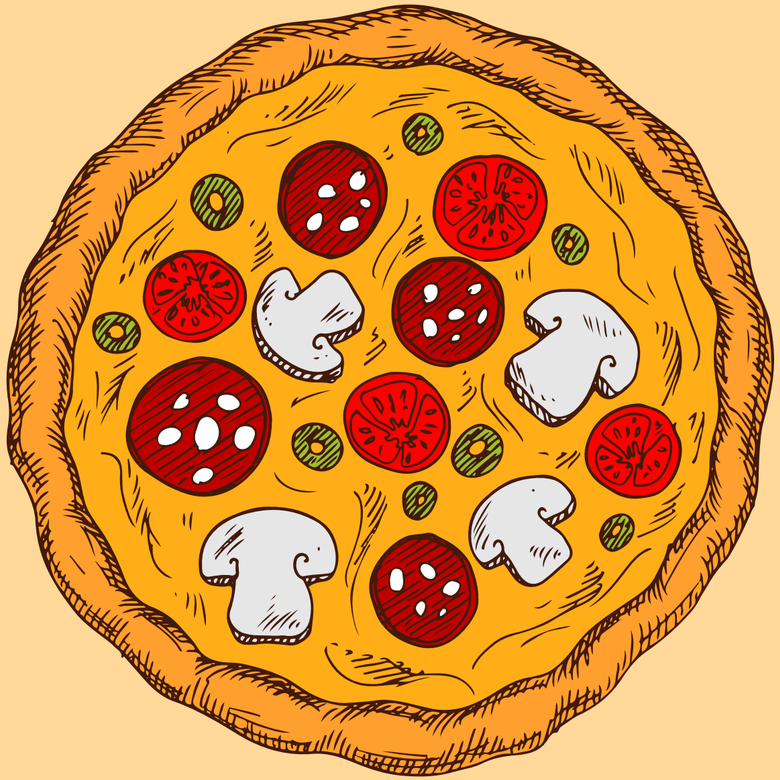


Delicious Pizza for Everyone!

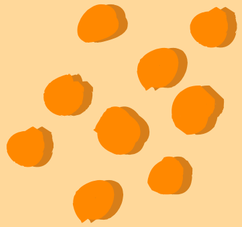
PIZZA SALES

using SQL

This project focuses on data analysis of pizza sales using SQL. The dataset consists of four key tables:

- orders: Captures information about customer orders, including order ID, date, and customer details.
- orders_details: Details the items in each order, including the quantity and size of each pizza ordered.
- pizza_type: Describes the different types of pizzas available, including the pizza type ID, name, and ingredients.
- pizzas: Contains information about individual pizzas, including their ID, type, and price.

The goal of the project is to perform various analyses, such as identifying the most popular pizza types, peak order times, and sales trends, to derive actionable insights that can enhance business decision-making.



This SQL-based project focuses on analyzing pizza sales data, utilizing four key tables: orders, orders_details, pizza_type, and pizzas. Through this project, we aim to explore and solve a range of questions at varying levels of difficulty. These questions will involve tasks such as:

- Identifying the most popular pizza types.
- Analyzing sales trends over time.
- Determining peak order times.
- Calculating revenue generated by different pizza sizes and types.

By solving these questions, we will demonstrate different SQL techniques, from basic querying to more complex data analysis, providing a comprehensive understanding of how SQL can be used to extract meaningful insights from sales data.

RETRIVE THE TOTAL NUMBER OF ORDERS PLACED

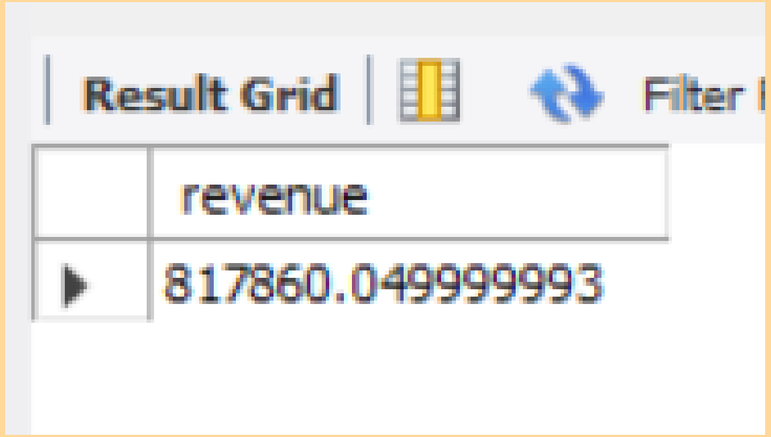
```
select count(order_id) from orders;
```

Result Grid

	count(order_id)
▶	21350

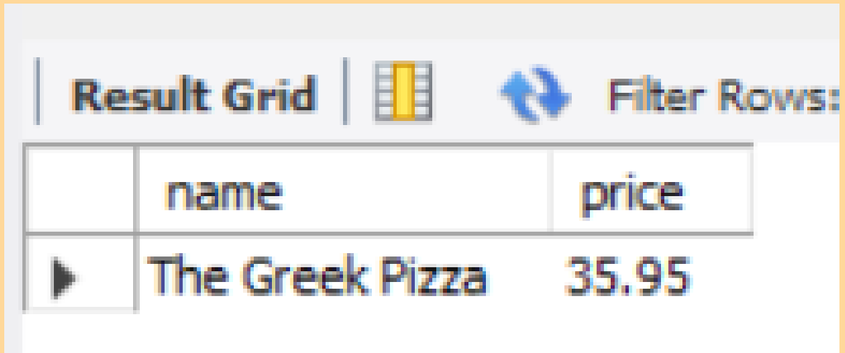
Total revenue genertaed from pizza sale

```sql
SELECT
    SUM(orders_details.quantity * pizzas.price) AS revenue
FROM
    orders_details
        JOIN
    pizzas ON pizzas.pizza_id = orders_details.pizza_id
```



Highest price 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;
```



	name	price
▶	The Greek Pizza	35.95

Most common pizza size ordered

```sql
SELECT
    pizzas.size, COUNT(orders_details.order_details_id) AS c
FROM
    pizzas
        JOIN
    orders_details ON pizzas.pizza_id = orders_details.pizza_id
GROUP BY pizzas.size
ORDER BY c DESC
LIMIT 1;
```

Result Grid

size	c
L	18526

Top 5 most ordered pizza types along with their quantities

```sql
SELECT
    pizza_types.name, SUM(orders_details.quantity) AS s
FROM
    pizzas
        JOIN
    pizza_types 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 s DESC
LIMIT 5;
```

Result Grid | Filter Rows:

name	s
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

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

```sql
SELECT
    pizza_types.category, SUM(orders_details.quantity) AS q
FROM
    pizzas
        JOIN
    pizza_types 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 q;
```

Result Grid

category	q
Chicken	11050
Veggie	11649
Supreme	11987
Classic	14888

Determine the distribuion of orders by hour of the day

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

Result Grid | Filter Rows:

HOUR(order_time)	COUNT(order_id)
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

Join relevant table to find the category wise distribution of pizzas

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

Result Grid | Filter Ro

	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(q), 0)
FROM
    (SELECT
        orders.order_date, SUM(orders_details.quantity) AS q
    FROM
        orders
    JOIN orders_details ON orders.order_id = orders_details.order_id
    GROUP BY orders.order_date) AS date
```

Result Grid

	ROUND(AVG(q), 0)
▶	138

Determine the top 3 most ordered pizza types on the basis of revenue

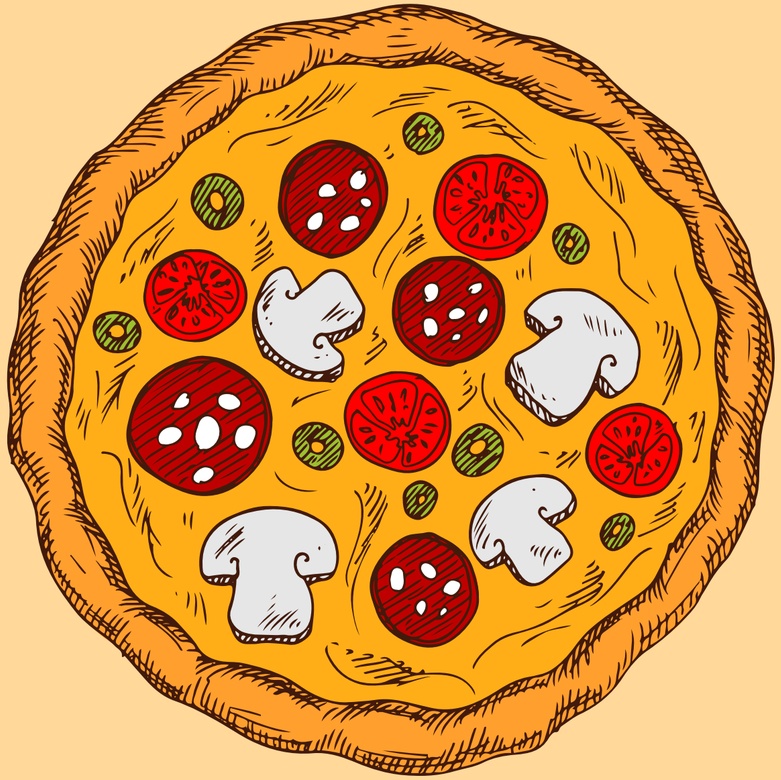
```sql
SELECT
    pizza_types.name,
    SUM(orders_details.quantity * pizzas.price) AS s
FROM
    pizzas
        JOIN
    pizza_types ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    orders_details ON orders_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY s DESC
LIMIT 3
```

name	s
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5

Determine the top 3 most ordered pizza types on the basis of revenue for each pizza category

```sql
select pizza_types.category as c , pizza_types.name as d ,
sum(orders_details.quantity * pizzas.price) as s
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 c , d
order by s desc
limit 3
```

c	d	s
Chicken	The Thai Chicken Pizza	43434.25
Chicken	The Barbecue Chicken Pizza	42768
Chicken	The California Chicken Pizza	41409.5



PIZZA SALES ANALYSIS

THANK YOU

JAYESH PATIL