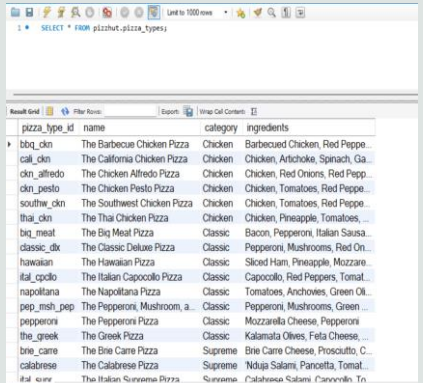
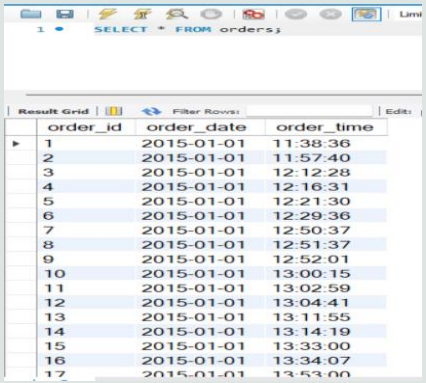
# Pizza Sale Data Analysis Project

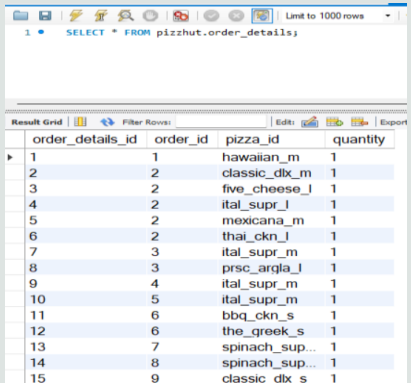
Name- Akash Gunjal

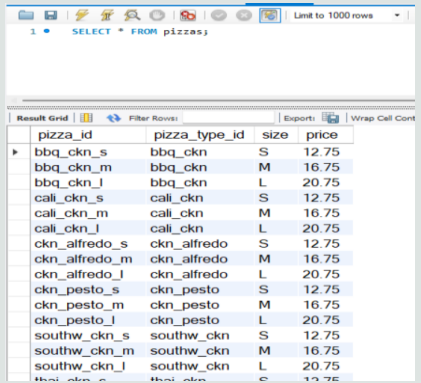
# Introduction

- This project is to design the DB Tables, load the data into DB and to solve the questions related data analysis of pizza sale using sql queries.
- Also visualize in power BI where It help to understand the insights of the data very clearly.



pizzhut orders

order_date

order_id

order_time

Collapse

pizzhut pizzas

pizza_id

pizza_type_id

price

size

Collapse

pizzhut order_details

order_details_id

order_id

pizza_id

quantity

Collapse

pizzhut pizza_types

category

ingredients

name

pizza_type_id

Collapse

1) Retrieve the total number of orders placed.

```
-- 1) Retrieve the total number of orders placed.

select count(order_id)as total_orders from orders;
```

| total_orders |
| --- |
| 21350 |

2)Calculate the total revenue generated from pizza sales.

```
-- 2)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
```

| total_sales |
| --- |
| 817860.05 |

3)Identify the highest-priced pizza.

```
-- 3)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;
```

| name | price |
|---|---|
| The Greek Pizza | 35.95 |

4)Identify the most common pizza size ordered.

```
-- 4)Identify the most common pizza size ordered.

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;
```

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

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

```sql
-- 5)List the top 5 most ordered pizza types along with their
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;
```

| 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 |

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

```sql
-- 6)Join the necessary tables to find the total quantity
--    of each pizza category ordered.
SELECT
    pizza_types.category,
    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.category
ORDER BY quantity DESC;
```

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

7)Determine the distribution of orders by hour of the day.

```
1   -- 7)Determine the distribution of orders by hour of the day
2
3   SELECT
4       HOUR(order_time), COUNT(order_id)
5   FROM
6       orders
7   GROUP BY HOUR(order_time);
```

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

| 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 |

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

```
1   -- 8)Join relevant tables to find the category-wise distribution of pizzas
2
3   SELECT
4       category, COUNT(name)
5   FROM
6       pizza_types
7   GROUP BY category;
```

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

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

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

```sql
-- 9)Group the orders by date and calculate the average number of pizzas ordered per day

SELECT
    ROUND(AVG(quantity), 0)as avg_pizza_ordered_per_day
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
```

| avg_pizza_ordered_per_day |
|---|
| 138 |

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

```sql
-- 10)Determine the top 3 most ordered pizza types based on revenue.

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

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

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

```sql
-- 11) Calculate the percentage contribution of each pizza type to total revenue.
SELECT
    pizza_types.category,
    ROUND(SUM(order_details.quantity * pizzas.price) / (SELECT
                ROUND(SUM(order_details.quantity * pizzas.price),
                        2) AS sales
            FROM
                order_details
                    JOIN
                pizzas ON order_details.pizza_id = pizzas.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;
```

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

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

12)Analyze the cumulative revenue generated over time.

```sql
-- 12)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
```

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

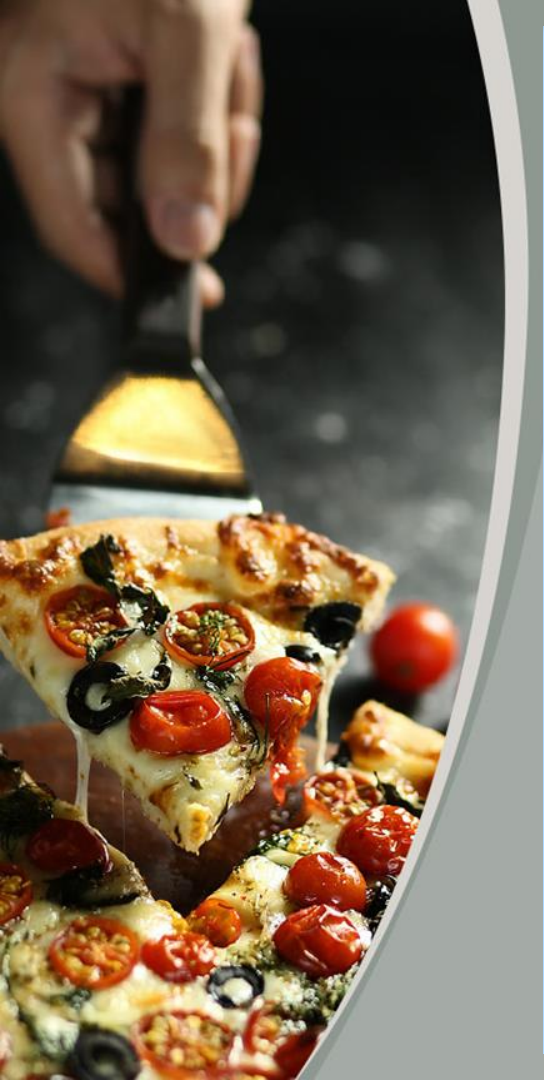
| order_date | cum_revenue |
|---|---|
| 2015-01-01 | 2713.8500000000004 |
| 2015-01-02 | 5445.75 |
| 2015-01-03 | 8108.15 |
| 2015-01-04 | 9863.6 |
| 2015-01-05 | 11929.55 |
| 2015-01-06 | 14358.5 |
| 2015-01-07 | 16560.7 |
| 2015-01-08 | 19399.05 |
| 2015-01-09 | 21526.4 |
| 2015-01-10 | 23990.350000000002 |
| 2015-01-11 | 25862.65 |
| 2015-01-12 | 27781.7 |
| 2015-01-13 | 29831.300000000003 |

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

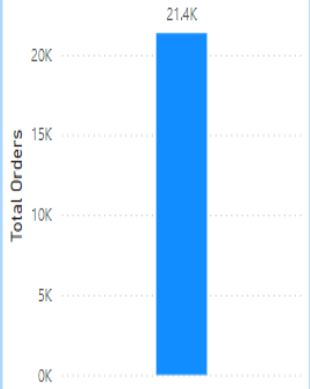
```sql
-- 13)Determine the top 3 most ordered pizza types based on revenue for each pizza category.

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;
```

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

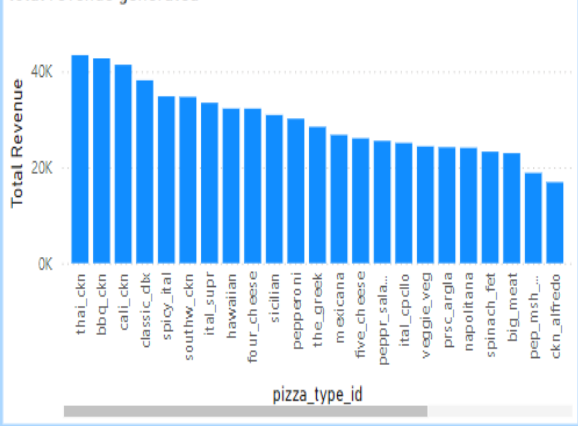
| name | revenue |
|---|---|
| The Thai Chicken Pizza | 43434.25 |
| The Barbecue Chicken Pizza | 42768 |
| The California Chicken Pizza | 41409.5 |
| The Classic Deluxe Pizza | 38180.5 |
| The Hawaiian Pizza | 32273.25 |
| The Pepperoni Pizza | 30161.75 |
| The Spicy Italian Pizza | 34831.25 |
| The Italian Supreme Pizza | 33476.75 |
| The Sicilian Pizza | 30940.5 |
| The Four Cheese Pizza | 32265.70000000065 |



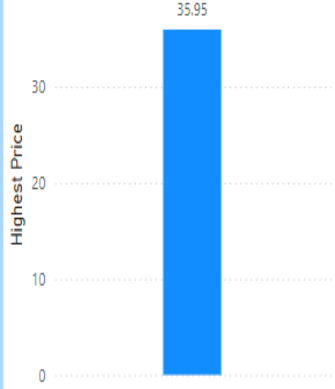
## Total Orders



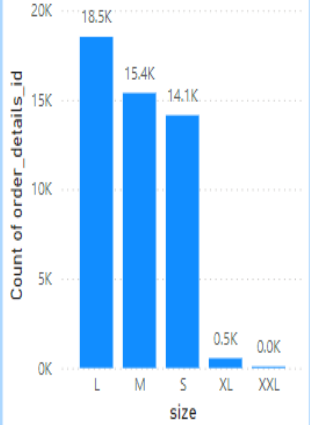

## total revenue generated



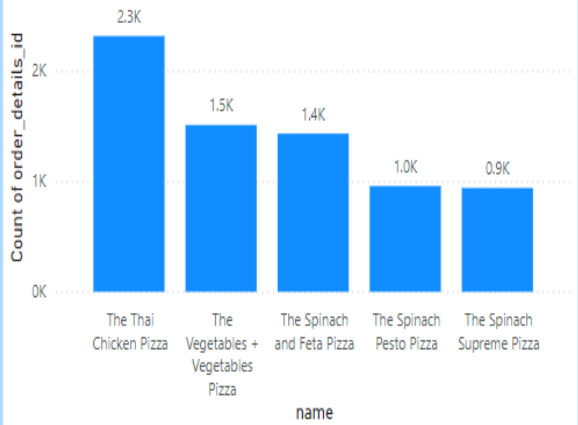

## Highest Price pizza



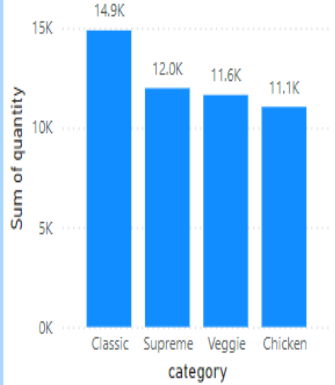

## Most common pizza size ordered



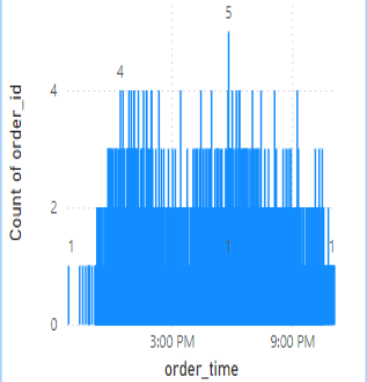

## most ordered pizza types along with their quantities



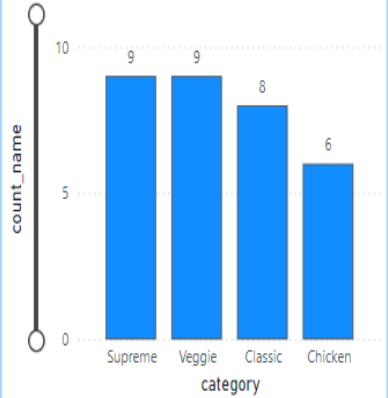

## total quantity of each pizza category ordered



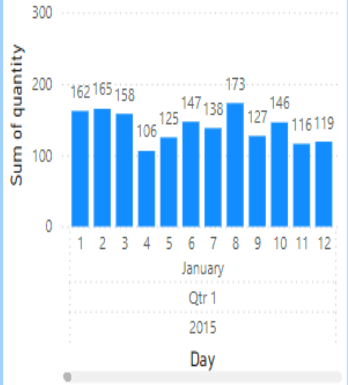
the distribution of orders by hour of the day.



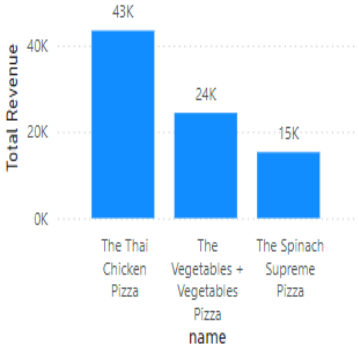

category-wise distribution of pizzas.



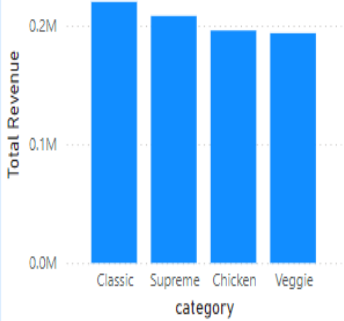

average number of pizzas ordered per day



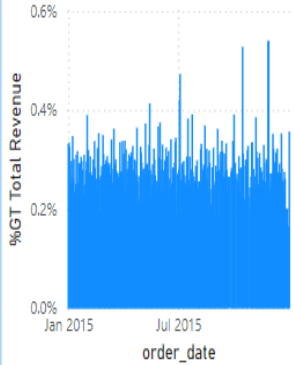

3 most ordered pizza types based on revenue



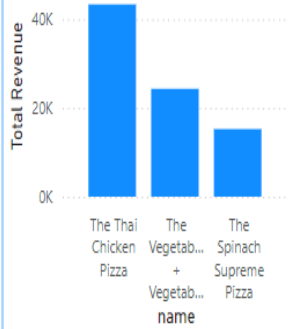

contribution of each pizza type to total revenue




cumulative revenue generated over time




3 most ordered pizza types based on revenue



# Thank You

I Appreciate the opportunity to share this project with you.