

PIZZA

sql project on pizza sales

DISCOUNT · DISCOUNT · DISCOUNT · DISCOUNT · DISCOUNT · DISCOUNT
25%



WELCOME OUR PIZZA

description here



The Pizza Sales Analysis Using SQL project focuses on analyzing pizza shop sales data through SQL queries. It involves creating tables for customers, pizzas, orders, and order details to manage and retrieve data efficiently. Using SQL commands like joins, group by, and aggregate functions, the project identifies best-selling pizzas, total revenue, top customers, and sales trends. This project demonstrates how SQL can turn raw sales data into meaningful business insights.

QUE.1 - Retrieve the total number of orders placed

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

Result Grid	
	total_orders
▶	21350



QUE.2 - Identify the highest-priced pizza

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

Result Grid | Filter Rows:

	name	price
▶	The Greek Pizza	35.95



QUE.3 - List the top 5 most ordered pizza types along with their quantities

```
SELECT pt.name, SUM(od.quantity) AS quantity
FROM pizza_types pt
    JOIN pizzas p ON pt.pizza_type_id = p.pizza_type_id
    JOIN order_details od ON od.pizza_id = p.pizza_id
GROUP BY pt.name
ORDER BY 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



QUE.4 - Join the necessary tables to find the total quantity of each pizza category ordered

```
SELECT pt.category, SUM(od.quantity) AS quantity
FROM pizza_types pt
JOIN pizzas p ON pt.pizza_type_id = p.pizza_type_id
JOIN order_details od ON od.pizza_id = p.pizza_id
GROUP BY pt.category
ORDER BY quantity DESC;
```

Result Grid | Filter

	category	quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050



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

SELECT

HOUR(order_time) AS hour, COUNT(order_id) AS order_count

FROM

orders

GROUP BY hour

	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



QUE.6 - Join relevant tables to find the category-wise distribution of pizzas

- ```
SELECT DISTINCT
 (category) AS category, COUNT(name) AS count
FROM
 pizza_types
GROUP BY category
```

Result Grid | Filter

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



QUE.7 - Group the orders by date and calculate the average number of pizzas ordered per day

```
SELECT
 ROUND(AVG(quantity), 0) AS avg_pizza_ordered_per_day
FROM
 (SELECT
 o.order_date AS order_date, SUM(od.quantity) AS quantity
 FROM
 orders o
 JOIN order_details od ON o.order_id = od.order_id
 GROUP BY order_date) AS order_quantity
```

|   | Result Grid               |  Filter Rows: |
|---|---------------------------|----------------------------------------------------------------------------------------------------|
|   | avg_pizza_ordered_per_day |                                                                                                    |
| ▶ | 138                       |                                                                                                    |



# QUE.8 - Determine the top 3 most ordered pizza types based on revenue

**SELECT**

```
pt.name, SUM(od.quantity * p.price) AS revenue
```

**FROM**

```
pizza_types PT
```

**JOIN**

```
pizzas P ON pt.pizza_type_id = p.pizza_type_id
```

**JOIN**

```
order_details od ON od.pizza_id = p.pizza_id
```

**GROUP BY** pt.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  |



# QUE.9 - Calculate the percentage contribution of each pizza type to total revenue

**SELECT**

```
pt.category,
ROUND(SUM(od.quantity * p.price) / (SELECT
 ROUND(SUM(od.quantity * p.price), 2) AS total_sales
FROM
 order_details od
 JOIN
 pizzas p ON p.pizza_id = od.pizza_id) * 100,
2) AS revenue
FROM
pizza_types PT
 JOIN
 pizzas P ON pt.pizza_type_id = p.pizza_type_id
 JOIN
 order_details od ON od.pizza_id = p.pizza_id
GROUP BY pt.category
ORDER BY revenue DESC;
```

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



# QUE.10 - Analyze the cumulative revenue generated over time

```
SELECT order_date , revenue,
SUM(revenue) OVER (ORDER BY order_date) AS cum_revenue
FROM
(SELECT o.order_date AS order_date , ROUND(SUM(od.quantity * p.price),2) AS revenue
FROM order_details od JOIN pizzas p
ON od.pizza_id = p.pizza_id
JOIN orders o
ON o.order_id = od.order_id
GROUP BY order_date
ORDER BY revenue) AS sales
```

|   | order_date | revenue | cum_revenue        |
|---|------------|---------|--------------------|
| ▶ | 2015-01-01 | 2713.85 | 2713.85            |
|   | 2015-01-02 | 2731.9  | 5445.75            |
|   | 2015-01-03 | 2662.4  | 8108.15            |
|   | 2015-01-04 | 1755.45 | 9863.6             |
|   | 2015-01-05 | 2065.95 | 11929.55           |
|   | 2015-01-06 | 2428.95 | 14358.5            |
|   | 2015-01-07 | 2202.2  | 16560.7            |
|   | 2015-01-08 | 2838.35 | 19399.05           |
|   | 2015-01-09 | 2127.35 | 21526.399999999998 |
|   | 2015-01-10 | 2463.95 | 23990.35           |
|   | 2015-01-11 | 1872.3  | 25862.649999999998 |
|   | 2015-01-12 | 1919.05 | 27781.699999999997 |
|   | 2015-01-13 | 2049.6  | 29831.299999999996 |
|   | 2015-01-14 | 2527.4  | 32358.699999999997 |
|   | 2015-01-15 | 1984.8  | 34343.5            |
|   | 2015-01-16 | 2594.15 | 36937.65           |
|   | 2015-01-17 | 2064.1  | 39001.75           |
|   | 2015-01-18 | 1976.85 | 40978.6            |



THANK YOU