SQL PROJECT ON PIZZA SALES

ORDER NOW

Start Your Slide

ABOUT THE PROJECT

Pizza Sales Analysis

This project focuses on analyzing pizza sales data using SQL to derive valuable insights into sales trends, customer preferences, and business performance.

By querying the data, this project aims to identify popular pizza flavors, track sales over time, and optimize revenue management. Through various SQL operations, the project demonstrates the power of database management and query techniques to inform strategic business decisions in the pizza industry.

DATABASE : PIZZAHUT

Short Info About Database

A database in SQL is an organized collection of structured data stored in tables, which can be easily accessed, managed, and updated using SQL (Structured Query Language). It allows for the efficient storage, retrieval, and manipulation of data in a systematic way, ensuring data integrity, security, and scalability.

The database PIZZAHUT contains tables with information on customer orders, date, time, pizza types, sales transactions, quantity, ingredients and much more.

Q1. RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

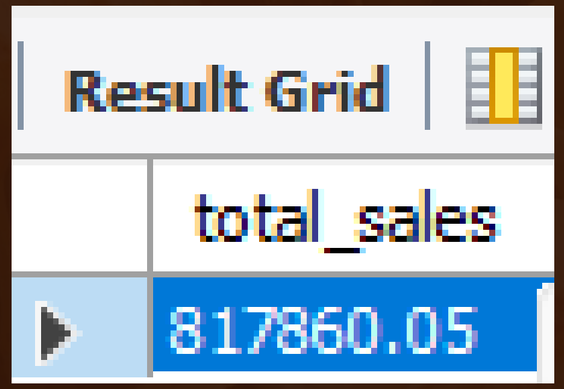
```sql
SELECT
    COUNT(order_id) AS total_orders
FROM
    orders;
```

Result Grid

	total_orders
▶	21350

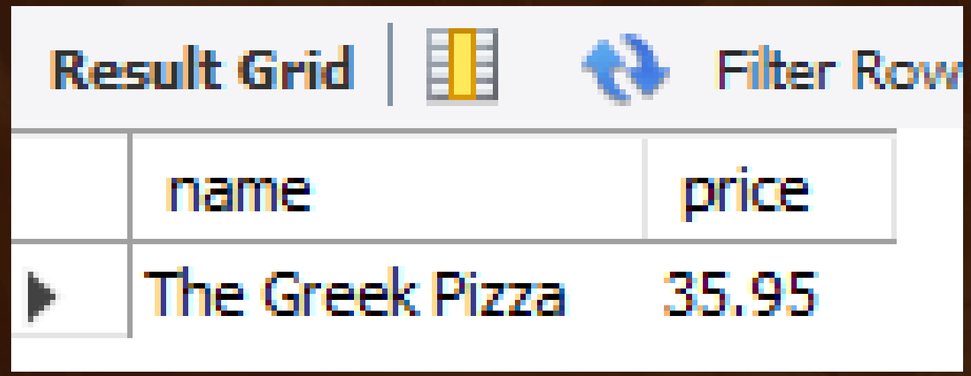
Q2. CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

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



Q3. IDENTIFY THE HIGHEST-PRICED 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

Q4. IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

```sql
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;
```

Result Grid

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

Q5. LIST THE TOP 5 MOST ORDERED PIZZA TYPES ALONG WITH THEIR QUANTITIES.

```sql
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;
```

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

Q6. JOIN THE NECESSARY TABLES TO FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED.

```sql
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;
```

Result Grid | Filte

category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

Q7. 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(order_time);
```

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

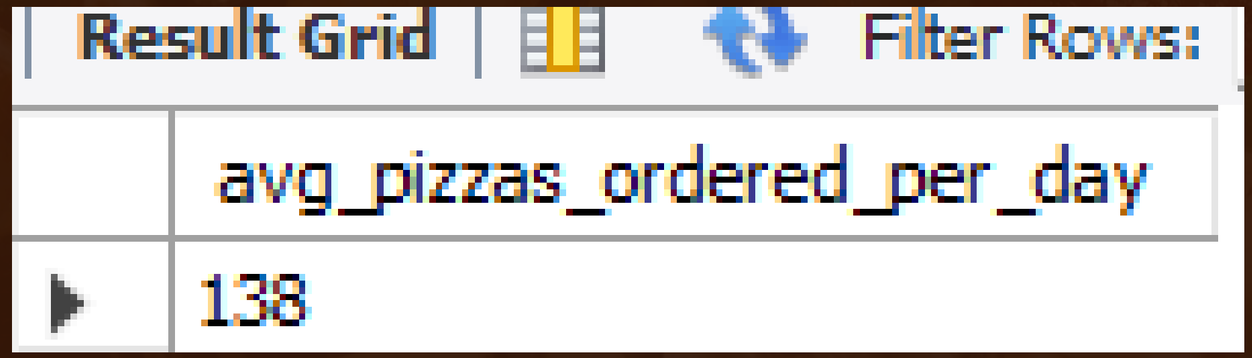
Q8. JOIN RELEVANT TABLES TO FIND THE CATEGORY-WISE DISTRIBUTION OF PIZZAS.

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

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

Q9. 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;
```



Q10. 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;
```

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

Q11. ANALYZE THE REVENUE GENERATED OVER TIME.

```sql
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;
```

order_date	revenue
2015-01-01	2713.8500000000004
2015-01-02	2731.8999999999996
2015-01-03	2662.3999999999996
2015-01-04	1755.4500000000003
2015-01-05	2065.95
2015-01-06	2428.95
2015-01-07	2202.2000000000003
2015-01-08	2838.3499999999995
2015-01-09	2127.3500000000004
2015-01-10	2463.95
2015-01-11	1872.3000000000002
2015-01-12	1919.0500000000002
2015-01-13	2049.6000000000004
2015-01-14	2527.3999999999996
2015-01-15	1984.8000000000002
2015-01-16	2594.15
2015-01-17	2064.1000000000004
2015-01-18	1976.8500000000001
2015-01-19	2387.1499999999996
2015-01-20	2397.9000000000005
2015-01-21	2040.5500000000002
2015-01-22	2496.7000000000003
2015-01-23	2423.7
2015-01-24	2289.25
2015-01-25	1617.5500000000002
2015-01-26	1884.4
2015-01-27	2528.0499999999997
2015-01-28	2016
2015-01-29	2045.3000000000002
2015-01-30	2270.3
2015-01-31	2417.85

and so on.....

Q12. CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE.

```sql
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 pizzas.pizza_id = orders_details.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;
```

category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

and so on.....

CONCLUSION

- In year 2015, Pizza hut placed a total of 21350 orders.
- An estimated total sales of 817,860 dollars.
- The classic deluxe pizza was the most ordered pizza in the year 2015.
- Noon and the evening hours were the most rush hours in the day.
- The avg no of pizzas ordered were 138.

Thank You.