Database Case 7

Feldspar Contractors

Problem:	Develop a payroll system
Management Skill:	Control
Access skills:	Select Queries (calculated fields) Append Queries Reports (wizards)

Data Tables:
CONTRACT
1_MONTH
2_MONTH
3_MONTH

After twelve years working for a general contractor, Martin Feldspar struck out on his own in 1990 to start his own road construction firm, Feldspar Contractors. For a start-up firm, Feldspar had done extremely well: in 2000 they showed gross revenues of $15 million and a net profit of $950,000.

Before starting his own company, Martin had gained a sound understanding of the road building and public sector market working as an estimator for a general contractor in Boston, Massachusetts. He built up a network of contacts among local government officials, and learned the procedures and pitfalls of responding to government contract solicitations. Fortunately, the local economy grew very rapidly as major corporations, government agencies and related housing created a boom market for road construction.

The key to Martin's success has been in keeping permanent employee numbers low, relying on sub-contractors, and keeping his capital costs very low by renting or leasing heavy equipment. Automated tools have also played an important role.

About 80% of Feldspar's business is in road construction. Feldspar responds to government agency solicitations for bids on construction projects. Feldspar, and other firms, prepare competitive proposals and tender bids. A large part of the bid preparation process, especially the design, cost estimation and technical specification, is aided using a proprietary decision support system (DSS) written for the road construction industry. The DSS runs on a Pentium PC in Feldspar's office.

Payroll and personnel--along with other office administrative tasks--are somewhat more chaotic. There are only 5 full time employees (Martin, two engineers, and two office staff). However, about 40 sub-contractors work for the firm during a typical monthly pay-roll period. More than 400 individuals will work for the firm in a year.

Currently, one of the office staff spends 80% of her time keeping track of the firm’s part time and full time employees. Originally Martin had considered outsourcing his payroll, but this

proved far too expensive an option. Martin is convinced the answer lies in building a simple in-house personnel system to handle employee wages, and related payroll information.

Last year, the situation became serious when the Internal Revenue Service (IRS) audited the firm's books. The IRS wondered why Feldspar had more than 400 employees but showed very low tax withholdings. Martin explained that most of the employees were short term, part time workers and/or sub-contractors. In recent times the IRS has taken a dim view of employees being declared as "contract workers" because federal tax is not withheld for contract workers and it is difficult for the IRS to collect from them later.

While Feldspar's contract workers appeared to be legitimate, examination of the records found a number of errors in withholding statements for regular employees who were part time workers. For 35 part time workers, no withholding was deducted or submitted to the IRS because of clerical errors. The IRS has given Martin one year to straighten out the records.

Martin has decided that a PC-based database system would be ideal for a small payroll system. He would like you to design such a system for a sample quarter. Employees are paid monthly.

A sample of the employee database showing the name, address, ID number, and age can be seen by loading the CONTRACT data table from your *SolveIt!* diskette. For each monthly payroll period a monthly file is created. Three hand calculated files for the first quarter have been built for you in Access: 1_MONTH, 2_MONTH, and 3_MONTH.

Tasks There are five tasks in this case:

1. Combine the three individual monthly reports (i.e.: 1_MONTH, 2_MONTH, 3_MONTH) into a single data table representing the first quarter.

   Hint: Use Append and Action queries to do this. You should end up with 26 records in a single, new data table.

2. (a) Develop a query, which will calculate the gross, pay, federal and state withholdings, FICA, and resulting net pay for all employees in the first quarter.

   Set federal withholdings at 17.4% of gross, FICA at 2.7% of gross, state withholding at 8% of gross, and calculate net pay as gross pay minus all deductions.

   (b) Now use this query to create a report, which shows the gross pay, all deductions, and net pay for all employees for the quarter.

3. Develop a payroll report, which shows just the firm's grand totals for the quarter for gross pay, all the deductions and net pay.

4. Develop a report which shows for each employee the earnings and deductions for the quarter, giving subtotals by employee. At the end, the report should also show firm totals.

*5. Produce a report for the IRS for the first quarter, which shows the employees' name, address, all deductions and net pay.

You will need the data table CONTRACT on the *Solve it!* diskette to do this task. This table contains the names and addresses of Feldspar's employees.

Time Estimates (excluding task marked with *):

Expert: 1.5 hour
Intermediate: 2.5 hours
Novice: 4+ hours

Tutorial For Database Case 7 Using Access 2000

Copying and Pasting Table Structures

From the Database window, you can create an empty copy of an existing table (i.e.: field structure only), as a first step in merging records from a number of different tables to a single table.

Let's practice this by creating an empty copy of the FRIENDS table:

1. Load FRIENDS.MDB. From the Database window, click to highlight the FRIENDS table.
2. Select EDIT/COPY from the menu.
3. Select EDIT/PASTE from the menu.
4. From the *Paste Tables As* window, give the table a new name (e.g.: Tute7 CopyTable)
5. Under the Paste Options section, click on *Structure Only* and then click OK. Access creates an empty table with an identical field structure to the FRIENDS table.

Action Queries

When you create a new query via the Simple Query Wizard or Design View options, Access generates a Select query by default. *Select* queries simply retrieve and display data from tables according to a specified user criteria. We have used Select queries for *Solve it!* database cases 1 to 6. In contrast, *Action* queries, which are constructed within the Select query window, actually alter data in tables. Action queries can be used to add, delete, or change data, and to create new tables from existing records. For example, a Delete action query removes obsolete records from a table. Access uses four types of action queries. A brief description of each type and the graphical objects that identify them within the database envelope, is shown in Figure 5-77.

In this tutorial, we will concentrate on *Append* action queries. Other Action query types will be covered in later tutorials.

Figure 5-77

Types of Queries

Access query types include:

Select queries select a group of records from one or more tables.

Action Queries

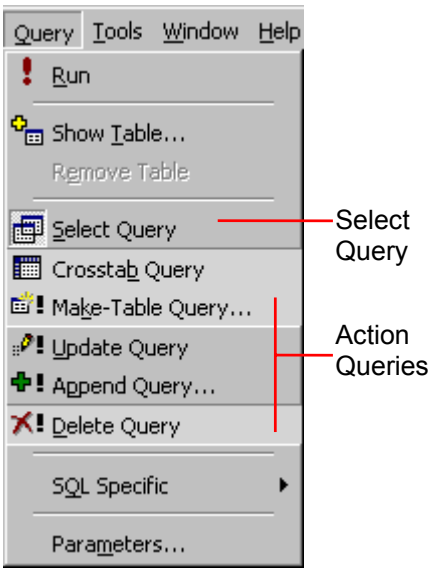
The following action queries change the data in your tables:

Make-table queries create a new table from all or parts of other tables.

Delete queries delete records from one or more tables.

Append queries add a group of records to a table.

Update queries make changes to data in a group of records.



Append Queries

Append queries copy all or some records from one table to the bottom of another existing table. This is especially useful if you use separate tables to manage certain data (e.g.: you keep payroll data on a month by month basis, and need to merge it every financial quarter). Append queries are also handy for storing historical data. For instance in an Orders table, completed orders could be separated from active, uncompleted orders to prevent the Orders table from becoming too large, and to provide a useful backup for future reference.

When data is appended, the tables involved in the query do not need to have the same structure, but the data types of the appended fields must match. (The exception is Counter data types, which may be appended to Long Integer data types). When records are appended to a table, the records in the original table remain intact. Access does not delete the original records.

Note:	if two tables involved in an Append query have identical field structures, the * in the table field list can be used instead of the individual field names in the Field: bar of the QBE Grid

Let's use the FRIENDS table and the empty copy of the FRIENDS table (e.g.: TUTE7 COPYTABLE) we created earlier in this tutorial to generate an Append action query.

1. From the Database window, create a new query (using the Design View option) based on the FRIENDS table.

2. From the query design window, select QUERY/APPEND from the menu or click on the Append query toolbar button. The results of this action will turn the Select query into an Append query, and cause Access to add an *Append to:* line to the QBE Grid.

3. In the *Append* dialog window, select and set the *Append to Table Name* box to the TUTE7 COPYTABLE we created earlier, and click OK. This means that we are going to append records from the FRIENDS table to the TUTE7 COPYTABLE.

4. From the Append query design window, select and drag the ✳ symbol in the FRIENDS field list down onto the Field: bar of the QBE Grid. This tells Access two things: that all FRIENDS fields should be included in the query, and that the field structure of TUTE7 COPYTABLE is identical to FRIENDS. Access immediately adds notation to this effect in the *Append to:* cell.

5. Test the Append query before committing to its execution. One of the things to note about any Action query is that it changes table data in some way. For this reason, it is good practice to trial the query before running it. Trialling also allows you to check for errors in the query setup.

To trial your Append query, click on the button. The resulting dynaset should display all the records in the FRIENDS table. Return to query design view, and run the query. Access should append all FRIENDS table records to TUTE7 COPYTABLE.

6. Save your query (e.g.: TUTE7 APPEND QUERY), and return to the Database window. Open the TUTE7 COPYTABLE and check the contents.

Hint:	You will need to use both the Copy/Paste table structure sequence and Append queries to complete Task 2 of Case 7.

Tutorial For Database Case 7 Using Access 97

Copying and Pasting Table Structures

From the Database window, you can create an empty copy of an existing table (i.e.: field structure only), as a first step in merging records from a number of different tables to a single table.

Let's practice this by creating an empty copy of the FRIENDS table:

1. Load FRIENDS.MDB. From the Database window, click to highlight the FRIENDS table.
2. Select EDIT/COPY from the menu.
3. Select EDIT/PASTE from the menu.
4. From the *Paste Tables As* window, give the table a new name (e.g.: Tute7 CopyTable)

5. Under the Paste Options section, click on *Structure Only* and then click OK. Access creates an empty table with an identical field structure to the FRIENDS table.

Action Queries

When you create a new query via the Simple Query Wizard or Design View options, Access generates a Select query by default. *Select* queries simply retrieve and display data from tables according to a specified user criteria. We have used Select queries for *Solve it!* database cases 1 to 6. In contrast, *Action* queries, which are constructed within the Select query window, actually alter data in tables. Action queries can be used to add, delete, or change data, and to create new tables from existing records. For example, a Delete action query removes obsolete records from a table. Access uses four types of action queries. A brief description of each type and the graphical objects that identify them within the database envelope is shown in Figure 5-78. In this tutorial, we will concentrate on *Append* action queries. Other Action query types will be covered in later tutorials.

Figure 5-78

Types of Queries

Access query types include:

Select queries select a group of records from one or more tables.

Action Queries

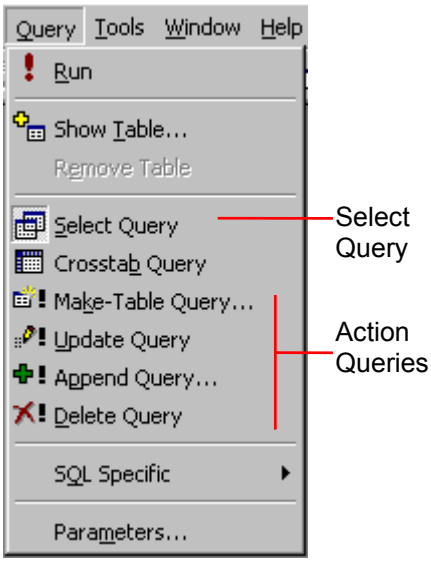
The following action queries change the data in your tables:

Make-table queries create a new table from all or parts of other tables.

Delete queries delete records from one or more tables.

Append queries add a group of records to a table.

Update queries make changes to data in a group of records.



Append Queries

Append queries copy all or some records from one table to the bottom of another existing table. This is especially useful if you use separate tables to manage certain data (e.g.: you keep

payroll data on a month by month basis, and need to merge it every financial quarter). Append queries are also handy for storing historical data. For instance in an Orders table, completed orders could be separated from active, uncompleted orders to prevent the Orders table from becoming too large, and to provide a useful backup for future reference.

When data is appended, the tables involved in the query do not need to have the same structure, but the data types of the appended fields must match. (The exception is Counter data types, which may be appended to Long Integer data types). When records are appended to a table, the records in the original table remain intact. Access does not delete the original records.

Note:	if two tables involved in an Append query have identical field structures, the * in the table field list can be used instead of the individual field names in the Field: bar of the QBE Grid

Let's use the FRIENDS table and the empty copy of the FRIENDS table (e.g.: TUTE7 COPYTABLE) we created earlier in this tutorial to generate an Append action query.

1. From the Database window, create a new query (using the Design View option) based on the FRIENDS table.

2. From the query design window, select QUERY/APPEND from the menu or click on the Append query toolbar button. The results of this action will turn the Select query into an Append query, and cause Access to add an *Append to:* line to the QBE Grid.

3. In the *Append* dialog window, select and set the *Append to Table Name* box to the TUTE7 COPYTABLE we created earlier, and click OK. This means that we are going to append records from the FRIENDS table to the TUTE7 COPYTABLE.

4. From the Append query design window, select and drag the ✳ symbol in the FRIENDS field list down onto the Field: bar of the QBE Grid. This tells Access two things: that all FRIENDS fields should be included in the query, and that the field structure of TUTE7 COPYTABLE is identical to FRIENDS. Access immediately adds notation to this effect in the *Append to:* cell.

5. Test the Append query before committing to its execution. One of the things to note about any Action query is that it changes table data in some way. For this reason, it is good practice to trial the query before running it. Trialling also allows you to check for errors in the query setup.

To trial your Append query, click on the button. The resulting dynaset should display all the records in the FRIENDS table. Return to query design view, and run the query. Access should append all FRIENDS table records to TUTE7 COPYTABLE.

6. Save your query (e.g.: TUTE7 APPEND QUERY), and return to the Database window. Open the TUTE7 COPYTABLE and check the contents.

Hint:	You will need to use both the Copy/Paste table structure sequence and Append queries to complete Task 2 of Case 7.