



# PIZZA SALES SQL PROJECT

PRESENTED BY  
YOGESH GUNJAL



# PIZZA SALES SQL PROJECT OVERVIEW

**Objective:** Analyze pizza sales data to derive business insights using SQL.

**Project Workflow:**

## 1. Data Preparation:

- Created and populated tables with pizza sales data.

## 2. Basic Analysis:

- Total Orders: Retrieved the number of orders.
- Revenue: Calculated total sales revenue.
- Pricing: Identified the highest-priced pizza.
- Order Patterns: Analyzed the most common pizza size and top 5 ordered pizzas.

## 3. Intermediate Analysis:

- Category Insights: Total quantity per pizza category.
- Time Analysis: Order distribution by hour.
- Sales Trends: Average pizzas ordered per day and top 3 pizzas by revenue.

## 4. Advanced Analysis:

- Revenue Breakdown: Percentage contribution per pizza type.
- Cumulative Trends: Revenue growth over time.
- Detailed Insights: Top 3 pizzas by revenue in each category.

# QUESTIONS

## 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.

## 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.



## Retrieve the total number of orders placed.

```
## 1) Retrieve the total number of orders placed.
```

```
SELECT
```

```
    COUNT(order_id) AS total_number
```

```
FROM
```

```
    pizzahut.orders;
```



total_number
21350

## Calculate the total revenue generated from pizza sales.

```
## 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 order_details.pizza_id = pizzas.pizza_id;
```



Result Grid	
	Total_Sales
▶	817860.05

## Identify the highest-priced pizza.

```
## 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;
```



	pizza_type_id	price
▶	the_greek	35.95

## Identify the most common pizza size ordered.

```
## 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;
```



Result Grid |

	size	order_count
▶	L	18526
◀	M	15385
▴	S	14137
▾	XL	544

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

```
## 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 pizzas.pizza_id = order_details.pizza_id  
GROUP BY pizza_types.name  
ORDER BY quantity DESC  
LIMIT 5;
```



Result Grid |  Filter Rows:

	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

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

```
##6)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 pizzas.pizza_type_id = pizza_types.pizza_type_id  
    JOIN  
    order_details ON pizzas.pizza_id = order_details.pizza_id  
GROUP BY (pizza_types.category);
```



Result Grid | Filter

	category	quantity
▶	Classic	14888
	Veggie	11649
	Supreme	11987
	Chicken	11050

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

```
##7)Determine the distribution of orders by hour of the day.
```

```
SELECT
```

```
    HOUR(order_time), COUNT(order_id) AS Total_orders
```

```
FROM
```

```
    orders
```

```
GROUP BY HOUR(order_time);
```



	HOUR(order_time)	Total_orders
▶	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

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

```
##8)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

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

```
##9)Group the orders by date and calculate the average number of pizzas ordered per day.  
SELECT  
    ROUND(AVG(quantity), 0) as Avg_number_of_pizzas_ordered_per_day  
FROM  
(SELECT  
    orders.order_date AS date,  
    SUM(order_details.quantity) AS quantity  
FROM  
    orders  
JOIN order_details ON orders.order_id = order_details.order_id  
GROUP BY (date)) AS Total_quantity_per_day;
```



Result Grid			Filter Rows:	<input type="text"/>	E
Avg_number_of_pizzas_ordered_per_day					
138					

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

```
##10)Determine the top 3 most ordered pizza types based on revenue.  
SELECT  
    pizza_types.name,  
    SUM(pizzas.price * order_details.quantity) 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;
```



Result Grid | Filter Rows:

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

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

```
##11) 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(pizzas.price * order_details.quantity),  
            2) AS Tptal_Sales  
    FROM  
        pizzas  
        JOIN  
            order_details ON pizzas.pizza_id = order_details.pizza_id)*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 Rows:

	category	revenue
▶	Classic	26.91
	Veggie	23.68
	Supreme	25.46
	Chicken	23.96

## Analyze the cumulative revenue generated over time.

```
##12)Analyze the cumulative revenue generated over time.  
SELECT order_date, total_revenue, SUM(total_revenue) over(order by order_date) as cum_revenue from  
(SELECT  
    orders.order_date,  
    ROUND(SUM(order_details.quantity * pizzas.price),  
        0) AS total_revenue  
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 sales;
```



Result Grid		Filter Rows:	E
	order_date	total_revenue	cum_revenue
1	2015-01-01	2714	2714
2	2015-01-02	2732	5446
3	2015-01-03	2662	8108
4	2015-01-04	1755	9863
5	2015-01-05	2066	11929
6	2015-01-06	2429	14358
7	2015-01-07	2202	16560
8	2015-01-08	2838	19398
9	2015-01-09	2127	21525
10	2015-01-10	2464	23989
11	2015-01-11	1872	25861
12	2015-01-12	1919	27780
13	2015-01-13	2050	29830
14	2015-01-14	2527	32357
15	2015-01-15	1985	34342
16	2015-01-16	2594	36936
17	2015-01-17	2064	39000
18	2015-01-18	1977	40977
19	2015-01-19	2387	43364
20	2015-01-20	2398	45762
21	2015-01-21	2041	47803
22	2015-01-22	2497	50300

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

```
##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, round(sum(order_details.quantity*pizzas.price),0) 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 b) as a where rn<=3;
```



	category	name	revenue
▶	Chicken	The Thai Chicken Pizza	43434
	Chicken	The Barbecue Chicken Pizza	42768
	Chicken	The California Chicken Pizza	41410
Classic	Classic	The Classic Deluxe Pizza	38180
	Classic	The Hawaiian Pizza	32273
Supreme	Classic	The Pepperoni Pizza	30162
	Supreme	The Spicy Italian Pizza	34831
Supreme	Supreme	The Italian Supreme Pizza	33477
	Supreme	The Sicilian Pizza	30940
	Veggie	The Four Cheese Pizza	32266
Veggie	Veggie	The Mexicana Pizza	26781
	Veggie	The Five Cheese Pizza	26066

Thank  
you!