

Lab 5 Subquery and Summary Query

All questions are getting data from the MGS Schema.

1. Write a SELECT statement that returns the same result set as the SELECT statement below, but doesn't use a join. Instead, use a subquery in a WHERE clause that uses the IN keyword.

```
SELECT DISTINCT category_name
FROM categories_mgs c JOIN products_mgs p
ON c.category_id = p.category_id
ORDER BY category_name
```

Output

	CATEGORY_NAME
1	Basses
2	Drums
3	Guitars

2. Write a SELECT statement that answers this question:

Which products have a list price that's greater than the average list price for all products?

Return the product_name and list_price columns for each product.

Sort the results by the list_price column in descending sequence.

Output

	PRODUCT_NAME	LIST_PRICE
1	Gibson SG	2517
2	Gibson Les Paul	1199

3. Write a SELECT statement that returns the category_name column from the Categories_mgs table. Return one row for each category that has never been assigned to any product in the Products_mgs table. To do that, use a subquery introduced with the NOT EXISTS operator.

Output

	CATEGORY_NAME
1	Keyboards

4. Write a SELECT statement that returns three columns: email_address, order_id, and the order total for each customer. To do this, you can group the result set by the email_address and order_id columns. In addition, you must calculate the order total from the columns in the Order_Items_mgs table.

Write a second SELECT statement that uses the first SELECT statement in its FROM clause. The main query should return two columns: the customer's email address and the largest order for that customer. To do this, you can group the result set by the email_address.

NOTE – This question should be attempted after the lecture on Thursday when subquery in FROM clause will be discussed.

Output

	EMAIL_ADDRESS	MAX_ORDER_TOTAL
1	david.goldstein@hotmail.com	489.3
2	gary_hernandez@yahoo.com	679.99
3	barryz@gmail.com	303.79
4	erinv@gmail.com	299
5	allan.sherwood@yahoo.com	1461.31
6	christineb@solarone.com	1678.6
7	frankwilson@sbcglobal.net	1539.28

5. Write a SELECT statement that returns the name and discount percent of each product that has a unique discount percent. In other words, don't include products that have the same discount percent as another product.

Sort the results by the product_name column.

Output

	PRODUCT_NAME	DISCOUNT_PERCENT
1	Gibson SG	52
2	Hofner Icon	25
3	Rodriguez Caballero 11	39
4	Tama 5-Piece Drum Set with Cymbals	15
5	Washburn D10S	0
6	Yamaha FG700S	38

6. Use a correlated subquery to return one row per customer, representing the customer's oldest order (the one with the earliest date). Each row should include these three columns: email_address, order_id, and order_date.

Output

	EMAIL_ADDRESS	ORDER_ID	ORDER_DATE
1	allan.sherwood@yahoo.com	1	28-MAR-22
2	barryz@gmail.com	2	28-MAR-22
3	christineb@solarone.com	4	30-MAR-22
4	david.goldstein@hotmail.com	5	31-MAR-22
5	erinv@gmail.com	6	31-MAR-22
6	frankwilson@sbcglobal.net	7	01-APR-22
7	gary_hernandez@yahoo.com	8	02-APR-22