Pizza sales Analysis

Each slice offers a taste of pure perfection.

An SQL project by Vishal Raj.

Objetives:

This project is to analyze pizza sales data using SQL to uncover valuable insights into sales trends, customer preferences, and revenue patterns. By utilizing SQL queries, the project aims to extract meaningful information related to top-selling pizzas, peak sales periods, customer ordering behavior, and overall business performance. This analysis will help in optimizing inventory management, improving marketing strategies, and enhancing decision-making for better profitability in the pizza business.

Dataset Overview

Table: pizza_types

Columns:

pizza_type_id	text
name	text
category	text
ingredients	text

Table: order_details

Columns:

order_details_id	int PK
order_id	int
pizza_id	te:
quantity	int

Table: pizzas

Columns:

pizza_id	text
pizza_type_id	text
size	text
price	double

Table: orders

Columns:

orders_id	int PK
orders_date	date
orders_time	time

Q1. Find the total number of order placed.

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

Result Grid

	total_orders
▶	21350

Q2. Find the total revenue generated from pizza sales.

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

Result Grid

	total_sales
▶	817860.05

Q3. Identify the hightest 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;
```

Result Grid | Filter Rows:

	name	price
▶	The Greek Pizza	35.95

Q4. Find the most common pizza size ordered.

```sql
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;
```

Result Grid | Filt

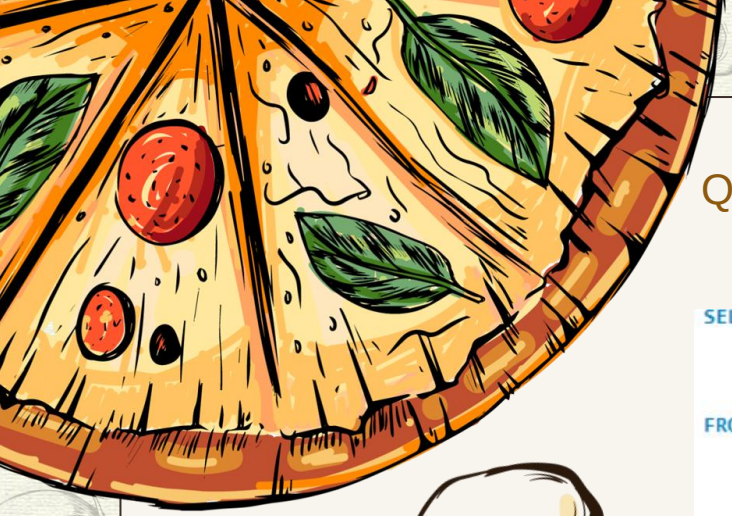
	size	order_count
▶	L	18526

Q5. Find the top 5 most ordered pizza types along with their quantities.

```sql
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;
```

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. Find the total quantity of each pizza category ordered.

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

Result Grid | Filter Row

total_quantity	category
14888	Classic
11987	Supreme
11649	Veggie
11050	Chicken

Q7. Determine the distribution of orders by hour of the day.

```
SELECT
    HOUR(orders_time) AS hour, COUNT(orders_id) AS order_count
FROM
    orders
GROUP BY HOUR(orders_time) order by hour asc;
```

Result Grid | Filter

hour	order_count
9	1
10	8
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

Q8. Find the category-wise distribution of pizza.

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

Result Grid | Filter Rows

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), 2) as avg_pizza_ordered_per_day
FROM
    (SELECT
        orders.orders_date, SUM(order_details.quantity) AS quantity
    FROM
        orders
    JOIN order_details ON orders.orders_id = order_details.order_id
    GROUP BY orders.orders_date) AS order_quantity;
```

Result Grid | Filter Rows:

avg_pizza_ordered_per_day
138.47

Q10. Find the top 3 most ordered pizza types based on revenue.

```
SELECT
    pizza_types.name,
    ROUND(SUM(pizzas.price * order_details.quantity),
            2) AS revenue
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 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

Q11. Calculate the percentage contribution of each pizza type to total revenue.

```sql
select pizza_types.category,round(sum(pizzas.price * order_details.quantity)/(select
    ROUND(SUM(order_details.quantity * pizzas.price),
            2) AS total_sales
FROM
    order_details
        JOIN
    pizzas ON order_details.pizza_id = pizzas.pizza_id)*100,2) as total_revenue
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 total_revenue desc;
```

Result Grid | Filter Rows:

category	total_revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

Q12. Analyze the cumulative revenue generated over time.

```
select orders_date, sum(revenue) over(order by orders_date) as cum_revenue
from
(select orders.orders_date, sum(pizzas.price * order_details.quantity) as revenue
from orders join order_details
on orders.orders_id= order_details.order_id
join pizzas on pizzas.pizza_id= order_details.pizza_id
group by orders.orders_date) as sales ;
```

Result Grid | Filter Rows:

orders_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

Q13. Determine the top 3 most ordered pizza types based on revenue for each pizza category.

```sql
select name, category, revenue
from (select category, name, revenue, rank() over (partition by category order by revenue desc) as rn
from
(select pizza_types.name, pizza_types.category,sum(pizzas.price * order_details.quantity) as revenue
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,pizza_types.category) as a) as b
where rn<=3;
```

Result Grid | Filter Rows: | Export:

name	category	revenue
The Thai Chicken Pizza	Chicken	43434.25
The Barbecue Chicken Pizza	Chicken	42768
The California Chicken Pizza	Chicken	41409.5
The Classic Deluxe Pizza	Classic	38180.5
The Hawaiian Pizza	Classic	32273.25
The Pepperoni Pizza	Classic	30161.75
The Spicy Italian Pizza	Supreme	34831.25
The Italian Supreme Pizza	Supreme	33476.75
The Sicilian Pizza	Supreme	30940.5
The Four Cheese Pizza	Veggie	32265.70000000065
The Mexicana Pizza	Veggie	26780.75
The Five Cheese Pizza	Veggie	26066.5

Key Insights:

- Peak Sales Days: The highest number of orders occur on Fridays and Saturdays.
- Top Performing Months: Sales peak in January and July.
- Best-Selling Category: Classic pizzas dominate both sales and order numbers.
- Most Profitable Size: Large pizzas contribute the most revenue.
- Top Revenue-Generating Pizza: The Thai Chicken Pizza brings in the highest earnings.
- Lowest-Selling Pizza: The Brie Carre has the least number of sales and orders.
- Most Ordered Pizza: The Classic Deluxe Pizza has the highest order volume.

THANK YOU FOR YOUR ATTENTION