Sales Analysis

This presentation will analyze sales data from a pizza restaurant, providing insights into key metrics and trends. We will explore various aspects of the business, including order volume, revenue, pizza popularity, and customer behavior.

Total Number of Orders Placed

SQL Query: -

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

Total Revenue

SQL Query: -

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

Highest Priced Pizza

SQL Query: -

```
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;
```

Most Common Pizza Size Ordered

SQL Query: -

```
SELECT
    pizzas.size,
    COUNT(order_details.order_details_id) AS Most_Common_Pizza_Size
FROM
    pizzas
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizzas.size
ORDER BY Most_Common_Pizza_Size DESC
LIMIT 1;
```

Top 5 Most Ordered Pizza Types

SQL Query: -

```sql
SELECT
    pizza_types.name,
    SUM(order_details.quantity) AS Quantity_Ordered
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 Quantity_Ordered DESC
LIMIT 5;
```

Total Quantity of each Pizza Category Ordered

SQL Query: -

```
SELECT
    pizza_types.category,
    SUM(order_details.quantity) AS Quantity_Ordered
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 Quantity_Ordered DESC;
```

Distribution of Orders by Hour of the Day

SQL Query: -

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

Category-Wise Distribution of Pizzas

SQL Query: -

```
SELECT
    pizza_types.category AS Category,
    COUNT(pizza_types.name) AS No_of_Pizzas
FROM
    pizza_types
GROUP BY Category;
```

Average Number of Pizzas Ordered Per Day

SQL Query: -

```
SELECT
    ROUND(AVG(Quantity), 0) AS Avg_Order_Per_Day
FROM
    (SELECT
        orders.order_date, SUM(order_details.quantity) AS Quantity
    FROM
        orders
    JOIN order_details ON orders.order_id = order_details.order_id
    GROUP BY orders.order_date) AS Order_Quantity;
```

Top 3 Most Ordered Pizza Types Based on Revenue

SQL Query: -

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

Percentage Contribution of each Pizza Type to Total Revenue

SQL Query: -

```
SELECT
  pizza_types.category,
  CONCAT(ROUND((SUM(order_details.quantity * pizzas.price) / (SELECT
              ROUND(SUM(order_details.quantity * pizzas.price),
                    2) AS Total_Revenue
            FROM
              order_details
                JOIN
              pizzas ON pizzas.pizza_id = order_details.pizza_id)) * 100,
          2),
      ' %') AS Percentage_Contribution_in_Revenue
FROM
  pizza_types
    JOIN
  pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
    JOIN
  order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category;
```

Cumulative Revenue Generated Over Time

SQL Query: -

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

Top 3 Most Ordered Pizza Types Based on Revenue for each Pizza Category

SQL Query: -

```
SELECT
  category,
  name,
  Total_Revenue
FROM
  (SELECT
    category,
    name,
    Total_Revenue,
    RANK() OVER(PARTITION BY category ORDER BY Total_Revenue DESC) AS rn
  FROM
    (SELECT
      pizza_types.category,
      pizza_types.name,
      SUM(order_details.quantity * pizzas.price) AS Total_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;
```