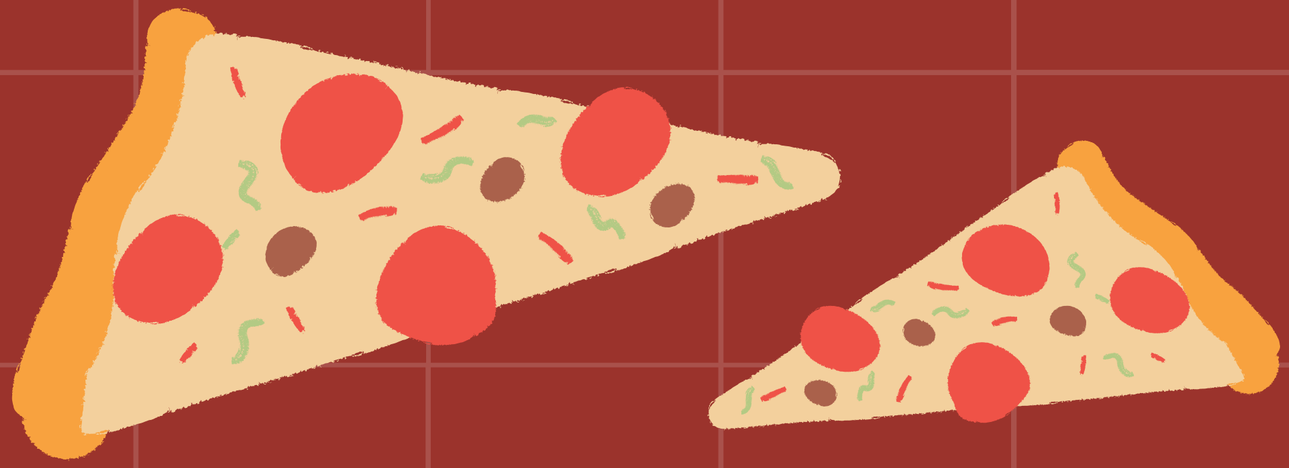
PIZZA SALES SQL REPORT



PROJECT OVERVIEW

This project uses SQL to analyze the dataset of pizza sales. The analysis covers various aspects such as pizza types,order values,order quantities and order date and time.

Steps

DATABASE SCHEMA

1.Order_details

order_details_id
order_id
pizza_id
quantity

2.orders

order_date
order_id
order_time

3.pizzas

pizza_id
pizza_type_id
size
price

4.pizza_types

pizza_type_id
name
category
ingredients

Problems on which are going to apply our SQL Queries:

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.

Retrieve the total number of orders placed

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

Result Grid

total_orders
21350

Calculate the total revenue generated from pizza sales.

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

Result Grid

total_sales
817860.05

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
```

Result Grid | Filter Rows:

name	price
The Greek Pizza	35.95

Identify the most common pizza size ordered.

```sql
SELECT
    pizzas.size,
    COUNT(orders_details.order_details_id) total_count
FROM
    pizzas
        JOIN
    orders_details ON pizzas.pizza_id = orders_details.pizza_id
GROUP BY pizzas.size
ORDER BY total_count DESC
LIMIT 3
```

Result Grid

size	total_count
L	18526
M	15385
S	14137

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

```
    pizza_types.name,
    SUM(orders_details.quantity) total_quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    orders_details ON pizzas.pizza_id = orders_details.pizza_id
GROUP BY pizza_types.name
ORDER BY total_quantity DESC
LIMIT 5
```

Result Grid | Filter Rows:

name	total_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
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 pizzas.pizza_id = orders_details.pizza_id
GROUP BY pizza_types.category
```

Result Grid

category	quantity
Classic	14888
Veggie	11649
Supreme	11987
Chicken	11050

Determine the distribution of orders by hour of the day.

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

Result Grid

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
10	8
9	1

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

```sql
SELECT
    category, COUNT(name)
FROM
    pizza_types
GROUP BY category
```

Result Grid | Filter Rov

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

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

```sql
SELECT
    round(AVG(quantity),0) as avg_pizza_ordered_perday
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_quantity
```

Result Grid | Filter Rows:

avg_pizza_ordered_perday
138

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

```sql
SELECT
    pizza_types.name,
    SUM(pizzas.price * orders_details.quantity) 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.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

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 pizza_types.pizza_type_id=pizzas.pizza_type_id
join orders_details
on pizzas.pizza_id=orders_details.pizza_id
group by pizza_types.category
order by revenue desc
```

Result Grid

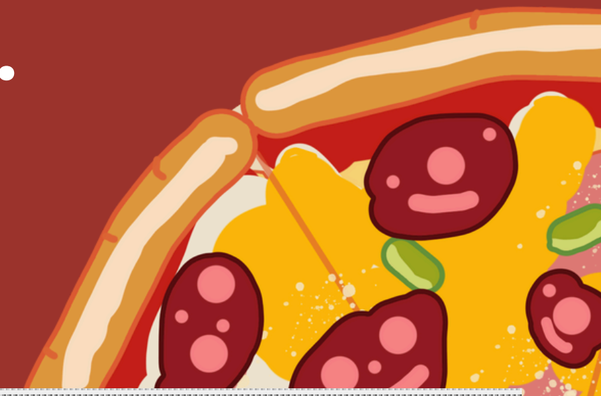
category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

Analyze the cumulative revenue generated over time.

```sql
select order_date,
round(sum(revenue) over(order by order_date),2) as comulative_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
```

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

```
(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


SQL PIZZA SALES
PROJECT
THANK YOU