

Lab 5: Data from Multiple Tables (8% of total grade)

Submission: Use the included .sql file to put your answers in, then upload only the SQL file to Blackboard (Assessments > Lab 5 - Multiple Tables).

Name your file: **HTTP5126-L5-MultipleTables-LastNameFirstName.SQL**, replace *LastNameFirstName* with your name as displayed in Blackboard.

Purpose: To practice data retrieval from multiple tables by joining tables in result sets.

Requirements: For this assignment, you will use the updated Pet Store data tables.

NOTE: Run your queries on your database to make sure desired results are retrieved. Also import and execute your sql file to ensure it runs all your queries before submitting.

Pre-Lab:

1. Start your mySQL server and open phpMyAdmin or Adminer.
2. Create a database for this lab (eg. http5126_lab5). Set the collation as utf8_unicode_ci.
3. Select the DB then import and execute the SQL file 'lab5_pet_store_tables.sql'. This should create 3 tables in your database (employee, stock_item, & sale) which will be used for the queries you create below.

Part 1: What are the sales for a particular item? (1%)

- A. Select all of the rows from the sale table with an stock item id value of 1014.
- B. Select date (from sale), item id (from stock_item) to get the items id with a value of 1014.

Part 2: What are the sales for a particular team member? (1%)

- A. Select all of the rows from the sale table with an employee id value of 111.
- B. Select date (from sale), first and last name (from employee), item id (from sale) to show the sales for the employee with an id of 111.

Part 3: Recent sales (1%)

- A. List the date (from sale) and item name (from stock_item) for the last 5 items sold. Ensure your results show the most recent sale first.
- B. For the 5 most recent sales, display the first and last name of the employee (from employee) who processed them. Also include the date of the sale (from sale) ordered from most recent to least recent. And the name and price of the item (from stock_item).

Part 4: What a week! and Go Team! (2%)

- A. Manager: "I need to review sales activity for a specific period . Can you provide me with a list of all sales made between January 12 and January 18 (inclusive) of this year? The report should include the date of each sale, the item sold, and the first name of the employee who processed the transaction."
- B. Manager: "I want to see how well each salesperson is performing. Can you provide a report that shows the total number of sales made by each employee, grouped by their first and last name? The results should be ordered from the highest number of sales to the lowest."

Part 5: Challenge (2%)

- A. Manager: "I want to take a closer look at our top-performing salesperson. For all sales made by the top-seller, provide the date of the sale, the item name, its price, and category, along with the employee's first name." *Use aliases to rename your tables so your query is shorter to write. For this and all future labs you may use these table aliases in queries if you would like to.*
- B. Manager: "I need a report that shows the 'Times Sold' for each item. Please include the item ID, item name, price, and category for each item, even if it hasn't been sold. The results should be ordered by item ID."

Part 6: Challenge Yourself (1%)

Create your own "human focused question" with the data set provided. Then write a query to answer the question. The question should be written as if you are either a customer of the store or an employee of the store (Manager, Sales Associate, Stock Person). Refer to the way some of the questions are written in this and earlier labs for an idea of how to write your "human focused question".

Your query should use at least 1 **JOIN** and should be **ordered** so data is easy to parse. Preferably you should use an **aggregate function** to make your query more interesting.

- A. Write the question you have created next to the comment on line 35 of the provided sql file. eg. "-- A <your question here>".
- B. Create the query that returns the result your question asks for.