



KISHAN KUMAR
NIT JAMSHEDPUR

PIZZA SALES ANALYSIS

MY SQL

DATA FILES

Data File -

<https://github.com/Kishankumar2100/Pizza-Sales-Analysis>

01

PIZZAS.CSV-

This file includes details about individual pizzas, with columns for pizza_id, pizza_type_id, size, and price. It provides the basic structure needed to understand the different types of pizzas, their sizes, and their pricing.

02

PIZZA_TYPES.CSV

This file categorizes the pizzas further by providing pizza_type_id, name, category, and ingredients. It helps in identifying the type and category of each pizza, along with a detailed list of ingredients.

03

ORDERS.CSV

This file captures the order details, including order_id, date, and time. It is essential for analyzing when orders were placed and understanding sales trends over time.

04

ORDER_DETAILS.CSV

This file connects orders with specific pizzas, containing order_details_id, order_id, pizza_id, and quantity. It provides a detailed view of what was ordered in each transaction and is crucial for analyzing order patterns and customer preferences.

Q1. Retrieve the total number of orders placed.

SQL QUERY

```
SELECT COUNT(order_id) AS Total_Order from orders;
```



Total_Order
21350

OUTPUT

Q2.Calculate the total revenue generated from pizza sales.

SQL QUERY

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



	Total_Revenue
▶	817860.05

OUTPUT

Q3.Identify the highest-priced pizza.

SQL QUERY

```
SELECT pizza_types.name AS pizza , pizzas.price AS price  
FROM pizza_types  
JOIN pizzas  
    ON pizza_types.pizza_type_id = pizzas.pizza_type_id  
ORDER BY price DESC  
LIMIT 1;
```

OUTPUT

pizza	price
The Greek Pizza	35.95

Identify the most common pizza size ordered.

SQL QUERY

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

OUTPUT

size	order_count
L	18526

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

SQL QUERY

```
SELECT pizza_types.name, sum(order_details.quantity) AS quantity
FROM pizza_types
JOIN pizzas
ON pizza_types.pizza_type_id = pizzas.pizza_type_id
JOIN order_details
ON order_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 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 QUERY

```
SELECT pizza_types.category,  
       SUM(order_details.quantity) AS quantity  
  FROM pizza_types  
 JOIN pizzas  
    ON pizza_types.pizza_type_id = pizzas.pizza_type_id  
 JOIN order_details  
    ON order_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 hour of the day.

SQL QUERY

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

OUTPUT

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 QUERY

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

OUTPUT

category	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 QUERY

```
SELECT round(AVG(quantity),0)
FROM
    (SELECT orders.order_date, SUM(order_details.quantity) as quantity
     FROM orders join order_details
     ON orders.order_id = order_details.order_id
     GROUP BY orders.order_date) as order_quantity;
```

OUTPUT

round(AVG(quantity),0)
138

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

SQL QUERY

```
SELECT pizza_types.name,  
       SUM(order_details.quantity * pizzas.price) as revenue  
  FROM pizza_types  
  JOIN pizzas  
    ON pizzas.pizza_type_id = pizza_types.pizza_type_id  
  JOIN order_details  
    ON order_details.pizza_id = pizzas.pizza_id  
 GROUP BY pizza_types.name  
 ORDER BY revenue DESC  
 LIMIT 3;
```

OUTPUT

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

THANK YOU

Github Link -

<https://github.com/Kishankumar2100/Pizza-Sales-Analysis>

