

Pizza Sales Data Analysis: An SQL Deep Dive

Goal: Utilizing Advanced SQL Techniques (Joins, Sub-queries, Window Functions)

- **Primary Tool:** SQL (MySQL Workbench)
- **Data Source:** Pizza Sales Dataset
- **Reporting/Visualization:** Canva

Presented By: **Riya Bhatt**

LinkedIn: <https://www.linkedin.com/in/riya-bhatt-3246852b8/>

Gmail: riyabhatt005@gmail.com

Project Objectives and Data Schema Overview

Objective : To transform raw, multi-table sales data into meaningful business metrics using advanced SQL queries.

Data Schema Overview: There are 4 tables orders, orders_details, pizzas, pizza_types

1. **orders** : Records of each transaction (Date and Time). order_id, order_date, order_time
-

2. ***order_details*** : Details of items within each order. *order_details_id*, *order_id* (FK), *pizza_id* (FK), quantity.
 3. ***pizzas*** : Lists all unique pizzas, their sizes, and price. *pizza_id*, *pizza_type_id* (FK), size, price.
 4. ***pizza_types*** : Lists the names, categories, and ingredients for each pizza type. *pizza_type_id*, name, category.
-

Key Business Questions & Analysis Tiers

Basic:

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

Intermediate:

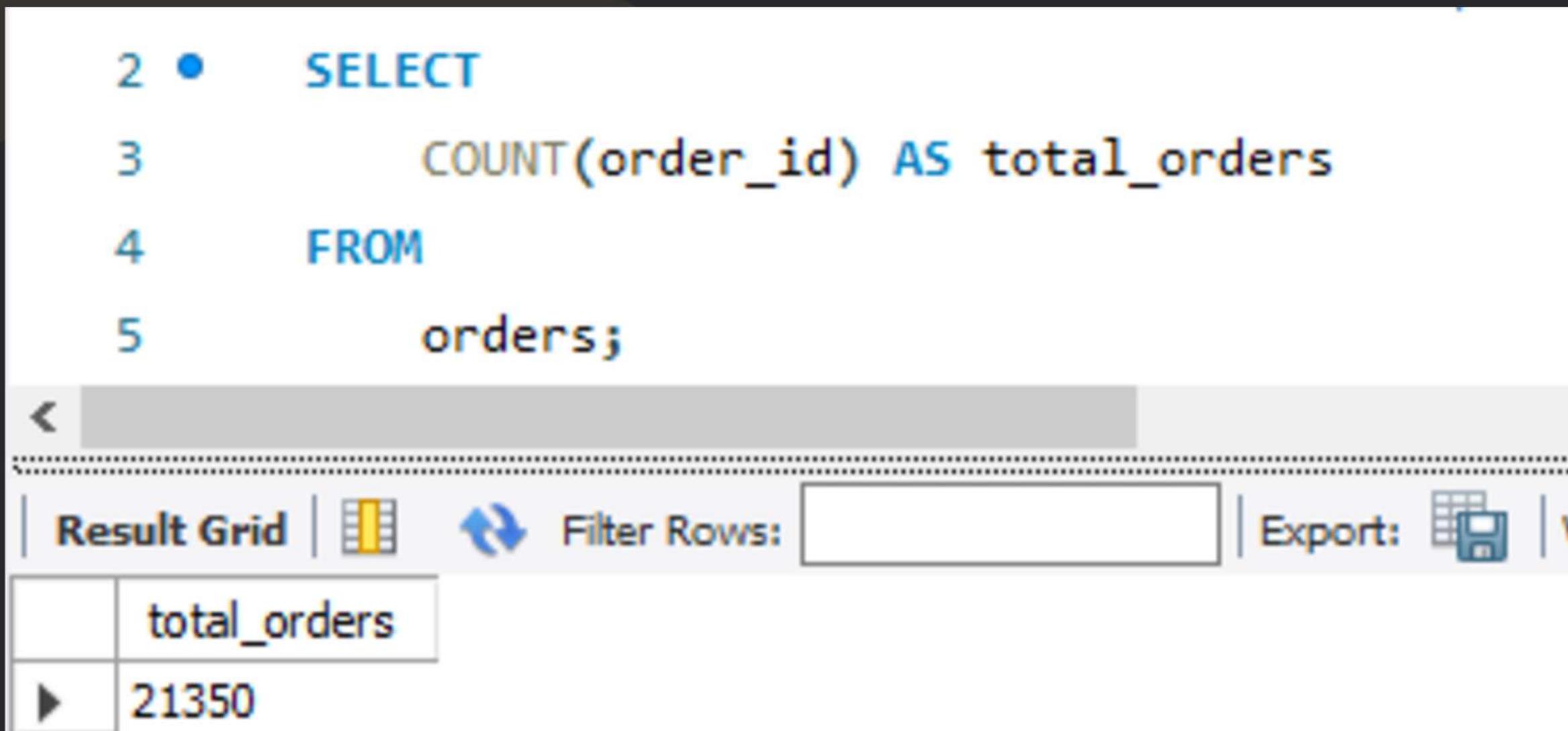
- Join the necessary tables to find the total quantity of each pizza category ordered.
 - Determine the distribution of orders by hour of the day.
 - Join relevant tables to find the category-wise distribution of pizzas.
 - Group the orders by date and calculate the average number of pizzas ordered per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

Advanced:

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

1. Retrieve the total number of orders placed.

```
2 •   SELECT  
3       COUNT(order_id) AS total_orders  
4   FROM  
5       orders;
```



The screenshot shows the MySQL Workbench interface. At the top, there is a code editor window containing the SQL query. Below it is a results grid window titled "Result Grid". The results grid displays a single row with one column, labeled "total_orders", which contains the value "21350". There are also buttons for "Filter Rows:" and "Export:".

total_orders
21350

2. Calculate the total revenue generated from pizza sales.

```
2 •  SELECT
3      SUM(orders_details.quantity * pizzas.price) AS total_sales
4  FROM
5      orders_details
6      JOIN
7      pizzas ON pizzas.pizza_id = orders_details.pizza_id;
```

The screenshot shows a MySQL query editor interface. At the top, there is a code editor containing the SQL query. Below the code editor is a toolbar with buttons for 'Result Grid' (selected), 'Filter Rows', 'Export', and 'Wrap Cell Content'. The main area displays a result grid with one row. The first column is labeled 'total_sales' and contains the value '817860.049999993'.

	total_sales
▶	817860.049999993

3. Identify the highest-priced pizza.

```
2 •   SELECT
3       pizza_types.name, pizzas.price
4   FROM
5       pizza_types
6       JOIN
7       pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
8   ORDER BY pizzas.price DESC
9   LIMIT 1;
```

The screenshot shows a MySQL query editor interface. At the top, there is a code editor containing the SQL query. Below the code editor is a toolbar with several buttons: 'Result Grid' (selected), 'Filter Rows', 'Export', 'Wrap Cell Content', and 'Fetch rows'. The main area displays a table with two columns: 'name' and 'price'. A single row is shown, representing the highest-priced pizza found by the query.

	name	price
▶	The Greek Pizza	35.95

4. Identify the most common pizza size ordered.

```
2 •   SELECT
3       pizza_types.name, pizzas.price
4   FROM
5       pizza_types
6           JOIN
7       pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
8   ORDER BY pizzas.price DESC
9   LIMIT 1;
```

The screenshot shows a MySQL query results interface. At the top, there is a code editor window containing the SQL query. Below it is a toolbar with various buttons: 'Result Grid' (selected), 'Filter Rows', 'Export', 'Wrap Cell Content', and 'Fetch rows'. The main area displays a table with one row of data. The table has two columns: 'name' and 'price'. The data row contains 'The Greek Pizza' in the 'name' column and '35.95' in the 'price' column. There are navigation arrows at the bottom left of the table.

	name	price
▶	The Greek Pizza	35.95

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

```
3 •   SELECT
4     pizza_types.name, SUM(orders_details.quantity) AS qty
5   FROM
6     pizza_types
7     JOIN
8       pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
9     JOIN
10       orders_details ON orders_details.pizza_id = pizzas.pizza_id
11   GROUP BY pizza_types.name
12   ORDER BY qty DESC
13   LIMIT 5;
```

Result Grid		Filter Rows:
	name	qty
▶	The Classic Deluxe Pizza	2453
	The Barbecue Chicken Pizza	2432
	The Hawaiian Pizza	2422
	The Pepperoni Pizza	2418
	The Thai Chicken Pizza	2371

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

```
1 -- Join the necessary tables to find the total quantity of each pizza category ordered.
2 • SELECT
3     pizza_types.category,
4     SUM(orders_details.quantity) AS quantity
5   FROM
6     pizza_types
7       JOIN
8     pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
9       JOIN
10    orders_details ON orders_details.pizza_id = pizzas.pizza_id
11   GROUP BY pizza_types.category
12   ORDER BY quantity DESC;
```

	category	quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

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

```
2 •   SELECT
3       HOUR(order_time) AS hour, COUNT(order_id) AS order_count
4   FROM
5       orders
6   GROUP BY HOUR(order_time);
```

hour	order_count
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

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

```
3 •   SELECT
4       category, COUNT(name)
5   FROM
6       pizza_types
7   GROUP BY category;
```

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

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

```
2 •   SELECT
3     AVG(quantity)
4   FROM
5     (SELECT
6       orders.order_date, SUM(orders_details.quantity) AS quantity
7     FROM
8       orders
9     JOIN orders_details ON orders.order_id = orders_details.order_id
10    GROUP BY orders.order_date) order_quantity;
```

avg_pizza_order_perDay
138.4749

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

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

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

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

```
2 • SELECT
3     pizza_types.category,
4     ROUND(SUM(orders_details.quantity * pizzas.price) / (SELECT
5         ROUND(SUM(orders_details.quantity * pizzas.price),
6             2) AS total_sales
7     FROM
8         orders_details
9         JOIN
10        pizzas ON pizzas.pizza_id = orders_details.pizza_id) * 100,
11    2) AS revenue
12
13    FROM
14        pizza_types
15        JOIN
16        pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
17        JOIN
18        orders_details ON orders_details.pizza_id = pizzas.pizza_id
19    GROUP BY pizza_types.category
20    ORDER BY revenue DESC;
```

	category	revenue
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68

12. 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(orders_details.quantity*pizzas.price) as revenue  
        from orders_details join pizzas  
          on orders_details.pizza_id=pizzas.pizza_id  
        join orders  
          on orders.order_id=orders_details.order_id  
       group by orders.order_date) as sales;
```

	order_date	cum_revenue
▶	2015-01-01	2713.8500000000004
	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
	2015-01-13	29831.30000000003
	2015-01-14	32358.70000000004
	2015-01-15	34343.50000000001
.		

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

```
select category, 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((orders_details.quantity)*pizzas.price) as revenue
from pizza_types join pizzas
on pizza_types.pizza_type_id=pizzas.pizza_type_id
join orders_details
on orders_details.pizza_id=pizzas.pizza_id
group by pizza_types.category,pizza_types.name)as a)as b
where rn<=3;
```

category	name	revenue
Chicken	The Thai Chicken Pizza	43434.25
Chicken	The Barbecue Chicken Pizza	42768
Chicken	The California Chicken Pizza	41409.5
Classic	The Classic Deluxe Pizza	38180.5
Classic	The Hawaiian Pizza	32273.25
Classic	The Pepperoni Pizza	30161.75
Supreme	The Spicy Italian Pizza	34831.25
Supreme	The Italian Supreme Pizza	33476.75
Supreme	The Sicilian Pizza	30940.5
Veggie	The Four Cheese Pizza	32265.7000
Veggie	The Mexicana Pizza	26780.75
Veggie	The Five Cheese Pizza	26066.5

Overall Business Performance

- **Total Orders :** 21025
- **Total Sales :** \$817860
- **5 Most Ordered Pizzas :**

The Classic Deluxe Pizza - 2453

The Barbecue Chicken Pizza - 2432

The Hawaiian Pizza - 2422

The Pepperoni Pizza - 2418

The Thai Chicken Pizza - 2371

- **Average number of pizzas ordered per day :** 138
-

Thank You

LinkedIn : <https://www.linkedin.com/in/riya-bhatt-3246852b8/>

Gmail : riyabhatt005@gmail.com

GitHub :<https://github.com/riyabhatts>