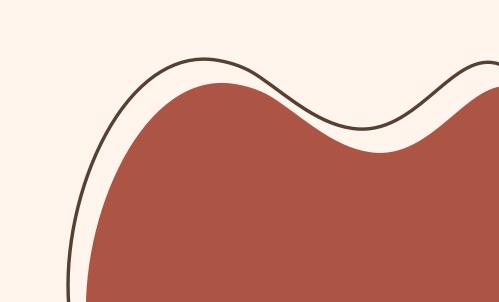
Pizza sales SQL Project

Presented by Dua Hussain







Preface



This project focuses on pizza sales analysis, showcased on a SQL platform. It demonstrates the utilization of aggregate functions, joins, Common Table Expressions (CTEs), and more to extract insights from the pizza database.

Retreving Total No of Order Placed

```
SELECT

COUNT(*) AS total_no_of_orders

FROM

orders;
```

	total_no_of_orders
>	21350

Calculate the total revenue generated from pizza sales.

```
SELECT

ROUND(SUM(p.price * o.quantity), 2) AS revenue_genrated

FROM

pizzas AS p

LEFT JOIN

order_details AS o USING (pizza_id);
```

	revenue_genrated
•	817860.05

Identify the highest-priced pizza.

```
SELECT

pt.name, p.price

FROM

pizzas AS p

JOIN

pizza_types AS pt USING (pizza_type_id)

ORDER BY p.price DESC

LIMIT 1;
```

	name	price
-	The Greek Pizza	35.95

Identify the most common pizza size ordered

```
p.size,
    COUNT(o.order_details_id) AS count_of_pizza_size_wise

FROM
    pizzas AS p
        LEFT JOIN
    order_details AS o USING (pizza_id)

GROUP BY p.size

ORDER BY count_of_pizza_size_wise DESC

LIMIT 1;
```

	size	count_of_pizza_size_wise
>	L	18526

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

```
SELECT
    pt.`name`, SUM(o.quantity) total_quantity
FROM
    pizzas AS p
        JOIN
    pizza_types AS pt USING (pizza_type_id)
        JOIN
    order_details AS o USING (pizza_id)
GROUP BY pt.`name`
DRDER BY total_quantity DESC
LIMIT 5;
```

	2453
The Parheeue Chicken Diago	
The barbecue Chicken Pizza 2	2432
The Hawaiian Pizza 2	2422
The Pepperoni Pizza 2	2418
The Thai Chicken Pizza 2	2371

Join the necessary tables to find the total quantity of each pizza category ordered

```
SELECT

SUM(o.quantity) total_quantity, category

FROM

pizza_types AS pt

LEFT JOIN

pizzas AS p USING (pizza_type_id)

LEFT JOIN

order_details AS o USING (pizza_id)

GROUP BY category

ORDER BY total_quantity DESC;
```

	total_quantity	category
•	14888	Classic
	11987	Supreme
	11649	Veggie
	11050	Chicken

Determine the distribution of orders by hour of the day.

```
SELECT

HOUR(time) AS hours, COUNT(order_id) AS Total_no_of_orders

FROM

orders

GROUP BY hours

ORDER BY Total_no_of_orders DESC;
```

hours	Total_no_of_orders
12	2520
13	2455
18	2399
17	2336
19	2009
16	1920
20	1642
14	1472
15	1468
11	1231
21	1198
22	663
23	28
10	8
9	1

Join relevant tables to find the category-wise distribution of pizzas.

```
SELECT

category, COUNT(category)

FROM

pizza_types

GROUP BY category;
```

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

Group the orders by date and calculate the average number of pizzas ordered per day.

```
SELECT

AVG(total_count_of_order_per_day)

FROM

(SELECT

date, COUNT(quantity) AS total_count_of_order_per_day

FROM

orders

JOIN order_details AS o USING (order_id)

GROUP BY date) avg_order_per_day;

-- the above is common table expression
```

```
avg(total_count_of_order_per_day)
135.8101
```

Determine the top 3 most ordered pizza types based on revenue.

```
p.pizza_type_id, SUM((p.price * o.quantity)) AS total_price

FROM

pizza_types AS pt

LEFT JOIN

pizzas AS p USING (pizza_type_id)

LEFT JOIN

order_details AS o USING (pizza_id)

GROUP BY p.pizza_type_id

ORDER BY total_price DESC

LIMIT 3;
```

	pizza_type_id	total_price
•	thai_ckn	43434.25
	bbq_ckn	42768
	cali_ckn	41409.5

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

```
SELECT
   category,
   ROUND((SUM(p.price * o.quantity) / (SELECT
                    SUM(p.price * o.quantity)
                FROM
                    order_details AS o
                        LEFT JOIN
                    pizzas AS p USING (pizza_id))) * 100,
            2) AS percentage of total category wise
FROM
   pizza_types A5 pt
       LEFT JOIN
   pizzas AS p USING (pizza_type_id)
       LEFT JOIN
   order_details AS o USING (pizza_id)
GROUP BY category;
```

category	percentage_of_total_category_wise
Chicken	23.96
Classic	26.91
Supreme	25.46
Veggie	23.68

THANK YOU SO MUCH!