Chapter 7

How to insert, update, and delete data

Exercises

1. Write an INSERT statement that adds a row to the Departments table for a department named “History”.

   (Hint: Remember that you do not need to specify a value for identity columns.)

2. Write a single INSERT statement that adds two rows to the Instructors table with the following values:

InstructorID:	The next automatically generated ID
LastName:	Benedict
FirstName:	Susan
Status:	P
DepartmentChairman:	0
HireDate:	The current date
AnnualSalary:	34000.00
DepartmentID:	9

InstructorID:	The next automatically generated ID
LastName:	Adams
FirstName:	null
Status:	F
DepartmentChairman:	1
HireDate:	The current date
AnnualSalary:	66000.00
DepartmentID:	9

   The statement should not use a column list.

3. Write an UPDATE statement that changes the AnnualSalary column for Susan Benedict (the first instructor you added in exercise 2) from 34,000 to 35,0000. To identify the row, use the InstructorID column.

4. Write a DELETE statement that deletes Adams (the second instructor you added in exercise 2). To identify the row, use the InstructorID column.

5. Write a DELETE statement that deletes the row in the Departments table that has an ID of 9. Then, run the statement. It will produce an error since the department has related rows in the Instructors table. To fix that, precede the DELETE statement with another DELETE statement that deletes all instructors in this department from the Instructors table.

6. Write an UPDATE statement that increases the annual salary for all instructors in the Education department by 5%. (Hint: join the Departments and Instructors tables and then filter the rows by the department name.)

7. Write a DELETE statement that deletes instructors who aren’t teaching any courses. To do that, use a subquery in the WHERE clause.

8. Open the script named CreateGradStudents.sql that’s in the MC Exercise Starts directory. Run this file to create a table named GradStudents. This table has the same columns as the Students table, but the StudentID column isn’t defined as an identity column.

Make sure that when you run the script, the active database is set to MurachCollege.

9. Write an INSERT statement that inserts rows from the Students table into the GradStudents table. Include only the rows for students who have graduated, and don't use a column list.
10. Open the script named CreateMurachCollege.sql that's in the MC Exercise Starts directory. Then, run this script. That should restore the data that's in the database. If an error message is displayed indicating that the database is in use, you'll need to close and restart Management Studio and then run the script again.