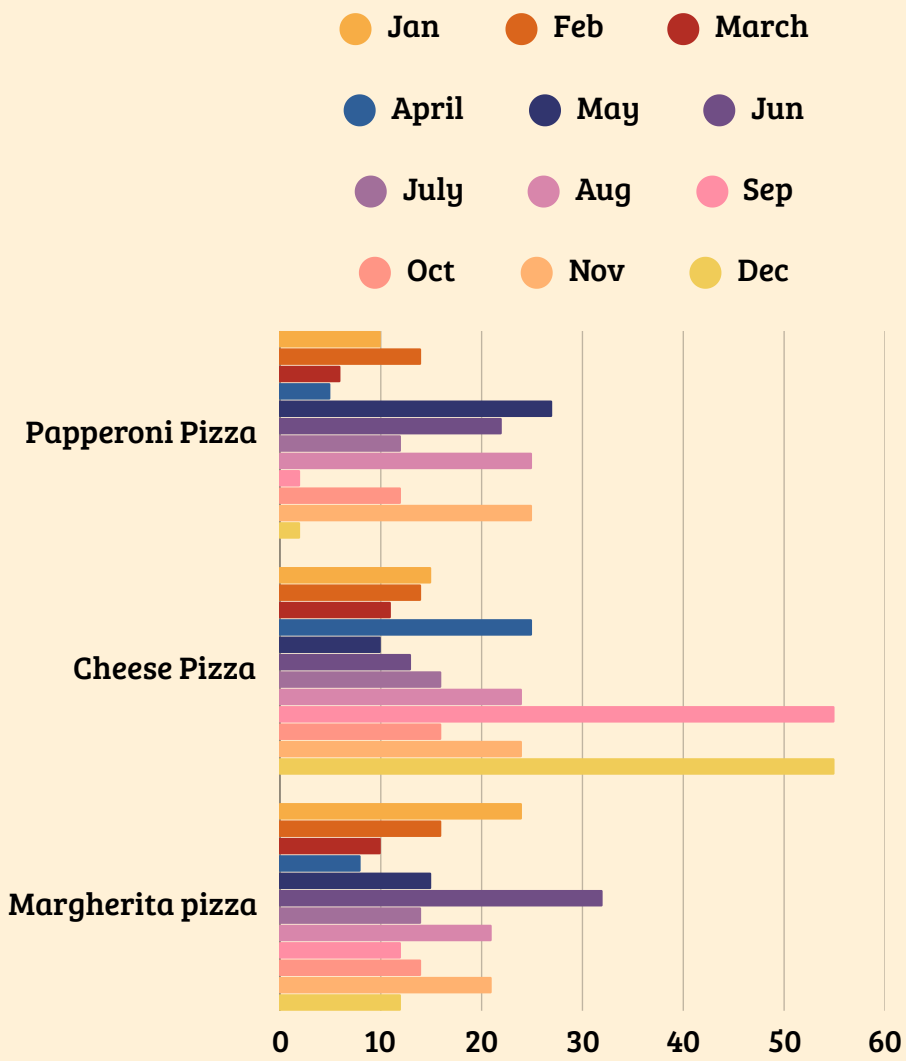
PIZZA HUT

PIZZA SALES
IN 1 YEAR



DATA ANALYSIS-SQL

AREEJ BADAR

Calculate the total revenue generated from pizza sales.

```sql
-- 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;
```

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

total_sales
817860.05

Retrieve the total number of orders placed.

```sql
-- Retrieve the total number of orders placed.
SELECT
    COUNT(order_id) AS total_order
FROM
    orders;
```

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

total_order
21350

Identify the highest-priced pizza.

```sql
SELECT
    (pizza_types.name) AS pizza_name,
    (pizzas.price) AS highest_price
FROM
    pizza_types
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
ORDER BY highest_price DESC
LIMIT 1
;
```

pizza_name	highest_price
The Greek Pizza	35.95

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

```sql
-- List the top 5 most ordered pizza types along with their quantities.
SELECT
    pizza_types.name AS pizza_name,
    SUM(order_details.quantity) AS pizza_quantity
FROM
    pizzas
        JOIN
    pizza_types ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizza_name
ORDER BY pizza_quantity desc limit 5;
```

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

pizza_name	pizza_quantity
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

Identify the most common pizza size ordered.

```sql
-- Identify the most common pizza size ordered.

SELECT
    (pizzas.size) AS pizza_size,
    COUNT(order_details_id) AS total_order
FROM
    pizzas
        JOIN
    order_details ON pizzas.pizza_id = order_details.
GROUP BY pizza_size
ORDER BY total_order DESC
LIMIT 1;
```

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

pizza_size	total_order
L	18526

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

```sql
-- Join the necessary tables to find the total quantity of each pizza category ordered.
SELECT
    pizza_types.category AS 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 pizzas.pizza_id = order_details.pizza_id
GROUP BY category
ORDER BY quantity DESC;
```

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

category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

Determine the distribution of orders by hour of the day.

```sql
-- Determine the distribution of orders by hour of the day.
SELECT
    HOUR(order_time) AS hour, COUNT(order_id) AS orders
FROM
    orders
GROUP BY hour;
```

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

hour	orders
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

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

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

SELECT
    ROUND(AVG(quantity), 0) as avg_order_per_day
FROM
    (SELECT
        (orders.order_date) AS order_date,
            SUM(order_details.quantity) AS quantity
    FROM
        orders
    JOIN order_details ON orders.order_id = order_details.order_id
    GROUP BY order_date) AS order_quantity;
```

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

avg_order_per_day
138

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

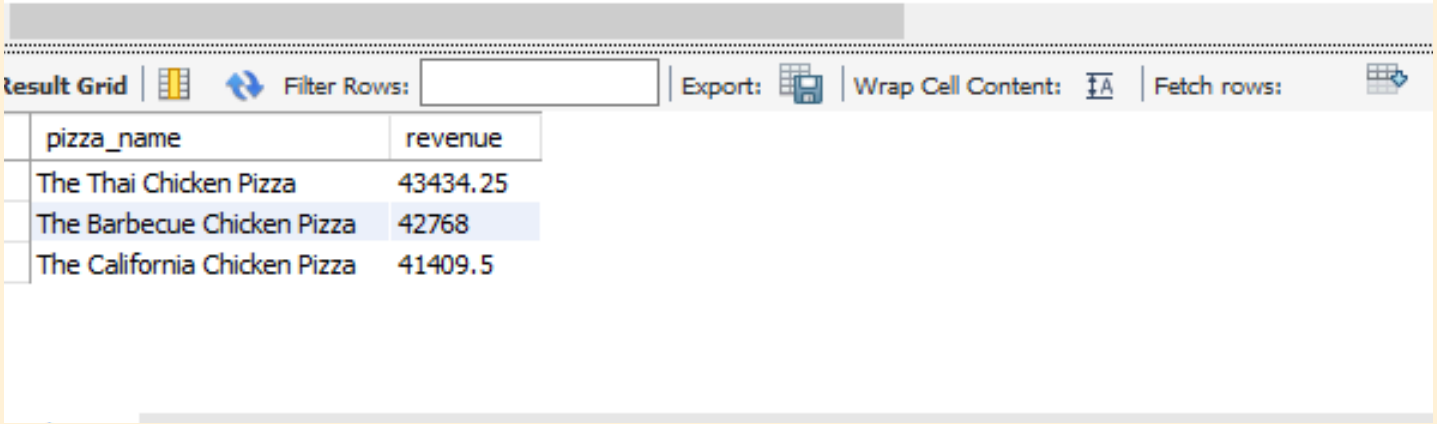
```sql
-- Join relevant tables to find the category-wise distribution of pizzas.
SELECT
    category, COUNT(name) AS quantity
FROM
    pizza_types
GROUP BY category
ORDER BY quantity;
```

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

category	quantity
Chicken	6
Classic	8
Supreme	9
Veggie	9

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

```sql
-- Determine the top 3 most ordered pizza types based on revenue.
SELECT
    (pizza_types.name) AS pizza_name,
    SUM(pizzas.price * order_details.quantity) 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_name
ORDER BY revenue DESC
LIMIT 3
;
```



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

Analyze the cumulative revenue generated over time.

```sql
-- 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) as order_date ,
sum(order_details.quantity * pizzas.price) as revenue
from orders join order_details on
orders.order_id = order_details.order_id
join pizzas on
order_details.pizza_id = pizzas.pizza_id
group by 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

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

```sql
-- Determine the top 3 most ordered pizza types based on revenue for each pizza category.
select name, revenue from
(select name, category, revenue,
rank() over(partition by category order by revenue) as rn
from
(select pizza_types.category, pizza_types.name,
sum(pizzas.price * order_details.quantity) 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 Chicken Pesto Pizza	16701.75
The Chicken Alfredo Pizza	16900.25
The Southwest Chicken Pizza	34705.75
The Pepperoni, Mushroom, and Peppers Pizza	18834.5
The Big Meat Pizza	22968

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

```sql
-- Calculate the percentage contribution of each pizza type to total revenue.
SELECT
    (pizza_types.name) AS pizza_name,
    ROUND(SUM(pizzas.price * order_details.quantity) / (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_name
ORDER BY revenue
;
```

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

pizza_name	revenue
The Brie Carre Pizza	1.42
The Green Garden Pizza	1.71
The Spinach Supreme Pizza	1.87
The Mediterranean Pizza	1.88
The Spinach Pesto Pizza	1.91
The Calabrese Pizza	1.95
The Italian Vegetables Pizza	1.96
The Soppressata Pizza	2.01
The Chicken Pesto Pizza	2.04
The Chicken Alfredo Pizza	2.07
The Pepperoni, Mushroom,...	2.3
The Big Meat Pizza	2.81
The Spinach and Feta Pizza	2.85
The Napolitana Pizza	2.95
The Prosciutto and Arugula...	2.96
The Vegetables + Vegetabl...	2.98
The Italian Capocollo Pizza	3.07
The Pepper Salami Pizza	3.12
The Five Cheese Pizza	3.19

Project Summary
Pizza Hut Sales Analysis using SQL

What I Did:

- Analyzed Pizza Hut's sales dataset using SQL to extract business insights.
- Used basic to advanced SQL queries involving aggregation, filtering, joins, grouping, and window functions.
- Explored sales performance by orders, revenue, product type, time trends, and category-wise breakdowns.

Key Learnings:

- Applied real-world SQL skills such as:
  - Data retrieval & aggregation (SUM, COUNT, AVG)
  - Table relationships through joins
  - Time-based analysis using DATE & TIME functions
  - Advanced techniques like cumulative totals and percentage revenue share
- Developed the ability to transform raw data into actionable insights for business decision-making.

How This Project Supports My Career:

- Strengthens my profile as an aspiring Data Analyst with hands-on project experience.
- Demonstrates my ability to perform data-driven storytelling using SQL.
- Prepares me for real-world scenarios in industries like retail, food services, and e-commerce.
- Enhances my portfolio with a practical project showing problem-solving, logic, and analytics.

"This project marks a strong step in my data analytics journey, turning raw data into real insights with SQL."