



QUESTIONS AND
QUERIES

Pizza Sales by using SQL queries

BY ADARSH KUMAR CHOUDHARY

Objectives:

- **Data Analysis:** Utilize SQL queries to extract meaningful insights from the pizza sales data.
- **Performance Metrics:** Identify key metrics such as top-selling pizzas, peak order times, and customer preferences.
- **Business Insights:** Provide actionable insights to improve sales strategy and operational efficiency

Project Overview:

This presentation delves into the analysis of pizza sales data using MySQL. The project is based on a comprehensive dataset that includes:

- Order Details: Information about individual items in each order.
- Orders: Data regarding the orders placed, including timestamps and order IDs.
- Pizza Types: Details about different pizza types offered, including ingredients and category.
- Pizzas: Information on pizza sizes and their respective pricing.

Tools and Technologies:

Database Management System: MySQL

Data Source: Excel Files (Order Details, Orders, Pizza Types, Pizzas)

Analysis Method: Structured Query Language (SQL)

QUESTION BANK

Basic:

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.

Basic:

6. Retrieve the total number of orders placed.
7. Calculate the total revenue generated from pizza sales.
8. Identify the highest-priced pizza.
9. Identify the most common pizza size ordered.
10. List the top 5 most ordered pizza types along with their quantities.

Intermediate:

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..

Advanced:

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.

BASIC QUESTIONS

1. Retrieve the total number of orders placed.

SELECT

COUNT(order_id) AS total_Order

FROM

orders;

OUTPUT =>

Result Grid	
	total_Order
▶	21350

2. Calculate the total revenue generated from pizza sales.

```
SELECT  
    ROUND(SUM(order_details.quantity * pizzas.price),  
        2) AS Total_Sales  
01   FROM  
        order_details  
        JOIN  
        pizzas ON order_details.pizza_id = pizzas.pizza_id;  
05
```

OUTPUT =>

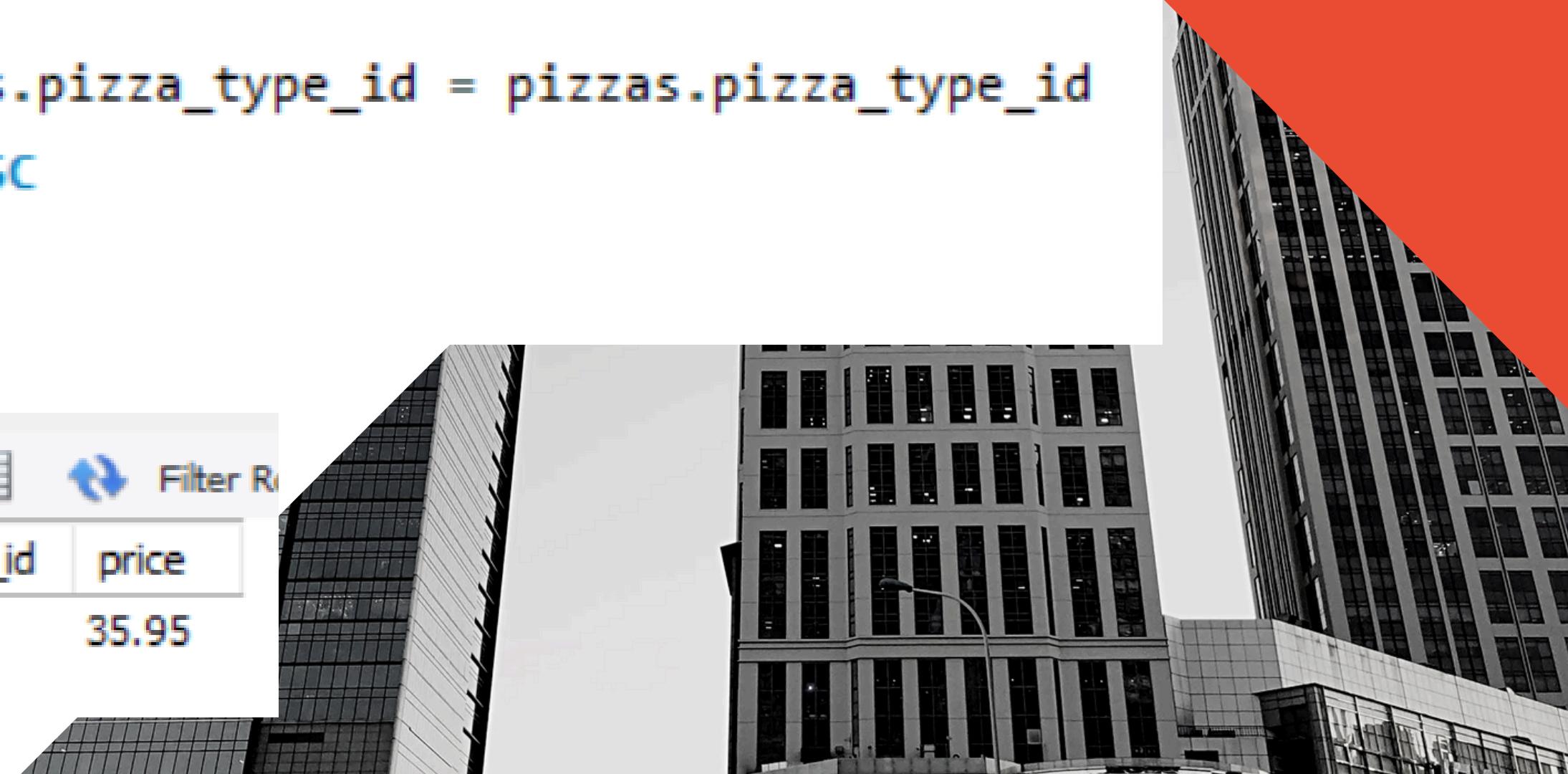
Result Grid	
	Total_Sales
▶	817860.05

3. Identify the highest-priced pizza.

```
SELECT  
    pizza_types.pizza_type_id, pizzas.price  
FROM  
    pizza_types  
        JOIN  
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id  
ORDER BY pizzas.price DESC  
LIMIT 1;
```

OUTPUT =>

	pizza_type_id	price
▶	the_greek	35.95



4. Identify the most common pizza size ordered.

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

OUTPUT =>

Result Grid |

	size	Order_Count
▶	L	18526

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

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

OUTPUT =>

	name	Quantity
▶	The Classic Deluxe Pizza	2453
	The Barbecue Chicken Pizza	2432
	The Hawaiian Pizza	2422
	The Pepperoni Pizza	2418
	The Thai Chicken Pizza	2371

INTERMEDIATE QUESTIONS

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

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

Result Grid		
	category	quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

<= OUTPUT

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);
```

OUTPUT =>

	HOUR(order_time)	COUNT(order_id)
▶	11	1231
	12	2520
	13	2455
	14	1472
	15	1468
	16	1920
	17	2336
	18	2399
	19	2009
	20	1642
	21	1198
	22	663
	23	28
	10	8
	9	1

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

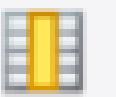
```
SELECT category, COUNT(name) FROM pizza_types  
GROUP BY category;
```

OUTPUT =>

	category	COUNT(name)
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9

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

```
SELECT  
    ROUND(AVG(quantity), 0)  
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  
    ORDER BY quantity) AS order_quantity;
```

Result Grid		 Filter Row
<pre>ROUND(AVG(quantity), 0)</pre>		
▶	138	

<= **OUTPUT**

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

SELECT

```
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.name

ORDER BY total_revenue DESC

LIMIT 3;

OUTPUT =>

	name	total_revenue
▶	The Thai Chicken Pizza	43434.25
▶	The Barbecue Chicken Pizza	42768
▶	The California Chicken Pizza	41409.5

ADVANCE QUESTIONS

1. 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 order_details.pizza_id = pizzas.pizza_id) * 100,  
    2) 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  
ORDER BY total_revenue DESC;
```

OUTPUT =>

Result Grid | Filter Rows

	category	total_revenue
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68

2. Analyze the cumulative revenue generated over time.

```
SELECT order_date,  
       SUM(total_price) OVER(ORDER BY order_date) AS cum_revenue  
  FROM  
(SELECT orders.order_date , SUM(order_details.quantity * pizzas.price) as total_price  
    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 revenue_perday;
```

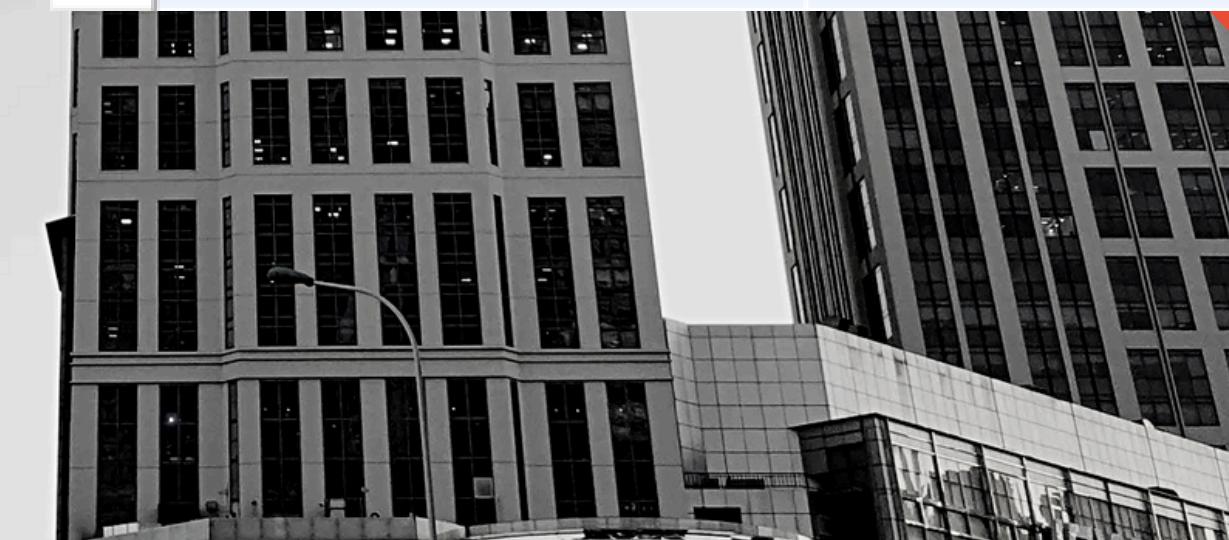
OUTPUT =>

	order_date	cum_revenue
▶	2015-01-01	2713.850000000004
	2015-01-02	5445.75
	2015-01-03	8108.15
	2015-01-04	9863.6
	2015-01-05	11929.55
	2015-01-06	14358.5
	2015-01-07	16560.7
	2015-01-08	19399.05
	2015-01-09	21526.4
	2015-01-10	23990.35000000002
	2015-01-11	25862.65
	2015-01-12	27781.7

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

Result Grid		
	name	revenue
▶	The Thai Chicken Pizza	43434.25
	The Barbecue Chicken Pizza	42768
	The California Chicken Pizza	41409.5
	The Classic Deluxe Pizza	38180.5
	The Hawaiian Pizza	32273.25
	The Pepperoni Pizza	30161.75
	The Spicy Italian Pizza	34831.25
	The Italian Supreme Pizza	33476.75
	The Sicilian Pizza	30940.5
	The Four Cheese Pizza	32265.7000000006





Thank You