

بازار

SQL QUERIES

Name: Hayah Ahmed

Email Address: hayahahmeduser2@gmail.com

Phone no.: +92-3363717227

```
SELECT * FROM SALES_DATA
```

--TASK DATA EXPLORATION--

--Total number of orders

```
SELECT COUNT(*) AS total_orders  
FROM SALES_DATA_FINAL;
```

--Total sales revenue

```
SELECT Round(SUM(amount_per_unit * ordered_quantity -  
item_discount), 3) AS total_sales_revenue  
FROM SALES_DATA_FINAL  
WHERE order_status = 'CLOSED';
```

--Average Order Quantity:

```
SELECT ROUND(AVG(ordered_quantity), 2) AS average_order_quantity  
FROM SALES_DATA_FINAL;
```

--Distribution of orders by warehouse and store

```
SELECT order_warehouse_id, store_id, COUNT(*) AS order_count  
FROM SALES_DATA_FINAL  
GROUP BY order_warehouse_id, store_id  
ORDER BY order_count DESC;
```

--Top Selling items

```
SELECT DISTINCT item_id, COUNT(ordered_quantity) AS order_count
FROM SALES_DATA_FINAL
GROUP BY item_id
ORDER BY order_count DESC;
```

-- TASK Analytical Questions--

--What is the overall discount rate (average discount per item sold)?

```
SELECT ROUND(AVG(item_discount / (amount_per_unit *
ordered_quantity)),2) AS avg_discount_rate
FROM SALES_DATA_FINAL
WHERE amount_per_unit > 0 AND ordered_quantity > 0 AND
item_discount > 0;
```

--Which warehouse has the highest average order value (total sales revenue divided by number of orders)?

```
SELECT order_warehouse_id, ROUND(AVG(amount_per_unit *
ordered_quantity),2) AS avg_order_value
FROM SALES_DATA_FINAL
GROUP BY order_warehouse_id
HAVING order_warehouse_id is not NULL
ORDER BY avg_order_value DESC;
```

--What is the total revenue generated by each store?

```
SELECT store_id,
```

```
SUM(amount_per_unit * ordered_quantity - item_discount) AS
total_revenue
FROM SALES_DATA_FINAL
WHERE order_status = 'CLOSED'
GROUP BY store_id
ORDER BY total_revenue DESC;
```

-- Identify the top 5 customers (based on total amount spent).

```
SELECT order_number AS top_customers,
       SUM(amount_per_unit * ordered_quantity) AS total_order_value
FROM SALES_DATA_FINAL
GROUP BY order_number
ORDER BY total_order_value DESC;
```

-- Calculate the month-over-month growth rate of sales revenue.

```
SELECT
  TO_CHAR(TO_DATE(order_date, 'DD/MM/YYYY'), 'IW') AS week,
  SUM(amount_per_unit * ordered_quantity - item_discount ) AS
weekly_revenue,
  ROUND(
    (
      SUM(amount_per_unit * ordered_quantity * (1 - item_discount / 100)) -
      LAG(SUM(amount_per_unit * ordered_quantity * (1 - item_discount /
100))) OVER (ORDER BY TO_DATE(order_date, 'DD/MM/YYYY'))
    ) /
```

```
LAG(SUM(amount_per_unit * ordered_quantity * (1 - item_discount /
100))) OVER (ORDER BY TO_DATE(order_date, 'DD/MM/YYYY')) *
100,
2) AS growth_rate
FROM
SALES_DATA_FINAL
WHERE -- Add condition for closed orders
order_status = 'CLOSED' AND
TO_DATE(order_date, 'DD/MM/YYYY') >= TO_DATE('01/02/2023',
'DD/MM/YYYY') AND TO_DATE(order_date, 'DD/MM/YYYY') <=
TO_DATE('28/02/2023', 'DD/MM/YYYY')
GROUP BY
TO_CHAR(TO_DATE(order_date, 'DD/MM/YYYY'), 'IW')
ORDER BY
TO_CHAR(TO_DATE(order_date, 'DD/MM/YYYY'), 'IW');
```

--6. Determine the percentage of orders that were canceled.

```
SELECT
COUNT(CASE WHEN order_status = 'CANCELLED' THEN 1 ELSE
NULL END) AS canceled_orders,
COUNT(*) AS total_orders,
ROUND((COUNT(CASE WHEN order_status = 'CANCELLED' THEN 1
ELSE NULL END) / COUNT(*)) * 100, 2) AS cancellation_rate_percent
FROM SALES_DATA;
```

--Additional findings--

```
SELECT
    SUM(CASE WHEN order_status = 'CLOSED' THEN amount_per_unit *
ordered_quantity - item_discount ELSE 0 END) AS total_sales,
    SUM(CASE WHEN order_status = 'CANCELLED' THEN
amount_per_unit * ordered_quantity - item_discount ELSE 0 END) AS
cancelled_sales,
    CASE WHEN SUM(amount_per_unit * ordered_quantity) > 0 THEN
        SUM(CASE WHEN order_status = 'CANCELLED' THEN
amount_per_unit * ordered_quantity - item_discount ELSE 0 END) /
        SUM(amount_per_unit * ordered_quantity) * 100
    ELSE 0 END AS sales_decrease_percent
FROM SALES_DATA_FINAL;
```

--

```
SELECT item_id, COUNT(DISTINCT order_number) AS purchase_count
FROM SALES_DATA_FINAL
WHERE order_status = 'CLOSED'
GROUP BY item_id
ORDER BY purchase_count ASC;
```