📈Customer & Sales Performance Analysis Questions

1. **Which products generate the highest and lowest total sales revenue?**
   → Group by product or category, order by total sales.

```
Select * from order_items;
Select * from orders;
Select * from products;

SELECT
   OI.product_id,
   P.product_name,
   SUM(OI.quantity) AS total_quantity_sold,
   SUM(OI.quantity * OI.list_price) AS total_revenue
FROM order_items OI
JOIN orders O ON OI.order_id = O.order_id
JOIN products P ON OI.product_id = P.product_id
GROUP BY OI.product_id, P.product_name
ORDER BY total_revenue DESC; --use when show highest revenue
ORDER BY total_revenue ASC;  --use when show lowest revenue
```

2. **What are the top 5 selling product categories in terms of quantity and revenue?**
   → Use aggregation + ranking (window functions or `LIMIT`).

```
Select * from orders;
Select * from order_items;
Select * from products;
Select * From categories;

Select C.category_id, C.category_name,
          SUM(OI.quantity) As Total_quantity_sold,
          Sum(OI.quantity * OI.list_price) As Total_revenue
From order_items OI
Join orders O on O.order_id = OI.order_id
Join products P on OI.product_id = P.product_id
Join categories C on P.category_id = C.category_id
Group by C.category_id, C.category_name
Order by Total_revenue Desc
```

```
Limit 5;
```

3. **Which customers are the top buyers in terms of total purchase value?**
   → Join customers and sales data, group by customer, use `SUM` and `RANK`.

```
Select * from customers;
Select * from order_items;
Select * from orders;

Select C.customer_id, C.first_name,
        Sum(OI.list_price*OI.quantity) as Total_purchase
From order_items OI
Join orders O  on O.order_id = OI.order_id
Join customers C on O.customer_id = C.customer_id
Group by C.customer_id,C.first_name
Order by Total_purchase Desc;
```

4. **What are the monthly sales trends over the past year?**
   → Use date functions and grouping by `MONTH`.

```
Select * from order_items;
Select * from orders;

Select
        DATE_TRUNC('month',O.order_date) as month,
        Sum(OI.list_price * OI.quantity) as Total_Revenue
From order_items OI
Join Orders O on O.order_id = OI.order_id
group by month
order by month;
```

5. **Which stores are underperforming based on total revenue or average order value?**
   → Group by store, calculate total and average order values.

```
Select * from order_items;
Select * from orders;
```

```
select * from stores;

Select S.store_id, S.store_name,
        Sum(OI.list_price * OI.quantity) as Total_Revenue,
        Avg(OI.list_price * OI.quantity) as Avg_Order_Value
From order_items OI
Join Orders O on O.order_id = OI.order_id
Join stores S on S.store_id = O.store_id
group by S.store_id, S.store_name
HAVING
  SUM(OI.list_price * OI.quantity) < 100000 -- threshold for total revenue (can be adjusted)
  OR AVG(OI.list_price * OI.quantity) < 1800  -- threshold for average order value (can be adjusted)
Order by Total_Revenue Desc;
```

6. **How do different customer demographics (e.g., state, city, gender) affect purchase behavior?**
 → Group by customer location or gender, analyze average and total sales.

```
Select * from customers;
Select * from order_items;
Select * from orders;

Select C.State, C.city,
        Sum(OI.list_price * OI.quantity) as Total_Revenue,
        Avg(OI.list_price * OI.quantity) as Avg_Order_Value
From order_items OI
Join Orders O on O.order_id = OI.order_id
Join customers C on C.customer_id = O.customer_id
group by C.State, C.city
order by Total_Revenue Desc;
```

📈 Operational Insights & Optimization Questions

7. **Which staff members are generating the most sales?**
→ Join staff and sales tables, group and rank.

```
Select * from order_items;
Select * from orders;
select * from staffs;

Select S.store_id, S.staff_id, S.first_name,S.last_name,
   COUNT(DISTINCT O.order_id) AS Orders_Handled,
   SUM(OI.list_price * OI.quantity) AS Total_Revenue_Staff,
   RANK() OVER (ORDER BY SUM(OI.list_price * OI.quantity) DESC) AS Sales_Rank
from order_items OI
Join Orders O on O.order_id = OI.order_id
Join Staffs S on O.staff_id = S.staff_id
group by S.store_id, S.staff_id, S.first_name,S.last_name
order by Total_Revenue_Staff Desc;
```

8. **What is the average order value by store and by product category?**
→ Join tables, group by store and category.

```
Select * From categories;
Select * from order_items;
Select * from orders;
Select * from products;
select * from stores;


Select S.store_id,S.store_name,C.category_name,
          Avg(OI.quantity * OI.list_price) as Avg_order_value
from order_items OI
join orders O on O.order_id = OI.order_id
join products P on OI.product_id = P.product_id
join categories C on C.category_id = P.category_id
Join stores S on S.store_id = O.store_id
group by S.store_id,S.store_name,C.category_name
order by Avg_order_value Desc;
```

9. **Are there any seasonal trends in product sales (e.g., certain categories doing better in certain months)?**
   → Use date filtering, group by month and category.

```
Select * From categories;
Select * from order_items;
Select * from orders;
Select * from products;


SELECT
        to_Char(date_trunc('month',O.order_date), 'Mon YYYY') as Month_year,
        C.category_id, C.category_name,
        SUM(OI.list_price * OI.quantity) AS Total_monthly_revenue
from order_items OI
Join orders O on O.order_id = OI.order_id
Join products P on OI.product_id = P.product_id
Join categories C on P.category_id = C.category_id
Group by Month_year, C.category_id, C.category_name
Order by Total_monthly_revenue Desc;
```

10. **What is the repeat customer rate, and who are the most loyal customers?**
    → Count distinct invoices per customer, identify those with multiple purchases.

```
Select * from customers;
Select * from order_items;
Select * from orders;

Select
        C.Customer_id, C.first_name, C.last_name,
        COUNT(DISTINCT O.order_id) AS total_orders,
        sum (OI.quantity*OI.list_price) as total_purchase
from order_items OI
join orders O on O.order_id = OI.order_id
Join customers C on C.customer_id = O.customer_id
group by C.Customer_id, C.first_name, C.last_name
HAVING COUNT(DISTINCT O.order_id) > 1
order by total_purchase Desc;

SELECT
```

```
    ROUND(
        COUNT(DISTINCT CASE WHEN order_count > 1 THEN customer_id END)::DECIMAL
        / COUNT(DISTINCT customer_id), 2
    ) AS repeat_customer_rate
FROM

(SELECT C.customer_id,
    COUNT(DISTINCT O.order_id) AS order_count
    FROM customers C
    JOIN orders O ON O.customer_id = C.customer_id
    GROUP BY C.customer_id
) AS customer_orders;
```