
Assignment 8: Group Functions (40 points)
Due Date: Monday, March 9th, 11:59 PM

Objectives:

This assignment focuses on the use of the aggregate functions. These functions combine all the results to perform a statistical operation that returns a single value. This assignment reinforces the following objectives:

- Write queries with aggregate functions: SUM, AVG, COUNT, MAX, and MIN.
- Use the GROUP BY clause to answer complex managerial questions.
- Nest aggregate functions.
- Use the GROUP BY clause with NULL values.
- Use the GROUP BY clause with the WHERE and ORDER BY clauses.
- Use the HAVING clause to filter out rows from a result table.

This assignment uses the tables associated with the *bookstore* database.

Query 1:

Write a SELECT statement that returns these columns:

- The count of the number of orders in the ORDERS table
- The sum of the tax_amount column in the ORDERS table

Query 2:

Write a SELECT statement that displays the product ID and the number of orders placed for each product. Show the results in decreasing order and label result column NumOrders.

Sample run:

```
+------------+-----------+
| product_id | NumOrders |
+------------+-----------+
|       1234 |         3 |
|       5234 |         3 |
|       2234 |         2 |
|      10234 |         2 |
```

Query 3:

Write a SELECT statement that returns one row for each category that has products with these columns:

- The category_name column from the CATEGORY table
- The count of the products in the PRODUCT table
- The list price of the most expensive product in the PRODUCT table

Sort the result set so the category with the most products appears first.

Sample run:

```
+---------------+---------------+------------------------+
| category_name | product_count | most_expensive_product |
+---------------+---------------+------------------------+
| Computer      |             6 |                2517.00 |
| DVD           |             3 |                  49.99 |
| printer       |             3 |                 599.00 |
```

Query 4:

Write a SELECT statement that returns one row for each customer that has orders with these columns:

- The email_address column from the CUSTOMER table
- The sum of the item price in the ORDERITEMS table multiplied by the quantity in the ORDERITEMS table
- The sum of the discount amount column in the ORDERITEMS table multiplied by the quantity in the ORDERITEMS table

Sort the result set in descending sequence by the item price total for each customer.

Sample run:

```
+----------------------------+------------------+-----------------------+
| email_address              | item_price_total | discount_amount_total |
+----------------------------+------------------+-----------------------+
| allan.sherwood@yahoo.com   |          6113.98 |               1883.39 |
| barryz@gmail.com           |          3250.92 |                821.14 |
| christineb@solarone.com    |          2411.95 |                722.19 |
| david.goldstein@hotmail.com|          2347.99 |                209.86 |
| frankwilson@sbcglobal.net  |          1614.97 |                309.22 |
| erinv@gmail.com            |          1299.00 |                  0.00 |
| gary_hernandez@yahoo.com   |            49.99 |                  7.49 |
+----------------------------+------------------+-----------------------+
```

Query 5:

Write a SELECT statement that returns one row for each customer that has orders with these columns:

- The email_address from the CUSTOMER table
- A count of the number of orders
- The total amount for each order (Hint: First, subtract the discount amount from the price. Then, multiply by the quantity.)

Return only those rows where the customer has more than 1 order. Sort the result set in descending order by the sum of the line item amounts.

Sample run:

email_address	order_count	order_total
allan.sherwood@yahoo.com	5	4230.59
barryz@gmail.com	6	2429.78
david.goldstein@hotmail.com	3	2138.13
christineb@solarone.com	2	1689.76
frankwilson@sbcglobal.net	4	1305.75

Query 6:

Modify the solution to query 5 so it only counts and totals line items that have an item_price value that's greater than 400.

Query 7:

Write a SELECT statement that answers this question: Which customers have ordered more than one product? Return these columns:

- The email address from the CUSTOMER table
- The count of distinct products from the customer's orders

Sample run:

email_address	number_of_products
allan.sherwood@yahoo.com	5
barryz@gmail.com	6
christineb@solarone.com	2
david.goldstein@hotmail.com	3
frankwilson@sbcglobal.net	4

Query 8:

For each vendor, retrieve the vendor ID and the number of products with a list price of $100 or higher supplied by the vendor.

Sample run:

vendor_id	COUNT(*)
2	5
3	1
5	3

Query 9:

Consider the groups of products where each group contains the products that are from the same category and supplied by the same vendor. For each such group that has more than one product, retrieve the vendor id, product category id, number of products in the group, and average price of the products in the group.

	Sample run: +-----------+-------------+--------------------+---------------+ | vendor_id | category_id | number of products | Average price | +-----------+-------------+--------------------+---------------+ | 2 | 11 | 5 | 1423.998000 | | 5 | 41 | 2 | 24.975000 | | 5 | 61 | 3 | 433.660000 | | 9 | 31 | 2 | 31.970000 | +-----------+-------------+--------------------+---------------+
Query 10:	Determine how many products are in the Printer category.

Submission

- For each of the queries above, submit the query and the result from running the query.
- You will need to label your assignment with your first initial, last name, and the name of the assignment. **Example:** hibrahim_assignment8.sql and hibrahim_assignment8.txt
- Zip the files to upload to Canvas. **Example:** hibrahim_assignment8.zip
- Submit the zipped file containing the script and output TXT via Canvas.
- Remember to include the query number as a comment at each step.
- Read your output TXT file before you submit it.