The Whole Picture Exercise

Objectives

To implement a multiple table database application. By the end of this you will have:

1. Created tables
2. Set data types
3. Set a primary key
4. Set up a relationship
5. Created queries
6. Created a form
7. Created a report

Introduction

This exercise is designed to be done with Access 2000. You are directed to many of the help pages which give a gentle but shallow introduction to the important aspects of relational database systems. For some of you, these will be revision; for some, they will be your first exposure to the ideas. We will be expecting you all to be familiar with this material so use it appropriately. There is probably a special word of warning to those of you with significant database experience - it is easy to overlook things you have not met, when they are surrounded by things which you have met.

We are going to implement a small 'tours' database application. Use the pointers to the **Help** menu **Contents** pages at the key stages indicated. The help pages selected provide a good overview. You may explore further if you have time. Each reference to a help page is a path through the contents hierarchy. For example to get to:

Creating and Working with Databases ⇨**Databases: What they are and how they work**

Go the **Help** menu, then **Microsoft Access Help.** Click on the **Contents tab** (if necessary)**.** Click on the plus symbol by **Creating and Working with Databases**. The next part of the hierarchy is displayed, then double-click on **Databases: What they are and how they work.** Sometimes you will then have to click again to open th etopic.

If you do not know how to complete the task mentioned, first read the appropriate help pages. If you still are unsure, please ask us. Remember that **Shift + F1** gives a different cursor that you can use to find out what a menu command or GUI widget does and that the F1 key gives context sensitive help.

*Task: Read Creating and Working with Databases ⇨ **Databases: What they are and how they work** (Explore the hotspots)*

*Create a new database called **tours.mdb** Note that (unlike a word processor) you have to give a file name at the start rather than when you save or close the "file".*

Data provided

Various operators arrange tours to several places on the Malaysian peninsula (or just offshore). The tours are round trips, stopping at the second place. There are three grades of seat / accommodation, gold, silver and standard. Duration times are for the outward journey only. Fares are for the round trip. You can assume that the return trip will take the same time.

OPERATORS

Code	Name	Telephone	FAX	Charter Rates
EOE	Eastern & Oriental Express	2272068	2249265	No
PLS	Peninsular Line Services Pte Ltd	2242588	2241721	Yes
ABF	Auto Batam Ferries	27144866		
CHH	Channel Holidays	2702228		
DSH	Dino Shipping	276922		

TOURS

Start	Visiting	Duration hour:min	Gold S$	Silver S$	Standard S$	Child discount %	Single Person Supplement S$	Transport	Operator
Singapore	Bangkok		10,380	5,580	4,000	20	2,800	Rail	EOE
Singapore	Johor			395	370		75	Rail	PLS
Singapore	Batam Island	0:40			27	33		Ferry	ABF
Kuala Lumpur	Bangkok		8,240	4,440	3,160	20	2,220	Rail	EOE
Singapore	Butterworth		6,560	3,540	2,500	20	2,220	Rail	EOE
Butterworth	Bangkok		6,560	3,540	2,500	20	2,220	Rail	EOE
Singapore	Malacca			630	590		75	Rail	PLS
Singapore	Batam Island	0:40			30	40		Ferry	CHH
Singapore	Kuala Lumpur		3,600	1,960	1,380	20	1,120	Rail	EOE
Singapore	Bintan	1:30			63	33		Ferry	ABF
Kuala Lumpur	Butterworth		4,640	1,480	1,780	20	1,120	Rail	EOE
Singapore	Malacca				196			Air	PLS
Singapore	Bintan	1:30			63	34		Ferry	DSH

Tables

Task: *Open the Help menu Contents and read the overview:*
Creating and Designing Tables ⇨ ***Tables: What they are and how they work***

You should then read:
Creating and Designing Tables ⇨ *Adding Fields and Choosing Data Types* ⇨ ***What data type should I use for a field in my table****?*

Create the two tables below. If no data type or size is given, choose a suitable one of your own. Leave the tables open for the time being: ***i.e. DO NOT SAVE OR CLOSE THEM****.*

OPERATOR

Field Name	Data type	Size
Code	Text	3
CompanyName		
Telephone		
FAX		
CharterRates	Yes/No	

TOUR

Field Name	**Data type**	**Size**	**Special points**
TourID	Autonumber		Note that this is a long integer
OperatorCode			Must match the code in the OPERATOR table
Transport	text	5	Use a Lookup Wizard to create 1 column containing the values Air Ferry Rail. Set Limit to list to Yes
StartPlace			Use a Lookup Wizard to create 1 column containing the place names Set Limit to list to Yes
PlaceVisited			Use a Lookup Wizard to create 1 column containing the place names Set Limit to list to Yes
Duration	Date/Time		Format Short time
GoldFare			The fares and supplement should all be
SilverFare			The same data type and format
StandardFare			
ChildDiscount	Number	Single	Format percentage, 2 decimal places
Supplement			

Primary keys

Remember that the primary key must be able to UNIQUELY identify a record.

Task: Read:

Creating and Designing Tables ⇨ Working with Primary Keys and Indexes ⇨ ***What kind of primary key should I use?***

Beware the seductive comments on Autonumber primary keys – they are appropriate only in a small minority of situations.

Set ***Code*** *as the primary key for the* ***OPERATOR*** *table. Note that the Indexed property is changed to Yes (No Duplicates).*
Set the TourID as the primary key for the TOUR table. Save the tables with the names we have suggested and close them.

Creating Relationships

Read:
Creating and Designing Tables ⇨ Defining Relationships and Setting Referential Integrity Options ⇨ **About relationships in an Access database**.
In this enterprise (or business), a given operator potentially has many tours. A given tour is arranged by only one operator. OPERATOR therefore has a one to many relationship with TOUR

Task: Click on the ***Relationships*** *icon on the toolbar and* ***Add*** *both tables. Drag the Code field from the OPERATOR table (the one side of the relationship) to the OperatorCode field of the TOUR table (the many side). Enforce referential integrity, then click on* ***Create.***

Note that a value for Code must occur once and only once in the OPERATOR table (it is the primary key), however a given operator organises many tours, so the matching code can occur many times in the TOUR table.

Task: Close the window, saving the changes.

Data validation

Errors in entering data can be reduced by using Input Masks, or writing application program code that validates the data as it is entered.

Task: Open the OPERATOR table. Select the Code field. Click on the Input Mask Property. Use the Input Mask Wizard (…) to create a mask that forces the user to type an alphabetic character. (Use the help) Change to datasheet view and enter the operator data.

Using the lookup field and limiting data entry to the list reduces entry errors as well. Notice that there is a BAD use of the lookup wizard in our TOURS table as well. The Start and Visiting fields both hold the same list of data. Duplicated data is redundant and introduces the potential for inconsistency in the data. In a more developed example, where more information on cities visited would be stored, a separate table would be better practise. We shall come on to Entity Relationship Modelling, Implementation and Normalisation later on. The normalisation process removes such anomalies from the database design.

Creating Forms

Read:
Working with Forms ⇨ **Forms: What they are and how they work**
Forms can be used to present a view of the data to the user. What the user sees is determined by the requirements of the enterprise or business. Different users may see different collections of data, or the same data in different ways. For example, one employee may have permissions to enter data, another may have permission to view the data but not to edit it.

Task: Create a new form using the form wizard. Base the form on the OPERATOR table. Add all the fields from this table AND all the fields from the TOUR table. View the data by OPERATOR with a form/subform. Select a layout for the subform, a style. In the next dialogue box choose to modify the form's design, then finish.

Notice that each field has a pair of associated 'controls'. The field name is in a **label**. The data are displayed via a **text box** that is BOUND to the appropriate field of the underlying table. By default, data are bound to the fields in the table that is bound to the form (i.e. is the Record Source). The visible fields come from the OPERATOR table and are on the MAIN form. This form has a control called a subform. This main form and subform are linked by the Code field of the main form (known in this context as the Master Field) and the OperatorCode field of the subform (Child Field).

Task: Change to form view. Notice that not all the subform is visible. Make the following changes:

On the OPERATOR form: Resize the form and its subform to full screen width. Delete the CharterRates Control and add it back to the form using the check box tool. Change its label to Charter Rates Available? Save your changes and close the form.

On the subform: The TourId field is incremented automatically and need not be displayed. Delete this control and the OperatorCode field from the form and resize the controls so that all data are visible. You could also do this by adjusting the column width in form view. Save the subform.

Text boxes can also be used to calculate and display values from other data.

Task: Add an unbound text box to your subform. Label it Child Fare. Use the expression builder to write an expression calculating the standard child fare. Note that Access expects field names to be enclosed in square brackets. Format this as Currency with 2 decimal places

Close your subform, saving the changes. Now open the OPERATOR form and enter the tour data. Notice that the OperatorCode is entered automatically.

Creating Queries

Queries retrieve subsets of information, which can be presented in a form or report to the user. The data can also be displayed directly, however the form interface can present the data better and be used to protect the data.
Read:
Working with Queries ⇨ **Queries: What they are and how they work**
As you work on each of the following queries, dragging fields into the QBE grid, change views frequently to see the effect on the data retrieved.

Retrieving a subset of data from a table

Task: Start a new query. Add the CompanyName and Telephone fields from the OPERATOR table. View the data retrieved. Save this query as Retrieving_Columns.

Sorting

The next task looks at sorting, using data from one table.

Task Create a new query displaying the data from the OPERATOR table. Sort it in reverse alphabetical order by CompanyName. Save the query as Sort_example.

Data can be sorted on more than one field. Access sorts on the left-hand field first, then subsorts by the next sort field it finds to the right and so on.

Task: Create a new query, adding both tables. Show the CompanyName, Start, Visiting and StandardFare fields. Show the fares in Ascending order and subsort by CompanyName. Save the query as Sort_example_2

If you don't like the layout you could add a second StandardFare field at the end and hide the one that the data is sorted by. A neater way would be to create a form with the query as its Record Source.

Setting Criteria

So far we have retrieved all records, but only displayed certain fields. Data retrieved can be restricted still further by picking out only certain records. To do this, criteria are applied to the fields.

Task: Create a query that will show the CompanyName , Start, Visiting, Transport and StandardFare fields. Set a criterion that will show only those records that start from Singapore. Save it as Criterion_example

Criteria set on the same row of the QBE grid are equivalent to 'AND'
Those set on different rows are equivalent to 'OR'

Task: Change this query so that only ferry tours starting from Singapore are displayed. Save it as Criteria_example_and

Change the query again so that only rail tours from Singapore visiting Kuala Lumpur or Bangkok are displayed. Save this as Criteria_example_or

We have calculated the Standard child fare on a form, using an unbound text box. An alternative strategy would be to calculate these derived data in a query, and base a form on that query.

Task: Add the CompanyName, Transport, Start, and Visiting fields to the QBE grid of a new query. Design a query that will display only these 4 fields listed above and calculate the CHILD fares. If there is no equivalent adult fare, nothing should be displayed for that fare. Save this as Calculated_example_1

In a similar way, a set of fares for a single person who will be charged the supplement fare as well. If a fare field is empty, nothing should be displayed Save this as Calculated_example_2

Did you have a problem with the last query? Check the results. Some data may be missing. The problem involves null values and how operators treat null values. Did you tab over the field (leaving it null) or enter a 0? 0 is not the same as null It would be appropriate for the supplement field but not the other fare fields where a blank would indicate that the fare wasn't available. The alternative is return a Null in your calculation instead. This is a classic problem area, which is best handled by careful database design and implementation.

Grouping

Queries can be used to group data and perform simple calculations on the data groups.

Task: Create a query grouped by company that counts the number of tours it runs. Save this as Grouped_example_1

Alter this query to count the tours visiting each place for each Company. Save this as Grouped_example_2

Alter this query to group by transport, then company. Save this as Grouped_example_3

This sort of data can sometimes be displayed better as a crosstab query.

Task: Convert the last two queries to crosstab queries. Save them as Crosstab_example_1 and Crosstab_example_2 respectively

More about forms

Forms present the user with a view of the data. They can be based on queries as well as tables.

Task: Create a form based on the Calculated_example_1 query. Call this form Child_Fares

Reports

Reports produce a view of the data in a suitable format for printing. They can also be previewed on screen.
Read:
Working with Reports ⇨ **Reports: what they are and how they work**

Task: Produce a tabular report showing the operator details using a report wizard. Sort it by Code. Save it as Operator Details

Notice that, in this tabular format report, one line is displayed per record

*Task: Produce a report based on the OPERATOR table, add the CompanyName, Telephone and FAX details. From the TOUR table add the Start, Visiting and StandardFare fields. **View** the report by **Operator** and **Group** it by **Visiting**. Sort it by Start. Use the summary options to calculate a Min StandardFare. Save it as Grouped_report_1*

The resulting groupings do not make a huge impact with such a small data set, however you should be able to see the structure of this sort of report. If you do not understand it, please ask.

Task: If *you have time, Look at:*
Working with Controls on Forms, Reports and Data Access Pages ⇨ ***Controls: What they are and how they work***

then experiment with customising COPIES of your forms and reports. Go over any points that you found difficult again. You may want to read some of the other pages under "Working with Controls on Forms, Reports and Data Access Pages"

General Points about Forms and Reports

1. They are very similar in concept.
2. All items on the form are 'controls'
3. Controls may be bound to an underlying field, or unbound.
4. They consist of various Headers and Footers plus a Detail section which is repeated
5. Similar principles apply in form and report design, although fonts and colours used may be different.