



SQL

Pizza Sales

SQL PROJECT ON PIZZA SALES

- WHERE EVERY PAGE TELLS A NEW CONCEPT



SQL

DML

DDL

Pratap Mishra



SQL

Pizza Sales

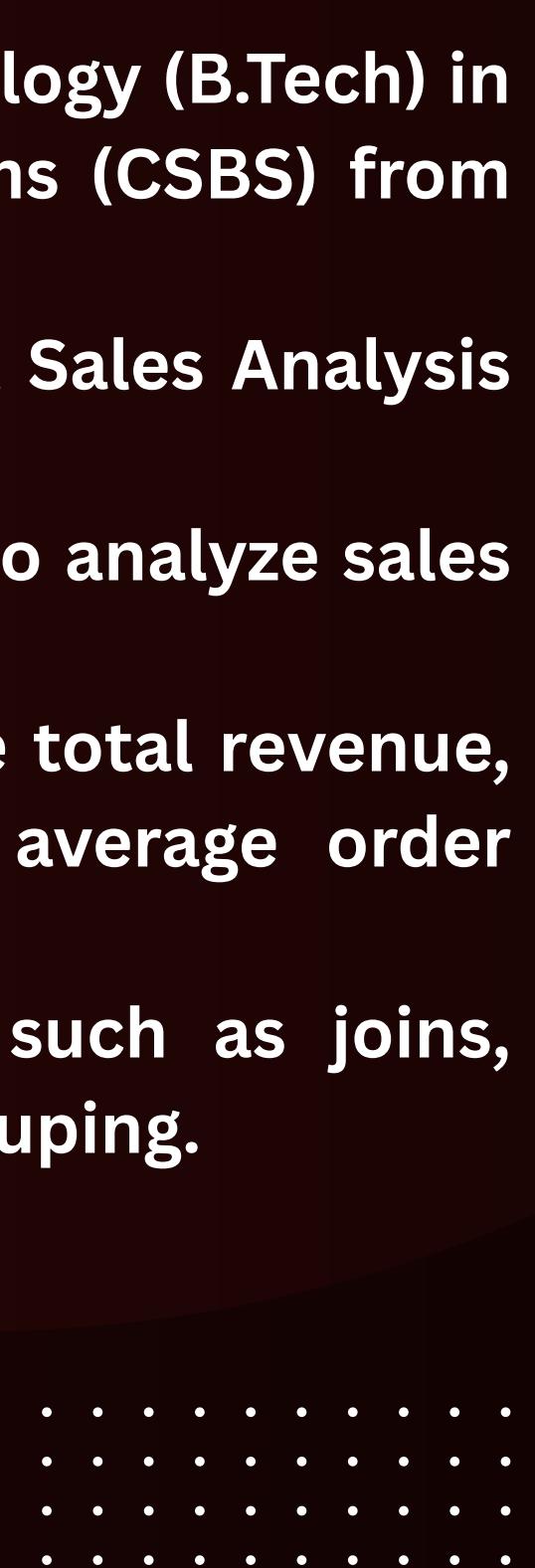
Home

About

Contact

INTRODUCTION

- My name is Pratap Kumar Mishra.
- Currently pursuing a Bachelor of Technology (B.Tech) in Computer Science and Business Systems (CSBS) from the 2022–2026 batch.
- I have worked on a project titled “Pizza Sales Analysis Using SQL.”
- In this project, I solved 13 SQL queries to analyze sales data.
- The queries helped me find insights like total revenue, best-selling pizzas, sales trends, and average order values.
- This project improved my SQL skills such as joins, aggregate functions, subqueries, and grouping.





SQL

Pizza sales

DATASET DESCRIPTION

About

Contact

Dataset: Pizza Sales Data

Key Tables:

- **Pizzas** (`pizza_id`, `pizza_type_id`, `size`, `price`)
- **Pizza_Types** (`pizza_type_id`, `name`, `category`, `ingredients`)
- **Orders** (`order_id`, `order_date`, `order_time`)
- **Order_Details** (`order_details_id`, `order_id`, `pizza_id`, `quantity`)



SQL

Pizza sales

SQL CONCEPTS USED

Home

About

Contact

- Basic **SELECT** queries
- Filtering with **WHERE**
- Joins
- GROUP BY and aggregate functions (SUM, COUNT, AVG, MAX, MIN)
- Subqueries
- ORDER BY and LIMIT
- Aliasing



SQL

Pizza sales

SAMPLE QUESTIONS SOLVED (13 QUESTIONS)

About

Contact

1. Retrieve the total number of order 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 there quantities.
6. Join the necessary tables to find the total quantity of each pizza category ordered.
7. Determine the distribution of orders by hour of the day.
8. Join relevant tables to find the category-wise distribution of pizzas.
9. Group the orders by date and calculate the average number of pizzas ordered per day.
10. Determine the top 3 most ordered pizza types based on revenue.
11. Calculate the percentage contribution of each pizza type to total revenue.
12. Analyze the cumulative revenue generated over tiem.
13. Determine the top 3 most ordered pizza types based on revenue for each pizza category.



SQL

Pizza sales

[Home](#)[About](#)[Contact](#)

1. Retrieve the total number of order placed.

```
SELECT  
    COUNT(order_id) AS total_numberOfOrder  
FROM  
    orders;
```

Result Grid	
	total_numberOfOrder
▶	21350



SQL

Pizza sales

[Home](#)[About](#)[Contact](#)

2. Calculate the total revenue generated from pizza sales.

```
-- 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;
```

Result Grid		Filter Rows:
	total_sales	
▶	817860.05	



SQL

Pizza sales

3. Identify the highest priced pizza.

[Home](#)[About](#)[Contact](#)

```
-- Identify the highest priced pizza
• SELECT
    pizza_types.name, pizzas.price AS highest_priced_pizza
  FROM
    pizza_types
    JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
 ORDER BY pizzas.price DESC
 LIMIT 1;
```

Result Grid | Filter Rows:

	name	highest_priced_pizza
▶	The Greek Pizza	35.95



SQL

Pizza sales

[Home](#)[About](#)[Contact](#)

4. Identify the most common pizza size ordered.

```
-- 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;
```

Result Grid | Filter Rows:

	size	order_count
▶	L	18526



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

```
-- 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;
```

	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



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

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

	category	qty
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050



SQL

Pizza sales

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

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

```
SELECT  
    HOUR(order_time) AS hour, COUNT(order_id) AS order_count  
FROM  
    orders  
GROUP BY HOUR(order_time)  
ORDER BY HOUR(order_time) ASC;
```

	hour	order_count
▶	9	1
	10	8
	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.

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

```
SELECT
```

```
    category, COUNT(name)
```

```
FROM
```

```
    pizza_types
```

```
GROUP BY category;
```

Result Grid | Filter Rows:

	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.

```
-- Group the orders by date and  
-- calculate the average number of pizzas ordered per day.  
  
• SELECT ROUND(AVG(quantity),0) AS daily_order_avg  
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 daily_order;
```

	daily_order_avg
▶	138



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

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

Result Grid | Filter Rows:

	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.

```
-- 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  
    ) * 100,  
    2) 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;
```

Result Grid | Filter

	category	revenue
▶	Classic	26.91
	Veggie	23.68
	Supreme	25.46
	Chicken	23.96



SQL

Pizza sales

12. Analyze the cumulative revenue generated over time.

```
-- Analyze the cumulative revenue generated over time.  
SELECT order_date,  
ROUND(SUM(revenue) OVER(ORDER BY order_date),2) AS cum_revenue  
FROM  
(SELECT  
    orders.order_date,  
    ROUND(SUM(order_details.quantity * pizzas.price),  
         2) AS revenue  
  FROM  
    order_details  
    JOIN  
    pizzas ON order_details.pizza_id = pizzas.pizza_id  
    JOIN  
    orders ON order_details.order_id = orders.order_id  
  GROUP BY orders.order_date) AS sales;
```

order_date	cum_revenue
2015-01-01	2713.85
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	23990.35
2015-01-11	25862.65
2015-01-12	27781.7
2015-01-13	29831.3
2015-01-14	32358.7
2015-01-15	34343.5
2015-01-16	36937.65
2015-01-17	39001.75
2015-01-18	40978.6
2015-01-19	43365.75
2015-01-20	45763.65
2015-01-21	47804.2
2015-01-22	50300.9
2015-01-23	52724.6
2015-01-24	55013.85
2015-01-25	56631.4

order_date	cum_revenue
2015-01-25	56631.4
2015-01-26	58515.8
2015-01-27	61043.85
2015-01-28	63059.85
2015-01-29	65105.15
2015-01-30	67375.45
2015-01-31	69793.3
2015-02-01	72982.5
2015-02-02	75311.1
2015-02-03	77925.9
2015-02-04	80159.8
2015-02-05	82375.6
2015-02-06	84885.55
2015-02-07	87123.2
2015-02-08	89158.2
2015-02-09	91353.55
2015-02-10	93410.05
2015-02-11	95870.05
2015-02-12	98028.85
2015-02-13	100783.35
2015-02-14	103102.5
2015-02-15	105243.75
2015-02-16	107212.55
2015-02-17	109334.45
2015-02-18	111977.3
2015-02-19	114007.55

order_date	cum_revenue
2015-02-19	114007.55
2015-02-20	116898.7
2015-02-21	119009.7
2015-02-22	120589.65
2015-02-23	122758.2
2015-02-24	124952.75
2015-02-25	127294.05

2015-02-26	129555.35
2015-02-27	132413.3
2015-02-28	134952.9
2015-03-01	136551.45
2015-03-02	138930.5
2015-03-03	141218.4
2015-03-04	143662.7
2015-03-05	146013.35
2015-03-06	148527.3
2015-03-07	150027.75



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

```
-- Determine the top 3 most ordered pizza types
-- based on revenue for each pizza category
SELECT rn, category, name, revenue
FROM
(SELECT category, name, revenue,
RANK() OVER(PARTITION BY category ORDER BY revenue DESC) as rn
FROM
(SELECT
pizza_types.name,
pizza_types.category,
SUM(order_details.quantity * pizzas.price) AS revenue
FROM
order_details
JOIN
pizzas ON order_details.pizza_id = pizzas.pizza_id
JOIN
pizza_types ON pizza_types.pizza_type_id = pizzas.pizza_type_id
GROUP BY category , name) AS a) AS b
WHERE rn <=3;
```

	rn	category	name	revenue
▶	1	Chicken	The Thai Chicken Pizza	43434.25
	2	Chicken	The Barbecue Chicken Pizza	42768
	3	Chicken	The California Chicken Pizza	41409.5
	1	Classic	The Classic Deluxe Pizza	38180.5
	2	Classic	The Hawaiian Pizza	32273.25
	3	Classic	The Pepperoni Pizza	30161.75
	1	Supreme	The Spicy Italian Pizza	34831.25
	2	Supreme	The Italian Supreme Pizza	33476.75
	3	Supreme	The Sicilian Pizza	30940.5
	1	Veggie	The Four Cheese Pizza	32265.70000000065
	2	Veggie	The Mexicana Pizza	26780.75
	3	Veggie	The Five Cheese Pizza	26066.5



SQL

Pizza sales

SQL

DML

DDL

THANK YOU

FOR ATTENTION

- PIZZA SALES PRESENTATION