Murach 4e Chapter 5 Coding Assignment Instructions

Exercises to be Completed

Complete exercises as follows:

- Exercise 1 through 8 are regular exercises.
- Exercise 9 is the challenge exercise.
- Note that Exercise 10 is not a proper exercise but rather a reminder to reset the database after you finish with exercises 1 through 9.

General Instructions

My expectations for your work on coding assignment exercises will grow as we progress through the course. In addition to applying any new coding techniques that have been covered in the current chapter, I will be expecting you to follow all of the good practices that we have adopted in the preceding weeks. Here is a quick summary of good practices that we have covered so far:

- Begin each script file that accesses the database with a USE statement (e.g., USE my_guitar_shop;).
- Use the *beautify* feature of the MySQL Workbench to *pretty-print* your code.
- End each statement in your script with a semicolon.
- Use the SQL features requested in the exercise description and/or covered in the chapter.
- Always include an ORDER BY in SELECT statements unless directed otherwise. If the exercise instructions ask for a particular order, then use that. Otherwise, choose any reasonable order.
- In SELECT statements that use JOIN, always use the explicit (SQL-92) JOIN syntax implemented in the FROM clause. Do NOT use the implicit JOIN syntax implemented using the WHERE clause.
- Do NOT include extra or unnecessary code in the script.

Tools

Use MySQL Workbench to create and test all scripts.

Submission Method

Use the following process to submit your work for this assignment:

- Locate the properly named directory associated with your assignment in the file system (see *File and Directory Naming,* below).
- Compress that directory into a single .ZIP file using a utility program. NOTE: Only one file may be submitted. File types other than .ZIP will not be graded.
- Submit the properly named .ZIP file to the submission activity for this assignment.

File and Directory Naming

Please note that file and directory names must be in all lower case. Deductions will be made for submissions that do not follow this standard.

Please use the following naming scheme for the directory that holds your scripts:

```
surname_givenname_mgs_chap_05
```

If this were my own project, I would name my scripts directory as follows:

```
trainor_kevin_mgs_chap_05
```

A separate solution script file must be submitted for each exercise. Solution scripts must be named using the following form: ex_xx_yy.sql (where xx is the two-digit chapter number [04] and yy is the two-digit exercise number [01]). So, an example of a properly formed solution script file example would be:

```
ex_05_01.sql
```

Use a zip utility to create one zip file that contain the scripts directory. The zip file should be named according to the following scheme:

```
surname_givenname_mgs_chap_05.zip
```

If this were my own project, I would name the zip file as follows:

```
trainor_kevin_mgs_chap_05.zip
```

Due By

Please submit this assignment by the date and time shown in the Weekly Schedule.

Last Revised

2025-02-06

Please see the exercises on the attached sheets!

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.

2. Write an UPDATE statement that modifies the row you just added to the Categories table. This statement should change the category column to "Woodwinds", and it should use the category_id column to identify the row.

3. 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.

4. 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 and do not specify values for columns that have default values.

5. 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%.

6. 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.)

7. 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.

8. 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.
9. [*Challenge Exercise*] 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.)
10. Open the script named create_my_guitar_shop.sql that’s in the mgs_ex_starts directory. Then, run this script. That should restore the data that’s in the database.