



PIZZA

Pizza_Sales
Using
SQL_Queries

PIZZA

INTRODUCTION

The project involves analyzing a pizza sales dataset using SQL queries to uncover trends and insights, such as identifying best-selling pizzas, peak sales periods, and customer purchasing behavior. The goal is to optimize inventory, enhance marketing strategies, and improve overall business performance based on the data analysis.

PIZZA

PROBLEMS

- 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.
- 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.
- 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.

PIZZA

Retrieve the total number
of orders placed?

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

Result Grid	
	total_orders
▶	21350

PIZZA

CALCULATE THE TOTAL
REVENUE GENERATED FROM
PIZZA SALES.

```
SELECT  
sum(order_details.quantity*pizzas.price) AS total_sales  
FROM order_details JOIN pizzas  
ON pizzas.pizza_id=order_details.pizza_id;
```

Result Grid



	total_sales
▶	817860.04999

PIZZA

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;
```

Result Grid



Filter Rows:

	name	price
▶	The Greek Pizza	35.95

PIZZA

IDENTIFY THE MOST
COMMON PIZZA SIZE
ORDERED.

```
SELECT count(order_details.order_details),pizzas.size  
FROM order_details Join pizzas  
ON order_details.pizza_id=pizzas.pizza_id  
GROUP BY pizzas.size  
ORDER BY count(order_details.order_details) DESC limit 1;
```

Result Grid			Filter Rows:
	count(order_details.order_details)	size	
▶	18526	L	

PIZZA

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;
```

Result Grid			Filter Rows:	Export:
	name	sum(order_details.quantity)		
▶	The Classic Deluxe Pizza	2453		
	The Barbecue Chicken Pizza	2432		
	The Hawaiian Pizza	2422		
	The Pepperoni Pizza	2418		
	The Thai Chicken Pizza	2371		

PIZZA

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 pizzas.pizza_id=order_details.pizza_id
GROUP BY pizza_types.category
ORDER BY sum(order_details.quantity) ;
```

Result Grid			Filter Rows:
	category	sum(order_details.quantity)	
▶	Chicken	11050	
	Veggie	11649	
	Supreme	11987	
	Classic	14888	

PIZZA

DETERMINE THE DISTRIBUTION
OF ORDERS BY HOUR OF THE
DAY.



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

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

PIZZA

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		

PIZZA

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

```
SELECT ROUND(AVG(quantity),0)AS total_no_of_pizza
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;
```

Result Grid

	total_no_of_pizza
▶	138

PIZZA

DETERMINE THE TOP 3 MOST
ORDERED PIZZA TYPES BASED ON
REVENUE FOR EACH PIZZA CATEGORY.

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

Result Grid

Filter Rows:

Export:

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

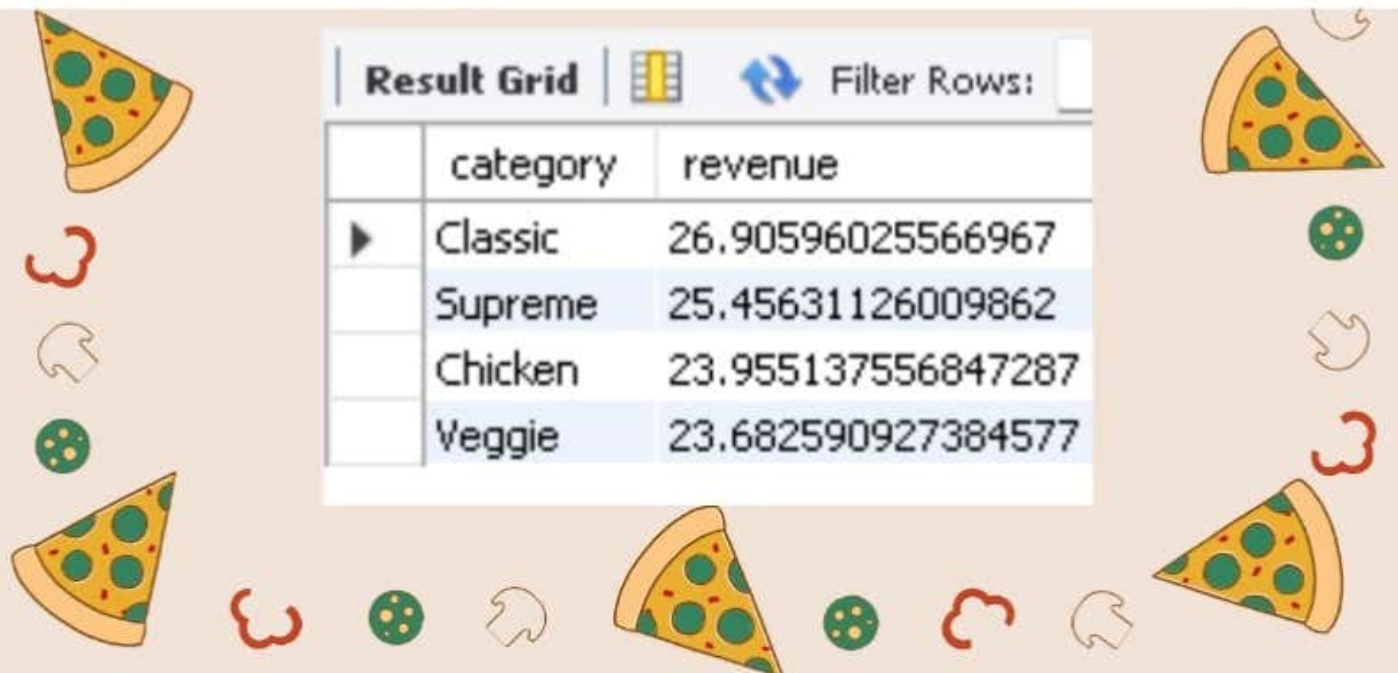


PIZZA

CALCULATE THE PERCENTAGE
CONTRIBUTION OF EACH PIZZA
TYPE TO TOTAL REVENUE

```
SELECT pizza_types.category,  
(SUM(order_details.quantity*pizzas.price)/(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 order_details.pizza_id=pizzas.pizza_id  
GROUP BY pizza_types.category  
ORDER BY revenue DESC;
```

Result Grid			Filter Rows:
	category	revenue	
▶	Classic	26.90596025566967	
	Supreme	25.45631126009862	
	Chicken	23.955137556847287	
	Veggie	23.682590927384577	



PIZZA

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

Result Grid			Filter Rows:
	order_date	cum_revenue	
▶	2015-01-01	2713.8500000000000004	
	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	

PIZZA

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 pizzas.pizza_type_id=pizza_types.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