SQL

Mysql workbench

Home Dataset Queries

PIZZA SALES ANALYSIS

SHODWE
Pizza Resto

Home About Contact

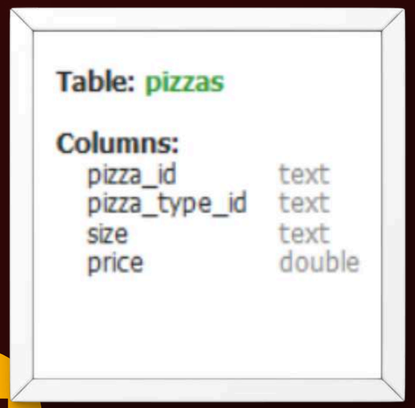
ABOUT ME

I am Divya, and i used SQL to analyze pizza sales data by solving queries related to revenue, top-selling pizza, pizza sales. This project showcases my SQL skills and ability to extract insights from data

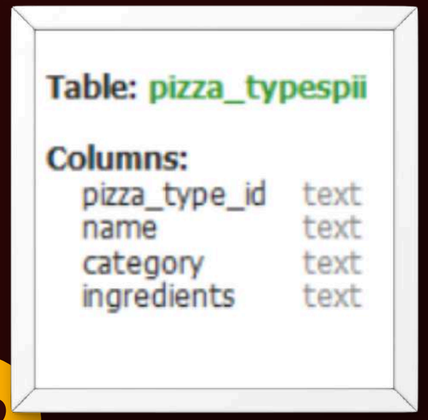
SHODWE
Pizza Resto

Home About Contact

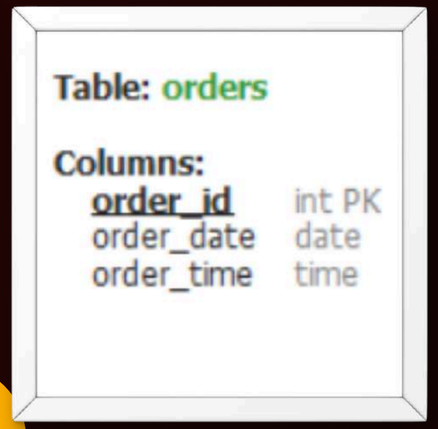
DATASET DETAILS



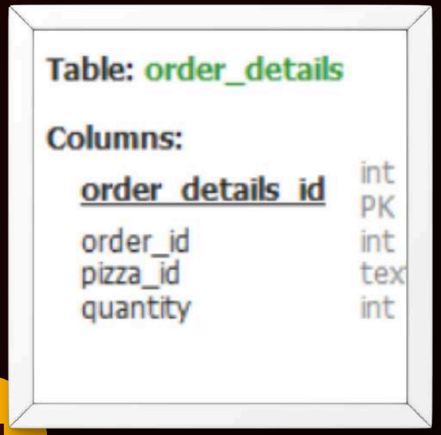
01 PIZZAS



02 PIZZA_TYPESPII



03 ORDER



04 ORDER_DETAILS

SQL
Mysql workbench

Home | Dataset | Queries

RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED

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

Result Grid

	total_orders
▶	2

SQL
Mysql workbench

Home | Dataset | Queries

CALCULATE THE TOTL REVENUE GENERATED FROM PIZZA SALES

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

Result Grid

	total_sales
▶	817860.05

SQL

Mysql workbench

Home | Dataset | Queries

IDENTIFY THE HIGHTEST-PRICED PIZZA.

```
SELECT
    pizza_typespii.name, pizzas.price
FROM
    pizza_typespii
        JOIN
    pizzas ON pizza_typespii.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

SQL
Mysql workbench

Home | Dataset | Queries

IDENTIFY THE MOST COMMOM PIZZA SIZE ORDERED

```sql
USE PIZZAHUT;
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;
```

Result Grid | Filte

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

SQL
Mysql workbench

Home | Dataset | Queries

LIST THE TOP 5 MOST ORDERED PIZZZA_TYPES WITH THEIR QUANTITES

```sql
SELECT
    pizza_typespii.name, SUM(order_details.quantity) AS quantity
FROM
    pizza_typespii
        JOIN
    pizzas ON pizza_typespii.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_typespii.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

SQL

Mysql workbench

Home | Dataset | Queries

TOTAL QUANITY OF EACH PIZZA CATEOGORY ORDERED.

```sql
SELECT
    pizza_typespii.category,
    SUM(order_details.quantity) AS quantity
FROM
    pizza_typespii
        JOIN
    pizzas ON pizza_typespii.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_typespii.category
ORDER BY quantity DESC;
```

Result Grid | Filter

category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

SQL
Mysql workbench

Home | Dataset | Queries

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

SQL
Mysql workbench

Home | Dataset | Queries

JOIN RELEVANT TABLES TO FIND THE CATEGORY-WISE DISTRIBUTION OF PIZZAS

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

Result Grid | Filter Row

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

SQL
Mysql workbench

Home | Dataset | Queries

GROUP THE ORDERS BY DATE AND CALCULATE THE AVERAGE OF NUMBER OF PIZZAS ORDERED PER DAY

```sql
SELECT
    ROUND(AVG(quantity), 0) AS Avg_pizza_ordered_per_day
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 order_date) AS order_quantity
```

Result Grid | Filter Rows:

Avg_pizza_ordered_per_day
138

SQL
Mysql workbench

Home | Dataset | Queries

DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON THE REVENUE

```sql
SELECT
    pizza_typespii.name,
    SUM(order_details.quantity * pizzas.price) AS revenue
FROM
    pizza_typespii
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_typespii.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY 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

SQL
Mysql workbench

Home | Dataset | Queries

CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOATAL REVENUE

```sql
SELECT
    pizza_typespii.category,
    ROUND(SUM(order_details.quantity * price) / (SELECT
                    ROUND(SUM(order_details.quantity * pizzas.price),
                                2) AS total_sales
                FROM
                    order_details
                        JOIN
                    pizzas ON pizzas.pizza_id = order_details.pizza_id) * 100,
            2) AS revenue
FROM
    pizza_typespii
        JOIN
    pizzas ON pizza_typespii.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY category
ORDER BY revenue DESC;
```

Result Grid | Filter Rows:

category	revenue_percentage
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

SQL
Mysql workbench

Home | Dataset | Queries

ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

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

Result Grid | Filter Rows:

order_date	cum_revenue
2015-01-01	2713.8500000000004
2015-01-02	5445.75
2015-01-03	8108.15
2015-01-04	9863.6
2015-01-05	11929.55
2015-01-06	14358.5
2015-01-07	16560.7
2015-01-08	19399.05
2015-01-09	21526.4
2015-01-10	23990.350000000002
2015-01-11	25862.65

SQL

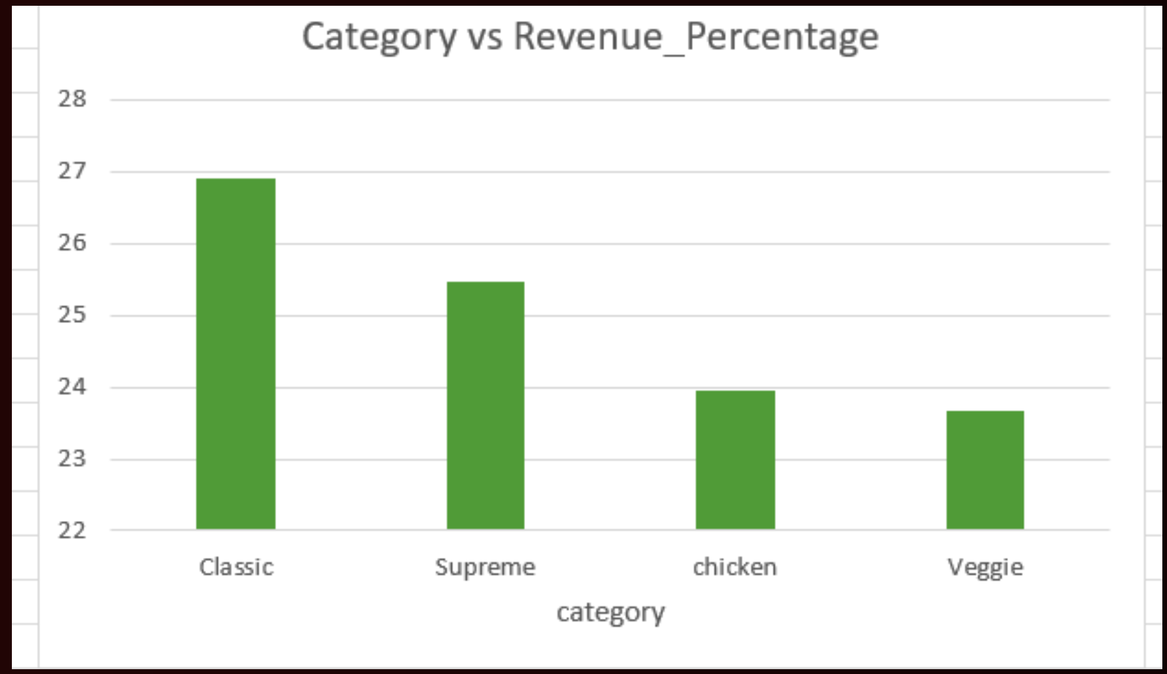
Mysql workbench

Home | Dataset | Queries

SALES REPORT

REVENUE CONTRIBUTION BY PIZZA CATEGORY

This chart shows how much revenue each pizza category generated. It highlights the top-earning categories and helps understand customer preferences



SQL

Mysql workbench

Home | Dataset | Queries

CONCLUSION

- Successful Successfully analyzed pizza sales data using SQL.
- Identified top-selling pizzas, revenue patterns, and category performance.
- Applied key SQL concepts like joins, aggregations, and window functions.
- Improved ability to solve real business problems using data.
- Strengthened overall confidence in SQL and data analysis skills.

SQL
Mysql workbench

Home | Dataset | Queries

THANK YOU

FOR ATTENTION

PIZZA SALES DATA ANALYSIS
PROJECT-DIVYA