



Pizza Sales Project using SQL



Project Description

The **Pizza Sales Analysis using SQL** project is all about using data to help a pizza business understand its sales better and run more smoothly. It involves building a database with key information like customer details, orders, pizzas, and ingredients. With SQL queries, you can easily see which pizzas are the most popular, track sales over time, and keep an eye on inventory levels to avoid running out of key ingredients. This project helps the business spot trends, understand customer preferences, and improve things like stock management and delivery times, ultimately leading to smarter decisions and better customer service.



Questions to solve

- 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.

Questions to solve

- Join the necessary tables to find the total quantity of each pizza category ordered.
- Determine the distribution of orders by top 5 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.

Questions to solve

- 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.

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



Output

| | |
|---|---------------------|
| | total_orders |
| ▶ | 21350 |



-- 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 orders_details.pizza_id =  
            pizzas.pizza_id;
```

Output

| | |
|---|--------------------|
| | total_sales |
| ▶ | 817860.05 |



-- 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;
```

| | name | price |
|---|-----------------|--------------|
| ▶ | The Greek Pizza | 35.95 |

Output

| | name | price |
|---|-----------------|--------------|
| ▶ | The Greek Pizza | 35.95 |



-- 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;
```

Output

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



-- 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;
```

Output

| | name | quantity |
|---|-----------------------------------|-----------------|
| ▶ | The Classic Deluxe Pizza | 2453 |
| | The Barbecue Chicken Pizza | 2432 |
| | The Hawalian Pizza | 2422 |
| | The pepperoni Pizza | 2418 |
| | The Thsi Checken Pizza | 2371 |



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

```
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;
```

Output

| | category | quantity |
|---|----------|----------|
| ▶ | Classic | 14888 |
| | Supreme | 11987 |
| | Veggie | 11649 |
| | Chicken | 11050 |



-- Determine the distribution of orders by top 5 hour of the day.

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

Output

| | hour | order_count |
|---|------|-------------|
| ▶ | 12 | 2520 |
| | 13 | 2455 |
| | 18 | 2399 |
| | 17 | 2336 |
| | 19 | 2009 |



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

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

Output

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

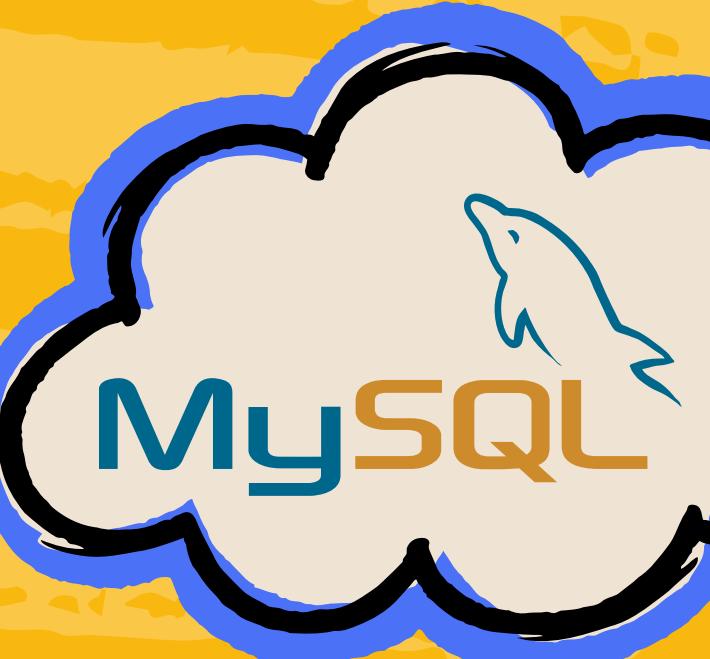


-- Group the orders by date and calculate the average number of pizzas ordered per day.

```
SELECT
    ROUND(AVG(quantity), 0) AS avg_pizzas_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_quantity;
```

Output

| | |
|---|----------------------------------|
| | avg_pizza_ordered_per_day |
| ▶ | 138 |





-- 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;
```

Output

| | name | revenue |
|---|-------------------------------------|-----------------|
| ▶ | The Thai Checken 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 orders_details.pizza_id = pizzas.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 orders_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY revenue DESC;
```

Output

| | category | revenue |
|---|----------|---------|
| ▶ | Classic | 26.91 |
| | Supreme | 25.46 |
| | Veggie | 23.96 |
| | Chicken | 23.68 |



-- 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  
order by cum_revenue limit 5;
```

Output

| | order_date | cum_revenue |
|---|-------------------|--------------------|
| ▶ | 2015-01-01 | 2713.85 |
| | 2015-01-02 | 5445.75 |
| | 2015-01-03 | 8180.15 |
| | 2015-01-04 | 9863.6 |
| | 2015-01-05 | 11929.55 |



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

```
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  
order by cum_revenue limit 5;
```

Output

| | name | revenue |
|---|-------------------------------------|-----------------|
| ▶ | The Thai Checken Pizza | 43434.25 |
| | The Barbecue Chicken Pizza | 42768 |
| | The California Chicken Pizza | 41409.5 |
| | The Classic Deluxe Pizza | 38180.5 |
| | The Hawailian Pizza | 32273.25 |

Thank You!

For any questions or feedback, feel free to reach out

amarhadke2726@gmail.com

