SQL PROJECT ON PIZZA SALES

WELCOME TO

🔥"PIZZA SALES ANALYSIS":

This project explores the sales data of a pizza restaurant to uncover meaningful business insights. Using SQL, we analyze various aspects of the dataset, such as order details, pizza types, quantities sold, and order times.

The goal is to identify trends like the most popular pizzas, best-performing days, and peak hours of operation. These insights help in understanding customer behavior and improving overall business strategy.

BASIC ANALYSIS

- 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

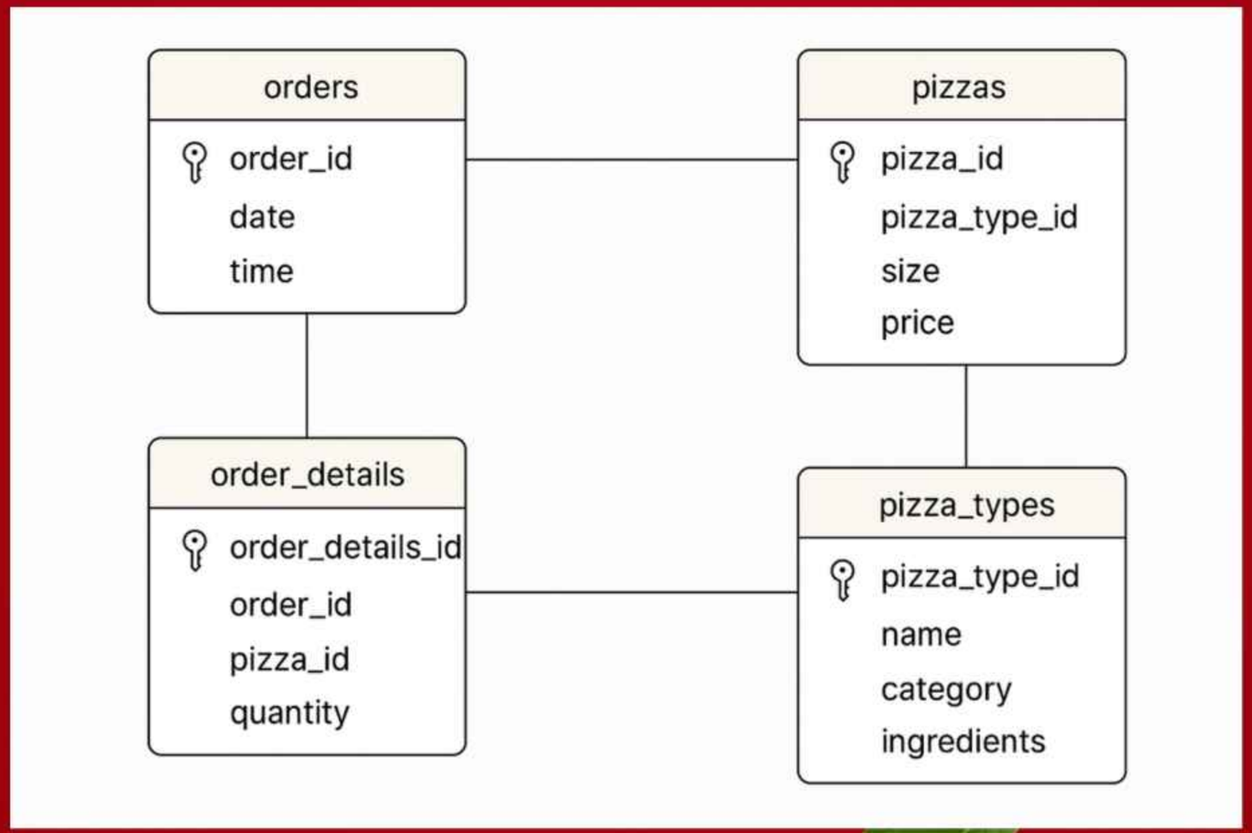
INTERMEDIATE ANALYSIS

- Join necessary tables to find 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 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 ANALYSIS

- 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

DATABASE SCHEMA OVERVIEW



QUS:1 RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

```sql
-- Retrieve the total number of orders placed.

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

OUTPUT :

	total_orders
▶	21350

QUS:2 CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

```sql
-- Calculate the total revenue generated from pizza sales.

SELECT
    ROUND(SUM(orders_details.quantity * pizzas.price),
            2) AS total_sales
FROM
    orders_details
        JOIN
    pizzas ON pizzas.pizza_id = orders_details.pizza_id
```

OUTPUT:

Result Grid | Filter Rows:

	total_sales
▶	817860.05

QES:3 IDENTIFY THE HIGHEST-PRICED PIZZA.

```sql
-- 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;
```

OUTPUT :

	name	price
▶	The Greek Pizza	35.95

QES:4 IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

```sql
-- Identify the most common pizza size ordered.

SELECT
    pizzas.size,
    COUNT(orders_details.order_details_id) AS order_count
FROM
    pizzas
        JOIN
    orders_details ON pizzas.pizza_id = orders_details.pizza_id
GROUP BY pizzas.size
ORDER BY order_count DESC;

```

OUTPUT :

	size	order_count
▶	L	18526
	M	15385
	S	14137
	XL	544
	XXL	28

QUES:5 LIST THE TOP 5 MOST ORDERED PIZZA TYPES ALONG WITH THEIR QUANTITIES.

```sql
SELECT
    pizza_types.name, SUM(orders_details.quantity) AS quantity
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 quantity DESC
LIMIT 5;
```

OUTPUT:

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

QES:6 JOIN THE RELEVANT TABLES TO CALCULATE THE TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED.

★★★★★

```sql
SELECT
    pizza_types.category,
    SUM(orders_details.quantity) AS quantity
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
ORDER BY quantity DESC;
```

category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

MOZAREL

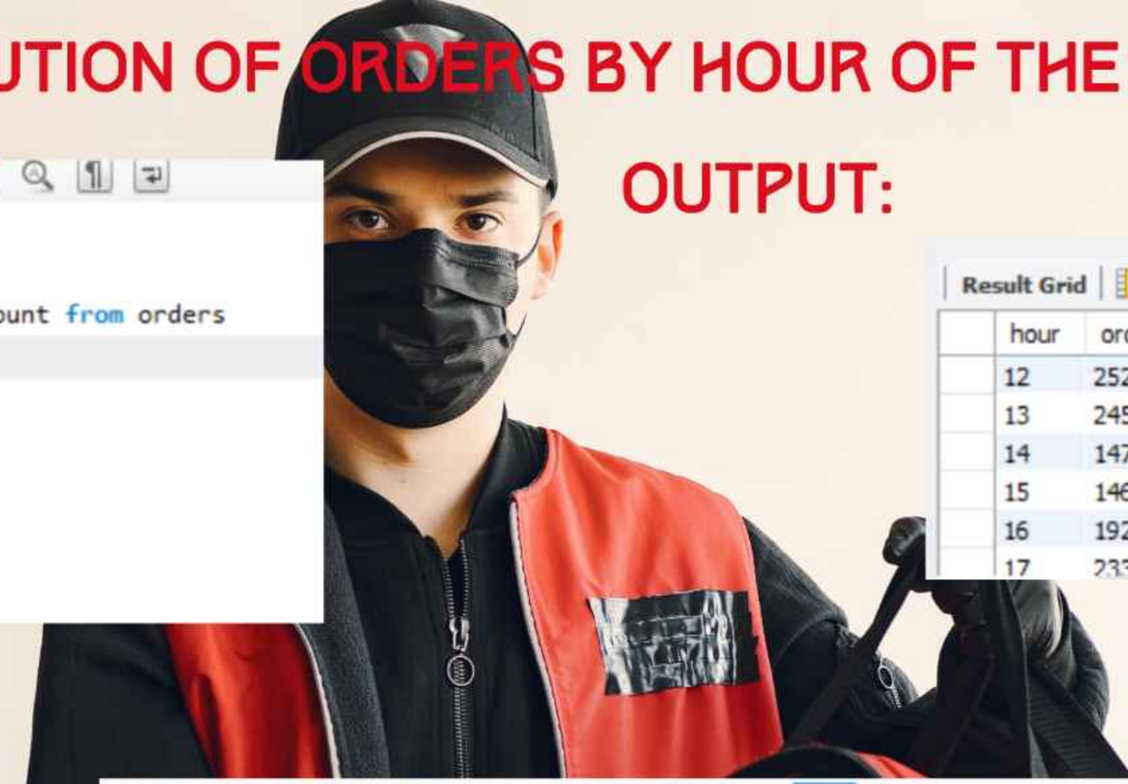
QES:7 DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

```sql
-- 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) ;
```

OUTPUT:

hour	order_count
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336



OUTPUT:

category	COUNT(name)
Chicken	6
Classic	8
Supreme	9
Veggie	9

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

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

OUTPUT:

Result Grid | Filter Rows:

avg_pizza_ordered_per_day
138

QES:9 GROUP THE ORDERS BY DATE AND CALCULATE THE AVERAGE NUMBER OF PIZZAS ORDERED PER DAY.

```sql
SELECT
    ROUND(AVG(quantity), 0) as avg_pizza_ordered_per_day
FROM
    (SELECT
        orders.order_date, SUM(orders_details.quantity) AS quantity
    FROM
        orders
    JOIN orders_details ON orders.order_id = orders_details.order_id
    GROUP BY orders.order_date) AS order_quantity;
```

OUTPUT:

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

QES:10 DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE.

```sql
SELECT
    pizza_types.name,
    SUM(orders_details.quantity * pizzas.price) AS revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.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;
```

QES:11 CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE.

```sql
SELECT
    pizza_types.category,
    (SUM(orders_details.quantity * pizzas.price) / (SELECT
    ROUND(SUM(orders_details.quantity * pizzas.price),
            2) AS total_sales
FROM
    orders_details
        JOIN
    pizzas ON pizzas.pizza_id = orders_details.pizza_id))*100  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
ORDER BY revenue DESC
LIMIT 4;
```

category	revenue
Classic	26.90596025566967
Supreme	25.45631126009862
Chicken	23.955137556847287
Veggie	23.682590927384577

QES:12 DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE FOR EACH PIZZA CATEGORY.

```sql
select 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;
```

OUTPUT:

Result Grid | Filter Rows:

name	revenue
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5
The Classic Deluxe Pizza	38180.5
The Hawaiian Pizza	32273.25
The Pepperoni Pizza	30161.75
The Spicy Italian Pizza	34831.25
The Italian Supreme Pizza	33476.75
The Sicilian Pizza	30940.5

✅ CONCLUSION

🔍 Key Takeaways

- Successfully analyzed pizza sales data using SQL across four tables: orders, order_details, pizzas, and pizza_types.
- Extracted meaningful insights such as:
  - Total revenue and top-performing pizzas
  - Most popular pizza sizes and categories
  - Peak sales hours and daily order trends

🧠 Skills Demonstrated

- SQL querying, joins, grouping, aggregation, and time-based analysis
- Real-world data interpretation for business strategy
- Clear visualization and structured reporting

📈 Business Value

- Helped identify customer preferences and top-selling products
- Supported decisions on menu planning and promotions
- Enabled data-driven improvements for pizza sales strategy

🙏 THANK YOU!
PRESENTED BY: PRINCE
PIZZA SALES DATA ANALYSIS USING SQL
JUNE 2025