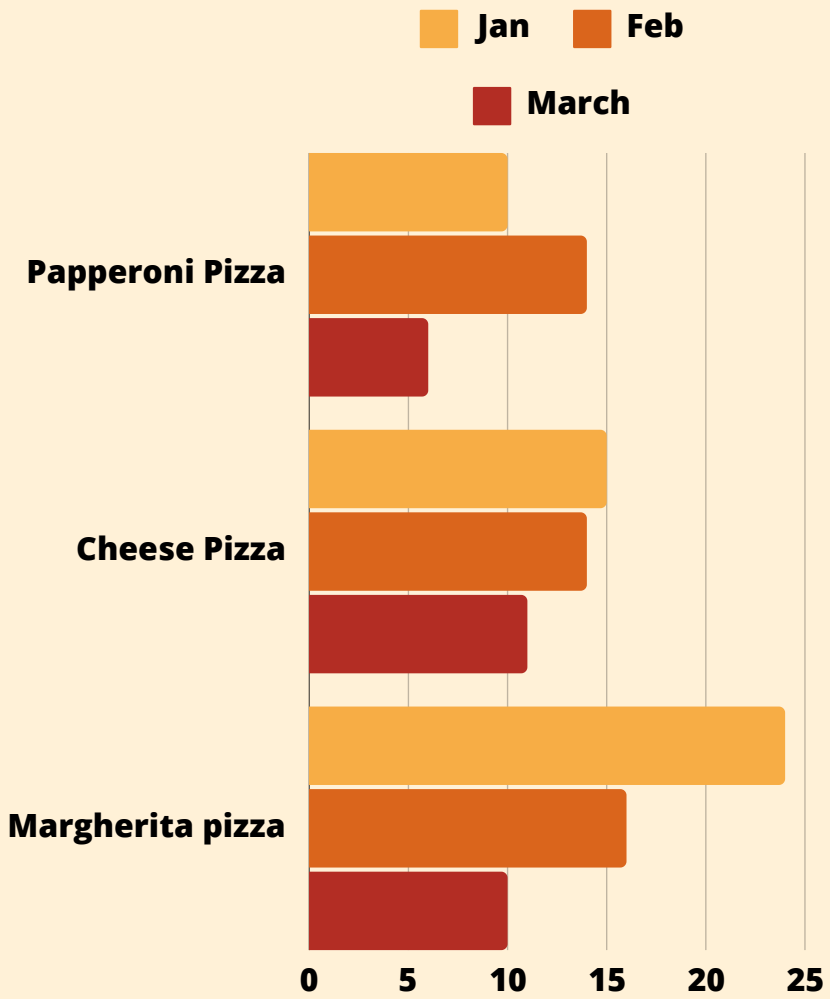
LICERIA
& CO.

PIZZA SALES

IN 3 MONTH



PRESENTED BY
CLAUDIA ALVES

HELLO

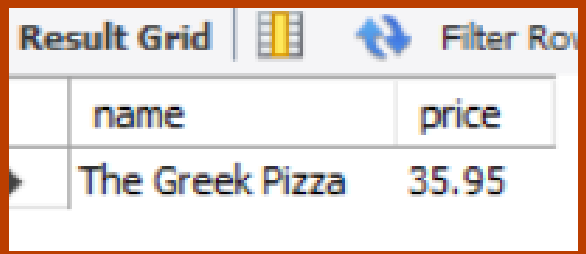
I am Upasana Bajpai ,a 4th year B-tech student. In this project I utilized SQL queries to analyze pizza sales data, extracting key insights such as the revenue contribution of different pizza types.

SQL DATABASE

A SQL database is a structured system that uses tables to store data in rows and columns, ensuring organized and easily manageable information. It utilizes SQL (Structured Query Language) for defining, manipulating, and querying data. Key features include primary keys for unique identification, foreign keys for establishing relationships, and indexes for efficient data retrieval.

Identify the highest-priced pizza

```sql
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

Identify the most common pizza size ordered

```sql
SELECT
    pizzas.size,
    COUNT(orders_detail.orders_detail_id) AS order_count
FROM
    pizzas
        JOIN
    orders_detail ON pizzas.pizza_id = orders_detail.pizza_id
GROUP BY pizzas.size
ORDER BY order_count DESC;
```

Result Grid

size	order_count
L	18526
M	15385
S	14137
XL	544

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

```sql
SELECT
    pizza_types.name, SUM(orders_detail.quantity) AS 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 quantity DESC
LIMIT 5;
```

Result Grid | Filter Rows:

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

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

```sql
use pizzahut;
SELECT
    pizza_types.category,
    SUM(orders_detail.quantity) AS 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 quantity DESC;
```

Result Grid

category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

Determine the distribution of orders by hour of the day

```
SELECT
    HOUR(order_time), COUNT(order_id)
FROM
    orders
GROUP BY HOUR(order_time);
```

Result Grid | Filter Rows:

HOUR(order_time)	COUNT(order_id)
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920

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

```
select category, count(name) from pizza_types
group by category;
```

Result Grid | Filter

category	count(name)
Chicken	6
Classic	8
Supreme	9
Veggie	9

Group the orders by date and calculate the average unumber of pizzas ordered per day

```sql
SELECT
    AVG(quantity) as avg_pizza_ordered_per_day
FROM
    (SELECT
        orders.order_date, SUM(orders_detail.quantity) AS quantity
    FROM
        orders
    JOIN orders_detail ON orders.order_id = orders_detail.order_id
    GROUP BY orders.order_date) AS order_quantity;
```

Result Grid | Filter Rows

avg_pizza_ordered_per_day
138.4749

Determine the top 3 most ordered pizza types based on revenue

```
use pizzahut;
select pizza_types.name,
sum(orders_detail.quantity * pizzas.price) as revenue
from pizza_types join pizzas
on pizzas.pizza_type_id = pizza_types.pizza_type_id
join orders_detail
on orders_detail.pizza_id = pizzas.pizza_id
group by pizza_types.name order by revenue desc limit 3;
```

Result Grid | Filter Rows:

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

Analyse the cumulative revenue generated over select over time

```sql
SELECT order_date,
       SUM(revenue) OVER (ORDER BY order_date) AS cum_revenue
FROM (
    SELECT orders.order_date,
           SUM(orders_detail.quantity * pizzas.price) AS revenue
    FROM orders_detail
    JOIN pizzas ON orders_detail.pizza_id = pizzas.pizza_id
    JOIN orders ON orders.order_id = orders_detail.order_id
    GROUP BY orders.order_date
) AS sales;
```

Result Grid | Filter Rows:

order_date	cum_revenue
2015-01-01	2713.850000000004
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

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

```sql
use pizzahut;
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((orders_detail.quantity) * pizzas.price) as revenue
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, pizza_types.name) as a) as b
where rn <=3;
```

Result Grid | Filter Rows:

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

Calcutae the total revenue generated from pizza sales.

```
USE pizzahut;
SELECT
    ROUND(SUM(orders_detail.quantity * pizzas.price),
            2) AS total_sales
FROM
    orders_detail
        JOIN
    pizzas ON pizzas.pizza_id = orders_detail.pizza_id;
```

Result Grid

	total_sales
▶	817860.05