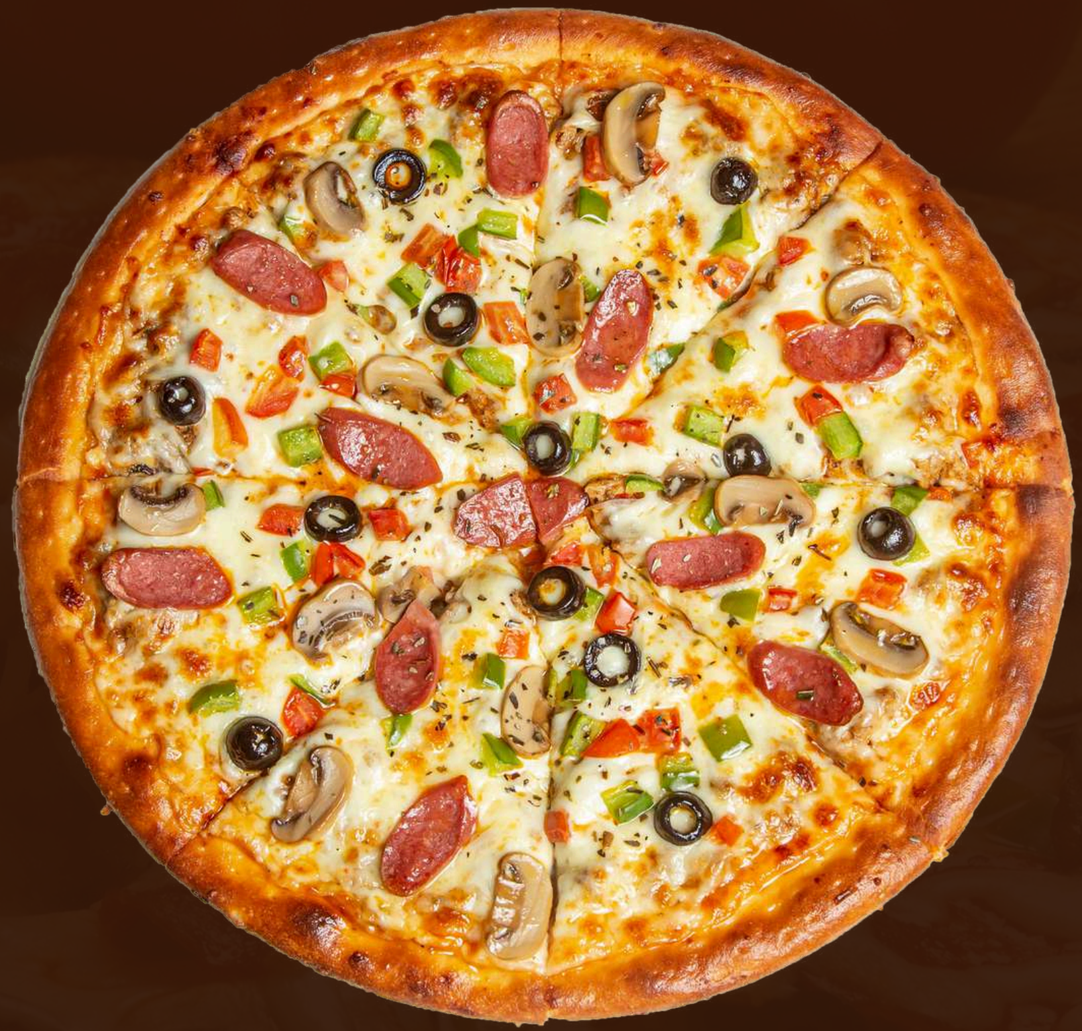
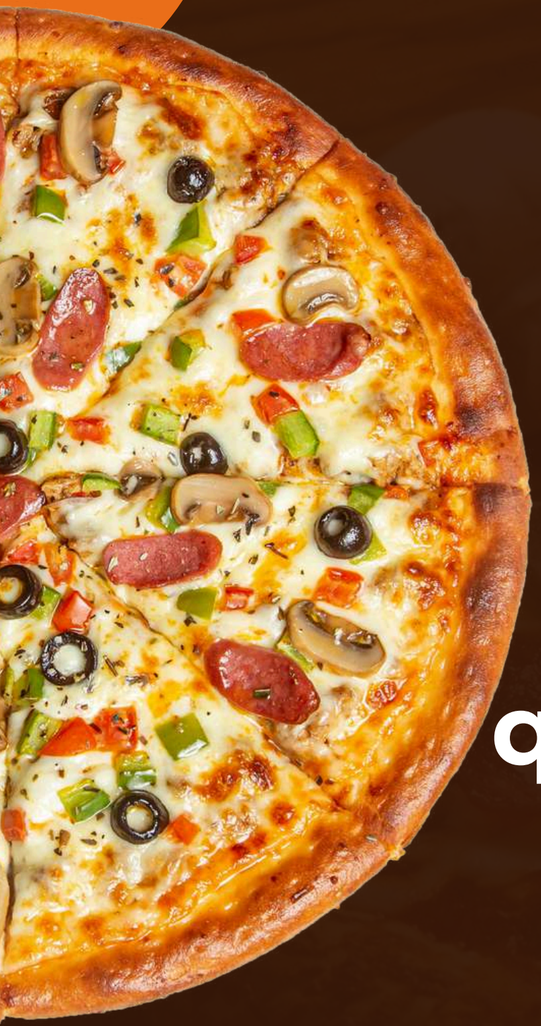
SQL PROJECT ON PIZZA SALES

Presenter : Vishal Patil

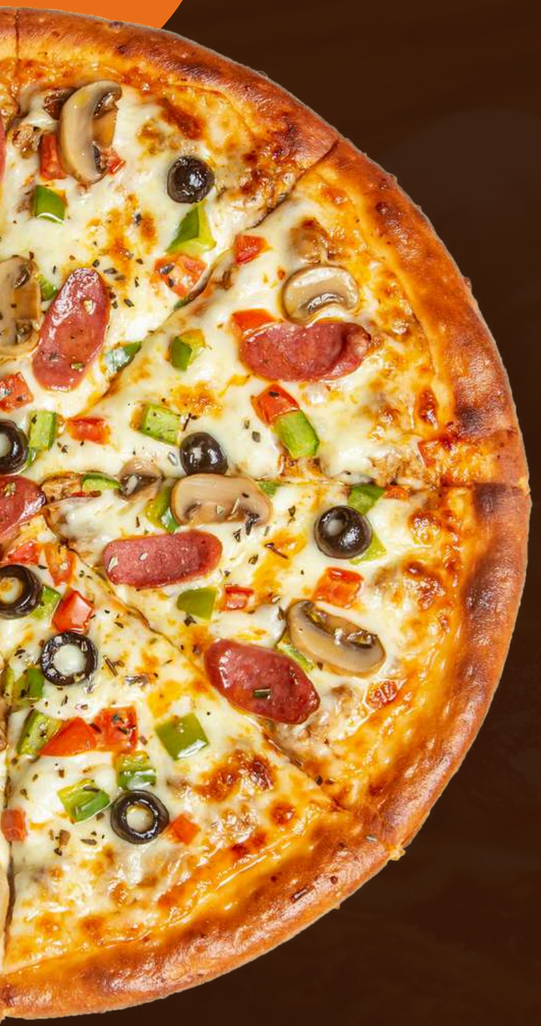




INTRODUCTION

Hi, I am Vishal Patil.
In this project I have utilized SQL queries to solve problems related to pizza sales.

During this project I have created a database and its corresponding tables. Performed necessary operations from beginner to advance level to achieve expected results.



Solved Questions

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

- **Intermediate:**
- Join the necessary tables to find the total quantity of each pizza category ordered.
- Determine the distribution of orders by hour of the day.
- Join relevant tables to find the category-wise distribution of pizzas.
- 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.

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

Q. Calculate the total revenue generated from pizza sales.

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

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

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

total_sales
817860.05

Q. 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, SUM(order_details.quantity) AS quantity
FROM
    order_details JOIN pizzas
    ON pizzas.pizza_id = order_details.pizza_id
        JOIN
    pizza_types
    ON pizzas.pizza_type_id = pizza_types.pizza_type_id
GROUP BY pizza_types.name
ORDER BY SUM(order_details.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

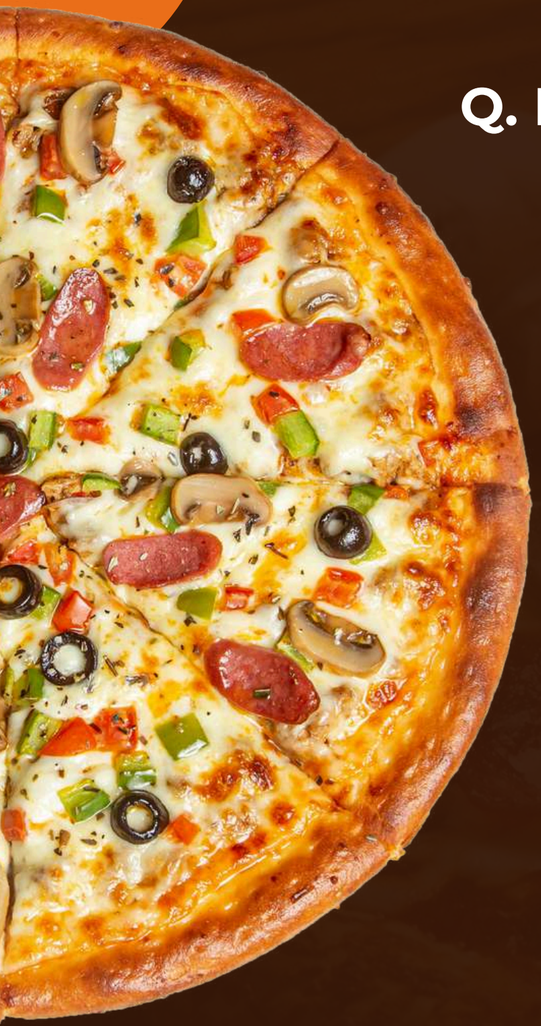
Q. Determine the top 3 most ordered pizza types based on revenue.

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

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

Result Grid | Filter Rows:

name	revenue
The Classic Deluxe Pizza	38180.5
The Spicy Italian Pizza	34831.25
The Southwest Chicken Pizza	34705.75
The Italian Supreme Pizza	33476.75
The Hawaiian Pizza	32273.25



Q. 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) hours, COUNT(order_id) AS order_count
FROM
    orders
GROUP BY hours
ORDER BY order_count DESC;
```

Result Grid | Filter Rows:

hours	order_count
12	2520
13	2455
18	2399
17	2336
19	2009
16	1920
20	1642
14	1472
15	1468
11	1231
21	1198
22	663
23	28

Q. 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.category as category ,
round((sum(order_details.quantity*pizzas.price)/
(SELECT SUM(pizzas.price * order_details.quantity) AS total_sales
FROM pizzas JOIN order_details
ON pizzas.pizza_id = order_details.pizza_id))*100, 2) as revenue

from order_details join pizzas
on order_details.pizza_id = pizzas.pizza_id

join pizza_types
on pizzas.pizza_type_id = pizza_types.pizza_type_id

group by category
order by revenue desc;
```

Result Grid | Filter Rows

category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

Q. Analyze the cumulative revenue generated over time.

```sql
-- Analyze the cumulative revenue generated over time.
select date,
sum(revenue) over(order by date) as cum_revenue
from
(select orders.order_date as date, round(sum(order_details.quantity*pizzas.price), 2) 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 date) as sales;
```

Result Grid | Filter Rows:

date	cum_revenue
2015-01-01	2713.85
2015-01-02	5445.75
2015-01-03	8108.15
2015-01-04	9863.6
2015-01-05	11929.55

Q. 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 category,name, revenue
from
(select category, name, revenue,
rank() over(partition by category order by revenue desc) as rn
from
(select
pizza_types.category as category,
pizza_types.name as 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 pizzas.pizza_id = order_details.pizza_id
group by category, name) as a) as b
where rn<=3;
```

Result Grid | Filter Rows: | Export:

category	name	revenue
Supreme	The Spicy Italian Pizza	34831.25
Supreme	The Italian Supreme Pizza	33476.75
Supreme	The Sicilian Pizza	30940.5
Veggie	The Four Cheese Pizza	32265.70000000065
Veggie	The Mexicana Pizza	26780.75

THANK YOU