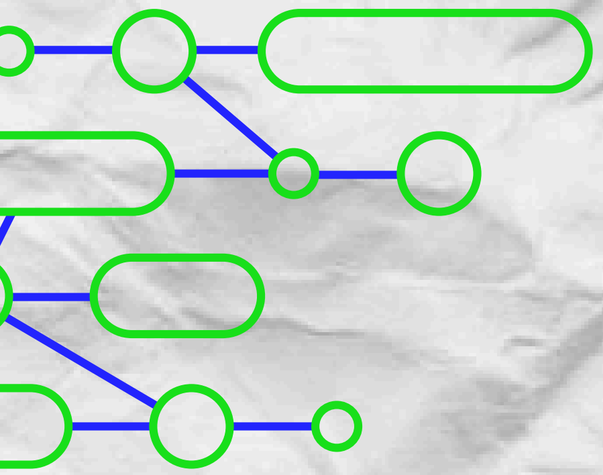
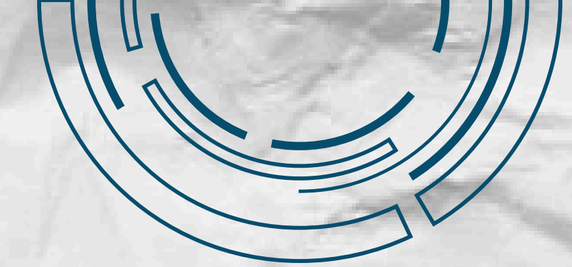
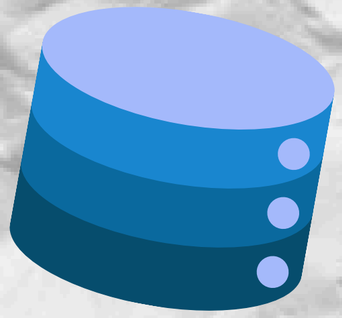
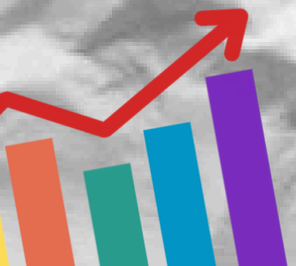
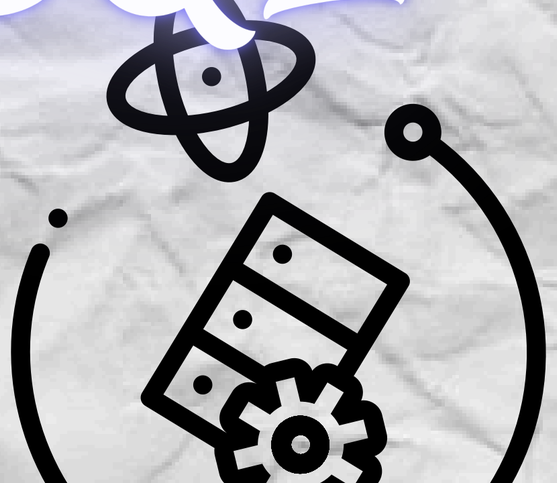




Data Analytics with SQL







Problem Statement

- The goal of this project was to conduct an in-depth analysis of a pizza sales dataset to provide valuable business insights and support data-driven decision-making processes.
- The project involved querying and manipulating the data using SQL to extract, analyze, and visualize various aspects of the pizza sales.
- The analysis was divided into three categories: Basic, Intermediate, and Advanced, each with specific objectives to be achieved.

The total number of orders placed

Input

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

Output

COUNT(order_id)
21350

The total revenue generated from pizza sales.

Input

```sql
SELECT
    SUM(pizzas.price * orders_detail.quantity) AS Revenue
FROM
    pizzas
        JOIN
    orders_detail ON orders_detail.pizza_id = pizzas.pizza_id;
```

Output

Result Grid

	Revenue
▶	817860.049999993

The highest priced pizza

Input

```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;
```

Output

name	price
The Greek Pizza	35.95

Most common Pizza size ordered

Input

```sql
SELECT
    pizzas.size, COUNT(orders_detail.quantity) AS Quantity
FROM
    pizzas
        JOIN
    orders_detail ON pizzas.pizza_id = orders_detail.pizza_id
GROUP BY pizzas.size
ORDER BY Quantity DESC
LIMIT 1;
```

Output

size	Quantity
L	18526

Top 5 most ordered Pizza type along with their quantity

Input

```
    pizza_types.name, SUM(orders_detail.quantity) AS quantity
FROM
    pizzas
        JOIN
    pizza_types ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    orders_detail ON pizzas.pizza_id = orders_detail.pizza_id
GROUP BY pizza_types.name
ORDER BY quantity DESC
LIMIT 5;
```

Output

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

Count of Pizza ordered according to Quantity

Input

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

Output

category	total_quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

Distribution of order by hour of the day

Input

```sql
SELECT
    HOUR(Order_time), COUNT(order_id)
FROM
    orders
GROUP BY HOUR(Order_time);
```

Output

HOUR(Order_time)	COUNT(order_id)
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

Average order per day

Input

```sql
SELECT
    ROUND(AVG(Quantity), 0) AS Average_Order_Perday
FROM
    (SELECT
        SUM(orders_detail.quantity) AS Quantity, orders.Order_date
    FROM
        orders
    JOIN orders_detail ON orders_detail.order_id = orders.order_id
    GROUP BY Order_date) AS Total_Sales
```

Output

Average_Order_Perday
138

Category wise distribution of Pizza

Input

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

Output

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

Top 3 Pizza by Revenue

Input

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

Output

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

Percentage wise contribution of Pizza Type in Revenue

Input

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

Output

category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

Cumulative Revenue generated overtime

Input

```
Select order_date,
sum(revenue) over (order by order_date) as cum_revenue
from
(Select orders.Order_date,
sum(orders_detail.quantity * pizzas.price) AS Revenue
from orders_detail join pizzas on orders_detail.pizza_id = pizzas.pizza_id
join orders on orders_detail.order_id = orders.order_id
group by orders.Order_date) AS sales;
```

Output

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

Top 3 most ordered pizza based on revenue

Input

```sql
select category, name, revenue from
(select category, name, revenue,
rank() over (partition by category order by revenue desc) AS rn
from
(SELECT
    pizza_types.category,
    pizza_types.name,
    SUM(orders_detail.quantity * pizzas.price) AS Revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    orders_detail ON orders_detail.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category , pizza_types.name) as a ) as b
where rn <=3;
```

Output

category	name	revenue
Chicken	The Thai Chicken Pizza	43434.25
Chicken	The Barbecue Chicken Pizza	42768
Chicken	The California Chicken Pizza	41409.5
Classic	The Classic Deluxe Pizza	38180.5
Classic	The Hawaiian Pizza	32273.25
Classic	The Pepperoni Pizza	30161.75
Supreme	The Spicy Italian Pizza	34831.25

Thanks !