SQL ASSIGNMENT-1

SUBMITED BY
KALAIYARASAN DASS

Task 1: Understanding the data in hand

1. **Describe the data in hand in your own words.**

→The dataset was provided in "csv" format thus it's cleaned before importing into SQL workbench.

→In the given Database of Superstore, in all there are 5 tables.
Cust_dimen, market_fact, orders_dimen, prod_dimen and shipping_ dimen.

Cust_dimen: table provides the overall information about the customers.

Prod_dimen: table provides the information about the categories and sub-categories of products.

Shipping_dimen: table provides the shipping information about the orders.

Orders_dimen: table provides the dates of orders received with their priorities.

Market_fact: shows the detail of sales along with profit, shipping cost, discount and product base margin.

2. **Identify and list the Primary Keys and Foreign Keys for this dataset provided to you.**

→

Table Name	Primary Key	Foreign Key
Cust_dimen	**Cust_id**	**NA**
Market_fact	**NA**	**Cust_id, Ord_id, Prod_id, Ship_id**
Orders_dimen	**Ord_id**	**NA**
Prod_dimen	**Prod_id**	**NA**
Shipping_dimen	**Ship_id**	**Order_id**

Task 2: Basic and advance analysis

1. Write a query to display the Customer_Name and Customer Segment using alias name “Customer Name", "Customer Segment" from table Cust_dimen.

→ SELECT customer_name as 'Customer Name', customer_segment as 'Customer Segment' from cust_dimen;

2. Write a query to find all the details of the customer from the table cust_dimen order by desc.

→SELECT * FROM cust_dimen ORDER BY cust_id DESC;

3. Write a query to get the Order ID, Order date from table orders_dimen where ‘Order Priority’ is high.

→SELECT order_id, order_date from orders_dimen where order_priority='high';

4. Find the total and the average sales (display total_sales and avg_sales)

→SELECT sum(sales) as total_sales, avg(sales) as avg_sales from market_fact;

5. Write a query to get the maximum and minimum sales from market_fact table.

→SELECT max(sales) as 'maximum sales', min(sales) as 'minimum sales' from market_fact;

6. Display the number of customers in each region in decreasing order of no_of_customers. The result should contain columns Region, no_of_customers.

→SELECT region, COUNT(*) AS no_of_customers FROM cust_dimen GROUP BY region ORDER BY no_of_customers DESC;

7. Find the region having maximum customers (display the region name and max(no_of_customers).

→ SELECT region, COUNT(*) AS no_of_customers FROM cust_dimen GROUP BY region HAVING no_of_customers >= ALL (SELECT COUNT(*) AS no_of_customers FROM cust_dimen GROUP BY region);

8. Find all the customers from Atlantic region who have ever purchased 'TABLES' and the number of tables purchased (display the customer name, no_of_tables purchased).

```
→SELECT
   c.customer_name, COUNT(*) AS no_of_tables_purchased
        FROM
   market_fact m
        INNER JOIN
   cust_dimen c ON m.cust_id = c.cust_id
        WHERE
   c.region = 'atlantic'
        AND m.prod_id = ( SELECT
                          prod_id
                          FROM
                          prod_dimen
                          WHERE
                          product_sub_category = 'tables')
        GROUP BY m.cust_id, c.customer_name;
```

9. Find all the customers from Ontario province who own Small Business. (display the customer name, no of small business owners).

→ select customer_name, COUNT(*) as no_of_small_business_owner from cust_dimen where province='ontario' and customer_segment='small business';

10. Find the number and id of products sold in decreasing order of products sold (display product id, no_of_products sold)

```
→ SELECT
   prod_id AS product_id, COUNT(*) AS no_of_products_sold
        FROM
   market_fact
        GROUP BY prod_id
        ORDER BY no_of_products_sold DESC;
```

11. Display product Id and product sub category whose product category belongs to Furniture and Technology. The result should contain columns product id, product sub category.
→SELECT prod_id, product_sub_category from prod_dimen where product_category='furniture' OR product_category='technology';

12. Display the product categories in descending order of profits (display the product category wise profits i.e. product_category, profits)?
→SELECT
p.product_category, SUM(m.profit) AS profits
FROM
market_fact m
INNER JOIN
prod_dimen p ON m.prod_id = p.prod_id
GROUP BY p.product_category
ORDER BY profits DESC;

13. Display the product category, product sub-category and the profit within each subcategory in three columns.
→SELECT
p.product_category, p.product_sub_category, SUM(m.profit) AS profits
FROM
market_fact m
INNER JOIN
prod_dimen p ON m.prod_id = p.prod_id
GROUP BY p.product_category, p.product_sub_category;

14. Display the order date, order quantity and the sales for the order.
→select a.order_date, b.order_quantity ,b.sales from orders_dimen a INNER JOIN market_fact b ON a.ord_id=b.ord_id;

15. Display the names of the customers whose name contains the i) Second letter as ‘R’ ii) Fourth letter as ‘D’
→SELECT customer_name FROM cust_dimen WHERE customer_name LIKE '_r_d%';

16. Write a SQL query to make a list with Cust_Id, Sales, Customer Name and their region where sales are between 1000 and 5000.

→SELECT b.cust_id, b.customer_name, b.region, a.sales
FROM cust_dimen b
INNER JOIN market_fact a
WHERE b.cust_id=a.cust_id
AND a.sales BETWEEN 1000 AND 5000;

17. Write a SQL query to find the 3rd highest sales.
→ SELECT sales FROM market_fact ORDER BY sales DESC
LIMIT 2, 1;

18. Where is the least profitable product subcategory shipped the most? For the least profitable product sub-category, display the region-wise no_of_shipments and the profit made in each region in decreasing order of profits (i.e. region, no_of_shipments, profit_in_each_region)

Note: You can hardcode the name of the least profitable product subcategory.

→ SELECT
c.region, COUNT(distinct s.ship_id) AS no_of_shipments, SUM(m.profit) AS profit_in_each_region
FROM
market_fact m
INNER JOIN
cust_dimen c ON m.cust_id = c.cust_id
INNER JOIN
shipping_dimen s ON m.ship_id = s.ship_id
INNER JOIN
prod_dimen p ON m.prod_id = p.prod_id
WHERE
p.product_sub_category IN (SELECT p.product_sub_category FROM market_fact m INNER JOIN prod_dimen p ON m.prod_id = p.prod_id GROUP BY p.product_sub_category HAVING SUM(m.profit) <= ALL
(SELECT
SUM(m.profit) AS profits
FROM
market_fact m
INNER JOIN

```
        prod_dimen p ON m.prod_id = p.prod_id
        GROUP BY p.product_sub_category
)
)
GROUP BY c.region
ORDER BY profit_in_each_region DESC;
```