

```
-- 1. Total Number of Orders
SELECT COUNT(*) AS Total_Orders FROM ORDERS;

-- 2. Calculate total revenue generated from pizza sales
SELECT
    ROUND(SUM(OD.quantity * P.price),2) AS Total_Sales
FROM
    order_details OD
JOIN
    pizzas P ON OD.pizza_id = P.pizza_id;

-- 3. Identify the highest-priced pizza
SELECT TOP 1
    PT.name,
    P.price
FROM
    pizza_types PT
JOIN
    pizzas P ON PT.pizza_type_id = P.pizza_type_id
ORDER BY
    P.price DESC;

-- 4. Identify the most common pizza size ordered
SELECT TOP 1
    size,
    COUNT(*) AS order_count
FROM
    pizzas P
JOIN
    order_details OD ON P.pizza_id = OD.pizza_id
GROUP BY
    size
ORDER BY
    order_count DESC;

-- 5. List the top 5 most ordered pizza types along with their quantities
SELECT TOP 5
    PT.name,
    SUM(OD.quantity) AS Quantity
FROM
    pizza_types PT
JOIN
    pizzas ON PT.pizza_type_id = pizzas.pizza_type_id
JOIN
    order_details OD ON OD.pizza_id = pizzas.pizza_id
GROUP BY
    PT.name
ORDER BY
    Quantity DESC;
```

-- 6. Find the total quantity of each pizza category ordered

```
SELECT
    PT.category,
    SUM(OD.quantity) AS Quantity
FROM
    pizza_types PT
JOIN
    pizzas P ON PT.pizza_type_id = P.pizza_type_id
JOIN
    order_details OD ON OD.pizza_id = P.pizza_id
GROUP BY
    PT.category
ORDER BY
    Quantity DESC;
```

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

```
SELECT
    DATEPART(HOUR, time) AS Hour,
    COUNT(ORDER_ID) AS Order_count
FROM
    ORDERS
GROUP BY
    DATEPART(HOUR, time) ORDER BY Hour;
```

-- 8. Find the category-wise distribution of pizzas

```
SELECT category, COUNT(name) FROM pizza_types GROUP BY category;
```

--9. Average number of pizzas ordered per day

```
SELECT AVG(quantity) AS Average_Quantity
FROM (
    SELECT
        orders.date,
        SUM(OD.quantity) AS quantity
    FROM
        orders
    JOIN
        order_details OD ON orders.order_id = OD.order_id
    GROUP BY
        orders.date
) AS Order_Quantity;
```

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

```
SELECT TOP 3
    PT.name,
    SUM(OD.quantity * pizzas.price) AS Revenue
FROM
    pizza_types PT
```

```
JOIN
    pizzas ON pizzas.pizza_type_id = PT.pizza_type_id
JOIN
    order_details OD ON OD.pizza_id = pizzas.pizza_id
GROUP BY
    PT.name
ORDER BY
    Revenue DESC;

-- 11. Calculate the percentage contribution of each pizza type to total revenue
WITH TotalSales AS (
    SELECT SUM(OD.quantity * pizzas.price) AS Total_Sales
    FROM order_details OD
    JOIN pizzas ON pizzas.pizza_id = OD.pizza_id
)
SELECT
    PT.category,
    ROUND(SUM(OD.quantity * pizzas.price) / TotalSales.Total_Sales * 100, 2) AS Revenue_Percentage
FROM
    pizza_types PT
JOIN
    pizzas ON PT.pizza_type_id = pizzas.pizza_type_id
JOIN
    order_details OD ON OD.pizza_id = pizzas.pizza_id
CROSS JOIN
    TotalSales
GROUP BY
    PT.category, TotalSales.Total_Sales
ORDER BY
    Revenue_Percentage DESC;

-- 12. Analyze the cumulative revenue generated over time
SELECT
    date,
    SUM(Revenue) OVER(ORDER BY date) AS cum_revenue
FROM (
    SELECT
        orders.date,
        SUM(OD.quantity * pizzas.price) AS Revenue
    FROM
        order_details OD
    JOIN
        pizzas ON OD.pizza_id = pizzas.pizza_id
    JOIN
        orders ON orders.order_id = OD.order_id
    GROUP BY
        orders.date
) AS Sales;
```

-- 13. Top 3 most ordered pizza types based on revenue for each pizza category

```
SELECT
    Name,
    Revenue
FROM
(
    SELECT
        category,
        Name,
        Revenue,
        RANK() OVER(PARTITION BY category ORDER BY Revenue DESC) AS Ranks
    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
    Ranks <= 3;
```