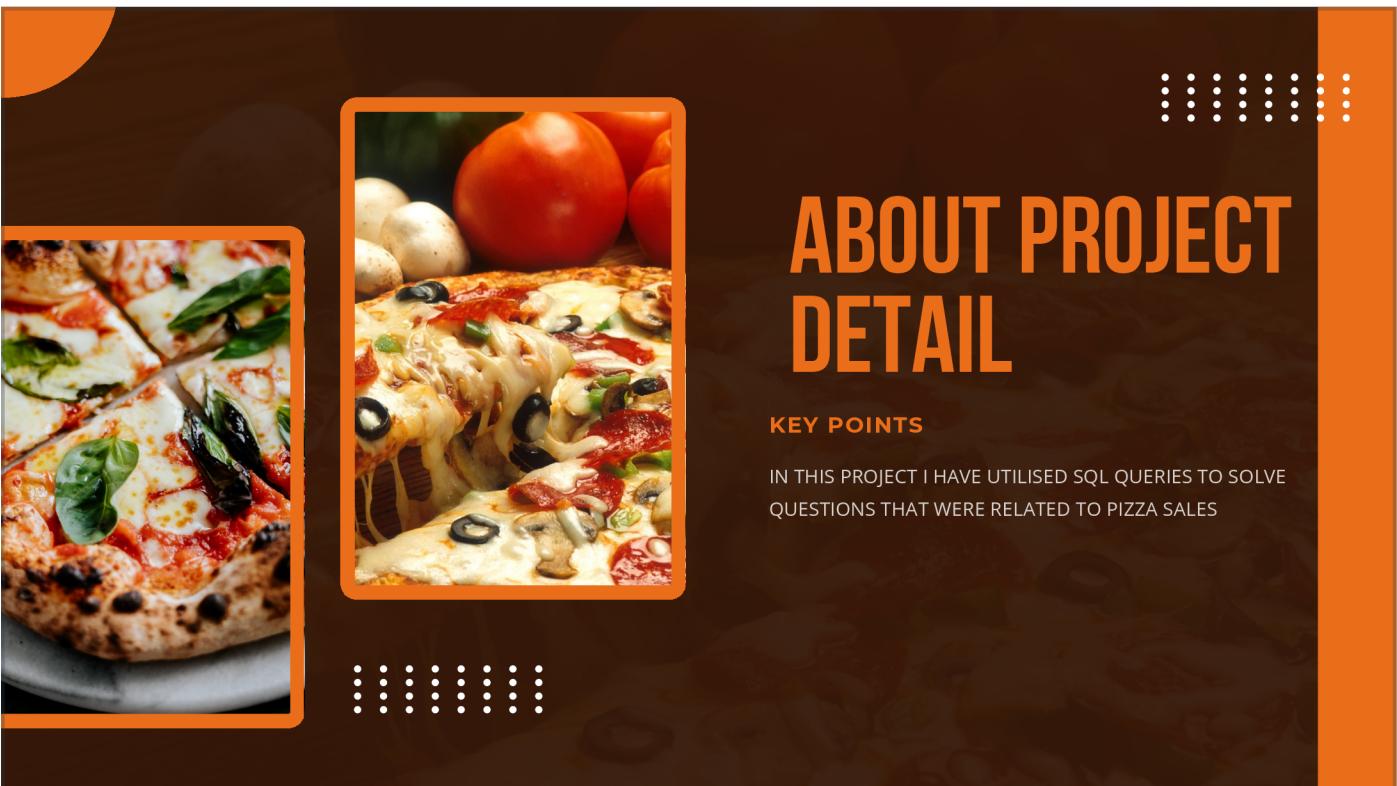


HELLO WELCOME TO MY NEW PROJECT

SLIDE FOR MORE
INFO





ABOUT PROJECT DETAIL

KEY POINTS

IN THIS PROJECT I HAVE UTILISED SQL QUERIES TO SOLVE QUESTIONS THAT WERE RELATED TO PIZZA SALES





- **Basic Questions:**
- **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.**



IN THIS PROJECT I BRIEFLY SOLVED SQL QUERIES AND ALL QUESTIONS FROM BASIC TO ADVANCE LEVEL

Retrieve the total number of orders placed.

```
1  -- --Q1 --retrieve total number of orders placed .
2 •  SELECT
3      *
4  FROM
5      orders;
6 •  SELECT
7      COUNT(order_id) AS total_orders
8  FROM
9      orders;
```

OUT_PUT

Basic questions

Result Grid		Filter Rows:
	total_orders	
▶	5220	



Calculate the total revenue generated from pizza sales.

```
-- -Q2-- - 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;
```

OUT_PUT

Result Grid		Filter Rows:
	total_sales	
▶	38361.3	



Identify the highest-priced pizza.

```
-- --Q3 --identify the higest priced pizza --
SELECT
    pizza_types.name, pizzas.price
FROM
    pizza_types
    JOIN [REDACTED]
        pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
ORDER BY pizzas.price DESC
LIMIT 1;
```

OUT_PUT

Result Grid		
	name	price
▶	The Greek Pizza	35.95

Identify the most common pizza size ordered ?

```
28 ---Q4 ---identity most common pizza sized ordered ---
29 •  SELECT
30     pizzas.size,
31     COUNT(order_details.order_details_id) AS order_count
32   FROM
33     pizzas
34   JOIN
35     order_details ON pizzas.pizza_id = order_details.pizza_id
36   GROUP BY pizzas.size
37   ORDER BY order_count DESC;
```

OUT_PUT

	size	order_count
▶	L	885
	M	729
	S	630
	XL	29

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

```
-- Q5--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 order_details.pizza_id = pizzas.pizza_id  
GROUP BY pizza_types.name  
ORDER BY quantity DESC  
LIMIT 5;
```

OUT_PUT

Result Grid		Filter Rows:
	name	quantity
▶	The Pepperoni Pizza	138
	The Thai Chicken Pizza	111
	The Barbecue Chicken Pizza	111
	The Classic Deluxe Pizza	105

AFTER DISCUSS SOME BASIC LEVEL OF
QUESTIONS NOW WE ARE MOVING
AHED TO INTERMEDIATE QUESTIONS

- Intermediate:
- 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.



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

```
50  --Q6 -----join the nesscerry tables to find the total quantity of each pizza category orders--  
51 • SELECTe the selected portion of the script or everything, if there is no selection ]  
52     pizza_types.category,  
53     SUM(order_details.quantity) AS quantity  
54  FROM  
55     pizza_types  
56     JOIN  
57     pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id  
58     JOIN  
59     order_details ON order_details.pizza_id = pizzas.pizza_id  
60  GROUP BY pizza_types.category  
61  ORDER BY quantity DESC;
```

OUT_PUT

intermediate question

Result Grid		
	category	quantity
▶	Classic	683
	Supreme	578
	Veggie	554
	Chicken	504



Determine the distribution of orders by hour of the day.

```
-- ---Q7-DETERMINE THE DISTRIBUTION OF ORDERS BY HOURS OF THE DAY--  
SELECT  
    HOUR(order_time), COUNT(order_id) AS order_count  
FROM  
    orders  
GROUP BY HOUR(order_time);
```

OUT_PUT

HOUR(order_time)	order_count
11	283
12	631
13	578
14	419
15	346
16	466
17	604
18	576
19	481
20	404
21	278
22	149
23	4
10	1

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

```
-- ----Q8 JOIN RELEVANT TABLES TO FIND THE CATEGORY -WISE DISTRIBUTION OF PIZZAS ---  
• select category ,count(name) from pizza_types group by category ;
```

OUT_PUT

	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

```
-- ---Q9---GROUP THE ORDERS BY DATES AND CALCULATE THE AVERAGE NUMBER OF PIZZAS ORDERS 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;
```

OUT_PUT

	avg_pizza_order_perday
▶	136

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

```
1 -- ----Q10----DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE---
2 •     SELECT
3         pizza_types.name,
4             SUM(order_details.quantity * pizzas.price) AS revenue
5     FROM
6         pizza_types
7             JOIN
8                 pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
9             JOIN
10                order_details ON order_details.pizza_id = pizzas.pizza_id
11        GROUP BY pizza_types.name
12        ORDER BY revenue DESC
13        LIMIT 3;
```

OUT_PUT

Result Grid		Filter Rows:
	name	revenue
▶	The Thai Chicken Pizza	2063.25
	The Barbecue Chicken Pizza	1975.25
	The Italian Supreme Pizza	1775

Here we are discussing some dynamic sql query

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.



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

```
-- -----Q11CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPES TO TOTAL REVENUE -----
SELECT
    pizza_types.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_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;
```

OUT_PUT

	category	revenue
▶	Classic	26.24
	Sup Supreme	98
	Veggie	24.35
	Chicken	23.44

advance

Analyze the cumulative revenue generated over time.

```
-- Q12--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 ;
```

OUT_PUT

order_date	cum_revenue
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
2015-01-08	19399.05
2015-01-09	21526.4
2015-01-10	23990.350000000002
2015-01-11	25862.65
2015-01-12	27781.7
2015-01-13	29831.300000000003
2015-01-14	32358.700000000004
2015-01-15	34343.50000000001
2015-01-16	36937.65000000001
2015-01-17	39341.30000000001

Result 14 X

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

```
-- --Q13 ----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 desc ) as rn
from
(select pizza_types.category ,pizza_types.name,sum((order_details.quantity) *pizzas.price) 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 a )as b
where rn <=3 ;
```

OUT_PUT

name	revenue
The Thai Chicken Pizza	2063.25
The Barbecue Chicken Pizza	1975.25
The California Chicken Pizza	1756.5
The Pepperoni Pizza	1747
The Classic Deluxe Pizza	1651
The Greek Pizza	1423
The Italian Supreme Pizza	1775
The Spicy Italian Pizza	1737.5
The Sicilian Pizza	1597.75
The Five Cheese Pizza	1424.5
The Four Cheese Pizza	1361.1...
The Mexicana Pizza	1270.5

PIZZA SALE PRESENTATION

THANK YOU FOR ATTENTION

See You Next

[WWW.GITHUB.COM/IGANESHNAYAK](https://www.github.com/IGANESHNAYAK)

[WWW.LINKEDIN.COM/IGANESHNAYAK](https://www.linkedin.com/IGANESHNAYAK)

[WWW.INSTAGRAM.COM/IGANESHNAYAK](https://www.instagram.com/IGANESHNAYAK)

[WWW.X.COM/IGANESHNAYAK](https://www.x.com/IGANESHNAYAK)

