

Murach 4e Chapter 4 Coding Assignment Instructions

Exercises to be Completed

Complete exercises as follows:

- Exercise 1 through 6 are regular exercises.
- Exercise 7 is the 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_givename_mgs_chap_04

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

trainor_kevin_mgs_chap_04

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_04_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_givename_mgs_chap_04.zip

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

trainor_kevin_mgs_chap_04.zip

Due By

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

Last Revised

2025-02-03

Please see the exercises on the attached sheets!

Chapter 4

How to retrieve data from two or more tables

Exercises

1. Write a SELECT statement that joins the Categories table to the Products table and returns these columns: category_name, product_name, list_price.

Sort the result set by the category_name column and then by the product_name column in ascending sequence.
2. Write a SELECT statement that joins the Customers table to the Addresses table and returns these columns: first_name, last_name, line1, city, state, zip_code.

Return one row for each address for the customer with an email address of allan.sherwood@yahoo.com.
3. Write a SELECT statement that joins the Customers table to the Addresses table and returns these columns: first_name, last_name, line1, city, state, zip_code.

Return one row for each customer, but only return addresses that are the shipping address for a customer.

Hint: You can disregard the foreign key on the Addresses table that points to the Customers table. Instead, focus on the billing_address and shipping_address columns on the Customers table. These can both be used as ad hoc foreign keys that point to the Addresses table. Because there are two of them, they don't have the exact same names as the primary key on the Addresses table. Instead, they have an added prefix in their name that reflects their context.
4. Write a SELECT statement that joins the Customers, Orders, Order_Items, and Products tables. This statement should return these columns: last_name, first_name, order_date, product_name, item_price, discount_amount, and quantity.

Use aliases for the tables.

Sort the final result set by the last_name, order_date, and product_name columns.
5. Write a SELECT statement that returns the product_name and list_price columns from the Products table.

Return one row for each product that has the same list price as another product.
Hint: Use a self-join to check that the product_id columns aren't equal but the list_price columns are equal.

Sort the result set by the product_name column.

7 My Guitar Shop Exercises for *Murach's MySQL (4th Edition)*

6. Write a SELECT statement that returns these two columns:

category_name The category_name column from the Categories table

product_id The product_id column from the Products table

Return one row for each category that has never been used. *Hint: Use an outer join and only return rows where the product_id column contains a null value.*

7. [Challenge Exercise] Use the UNION operator to generate a result set consisting of three columns from the Orders table:

ship_status A calculated column that contains a value of SHIPPED or NOT SHIPPED

order_id The order_id column

order_date The order_date column

If the order has a value in the ship_date column, the ship_status column should contain a value of SHIPPED. Otherwise, it should contain a value of NOT SHIPPED.

Sort the final result set by the order_date column.