

1.Retrieve the total number of orders placed.

```
USE PIZZADB;  
SELECT COUNT(ORDER_ID) AS TOTAL_ORDERS FROM ORDERS;
```

2.Calculate the total revenue generated from pizza sales.

```
USE PIZZADB;  
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
```

3.Identify the highest-priced pizza.

```
USE PIZZADB;  
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
```

4.Identify the most common pizza size ordered.

```
USE PIZZADB;  
SELECT PIZZAS.SIZE , COUNT(ORDER_DETAILS.ORDER_DETAILS_ID) AS TOTAL_SALES  
FROM PIZZAS JOIN ORDER_DETAILS  
ON PIZZAS.PIZZA_ID = ORDER_DETAILS.PIZZA_ID  
GROUP BY PIZZAS.SIZE ORDER BY TOTAL_SALES DESC LIMIT 1;
```

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

```
SELECT PIZZA_TYPES.NAME, 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.NAME ORDER BY TOTAL_QUANTITY DESC LIMIT 5
```

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

```
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
```

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

```
SELECT HOUR(ORDER_TIME), COUNT(ORDER_ID) FROM ORDERS
GROUP BY HOUR(ORDER_TIME)
```

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

```
SELECT COUNT(NAME) , CATEGORY FROM PIZZA_TYPES
GROUP BY CATEGORY
```

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

```
SELECT ROUND(AVG(TOTAL_QUANTITY),0) AS AVG_PIZZAS_PER_DAY
FROM (SELECT ORDERS.ORDER_TIME,SUM(ORDER_DETAILS.QUANTITY) AS TOTAL_QUANTITY
FROM ORDERS JOIN ORDER_DETAILS
ON ORDERS.ORDER_ID = ORDER_DETAILS.ORDER_ID
GROUP BY ORDERS.ORDER_TIME) AS QUANTITY_PER_DAY;
```

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

```
SELECT 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.NAME ORDER BY REVENUE DESC LIMIT 3
```

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

```
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)*100,2) AS REVENUE_PERCENTAGE
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
```

12. Analyze the cumulative revenue generated over time.

```
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 ORDERS JOIN ORDER_DETAILS
ON ORDERS.ORDER_ID = ORDER_DETAILS.ORDER_ID
JOIN PIZZAS
ON PIZZAS.PIZZA_ID = ORDER_DETAILS.PIZZA_ID
GROUP BY ORDERS.ORDER_DATE) AS SALES;
```

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

```
SELECT CATEGORY, 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;
```