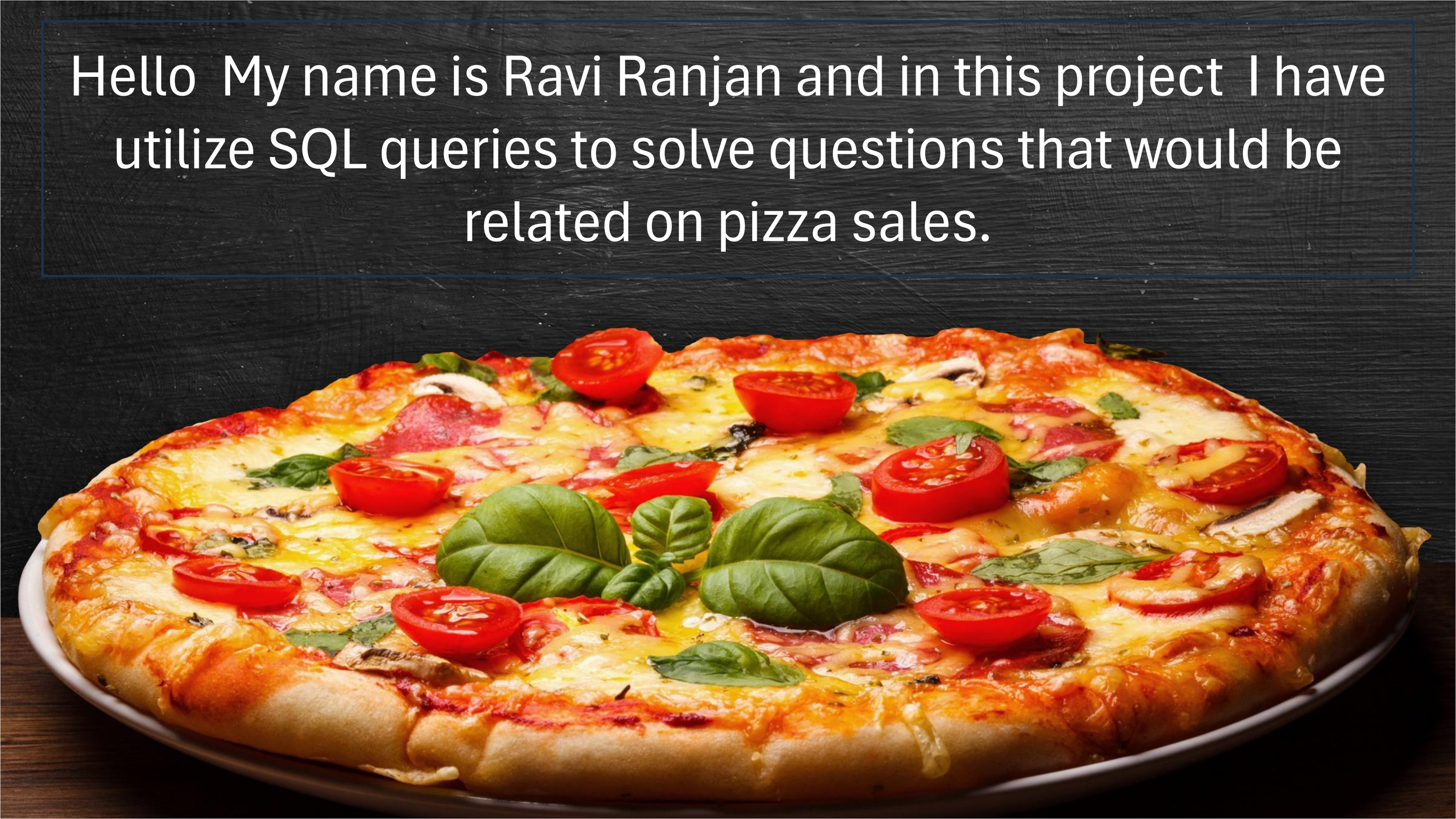
Hello My name is Ravi Ranjan and in this project I have utilize SQL queries to solve questions that would be related on pizza sales.



HERE WE DEFINE THE SCHEMAS BREIFLY.

Data base

In the sequence

Table names

order_details
orders
pizzas
pizza_typess

```
+-----------------+------+------+-----+---------+-------+
| Field           | Type | Null | Key | Default | Extra |
+-----------------+------+------+-----+---------+-------+
| order_details_id | int  | NO   | PRI | NULL    |       |
| order_id        | int  | NO   |     | NULL    |       |
| pizza_id        | text | NO   |     | NULL    |       |
| quantity        | int  | NO   |     | NULL    |       |
+-----------------+------+------+-----+---------+-------+
```

```
+------------+------+------+-----+---------+-------+
| Field      | Type | Null | Key | Default | Extra |
+------------+------+------+-----+---------+-------+
| order_id   | int  | NO   | PRI | NULL    |       |
| order_date | date | NO   |     | NULL    |       |
| order_time | time | NO   |     | NULL    |       |
+------------+------+------+-----+---------+-------+
```

```
+---------------+--------+------+-----+---------+-------+
| Field         | Type   | Null | Key | Default | Extra |
+---------------+--------+------+-----+---------+-------+
| pizza_id      | text   | YES  |     | NULL    |       |
| pizza_type_id | text   | YES  |     | NULL    |       |
| size          | text   | YES  |     | NULL    |       |
| price         | double | YES  |     | NULL    |       |
+---------------+--------+------+-----+---------+-------+
```

```
+---------------+------+------+-----+---------+-------+
| Field         | Type | Null | Key | Default | Extra |
+---------------+------+------+-----+---------+-------+
| pizza_type_id | text | YES  |     | NULL    |       |
| name          | text | YES  |     | NULL    |       |
| category      | text | YES  |     | NULL    |       |
| ingredients   | text | YES  |     | NULL    |       |
+---------------+------+------+-----+---------+-------+
```

Retrive the total number of order placed.

```
Select count(Order_id) as total_orders from order_details;
```

Result Grid

	total_orders
▶	2750

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);
```

Result Grid | Filter R

	hour	order_count
▶	11	93
	12	201
	13	194
	14	155
	15	127

Result 1

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

```sql
SELECT
    category,
    name,
    revenue,
    RANK() OVER (PARTITION BY category ORDER BY revenue DESC) AS rnn
FROM (
    SELECT
        pizza_typess.category,
        pizza_typess.name,
        SUM(order_details.quantity * pizzas.price) AS revenue
    FROM
        pizza_typess
    JOIN
        pizzas ON pizza_typess.pizza_type_id = pizzas.pizza_type_id
    JOIN
        order_details ON order_details.pizza_id = pizzas.pizza_id
    GROUP BY
        pizza_typess.category, pizza_typess.name
) AS a;
```

Result Grid | Filter Rows: | Export:

category	name	revenue	rnn
Chicken	The Barbecue Chicken Pizza	1961	1
Chicken	The Thai Chicken Pizza	1857.25	2
Chicken	The California Chicken Pizza	1778	3
Chicken	The Southwest Chicken Pizza	1557.25	4
Chicken	The Chicken Alfredo Pizza	757	5
Chicken	The Chicken Pesto Pizza	718.75	6
Classic	The Pepperoni Pizza	1673.75	1
Classic	The Classic Deluxe Pizza	1548	2

analyze the cumulative revenue generated over time.

```sql
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	2592.8
2015-01-02	4527.5
2015-01-03	5614.7
2015-01-04	6241.45
2015-01-05	7169.45
2015-01-06	8114.75
2015-01-07	8942.75
2015-01-08	10161.6

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

```sql
select pizza_typess.category,
round(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)*100,2)as revenue
from pizza_typess join pizzas
on pizza_typess.pizza_type_id=pizzas.pizza_type_id
join order_details
on order_details.pizza_id=pizzas.pizza_id
group by pizza_typess.category order by revenue desc;
```

Result Grid | Filter Rows:

category	revenue
Classic	26.18
Supreme	25.83
Veggie	25.48
Chicken	22.52

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

```sql
SELECT
    pizza_typess.name,
    SUM(order_details.quantity * pizzas.price) AS revenue
FROM
    pizza_typess
JOIN
    pizzas ON pizzas.pizza_type_id = pizza_typess.pizza_type_id
JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY
    pizza_typess.name
ORDER BY
    revenue DESC
LIMIT 3;
```

Result Grid | Filter Rows:

name	revenue
The Barbecue Chicken Pizza	1961
The Thai Chicken Pizza	1857.25
The California Chicken Pizza	1778

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

```sql
SELECT
    pizza_typess.category,
    SUM(order_details.quantity) AS quantity
FROM
    pizza_typess
JOIN
    pizzas ON pizza_typess.pizza_type_id = pizzas.pizza_type_id
JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY
    pizza_typess.category
ORDER BY
    quantity DESC;
```

Result Grid | Filter Rows:

category	quantity
Classic	678
Veggie	582
Supreme	573
Chicken	487

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

```sql
SELECT
    pizza_typess.name, SUM(order_details.quantity) AS quantity
FROM
    pizza_typess
        JOIN
    pizzas ON pizza_typess.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_typess.name
ORDER BY quantity DESC
LIMIT 5;
```

Result Grid | Filter Rows:

name	quantity
The Pepperoni Pizza	135
The Hawaiian Pizza	110
The Barbecue Chicken Pizza	108
The California Chicken Pizza	104
The Thai Chicken Pizza	103

generated by pizza sales:

```sql
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

	total_sales
▶	38320.55

Identify the highest-priced pizza. Corrected table name:

```sql
SELECT
    pizza_typess.name, pizzas.price
FROM
    pizza_typess
        JOIN
    pizzas ON pizza_typess.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

Identify the most common pizza size ordered.

```sql
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	888

This project involved analyzing a pizza sales dataset using SQL to derive actionable business insights. I performed data cleaning, joined multiple tables
*(**pizza_types**, **pizzas**, **order_details**, and **orders**), and wrote complex SQL queries to uncover key patterns and trends.*

THANK YOU!