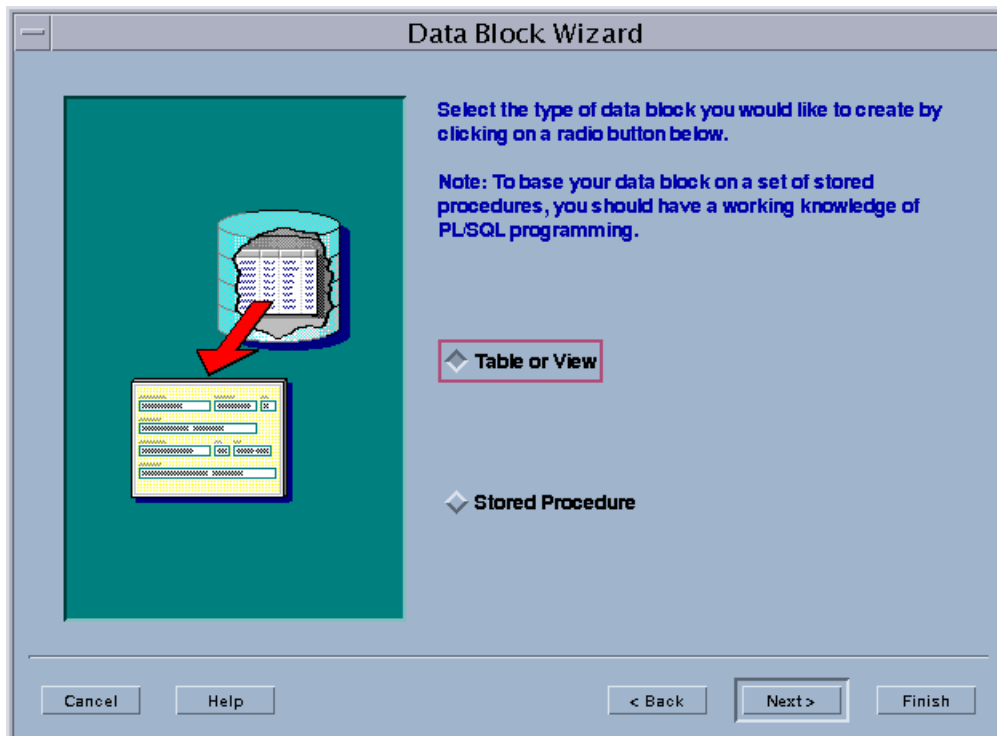
Change its name to Property (notice that Forms Builder automatically changes what you type to upper case)

STEP 2 - Create the Data Block

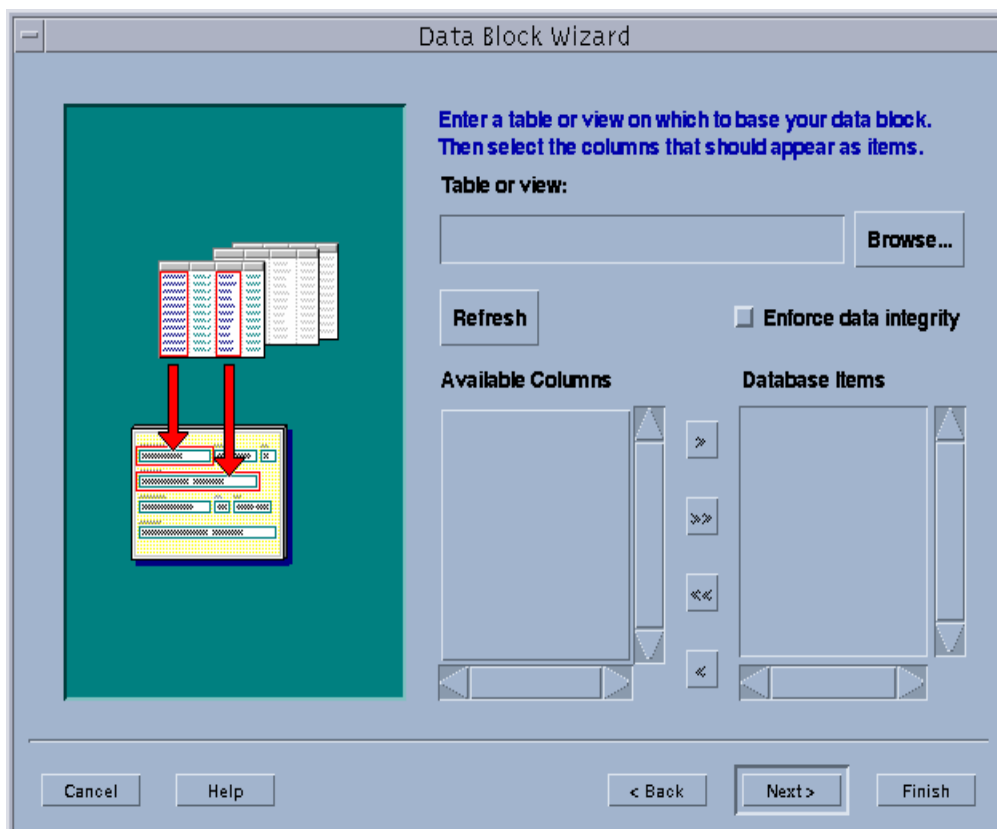
Select the **Data Block Wizard** from the Tools item on the Main Menu at the top of the screen. You will see the Data Block Wizard welcome screen.



Click on the Next button to continue and you will see a new dialogue box like the one below.

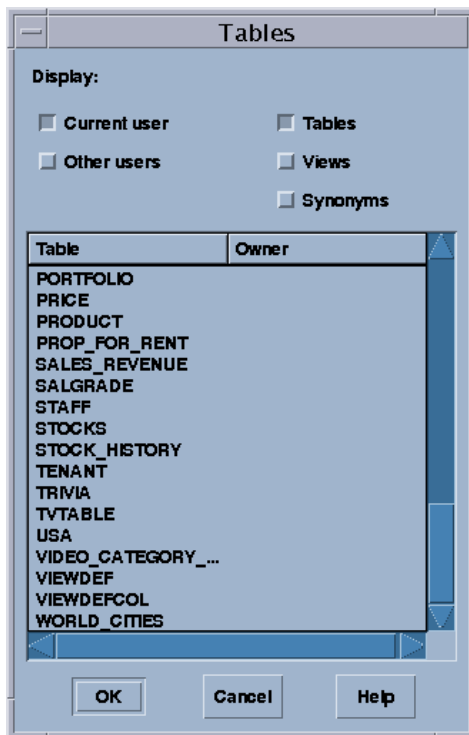


We want to create our form using a table so select *Table or View* and then click on the next button. The next dialogue box will enable you to choose the base table and the columns that you want to appear on the form.

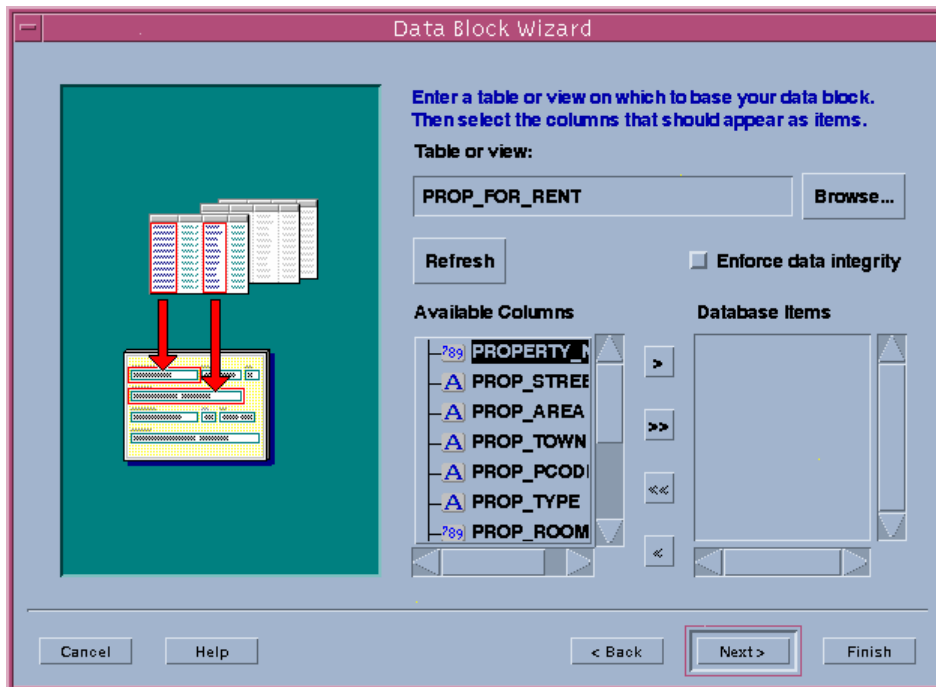


It also allows you to make sure that any integrity constraints set up when the tables were created are enforced, e.g. primary key or referential integrity.

In order to create the form based on the `prop_for_rent` table you can either type in the name of the table and click on refresh, or you can click on *Browse*. If you click on browse this will give a list of all the tables that you have access to and will look similar to the one below, except that you will not have access to as many tables.



Ensure that *Current User* and *Tables* are selected. Highlight the table that you want to use - in this case **prop_for_rent** - and then click on the OK button. You will now see that your property table and the names of its columns are displayed.

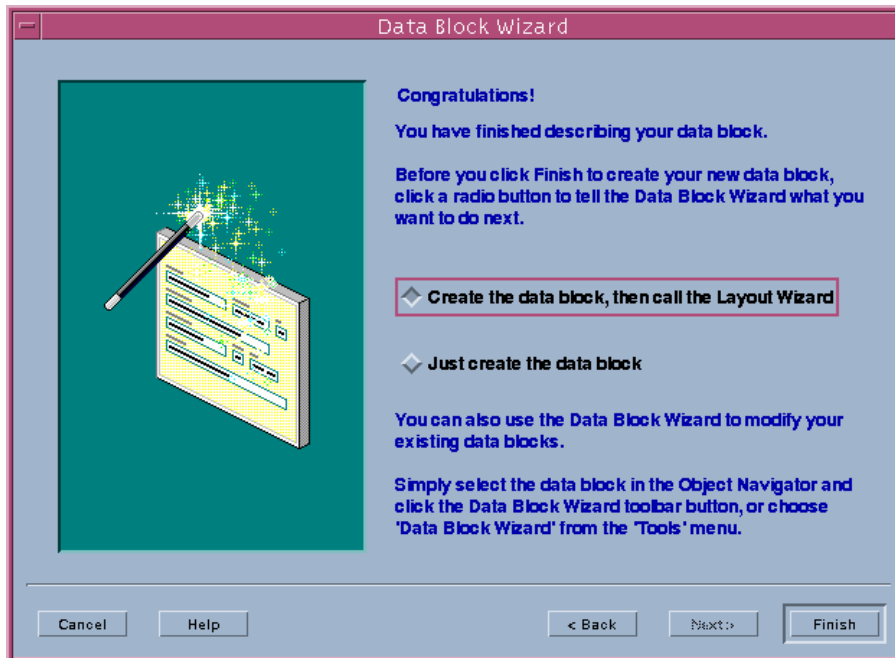


To include a column on the form, highlight the column name and click on the right arrow (>). To include all columns on the form, click the double arrows (>>). The name(s) of the column(s) will move over into the Database Items box and will then be included on the form.

Ensure that you select all of the columns.

Select the *Enforce data integrity* box. This will ensure that any integrity constraints placed on the tables when they were created, such as not null columns, are used in the form.

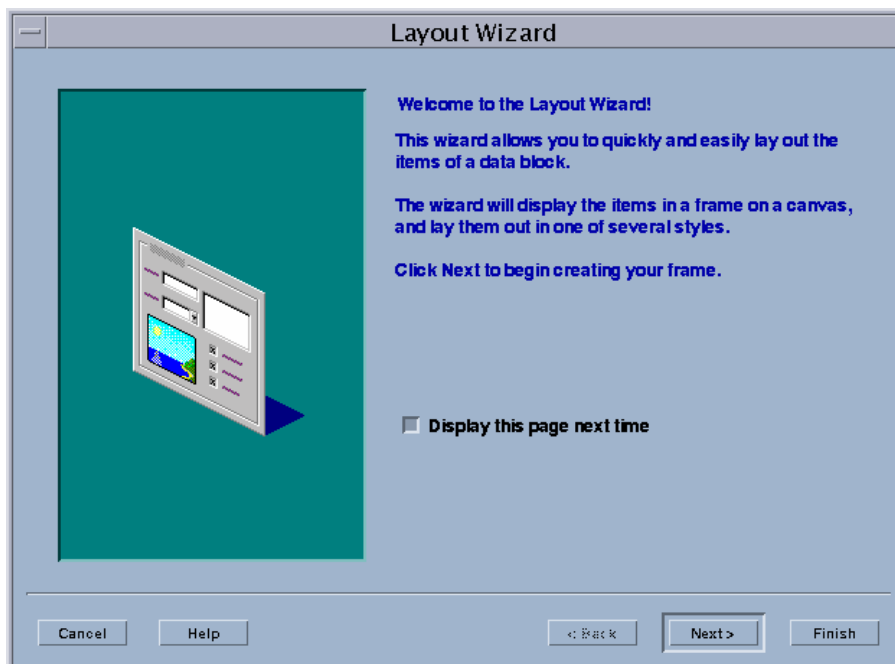
Click on the Next button and the last dialogue box will appear:



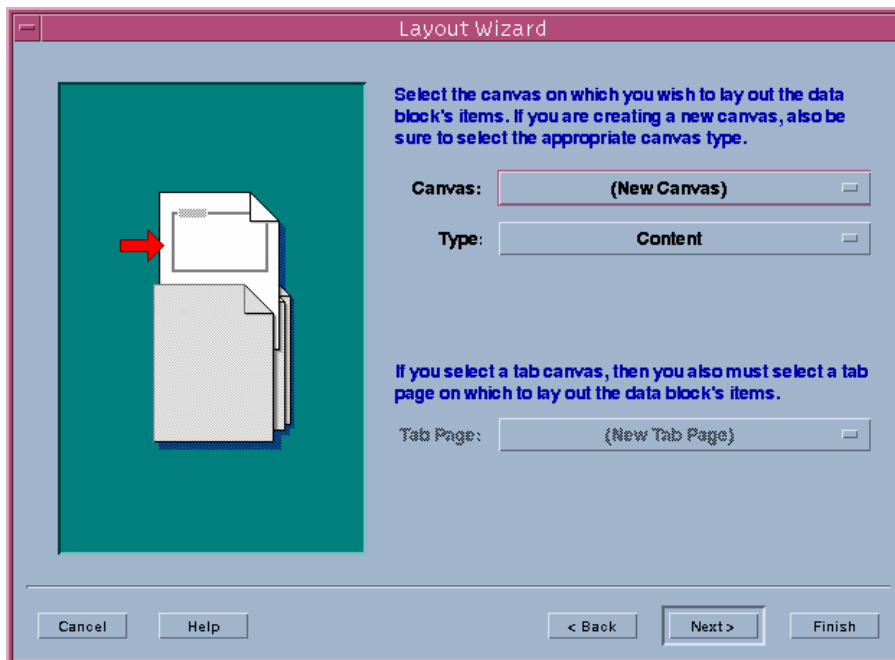
We want to continue to create our form so ensure that *Create the data block then call the Layout Wizard* is selected and click on the Finish Button

STEP 3 - Create the Layout

Since you selected the Layout Wizard in the last screen the first dialogue box in the Layout Wizard should appear.

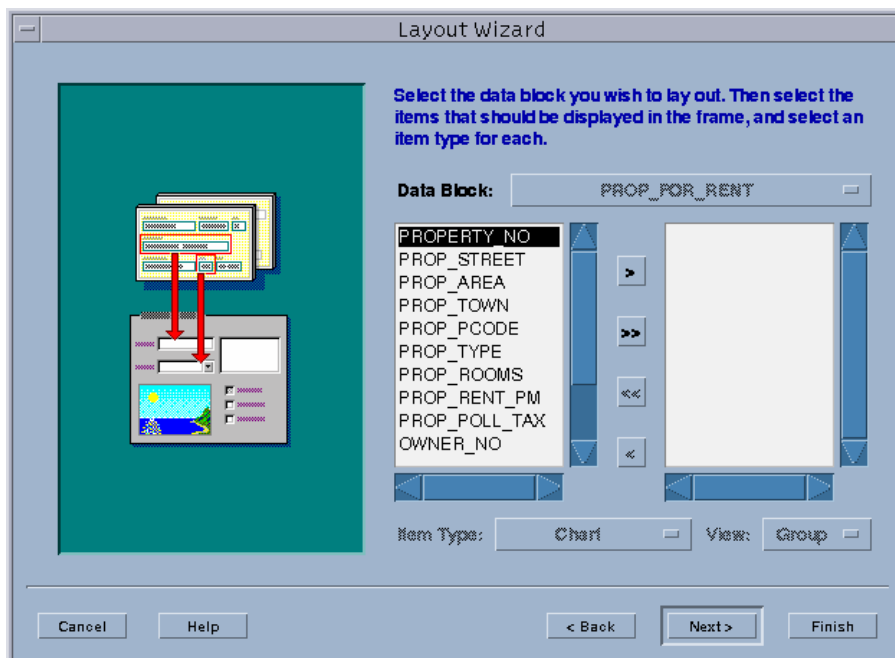


Click on the Next button to continue. You will see a dialogue box like the one below.



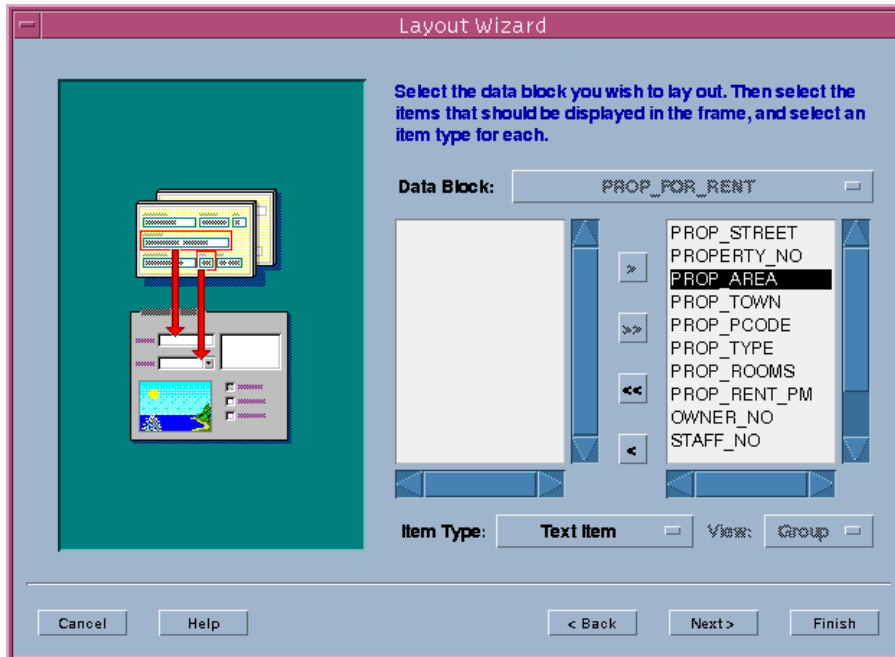
As this is a new form the columns will be placed on a new canvas. A canvas is simply the background on which the items will appear on the form. The type of canvas will be a Content canvas. This is the most common type and displays all of your information in one window.

Click on the Next button. You should see a screen like the one below which will enable you to select the columns that you want to be on the layout.



Now you need to say which Data Block you want to be laid out on the form. In this case it is our prop_for_rent block that we created earlier, and as this is the only data block on the form it appears by default.

Move all of the columns from the 'available' column to the 'Displayed items' column by clicking on the double right arrow and you should have a screen similar to the one below.



Click on the Next button.

Enter a prompt, width, and height for each item. The units for item width and height are Points.

Name	Prompt	Width	Height
PROPERTY_NO	Property No	45	17
PROP_STREET	Prop Street	166	17
PROP_AREA	Prop Area	134	17
PROP_TOWN	Prop Town	134	17
PROP_PCODE	Prop Poode	58	17
PROP_TYPE	Prop Type	13	17
PROP_ROOMS	Prop Rooms	26	17
PROP_RENT_PM	Prop Rent Pm	64	17
OWNER_NO	Owner No	45	17
STAFF_NO	Staff No	45	17
BRANCH_NO	Branch No	32	17

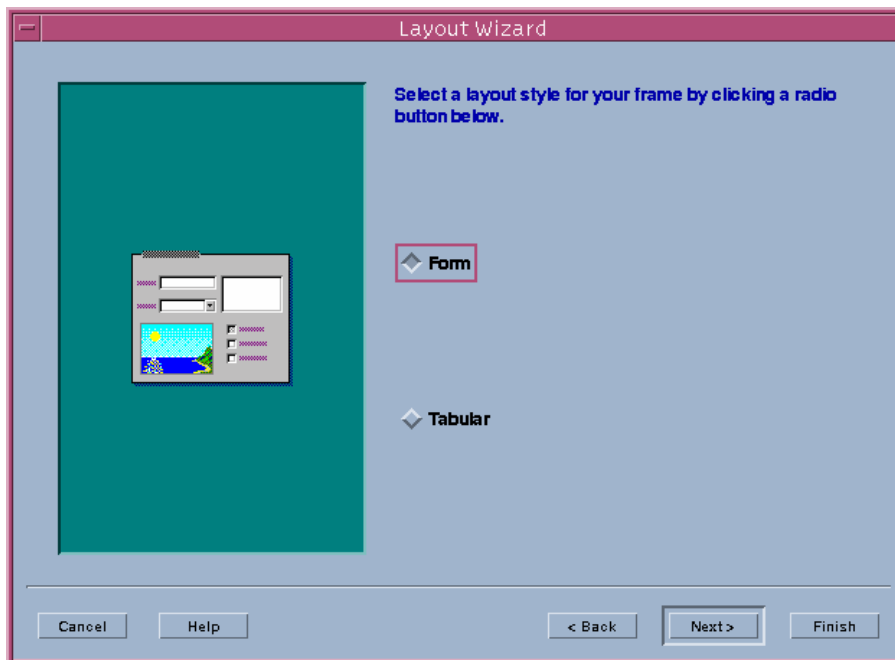
Buttons: Cancel, Help, < Back, Next >, Finish

In this dialogue box you can add a prompt for the column names. Note that the prompt is what will appear as column titles on the screen. For this reason they should be made meaningful.

EXERCISE 3

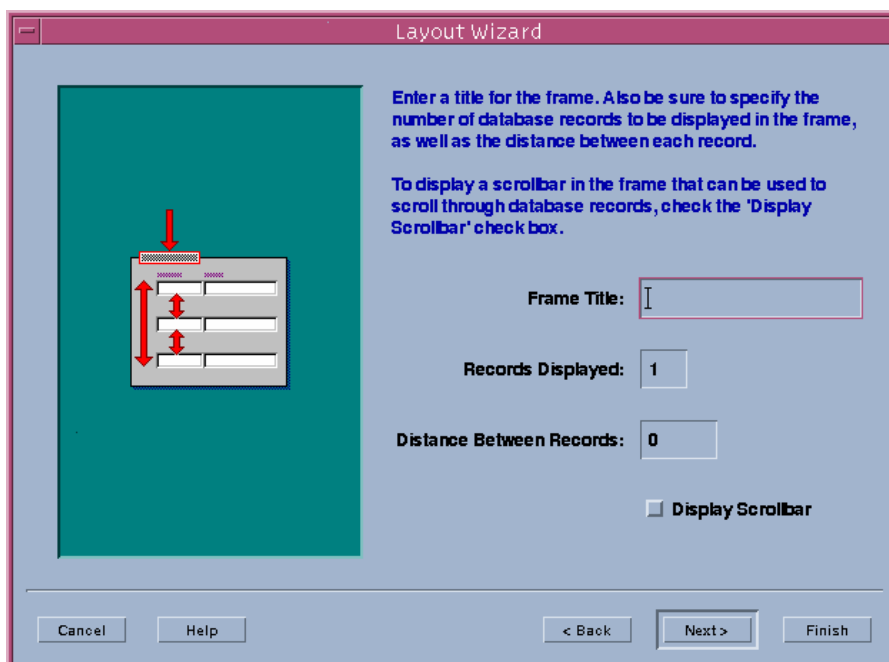
1. Change the Prompts so that they are meaningful
When you have completed this exercise, click on the next button.

On the next screen you can select a layout type for your form.



There are two styles available. You can either choose a **Form** layout or a **Tabular** layout. A form layout places the field names in front of the fields just like they would be on a paper form. Only one record is displayed at a time. A tabular layout on the other hand displays the field names above the fields and displays the fields in a table format - more like a spreadsheet. Usually this would be used to display several records at once.

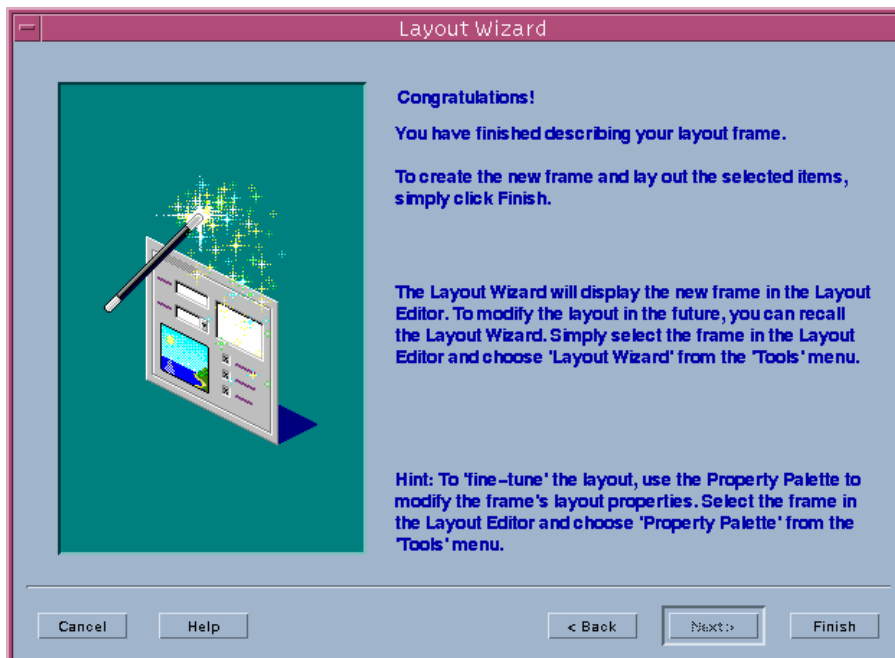
For our example choose the Form layout and then click on the Next button.



In this next dialogue box you can type a name for the frame. The frame is the area around the data block.

Add a suitable frame title, such as Property for Rent.

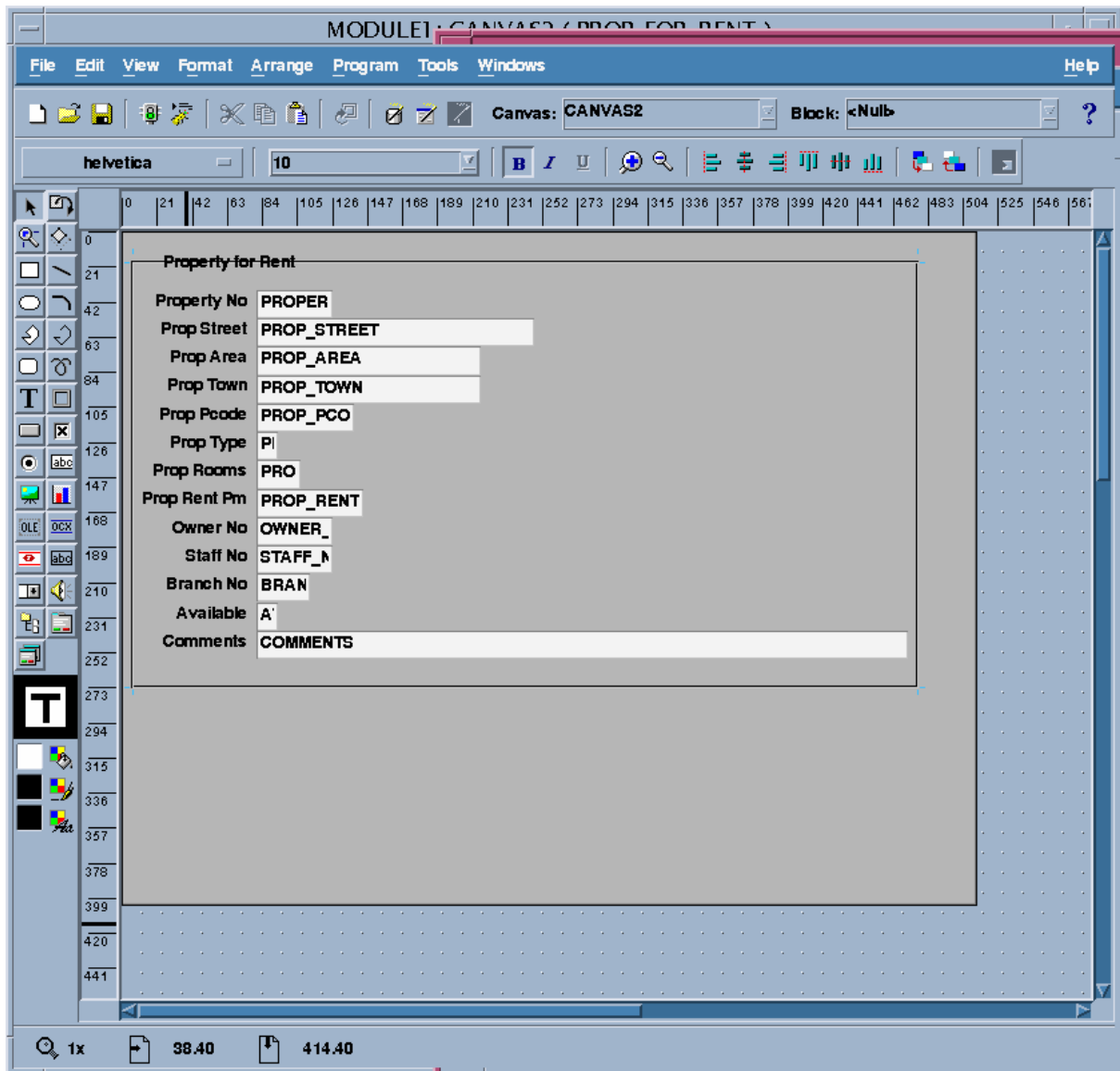
You can choose whether to have a scroll bar or not. A scroll bar is usually required on a **tabular** form so that the records can be scrolled through. It is not usually needed on a **Form** layout. The number of records to be displayed can also be chosen. Usually there will be one record on a **Form** layout and several records on a **Tabular** layout. When you have completed this dialogue box click on the next button and the final dialogue box will be displayed.



Click on the Finish button to create the layout and a new canvas will be created. Once you have finished, the layout editor will be displayed showing the layout of your new form.

2.3 The Layout Editor

The layout editor is used to customise the appearance of forms. It shows a **canvas** with its associated items, and enables you to move items around on the form, to format them by changing font and font size etc. The figure below shows the layout of the form you have just created.



The layout editor can be used to move fields around by highlighting and dragging them. You can add text, titles, buttons, boxes, lines etc to improve the layout of your form and to make it more user friendly. For the moment we will not make any changes. We will examine the layout editor in more detail in a later section.

2.4 Saving and running Forms

2.4.1 Saving a Form

The next step is to save the form that you have just created. You can either choose **[File/Save]** from the Menu, or you can click on the Save icon (the one that looks like a floppy disk) to save your file. Ensure that you save the file in an appropriate place. Call your file **property.fmb**

IT IS RECOMMENDED THAT YOU CREATE A NEW DIRECTORY TO KEEP YOUR FORMS IN AND THEN USE [FILE/SAVE AS] THE FIRST TIME YOU SAVE YOUR FORM.

YOU MUST REMEMBER TO SAVE YOUR FORMS AS YOU MAY BE ASKED TO RETURN TO THEM LATER IN THIS OR OTHER UNITS. YOU WILL FIND THEM MUCH EASIER TO FIND AND RETRIEVE IF THEY ARE IN A SEPARATE DIRECTORY.

2.4.2 Compiling a Form

Before a form can be run it has to be compiled. When you saved your file in the previous section you saved a file that had an extension of **.fmb**. This stores the source code in a binary format. However, you also have to produce a compiled version. This is the executable version - the version that runs. This file always has an extension of **.fmx**. To compile a form select

[File/Administration/Compile File] from the menu

You can also compile and run the form in one step. **To run the form click on the traffic lights icon.** This can be found on the top bar of the layout editor or on the button bar of the Object Navigator. If you do this now you will see the finished version of your form, which should look something like the one here:

To check that your form is working OK run a simple query. In Forms to run a query use the menu and select **[Query/Enter]**. This places the form into query mode. The status bar at the bottom of the screen will display 'Enter-Query'.

Then select **[Query/Execute]** to retrieve all of the records in the table that the block has been based on. The first record in the prop_for_rent table will be displayed. To see the next record, select **[Record/Next]** from the menu.

NOTE: In Oracle Forms there are frequently several different ways to perform the same action. For example, instead of using the menu to run and execute a query you can use the buttons at the top of the form – the buttons to use are the ones with a question mark on them

Enter Query is this button:



Execute query is this button:



It does not matter which method you choose. Use whichever you prefer.

Now that you have saved and compiled, and run your form you need to close it. To close your running form select **[Action/Exit]** from the menu. This will take you back to the Object Navigator. In the Object Navigator, highlight the name of the form and select **[File/Close]** from the menu. This will completely close your form.

You can have several forms open at the same time in the Object Navigator. However, when you first start using forms you will find it much less confusing if you save and close each form when you have finished with it.

EXERCISE 4

1. Put the Unit to one side and see if you can create a default form based on the lease table without looking it up.
Save the form.
Call it **lease.fmb**.
Compile and run it. Remember to close the form when you have finished.
 2. Which statement describes the main difference between your *lease.fmb* file and your *lease.fmx* file?
 - a. The FMB file is the form definition stored as a text file and the FMX is the previous version of this text file.
 - b. The FMB file is the form definition stored as a source code file and the FMX is the previous version of this source code file.
 - c. The FMB file is the form definition stored as a binary file and the FMX is the form definition stored as an executable version of this file.
 - d. The FMX file is the form definition stored as a binary file and the FMB is the form definition stored as an executable version of this file.
- N. B. Check your answer with your tutor, to ensure that you are correct.

2.5 Basic Concepts

Now that you have created a basic form, we will have a look at it in more detail.

2.5.1 Items

Items on a form are the objects that allow users to interact with the form. An item can be a button, a field on a form (called a text item in Oracle Forms), radio groups, a check box etc.

EXERCISE 5

If you have closed your **lease.fmb** form you will need to open it again. There is only one kind of object on this form. What type of item is it?

All items on a form have **properties**. Properties define what an item looks like, such as its width, its font, its font size etc. Properties can be changed to adjust how they look and how they behave. For example, you may wish a field to stand out from the rest and so could change its colour. You might want to display the value of a field on the form but might not want a user to be able to change it, so you could set its property to be a display item only. In forms there are often several ways of doing the same thing. For example, the properties can be changed through the Property Palette, through the layout editor or by using a program. We will use all of these in this and later units.

2.5.2 Canvases

A canvas is the object that we 'draw' the objects on. This is where you position fields, buttons etc. You created a canvas when you created your **property** form. You must have a canvas in order for items to be seen on a form.

2.5.3 Windows

A window is simply the part of forms that contains the canvas. Windows have properties, which can be changed, such as their size.

2.5.4 Blocks

A block is a logical container of items. It does not have physical characteristics such as position or height. It is just a way of logically grouping things together. When you

created your property form, you used one block. There are two types of block that can be used: **base table blocks** and **non-base table blocks**. Base table blocks are based on a database table. For example you based the block in your form on the prop_for_rent table. However, you can also create blocks that are not based on a database table. For example, you might want to have some buttons on your form to save or exit the form. These would be placed in a non-base table block.

EXERCISE 6

1. What is an item?
2. What is a base table block?
3. Which object on a form contains items?
4. If you want an item to be visible to a user you must place it on a canvas. True or False?
5. Which one of the following is not a physical object?
 - a) Buttons
 - b) Check boxes
 - c) Items
 - d) Blocks
 - e) Windows
- 6) How can you change properties?
 - a) Through the Property Palette
 - b) Through the Layout Editor
 - c) Through a Program
 - d) Through the Canvas

3 Manipulating data using oracle forms

3.1 Introduction

This section will explain how you can use ORACLE Forms to retrieve, insert, delete and update data. Every Oracle Form has default capabilities for performing these actions and these capabilities will be examined in this section through the use of Query by Example.

Throughout the examples used in this section, we will use the **property** form that you created earlier.

3.2 Objectives

When you have completed this section you will be able to:

- retrieve data using Forms
- insert data into tables using Forms
- update data using Forms
- delete data from tables using Forms
- list the advantages and disadvantages of SQL data manipulation statements and manipulation of data through Forms

3.3 Query By Example - Simple Queries

First, if you have closed it, you will need to load and run the form that you created earlier – this should have been called **property.fmb**. Otherwise, just run your form. To load a form select **[File/Open]** from the Main Menu and then select the form that you require. Then run it by clicking on the traffic lights icon.

When you first run the form no records appear until you query the database. When you run a query using SQL*Plus you type in the query directly. You state which columns you want from which tables. In a Forms environment you run queries by using Query By Example (QBE). When you use QBE the name of the table is already determined by the name of the table that has been used to create the form. The columns are also fixed by this. Since these two elements are fixed, all you need to do is specify the criteria to be used in retrieving data.

In Forms, to query data you must first place the form in *query mode*. To do this Select **[Query/Enter]**, or use the 'Enter Query' button. Once you are in this mode you can enter some example data in any of the fields on the form as criteria.

For example if you wanted to display all properties which are detached you could enter a 'D' in the Property Type field. Then run the query by selecting **[Query/Execute]**. The first record will be displayed, to see further records, you can click on the Next Record' button.

Try this using your property form.

All of the normal SQL operators can be used in Forms. For example you could find all monthly rents which are greater than £400.

Think about how you would do this and then try it.

You can query more than one field at once. For example, suppose we wanted to retrieve a list of properties which were detached and which had a rent greater than £400. You can probably already guess how to do this.

Try it. You need to type > 400 in the prop_rent_pm field AND type 'D' in the prop_type field. Make sure that you use a capital D.

3.4 Query By Example - Complex Queries

As you can see, Forms provide a quick and easy way of executing queries. However there are some types of query which are a little more difficult, and so cannot be entered into a Form in quite the same way as in the examples above.

3.4.1 Queries using OR

If we wanted to select a list of all properties where the rent was more than £400 **OR** where the prop_type was 'D' we would use the following SQL query

```
select *  
from prop_for_rent  
where prop_rent_pm > 400  
or prop_type = 'D';
```

However we cannot type > 400 into the prop_rent_pm field and 'D' into the prop_type field because this would give us an incorrect result. If we did this we would get the result of the previous query, i.e. rows where both conditions were true. We want to retrieve rows where either condition is true. So how do we get round this?

In order to build a query that uses an OR condition, you have to insert what is called a **placeholder variable** in the fields that you would use in the where clause.

However, before we can do this, we need to expand the size of the prop_type field so that it is large enough to accept our query conditions. To do this, go to the layout editor. (Select it from the Tools menu in the Object Navigator). In the layout editor expand the size of the field by dragging it, then double click on the prop_type field and you will see a properties palette. In the list of properties find the Query Length and change it to 5. You will meet the properties palette in more detail later in the Unit. Close the properties palette.

Now we can perform our complex queries:

STEP 1

Select **[Query/Enter]** as before

Then type in the placeholders as shown below.

The screenshot shows a window titled "WINDOW1" with a menu bar (Action, Edit, Query, Block, Record, Field) and a toolbar. Below the toolbar is a form titled "Property for Rent". The form contains the following fields:

- Property No
- Prop Street
- Prop Area
- Prop Town
- Prop Poode
- Prop Type (containing the placeholder :a)
- Prop Rooms
- Prop Rent Pm (containing the placeholder :b)
- Owner No
- Staff No
- Branch No
- Available
- Comments

The bottom status bar displays "Record: 1/1", "Enter-Query", and "Insert" buttons.

The placeholders have been placed into the two fields that we are going to use for our query. They are called **:a** and **:b** to distinguish them from each other.

STEP 2

Select **[Query/Execute]**. This will result in the QUERY/WHERE window being displayed. Type in exactly the conditions shown below and then select the O.K. button.



Notice that this is similar to the WHERE clause in SQL except that you do not use the word where, and you do not need the terminating semi-colon.

The placeholder variables can be in upper or lower case - the query will work using either.

Click on OK and the query will be executed.

It is also important to note that in this example we have used two variables (:a and :b), and so they have been given different names to distinguish them from one another. If you only had one variable, then you would only need to use the colon on its own. For example, you could place : in the prop_rent_pm field, and then execute a query to display the Query/Where Box. The query would be

: < 200 or :> 400

3.4.2 Queries using ORDER BY

The **order by** clause in SQL is optional and the words **asc** or **desc** are used to determine whether the sort will be in ascending or descending order. For example

```
select *  
from prop_for_rent  
order by prop_rent_pm desc;
```

will place the output in descending rent per month order.

You can do the same thing using the QUERY/WHERE box that you have just seen. You would place a colon in one of the screen fields (it doesn't matter which one because all you want to do is to display the Query/Where box), execute the query and enter the order by clause into the box. The next screen gives you an example. If you want to order the records in **desc** order, then just add this to your query.



The query/where box is used for the following SQL operators:

OR
Between
Order By

EXERCISE 7

Use the lease form that you created earlier for these exercises

1. Retrieve all properties where the lease has been paid for by cash. This will be stored as 'C' on the table.
2. Retrieve all leases where the deposit is more than £100 and the deposit has been paid.
3. Retrieve all leases that ended in 1997.
There are three ways of doing this - though some are more efficient than others. Try to find all three ways.
4. Retrieve all properties that are still leased.
5. Retrieve any leases that cost less than £500 a month or that have a deposit that has not been paid.
6. Retrieve all of the records in the lease table but ensure that they are ordered by the starting date of the lease.

3.4 Inserting Records

Using SQL or ORACLE Forms, data can be inserted into a new empty table, or into a table that already has data in it. We will add data to our existing tables.

Using SQL you have to write an insert statement for every row. This can be very time consuming and is prone to errors of typing. A popular use of Forms therefore is to insert data. Forms can be in one of two modes. By default the form is in insert mode. This means that you can start to enter a record. If there is a record already displayed on the screen you must clear it before entering a new record. The steps to insert a new record are given below. We will perform these steps in the next exercise.

Step 1

Select [**Record/Insert**] from the menu or use the button at the top of the form.

Step 2

Type in the details of your new record

Step 3

When you have typed the details, you need to save the record. Do this by selecting [**Action/Save**] from the main Menu.

EXERCISE 8

1. Using the **property.fmb** form that you created earlier, insert a new record of your own choosing into the prop_for_rent table. Do not forget to save it.
2. Use a query to check that the new record has been inserted.
3. Write down TWO methods of placing data into a database table in ORACLE. Give advantages and disadvantages of both methods.

METHOD 1

ADVANTAGES

DISADVANTAGES

METHOD 2

ADVANTAGES

DISADVANTAGES

3.5 Updating Records

Updating records in Forms is quite simple. First you need to retrieve the record that you want to update. When the appropriate record is displayed change whichever fields need updating. Then save the change by selecting **[Action/Save]**

EXERCISE 9

1. Property No 1007 has been given the wrong rent in error. The rent should be £350. Use Forms to update the prop_for_rent table in order to correct this. Do not forget to save your change to this record.

3.6 Deleting Records

Using ORACLE Forms to delete records is much simpler than SQL and more foolproof. Several keystrokes are required and although this may sometimes seem cumbersome it minimises the possibility of deleting records without thinking. To delete a record the sequence of actions is as follows:

- a) Select **[Query/Enter]** from the menu and type in selection criteria to find the appropriate record to delete.
- b) Select **[Query/Execute]**
- c) Select **[Record/Remove]**
- d) Select **[Action/Save]** to commit the changes to the database

EXERCISE 10

1. Delete the record that you created earlier.
2. Use a query to prove that it has gone.

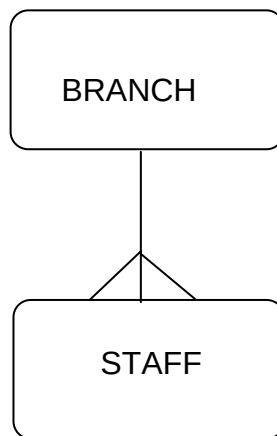
4 Creating Master/Detail Forms

4.1 Introduction

This section will explain how you can create Master/Detail forms. In the first unit on SQL you saw how tables could be joined together. In a business environment the idea of the

Master/Detail relationship is an important one. For example, customers often buy many items, they make many payments etc.

So far you have created forms which only display one record at a time. Master/Detail forms can be created using the same techniques as those used to create default forms. Master/Detail forms deal with one to many relationships between two entities. For example, in the Property Case Study, there is a one to many relationship between the branch entity and the staff entity, i.e. a branch contains many staff. If you look at the Data Model in the Appendix you will see a one to many relationship between the **branch** and the **staff** tables:



When we create a form to deal with this relationship we need to show ONE record for the branch and MANY records for staff. To represent this relationship on a form, we have to use what is called a master/detail form.

To do this we have to create two blocks. One for the branch (the master) and one for the staff (the detail). If a form is defined as a master/detail form Oracle Forms automatically creates all of the program statements that are needed to keep the two (or more!) required blocks in step e.g. to display only the staff who work in the branch. These program statements are called triggers, and you will meet these in a later Unit.

The basic master-detail form that we want to create is shown below.

WINDOW1

Action Edit Query Block Record Field Help

Branch No Br Street
Br Area Br Town
Br Poode Br Telno

Staff No	Branch No	Staff Surname	Staff Forenames	Staff Street	Staff Area	Staff Town	Staff Poode	Staff Telno

Record: 1/1 Insert

4.2 Objectives

When you have completed this section you will be able to:

- Understand the purpose of a master/detail form
- Be able to create master/detail forms

4.3 A Master/Detail Form

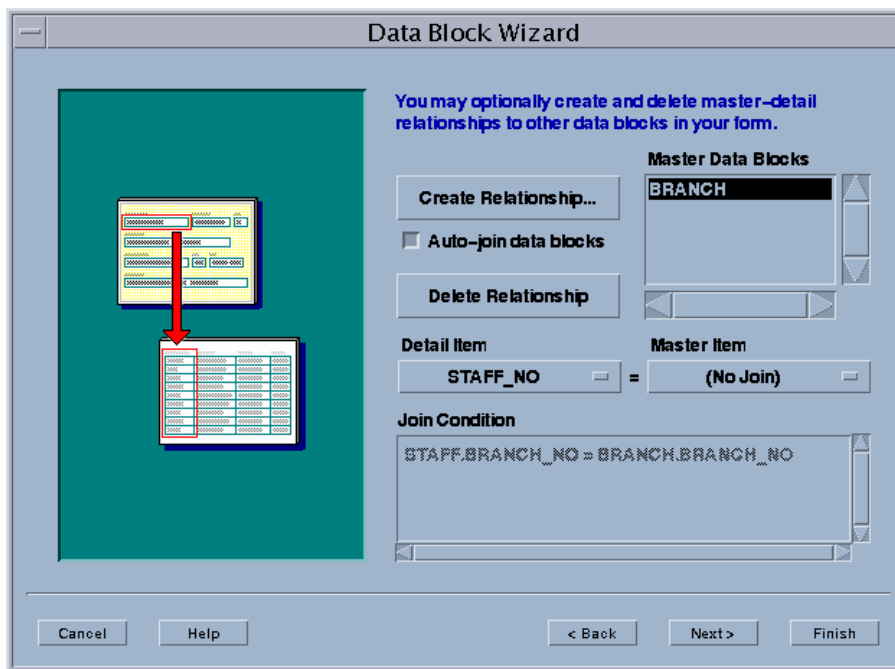
The steps needed to create a master/detail form are as follows:

STEP 1 - Create the Master Block

This is exactly the same procedure as when you created a default form. Start by creating a new form with a data block based on the BRANCH table. Select all of the columns in the branch table. Ensure that you choose a **Form** layout and that only ONE record is displayed. This creates the MASTER block. You can save your form and run a query to make sure that the master part works OK. Now though we have to create the detail block.

STEP 2 - Create the Detail Block

In the Object Navigator, **it is important** to ensure that you have the name of the form highlighted, and then select **[Tools/Data Block Wizard]**. Now we want to create a data block based on the STAFF table. Include all of the columns from the staff table. Because there is already one block on the form, the next step is to create a relationship between the two blocks. The wizard can do this automatically, or it can be done manually. In this case we will use the wizard.



To create the relationship you need to click on the '**Create Relationship**' button. When the relationship has been specified, create the data block by clicking on next and call the layout wizard. Make sure that you place the second data block on the same canvas as the first block. The display of the detail block should be tabular. Call the Frame Staff and display 5 records.

Save your form. Call it **branchstaff.fmb**. Compile the form and then run it. Perform a query on your form. To move between the master and detail parts of the form use **[Block/Next]** and **[Block/Previous]**. You can also use the buttons at the top of the form.

EXERCISE 11

1. Create a master/detail form based on the STAFF and INSPECTION tables. Compile and save your form. Call it **staffinspect.fmb**
2. Query the form to make sure that it works correctly.

5 Customising Forms - The Basics

5.1 Introduction

As you have seen, default forms can be created very quickly. To create them, you specify which database table is required and which columns you want to appear. These default forms can be used to perform queries, and to generally manipulate data. However, they are not necessarily very visually effective and they provide only basic functionality.

Forms can be customised not only to improve their appearance, but also to improve validation and ease of use. For example, you can

- change the order in which the cursor goes to fields
- change the display mode of items
- convert any input automatically to uppercase
- establish range checks
- create help messages for the user
- include programming statements (triggers) to satisfy user requirements that cannot be performed by in-built features.

5.2 Objectives

When you have completed this section you will be able to:

- use the layout editor to improve the appearance of forms
- use the basic facilities of the properties palette

5.3 The Layout Editor

In order to examine some of the facilities of Forms Builder which enable you to customise forms we will use the default master/detail form that you created earlier, called **branchstaff.fmb**.

The layout editor is used to customise the appearance of forms. It shows a **canvas** with its associated items, and enables you to move items around on the form, to format them by changing font and font size etc. I'm sure you would agree that the **branchstaff** form can be improved.

The layout editor has a number of tools that can be used to customise forms. There is a Menu Bar at the top that can be used for saving files, running forms etc. It has a toolbar at the top which will allow you to format text. This includes facilities to embolden text, italicise it etc. It also has a tool palette on the left hand side of the screen which allows you to create new objects, such as new fields, buttons etc. This tool palette gives you an alternative to the Object Navigator for creating objects.

5.3.1 Changing and Adding Text

Boilerplate text includes things like titles. So let us begin by adding new titles for the form and for the frames. You do this using the layout editor.

Ensure that you have opened the **branchstaff.fmb** form. Select **[Tools/Layout Editor]** to display the layout editor. On the left hand side of the layout editor is the tool palette. If you rest the cursor over each one, the name of the relevant tool will be displayed. We are going to add a title to the form so we need the **Text Tool**. Find this on the tool palette and click on it. Then click on the form where you want the title to appear. When you click on the form a box appears for you to type into.

EXERCISE 12

1. Type in the title 'Branch and Staff Details' at the top of the form
2. Save and run your form

NOTE: If you make a mistake when typing the title, you can edit it by clicking on the text tool and then clicking on the text you want to change. A box will appear around the existing text and you can make whatever change is required.

5.3.2 Moving Objects

Objects can be moved in the layout editor by selecting them and dragging them to their new location. To do this use the Select tool from the tool palette to highlight what you want to move, then move it. If you are not sure which tool to use, then remember that you can move the cursor over each tool to obtain a description.

EXERCISE 13

1. Move the items in the master block to a more central location.
To make life easier items can be grouped together before you move them, or you can move the frame and all of the items within the frame will move together.
As part of this exercise find out how to group items.
HINT: You will find it in one of the drop down menus.
2. Make some of the larger fields, such as the surname and address details, smaller by dragging them. Rearrange your form so that you improve its overall layout.

5.3.3 Changing Font and Text Size

Changing the font and size of text can be done quite easily. Highlight the item you wish to change, using the select tool and then choose **[Format/Font]** from the Main Menu Bar. Choose the font and size you require from the list.

EXERCISE 14

1. Change the font of the heading to a different and larger font.

5.3.4 Changing Colours

Change the colour of text by selecting the Text Colour Item from the tool palette. When you click on this a palette of colours will appear and you can choose the one you want.

EXERCISE 15

1. Change the colour of the heading to dark blue.

5.4 The Properties Palette

All objects within Forms, such as items, windows and buttons, have certain properties. You may have noticed that when you expanded some of the objects in the Object navigator a small icon appeared to the right of the + or - sign. If you double click on this a ***properties palette*** for the object will appear. Another way of displaying the properties is to go to the layout editor and double click on an item. The next screen shows part of the ***properties palette*** for the branch_no item.

Property Palette

File Edit Property Program Tools Windows Help

Find:

Item: **BRANCH_NO**

General	
Name	BRANCH_NO
Item Type	Text Item
Subclass Information	
Comments	
Help Book Topic	
Functional	
Enabled	Yes
Justification	Start
Implementation Class	
Multi-Line	No
Wrap Style	Word
Case Restriction	Mixed
Conceal Data	No
Keep Cursor Position	No
Automatic Skip	No
Popup Menu	<Null>
Navigation	
Keyboard Navigable	Yes
Previous Navigation Item	<Null>
Next Navigation Item	<Null>
Data	
Data Type	Number
Maximum Length	4
Fixed Length	No
Initial Value	
Required	Yes
Format Mask	
Lowest Allowed Value	
Highest Allowed Value	
Copy Value from Item	
Synchronize with Item	<Null>
Calculation	
Calculation Mode	None
Formula	
Summary Function	None
Summarized Block	<Null>
Summarized Item	<Null>
Records	

Internal name of the object.

Within this palette properties are organised into categories such as data, navigation, database and so on. These categories can be expanded or collapsed just like in the Object Navigator. You can obtain further information on all of the properties by highlighting the property and then pressing the help key.

5.4.1 Changing properties

Items can be changed one at a time or you can change several items together. For example, you might want to change the font size for three items on the form. To do this you would select all of the items to be changed and then go to the properties palette and change the size. The change would then be reflected in all of the items that you selected. You can specify, among other things, the length of the item, whether it will appear in upper or lower case, you can also change its colour using the properties palette instead of the layout editor etc.

5.4.2 Viewing more than one Properties Palette at once

Sometimes you may wish to view the properties of several items at the same time. To do this use the *freeze/unfreeze* button on the Property Palette's toolbar. This has a pin as its icon - so you 'pin' the properties of an item to keep it in place while you then open another. There is no restriction as to how many property palettes you can have open at once.

EXERCISE 16

1. Go to the property palette for the **staff** data block
Find the ORDER BY property
Type in the name of an item, e.g. staff_surname
Run your form, execute a query and you should find that the records have been ordered by whatever you specified in the properties.
2. You may have noticed when you run this form that the items spread far across the screen. You can use the property palette of the window to add a scrollbar so that you can see all items. Change the appropriate property.
3. Go to the properties of the **gender** item
Find the property to set Case Restriction
Ensure that the gender can only be inserted in upper case.
Run your form and enter some data to test this. Type the gender in lower case - it should automatically be converted to upper case.
4. Change the font of the staff surname through the property palette.
Write down which category of the properties the font is displayed under.

5. Open the properties palette for the staff forename. Pin this so that it remains open. Now open the properties palette for the staff surname. Now that they are both open you can compare them. You should be able to see that they have different fonts (If you successfully completed question 4)
6. Amend the format mask property of the salary so that commas and a decimal point are displayed, e.g. 32,456.00. Remember that you can press the help key when you are on the relevant property in order to get further information. Save and run your form. Execute a query to check if the salary is displayed correctly.

5.4.3 Adding a List Item

A list item is a short list of values that the user can select from. The list can be given labels that are understandable to the user. For example, on the property form that you created earlier, **property.fmb**, there is a field for the user to specify the type of property. The values that are allowed in this field are 'D', 'S', 'B' etc. These single letters are not very useful to a user. In your List item you can specify a full name to be displayed, e.g. Detached, but ensure that the correct value 'D' gets entered into the table. Let us see how this works:

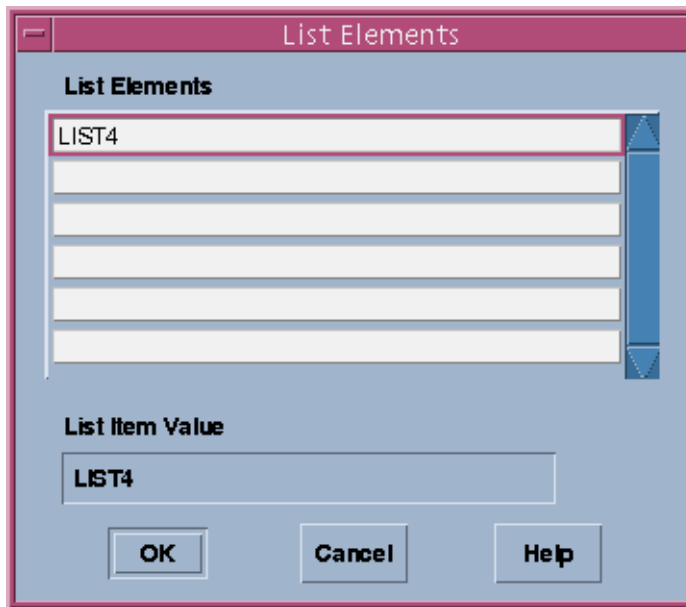
STEP 1

Open your **property.fmb** form. (Make sure that you have closed any other open forms first)

Go to the property palette for the **prop_type** item. In the property palette change the Item Type to *List Item*. You should notice that when you change the type of item, some of the properties change because they are only applicable to the type of item being specified.

STEP 2

Now you need to add the elements that will appear in the list. In the property palette click on Elements in List property and then click the 'more' button and you will see a screen like the one below:



There are two parts to this box. On the first line, under List Elements, add the word 'Detached' (You do NOT need the quotes). In the List Item value, type in a 'D' (Again, you do NOT need the quotes). This is the value that the table holds. If you look at the case study you will see that the database field holds only one character.

The values that we want to allow in our list item are:

- B – Bungalow
- C - Cottage
- D - Detached
- E - End terrace
- F - Flat
- S - Semi Detached

Create each of these in your list elements. You will also need to set the 'Initial Value' property. This is the default value that will be displayed. Set this to 'D'. We want to display the whole word 'Detached' on the screen, however the displayed field currently only holds one character. Therefore we have to increase the size of the field on the screen. Go to the layout editor and increase the size of the field so that it is large enough to display the word(s) we want.

Save your form as **property1.fmb**. Test your form. To test that you have done everything correctly, enter a record and save it. You may need to use SQL to find relevant data.

5.4.4 Adding Hint text

You can create friendly hint messages for the user. For example, you could prompt the user to enter the surname, or you could tell them the format that a date must be in. This

is done through the properties palette (toward the bottom of the list). You simply change the Hint to whatever text you want and change the Auto Hint to True. When the cursor moves to the field on the form the hint text appears at the bottom of the screen.

EXERCISE 17

1. Create some hint text on one or two of the items on your **property1.fmb** form.

5.4.5 Adding Tool tips

Tool tips are another way of providing some help to users. A tool tip appears in a small box below the item when the mouse enters the item.

EXERCISE 18

1. Using the properties palette, add an appropriate tool tip to the prop_type item.
2. Save and run your form. Hold the mouse over the prop_type item to check that your tool tip is there.

5.4.6 Adding Radio Groups

Radio groups are logical containers of radio buttons. Radio buttons present mutually exclusive selections for the user. For example on our property form we could present the prop_type item as radio buttons

EXERCISE 19

NOTE: Now that you are more familiar with using the property palette and the layout editor you will be given a little less help and encouraged to work out for yourself what to do.

Perform the tasks and write down your answers as you go along.

1. Create another default form based on the prop_for_rent table. Call this form **property2.fmb**
2. Change the prop_type item to Radio Group
3. Go to the layout editor and look at the form. Do you notice anything strange?
4. Why do you think that the prop_type item is no longer displayed on the canvas?
5. In the layout editor create some radio buttons.
How do you think you will do this?
You will need to rearrange the form so that there is space to place the buttons.

How many buttons do you think you will need?

6. Each of your buttons need a meaningful name to be displayed on the form.
Which property will you use to do this?
Give the buttons meaningful labels.
7. Each of your buttons will also need to have a meaningful object name for display in the Object Navigator.

How will you do this?
8. In the same way that your list items each had to have a value that will be placed in the database, each of your radio buttons will need to have value.
Which property will you need to use to?
Try it.

9. Save and run your form. You should get an error message. What does the error message say? What do you think you need to do to resolve the problem? Try it.
10. Save and run your form. Does it work? Enter some data to test the radio buttons.
If your form doesn't work, then retrace your steps to try to work out what went wrong.

5.4.7 Adding New Items

There will often be times when creating forms when you might want to display some information from another table. For example when displaying details of the lease, you might also want to include the name of the tenant (i.e. the person leasing the property). To do this you would need to create a new item on the form to display the name, and then create a trigger to select the information from the tenant table. Triggers will be examined in a later Unit. You can also use a new text item to display information that is not from any table at all. For example you could have a new item to display the date on the form. Or you might want to display a calculation. Let us begin by creating a new item to display the date at the top of the form. We want this to display today's date and we don't want the user to be able to change it.

Use your **property2.fmb** form

STEP 1

With your form displayed in the layout editor select the Display Item Tool. We use the display item because we only want to display the date, we don't want to allow the user to do anything else to it.

STEP 2

Position the cursor at the top left hand corner of where you want the new item to appear. Click once to create a default size item.

STEP 3

Use the property palette to define the characteristics for the new item, such as font size etc. Its data type needs to be DATE. You can look at some of the other items on the form to ensure that you use the same font and size. You will need to give the new item a name. So call it *today's_date*.

We also want to display today's date automatically. To do this set the **initial value** property to `$$DATE$$`. This is Forms' default system date.

As the date does NOT come from a table we need to set another property. This property is the **Database item property**. We need to set this to NO. This is VERY important. If you do not do this you will not be able to query or update data in the table and you will obtain an error message when you run your form.

EXERCISE 20

1. When you have added the date field, save and run your form. The date should be displayed as soon as you load up the form.
2. Place your form into query mode. (To do this select **[Query/Enter]** from the menu. What do you notice about the date?
You should notice that the date is no longer displayed. This is because the date is not an item from the table and as Forms prepares for a query from a table it no longer displays the date.
If we want the date to be displayed more permanently we would need to use some code – which will be discussed in a later Unit.

5.4.8 Calculating the value of an item

The value of an item can be calculated automatically through the property palette. This is a simple way of performing calculations without having to resort to programming.

EXERCISE 21

In this exercise we will calculate an annual rent.

1. Use your property2.fmb form. Add a new item to hold the calculated value. Call the item Annual Rent.
Do you think this should be a text item or a display item?

Remember to change the **Database item** property to NO.
In the layout editor, add a label to the item.

2. Go to the properties of the item that you have just created. Find the Calculation mode property and set it to *formula*.
3. Now you have to add the formula to calculate the annual rent. Until you do this you will not be able to run your form. The formula you should enter into the formula property is

`:prop_for_rent.prop_rent_pm * 12`

Notice the way that this is written. First, there is the name of the block (prop_for_rent) with a colon in front of it, followed by a full stop and then the name of the item.

Items on forms are always preceded by the name of the block in order to identify them as belonging to the form. You will meet this again in the next unit.

4. Save and run your form. Test that the calculation is working correctly by running a query.

5.4.9 Creating a List of Values

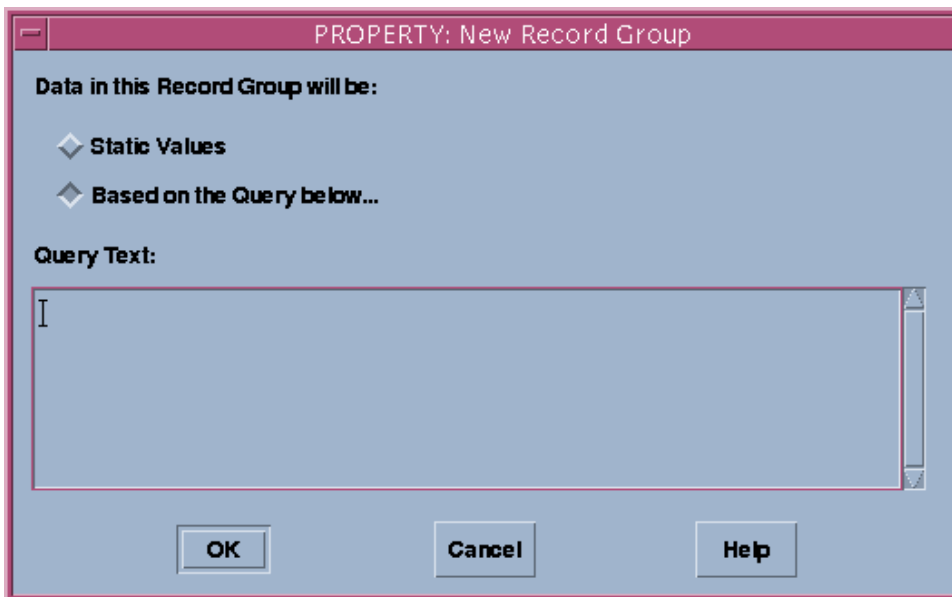
You have already created a list item, which is used for short list and only displays one field. Sometimes you may want to display a much longer list and be able to show values from several fields. For example, it can often be difficult to remember lists of numbers or codes when entering data, so you may want to show the property numbers along with an address. One way to do this is to use a list of values. A list of values is simply a list of valid values from a table and they are helpful in a number of ways. For example, they make data entry easier - the user doesn't have to type, but can just select from a list. Also, they ensure that data entered is valid. A List of Values appears in its own window. Lists of Values are normally attached to a text item (a field). In Forms, a List of Values is based on a record group. A record group is simply a query that returns a set of records. The record group is then used to populate the List of Values.

N.B. Creating a List of Values in Forms can be quite difficult for a new user so please read the following section carefully and don't miss any steps out.

To create an LOV you can use the LOV wizard, or you can create them manually. In this case we will create our LOV manually using the steps below:

STEP 1 – Create the Record Group

Use the **property2.fmb** form. In the Object Navigator, highlight Record Groups, and then click on create (the green plus sign). You will see a dialogue box like the one below:



Now we need to write a query in order to produce the record group. In this case we want to display all of the property numbers along with the street in the LOV. To do this, type in the following query text in the box:

```
select property_no, prop_street  
from prop_for_rent
```

Notice that you do not need the semi-colon. Ensure that the 'Based on the Query below...' is checked. Then click on OK. This will create a Record Group for us to base the LOV on. In the Object Navigator, give your record group a meaningful name such as 'PROPERTY_GROUP'.

STEP 2 - Create a new LOV

In the Object Navigator select the LOV node. You can select it by highlighting the LOV node and using the CREATE icon (the green plus sign). A New LOV dialogue box like the one below will be displayed.



By default, 'Use LOV Wizard' will be selected. In this case you need to select 'Build new LOV manually' and then click on the OK button.

STEP 3 – Create the LOV

In the Object Navigator, highlight the LOV that you have just created and change its name to PROPERTYNO_LOV.

Go to the properties palette for the LOV (by double clicking on the small icon at the side of it), and find the Record Group property. Click on the Record Group property and select the record group name that you have just created from the drop down list..

Next, select the column mapping property, and click on 'more'. You will see a box like the one below:

The screenshot shows a dialog box titled "LOV Column Mapping". It contains several input fields and buttons. The "Column Names" section has a list box with "COLUMN6" and several empty slots. The "Return Item" section has a text box and a "Browse..." button. The "Display Width" section has a text box with the value "1". The "Column Title" section has a text box. At the bottom are "OK", "Cancel", and "Help" buttons.

In this box you will need to type in the name of the column and its return item. The return item is just the name of the item on the form. When the user clicks on the property_no that they want, we would like the property_no to be placed automatically in the property_no field on the form and the street to be placed in the street field on the form. You therefore need to ensure that these are typed into the return items. To do this:

- a) Type in property_no as the first column name
- b) Click on the browse button to see all of the items on the form and select property_no from the list. This places it in the return item. You will also need to set the Display Width of the property_no. Set this to 65. This is the width of the field displayed in the LOV window when it pops up. If you don't do this, the full property number will not be displayed in the LOV.
The Column Title is the heading that will be displayed at the top of the LOV so type in Property Number.
- c) Go back up to column names and type in prop_street as the second column
- d) Click on the browse button and select the street from the list.
Add a column title of 'Street' and then click on OK.
- e) You should not have to change the Display Width of the street.

STEP 4 - Attach the LOV to an item

We want the LOV to be displayed when the user goes to the property_no field. To do this we have to 'attach' the LOV to the property_no. To do this, in the Object Navigator select

the PROPERTY_NO text item. In the property palette for this enter the name of the LOV in the List of Values property.

Save your form under a new name. Call it **property3.fmb**

STEP 5 - Test your LOV

Test your LOV. You can test it by using a query. Select **[Query/Enter]** from the main menu, then use the appropriate function keys to display the LOV (use the help to find out which key or keys this is) Select a property_no from the list and click on OK. The chosen property_no will appear in the property_no field along with the appropriate street. Press **[Query/Execute]**. The details for that property_no will be displayed.

EXERCISE 22

For the final exercise in this unit you will need a copy of a form which contains numerous errors. The form is called **property4.fmb** and is based on the `prop_for_rent` table.

The requirements for the form (in addition to the defaults) are:

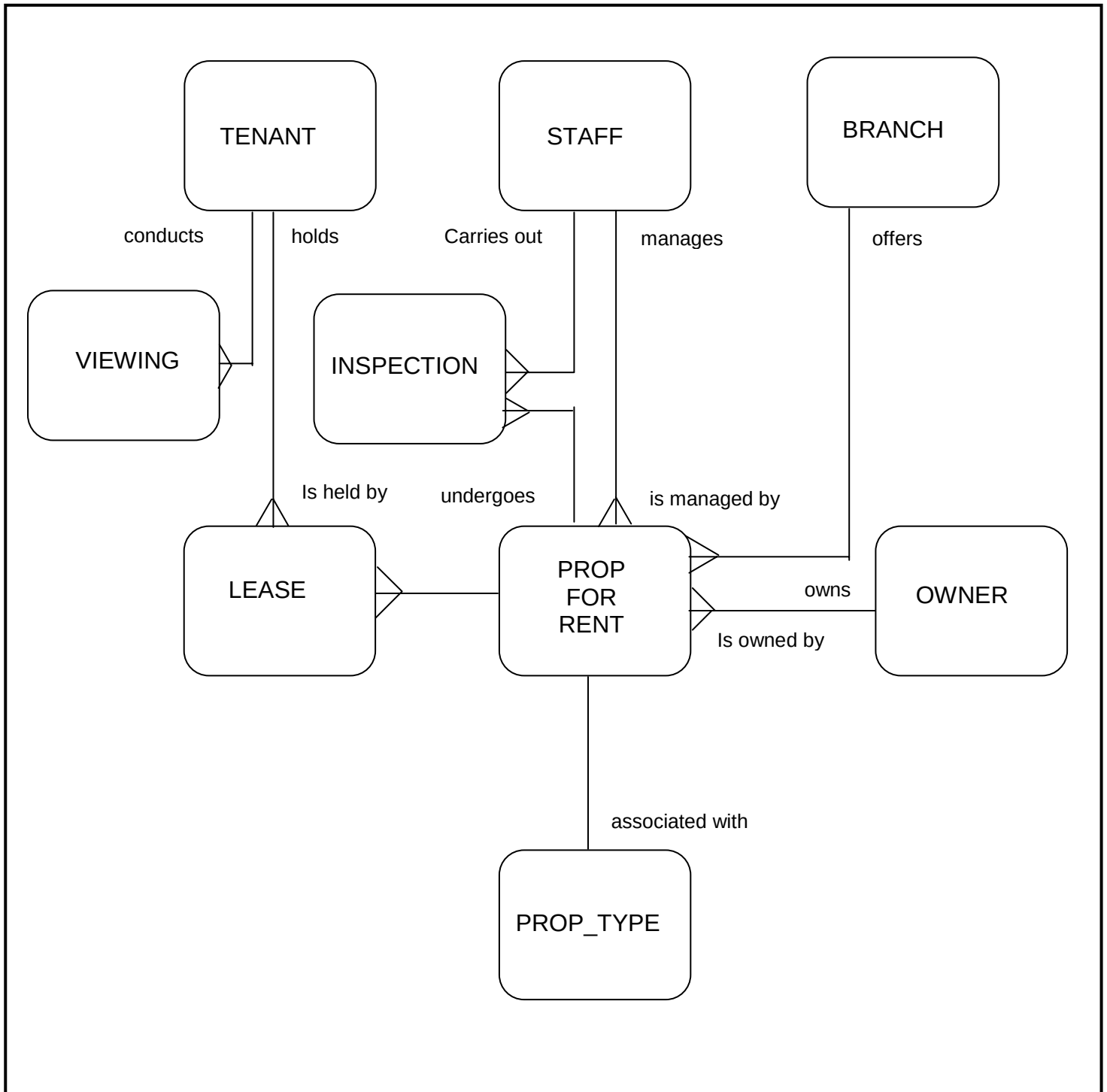
- a) A monthly poll_tax item, which is calculated as the poll tax divided by 12.
- b) Radio buttons for the Available field
- c) A date field to be displayed at the right hand top of the form
- d) An LOV for the property number and street
- e) To be visually effective

The form has been developed by an inexperienced developer and already has some of the required features built in. However, the developer cannot get the form to run. Your task is to correct the form, so that it will run, to meet the requirements and to make the form visually effective.

What were the errors and how did you correct them?

CASE STUDY

PROPERTY RENTAL SERVICES



LEASE

Name	Null?	Type
-----	-----	-----
LEASE_NO	NOT NULL	NUMBER(5)
*PROPERTY_NO	NOT NULL	NUMBER(5)
*TENANT_NO	NOT NULL	NUMBER(5)
RENT_PM		NUMBER(6, 2)
PAYMENT_METHOD	NOT NULL	CHAR(1)
DEPOSIT_AMOUNT		NUMBER(6, 2)
DEPOSIT_PAID		CHAR(1)
STDATE	NOT NULL	DATE
ENDDATE		DATE

VIEWING

Name	Null?	Type
-----	-----	-----
PROPERTY_NO	NOT NULL	NUMBER(5)
TENANT_NO	NOT NULL	NUMBER(5)
DATE_VIEW		DATE
COMMENTS		VARCHAR2(50)

TENANT

Name	Null?	Type
-----	-----	-----
TENANT_NO	NOT NULL	NUMBER(5)
T_SURNAME	NOT NULL	VARCHAR2(25)
T_FORENAMES	NOT NULL	VARCHAR2(25)
T_STREET	NOT NULL	VARCHAR2(25)
T_AREA		VARCHAR2(20)
T_TOWN	NOT NULL	VARCHAR2(20)
T_POSTCODE	NOT NULL	VARCHAR2(8)
T_TELNO		VARCHAR2(12)
T_PREF_TYPE		CHAR(1)
T_MAX_RENT		NUMBER(6, 2)

INSPECTION

Name	Null?	Type
-----	-----	-----
PROPERTY_NO	NOT NULL	NUMBER(5)
STAFF_NO	NOT NULL	NUMBER(5)
INSPECT_DATE	NOT NULL	DATE
COMMENTS		VARCHAR2(50)

PROP_FOR_RENT

Name	Null?	Type
-----	-----	-----
PROPERTY_NO	NOT NULL	NUMBER(5)
PROP_STREET	NOT NULL	VARCHAR2(25)
PROP_AREA		VARCHAR2(20)
PROP_TOWN	NOT NULL	VARCHAR2(20)
PROP_PCODE	NOT NULL	VARCHAR2(8)
PROP_TYPE	NOT NULL	CHAR(1)
PROP_ROOMS	NOT NULL	NUMBER(2)
PROP_RENT_PM	NOT NULL	NUMBER(7, 2)
PROP_POLL_TAX		NUMBER(6, 2)
*OWNER_NO	NOT NULL	NUMBER(5)
*STAFF_NO		NUMBER(5)

*BRANCH_NO	NOT NULL	NUMBER(3)
AVAILABLE		CHAR(1)
COMMENTS		VARCHAR2(150)

PROP_TYPE

Name	Null?	Type
-----	-----	-----
PROP_TYPE	NOT NULL	CHAR(1)
TYPE_DESC	NOT NULL	VARCHAR2(15)

OWNER

Name	Null?	Type
-----	-----	-----
OWNER_NO	NOT NULL	NUMBER(5)
OW_SURNAME	NOT NULL	VARCHAR2(25)
OW_FORENAMES	NOT NULL	VARCHAR2(25)
OW_STREET	NOT NULL	VARCHAR2(25)
OW_AREA		VARCHAR2(20)
OW_TOWN	NOT NULL	VARCHAR2(20)
OW_PCODE	NOT NULL	VARCHAR2(8)
OW_FEE		NUMBER(6, 2)

STAFF

Name	Null?	Type
-----	-----	-----
STAFF_NO	NOT NULL	NUMBER(5)
*BRANCH_NO		NUMBER(3)
STAFF_SURNAME	NOT NULL	VARCHAR2(25)
STAFF_FORENAMES	NOT NULL	VARCHAR2(25)
STAFF_STREET	NOT NULL	VARCHAR2(25)
STAFF_AREA		VARCHAR2(20)
STAFF_TOWN	NOT NULL	VARCHAR2(20)
STAFF_PCODE	NOT NULL	VARCHAR2(8)
STAFF_TELNO		VARCHAR2(12)
STAFF_GENDER		CHAR(1)
STAFF_SALARY		NUMBER(8, 2)

BRANCH

Name	Null?	Type
-----	-----	-----
BRANCH_NO	NOT NULL	NUMBER(3)
BR_STREET	NOT NULL	VARCHAR2(25)
BR_AREA		VARCHAR2(20)
BR_TOWN	NOT NULL	VARCHAR2(20)
BR_PCODE	NOT NULL	VARCHAR2(8)
BR_TELNO		VARCHAR2(12)