



SQL_PROJECT

ON

PIZZA_SALES

BY --RAJ SHARMA..





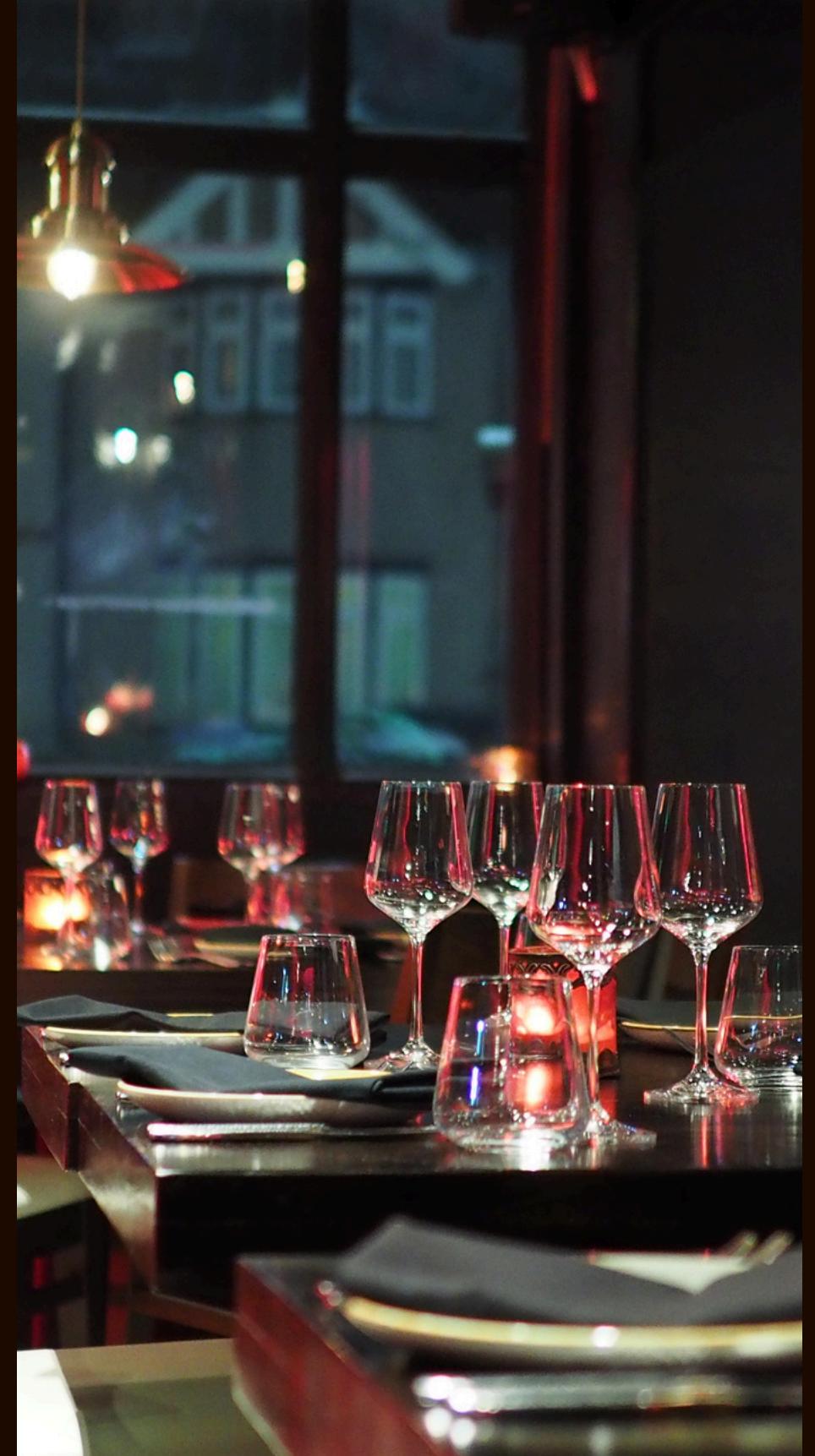
BORCELLE
R E S T A U R A N T

ABOUT US

Hello ! , my name is Raj Sharma, and in this project i have utilize SQL queries to solve questions that were related to pizza sa

HERE ARE SOME BASIC SALES ANALYSIS..

1. Retrieve the total number of orders placed.
2. Calculate the total revenue generated from pizza sales.
3. Identify the highest-priced pizza.
4. Identify the most common pizza size ordered.
5. List the top 5 most ordered pizza types along with their quantities.



1. RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

```
• SELECT  
|   COUNT(order_id) AS total_orders  
• FROM  
|   orders;
```



2. CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

```
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
```



SQL_PROJECT
ON
PIZZA_SALES

3. 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;
```



SQL_PROJECT
ON
PIZZA_SALES

4. IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

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



5. LIST THE TOP 5 MOST ORDERED PIZZA TYPES ALONG WITH THEIR QUANTITIES.

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



HERE ARE SOME INTERMEDIATE SALES ANALYSIS..

1. Join the necessary tables to find the total quantity of each pizza category ordered.
2. Determine the distribution of orders by hour of the day.
3. Join relevant tables to find the category-wise distribution of pizzas.
4. Group the orders by date and calculate the average number of pizzas ordered per day.
5. Determine the top 3 most ordered pizza types based on revenue.



1. JOIN THE NECESSARY TABLES TO FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED.

```
SELECT
    pizza_types.category, SUM(order_details.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;
```



2. DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

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



SQL_PROJECT

ON
PIZZA_SALES

3. JOIN RELEVANT TABLES TO FIND THE CATEGORY-WISE DISTRIBUTION OF PIZZAS.

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



4. GROUP THE ORDERS BY DATE AND CALCULATE THE AVERAGE NUMBER OF PIZZAS ORDERED PER DAY.

```
SELECT
    ROUND(AVG(quantity), 0) as Average_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;
```



5. 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 pizzas.pizza_id = order_details.pizza_id
GROUP BY pizza_types.name
ORDER BY revenue DESC
LIMIT 3;
```





SQL_PROJECT

PIZZA^{ON}_SALES

HERE ARE SOME ADVANCED SALES ANALYSIS..

1. Calculate the percentage contribution of each pizza type to total revenue.
2. Analyze the cumulative revenue generated over time.
3. Determine the top 3 most ordered pizza types based on revenue for each pizza category.



1. CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE.

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



2. 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
    |     |     pizzas
    JOIN order_details
    ON
    |     pizzas.pizza_id  = order_details.pizza_id
    JOIN
    orders
    ON
    |     order_details.order_id= orders.order_id
GROUP BY
    |     orders.order_date) AS sales;
```



3. 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(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 pizzas.pizza_id = order_details.pizza_id
    GROUP BY pizza_types.category, pizza_types.name) as a) as b
    WHERE rn<=3;
```





CONTACT US

You can order the food from the mobile or by email.

+91 9352197049

sharmarj0809@gmail.com

175. Lajpath Nagar, Sch. 2 ,Alwar (Raj.)





SQL_PROJECT
ON
PIZZA_SALES

THANK YOU

