Using a Database Program to Create a Solution

Prepared by:
Mike Thomas
theexceltrainer.co.uk

LORMAN®
Published on www.lorman.com - March 2020

Using A Database Program To Create A Solution

Database Design

Microsoft Access is a very large and powerful program with which very complex company wide applications can be developed. Database design can be very difficult and is a subject in its own right, however if you are landed with the problem of creating a database by following the basic rules it is possible to create a solution.

1. **Decide on the requirements of your database**. It will be necessary to talk to possible users to determine their requirements and expectations. Examine the way in which information is collected, stored and retrieved at present. Make detailed notes on the reports that need to be produced.
2. **Design your tables**. Using the information you have gathered break the elements down into subjects such as Purchases, Suppliers or Stock Items. These subjects then become table names. You then decide what information needs to be held about each subject. Each item of information becomes a field.
3. **Create Relationships between your tables**. Examine each table carefully and think about the way in which the subjects are related. Add fields and tables if necessary.
4. **Try your design**. Create queries and forms to make sure you can store and retrieve data in the way that is required. Check for errors in calculated fields and that information is not being duplicated. If necessary redesign your table layout. Continue to add objects to your design and repeat the checking process. When you are satisfied test the application in a 'live' situation.

It is **important** to get the design of your tables and the relationships correct before you continue to design forms and reports and maybe even begin to write functions. If it is necessary to redesign tables any extra design work may have to be done again.

The Wings Tour Company

To help you to understand the elements of Microsoft Access and the way in which a database can be used to provide a solution we are going to create a fictitious scenario. Following through this will help you to understand the tools Access provides.

The Wings Travel Company operate at an international airport and sell passengers seats on flights to international destinations.

They require a computerised system of booking passengers onto scheduled flights to international destinations. The bookings can be made well in advance of the flight date and it is likely that seat prices may change with currency fluctuations and other factors. The passengers price at time of booking will be fixed of course.

The system must be able to make sure that there are enough seats on the flight for all of the passengers booked. There will be three rates of travel available to each destination, namely economy, business and first class.

A passenger list must be able to be printed for each flight along with a boarding pass for each passenger. A listing of scheduled flights will also be necessary which shows current prices.

It would be useful to be able to respond to customer enquiries about information local to the destination country.

We will use Access to design a database that fits the above requirements.

Deciding On The 'Wings' Tables

Now that we have the requirements of the Wings database we can begin to decide on the tables that are needed.

The first subject for a table that comes to mind is of **destinations**, each airport to fly to will have a separate record. Another obvious table is one for **passengers** as the system will have to record the name of each passenger when they make a booking.

Each scheduled flight will also be a separate record and these will be stored in the **flights** table. If the system is to be able to report on the number of seats available it would make sense to have a table holding the types of different **aircraft** in use. Each record could hold the number of seats on each aircraft.

One last table will hold data about **countries**. This example will be used to show relationships of information. An airport can be associated with a country as can a passenger be a patriot of a country, it makes sense therefore to relate both the destination and the passenger tables to the countries table.

We now then have a list of five tables to go into our initial design of the Wings database, these are;

Destinations
Flights
Passengers
Aircraft
Countries

Adding Fields To The Tables

We now need to decide upon the pieces of information that will be held for each record of our tables. Always make sure that the information you wish to store in a table relates directly to the subject of that table. One exception to this is Foreign Keys which will be discussed later.

Use common sense when thinking of fields, for example the flights table will need to have a date and a time so passengers will know when to arrive. It must hold data as to the destination and which type of aircraft will be flying. There will also need to be a unique flight number as an identifier to both passengers and airport staff.

The fields we have deduced for the flights table will therefore be...

Flight Number , Destination , Aircraft , Date, Time.

The design process is served best by sketching your tables on a piece of paper. This helps you to visualise the data and begin to see which tables should be related. You will add the necessary fields as the tables are being created in the following sections.

Creating Access Tables

Starting A New Database

If you have not already done so begin the Access program.

The Access opening screen will be displayed and various Add-ins are loaded, when these are done Access is ready for you to begin. At this point there is no database open. You may now open an existing database or create a new database.

- From the 'File' menu choose 'New Database...'

This opens the New Database dialogue box and allows you to name the new file.

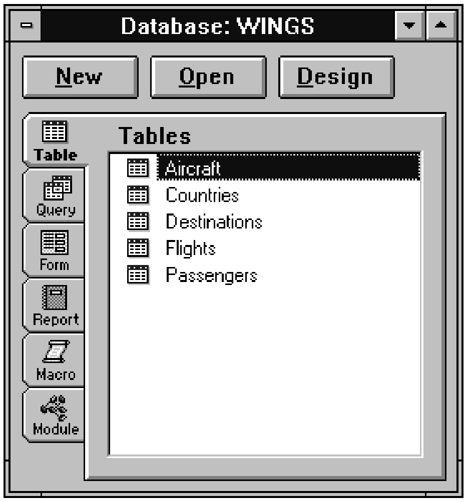
- In the 'File Name' box type WINGS.MDB (MDB is the file extension for an Access database).
- Choose the directory in which you want to store the file.

- Choose OK

The new database file is then created even though at present it is empty.

The Database Window

Your new Wings database is now open and you are presented with the Database window for this file. Note that only one database can be open at a time in Access.



The database window is similar in function to the Windows File Manager. It allows you to display, create new and open existing objects in your database. The window can be maximised or minimised as any other window. Closing the database window also closes your database and returns to the Access opening screen.

Creating A New Table

To create a new table in the database window;

- If the tables section is not displayed click on the ‘Tables’ label in the database window.
- Click on the ‘New’ button in the database window.
- The ‘New Table’ dialogue box will appear
- Choose the ‘New Table’ button, the Table Wizard is discussed later.
- You are then presented with the Table Design Window. This is where you enter information about the fields for each record in your table.

Field Name

A field name can be 64 characters long and include letters, numbers and spaces, however it is practical to keep the field name as short as possible. Shorter names are easier to use in expressions and Access Basic.

Data Type

It is very important to understand the different types of data available. This will allow your database design to be more efficient and increase performance. The following table describes the data type options.

Data Type	Holds	Size
Text	Alphanumeric Characters	1 Byte per character (Maximum 255 characters per string)
Memo	Alphanumeric Characters (Usually large blocks of descriptive text)	Maximum 64,000 characters

Number	Numeric Values (integers or floating point)	1,2,4 or 8 bytes depending on type
Date/Time	Date and Time values	8 bytes
Currency	Monetary Values	8 bytes
Yes/No	Boolean (True/False)	1 bit
OLE object	OLE objects, graphics or binary data	Maximum 1 gigabyte (Limited by disk space)
Counter	Long Integer which Access increments when a record is added	4 bytes

The way you intend to use the data stored is also an important factor when deciding on data type. For example a telephone number should be stored as text, as no mathematical function will be used spaces can be added to simplify reading.

Description

This is displayed in the status bar at the base of the screen when entering data in datasheet view or to a control that is bound to the field.

Moving and Deleting Fields

To delete a field, click on the field selector and press the 'Delete' key or choose 'Delete Row' form the 'Edit' menu.

To move the position of a field click once on the record selector. Then click again and hold, drag the field up or down to the position required.

Creating A Primary Key For The Table

Access performs at its best when a primary key is defined in a table. Indeed you should always specify one field as a primary key.

Setting a primary key automatically creates an index for the field. Using an index speeds up retrieval times for each record.

A primary key must be a unique value, that is no other record in the table can have the same value in that field. Generally a primary key can be an ID number, an abbreviated description or a counter field.

The default view of records in a datasheet or form is in order of the primary key.

If you are relating tables, the one side field of a one-to-many relationship must be a primary key.

The primary key is set by;

- Clicking on the field you want to set as the primary key (if a multiple key is required hold down the CTRL key whilst selecting).
- Click the 'Set Primary Key' button or choose 'Set Primary Key' from the 'Edit' menu.

Saving The Table

When you have added all your fields you must save the table design.

- Choose 'Save' ('Save As' - first time) from the 'File' Menu, or click the 'Save' button on the toolbar.
- Enter a name of up to 64 characters (again the shorter the better).
- Choose OK.

When saved your table name will appear in the tables list in the database window.

Creating The Wings Tables

Wings Table Fields

The fields required for each table are listed below. Create a new table for each and set the bold field as the primary key.

Flights

Flight_ID	Text	6
Dest_ID	Number	Long Integer
Plane_ID	Text	6
Date	Date/Time	
Time	Date/Time	

Passengers

Pass_ID	Counter	
Country_ID	Text	3
Salut	Text	4
FirstName	Text	25
LastName	Text	25
Charge	Currency	
SeatType	Text	6
Flight_ID	Text	6

Destinations

Dest_ID	Counter	
Country_ID	Text	3
Airport	Text	25
City	Text	25
Bus	Currency	
Econ	Currency	
First	Currency	

Countries

Country_ID	Text	3
Country	Text	30
Currency	Text	25

Aircraft

Plane_ID	Text	6
Bus	Number	Integer
Econ	Number	Integer
First	Number	Integer

Foreign Keys

You may be wondering why (for example) the Destinations table has a Country_ID field, when Country_ID is in the Countries table. This is the method by which records from one table are related to another, these fields must be the same data type and size in both tables. When a primary key from one table is recorded in another it is known as a foreign key.

The field name can be different in each table but it is good practice to maintain the same name throughout for clarity. Also when adding tables to a query a join is automatically created when two field names are the same and their data types and size match.

Creating Relationships Between The Tables

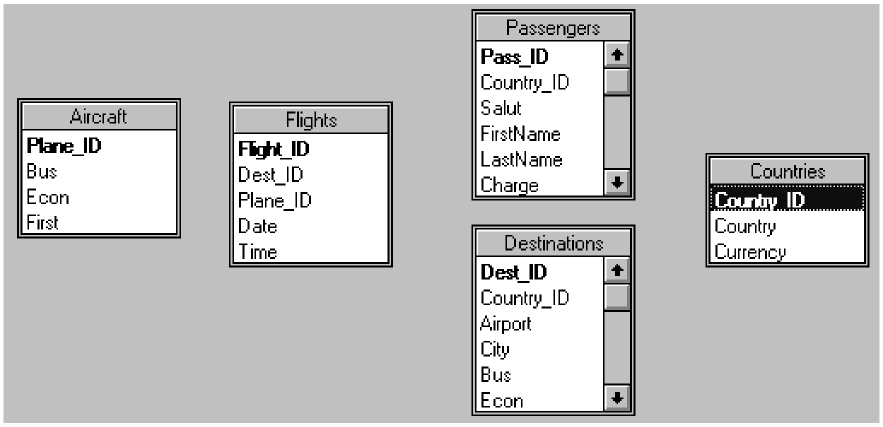
Although we have created the tables and put Foreign Keys in the correct places the tables have not yet been related. To do this you must create and specify the type of relationship that will exist between the tables.

The Relationships Window

To create and edit relationships you must open the Relationships Window. To do this you must have the database window as the current window.

- Change to the database window by choosing ‘Database:’ from the window menu or click on the database window button on the toolbar.
- Open the Relationships Window by clicking on the relationships window or by choosing ‘Relationships...’ from the ‘Edit’ menu. The relationships window will then open. Maximise this if necessary.
- Choose ‘Add Table...’ from the ‘Relationships’ menu or click on the ‘Add table’ button on the toolbar. The ‘Add Table’ dialogue box will open.
- Make sure the Tables option button is selected, and the five tables should be listed.
- To add a table, select it and choose the add button. Alternatively, double-click on the table name in the list.
- You can also select all tables at once by clicking on the first table in the list then holding the SHIFT key and clicking on the last table in the list.
- When all tables have been added click on the ‘Close’ button.

Using the mouse you can click and drag the tables position within the relationships window. Arrange the tables as shown in the diagram below.



Types Of Relationships

One-To-Many

You have seen above that the method of defining the field used in the relationship is to insert a foreign key into a table. As each foreign key is available to us we can now use it to specify the type of relationship the will exist between the two tables.

The most common relationship in a relational database is a one-to-many. This is where a record in the ‘One’ table can have any number of records in the ‘Many’ table which have the same value in the foreign key.

This also specifies that each record in the 'Many' table only has one record in the 'One' table with the same value in the foreign key. It is important to remember this point when creating relationships.

An example of a one-to-many relationship would be a Customers table that has many related records in an Orders table. In our example the Countries table will have many Destinations.

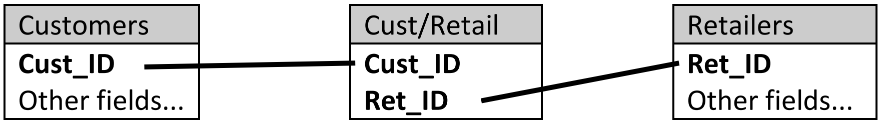
Many-to-Many

A many-to-many relationship is where a record in the first table can have one or more records in the second table, and each record in the second table can have one or more records in the first.

This situation is common place in reality, for example as a customer you buy from many retailers however each retailer has many customers.

You cannot define a many-to-many relationship directly in Access, this is because of the logic involved in managing such a relationship and is not a failing of Access. Instead you must design your tables to reflect the nature of the relationship.

This is quite simply done by creating a third table which sits between the two originals. The fields in this table are normally the primary keys from the two others.



The diagram above shows that by having a 'Cust/Retail' table you can now record the Customers who use each retailer.

One thing to avoid in the 'Cust/Retail' table is that a record is not entered for a customer and a retailer twice. This can be solved by using 'Cust_ID' and 'Ret_ID' as a multiple primary key. This means that both fields make up the primary key, so if a record were to be entered again it would not be saved as it would not be a unique value.

One-To-One

A one-to-one relationship is where a record in each table can have only one matching record in the other. In many situations the two tables could be combined to make one. However there are occasions where it may be necessary to join two tables in this way.

One scenario may be where you want to restrict access to certain information. For example, you may have an employees table showing their telephone extension number and other day to day information. It may be necessary to store their salary details in a separate table to prevent access by casual users (and possibly the employees themselves!).

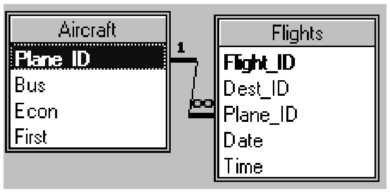
Another use of a one-to-one relationship would be to avoid having many empty records in a table. If only a few employees had company cars it would be wasteful to add 'Company Car' fields to the employees table. Instead create a 'Car' table with the 'Employee_ID' field.

The 'Relationships' Dialogue Box

Now you have all five tables in the relationships window you are ready to specify the relationships that will exist.

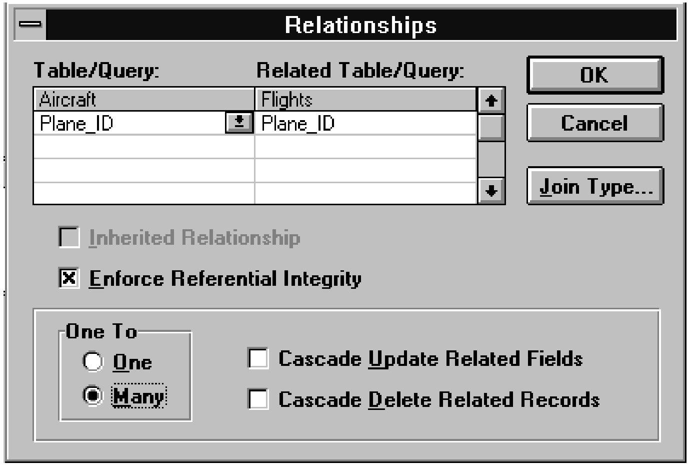
We will begin by relating the ‘Aircraft’ table to the ‘Flights’ table.

- Click and hold the mouse pointer on the ‘Plane_ID’ field in the ‘Aircraft’ table and drag this across to the ‘Plane_ID’ field in the flights table.
- Release the mouse button and the relationships dialogue box will open.
- Check the ‘Enforce Referential Integrity’ option
- Make sure the one-to-many (default) option is set
- Click on the ‘Create’ button.
- Access will create the relationship and display the join line in the relationships window.



The ‘1’ represents the ‘one’ side of the relationship and the infinity symbol represents the ‘many’ side.

To edit the relationship simply double-click on the join line to open the ‘Relationships’ dialogue box. Alternatively, select the join line and choose ‘Edit Relationship...’ from the ‘Relationships’ menu.



Table/Query:	Related Table/Query:
Aircraft	Flights
Plane_ID	Plane_ID

☐ Inherited Relationship

☒ Enforce Referential Integrity

One To
○ One
● Many

☐ Cascade Update Related Fields

☐ Cascade Delete Related Records

Relationship Options

Enforce Referential Integrity

Checking this option ensures that relationships are maintained between the joined tables when you add or delete records.

With referential integrity enforced you must follow these rules;

- You may only add a record to a ‘many’ table when a related record exists in the ‘one’ table.
- You cannot delete a record from the ‘one’ table if matching records exist in the ‘many’ table. (Unless Cascade Delete is set)

Cascade Delete

If the Cascade Delete option is set, deleting a record from the ‘one’ table will delete any matching records in the ‘many’ table.

Cascade Update

If the Cascade Update option is set changing the value of the primary key in the ‘one’ table will automatically change the values held in the foreign key of the ‘many’ table records.

Setting Relationships In The Wings Database

Remember that the Wings database is fictitious and although many of the options that follow will be valid as part of this exercise there may be different priorities in a real situation of this type.

Aircraft to Flights

You have already created a one-to-many relationship between these two tables so lets look at our options.

- You need to enforce referential integrity because a flight must have an aircraft.
- If the Plane_ID changes this should be reflected in the flights table, therefore select to cascade update.
- If an Aircraft record were to be deleted you would not want all the related flights deleted, therefore do not set Cascade Delete.

Flights to Passengers

For this example exercise the Flights table will have many passengers meaning that a passenger record must be entered for each person booking on a flight.

- Drag the ‘Flight_ID’ field on the Flights table to the ‘Flight_ID’ field on the passengers table. Set the one-to-many option.
- Enforce referential integrity as a passenger must have a flight.
- Set Cascade Update as the Flight_ID may change, but the passengers will still be booked on it.
- Set Cascade Delete, as when a flight is deleted so must all the passengers.

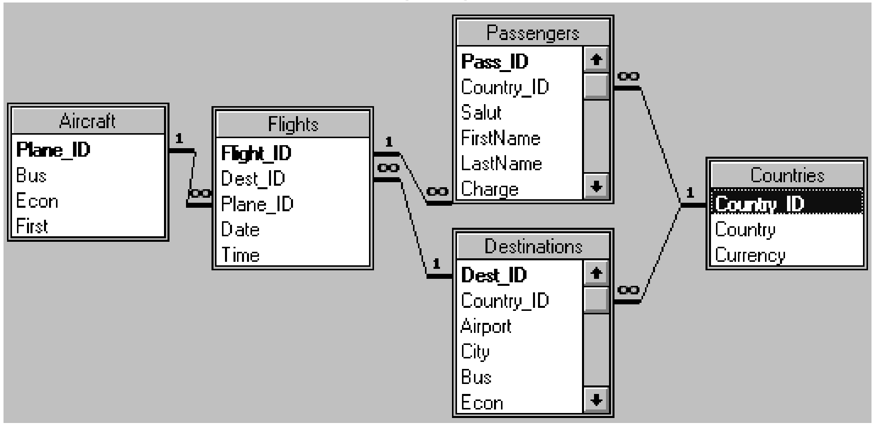
Destinations to Flights

- A flight can only have one destination, but a destination can have many flights. Create the relationship.
- Enforce referential integrity as a flight must have a destination.
- Do not set Cascade Update as DEST_ID is a counter field.
- Do not set Cascade Delete. It is possible for a destination to become invalid, in the case of bad weather or war. Deleting all the flights would remove all passenger information, it would be preferable to change the destination of the flight/s. Then the destination record could be deleted.

Countries to Passengers and Countries to Destinations

- A country can have many patriots and also many airports as destinations. Create the same relationship for both of these tables.
- Enforce referential integrity as both tables need to be referenced to a country.
- Set Cascade Update. As we have only set 3 characters for the Country_ID it may be necessary to change the ID at some time.
- Do not set Cascade Delete. It would be disastrous if a country record was mistakenly deleted, it would remove all related passenger and destination information.

The database window should look like the following diagram.



- Close the relationships window

- Access will ask if you want to save the changes to the layout of the window. Select YES.
- This saves the layout of the relationship window. The relationships you have specified have already been saved.

Adding Data To Tables

Now we have our design established and tables created and related it is time to test the database by entering some information.

Field Properties In Tables

Before we start to enter data it would be wise to think about how we can ensure the data is checked and that enough data has been added.

You can set options for fields by setting the properties section of the table design box. As an exercise you will now set some properties for the Flights table.

- From the database window select the 'Flights' table and click the 'Design' button.

The table design window will open.

As a flight record will only be complete when all the fields have been filled it will be necessary to set all the field properties to 'Required'

- Click on the Dest_ID field.
- In the properties list click in the 'Required' box.
- Click on the 'Drop down list' arrow and choose 'Yes' from the list.
- Repeat the above for the Plane_ID, Date, and Time fields.

A flight can now only be scheduled when all data has been entered.

Display Format

The properties box also allows you to set the way in which data is displayed (not stored) by Access. Setting this property in a table will set the defaults for the fields display when it is bound to a control on a form or report which will be explained later.

In the Flights table set the display formats for the Date and Time fields as follows;

- Click on the 'Date' field.
- In the properties list click on the 'Format' box.
- Click on the drop-down list arrow and choose the 'Medium Date' option from the list.
- Click on the 'Time' field.
- In the properties list click on the 'Format' box.
- Click on the drop-down list arrow and choose the 'Short Time' option.

These fields will now be displayed as 12-May-95 and 12:30.

You can also create custom formats, for more information search Help for 'Format'.

Creating an Input Mask

The Flight_ID field is one in which (for this exercise) we need to be formatted in a special way. That is it will be a letter followed by a dash followed by up to 4 numbers. To achieve this we can use the input mask wizard.

- In the Flights table design window click on the Flight_ID field.
- In the properties list click on the 'Input Mask' box.

- Then click on the '...' button (builder).
- The wizard gives you a list of options, for now click on 'Next>'.
- In the 'Input Mask' box delete the example given.
- Type L-9999
- Click on the 'Try it' box.
- Enter a number for the first character. Access will beep and ignore your keystroke.
- You can change the place holder to your preference. 'Try it' again.
- Choose 'Next >'.
- Select the option to store 'With the symbols in mask'.
- Then choose 'Finish'.

Save the table. Change to datasheet view and enter a flight_ID. Escape to abandon the record as you do not yet have related fields to enter.

For more information on the characters used in the mask search Help for 'Input Mask' then goto the 'InputMask Property'.

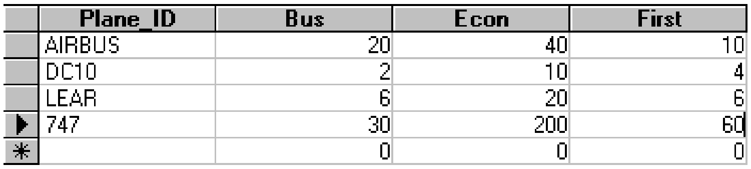
Entering Aircraft Records

You will now enter some data into the Aircraft table so that you will know the number of seats on each type of plane.

- In the database window click on the Tables label if it is not already current.
- Select the 'Aircraft' table and click 'Open' or double-click on the 'Aircraft' table. This will display the Aircraft table in datasheet view.

Each table or form has a blank record which follows the last existing record. If a table or form have no records only the blank record will be displayed. This blank record is the one into which you enter new data.

* The asterix symbol in the record selector denotes the blank record



	Plane_ID	Bus	Econ	First
	AIRBUS	20	40	10
	DC10	2	10	4
	LEAR	6	20	6
▶	747	30	200	60
*		0	0	0

Enter the data in the as shown in the table above.

- In the first field type 'AIRBUS' and press the TAB key or ENTER.
- Then type '20' and press TAB or ENTER again.
- Continue until all fields are completed.

Moving Around Within The Table

When entering data to a table you will generally use the keyboard so we will take a look at navigating using the keyboard.

- Use the Arrow or Cursor keys to move up and down one row at a time and to move left and right one column at a time.
- If you use the TAB key when entering data you can use SHIFT + TAB to move back to the previous field.
- To move to the first field in the table use CTRL+HOME.
- To move to the last field in the table use CTRL+END
- If viewing a large table use PAGE UP and PAGE DOWN to view the next screen full of records.

Notice that when you move into a field the whole field is selected. If you begin to type the existing data is deleted and your new keystroke inserted. To 'Edit' the data in the field you must press the **F2** key. However, if you move the cursor using the mouse the field changes directly to edit mode.

Saving A Record

You do not have to explicitly save a record in an Access table, the record is saved automatically when you move to another record. This is unlike the way a spreadsheet is saved.

When you are editing the current record the record selector displays the pencil symbol.
✎ This means that you have not yet saved the data in the current record.

- In the 'Aircraft' table move to the 'LEAR' record and move to the 'Bus' field.
- Change the value to 7.
- Press TAB.
- Notice the pencil symbol and that '7' is displayed in the 'Bus' field.
- Press the ESC key.

The record selector changes to display the current record symbol ▶ and your changes have been discarded. If you repeat the above but instead of pressing ESC move to a different record (row) your changes will be saved.

You can save changes to a record as you edit by pressing SHIFT+ENTER or by choosing 'Save Record' from the file menu.

NOTE : You cannot disregard any changes you have made by not saving your file, changes are constantly updated in the database file. This is unlike a Word Processor for example that would allow you to CLOSE a file without saving your changes.

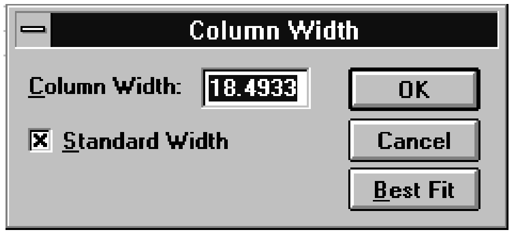
Setting Column Width and Row Height

Using the mouse you can adjust the Column Width and the Row height.

- Move the mouse pointer over the line break between a row or a column.
- The cursor will change to a thick black line with an arrow head in each direction.
- Click and drag this cursor to resize the row or column.
- If you move a column break the column to the left of the cursor will change size.
- If you move a row break the height of all of the rows will change size.

The Column Width and Row Height can also be set through a dialogue box for each option.

- Select a column by clicking on the field name.
- From the 'Format' menu choose 'Column Width..."
- In the dialogue box click the 'Best Fit' button.



Column Width

Column Width: 18.4933

☒ Standard Width

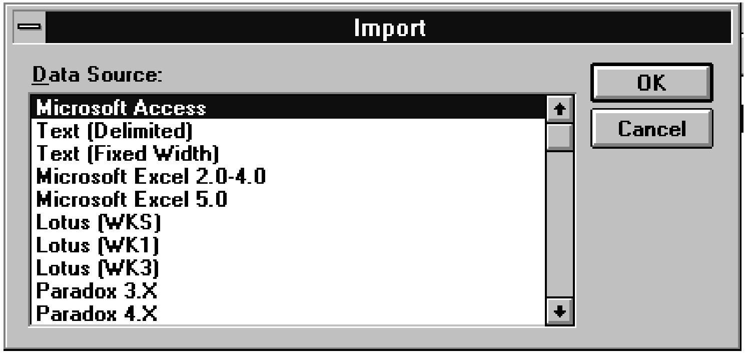
Although any changes you have made to the data stored in the Aircraft table have already been saved your layout changes have not. To save your changes choose 'Save Table' from the 'File' menu.

Importing Data From Other Sources

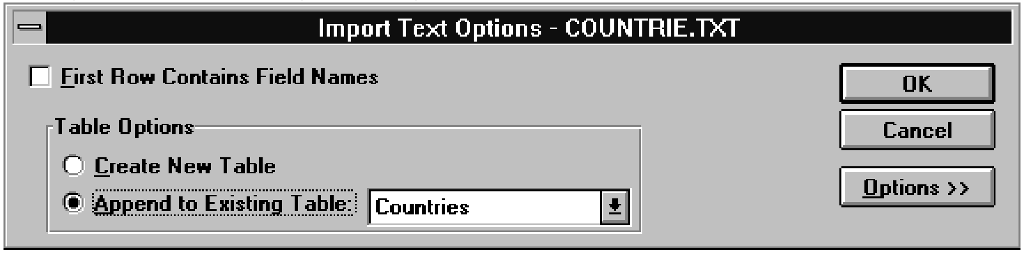
Microsoft Access can import data from many other sources such as other database formats and text formats from spreadsheets and other programs.

You will now import some data into the tables you have created. This is for two reasons, the first to show how referential integrity works, and secondly to save yourself some typing.

- From the database window click on the 'Import' button
- To open the import options dialogue box...



- Choose Text (Delimited) and click OK.
- Then choose the FLIGHTS.TXT file in the C:\EXERCISE directory.
- Select to Append to Existing Table - Flights, and choose OK.
- You should be greeted by an error message informing you that referential integrity will be violated. Cancel the operation and start again. The reason for the error is the Flights table requires a matching record to be in the Destinations table. Because of the relationships specified you will have to import the files in a specific order.
- Repeat the above steps but import the COUNTRIE.TXT file first.


Import Text Options - COUNTRIE.TXT

□ First Row Contains Field Names

Table Options

○ Create New Table

● Append to Existing Table: Countries

- This should import correctly and report no errors
- Then import the DESTINAT.TXT
- Then import the FLIGHTS.TXT
- Then import the PASSENGE.TXT

You may like to ponder for a while on the order of import. The Countries table had to be imported first because both the Destinations and Passengers tables needed a matching record in this table.

Then Destinations could be added, this had to go in before any Flights records that require a Destination. Passengers must still wait as they need a matching record in the Flights table. Therefore the Flights table must be imported before Passengers.

The Aircraft table (as a 'one') table does not come into it as it does not care if any flight records exist or not.

Creating Forms

A Form, as was mentioned before, allows you to view and edit your data one record at a time. It also allows you to add various controls to make the use of your database easier. A Form can be based on a table or a query. You will see next how a form can make data entry a more organised process.

A Form can be created for you by Access or you can create your own blank form and start your design from scratch. You will learn how to do both in the following exercises.

Forms to View More Than One Table

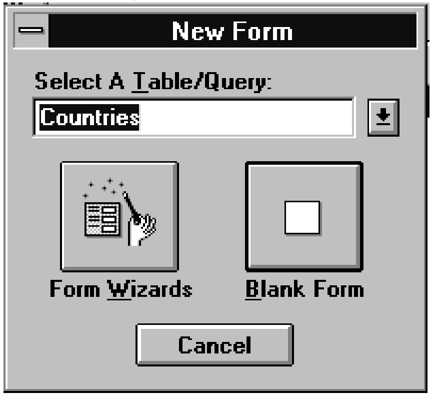
Whilst having a relationship enforced is good for your database design it makes data entry to the 'many' tables very difficult. If you open the 'Destinations' table and add a new airport you will have to remember the three letter 'Country_ID' otherwise Access will not allow you to enter the record. If this does not seem all that difficult imagine taking an order and trying to remember 3,500 customer ID's.

As a solution to this problem you can create a form based on the Countries table which displays only the related records in the Destinations table.

Using A Form Wizard To Create The Countries/Dest Form

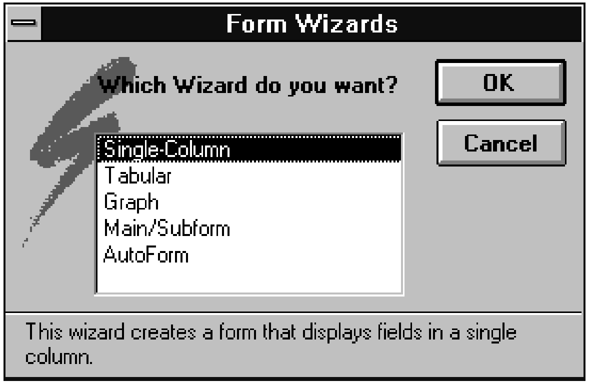
You will now use a Form Wizard to automate the task of creating a form. Whilst Wizards simplify many tasks they only go so far as we will see later on.

- In the database window click on the forms label to open the forms list and click on the 'New' button. Alternatively, choose the 'New Form' button on the toolbar . You can also choose 'New' from the 'File' menu and then 'Form' from the flyout menu.
- Any method will open the 'New Form' dialogue box.



Select A Table/Query: Countries

- Select the Countries Table and then click the large 'Form Wizards' button.



Which Wizard do you want? Single-Column

This wizard creates a form that displays fields in a single column.

You will then have the option of which type of form you wish to create. Clicking on each option will give a short description at the base of the dialogue box. The option you require is Main/Subform.

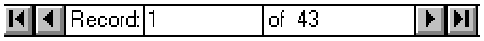
- In the Form Wizards Dialogue box select Main/Subform then click OK.
- You have already specified the Main form as 'Countries' so you now have to supply the form you wish to use as a subform. Choose 'Destinations' from the list.

- Then choose ‘Next >‘.
- You are now asked which fields you want to display on the MAIN form. All three fields are required so add each on to the list.
- Again choose ‘Next >‘.
- You must then supply the fields to display on the SUBFORM. You need to add all of the fields except Dest_ID and County_ID. Dest_ID is a counter field and so can be hidden from the user. As Country_ID links the two forms it will automatically entered and so is not needed on the subform.
- Choose ‘Next >‘.
- Then select the ‘Embossed’ option.
- Choose ‘Next >‘.
- You can then give the form the title ‘Countries/Dest’.
- Choose to open the form with data in it.
- Choose ‘Finish’.
- The form is then created and you will be prompted to save the subform before the wizard can proceed. Name the subform ‘DestSubForm’
- The wizard will then complete the form and it will open in data view.

Entering Data To Multiple Forms

You will now see the Destinations records displayed within the Countries records. Although the Destinations form looks like a table it is actually a form in datasheet view, but more of that later.

Notice the two sets of record navigation bars, one for the Countries form and one for the Destinations records.


Record: 1 of 43

The buttons on the navigation bar move to first, previous, next and last record respectively.

The large bar on the left hand side of the main form is a record selector and is used in the same way as in a table.

You now need to enter a record for India and three airports in that country. You will first need to move to the blank record to allow you to enter data.

- To create a new record for a country first make sure that the cursor is in the main form.
- Choose the ‘New Record’ button from the toolbar ▶*. Or on the main form navigation bar click the last record button and then the next record button to move to the blank record at the end of the table.
- Enter the Country_ID as ‘IND’. Press TAB.
- Enter the Country as ‘India’. Press TAB.
- Enter the Currency as ‘Indian Rupee’. Then press TAB again. This will move you into the destinations form.
- You can now enter the following destination information. Press TAB after each entry.

Airport	City	Bus	Econ	First
Indira Gandhi	Delhi	300.00	245.00	345.00
Dum Dum	Calcutta	280.00	235.00	320.00
Bombay International	Bombay	310.00	285.00	350.00

- To move the cursor back to the main from a subform press CTRL+SHIFT+TAB. Or use the mouse to move the cursor to a field on the main form.
- Save the ‘Countries/Dest’ form.

The records entered to the Countries/Dest form are now saved in the Countries and Destinations tables respectively. To view the airports you have added open the Destinations table.

- From the database window click the Tables label.
- Select the ‘Destinations’ table and click ‘Open’.
- Move to the last records in the table.
- Notice that the Country_ID ‘IND’ has been entered automatically. These airports are now related to the India country record.

Editing A Form Created By A Wizard

Although Wizards make the creation of forms simple they often create a form which needs to be redesigned to make it workable. In the form you have just created you will have had to scroll across the form to be able to view the sub form. In every day use this would become very tiresome. The form therefore needs to be redesigned.

Once a wizard has created a form it is no different to any other and can be opened in design view. A wizard simply automates the tasks you would go through when creating a form manually, and it has to be said that you must understand the elements of form design before you can answer the wizards questions.

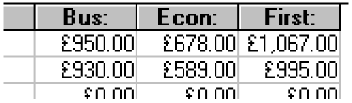
You will now open the Countries/Dest form in design view.

- From the database window click the Forms label to display the forms list.
- Select the Countries/Dest form and click ‘Design’.
- The Countries/Dest form will open in Design view. Maximise the screen if necessary.

The problem with scrolling the form is actually caused by the ‘DestSubForm’ subform. This is due to the datasheet being too wide to view.

To although you can edit a subform from the database window the easiest way to re-design a sub form is from the main form.

- Double-click the ‘DestSubForm’ control in the Countries/Dest form. (Anywhere in the large white box)
- The DestSubForm opens in design view.
- Change to datasheet view by clicking on the ‘Datasheet View’ button on the toolbar.
- As with a table you need to narrow the width of the columns. The ‘Airport’ and ‘City’ columns are okay but the columns showing the costs of the flights could be narrowed. We could reasonably expect the cost of a flight to be under £9999.99 so you can narrow the columns until they will just display this figure.
- As with a table you can drag the width of the columns, but if you want a neat uniform width for costs do the following.
- For each cost field (Bus, Econ, First) change the amount to four figures, i.e. £9999.
- Holding the SHIFT key click on each cost field column selector.
- Each selection should highlight the whole column.
- From the ‘Format’ menu select ‘Column Width...’
- All three columns should now neatly fit four figure sums.



Bus:	Econ:	First:
£950.00	£678.00	£1,067.00
£930.00	£589.00	£995.00
£0.00	£0.00	£0.00

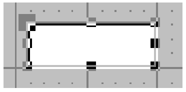
- If you have changed values return them to their originals.
- Now save the form to record your changes.
- From the ‘File’ menu choose ‘Save Form’.
- Then choose ‘Close’ from the ‘File’ menu.

You will return to design view in the ‘Countries/Dest’ form.

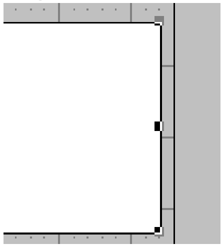
The 'DestSubForm' has now been narrowed but the Control which displays it is still too wide for the form. You need to reduce the width of this so the form will fit in the width of the screen.

Each Control on a form is a separate object, if you want to edit a specific Control you must first select it by clicking once on the control with the mouse.

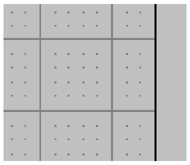
- An object is selected when its Control Handles are displayed as in the picture.



- To change the size of an object you simply click and drag on one of the smaller handles.
- To move the object independently of another click and drag the larger handle in the top left corner.
- Select the 'DestSubForm' control if it is not already selected. (If you click twice and the pointer changes to a text cursor press ESC)
- Scroll the form window so you can see the right hand side of the 'DestSubForm' control.



- Move the mouse pointer over the handle in the centre of the right hand side. When in the correct position the pointer will change to display left and right arrows.
- Hold and drag to the left.
- Re-size the Control to be less then the width of the screen.
- Check by scrolling the window so you can see the left hand side of the form. If necessary repeat the operation.
- When satisfied all that remains is to reduce the width of the 'Detail' section of the form. This is the gridded area used to display controls.



- Move the mouse pointer over the right hand edge of the Detail section.
- As when resizing a column the cursor will change to a thick black line with a left and right arrow indicating you can resize the object.
- Click and drag the edge of the Detail section to the left until it fits on screen.
- Change to Form View to see the changes you have made by clicking on the 'Form View' button .
- Choose 'Save Form' from the 'File' menu to save your changes to the form.

You have now created a form based on two tables and had a brief introduction to the Design View.

Creating A Form to Choose Options From Other Tables

A Wizard can only take your form design so far. If you need to create a form to solve a specific problem you will most likely need to add other types of control to the form. The Form Wizard is limited in this respect, but you can still use it to do some of the work.

Create A Flights Main Form With A Passengers Sub Form

- Using the Main/Sub form Form Wizard create Flights form based on the Flights table.
- Use the Passengers table as the base for the 'PassSubForm' sub form.
- Add all of the fields from the Flights Table.
- Add 'Salut', 'FirstName', 'LastName', 'Country_ID', 'SeatType' and 'Charge' from the Passengers table.

- Choose the Embossed style.
- Name the form Flights.
- Choose to modify the forms design.
- Choose ‘Finish’.

The form will open in design view. Maximise the form window if necessary.

Elements Of The Form Design View

Before you continue take a look at the elements which make up the Form Design View and how you can use them.

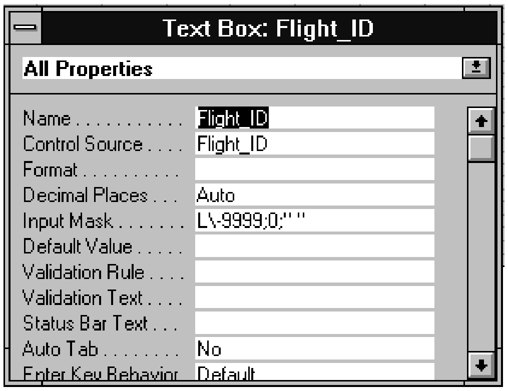
The **Detail** section is the area on the form where each individual record from the underlying table or query is displayed.

The **Form Header** section is an area in which you can add controls also these can relate to the records in the Detail section. A Form Header is useful when the form is deeper than the screen or when the detail section shows continuous forms.

You can also add a **Page Header** to your form. This goes between the Form Header and the Detail section and is used when the forms cover many pages.

- From the format menu choose ‘Page Header/Footer’
- This will display the Page Header and Footer sections.
- Choose again to uncheck the option and remove the sections.
- The same can be done for the ‘Form Header/Footer’ but removing this will delete the text contained.

The **Properties** button opens the Properties box. Every object in the Access database has certain properties associated with it. As an example select the Flight_ID text box and click on the Properties button.



Text Box: Flight_ID

All Properties

Name	Flight_ID
Control Source	Flight_ID
Format	
Decimal Places . . .	Auto
Input Mask	L\-9999;0;" "
Default Value	
Validation Rule	
Validation Text	
Status Bar Text . . .	
Auto Tab	No
Enter Key Behavior	Default

Notice that the Input Mask has been ‘Inherited’ from the Flights table. Also notice that the text box itself is named Flight_ID.

You can choose to view types of Properties by choosing from the list at the top of the Properties box.

One important point to note at the moment is that the property setting applies to the control on the form not to the field in the table. For example you could remove the input mask from this control but it would not affect the way you enter data in the table.

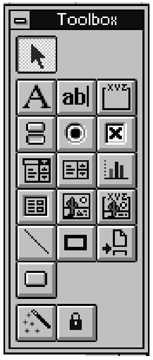
You will learn more about properties as you work through the exercises.

(For a more detailed description of Properties search Help for properties)

The **Field List** button opens and closes a box containing the fields that are available for use on the current form. You can drag a field onto the form using the mouse. If any properties are set in the table for the field they will be inherited when the field is placed on the form.

The **Code** button opens the Access Basic module for the current form. This is where functions and event procedures written in Access Basic can be created and edited.

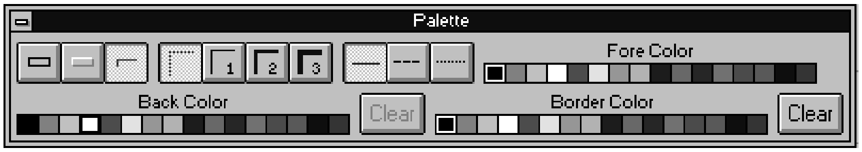
The **Toolbox** button opens and closes the toolbox. You use the toolbox to add the various controls to your forms design. This is also used to set the default properties for the controls. With the property box open click on the control in the toolbox to see its default property list.



The toolbox can 'float' as seen above or be positioned as a toolbar by dragging it to the top of the screen.

For a description of each control hold the mouse pointer over the button.

The **Palette** button is used to open and close the Palette. When a control is selected you can change the colour and appearance by choosing the options in the palette.



The Palette can also 'float' or be positioned as a toolbar.

Ruler. A ruler can be displayed to help position controls. Choose 'Ruler' from the View menu. The ruler is visible when there is a tick shown in the menu. Choose again to remove the ruler.

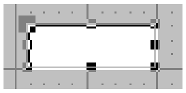
- Clicking on the ruler with the mouse selects all objects in a straight line. Try it.

Grid. This option is also toggled on and off in the 'View' menu. When 'on' the background of the form displays grid lines and grid points.

Note: To toggle grid snapping on and off choose 'Snap to grid' in the 'Format' menu.

Methods of Selecting Objects

A **Single** object can be selected by clicking once with the mouse. An object is shown as selected by the visibility of the selection 'Handles'.



Using the **TAB** key you can move to each object on the form in turn. Use SHIFT+TAB to move to the previous object.

You can select **Multiple Objects** in three ways.

- Click on the ruler to select objects through a straight line.
- Hold down the SHIFT key and select each object required. (Select the object again to deselect)
- Using the mouse drag a selection 'window' around or through the required objects.

Note: The default selection behaviour can be set in the 'Options' list box. Choose 'Options...' from the View menu. Then choose 'Form & Report Design'. The selection behaviour options are 'Fully enclosed' and 'Partially enclosed'.

Moving Objects

- **Un-selected** objects can be moved by clicking and dragging with the mouse pointer
- When **selected** an object or objects can be moved by the mouse when the mouse pointer turns into a 'hand'. Click and drag the object to the required position.
- To move an individual **selected** object move the mouse pointer over the large top left control handle until it changes to a pointing finger. Click and drag the object to the required position.

Note: When a text box has an attached label both objects will move together. To move these objects individually use the top left control handle.

The Flights Form

The design of the Flights form must include the following criteria;

- It must store the Date and Time of the flight.
- It must be able to designate an Aircraft and show the number of seats available.
- The destination will be chosen from any airport available.
- The current costs to the destination must be shown.
- You must be able to book a passenger on to this flight and see the total number of passengers.

These are the main criteria you will concentrate on initially. Other enhancements will be added later.

Adding A Combo-Box

The Form Wizard has produced a 'Flights' form with a 'PassSubForm' sub form that will allow passengers to be 'booked' onto each flight. So far so good.

- Change to form view and look at the current record.

The entry of Flight_ID, Time and Date pose no problems. However, as in table view you would have to remember both the number of the destination (Dest_ID) and also the type of aircraft (Plane_ID) if you wanted to enter a record. As you have seen before this is not a comfortable situation.

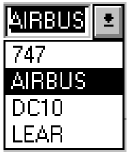
The solution would be to enable the user to choose each option from a list. Access provides this facility in both a list box and a combo-box.

You will now add two combo-boxes to your form.

Firstly the Plane_ID combo-box using a combo-box wizard.

- Delete both the Dest_ID and Plane_ID from the 'Flights' form. (Select each and press the delete key)
- Open both the Field List and the toolbox by clicking on their buttons.
- Make sure the Control Wizard button is depressed. This will invoke the control wizard when you add a control.
- Click on the combo-box button .
- Then from the field list box click and drag the 'Plane_ID' field onto the form and position it below the 'Flight_ID' field. Release the mouse button.
- The Combo-box Wizard will open.
- Choose to look up values from a table or query.
- Choose 'Next>'.
- Select the 'Aircraft' table and choose 'Next>'.
- Choose to show the Plane_ID field and choose 'Next>'.
- In the next dialogue drag the column width in to display just a little more than the six characters allowed. Choose 'Next>'.
- Choose to Store the value in the field. Make sure Plane_ID is selected. Choose 'Next>'.
- Accept the label and choose 'Finish'.

Change to form view and look at the values in the Plane_ID combo-box.

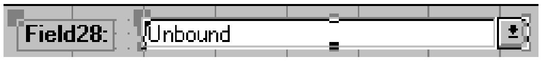


AIRBUS

747
AIRBUS
DC10
LEAR

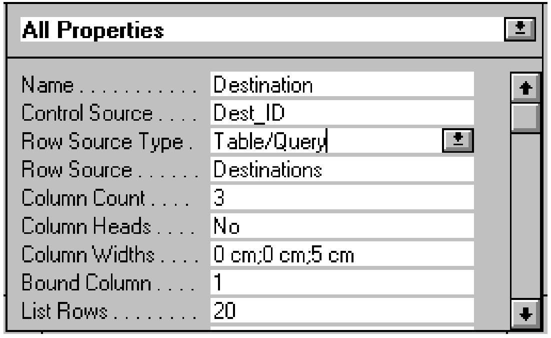
Now you will add the Dest_ID combo-box without using the Combo-box Wizard.

- Change to Design view in the 'Flights' form.
- Click on the Control Wizard button in the toolbox to turn this feature off.
- Click on the combo-box button.
- In the field list click on Dest_ID.
- Move the mouse pointer onto the form.
- The mouse pointer changes to the combo-box insertion symbol.
- On the right hand side of the form level with Plane_ID click and drag a combo-box over five large grid squares.
- It should look something like this



Field28: Unbound

- Open the Properties box and set the following values



All Properties

Name	Destination
Control Source	Dest_ID
Row Source Type .	Table/Query
Row Source	Destinations
Column Count	3
Column Heads	No
Column Widths	0 cm;0 cm;5 cm
Bound Column	1
List Rows	20

- Click on the label and change its caption property to 'Destination'
- View the form in Form View and look at the values in the list.

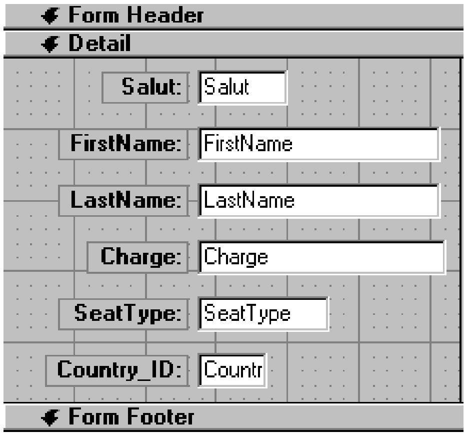
Now the form is beginning to look more useful. There is however much more that can be done to aid the entry of data.

Enhancing The Passengers Sub Form

The passengers subform needs to be improved to aid data entry. The Country_ID will have to be remembered at present, it would be better to choose this from a combo-box. And while we are about it why not choose the SeatType from a list of values too.

It would be preferable for the sub form to blend in more with the main form and this can be done by viewing the sub form in form view rather than datasheet view.

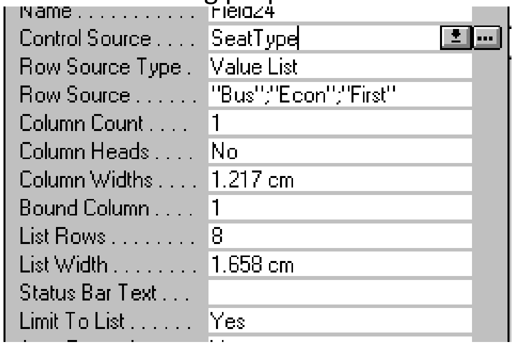
- Change to design view in the Flights form.
- Select the 'PassSubForm' control and make a note of the Width property value. This will come in handy later.
- De-select the 'PassSubForm' control and then Double-click on it to open it in design view.
- The view should be similar to this...



- Delete the Country_ID and the SeatType text boxes and all the labels.
- Using a Combo-box Wizard add a Country_ID combo-box.
- Choose 'Country_ID' and then 'Country' as the columns to display.
- Double-click the right hand side of the Country_ID column to set 'best fit'.
- Country_ID is to be stored in the table.
- Finish and delete the label.

Now add a SeatType Combo-box.

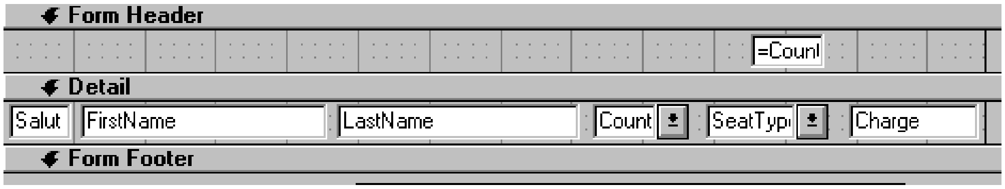
- Add this without a Wizard and set the following properties...



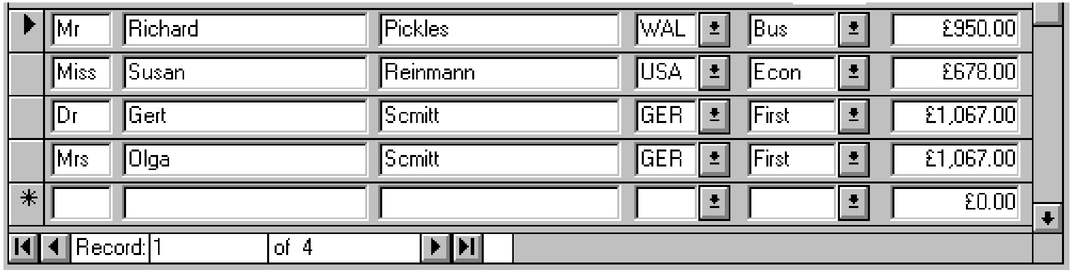
Name	Field24
Control Source	SeatType
Row Source Type .	Value List
Row Source	"Bus";"Econ";"First"
Column Count	1
Column Heads	No
Column Widths	1.217 cm
Bound Column	1
List Rows	8
List Width	1.658 cm
Status Bar Text . . .	
Limit To List	Yes

Note the Limit To List setting. This ensures only values in the list can be entered.

- Set the Form Width property to approx. 1cm less then the 'PassSubForm' control.
- Change the form layout to look like this...



- Change the forms Default View property to 'Continuous Forms'.
- Save and then Close the form.
- In the Flights form change to form view.
- The sub form should look something like this...



▶	Mr	Richard	Pickles	WAL	Bus	£950.00
	Miss	Susan	Reinmann	USA	Econ	£678.00
	Dr	Gert	Scmitt	GER	First	£1,067.00
	Mrs	Olga	Scmitt	GER	First	£1,067.00
*						£0.00

Record: 1 of 4

Using The Database

Sorting Data

Sorting data is very easy in Microsoft Access. You can sort both text fields in alphabetical order and values in size order.

There are two sort options these are;

AZ↓ **Sort Ascending** where text is sorted from A to Z and numeric values are sorted from the smallest to the greatest.

ZA↓ **Sort Descending** where text is sorted from Z-A and numeric values are sorted from the greatest to the smallest.

You use the sort buttons shown above to perform **temporary** sorts on tables and forms. Note that the sort order is not saved when you close the table or form.

The **permanent** sort order for tables is based on the primary key if one has been defined.

- Open the Destinations table in datasheet view.
- Select the 'First' field column.
- Click the ZA↓ sort descending button.
- The most expensive first class fare will be shown as the first record.
- Select the 'City' field column.
- Click the AZ↓ sort ascending button.
- The Cities will be listed in alphabetical order.
- Open the Flights form.
- Move to the Date field.
- Click the AZ↓ sort ascending button.
- The earliest Flight is now the first record.

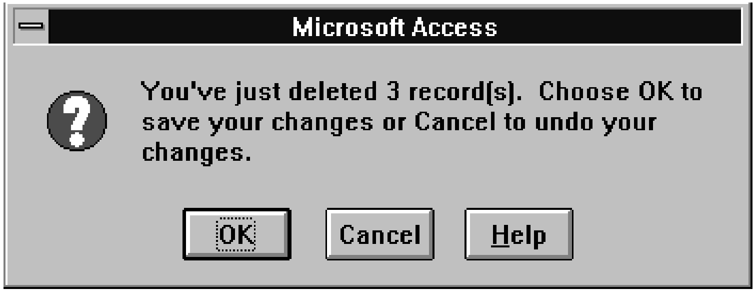
Deleting Records

When you delete a record in an Access database you cannot recover it by trying to close the file and 'not saving' changes as you may be able to do in a spreadsheet program. Once a record has been deleted it cannot be recovered, the same goes for any related records that would be deleted also.

Deleting Records From A Table

- Click on the record selector to highlight the whole record.
- Press the DELETE key or choose 'Delete' from the 'Edit' menu.
- To delete several records click and drag the mouse pointer over the record selectors of the record to be deleted.
- Or, click on the first record of a group to be deleted, hold the SHIFT key, and select the last record to be deleted.
- Again press the DELETE key or choose 'Delete' from the 'Edit' menu.

Access will then give you the message...

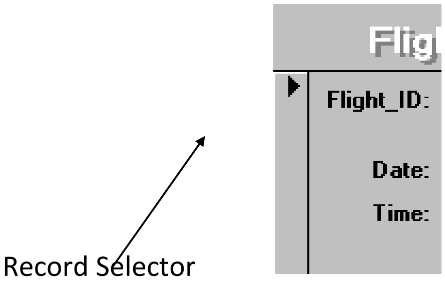


You've just deleted 3 record(s). Choose OK to save your changes or Cancel to undo your changes.

If you choose OK you will permanently lose the data in the deleted records.

Deleting Records From A Form

To delete a record in a form you must first select it. If the form has a record selector click on this and delete the record as you would in a table.



If there is no record selector on the form choose 'Select Record' from the 'Edit' menu and then delete the record. This deletes the current record on the form.

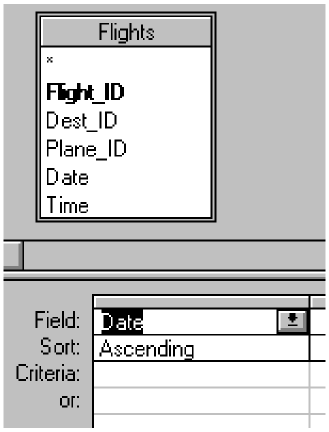
Again you are given the option to cancel your action.

Using A Filter

Access provides an easy way of working with just the records you want to view. This is known as a **Filter**.

A filter can be used on a table or a form in the same way. By specifying criteria you can view only the records needed. A filter is almost identical to a query except;

- It is temporary. (Although it can be saved)
- It always shows all of the fields in each record. Unlike a query where you can show only the fields you choose.



Field: Date
Sort: Ascending
Criteria:
or:

- Open the Flights table and click on the 'Edit Filter/Sort' button .
- Click in the first Field: box and choose 'Date' from the list.
- Click in the Sort: box and choose 'Ascending'.
- In the field list click on 'Time' and drag this to the second Field: box.
- Release the mouse and the 'Time' field is inserted.
- Select to sort the Time 'Ascending'.
- Click on the 'Apply Filter/Sort' button on the toolbar .
- The flight records will be ordered by ascending Date first and then by ascending Time for each Date.
- Now Click on the 'Edit Filter/Sort' button again.
- Click once on the Field Selector for the Time field. (The grey bar above the field name)
- Click on this again and drag the Time field to the left.
- Release the mouse and the Time field should now be to the left of the Date field.
- Click on the 'Apply Filter/Sort' button and view the results.

Records are sorted by fields from left to right in the filter grid. So as in the previous example the times of the records are sorted first and then if any flights occur at the same time the dates are sorted into chronological order.

The **Show All Records** button removes the filter and displays all of the records in their original sort order.

- Click on the 'Show All Records' button.
- Notice the records return to the original order.

Your filter settings still exist and you can reapply them by;

- Clicking on the 'Apply Filter/Sort' button.

Saving A Filter For Future Use

You may create a filter that would be useful in everyday use. It would be tedious to re-create this each time you wanted to use it so Access allows you to save the filter.

First let's make our filter more useful.

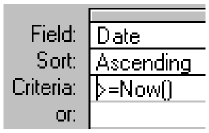
Seeing the flights in Date and then Time order is satisfactory, but viewing flights that have actually happened will be unnecessary.

Access provides many functions that can be used as criteria in the filter grid. One of these is the **Now()** function.

Now() returns the current system date and time set on your computer.

To use the Now() function;

- Click on the ‘Edit Filter/Sort’ button.
- Swap the sort order to be Date then Time.
- In the Date criteria box type ‘Now()’



Field: Date
Sort: Ascending
Criteria: >=Now()
or:

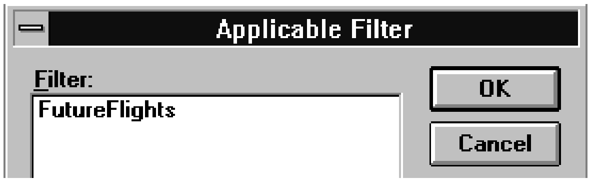
- Click on the ‘Apply Filter/Sort’ button to see the effect.

Now you will save the filter;

- Click on the ‘Edit Filter/Sort’ button.
- From the ‘File’ menu choose ‘Save As Query...’
- Name this ‘FutureFlights’.
- Choose OK.

The filter is now saved as a query.

- Close the Flights table.
- Open the Flights table.
- Click on the ‘Edit Filter/Sort’ button.
- From the ‘File’ menu choose ‘Load From Query...’



Filter:
FutureFlights

- Any queries that can be used as a filter will be displayed.
- Select ‘FutureFlights’ and choose OK. The criteria will be loaded into the filter grid.

Creating Queries

A query is similar to a filter but allows much more flexibilty when viewing data. You can choose to show only certain fields and include fields from other related tables.

A Form and Report can also be based on a query. This allows use to view and edit fields from multiple tables and to produce reports based on multiple tables.

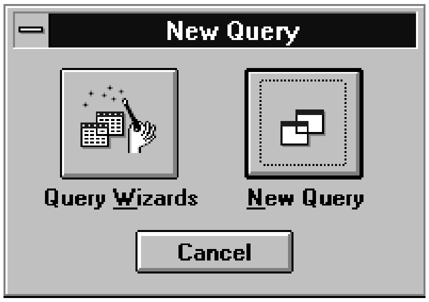
Select Queries

A select query extracts records from the table/s that match the criteria specified in the grid. Unlike a filter this is a ‘one time’ event that occurs when the query first runs. To change the criteria you must re-run the query.

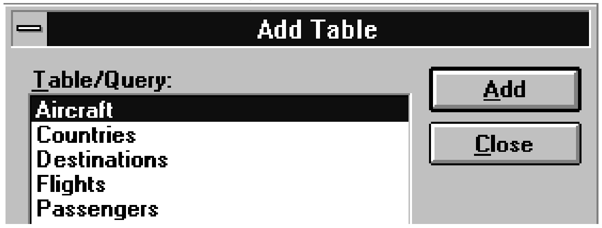
Creating A Select Query

- In the database window click on the Query label to display the list of queries.

- Click on the 'New' button.



- In the New Query dialogue box click on the large 'New Query' button.

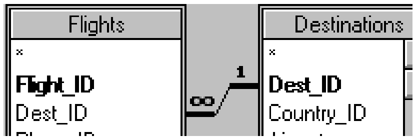


- You are then asked which tables to add to the query layout.

You will now create a query to show each flight. Along with the Flight_ID you need to see Date and Time, Airport, City and Country.

The fields you need to use come from three different tables, Flights, Destinations and Countries. These then need to be added to the query layout.

- Select Flights and click the 'Add' button.
- Select Destinations and click the 'Add' button.



- Notice that a join is automatically created. This is because you have created a relationship between the two tables.
- Select Countries and click the 'Add' button.
- Close the 'Add table' dialogue box.

Now all the fields you require are available they can be added to the grid. Note that if a table joins two others it must be added to the layout even though it may not hold any fields you need to use in your query.

- As with a filter add the following fields to the grid in the order from left to right as shown.
- From the Flights table - Flight_ID, Date and Time.
- From the Destinations table - Airport and City.
- From the Countries table - Country.
- Set the Date field sort order to 'Ascending'.
- Set the Time field sort order to 'Ascending'.
- Click on the 'Datasheet View' button or the 'Run query' button. To display the dynaset.
- Click on the 'Design' button to return to design view.

You will now save this query for future use;

- From the 'File' menu choose 'Save'.
- Name the query 'SortedFlights'.
- Choose OK.

AllPassFlights Query

We will now create a query to show each which flight each passenger is booked on. This will also need to show the Airport, City and Country of destination.

- Create a new query.
- Add the Passengers, Flights, Destinations and Countries table.
- Notice that the Countries table is joined to both the Passengers and the Destination. Having a join like this could cause unpredictable results and give erroneous data so it must be deleted.
- Click on the join line between the Countries and Passengers tables and press DELETE.
- Now add the following fields to the grid;
- From Passengers - Salut, FirstName, LastName, SeatType, Charge and Flight_ID.
- From Flights - Date and Time.
- From Destinations - Airport.
- From Countries - Country.

This list will be used to find a passenger by name. When looking a name up in a list it is proper to search for the Surname then the first name. To reflect this in your query;

- Move the LastName field to the left of the FirstName field.
- Set the LastName sort order to 'Ascending'.
- Set the FirstName sort order to 'Ascending'.
- Run the query.
- Save the query as 'AllPassFlights'.

Adding A Table To A Query

As you had to delete the join between the Countries and the Passengers table it would appear that it is impossible to display the country (text) the passenger comes from.

You can however add a table a second time to the query layout. The same process is also used to add a new table for the first time.

- In design view in the 'AllPassFlights' query .
- Click on the 'Add Table' button or choose 'Add table...' from the 'Query' menu.
- Select the Countries table and choose 'Add'.
- Close the 'Add Table' dialogue box.
- Notice that the table has been added to the layout with the name 'Countries_1'.

This table has not automatically been joined to another because it is referenced with a different name.

You must now create a join to tell the query which fields to link together.

As when creating a relationship you drag the field name from one table to the matching field in the other table.

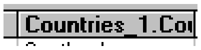
Note: In this situation as you are not creating a relationship it does not matter which table you select first when dragging the field.

- Drag the Country_ID field from 'Countries_1' to the Country_ID field in Passengers.
- Release the mouse and the join line will appear.
- You can now drag the Country field from the 'Countries_1' table into the grid.
- Drag the field and drop it onto the SeatType field.
- Release the mouse.
- Country will appear to the left of SeatType in the grid.

- Run the query !.

Changing The Field Name In A Query

Notice that the label at the top of the passengers country column displays the table name then the field name thus;

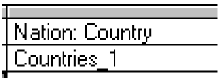


Countries_1.Cou

This is not very descriptive. You can assign a name to a field in a query. This is then used in any forms or reports that are based on the query.

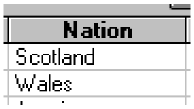
In this case change the name to 'Nation'.

- In design view click on the 'Table names' button . This displays an extra row in the grid showing the table name of the field.
- Click in the Country field from the Countries_1 table.
- Move the cursor the left most position in the field.
- Type 'Nation:' followed by a space, i.e. .



Nation: Country
Countries_1

- Run the query again.
- The column heading should now read 'Nation'.



Nation
Scotland
Wales

- Save the query.

Hiding Fields In A Query

There may be times when you need to use a field in a query but do not want that field displayed. For example, you may sort on a field or specify criteria for a field but it is not required in the output.

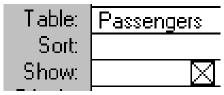


Table:	Passengers
Sort:	
Show:	☒

The **Show** box allows you to decide to display a field or not. When checked the field is shown when unchecked the field is not.

- Change to design view in the 'AllPassFlights' query.
- Uncheck the 'Show' boxes for the following fields;
- SeatType, Charge, Date, Time, Flight_ID, Airport, Country.
- Run the query.

Expressions And Functions In Queries

As with a filter you can enter an expression to the criteria box to specify the field value you want to search for.

You can also enter an exact value to find

- In design view in the 'AllPassFlights' query.
- Check the Show box for the 'Country' field.
- In the Criteria box type 'Australia'

- Run the query.
- Return to Design View.

Wildcard characters can also be used to specify values in the criteria. This is a character that can be used to replace others in a string. For example, the * will show all characters. If you use a wildcard character Access will take it to mean that you want to use the 'Like' function.

- In the Country field replace 'Australia' with 'A*'.
- Press the Down-arrow key.
- Access will replace your typing with - Like "A*".
- Run the query.
- You will see all the country names beginning with the letter A.
- If you remove the wildcard character no countries are found, because no countries are called A.

The Like function can be used to search for many patterns. The characters in the pattern are;

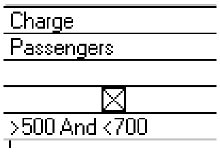
?	Any single character
*	Zero or more characters
#	Any single digit (0-9)
[*charlist*]	Any single character in *charlist*
[!*charlist*]	Any single character not in *charlist*

- Type Like "[!A-C]*" in the Country criteria box and run the query.
- You should see all the countries that do not begin with the letters A,B, or C.

For more information on the Like function search Help for 'Like'.

Searching for a range of values.
You can search for numeric values that have an upper or lower limit or that fall between a range of values.

- In design view in the 'AllPassFlights' query.
- Remove the Like criteria in the Country field.
- Check the Charge field Show box.
- In the charge criteria type '>500 And <700'.



Charge
Passengers

☒
>500 And <700

- Run the query.
- Then in the 'Or' box below the Charge criteria add - >900 And <1000
- Run the query again.

This will show any Charge greater than £500 and less than £700, it will also show those greater than £900 and less than £1000.

- Change the Charge criteria to read - Between 500 And 1000.
- Run the query.

Between will include the values £500 and £1000.

It can also be combined with the NOT operator. Try 'Not Between 500 And 1000'.

Date and Time fields can be used with the Between operator.

Close the AllPassFlights Query and DO NOT save changes

Totals Query

You may want to see totals of groups in your records. For example, how many customers purchased a particular product.

A Select query can be changed to a totals query by clicking on the ‘Totals’ button. Σ

- Create a new query
- Add the Passengers table
- Click the ‘Totals’ button on the toolbar.
- Notice the Totals row appear in the grid.
- Add ‘Country_ID’ to the first field. The Totals row should read ‘Group By.
- Add ‘Country_ID’ to the second field.
- Choose ‘Count’ from the Totals row list.
- Run the query.

You will see the total number of passengers from each country.

To see an the average paid by each person from each country;

- Add Charge to the third field.
- Set the Totals row to ‘Avg’

Crosstab Query

A Crosstab Query is similar to a Pivot Table for those people with spreadsheet experience. This creates a summary of data in your tables.

The Crosstab query Wizard is usually the quickest way of creating a Crosstab query, but to understand the mechanics you will now create one from a blank query.

The following exercise is quite simple but will show the use of a Crosstab query quite neatly.

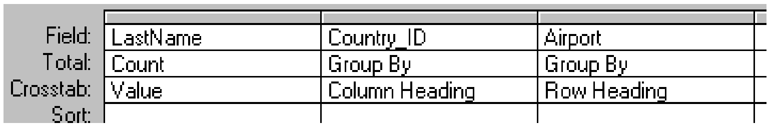
- Create a new query and add the Passengers table
- Click the Crosstab Query button on the toolbar
- Notice the Crosstab row appear in the query grid.
- Add SeatType to the first field box in the grid.
- Total should be ‘Group By’.
- Set the Crosstab row to read ‘Column Heading’.
- Add Country_ID to the second field box.
- Again Total should be ‘Group By’.
- Set the Crosstab row to read ‘Row Heading’.
- Add SeatType to the third field box.
- Set the Total to ‘Count’.
- Set the Crosstab row to ‘Value’.
- Run the query.
- Save the query as ‘CountrySeatCrosstab’.
- Close.

You will see the number of passengers from each country and the type of seats they choose.

A Crosstab query must have one Value, one Column Heading and at least one Row Heading.

The fields you enter must be sensibly selected or you may get errors or unpredictable results.

Try creating the following Crosstab query;



Field:	LastName	Country_ID	Airport
Total:	Count	Group By	Group By
Crosstab:	Value	Column Heading	Row Heading
Sort:			

Using A Select Query As AutoLookup

AutoLookup is used to automatically look up and display records. This works when two tables have a one-many relationship and the join field on the 'one' side has a unique index.

The secret to designing a lookup query is to include the join field from the 'many' side of the query.

In a query with a one-to-many relationship you can only change the value of the join field from the 'many', not the value from the 'one' side.

The Wings application has a need of AutoLookup as might many of the databases you will design in the future. First you will create a query that performs AutoLookup and then you will see how to apply this to your application.

- Create a new select query and add the following tables;
- Aircrafts, Flights, Destinations and Countries.
- Add the following fields to the grid;
- From Aircraft add...
- Bus, Econ, First.
- From Flights add...
- All fields.
- From Destinations add...
- Airport, City, Bus, Econ, First.
- From Countries add...
- Country.
- Run the query
- In the dynaset move to the last record.
- Move to the Plane_ID field.
- Type 'LEAR'
- Press ENTER or TAB.
- Notice the Bus, Econ and First fields have been filled automatically.
- In the Dest_ID field type '29'.
- Tab across the dynaset and look at the fields that have been filled.
- Press ESC to abandon your changes.
- Save the query as 'BaseForFlights'.

Remember the secret is to put in the join field from the '**many**' side. (The foreign key)

You will have seen when trying to enter data into a related table that it is difficult if not impossible to remember all of the values for the foreign key in the table. Entering data in the previous way is also prone to these pitfalls. You have seen on the Flights form however that by adding a combo-box you can select a value from the list.

Basing A Form on A Query

Return To The Flights Form

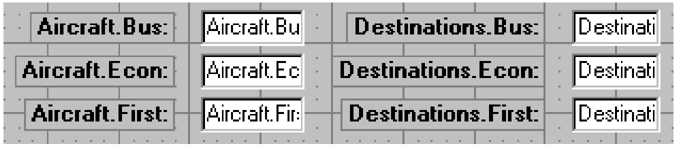
The Flights form is coming along reasonably but does not yet fulfil all of the criteria specified. For example it does not show how many seats are available on each plane and also needs to display prices to each destination.

You have seen that a query can be used to look up this automatically, and that you have just saved a query that meets this criteria.

The design of the Flights form is based on the Flights table however and you do not want to start redesigning this. Especially as much work has gone into the subform.

There is a solution to this;

- Open the Flights form in design view.
- Open the Properties box.
- Click on the record source box.
- From the list choose 'BaseForFlights'.
- Close the Properties box.
- Open the Field list.
- Look at the fields that are available.
- Add the Country field to the form just below the Destination.
- Add the following fields as in the picture;



Aircraft.Bus:	Aircraft.Bu	Destinations.Bus:	Destinati
Aircraft.Econ:	Aircraft.Ec	Destinations.Econ:	Destinati
Aircraft.First:	Aircraft.Fir	Destinations.First:	Destinati

- View the Form.
- Selecting a different destination and aircraft will now change the values in the newly added fields.

The Flights form now specifies most of its criteria.

The Flights Schedule Form

The remaining design work on the Flights form will involve expressions and some programming, so before you attempt that there is another form that is needed.

If a passenger were to call and enquire about a flight it would be time consuming to move to a particular flight using the flights form.

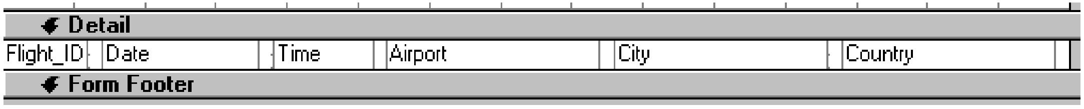
An enquiry for Munich for example would be difficult to find or filter because the form is based on the Dest_ID field which is a number. It would be useful to be able to see the flights in chronological order and then to be able to switch to a particular flights details.

This will also be a requirement in many of the databases you will design.

Again a query provides the solution.

You have already created a query called '**SortedFlights**'. This will now be used as the base for our scheduled flights form.

- Create a new blank form based on the 'SortedFlights' query.
- Add all of the fields to the form.
- Delete all of the labels.
- Reposition the fields to look something like this;



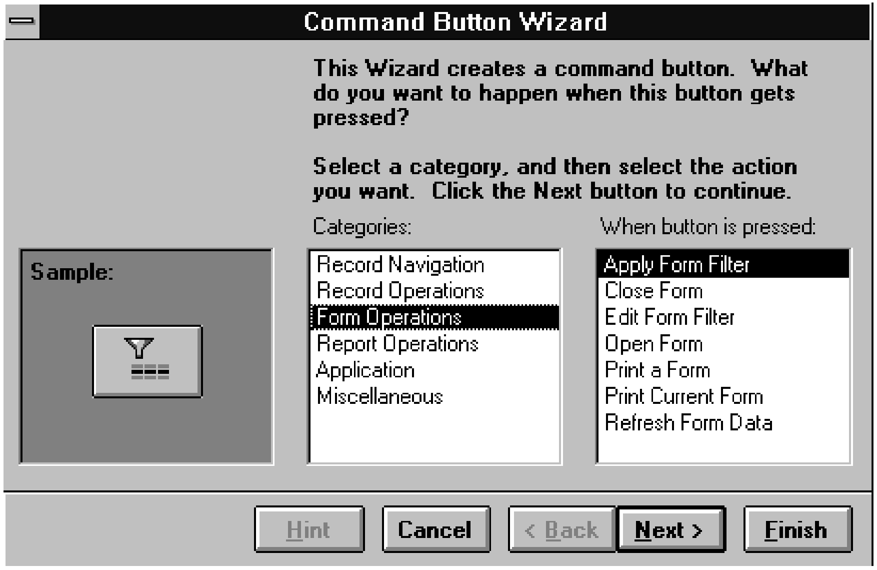
- The form should have a one square deep header.
- Set the forms Default View property to 'Continuous Forms'.
- Set the border of each field to be clear.
- View the form.
- Save the form as 'FlightSchedule'

Now you need to be able to select a record and go to the relevant flight detail.

Adding A Command Button

The easiest way to add a command button is to use the Command Button Wizard. You will now add a button that opens the flights form using the Flight_ID from the current record.

- Change to design view on the 'FlightSchedule' form.
- Open the toolbox.
- Make sure the Wizard button is 'on' (Depressed).
- Click on the Command button in the toolbox ▭.
- Move the pointer to the Form Header and drag a button across four squares.
- The Command Button Wizard will open...



This Wizard creates a command button. What do you want to happen when this button gets pressed?

Select a category, and then select the action you want. Click the Next button to continue.

- Choose 'Form Operations' and then 'Open Form'.
- Choose 'Next>'.
- Choose the Flights form as the form to open then choose 'Next>'.
- As you need to open the Flights form with the chosen Flight_ID you must select the option to 'Open the form and find specific data in it'.
- Then choose 'Next>'.
- In the next dialogue box you must select the fields which link the two forms.
- Select the Flight_ID field in both lists.
- Click the <-> button, and choose 'Next>'.
- You are then given the choice of displaying text or a picture on the button, you cannot display both at the same time.

- Choose the text option.
- Enter the text ‘View Flight Details’.
- Choose ‘Next>‘.
- Give the button the name ‘ViewFlights’.
- Then Finish.
- Resize and reposition the button if necessary.
- Save the form.

View the form, it should look a little like an information board. Select a flight and then click on the ‘View Flight Details’ button.

- Notice that the record navigation bar on the flights form shows only 1 record of 1. This is because the button action has filtered the records.
- To view all the records again click on the ‘Show All Records’ button.

If you want to return to the FlightSchedule form choose it from the ‘Window’ menu.

Expressions In Forms

As promised you will now return to the Flights Form and add the final enhancements.

The criteria stated that the form should show how many passengers were on the flight. This problem can be solved by adding a field to the ‘PassSubForm’ that counts the number of records. This value is then returned to the Flights form.

- Open the Flights form in design view.
- Double-click on the ‘PassSubForm’ control.
- Open the toolbox and click on the Text box tool ab|
- Move the pointer to the Form Header section and drag a three square box.
- Notice that the text box displays the word ‘Unbound’.

Unbound means that the control is not linked to any field. If you viewed the form the text box would appear blank.

You can add an unbound text box anywhere on your form.

An unbound text box is used to hold an **expression** or to display data as the result of an expression in another field.

An expression can consist of a function or be made of other field values. For example, an unbound control could display the result of a calculation from other controls on the form.

Typical functions are Sum(), Count() or Now()+30.

It is important to remember that a control on a form is referenced by *its* **name** and not the name of the field it displays.

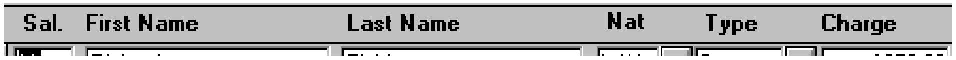
- Open the Properties box for the newly added text box.
- Set the ‘Name’ property to ‘PassTotal’.
- In the Control Source box type;   =Count([SeatType])
- View the form.

The text box should display the total number of passengers.

How then do we reference this to the main form?

The solution to this is to add another Unbound control to the main Flights form. Before this tidy up the sub form.

- In design view on the 'PassSubForm'...
- click on the 'PassTotal' text box.
- Set its visible property to 'No'
- From the palette set its back colour to red.
- Reduce the width of the text box to cover less than one square.
- Delete its label.
- Move it to the far right of the Form Header.
- Add labels to the Form Header as in the picture;



- Save the sub form and Close.

You now need to add a field to the Flights form to display the sub form total.

- Add a text box below the Date and Time on the Flights form.
- Open the Property box for this text box.
- Click on the Control Source box.
- Press SHIFT+F2 to open the Zoom box.
- Type in the following expression;

```
=[PassSubForm].[Form]![PassTotal]
```

- This expression will 'read' the value held in the 'PassTotal' control on the 'PassSubForm'.
- Change the label connected to the text box to read 'Passengers on flight'.
- View the Form.

A Bit Of Basic

There is one last niggling problem to solve and the only practical way to do this is to use an Access Basic procedure.

When entering a passengers booking the Charge is having to be entered as well. Not only is this time consuming but it is prone to errors. The Charge must be stored separately because the price to the destination could change. If it was just looked up each time a price change was made all the passengers Charges would change. Not good for business!

There are also three possible charges which makes entry even more prone to errors.

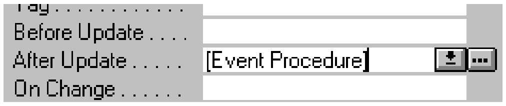
The only way around this is to add some code. Watch carefully.

- On the Flights form there are three fields that reference the destinations costs.
- Change the 'Name' property of each to 'Bus', 'Econ' and 'First' depending on which field they reference.

You will now use these control names on the PassSubForm.

- Double-click on the PassSubForm control.
- Open the property box for the 'SeatType' text box

.



Tag . . . .
Before Update . . . .
After Update . . . . . [Event Procedure]
On Change . . . . . .

- Click on the 'After Update' box. (You'll need to scroll down to this)

- Then click on the 'Build' button (with three dots).
- The Module window will open.
- Move the cursor to the end of the first line in the window and press ENTER.
- Type in the following code;

```
Select Case SeatType
   Case "Bus"
      Charge.Value = DLookup(Forms![Flights]![Bus], "Flights")
   Case "Econ"
      Charge.Value = DLookup(Forms![Flights]![Econ], "Flights")
   Case "First"
      Charge.Value = DLookup(Forms![Flights]![First], "Flights")
   End Select
```

- From the 'Run' menu choose 'Compile Loaded Modules' this will check your syntax.
- Close the Module window.
- Save and then Close the form.
- View the flights form and add a passenger record.
- Choosing a value in the Type column should add the correct price to the Charge box.
- Welcome to Access Basic.

It is beyond this course to go into detail about Access Basic programming. You will probably find it to be the only answer in certain so it is well worth your while to look into it.

To explain the code you have just entered, you must understand events. An event can be triggered by the user or by another piece of code. In your example the event is triggered by the change to the value in the SeatType combo-box when the user makes a choice. This is the **AfterUpdate** event. In other words when the field is changed, do something.

This causes the code in the AfterUpdate subroutine to run.

Select Case tells the routine to jump to whichever **Case** value matches the value in SeatType.
So if 'Bus' was chosen the 'Bus' Case will be executed.

Dlookup is a function that looks up the value held in a control, much the same as the way a query looks up a value automatically.

All you need to do then is explain which value to lookup. This is done by [Forms]![Flight]![Bus] which gets the value from the 'Bus' text box on the Flights form.

The value is then passed to the left hand side of the statement. **Charge** is the name of the control on the PassSubform and the value returned is deposited in the **value** property of this control. As the control is 'Bound' to the Charge field that is where the value ends up.

When the 'Bus' Case is done the routine jumps to the **End Select** statement and then the sub-routine finishes.

Creating Reports

Now that you have completed the design of your 'user interface' it is time to produce some reports using the data that is stored.

Access allows you to design a report to convey your data in exactly the way you want.

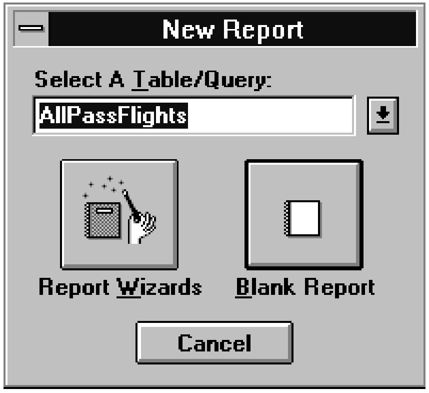
As with a form you can base a report on a table or a query. Using a query you can print fields from multiple tables.

Creating A Passenger List Report

A typical report that would be needed by the Wings Company is a list of the passengers on each flight. This could be given to the flight crew for example.

The tables you need for this report have already been used in the 'AllPassFlights' query so you will use this as a base for your report.

- In the database window click on the Report label to view the Report list.
- Click on 'New' . Alternatively click the 'New Report' button or choose 'New' from the 'File' menu then 'Report' from the flyout.
- Select the 'AllPassFlights' query then click the 'Blank Report' button.



New Report

Select A Table/Query: AllPassFlights

- The report design window is similar to the form design window and the toolbox, palette, field list, properties box and code buttons are the same.
- As with a form a report can have a **Report Header/Footer** and a **Page Header/Footer**.
- The **detail** section displays records in the same way as a form.
- A report can also have sections based on groups. Each of these appears around the detail section as a Header and Footer if you choose to display them. The Sorting and Grouping box can be opened by clicking the button on the toolbar
- As well as the normal Print Preview button there is also a Sample Preview button . This gives a quick preview of your report, but **beware** the data shown may not be what you expect.
- Defaults can be set as with forms.
- Set the Default Text box 'Auto Label' property to 'No'.

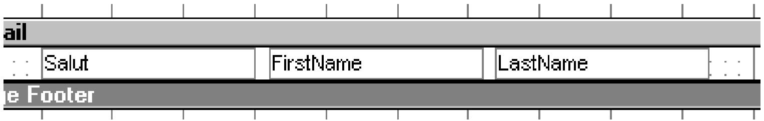
The position that you add fields onto the form will determine the data that is displayed. You will need to think of the grouping of your data as you would do in a Totals query. The highest group level corresponds to the left most field in a sort order.

The **details** you need to see in this report are the passengers names. The flight crew need to see which type of seat the passenger has booked, and obviously which flight each passenger is booked on.

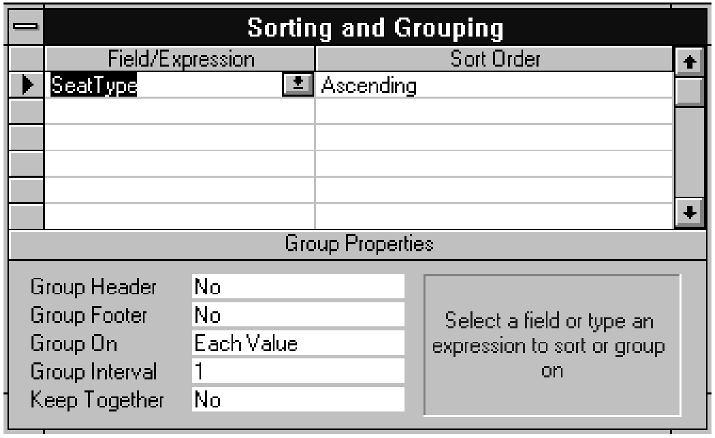
This immediately gives you three group levels ie.
Flights
SeatType
Passenger

To apply this structure to the report;

- Add the Salut, FirstName and LastName fields to the Detail section and reduce the size of the detail section as shown...



- Click on the Sorting and Grouping button to open the Sorting and Grouping box.



Field/Expression	Sort Order
SeatType	Ascending

Group Header	No
Group Footer	No
Group On	Each Value
Group Interval	1
Keep Together	No

Select a field or type an expression to sort or group on

- In the Field/Expression box add SeatType.
- In the Group Properties section set Group Header to 'Yes'.
- Add the SeatType field to the SeatType Header section on the report.
- In the second Field/Expression box add the Flight_ID field
- In the Group Properties section set Group Header to 'Yes'.
- Add the Flight_ID field to the Flight_ID Header section.
- **Notice** that the groups are the wrong way round. This is not a problem.
- The grey box next to each field in the Field/Expression box acts as a record selector.
- Click once next to the Flight_ID field. To highlight the row.
- Then click and drag the field up until it is above the SeatType field. Release the mouse.
- Notice the change this has made to your report design.
- Run the report by clicking on the Print Preview button
- Close the preview by clicking on

The result may not be very pretty but you should be able to see the affect of your grouping.

- Add the Date, Time And Airport fields to the Flight_ID Header.
- Select all of the above.
- From the toolbar set the text attribute as Bold and the point size as 12.
- From the 'Format' menu select 'Size' then 'To fit' from the flyout.
- Select the SeatType field.
- Set this to Italic and 10 point.
- Preview the report.

Whilst slightly better the names of the passengers are not looking good. This is because the three words making up the name are spaced apart.

The three fields can be printed better by using the '&' operator. This joins strings of text together.

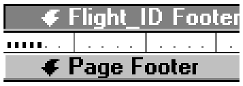
To see this;

- Delete the FirstName and LastName fields.
- Select the Salut field.
- Drag its width across the page.

- Edit the text in the box to read;
  =[Salut] & " " & [FirstName] & " " & [LastName]
- Preview the report.

In real life of course each Passenger list for each flight would need to be printed on a separate page. To do this you can force the report to throw a page at the end of each flight.

- In the Sorting and Grouping box select the Flight_ID Group.
- Set the Group Footer to 'Yes'
- in the toolbox click on the Page Break button
- Move the pointer to the Flight_ID footer and drag a small window.
- A series of dots will appear on the page...



- Preview the report.

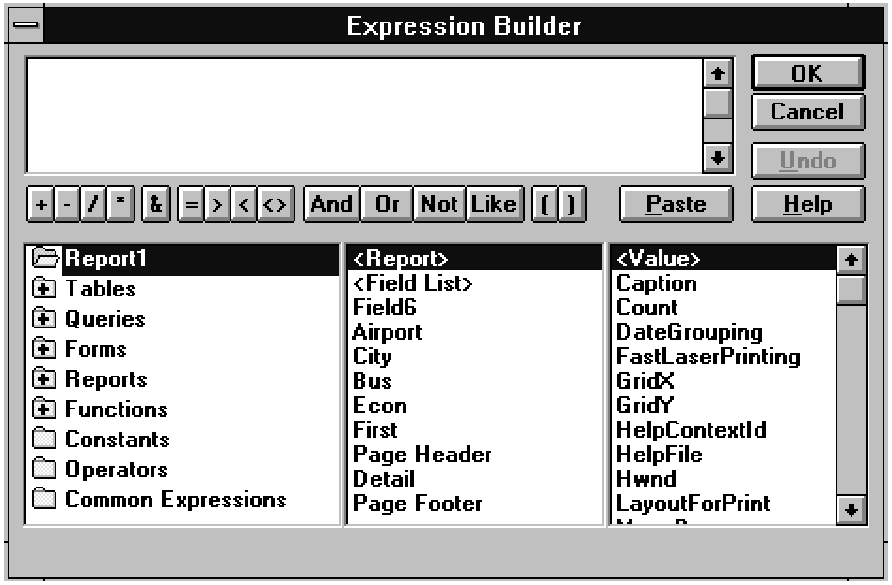
Each flight is printed on a separate page.

- Save the report as 'PassengerList' and Close.

The Destinations Report

Another simple report would be a list of destinations showing prices to each.

- From the database window create a new report based on the Destinations table
- Choose Blank report
- Set the Default Text box Auto Label property to 'No'.
- Add the City, Airport, Bus, Econ and First fields to the Detail section.
- Make the Page Header two squares deep.
- Add labels in the Header to describe the columns below.
- Add an unbound text box to the top right hand corner of the Page Header.
- In the properties box for this text box click on the Control Source box.
- Then click on the build button to open the expression builder...

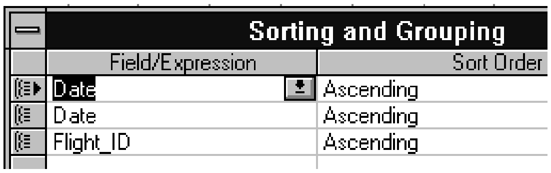


- Click on the Common Expressions folder.
- Choose 'Page N of M' from the list.
- Choose Paste to put the expression into the Expression Window.
- Choose OK.
- Print the report.

Flights Revenue Report

Another report that could be requested could be the amount of revenue generated by the flights.

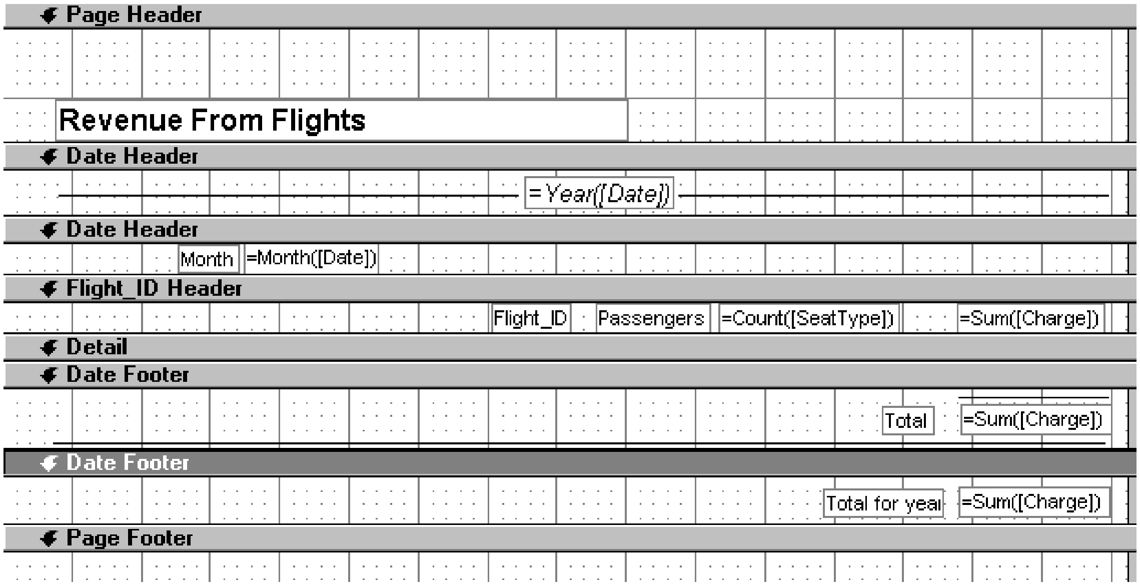
- Create a new report based on the ‘AllPassFlights’ query.
- Create the following groups;



Sorting and Grouping

Field/Expression	Sort Order
Date	Ascending
Date	Ascending
Flight_ID	Ascending

- For the first Date field create a Group Header and Footer.
- Choose to Group on ‘Year’.
- For the second Date field create a Group Header and Footer.
- Choose to Group on ‘Month’.
- For the Flight_ID field create a Group Header.
- Insert fields, labels and calculated fields as shown...



Page Header

Revenue From Flights

Date Header

=Year([Date])

Date Header

Month =Month([Date])

Flight_ID Header

Flight_ID Passengers =Count([SeatType]) =Sum([Charge])

Detail

Date Footer

Total =Sum([Charge])

Date Footer

Total for year =Sum([Charge])

Page Footer

- Set the ‘=Year([Date])’ field as centered, italic and 10 point.
- Set the ‘=Month([Date])’ and ‘=Count([SeatType])’ to be left justified.
- Draw the lines in the Date Header and Footer.
- Save the report as ‘FilghtRevenue’.
- Print the report (when you are satisfied with the preview)

The material appearing in this website is for informational purposes only and is not legal advice. Transmission of this information is not intended to create, and receipt does not constitute, an attorney-client relationship. The information provided herein is intended only as general information which may or may not reflect the most current developments. Although these materials may be prepared by professionals, they should not be used as a substitute for professional services. If legal or other professional advice is required, the services of a professional should be sought.

The opinions or viewpoints expressed herein do not necessarily reflect those of Lorman Education Services. All materials and content were prepared by persons and/or entities other than Lorman Education Services, and said other persons and/or entities are solely responsible for their content.

Any links to other websites are not intended to be referrals or endorsements of these sites. The links provided are maintained by the respective organizations, and they are solely responsible for the content of their own sites.