SQL Queries for PizzaHut Data Analysis

Mohd Hasan Khan

Basic Questions

Retrieve the total number of orders placed

```sql
SELECT
    COUNT(order_id) AS Total_Orders
FROM
    orders;
```

Calculate the total revenue generated from pizza sales

```sql
SELECT
    ROUND(SUM(a.price * b.quantity), 2) AS Total_Revenue
FROM
    pizzas AS a
    JOIN order_details AS b ON a.pizza_id = b.pizza_id;
```

Identify the highest-priced pizza

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

Identify the most common pizza size ordered

```sql
SELECT
    a.size, COUNT(b.order_details_id) AS order_count
FROM
```

```
    pizzas AS a
    JOIN order_details AS b ON a.pizza_id = b.pizza_id
GROUP BY a.size
ORDER BY order_count DESC;
```

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

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

Medium Questions

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

```
SELECT
    c.category AS category, SUM(b.quantity) AS total_quantity
FROM
    pizzas AS a
    JOIN order_details AS b ON a.pizza_id = b.pizza_id
    JOIN pizza_types AS c ON a.pizza_type_id = c.pizza_type_id
GROUP BY category
ORDER BY total_quantity DESC;
```

Determine the distribution of orders by hour of the day

```
SELECT
    HOUR(order_time) AS Hour, COUNT(order_id) AS total_count
FROM
    orders
GROUP BY Hour;
```

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

```
SELECT
    category, COUNT(name) AS Total_count
FROM
    pizza_types
GROUP BY category
ORDER BY Total_count DESC;
```

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

```
SELECT
    ROUND(AVG(Total_quantity), 2) AS Average_Number_Of_Pizzas_Order_PerDay
FROM
    (SELECT
        b.order_date AS Date, SUM(a.quantity) AS Total_quantity
    FROM
        order_details AS a
    JOIN orders AS b ON a.order_id = b.order_id
    GROUP BY Date) AS Order_Quantity;
```

Determine the top 3 most ordered pizza types based on revenue

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

Advanced Questions

Calculate the percentage contribution of each pizza type to total revenue

```
SELECT
    c.category AS Category,
    ROUND((SUM(a.price * b.quantity)) /
    (SELECT
        ROUND(SUM(a.price * b.quantity), 2) AS Total_Revenue
    FROM
        pizzas AS a
    JOIN order_details AS b ON a.pizza_id = b.pizza_id) * 100, 2) AS Revenue
FROM
    pizzas AS a
    JOIN order_details AS b ON a.pizza_id = b.pizza_id
    JOIN pizza_types AS c ON a.pizza_type_id = c.pizza_type_id
GROUP BY Category
ORDER BY Revenue DESC;
```

Analyze the cumulative revenue generated over time

```
SELECT Date, SUM(Total) OVER (ORDER BY Date) AS cumulative_revenue
FROM
    (SELECT date(a.order_date) AS Date, ROUND(SUM(b.quantity * c.price), 2) AS Total
     FROM orders AS a
     JOIN order_details AS b ON a.order_id = b.order_id
     JOIN pizzas AS c ON b.pizza_id = c.pizza_id
     GROUP BY Date) AS sales;
```

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 rank_
     FROM
        (SELECT b.category, b.name, SUM(a.price * c.quantity) AS revenue
         FROM pizzas AS a
         JOIN pizza_types AS b ON a.pizza_type_id = b.pizza_type_id
         JOIN order_details AS c ON a.pizza_id = c.pizza_id
         GROUP BY b.name, b.category) AS x) AS y
WHERE rank_ <= 3;
```