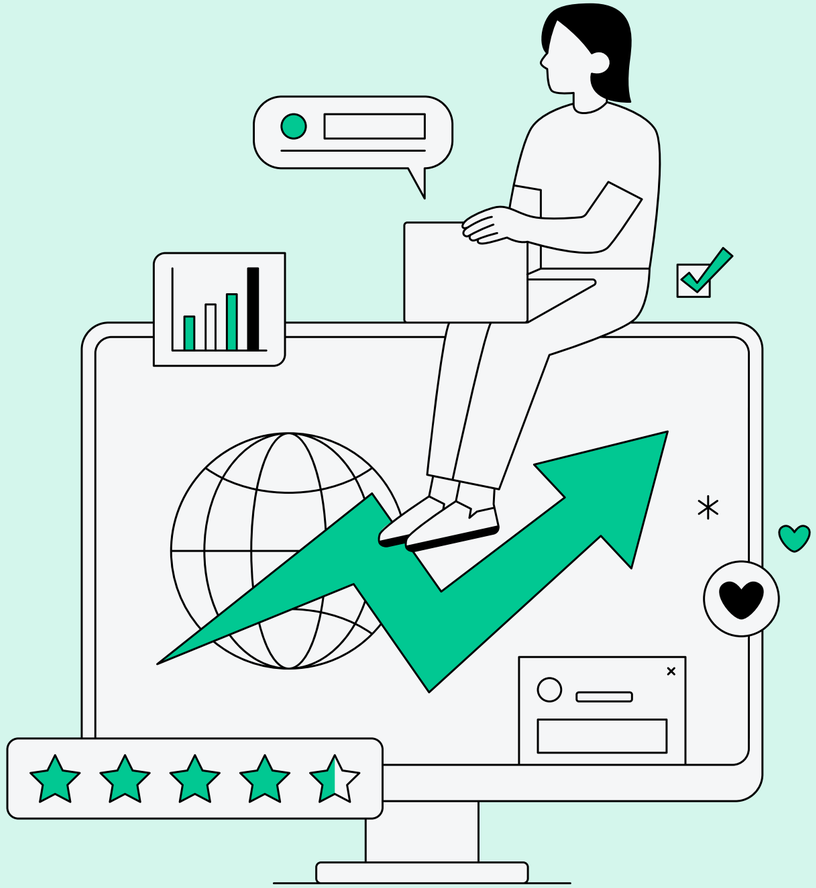
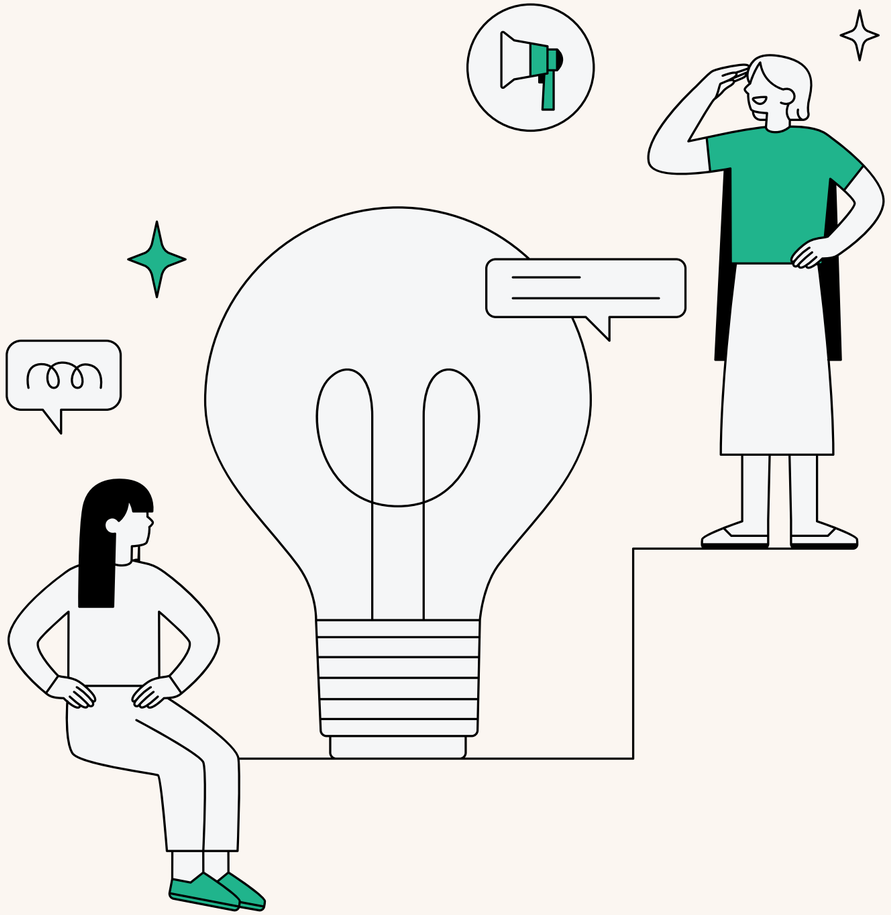
Presented by Ajith K

# PizzaHut Data - Analysis Project Report



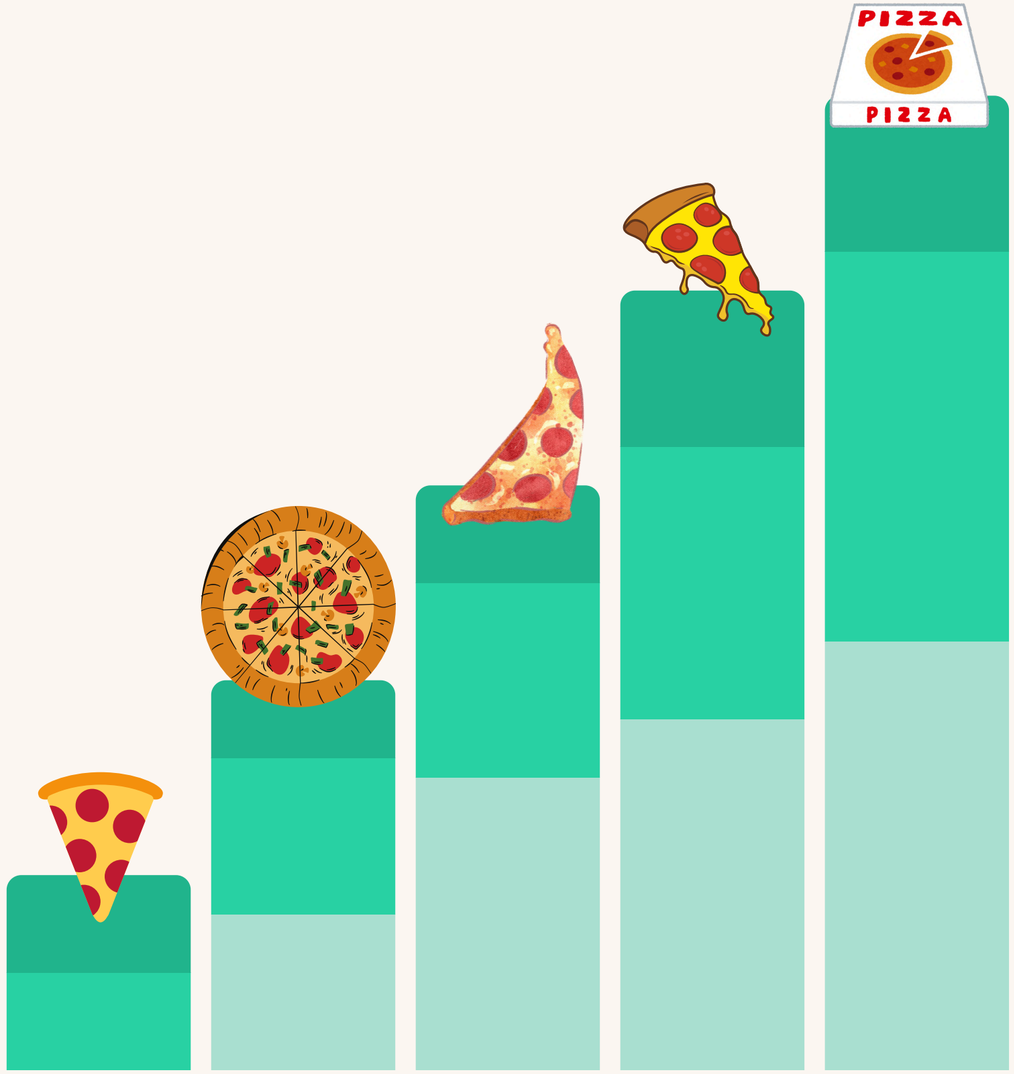
# Introduction to analysis

- This Data-Analysis project was developed by utilizing **PostgreSQL** database and **Structured Query Language** to query the database to generate useful business insights.
- Pizzahut sales data, which was available in CSV format was extracted and loaded to the PostgresSQL database in a structured manner for the analysis by leveraging **pgAdmin4** software.
- A new database named pizzahut was created and tables with proper schema and keys were structured and available sales data was imported using wizard as a part of the project.



# Methodology used in the analysis

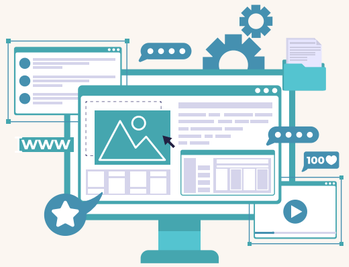
- pgAdmin4 was utilized for creating PostgreSQL database, structuring the available csv data into relational format and for querying the data insights.
- Basic to advanced SQL concepts like complex joins, aggregations, subqueries, window functions were leveraged and utilized for comprehensive analysis.



# Overview of Project Architecture

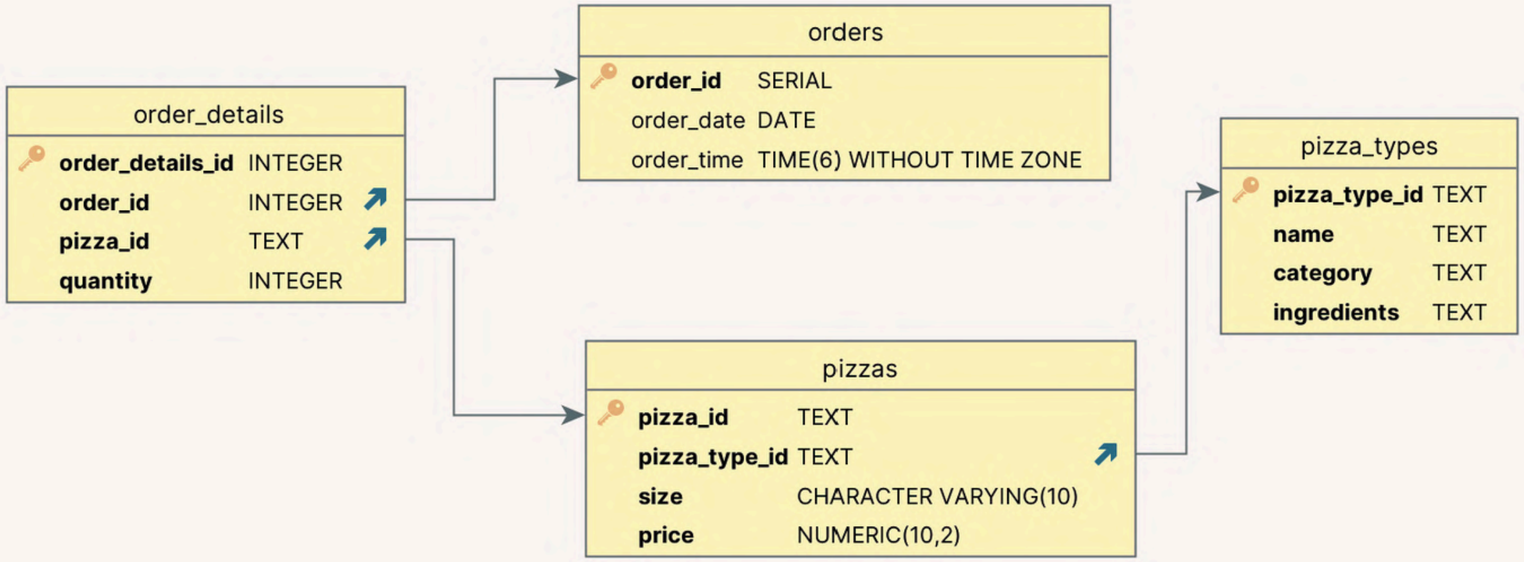

**PizzaHut Dataset**


**PostgreSQL Database**


**Business Insights**

# Schema Visualization of the Database




Database : pizzahut

# Challenges and proposed solutions

## 01.

CSV files directly loaded to Microsoft Excel can only give simple calculations which cant help in business driven decision making.

- Hence leveraged the possibilities of PostgreSQL and its complex analysis possibilities.

## 02.

Data stored in different tables cant bring overall insights together.

- Using complex joins, unions, subqueries and window functions, analysed maximum possibilities of the stored data in different tables

# Analysis and Insights.

The total number of orders placed.

**21,350**

Query | Query History

```
1  select
2      count(*) as total_orders
3  from orders;
```

Data Output | Messages | Notifications

| | total_orders bigint |
|---|---|
| 1 | 21350 |

# Analysis and Insights.

**The total revenue generated from pizza sales.**

**817860.05**

Query | Query History

```
1  select
2      sum(od.quantity * p.price) as total_revenue
3  from pizzas p join order_details od
4  on p.pizza_id = od.pizza_id;
```

Data Output | Messages | Notifications

| | total_revenue numeric |
|---|---|
| 1 | 817860.05 |

# Analysis and Insights.

The highest-priced pizza.

# The Greek Pizza

Query | Query History

```sql
select
    pt.name as highest_priced_pizza
from pizza_types pt
join pizzas p on pt.pizza_type_id = p.pizza_type_id
order by p.price desc
limit 1;
```

Data Output | Messages | Notifications

| | highest_priced_pizza text |
|---|---|
| 1 | The Greek Pizza |

# Analysis and Insights.

**The most common pizza size ordered.**

# Large

Query | Query History

```sql
select
    pizza_size as most_common_pizza_size
from (
      select
         p.size as pizza_size,
         count(p.size) as total_count
      from order_details od
      join pizzas p on od.pizza_id = p.pizza_id
      group by p.size
      order by total_count desc
     )
limit 1;
```

Data Output | Messages | Notifications

| | most_common_pizza_size<br>character varying (10) |
|---|---|
| 1 | L |

# Analysis and Insights.

**The top 5 most ordered pizza types along with their quantities.**

1. The Classic Deluxe Pizza
2. The Barbecue Chicken Pizza
3. The Hawaiian Pizza
4. The Pepperoni Pizza
5. The Thai Chicken Pizza

Query | Query History

```sql
select
    pt.name as pizza_type,
    sum(quantity) as total_quantity
from order_details od
join pizzas p on od.pizza_id = p.pizza_id
join pizza_types pt on p.pizza_type_id = pt.pizza_type_id
group by pt.name
order by total_quantity desc
limit 5;
```

Data Output | Messages | Notifications

| | pizza_type text | total_quantity bigint |
|---|---|---|
| 1 | The Classic Deluxe Pizza | 2453 |
| 2 | The Barbecue Chicken Pizza | 2432 |
| 3 | The Hawaiian Pizza | 2422 |
| 4 | The Pepperoni Pizza | 2418 |
| 5 | The Thai Chicken Pizza | 2371 |

# Analysis and Insights.

**The total quantity of each pizza category ordered.**

**Classic : 14888**
**Supreme : 11987**
**Veggie : 11649**
**Chicken : 11050**

Query | Query History

```sql
select
    pt.category as category,
    sum(quantity) as total_order_quantity
from order_details od
join pizzas p on od.pizza_id = p.pizza_id
join pizza_types pt on p.pizza_type_id = pt.pizza_type_id
group by pt.category
order by total_order_quantity desc;
```

Data Output | Messages | Notifications

| | category text | total_order_quantity bigint |
|---|---|---|
| 1 | Classic | 14888 |
| 2 | Supreme | 11987 |
| 3 | Veggie | 11649 |
| 4 | Chicken | 11050 |

# Analysis and Insights.

**The distribution of orders by hour of the day.**

Query | Query History

```
1  select
2      extract(hour from order_time) as hour,
3      count(order_id) orders_count from orders
4  group by extract(hour from order_time)
5  order by hour;
6
```

Data Output | Messages | Notificat

| | hour numeric | orders_count bigint |
|---|---|---|
| 1 | 9 | 1 |
| 2 | 10 | 8 |
| 3 | 11 | 1231 |
| 4 | 12 | 2520 |
| 5 | 13 | 2455 |
| 6 | 14 | 1472 |
| 7 | 15 | 1468 |
| 8 | 16 | 1920 |
| 9 | 17 | 2336 |
| 10 | 18 | 2399 |
| 11 | 19 | 2009 |
| 12 | 20 | 1642 |
| 13 | 21 | 1198 |
| 14 | 22 | 663 |
| 15 | 23 | 28 |

# Analysis and Insights.

**The category-wise distribution of pizzas ordered.**

Classic : 14,888
Supreme : 11,987
Veggie : 11,649
Chicken : 11,050

Query | Query History

```sql
select
    pt.category as category,
    sum(od.quantity) as total_quantity_ordered
from order_details od
join pizzas p on od.pizza_id = p.pizza_id
join pizza_types pt on p.pizza_type_id = pt.pizza_type_id
group by pt.category
```

Data Output | Messages | Notifications

| | category text | total_quantity_ordered bigint |
|---|---|---|
| 1 | Veggie | 11649 |
| 2 | Chicken | 11050 |
| 3 | Supreme | 11987 |
| 4 | Classic | 14888 |

# Analysis and Insights.

**The average number of pizzas ordered per day.**

**138**

Query | Query History

```sql
select
    round(avg(sum_orders_per_day)) as average_orders_per_day
from (
      select
        o.order_date as day,
        sum(od.quantity) as sum_orders_per_day
      from orders o
      join order_details od on o.order_id = od.order_id
      group by o.order_date
     )
```

Data Output | Messages | Notifications

| | average_orders_per_day numeric |
|---|---|
| 1 | 138 |

# Analysis and Insights.

**The top 3 most ordered pizza types based on revenue.**

1. **The Thai Chicken Pizza**
2. **The Barbecue Chicken Pizza**
3. **The California Chicken Pizza**

Query | Query History

```sql
select
    pt.name as pizza_type,
    sum(od.quantity * p.price) as total_revenue
from order_details od
join pizzas p on od.pizza_id = p.pizza_id
join pizza_types pt on p.pizza_type_id = pt.pizza_type_id
group by pt.name
order by total_revenue desc
limit 3;
```

Data Output | Messages | Notifications

| | pizza_type<br>text | total_revenue<br>numeric |
|---|---|---|
| 1 | The Thai Chicken Pizza | 43434.25 |
| 2 | The Barbecue Chicken Pizza | 42768.00 |
| 3 | The California Chicken Pizza | 41409.50 |

# Analysis and Insights.

**The percentage contribution of each pizza category to total revenue.**

**Veggie : 23.68%**
**Chicken : 23.96%**
**Supreme : 25.46%**
**Classic : 26.91%**

ery Query History

```
select
    pt.category as pizza_type,
    round(sum(od.quantity * p.price)/(
                                        select
                                            sum(od.quantity * p.price)
                                        from order_details od
                                        join pizzas p on od.pizza_id = p.pizza_i
                                       )*100,2) as percentage_of_revenue
from order_details od
join pizzas p on od.pizza_id = p.pizza_id
join pizza_types pt on p.pizza_type_id = pt.pizza_type_id
group by pt.category;
```

Data Output | Messages | Notifications

| | pizza_type text | percentage_of_revenue numeric |
|---|---|---|
| 1 | Veggie | 23.68 |
| 2 | Chicken | 23.96 |
| 3 | Supreme | 25.46 |
| 4 | Classic | 26.91 |

# Analysis and Insights.

**Analyze the cumulative revenue generated over time.**

Query | Query History

```sql
select
    order_date,
    sum(revenue) over(order by order_date) as cumulative_revenue
from (
    select
        o.order_date as order_date,
        sum(p.price * od.quantity) as revenue
    from pizzas p
    join order_details od on p.pizza_id = od.pizza_id
    join orders o on od.order_id = o.order_id
    group by o.order_date
```

Data Output | Messages | Notifications

| | order_date<br>date | cumulative_revenue<br>numeric |
|---|---|---|
| 1 | 2015-01-01 | 2713.85 |
| 2 | 2015-01-02 | 5445.75 |
| 3 | 2015-01-03 | 8108.15 |
| 4 | 2015-01-04 | 9863.60 |
| 5 | 2015-01-05 | 11929.55 |
| 6 | 2015-01-06 | 14358.50 |
| 7 | 2015-01-07 | 16560.70 |
| 8 | 2015-01-08 | 19399.05 |
| 9 | 2015-01-09 | 21526.40 |
| 10 | 2015-01-10 | 23990.35 |
| 11 | 2015-01-11 | 25862.65 |
| 12 | 2015-01-12 | 27781.70 |

Total rows: 358 of 358 | Query complete 00

# Analysis and Insights.

**The top 3 most ordered pizza types based on revenue for each pizza category.**

| | |
|---|---|
| Chicken | The Thai Chicken Pizza |
| | The Barbecue Chicken Pizza |
| | The California Chicken Pizza |
| Classic | The Classic Deluxe Pizza |
| | The Hawaiian Pizza |
| | The Pepperoni Pizza |
| Supreme | The Spicy Italian Pizza |
| | The Italian Supreme Pizza |
| | The Sicilian Pizza |
| Veggie | The Four Cheese Pizza |
| | The Mexicana Pizza |
| | The Five Cheese Pizza |

ery Query History

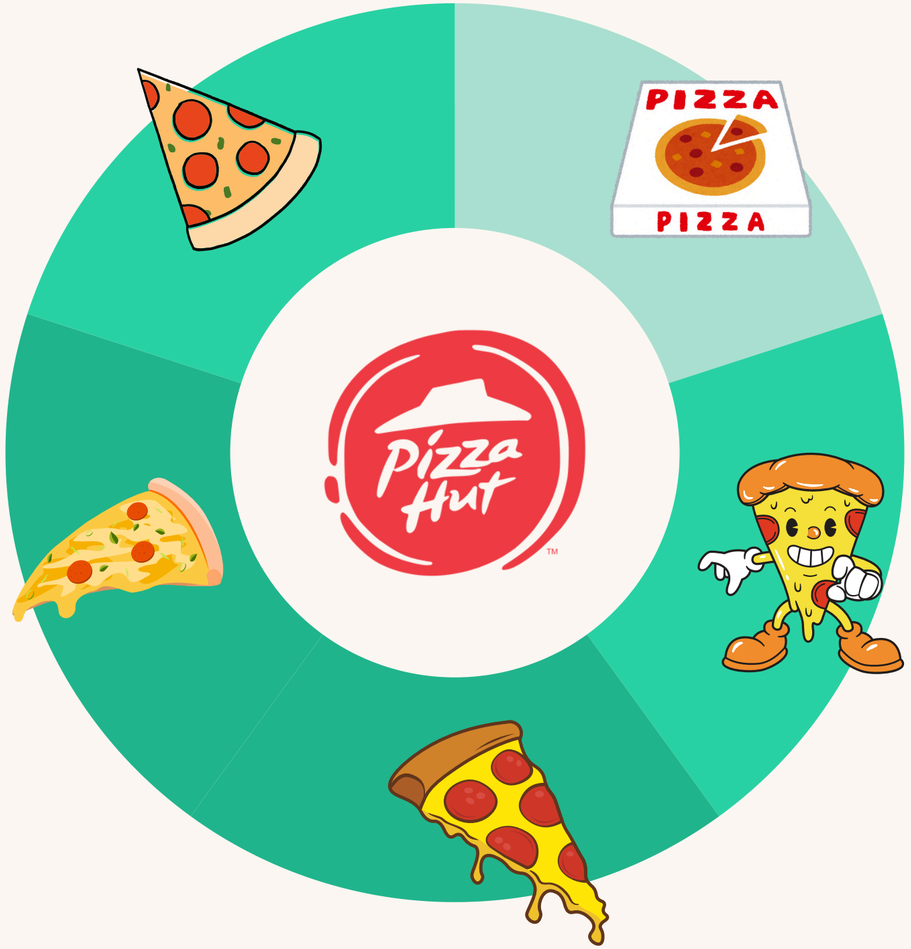
```sql
select
    category,
    name,
    revenue,
    position
from (
      select
          pt.category as category,
          pt.name as name,
          sum(od.quantity * p.price) as revenue,
          rank() over (partition by pt.category order by sum(od.quantity * p.price) desc) as positic
      from order_details od
      join pizzas p on od.pizza_id = p.pizza_id
      join pizza_types pt on p.pizza_type_id = pt.pizza_type_id
      group by pt.category, pt.name
      order by category, revenue desc
     )
where position in (1,2,3);
```

Data Output | Messages | Notifications

| | category text | name text | revenue numeric | position bigint |
|---|---|---|---|---|
| 1 | Chicken | The Thai Chicken Pizza | 43434.25 | 1 |
| 2 | Chicken | The Barbecue Chicken Pizza | 42768.00 | 2 |
| 3 | Chicken | The California Chicken Pizza | 41409.50 | 3 |
| 4 | Classic | The Classic Deluxe Pizza | 38180.50 | 1 |
| 5 | Classic | The Hawaiian Pizza | 32273.25 | 2 |
| 6 | Classic | The Pepperoni Pizza | 30161.75 | 3 |
| 7 | Supreme | The Spicy Italian Pizza | 34831.25 | 1 |
| 8 | Supreme | The Italian Supreme Pizza | 33476.75 | 2 |
| 9 | Supreme | The Sicilian Pizza | 30940.50 | 3 |
| 10 | Veggie | The Four Cheese Pizza | 32265.70 | 1 |
| 11 | Veggie | The Mexicana Pizza | 26780.75 | 2 |
| 12 | Veggie | The Five Cheese Pizza | 26066.50 | 3 |

# Conclusion and key learnings

Main goal of this pizzahut data analysis project was a comprehensive walkthrough of the pizza sales data and finding useful insights from the data that could help business to develop and increase marketcap and future-sales of their commodity.
Leveraging the SQL for the complex data analysis was the main takeaway from this real-world industry level project. The power of SQL in leveraging the data analysis and its application in real world scenario is the key learnings from this project.



Presented by Ajith K

# Thank you very much!