Welcome to the Pizza Sales Data Analytics Project

This project aims to provide insightful analysis into the sales data of a pizza restaurant. By leveraging data analytics techniques, we will uncover valuable insights to optimize business strategies and enhance decision-making processes.

Pizza Sales Analysis
By Abhay Pandey

Project Overview and Tasks

Basic Tasks:

1. **Basic Tasks:**
2. **Retrieve the total number of orders placed.**
3. **Calculate the total revenue generated from pizza sales.**
4. **Identify the highest-priced pizza.**
5. **Identify the most common pizza size ordered.**
6. **List the top 5 most ordered pizza types along with their quantities.**

Intermediate Tasks:

1. **Join the necessary tables to find the total quantity of each pizza category ordered.**
2. **Determine the distribution of orders by hour of the day.**
3. **Join relevant tables to find the category-wise distribution of pizzas.**
4. **Group the orders by date and calculate the average number of pizzas ordered per day.**
5. **Determine the top 3 most ordered pizza types based on revenue.**

Advanced Tasks:

1. **Calculate the percentage contribution of each pizza type to total revenue.**
2. **Analyze the cumulative revenue generated over time.**
3. **Determine the top 3 most ordered pizza types based on revenue for each pizza category.**

Explanation:

The database "pizzahut" serves as the foundation for our data analytics project. It comprises four essential tables: orders, orders_details, pizza_types, and pizzas. These tables contain vital information about orders, pizzas, and related details necessary for conducting comprehensive analysis.

Screenshot of database & tables :-

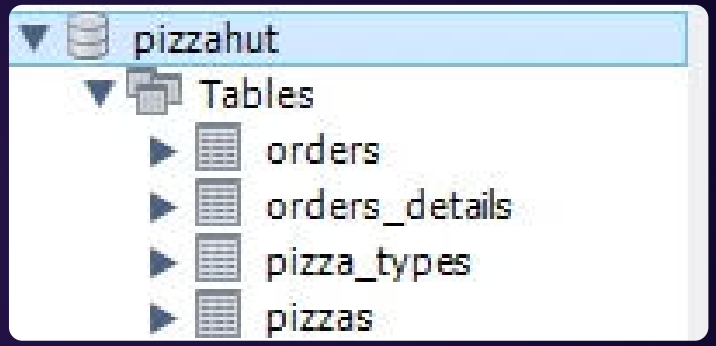
```sql
create database pizzahut;
use pizzahut;

create table orders (
order_id int not null,
order_date date not null,
order_time time not null,
primary key(order_id));

select * from  orders;

create table orders_details (
order_details_id int not null,
order_id int not null,
pizza_id text not null,
order_time time not null,
quantity int not null,
primary key(order_details_id));

alter table orders_details
drop column order_time;
select
```



Basic Tasks:

- **Retrieve the total number of orders placed.**
- **Calculate the total revenue generated from pizza sales.**

```sql
-- Retrieve the total number of orders placed.

SELECT
    COUNT(order_id) AS total_orders
FROM
    orders;
```

Result Grid | Filter Rows:

total_orders
21350

Result 1

```sql
-- Calculate the total revenue generated from pizza sales.

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

Result Grid | Filter Rows:

total_sales
817860.05

Result 1

Pizza Sales Analysis
By Abhay Pandey

- **Identify the highest-priced pizza.**
- **Identify the most common pizza size ordered.**

```sql
-- 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;
```

Result Grid | Filter Rows:

name	price
The Greek Pizza	35.95

Result 1

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


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

Result Grid | Filter Rows:

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

Result 1

Pizza Sales Analysis
By Abhay Pandey

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

Intermediate Tasks:

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

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

SELECT
    pizza_types.name, SUM(orders_details.quantity) AS quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    orders_details ON orders_details.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

Result 1

```sql
-- Join the necessary tables to find the total quantity of each pizza category category.

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

Result Grid | Filter Rows

category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

Result 1

Pizza Sales Analysis
By Abhay Pandey

- **Determine the distribution of orders by hour of the day.**
- **Join relevant tables to find the category-wise distribution of pizzas**

```sql
-- Determine the distribution of orders by hour of the day.

SELECT
    HOUR(order_time) AS hour, COUNT(order_id) AS order_count
FROM
    orders
GROUP BY HOUR(order_time);

```

Result Grid | Filter Rows:

hour	order_count
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

Result 1

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

SELECT
    category, COUNT(name)
FROM
    pizza_types
GROUP BY category;

```

Result Grid | Filter Rows:

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

Result 1

Pizza Sales Analysis
By Abhay Pandey

- **Group the orders by date and calculate the average number of pizzas ordered per day.**
- **Determine the top 3 most ordered pizza types based on revenue.**

```sql
-- 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(orders_details.quantity) AS quantity
    FROM
        orders
    JOIN orders_details ON orders.order_id = orders_details.order_id
    GROUP BY orders.order_date) AS order_quanitity;

```

Result Grid | Filter Rows:

avg_pizza_ordered_per_day
138

Result 1

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


SELECT
    pizza_types.name,
    SUM(orders_details.quantity * pizzas.price) AS revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    orders_details ON orders_details.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

Result 1

Pizza Sales Analysis
By Abhay Pandey

Advanced Tasks:

- Calculate the percentage contribution of each pizza type to total revenue.
- Analyze the cumulative revenue generated over time.

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


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

Result Grid | Filter Rows:

category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

Result 1

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


```

Result Grid | Filter Rows:

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

Result 1

Pizza Sales Analysis
By Abhay Pandey

- 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 category, name, revenue,
rank() over (partition by category order by revenue desc) as rn
from
(select pizza_types.category, pizza_types.name,
sum((orders_details.quantity) * pizzas.price) as revenue
from pizza_types join pizzas
on pizza_types.pizza_type_id = pizzas.pizza_type_id
join orders_details
on orders_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:

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
The Mexicana Pizza	26780.75
The Five Cheese Pizza	26066.5

Result 1

End

Pizza Sales Analysis
By Abhay Pandey