Chapter 5

How to insert, update, and delete data

Exercises

To test whether a table has been modified correctly as you do these exercises, you can write and run an appropriate SELECT statement.

1. Write an INSERT statement that adds this row to the Categories table:

category\_name: Brass

Code the INSERT statement so MySQL automatically generates the category\_id column.

1. Write an UPDATE statement that modifies the row you just added to the Categories table. This statement should change the category\_name column to “Woodwinds”, and it should use the category\_id column to identify the row.
2. Write a DELETE statement that deletes the row you added to the Categories table in exercise 1. This statement should use the category\_id column to identify the row.
3. Write an INSERT statement that adds this row to the Products table:

product\_id: The next automatically generated ID   
category\_id: 4  
product\_code: dgx\_640  
product\_name: Yamaha DGX 640 88-Key Digital Piano  
description: Long description to come.  
list\_price: 799.99  
discount\_percent: 0  
date\_added: Today’s date/time.

Use a column list for this statement.

1. Write an UPDATE statement that modifies the product you added in exercise 4. This statement should change the discount\_percent column from 0% to 35%.
2. Write a DELETE statement that deletes the Keyboards category. When you execute this statement, it will produce an error since the category has related rows in the Products table. To fix that, precede the DELETE statement with another DELETE statement that deletes all products in this category. (Remember that to code two or more statements in a script, you must end each statement with a semicolon.)
3. Write an INSERT statement that adds this row to the Customers table:

email\_address: rick@raven.com  
password: (empty string)  
first\_name: Rick  
last\_name: Raven

Use a column list for this statement.

1. Write an UPDATE statement that modifies the Customers table. Change the password column to “secret” for the customer with an email address of rick@raven.com.
2. Write an UPDATE statement that modifies the Customers table. Change the password column to “reset” for every customer in the table. If you get an error due to safe-update mode, you can add a LIMIT clause to update the first 100 rows of the table. (This should update all rows in the table.)
3. Open the script named create\_my\_guitar\_shop.sql that’s in the CIS243 SQL-Files folder on the desktop. Then, run this script. That should restore the data that’s in the database.