Murach 4e Chapter 9 Coding Assignment Instructions

Exercises to be Completed
Complete exercises as follows:

- Exercise 1 through 6 are regular exercises.
- Exercises 7 is a challenge exercise.

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_09`

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

`trainor_kevin_mgs_chap_09`

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_09_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_09.zip`

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

`trainor_kevin_mgs_chap_09.zip`

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

Last Revised
2025-04-07

Please see the exercises on the attached sheets!

Chapter 9
How to use functions

Exercises

1. Write a SELECT statement that returns these columns from the Products table:

   The list_price column

   The discount_percent column

   A column named discount_amount that uses the previous two columns to calculate the discount amount and uses the ROUND function to round the result so it has 2 decimal digits

2. Write a SELECT statement that returns these columns from the Orders table:

   The order_date column

   A column that uses the DATE_FORMAT function to return the four-digit year that's stored in the order_date column

   A column that uses the DATE_FORMAT function to return the order_date column in this format: Mon-DD-YYYY. In other words, use abbreviated months and separate each date component with dashes.

   A column that uses the DATE_FORMAT function to return the order_date column with only the hours and minutes on a 12-hour clock with an am/pm indicator

   A column that uses the DATE_FORMAT function to return the order_date column in this format: MM/DD/YY HH:MM. In other words, use two-digit months, days, and years and separate them by slashes. Use 2-digit hours and minutes on a 24-hour clock. And use leading zeros for all date/time components.

3. Write a SELECT statement that returns these columns from the Orders table:

   The card_number column

   The length of the card_number column

   When you get that working right, add the columns that follow to the result set. This is more difficult because these columns require the use of functions within functions.

   The last four digits of the card_number column

   A column that displays an X for each digit of the card_number column except for the last four digits. If the card number contains 16 digits, it should be displayed in this format: XXXX-XXXX-XXXX-1234, where 1234 are the actual last four digits of the number. If the card number contains 15 digits, it should be displayed in this format: XXXX-XXXXXX-X1234. *(Hint: Use an IF function to determine which format to use.)*

4. Write a SELECT statement that returns these columns from the Orders table:

   The order_id column

   The order_date column

   A column named approx_ship_date that's calculated by adding 2 days to the order_date column

   The ship_date column if it doesn't contain a null value

   A column named days_to_ship that shows the number of days between the order date and the ship date

   When you have this working, add a WHERE clause that retrieves just the orders for March 2022.

5. Write a SELECT statement that uses regular expression functions to get the username and domain name parts of the email addresses in the Administrators table. Return these columns:

   The email_address column

   A column named user_name that contains the username part of the email_address column (the part before the @ symbol)

   A column named domain_name that contains the domain name part of the email_address column (the part after the @ symbol)

   Note: The username part of the email addresses contains only letters, and the domain name part contains only letters and a period.

6. Write a SELECT statement that uses the ranking functions to rank products by the total quantity sold. Return these columns:

   The product_name column from the Products table

   A column named total_quantity that shows the sum of the quantity for each product in the Order_Items table

   A column named rank that uses the RANK function to rank the total quantity in descending sequence

   A column named dense_rank that uses the DENSE_RANK function to rank the total quantity in descending sequence

7. [*Challenge Exercise*] Write a SELECT statement that uses the analytic functions to get the highest and lowest sales by product within each category. Return these columns:

   The category_name column from the Categories table

   The product_name column from the Products table

   A column named total_sales that shows the sum of the sales for each product with sales in the Order_Items table

   A column named highest_sales that uses the FIRST_VALUE function to show the name of the product with the highest sales within each category

   A column named lowest_sales that uses the LAST_VALUE function to show the name of the product with the lowest sales within each category