Basics

Q1- Retrieve the total number of orders placed.

```
Ans - select count(order_id) as Total_orders from orders
```

Q2- Calculate the total revenue generated from pizza sales.

```
Ans - SELECT
    round(sum(order_details.quantity * pizzas.price),
            2) as Total_revenue
FROM
    order_details
        JOIN
 pizzas ON pizzas.pizza_id = order_details.pizza_id;
```

Q3- Calculate the total revenue generated from pizza sales.

```
 Ans - SELECT
    round(sum(order_details.quantity * pizzas.price),
            2) as Total_revenue
FROM
    order_details
```

```
       JOIN
 pizzas ON pizzas.pizza_id = order_details.pizza_id;
```

Q4 - Identify the highest-priced pizza.

```
Ans - SELECT
    pizza_types.name, pizzas.price
FROM
    pizza_types
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
    order by pizzas.price Desc
    limit 1;
```

Q5- Identify the most common pizza size ordered.

```
Ans - SELECT
    pizzas.size,
    COUNT(order_details.order_details_id) AS order_count
FROM
    pizzas
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizzas.size
ORDER BY order_count DESC;
```

Intermediate

Q6- List the top 5 most ordered pizza types along with their quantities.

Ans -

```
select pizza_types.name, sum(order_details.quantity) as quantity
from pizza_types
JOIN pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
JOIN order_details ON order_details.pizza_id = pizzas.pizza_id
group by pizza_types.name
order by quantity desc
limit 5;
```

Q7- Join the necessary tables to find the total quantity of each pizza category ordered

Ans -

```
SELECT
    pizza_types.category,
    SUM(order_details.quantity) AS total_quantity
FROM
    pizza_types
        JOIN
```

```
  pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
    JOIN
  order_details ON order_details.pizza_id = pizzas.pizza_id
  group by pizza_types.category
  order by total_quantity desc
```

Q8 - Determine the distribution of orders by hour of the day.

```
Ans - select extract(HOUR from order_time) as order_hour,
count(orders.order_id) as order_count
from orders
group by extract(HOUR from order_time)
order by order_count desc
```

Q9 - Join relevant tables to find the category-wise distribution of pizzas.

```
Ans - select pizza_types.category, count(name) from pizza_types
group by category
```

Q10 - Group the orders by date and calculate the average number of pizzas ordered per

```
Ans - SELECT
   ROUND(AVG(quantity),0) as avg_pizza_order_perday
FROM
   (SELECT
      orders.order_date, SUM(order_details.quantity) AS quantity
   FROM
      orders
   JOIN order_details ON orders.order_id = order_details.order_id
   GROUP BY orders.order_date) AS order_quantity;
```

Q11 - Determine the top 3 most ordered pizza types based on revenue.

Ans -

```
SELECT
    pizza_types.name,
    ROUND(SUM(order_details.quantity * pizzas.price),
            0) AS total_revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY total_revenue DESC
LIMIT 3;
```

Q12 - Calculate the percentage contribution of each pizza type to total revenue.

```
Ans - SELECT
    pizza_types.category,
    ROUND(SUM(order_details.quantity * pizzas.price) / (SELECT
                    ROUND(SUM(order_details.quantity * pizzas.price),
                                2) AS total_sales
                FROM
                    order_details
                        JOIN
                    pizzas ON pizzas.pizza_id = order_details.pizza_id),
            2) * 100 AS revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY revenue DESC
```

Q13 - Analyze the cumulative revenue generated over time.

```
Ans - select order_date,
sum(revenue) over(order by order_date) as cum_revenue from
(select orders.order_date ,
sum(order_details.quantity * pizzas.price) as revenue

from order_details join pizzas

ON order_details.pizza_id = pizzas.pizza_id

JOIN orders ON orders.order_id = order_details.order_id

group by orders.order_date) as sales;
```

Q14 - Determine the top 3 most ordered pizza types based on revenue for each pizza category.

```
Ans - select name, revenue from

(select category, name, revenue, rank() over(partition by category order by revenue desc) as rn

from

(select pizza_types.category, pizza_types.name,

sum(order_details.quantity * pizzas.price) as revenue

From pizza_types JOIN pizzas

ON pizza_types.pizza_type_id = pizzas.pizza_type_id

JOIN order_details On order_details.pizza_id = pizzas.pizza_id

group by pizza_types.category, pizza_types.name) as a) as b

where rn <=3 ;
```

Intermediate

--.

--

--

--

--

--

--