Q. Retrieve the total number of orders placed.

```
SELECT
   COUNT(order_id)
FROM
   orders;
```

Q. Calculate the total revenue generated from pizza sales.

```
SELECT
   ROUND(SUM(orders_details.quantity * pizzas.price))
FROM
   orders_details
      JOIN
   pizzas ON orders_details.pizza_id = pizzas.pizza_id;
```

Q. Identify the highest priced pizza.

```
SELECT
   pizza_types.name, pizzas.price
FROM
   pizza_types
      JOIN
   pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
ORDER BY pizzas.price DESC
LIMIT 1;
```

Q. Identify the most common pizza size ordered.

```
SELECT
   pizzas.size,
   COUNT(orders_details.order_detail_id) AS orders_count
FROM
   pizzas
      JOIN
```

```
    orders_details ON pizzas.pizza_id = orders_details.pizza_id
GROUP BY pizzas.size
ORDER BY orders_count DESC
LIMIT 3;
```

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

```
SELECT
    pizza_types.name, COUNT(orders_details.quantity) AS quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    orders_details ON pizzas.pizza_id = orders_details.pizza_id
GROUP BY pizza_types.name
ORDER BY quantity DESC
LIMIT 5;
```

Q. Join the necessary tables to find the total quantity of each pizza category ordered.

```
SELECT
    pizza_types.category,
    COUNT(orders_details.quantity) AS Quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    orders_details ON pizzas.pizza_id = orders_details.pizza_id
GROUP BY pizza_types.category
ORDER BY quantity DESC;
```

Q. Determine the distribution of orders by hour of the day.

```
SELECT
    HOUR(orders.order_time),
    COUNT(orders.order_id) AS order_count
FROM
    orders
GROUP BY HOUR(orders.order_time)
ORDER BY order_count DESC;
```

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

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

Q. Group the orders by date and calculate the average number of pizzas ordered per day.

```
SELECT
    AVG(quantity)
FROM
    (SELECT
        orders.order_date AS ord_date,
            SUM(orders_details.quantity) AS quantity
    FROM
        orders
    JOIN orders_details ON orders.order_id = orders_details.order_id
    GROUP BY ord_date) AS order_quantity;
```

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

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

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

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

Q. Analyze the cumulative revenue generated over time.

```
SELECT
   order_date,
   SUM(revenue) OVER (ORDER BY order_date) AS cumulative_revenue
FROM (
   SELECT
      orders.order_date,
      SUM(orders_details.quantity * pizzas.price) AS revenue
   FROM
      orders_details
   JOIN
      pizzas ON orders_details.pizza_id = pizzas.pizza_id
   JOIN
      orders ON orders.order_id = orders_details.order_id
   GROUP BY
      orders.order_date
) AS sales
ORDER BY  order_date;
```

Q. Determine the 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 rn
   FROM (
      SELECT
         pizza_types.category,
         pizza_types.name,
```

```
        SUM(orders_details.quantity * pizzas.price) AS revenue
    FROM
        pizza_types
    JOIN
        pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
    JOIN
        orders_details ON orders_details.pizza_id = pizzas.pizza_id
    GROUP BY
        pizza_types.category,
        pizza_types.name
  ) AS a
) AS b
WHERE rn <= 3;
```