Lab 4 Summary Query

Summary queries on MGS Schema

In these exercises, you'll enter and run your own SELECT statements.

1. Write a SELECT statement that returns the columns below and uses the column headings ORDER_COUNT, TAX_TOTAL in your result.

   The count of the number of orders in the Orders_mgs table

   The sum of the tax_amount column in the Orders_mgs table.

   Output

	ORDER_COUNT	TAX_TOTAL
1	9	122.24

2. Write a SELECT statement that returns one row for each category that has products with the columns below and uses the column headings PRODUCT_COUNT, MOST_EXPENSIVE_PRODUCT for the last two columns in your result.

   The category_name column from the Categories_mgs table

   The count of the products in the Products_mgs table

   The list price of the most expensive product in the Products_mgs table

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

   Output

	CATEGORY_NAME	PRODUCT_COUNT	MOST_EXPENSIVE_PRODUCT
1	Guitars	6	2517
2	Drums	2	799.99
3	Basses	2	799.99

3. Write a SELECT statement that returns one row for each customer that has orders with the columns below and uses the column headings ITEM_PRICE_TOTAL, DISCOUNT_AMOUNT_TOTAL for the last two columns in your result.

   The email_address column from the Customers_mgs table

   The sum of the item price in the Order_Items_mgs table multiplied by the quantity in the Order_Items_mgs table

   The sum of the discount amount column in the Order_Items_mgs table multiplied by the quantity in the Order_Items_mgs table

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

   HINT: to find out which customer has orders, you need a third table.

   Output

	EMAIL_ADDRESS	ITEM_PRICE_TOTAL	DISCOUNT_AMOUNT_TOTAL
1	allan.sherwood@yahoo.com	4131	1830.39
2	christineb@solarone.com	2398	719.4
3	frankwilson@sbcglobal.net	2198.98	659.7
4	david.goldstein@hotmail.com	998	209.7
5	gary_hernandez@yahoo.com	799.99	120
6	barryz@gmail.com	489.99	186.2
7	erinv@gmail.com	299	0

4. Write a SELECT statement that returns one row for each customer that has orders with the columns below and uses the column heakdings ORDER_COUNT, ORDER_TOTAL for the last two columns in your result.

   The email_address from the Customers_mgs 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 sequence by the total order amounts.

   HINT: to find out which customer has orders and compute the total amount for each order, you need three tables.

   Output

	EMAIL_ADDRESS	ORDER_COUNT	ORDER_TOTAL
1	allan.sherwood@yahoo.com	3	2300.61
2	frankwilson@sbcglobal.net	3	1539.28
3	david.goldstein@hotmail.com	2	788.3

5. Modify the solution to Question 4 so it only counts and totals line items that have an item_price value that's greater than 400 and everything else remains the same.

   HINT: think about the differences between WHERE clause and HAVING clause.

   Output

	EMAIL_ADDRESS	ORDER_COUNT	ORDER_TOTAL
1	allan.sherwood@yahoo.com	3	2300.61
2	frankwilson@sbcglobal.net	3	1539.28

6. Write a SELECT statement that answers this question: What is the total amount ordered for each product? Return the columns below and uses the column headings PRODUCT_TOTAL for the last column in your result.

   The product name from the Products_mgs table

   The total amount for each product in the Order_Items_mgs table
   (*Hint: You can calculate the total amount by subtracting the discount amount from the item price and then multiplying it by the quantity*)

   Use the ROLLUP operator to include a row in your result that gives the grand total.

   NOTE: Attempt at this question AFTER Lec#8 that covers ROLLUP.

HINT: The ROLLUP operator should apply to one column in this query.

Output

	PRODUCT_NAME	PRODUCT_TOTAL
1	Fender Precision	559.99
2	Fender Stratocaster	978.6
3	Gibson Les Paul	2517.9
4	Gibson SG	1208.16
5	Ludwig 5-piece Drum Set with Cymbals	489.99
6	Rodriguez Caballero 11	253.15
7	Tama 5-Piece Drum Set with Cymbals	679.99
8	Washburn D10S	598
9	Yamaha FG700S	303.79
10	(null)	7589.57

7. Write a SELECT statement that answers this question: Which customers have ordered more than one product? Return the columns below and uses the column headings NUMBER_OF_PRODUCTS for the last column in your result.

   The email address from the Customers_mgs table

   The count of **distinct** products from the customer's orders

   Sort the result set in increase sequence by the email addresses.

   HINT: Three tables are needed in this query. Your need to exclude duplicate products in your product count.

   Output

	EMAIL_ADDRESS	NUMBER_OF_PRODUCTS
1	allan.sherwood@yahoo.com	3
2	david.goldstein@hotmail.com	2
3	frankwilson@sbcglobal.net	3

8. Write a SELECT statement that answers this question: What is the total quantity purchased for each product within each category? Return the columns below and use the column heading QTY_PURCHASED for the last column in your result.

   The category_name column from the Categories_mgs table

   The product_name column from the Products_mgs table

   The total quantity purchased for each product with orders in the Order_Items_mgs table

   Use the ROLLUP operator to include rows in your result that give a summary for each category name as well as a row in your result that gives the grand total.

   Use the CASE and GROUPING functions to replace null values in the category_name and product_name columns with literal values ======== if they're for summary rows.

   NOTE: Attempt at this question AFTER Lec#8 that covers ROLLUP, CASE, and GROUPING.

   HINT: The ROLLUP operator should apply to two columns in this query.

   Output

	CATEGORY_NAME	PRODUCT_NAME	QTY_PURCHASED
1	Drums	Tama 5-Piece Drum Set with Cymbals	1
2	Drums	Ludwig 5-piece Drum Set with Cymbals	1
3	Drums	==========	2
4	Basses	Fender Precision	1
5	Basses	==========	1
6	Guitars	Gibson SG	1
7	Guitars	Washburn D10S	2
8	Guitars	Yamaha FG700S	1
9	Guitars	Gibson Les Paul	3
10	Guitars	Fender Stratocaster	2
11	Guitars	Rodriguez Caballero 11	1
12	Guitars	==========	10
13	==========	==========	13

9. Write a SELECT statement that u**ses an aggregate window function** to get the total amount of each order. Return the columns below and use the column headings ITEM_AMOUNT, ORDER_AMOUNT for the last two columns in your result.

   The order_id column from the Order_Items_mgs table

   The total amount for each order item in the Order_Items_mgs table (*Hint: You can calculate the total amount by subtracting the discount amount from the item price and then multiplying it by the quantity*)

   The total amount for each order

   Sort the result set in ascending sequence by the order_id column.

   NOTE: Attempt at this question AFTER Lec#8 that covers aggregate window function.

   HINT: This query is a single-table SELECT and partitions the data by one single column using PARTITION BY clause.

   Output

	ORDER_ID	ITEM_AMOUNT	ORDER_AMOUNT
1	1	839.3	839.3
2	2	303.79	303.79
3	3	1208.16	1461.31
4	3	253.15	1461.31
5	4	1678.6	1678.6
6	5	299	299
7	6	299	299
8	7	489.3	1539.28
9	7	559.99	1539.28
10	7	489.99	1539.28
11	8	679.99	679.99
12	9	489.3	489.3