Pizza Sales Analysis

Using SQL

Business Problem

• Analyze pizza sales data to find customer preferences, revenue patterns, and business insights.

• Why SQL? → Efficient querying & analysis on relational data

Dataset Used

File Name

orders.csv

order_details.csv

pizzas.csv

pizza_types.csv

View Dataset

Description

Contains all order information: order_id, Order_date, Order_Time, etc.

Contains item-level details for each order: Order_details, order_id, pizza_id, quantity

Contains pizza details: pizza_id, pizza_type_id, size, prize

Contains pizza category information: pizza_type_id, name, category, ingredient

Retrieve the total number of orders placed.



Calculate the total revenue generated from pizza sales.

```
SELECT

ROUND(SUM(order_details.quantity * pizzas.price),

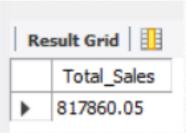
2) AS Total_Sales

FROM

order_details

JOIN

pizzas ON pizzas.pizza_id = order_details.pizza_id;
```



Identify the highest-priced pizza.



Identify the most common pizza size ordered.

Result Grid				Filte
	size	order	coun	t
•	L	18526		
	M	15385		
	S	14137		
	XL	544		
	XXL	28		

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			
	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.

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

R	esult Grid	Filter R
	category	Total_Quantity
١	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

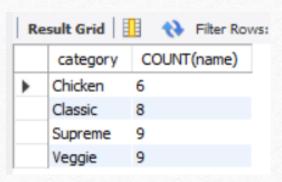
Determine the distribution of orders by hour of the day.

```
3 • SELECT
4 HOUR(order_time), COUNT(order_id)
5 FROM
6 orders
7 GROUP BY HOUR(order_time);
```

Re	Result Grid		
	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	

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

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



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

```
SELECT
ROUND(AVG(quantity), 0) AS Avg_Pizza_Order_PerDay
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
```



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

```
3 •
       SELECT
           pizza_types.name,
           SUM(order_details.quantity * pizzas.price) AS revenue
       FROM
           pizzas
               JOIN
           pizza_types ON pizzas.pizza_type_id = pizza_types.pizza_type_id
10
               JOIN
           order_details ON order_details.pizza_id = pizzas.pizza_id
11
12
       GROUP BY pizza types.name
       ORDER BY revenue DESC
13
14
       LIMIT 3;
```

R	esult Grid 🔠 🙌 Filter Ro	ws:
	name	revenue
١	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.

```
SELECT
           pizza_types.category,
           ROUND(SUM(order_details.quantity * pizzas.price) /(SELECT
                   SUM(order_details.quantity * pizzas.price)
               FROM
                   order_details
                       JOIN
                   pizzas ON order_details.pizza_id = pizzas.pizza_id) * 100,2) AS revenue
10
11
       FROM
           pizza_types
12
               JOIN
           pizzas ON pizza types.pizza type_id = pizzas.pizza type_id
15
               JOIN
           order_details ON order_details.pizza_id = pizzas.pizza_id
       GROUP BY pizza types.category
17
       ORDER BY revenue DESC
18
```

Result Grid			
	category	revenue	
•	Classic	26.91	
	Supreme	25.46	
	Chicken	23.96	
	Veggie	23.68	

Analyze the cumulative revenue generated over time.

```
select order_date ,
sum(revenue) over (order by order_date)
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;
```

R	esult Grid	Filter Rows:	port:
	order_date	sum(revenue) over (order by order_date)	
•	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	

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) as rn
       from
       (select pizza_types.category , pizza_types.name,
       sum(order_details.quantity * pizzas.price) as revenue
 9
       from order details join pizzas
10
       on order details.pizza id = pizzas.pizza id
11
12
       join pizza types
13
       on pizza types.pizza type id = pizzas.pizza type id
14
       group by pizza_types.category , pizza_types.name) as a) as b
15
       where rn <= 3;
```

Re	esult Grid	Export:
	name	revenue
•	The Chicken Pesto Pizza	16701.75
	The Chicken Alfredo Pizza	16900.25
	The Southwest Chicken Pizza	34705.75
	The Pepperoni, Mushroom, and Peppers Pizza	18834.5
	The Big Meat Pizza	22968
	The Napolitana Pizza	24087
	The Brie Carre Pizza	11588.4999999999
	The Spinach Supreme Pizza	15277.75
	The Calabrese Pizza	15934.25
	The Green Garden Pizza	13955.75
	The Mediterranean Pizza	15360.5
	The Spinach Pesto Pizza	15596

Key Insights

- Classic pizzas contributed the highest share of total orders.
- Large size pizzas were the most commonly ordered size.
- Thai Chicken Large was among the highest revenue-generating pizzas.
- **Peak ordering hours**: 12 PM–2 PM and 6 PM–8 PM.
- Monthly sales trend showed December as the best-performing month.

Conclusion

- SQL helped uncover **key insights** like top-selling pizzas, revenue drivers, and customer preferences.
- Showed how data can guide business decisions.
- This project improved my **SQL** + data storytelling skills.

Let's Connect





