A Structured Query Language (SQL)-Based Data Exploration

PIZZA SALES - SQL ANALYSIS REPORT

BY MAYANK

Aspiring Data Analyst

INTRODUCTION

PROJECT OVEVIEW

This project focuses on analyzing pizza sales using SQL to uncover patterns, trends, and actionable insights. It involves writing and executing queries on multiple related tables to answer real-world business questions.

DATASET OVERVIEW

PIZZAS: CONTAINS PIZZA SIZE AND PRICE DETAILS

PIZZA_TYPES : DESCRIBES THE TYPE AND CATEGORY OF PIZZAS

ORDER_DETAILS : RECORDS QUANTITY AND PIZZA ORDERED PER ORDER

ORDERS : PROVIDES ORDER DATE AND TIME

RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED

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

Result Grid

	TOTAL_ORDERS
▶	4314

CALCULATE THE TOTAL 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
581681.65

IDENTIFY THE HIGHEST PRICES PIZZA.

```
select pizza_types.name , pizzas.price as expensive_pizza
from pizza_types join pizzas on pizza_types.pizza_type_id
= pizzas.pizza_type_id
order by expensive_pizza desc limit 1;
```

Result Grid | Filter Rows:

	name	expensive_pizza
▶	The Greek Pizza	35.95

IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

```sql
SELECT
    pizzas.size,
    COUNT(order_details.order_details_id) AS total_order_detail_id
FROM
    pizzas
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizzas.size;
```

size	total_order_detail_
M	10957
L	13180
S	10028
XL	400

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

```sql
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 pizzas.pizza_id = order_details.pizza_id
GROUP BY pizza_types.category;
```

Result Grid | Filter

category	quantity
Classic	10557
Veggie	8372
Supreme	8503
Chicken	7816

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

```sql
SELECT
    pizza_types.category AS 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 category
ORDER BY quantity DESC;
```

category	quantity
Classic	10557
Supreme	8503
Veggie	8372
Chicken	7816

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

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

Result Grid

	category	name
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9

GROUP THE ORDERS BY DATE AND CALCULATE THE AVERAGE NUMBER OF PIZZAS PE DAY.

```sql
SELECT AVG(quantity_per_day) AS avg_pizzas_per_day
FROM (
    SELECT
        orders.date,
        SUM(order_details.quantity) AS quantity_per_day
    FROM orders
    JOIN order_details
        ON orders.order_id = order_details.order_id
    GROUP BY orders.date
) AS daily_totals;
```

Result Grid | Filte

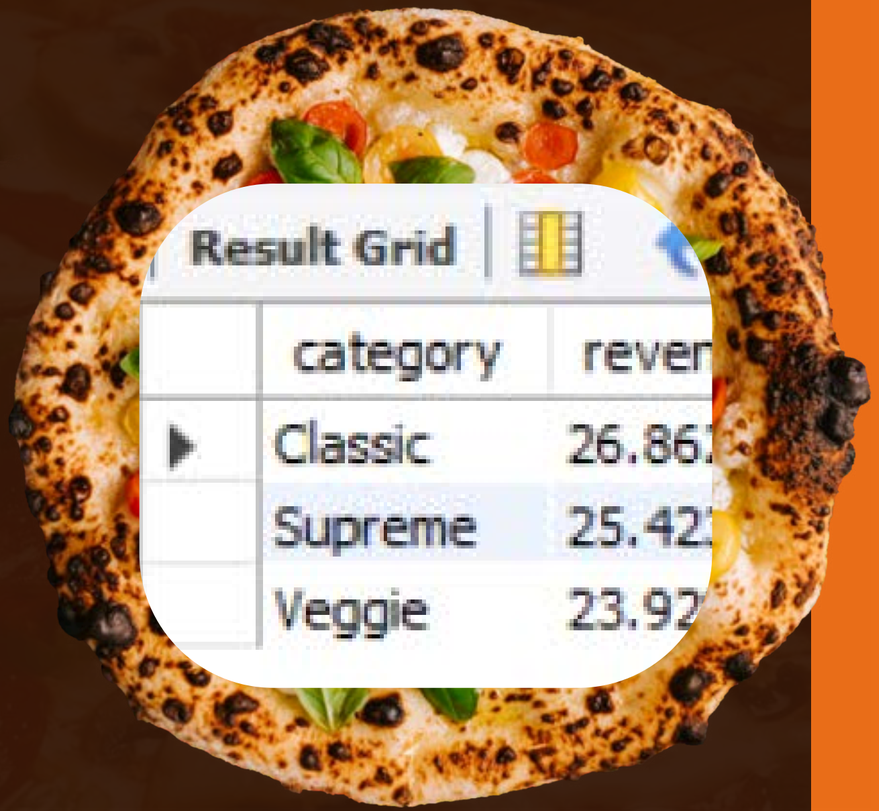
avg_pizzas_per_day
28.3889

DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE

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

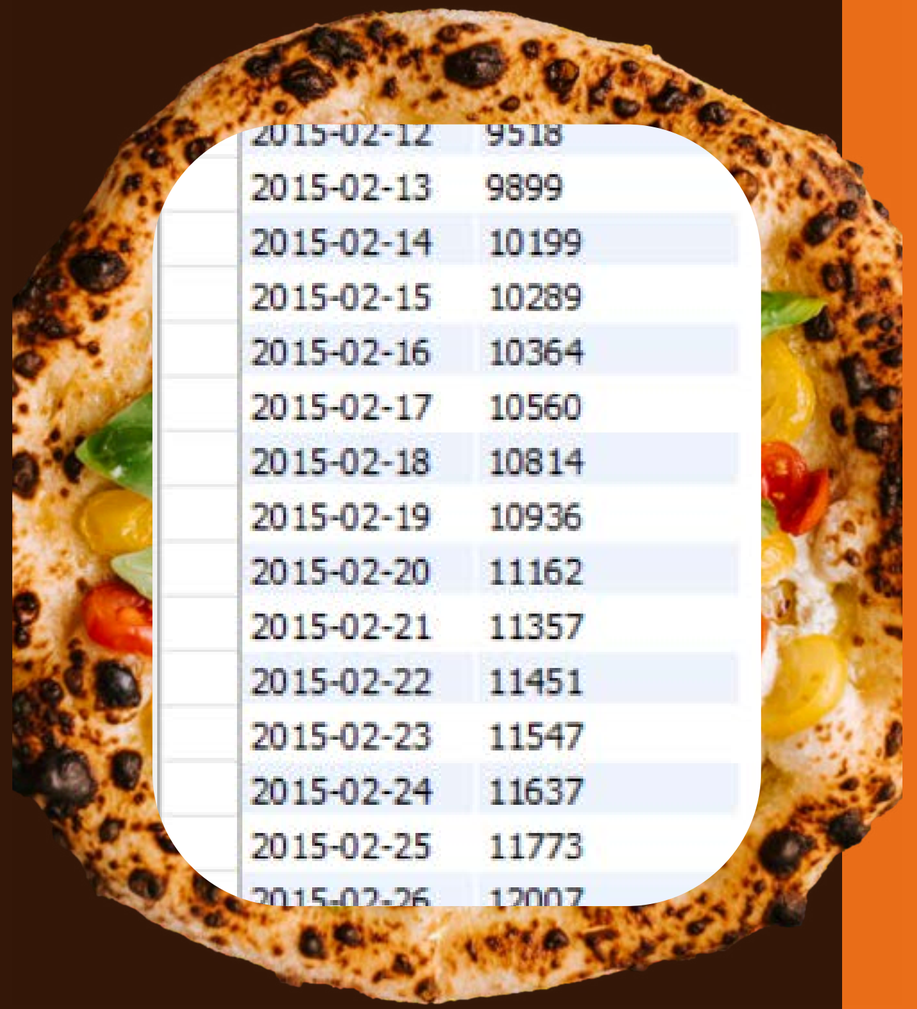
CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE.

```sql
SELECT pizza_types.category ,
    (SUM(order_details.quantity * pizzas.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 as revenue
FROM pizza_types join
        pizzas  on pizza_types.pizza_type_id = pizzas.pizza_type_id jo
        order_details
on pizzas.pizza_id = order_details.pizza_id
group by pizza_types.category order by revenue desc limit 3 ;
```



ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME

```sql
select date , sum(revenue) over (order by date) as cum_revenue
from
(Select orders.date,
        ROUND(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_details_id
group by orders.date) as sales;
```



2015-02-12	9518
2015-02-13	9899
2015-02-14	10199
2015-02-15	10289
2015-02-16	10364
2015-02-17	10560
2015-02-18	10814
2015-02-19	10936
2015-02-20	11162
2015-02-21	11357
2015-02-22	11451
2015-02-23	11547
2015-02-24	11637
2015-02-25	11773
2015-02-26	12007

DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE FOR EACH PIZZA CATEGORY.

```sql
SELECT
    category,
    name,
    revenue,
    rn
FROM (
    SELECT
        pizza_types.category,
        pizza_types.name,
        SUM(order_details.quantity * pizzas.price) AS revenue,
        RANK() OVER (PARTITION BY pizza_types.category ORDER BY SUM(order_details.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, pizza_types.name
) AS ranked_pizzas
WHERE rn <= 3;
```

esult Grid | Filter Rows: | Export: | Wra

category	name	revenue	rn
Chicken	The Barbecue Chicken Pizza	30841.75	1
Chicken	The Thai Chicken Pizza	30241.25	2
Chicken	The California Chicken Pizza	29398.25	3
Classic	The Classic Deluxe Pizza	26653.5	1
Classic	The Hawaiian Pizza	22562.25	2
Classic	The Pepperoni Pizza	21575	3
Supreme	The Spicy Italian Pizza	25034.25	1
Supreme	The Italian Supreme Pizza	24057.75	2
Supreme	The Sicilian Pizza	21550.5	3
Veggie	The Four Cheese Pizza	22805.150000000365	1
Veggie	The Five Cheese Pizza	19110.5	2
Veggie	The Mexicana Pizza	18787.75	3

THANK YOU

Thank you for reviewing my project! I am excited to continue learning and growing as a data analyst.

LINKEDLN https://www.linkedin.com/in/mayank-goyal-4b8756363/

GITHUB https://github.com/mayank-goyal09

EMAIL itsmaygal09@gmail.com