Murach 3e Chapter 12 Coding Assignment Instructions

Exercises to be Completed

Please complete Exercises 1 – 6. You will find those exercises on the attached sheet(s).

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 (ANSI) JOIN syntax implemented in the FROM clause. Do NOT use the implicit JOIN syntax implemented using the WHERE clause.
- When testing SELECT statements that use summary functions, always test with the ONLY_FULL_GROUP_BY setting set to ON.
- 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 accepted and will receive a grade of zero.
- 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_12
```

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

```
trainor_kevin_mgs_chap_12
```

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_12_01.sql
```

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

```
surname_givenname_mgs_chap_12.zip
```

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

```
trainor_kevin_mgs_chap_12.zip
```

Due By

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

Last Revised

2021-03-26

Please see the exercises on the attached sheets

Chapter 12

How to create views

Exercises

1. Create a view named customer_addresses that shows the shipping and billing addresses for each customer.

   This view should return these columns from the Customers table: customer_id, email_address, last_name and first_name.

   This view should return these columns from the Addresses table: bill_line1, bill_line2, bill_city, bill_state, bill_zip, ship_line1, ship_line2, ship_city, ship_state, and ship_zip.

   Hint: join the addresses table to the customers table twice. Once to get the billing address, then again to get the shipping address.

2. Write a SELECT statement that returns these columns from the customer_addresses view that you created in exercise 1: customer_id, last_name, first_name, bill_line1. The rows in the result should be sorted by the last_name and then first_name columns.

3. Write an UPDATE statement that updates the Customers table using the customer_addresses view you created in exercise 1. Set the first line of the shipping address to "1990 Westwood Blvd." for the customer with an ID of 8.

4. Create a view named order_item_products that returns columns from the Orders, Order_Items, and Products tables.

   This view should return these columns from the Orders table: order_id, order_date, tax_amount, and ship_date.

   This view should return the product_name column from the Products table.

   This view should return these columns from the Order_Items table: item_price, discount_amount, final_price (the discount amount subtracted from the item price), quantity, and item_total (the calculated total for the item).

5. Create a view named product_summary that uses the view you created in exercise 4. This view should return summary information about each product.

   Each row should include product_name, order_count (the number of times the product has been ordered) and order_total (the total sales for the product).

6. Write a SELECT statement that uses the view that you created in exercise 5 to get total sales for each of the five *best selling products*. Sort the result set by the order_total column in descending sequence. In this context, *best selling products* means the products with the highest dollar amount of total sales.