# SQL PROJECT ON PIZZA SALE

# INTRODUCTION

Hello, I am Abhilasha Singh, a Data Analyst with a background in Computer Science (BE). In this SQL project, I will analyze pizza sales data to uncover trends, insights, and key metrics that drive business decisions.

# IDENTITFY HIGHEST PRICE PIZZA

```sql
-- Identitfy highest price pizza
SELECT
    pt.name AS pizza_name, p.price
FROM
    pizza_types AS pt
        JOIN
    pizzas AS p ON p.pizza_type_id = pt.pizza_type_id
ORDER BY p.price DESC
LIMIT 1;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

| pizza_name | price |
|---|---|
| The Greek Pizza | 35.95 |

# CALCULATE THE TOTAL REVENUE GENERATED BY PIZZA SALES

```sql
SELECT
    ROUND(SUM(piz.price * d.quantity), 2) AS total_revenue
FROM
    pizzas AS piz
        JOIN
    orders_detail AS d ON piz.pizza_id = d.pizza_id;
```

Limit to 10000 rows

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

| total_revenue |
| --- |
| 817860.05 |

# IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED

```sql
SELECT
    p.size, COUNT(order_detail_id) AS Ordered_by_people, sum(quantity) as total_orders
FROM
    pizzas AS p
        JOIN
    orders_detail AS o ON p.pizza_id = o.pizza_id
GROUP BY p.size
ORDER BY Ordered_by_people DESC;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

| size | Ordered_by_people | total_orders |
|---|---|---|
| L | 18526 | 18956 |
| M | 15385 | 15635 |
| S | 14137 | 14403 |
| XL | 544 | 552 |
| XXL | 28 | 28 |

Result 1

# LIST TOP 5 MOST ORDERED PIZZA TYPE ALONG WITH THEIR QUANTITY

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

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

| name | total_quantity |
|---|---|
| The Classic Deluxe Pizza | 2453 |
| The Barbecue Chicken Pizza | 2432 |
| The Hawaiian Pizza | 2422 |
| The Pepperoni Pizza | 2418 |

# JOIN THE NECESSARY TABLE TO FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY

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

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

| category | total_quantity |
|---|---|
| Classic | 14888 |
| Supreme | 11987 |
| Veggie | 11649 |
| Chicken | 11050 |

# DETERMINE THE DISTRIBUTION OF THE ORDER BY HOUR OF THE DAY

```sql
SELECT
    HOUR(order_time) AS hours, COUNT(order_id) AS total_order
FROM
    orders
GROUP BY hours;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

| hours | total_order |
|---|---|
| 11 | 1231 |
| 12 | 2520 |
| 13 | 2455 |
| 14 | 1472 |
| 15 | 1468 |
| 16 | 1920 |
| 17 | 2336 |

# JOIN THE RELEVANT TABLE TO FIND THE PIZZA CATEGORY WISE DISTRIBUTION

```sql
SELECT
    category, COUNT(name) AS total_pizza
FROM
    pizza_types
GROUP BY category
ORDER BY total_pizza DESC;
```

Limit to 10000 rows

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

| category | total_pizza |
|---|---|
| Supreme | 9 |
| Veggie | 9 |
| Classic | 8 |
| Chicken | 6 |

# RETRIVE TOTAL NUMBER OF ORDER PLACED

```
1  SELECT
2      COUNT(order_id) AS total_orders
3  FROM
4      orders;
5
```

Result Grid | Filter Rows: | Export: | Wrap

| total_orders |
| --- |
| 21350 |

# GROUP THE ORDERS BY DATE AND CALCULATE THE AVERAGE NUMBER OF PIZZA ORDERED PER DAY

```sql
SELECT
    ROUND(AVG(number_of_pizza), 2) AS avg_pizza_ordered
FROM
    (SELECT
        orders.order_date,
            SUM(orders_detail.quantity) AS number_of_pizza
    FROM
        orders
    INNER JOIN orders_detail ON orders.order_id = orders_detail.order_id
    GROUP BY orders.order_date) AS order_quantity;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

| avg_pizza_ordered |
| --- |
| 138.47 |

# DETERMINE THE TOP THREE MOST ORDERED PIZZAS TYPE BASE ON THE REVENUE

```sql
SELECT
    pizza_types.name,
    SUM(pizzas.price * orders_detail.quantity) AS total_revenue
FROM
    pizzas
        INNER JOIN
    pizza_types ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        INNER JOIN
    orders_detail ON pizzas.pizza_id = orders_detail.pizza_id
GROUP BY pizza_types.name
ORDER BY total_revenue DESC
LIMIT 3;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

| name | total_revenue |
|---|---|
| The Thai Chicken Pizza | 43434.25 |
| The Barbecue Chicken Pizza | 42768 |
| The California Chicken Pizza | 41409.5 |

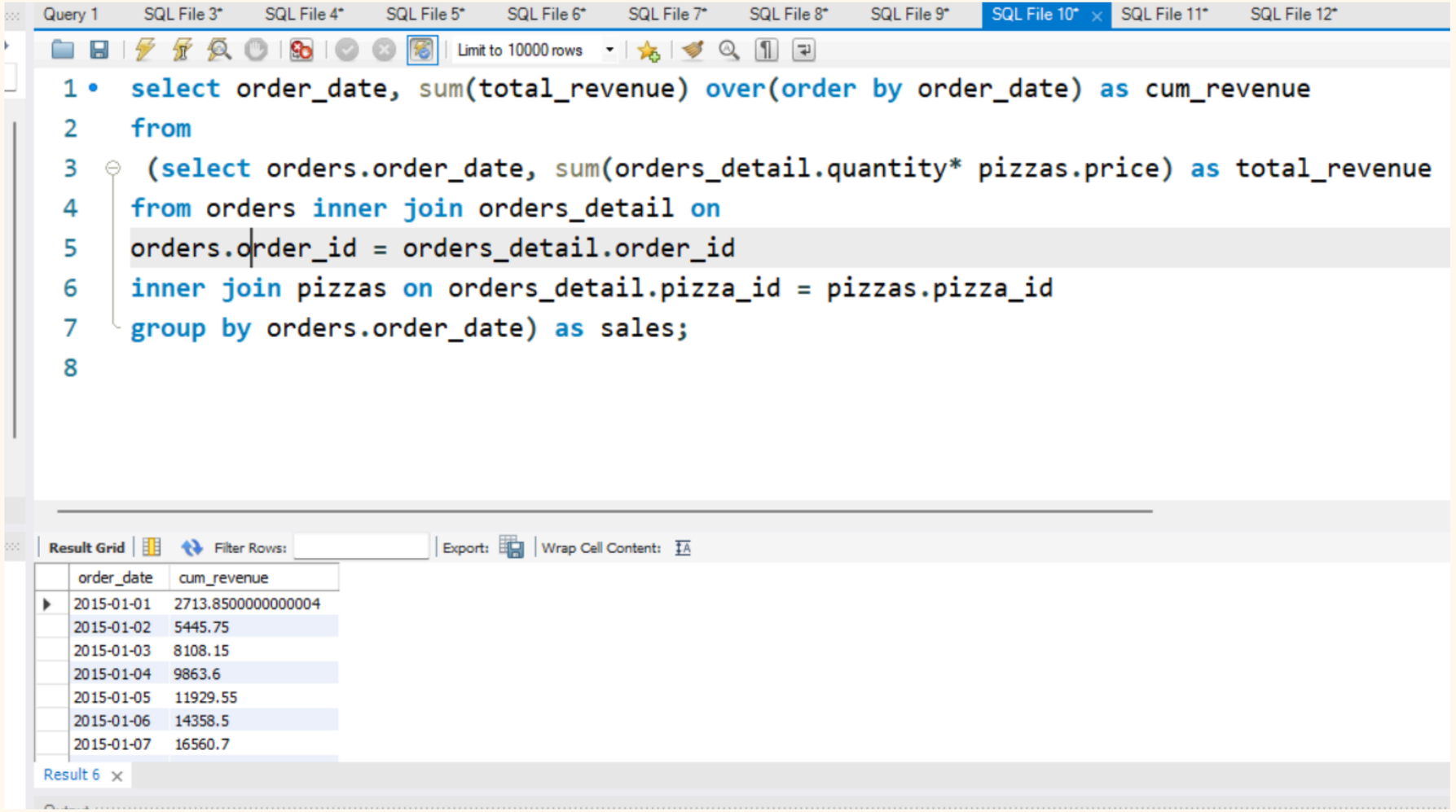
# CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE

```sql
SELECT
    pizza_types.category,
    ROUND(SUM(orders_detail.quantity * pizzas.price) / (SELECT
                    SUM(orders_detail.quantity * pizzas.price)
                FROM
                    orders_detail
                        INNER JOIN
                    pizzas ON pizzas.pizza_id = orders_detail.pizza_id) * 100,
            2) AS total_percent
FROM
    pizza_types
        INNER JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        INNER JOIN
    orders_detail ON orders_detail.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY total_percent DESC;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

| category | total_percent |
|---|---|
| Classic | 26.91 |
| Supreme | 25.46 |
| Chicken | 23.96 |
| Veggie | 23.68 |

# ANALYZE THE CUMMULATIVE REVENUE GENERATED OVER TIME



# DETERMINE THE TOP THREE MOST ORDERED PIZZAS TYPE BASED ON REVNEUE FOR EACH PIZZA_TYPE

Query 1 | SQL File 3* | SQL File 4* | SQL File 5* | SQL File 6* | SQL File 7* | SQL File 8* | SQL File 9* | SQL File 10* | SQL File 11* | SQL File 12*

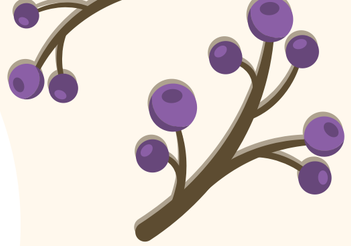
Limit to 10000 rows

```sql
select name, total_revenue from
(select category, name, total_revenue, rank() over( partition by category order by total_revenue desc) as rn
from
(select pizza_types.category, pizza_types.name, sum(pizzas.price * orders_detail.quantity) as total_revenue
from pizza_types inner join pizzas on pizza_types.pizza_type_id = pizzas.pizza_type_id
inner join orders_detail on pizzas.pizza_id = orders_detail.pizza_id
group by pizza_types.category, pizza_types.name) as sales) as b
where rn >=3;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

| name | total_revenue |
|---|---|
| The California Chicken Pizza | 41409.5 |
| The Southwest Chicken Pizza | 34705.75 |
| The Chicken Alfredo Pizza | 16900.25 |
| The Chicken Pesto Pizza | 16701.75 |
| The Pepperoni Pizza | 30161.75 |
| The Greek Pizza | 28454.100000000013 |
| The Italian Capocollo Pizza | 25094 |

Result 7



# THANK YOU FOR YOUR ATTENTION