Pizza Sales Analysis - SQL Questions & Solutions

1) Retrieve the total number of orders placed.

```
SELECT COUNT(*) AS order_placed FROM orders;
```

2) Calculate the total revenue generated from pizza sales.

```
SELECT SUM(p.price * od.quantity) AS total_revenue
FROM pizzas AS p
JOIN order_details AS od ON p.pizza_id = od.pizza_id;
```

3) Identify the highest-priced pizza.

```
SELECT pizza_id, price AS Highest_priced_pizza
FROM pizzas
ORDER BY price DESC
LIMIT 1;
```

4) Identify the most common pizza size ordered.

```
SELECT size, COUNT(*) AS Total_Pizza
FROM pizzas
GROUP BY size
ORDER BY Total_Pizza DESC;
```

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

```
SELECT pt.name, SUM(od.quantity) AS Total_Quantity
FROM pizzas AS p
JOIN order_details AS od ON p.pizza_id = od.pizza_id
JOIN pizza_types AS pt ON pt.pizza_type_id = p.pizza_type_id
GROUP BY pt.name
ORDER BY Total_Quantity DESC
LIMIT 5;
```

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

```
SELECT pt.category, SUM(od.quantity) AS total_ordered
FROM pizzas AS p
JOIN order_details AS od ON p.pizza_id = od.pizza_id
JOIN pizza_types AS pt ON p.pizza_type_id = pt.pizza_type_id
GROUP BY pt.category
```

```
ORDER BY total_ordered DESC;
```

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

```
SELECT HOUR(order_time) AS order_hour, COUNT(*) AS total_orders
FROM orders
GROUP BY order_hour
ORDER BY order_hour;
```

8) Find the category-wise distribution of pizzas.

```
SELECT category, COUNT(*) AS pizza_distribution
FROM pizza_types
GROUP BY category
ORDER BY pizza_distribution;
```

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

```
SELECT o.order_date, AVG(od.quantity) AS avg_pizzas_per_day
FROM orders o
JOIN order_details od ON o.order_id = od.order_id
GROUP BY o.order_date
ORDER BY o.order_date;
```

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

```
SELECT pt.name AS Pizza_type, SUM(p.price * od.quantity) AS Total_Revenue
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.name
ORDER BY Total_Revenue DESC
LIMIT 3;
```

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

```
SELECT pt.category AS Pizza_Category, SUM(p.price * od.quantity) AS Total_Revenue,
(SUM(p.price * od.quantity) * 100.0) /
(SELECT SUM(p.price * od.quantity) FROM pizzas p JOIN order_details od ON od.pizza_id =
p.pizza_id) AS Revenue_Percentage
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 Total_Revenue DESC;
```

12) Analyze the cumulative revenue generated over time.

```
SELECT o.order_date, SUM(p.price * od.quantity) AS Total_Revenue,
SUM(SUM(p.price * od.quantity)) OVER (ORDER BY o.order_date) AS Cumulative_Revenue
FROM orders o
JOIN order_details od ON o.order_id = od.order_id
JOIN pizzas p ON od.pizza_id = p.pizza_id
GROUP BY o.order_date
ORDER BY o.order_date;
```

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

```
WITH PizzaRevenue AS (
        SELECT pt.category, pt.name AS Pizza_Type, SUM(p.price * od.quantity) AS
Total_Revenue,
     RANK() OVER (PARTITION BY pt.category ORDER BY SUM(p.price * od.quantity) DESC) AS
rnk
    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, pt.name
)
SELECT category, Pizza_Type, Total_Revenue, rnk
FROM PizzaRevenue
WHERE rnk <= 3
ORDER BY category, rnk;
```