HTTP 5126 – Database Assignment 5.24SS

# **Full Name : Reshmi Harikumar Lathakumari**

**Section : A**

**Student # : N01614330**

**Date : June 5, 2024**

# Assignment 5-Retrieving Data from Multiple Tables (4% of total grade)

**Format:** Add your query in text and screenshot of your results table. Upload your assignment to Blackboard (folder : MySQL Assignments & Quizzes).

BE SURE TO PUT YOUR NAME IN THE FILENAME: e.g. **DB-Assign5-*YourName*.doc/pdf**

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

**Requirements:** For this assignment, you will use the provided Pet Store data tables. **NOTE: a new *sales* table has been added.**

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

1. Select *date* (from *sales*), *item* (from *stock*\_*items*) to get the items with a value of 1014.

**SELECT s.date, si.item**

**FROM sales s**

**INNER JOIN stock\_items si ON s.item = si.id**

**WHERE s.item = 1014;**

A screenshot of a computer

Description automatically generated

1. Find the employee numbers who sold item cat (feline) water dish. Include employee number, item number, category and item.

**SELECT s.employee, s.item, si.category, si.item**

**FROM sales s**

**INNER JOIN stock\_items si ON s.item = si.id**

**WHERE si.category = 'Feline ' AND si.item = 'Water dish';**

A screenshot of a computer

Description automatically generated

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

1. Select *date* (from *sales*), first and last name (from employees), *item* (from *sales*) to show the sales from the employee with a value of 111.

**SELECT s.date, e.first\_name, e.last\_name, s.item**

**FROM sales s**

**INNER JOIN employees e ON s.employee = e.id**

**WHERE e.id = 111;**

**A screenshot of a computer

Description automatically generated**

1. Find all sales by employees who have a SIN number that starts with either 258, 456 or 753. Include employee name (in one column), SIN, role, and item. Use right join.

**SELECT CONCAT(e.first\_name, ' ', e.last\_name) AS 'employee name', e.sin, e.role, s.item**

**FROM sales s**

**RIGHT JOIN employees e ON s.employee = e.id**

**WHERE e.sin LIKE '258%' OR e.sin LIKE '456%' OR e.sin LIKE '753%';**

A screenshot of a computer

Description automatically generated

A group of people in a row

Description automatically generated

## Part 3: What a week! and Go Team! (1%) sat 12 - fri 18

1. Provide the date, item (from sales), and employee first name for all sales in the range of Saturday June 12 to Friday June 18 (inclusive) of this year.

**SELECT s.date, s.item, e.first\_name**

**FROM sales s**

**INNER JOIN employees e ON s.employee = e.id**

**WHERE s.date BETWEEN '2021-06-12' AND '2021-06-18';**

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. Provide the count of all sales for each sales person (first name and last name) grouped by sales person. The results should show most to least number of sales for all employees – even those who have no sales.

**SELECT CONCAT(e.first\_name, ' ', e.last\_name) AS 'sales person', COUNT(s.id) AS sales\_count**

**FROM employees e**

**LEFT JOIN sales s ON e.id = s.employee**

**GROUP BY e.first\_name, e.last\_name**

**ORDER BY sales\_count DESC;**

**A screenshot of a computer

Description automatically generated**

HTTP 5126 – Database Assignment 5.24SS

## Part 4: Challenge (1%)

1. *Based on the results from 3B above*, the query will focus on the top salespersons with the highest sales. Include the date, item (from stock\_items), price, category, and employee first and last name with the top salesperson. Use aliases for your tables. Apply the concepts that we learned in Module 5 for the query.

**SELECT s.date, si.item, si.price, si.category, CONCAT(e.first\_name, ' ', e.last\_name) AS salesperson**

**FROM sales s**

**INNER JOIN stock\_items si ON s.item = si.id**

**INNER JOIN employees e ON s.employee = e.id**

**WHERE s.employee = (**

**SELECT e.id**

**FROM employees e**

**LEFT JOIN sales s ON e.id = s.employee**

**GROUP BY e.id**

**ORDER BY COUNT(s.id) DESC**

**LIMIT 1**

**);** A screenshot of a computer

Description automatically generated

1. Create a list of *unique* items (sales), item id (stock items), items

(stock\_items), prices and categories that must include all stock\_item items in the result set, whether there were any sales or not. This list should be ordered by the stock\_items id.

**SELECT DISTINCT si.id AS item\_id, si.item, si.price, si.category**

**FROM stock\_items si**

**LEFT JOIN sales s ON si.id = s.item**

**ORDER BY si.id;**

A screenshot of a computer

Description automatically generated

A white background with black and white clouds

Description automatically generated