Database Case 9

Country Roads Hotel

Problem: Design a hotel reservation system.

Management skill: Control

Access Skills:
- Forms
- Screen Design
- Macros
- Queries
- Functions

Data Tables:
- ROOMS
- VISITORS
- BOOKINGS

Linda O'Hara settled down at the table with the accounts and reservation book and started the monthly chore of calculating the month's takings at the hotel owned and operated jointly with her husband, Mike. When the Country Roads Hotel in Port Arthur, Texas, was left to Linda by her recently deceased grandparents, Linda and Mike were faced with the decision of whether to attempt to manage the business, or sell the hotel. As recent MBA graduates of the University of Texas at Austin, the couple prepared a complete business plan, as if they were deciding to purchase the business.

Linda found that while the 23-room hotel was profitable, it was experiencing steadily declining rentals. After talking with owners of nearby hotels, Linda discovered that up to 70% of hotel business in the area came from repeat business. Despite this, Country Roads had no list of past customers, either regular or otherwise, and records of past capacity utilization were patchy. In addition, the current booking and reservation system was manually based. Even so, Linda and Mike thought the business had great potential, given a more modern management approach. After a few years of successful operation, they believed there would be enough funds for expansion of the hotel. They decided to go ahead with running the hotel.

Linda contacted one of her Operations professors from UTA, who suggested the reservation and booking system was amenable to computerization. The professor recommended a PC-based database management system for tracking the usage of rooms, automatically establishing a history of customers, and recording a transaction list whereby revenues could be easily calculated. The list of past guests could then be used for mail-outs to remind potential repeat customers of the attractions of Port Arthur and particularly, the Country Roads Hotel.

Linda and Mike were excited at the prospect of owning and managing a business at such an early age. However, managing the hotel and its staff was a full-time responsibility, and they were not confident they could develop the required database system themselves. They contacted the placements office at UTA which directed them to Chet Martin, a first year MBA student and an IS major, proficient in PC database systems. Chet agreed that a simple menu-driven program could be developed to record guests booking in and out of the hotel.

Linda and Mike hired Chet to develop the management system. Chet decided that several data tables would be necessary for the hotel's reservation system. These would be:

1. a list of rooms and their features (ROOMS),
2. a list of past guests and their details (VISITORS),
3. a list of transactions of each guest's stay (BOOKINGS)

In addition, a lookup table ROOMS LOOKUP, would be generated from the ROOMS data table to provide a list of currently available rooms.

Chet believed appropriate options on a menu would be:

1. Registering a guest.
2. Checking out.
3. Listing vacant rooms.
4. Exit the program.

Chet explained how he intended to use these different options to operate the three data tables, using customer name as the common field. For example, when a guest arrived, the person at reception would select Option 1 from the menu. The vacant rooms would then be displayed from the ROOMS LOOKUP table. The receptionist would enter the guest's name and offer a room choice to the guest. Once the choice was made, the ROOMS table would then be amended to show the details of the new guest, their room number and room rate. The program would then check to see whether the guest had visited before by consulting the VISITORS table. If the guest were a newcomer, further details such as address would be entered and added to the VISITORS table. This would eventually build a significant guest list and be a valuable marketing tool.

The second option would be used at the end of a guest's stay. The relevant room in the ROOMS LOOKUP table would be changed to vacant status, and entries in the VISITORS, ROOMS and BOOKINGS tables would be amended or updated to reflect details of the current visit. At the same time a charge would calculated by subtracting the current date from the date of arrival and multiplying the number of days stayed by the room rate.

The third option would enable the user to examine a list of vacant rooms at any time using the ROOMS LOOKUP table. The fourth option would simply exit the hotel reservation system.

You have been supplied with three of the four data tables needed to complete this case: ROOMS, VISITORS and BOOKINGS on your *Solve it!* diskette. The fourth table ROOMS

LOOKUP, should be created within the database package you are using. Load these tables now and examine their structure.

Tasks:

There are five tasks in this case:

1. Create a form, and appropriate command buttons and macros to register a guest at the Country Roads Hotel. This will require the use of the ROOMS and VISITORS tables supplied on the *Solve it!* diskette. Include on your form all appropriate formatting, header information and instructional text.

2. Create a new table (ROOMS LOOKUP), which can be used to look up a list of all vacant rooms, their rate and features. This table will be based on some of the fields in the ROOMS table. Create a command button on the form you created in Task 1, which will allow access to this new table.

3. Create a procedure to check a guest out. This will involve appending a record to the BOOKINGS table from the ROOMS table, before deleting the relevant record from the ROOMS table. The BOOKINGS table should then be updated to calculate total charges owing. Be sure to use parameters for isolating specific records in this procedure.

4. Create a Main Menu form and appropriate command buttons to allow the user to choose from and use any of the four options described in the case. The four menu options should be displayed using forms, which all include an Exit procedure back to the main menu.

*5. Suggest and implement how the O'Haras can incorporate the following improvements:

    A mechanism to distinguish between guests of the same name,
    A future booking system whereby guests may reserve rooms ahead of time for peak periods, and
    A checking system so other guests cannot stay in reserved rooms.

Time Estimates (excluding task marked with *):

Expert: 2 hour
Intermediate: 3 hours
Novice: 5 hours

Tutorial For Database Case 9 Using Access 2000

More About Forms - Main/Subform

A subform is a form embedded inside another form (known as the main form). Main/sub forms are used to generate and display data from any two tables or queries that have a one-to-many relationship. The Main form should be based on the table that's on the "one" side of the relationship, and the subform on the "many" side of the relationship (e.g.: in Case 9, one guest can have many bookings).

The relationship between the two tables must exist before attempting to create the subform. In addition, the main form (the table on the "one side") should have a primary key, while the subform (the table on the "many side") needs to contain a field with the same name and data type as that primary key. Primary keys are usually unique identifiers. You will need to create primary key fields for some of the tables used in Case 9. Recap the section *Creating Relationships Between Tables* in Chapter 4 if you are unsure how to do this.

To create a Main/Subform:

1. Create a new form using the appropriate form wizard or design view option. This action creates the *main form*.
2. In Form Design view, activate the Control Wizards button in the Toolbox
3. Click the Subform/Subreport button in the Toolbox. *(Make sure the relationships between the tables are correct*)
4. Click the location of the subform and then follow the directions of the Subform Wizard to complete the creation of the subform object.

Delete Queries 

Delete queries are a type of action query which remove records from a table according to some specified search criteria. Delete queries are created in the same way as other action queries. Begin by creating a select query, which contains field criteria such as parameter statements for isolating the records targeted for deletion. *Warning: a delete query always deletes entire records, not just the contents of specific fields.*

Choose QUERY/DELETE from the menu or click on the Delete Query toolbar button to turn the select query to a delete query. This action generates display of a Delete: row in the QBE Grid of the query window. Test the query without committing to execution by using the Datasheet View button to preview the targeted records. Execute the query by clicking on the Run toolbar button or selecting QUERY/RUN from the menu. *Warning: Delete queries cannot be undone, so test thoroughly before committing to Run.*

Functions in Access

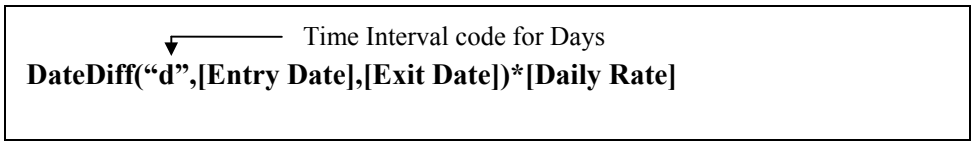
A *function* in Access performs some sort of calculation on data and then returns the result of that calculation. There are over 130 different functions available with Access. Functions can be used in Macros or Modules, in Query expressions, or as calculated controls in Form or Report objects. Case 9 involves using two different types of functions:

Type of Function	**Name of Function**	**Calculation Performed**
Date/Time	DateDiff	Number of time intervals between dates
Domain Aggregate	DLookup	Finds/Looks Up and displays a Value

DateDiff

The *DateDiff* function is used to determine how many time intervals (e.g.: days, weeks, years) exist between two dates. In the following example, the DateDiff function is used to calculate the number of days between an Entry and Exit date at a parking garage. The result of this calculation is then multiplied by the Daily Parking Rate to give a total amount owing by the driver. Required syntax is exactly as shown.

In Case 9, the DateDiff function is used in an Update action query, to firstly determine the number of days a guest has stayed at the hotel, and then to calculate the appropriate charge.



DLookup

DLookup returns a field value from a domain (a specific set of records as defined by a table or query). The following example firstly looks up the Tax Rate in a table called Personnel where data in the Skill field of a table called Staff matches data in the Skill field of the Personnel table. The appropriate Tax Rate is then added to the Tax Rate field in the Personnel table. Required syntax is exactly as shown.

```
DLookup("[Tax Rate]","[Personnel]","[Skill]"=[Staff]![Skill]")
```

In Case 9, the DLookup function is used in a Set Value macro sequence to look up a room rate in the ROOMS LOOKUP table and add it to the Rate field in the ROOMS table, as part of the registration procedure.

More About Macros

The following macro actions may be of assistance in completing the tasks associated with Case 9.

MACRO ACTION	WHAT IT DOES
Close	Closes the current active window
Echo	Hides or shows the result of a macro while it runs
MsgBox	Displays a message box with informational or warning text
Open Form	Opens a specified Form object
Open Query	Opens a select query and/or runs an action query
Set Value	Sets the value for a control field or property on a form or report
Set Warnings	Turns off all system messages such as warnings while a macro is running

Tutorial For Database Case 9 Using Access 97

More About Forms - Main/Subform

A subform is a form embedded inside another form (known as the main form). Main/sub forms are used to generate and display data from any two tables or queries that have a one to many relationship. The Main form should be based on the table that's on the "one" side of the relationship, and the subform on the "many" side of the relationship (e.g.: in Case 9, one guest can have many bookings).

The relationship between the two tables must exist before attempting to create the subform. In addition, the main form (the table on the "one side") should have a primary key, while the subform (the table on the "many side") needs to contain a field with the same name and data type as that primary key. Primary keys are usually unique identifiers. You will need to create primary key fields for some of the tables used in Case 9. Recap the section *Creating Relationships Between Tables* in Chapter 4 if you are unsure how to do this.

To create a Main/Subform:

1. Create a new form using the appropriate form wizard or design view option. This action creates the *main form*.
2. In Form Design view, activate the Control Wizards button in the Toolbox
3. Click the Subform/Sub report button in the Toolbox. *(Make sure the relationships between the tables are correct)*
4. Click the location of the subform and then follow the directions of the Subform Wizard to complete the creation of the subform object.

Delete Queries 

Delete queries are a type of action query, which remove records from a table according to some specified search criteria. Delete queries are created in the same way as other action queries. Begin by creating a select query, which contains field criteria such as parameter statements for isolating the records targeted for deletion. *Warning: a delete query always deletes entire records, not just the contents of specific fields.*

Choose QUERY/DELETE from the menu or click on the Delete Query toolbar button to turn the select query to a delete query. This action generates display of a Delete: row in the QBE Grid of the query window. Test the query without committing to execution by using the Datasheet View button to preview the targeted records. Execute the query by clicking on the Run toolbar button or selecting QUERY/RUN from the menu. *Warning: Delete queries cannot be undone, so test thoroughly before committing to Run.*

Functions in Access

A *function* in Access performs some sort of calculation on data and then returns the result of that calculation. There are over 130 different functions available with Access. Functions can be used in Macros or Modules, in Query expressions, or as calculated controls in Form or Report objects. Case 9 involves using two different types of functions:

Type of Function	**Name of Function**	**Calculation Performed**
Date/Time	DateDiff	Number of time intervals between dates
Domain Aggregate	DLookup	Finds/Looks Up and displays a Value

DateDiff

The *DateDiff* function is used to determine how many time intervals (e.g.: days, weeks, years) exist between two dates. In the following example, the DateDiff function is used to calculate the number of days between an Entry and Exit date at a parking garage. The result of this calculation is then multiplied by the Daily Parking Rate to give a total amount owing by the driver. Required syntax is exactly as shown.

In Case 9, the DateDiff function is used in an Update action query, to firstly determine the number of days a guest has stayed at the hotel, and then to calculate the appropriate charge.

Time Interval code for Days

DateDiff("d",[Entry Date],[Exit Date])*[Daily Rate]

DLookup

DLookup returns a field value from a domain (a specific set of records as defined by a table or query). The following example firstly looks up the Tax Rate in a table called Personnel where data in the Skill field of a table called Staff matches data in the Skill field of the Personnel table. The appropriate Tax Rate is then added to the Tax Rate field in the Personnel table. Required syntax is exactly as shown.

DLookup(“[Tax Rate]”,”[Personnel]”,”[Skill]”=[Staff]![Skill]”)

In Case 9, the DLookup function is used in a Set Value macro sequence to look up a room rate in the ROOMS LOOKUP table and add it to the Rate field in the ROOMS table, as part of the registration procedure.

More About Macros

The following macro actions may be of assistance in completing the tasks associated with Case 9.

MACRO ACTION	WHAT IT DOES
Close	Closes the current active window
Echo	Hides or shows the result of a macro while it runs
MsgBox	Displays a message box with informational or warning text
Open Form	Opens a specified Form object
Open Query	Opens a select query and/or runs an action query
Set Value	Sets the value for a control field or property on a form or report
Set Warnings	Turns off all system messages such as warnings while a macro is running