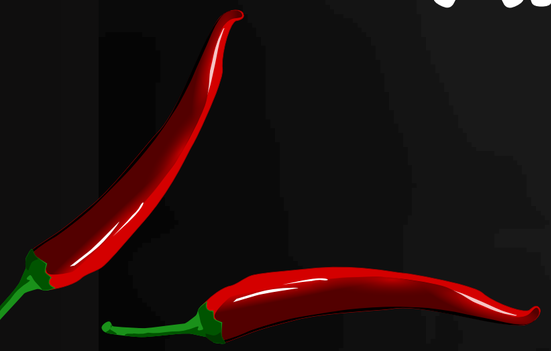
ALSO USED ADV SQL FOR QUERIES

SQL PROJECT

PIZZA SALES ANALYSIS



MySQL

FRIENDS, IN THIS PROJECT I HAD SOLVED 15 REQUIREMENTS

Which involves many SQL concepts

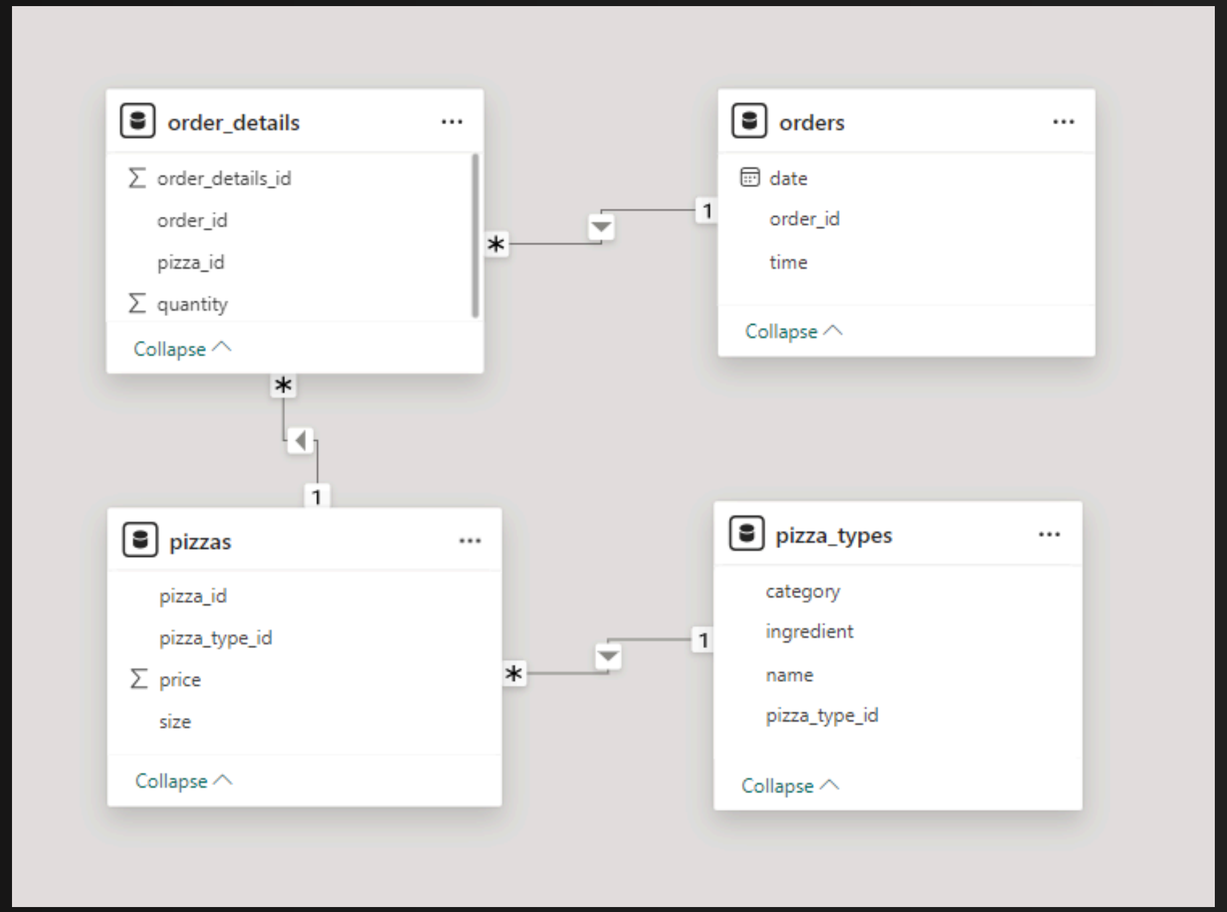
- **BASIC**
- **INTERMEDIATE**
- **ADVANCE**

WE HAVE **4** TABLES IN PROJECT

ER DIAGRAM

4 TABLES

- ORDERS
- ORDER_DETAILS
- PIZZAS
- PIZZA_TYPES



REQUIREMENT 1 RETRIVE THE TOTAL NO OF ORDER PLACED

```sql
SELECT
    COUNT(ORDER_ID) AS "TOTAL_ORDERS"
FROM
    ORDERS;
```

Result Grid | Filter Rov

	TOTAL_ORDERS
▶	21350

REQUIREMENT 2

Calculate the total revenue generated from pizza sales.

```
SELECT ROUND(SUM(O.QUANTITY * P.PRICE),2) AS TOTAL_REVENUE
FROM ORDER_DETAILS O
JOIN PIZZAS P
ON O.PIZZA_ID = P.PIZZA_ID;
```

TOTAL_REVENUE
817860.05

REQUIREMENT 3 Identify the highest-priced pizza.

```
SELECT PT.NAME, P.PRICE
FROM PIZZA_TYPES PT
JOIN PIZZAS P
ON PT.PIZZA_TYPE_ID = P.PIZZA_TYPE_ID
ORDER BY PRICE DESC
LIMIT 1;
```

	NAME	PRICE
▶	The Greek Pizza	35.95

REQUIREMENT 4

Identify the most common pizza size ordered.

```sql
SELECT P.SIZE, COUNT(*) AS ORDER_COUNT
FROM ORDER_DETAILS O
JOIN PIZZAS P
ON O.PIZZA_ID = P.PIZZA_ID
GROUP BY P.SIZE
ORDER BY ORDER_COUNT DESC
LIMIT 1;
```

SIZE	ORDER_COUNT
L	18526

REQUIREMENT 5

List 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;
```

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

REQUIREMENT 6

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

```
SELECT PT.CATEGORY, SUM(O.QUANTITY) QUANTITY FROM PIZZA_TYPES PT
JOIN PIZZAS P ON P.PIZZA_TYPE_ID = PT.PIZZA_TYPE_ID
JOIN ORDER_DETAILS O
ON O.PIZZA_ID = P.PIZZA_ID
GROUP BY CATEGORY
ORDER BY QUANTITY DESC;
```

CATEGORY	QUANTITY
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

REQUIREMENT 7

Determine the distribution of orders by hour of the day.

```
SELECT HOUR(ORDER_TIME) , COUNT(ORDER_ID) FROM
ORDERS
GROUP BY HOUR(ORDER_TIME);
```

H	COUNT(ORDER_ID)
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336
18	2399
19	2009
20	1642

REQUIREMENT 8

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

REQUIREMENT 9

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

```
SELECT ROUND(AVG(QUANTITY)) AVG_DAILY_QUANTITY FROM
(SELECT ORDERS.ORDER_DATE,
SUM(ORDER_DETAILS.QUANTITY)  QUANTITY FROM ORDERS
JOIN ORDER_DETAILS
ON ORDERS.ORDER_ID = ORDER_DETAILS.ORDER_ID
GROUP BY ORDERS.ORDER_DATE) AS ORDER_QUANTITY;
```

AVG_DAILY_QUANTITY
138

REQUIREMENT 10

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

```sql
SELECT PIZZA_TYPES.NAME, SUM(ORDER_DETAILS.QUANTITY * PIZZAS.PRICE) AS REVENUE
FROM PIZZA_TYPES JOIN PIZZAS
ON PIZZAS.PIZZA_TYPE_ID = PIZZA_TYPES.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
```

NAME	REVENUE
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5

REQUIREMENT 11

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

```
SELECT PT.CATEGORY,ROUND(SUM(O.QUANTITY * P.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) * 100,0) REVENUE
FROM PIZZA_TYPES PT
JOIN PIZZAS P ON PT.PIZZA_TYPE_ID = P.PIZZA_TYPE_ID
JOIN ORDER_DETAILS O ON O.PIZZA_ID = P.PIZZA_ID
GROUP BY CATEGORY
ORDER BY REVENUE DESC;
```

CATEGORY	REVENUE
Classic	27
Supreme	25
Veggie	24
Chicken	24

REQUIREMENT 12

Analyze the cumulative revenue generated over time.

```sql
SELECT ORDER_DATE, SUM(REVENUE) OVER(ORDER BY ORDER_DATE) AS CUM_REVENUE
FROM
(SELECT O.ORDER_DATE, SUM(OD.QUANTITY * P.PRICE) AS REVENUE
FROM ORDERS O JOIN ORDER_DETAILS OD
ON O.ORDER_ID = OD.ORDER_ID
JOIN PIZZAS P ON
P.PIZZA_ID = OD.PIZZA_ID
GROUP BY ORDER_DATE) AS SALES;
```

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
2015-01-08	19399.05

REQUIREMENT 13

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

```
SELECT NAME, REVENUE, RN FROM

(SELECT CATEGORY, NAME, REVENUE, RANK() OVER(PARTITION BY CATEGORY ORDER BY REVENUE DESC) AS RN
FROM
(SELECT PT.CATEGORY, PT.NAME, SUM((OD.QUANTITY) * P.PRICE) REVENUE
FROM PIZZA_TYPES PT JOIN
PIZZAS P
ON PT.PIZZA_TYPE_ID = P.PIZZA_TYPE_ID
JOIN ORDER_DETAILS OD
ON OD.PIZZA_ID = P.PIZZA_ID
GROUP BY PT.CATEGORY , PT.NAME) T1) T2
WHERE RN <= 3;
```

NAME	REVENUE	RN
The Thai Chicken Pizza	43434.25	1
The Barbecue Chicken Pizza	42768	2
The California Chicken Pizza	41409.5	3
The Classic Deluxe Pizza	38180.5	1
The Hawaiian Pizza	32273.25	2
The Pepperoni Pizza	30161.75	3
The Spicy Italian Pizza	34831.25	1
The Italian Supreme Pizza	33476.75	2
The Sicilian Pizza	30940.5	3
The Four Cheese Pizza	32265.70000000065	1

REQUIREMENT 14

Determine the revenue contribution of each pizza size (e.g., Small, Medium, Large) as a percentage of total revenue.

```
SELECT
    P.SIZE AS PIZZA_SIZE,
    ROUND(SUM(O.QUANTITY * P.PRICE) /
    (SELECT SUM(OD.QUANTITY * PZ.PRICE)
    FROM Order_Details OD
    JOIN Pizzas PZ
    ON OD.PIZZA_ID = PZ.PIZZA_ID) * 100, 2) AS REVENUE_PERCENTAGE
FROM
    Order_Details O JOIN Pizzas P ON O.PIZZA_ID = P.PIZZA_ID GROUP BY P.SIZE ORDER BY
    REVENUE_PERCENTAGE DESC;
```

PIZZA_SIZE	REVENUE_PERCENTAGE
L	45.89
M	30.49
S	21.77
XL	1.72
XXL	0.12

REQUIREMENT 15

Identify the change in total revenue from one order date to the next using the LEAD function.

```sql
SELECT
    O.ORDER_DATE,
    ROUND(SUM(OD.QUANTITY * P.PRICE), 2) AS DAILY_REVENUE,
    ROUND(LEAD(SUM(OD.QUANTITY * P.PRICE)) OVER (ORDER BY O.ORDER_DATE), 2) AS NEXT_DAY_REVENUE,
    ROUND(LEAD(SUM(OD.QUANTITY * P.PRICE)) OVER (ORDER BY O.ORDER_DATE) - SUM(OD.QUANTITY * P.PRICE), 2) AS REVENUE_CHANGE
FROM Orders O
JOIN Order_Details OD ON O.ORDER_ID = OD.ORDER_ID
JOIN Pizzas P ON OD.PIZZA_ID = P.PIZZA_ID
GROUP BY O.ORDER_DATE
ORDER BY O.ORDER_DATE;
```

ORDER_DATE	DAILY_REVENUE	NEXT_DAY_REVENUE	REVENUE_CHANGE
2015-01-01	2713.85	2731.9	18.05
2015-01-02	2731.9	2662.4	-69.5
2015-01-03	2662.4	1755.45	-906.95
2015-01-04	1755.45	2065.95	310.5
2015-01-05	2065.95	2428.95	363
2015-01-06	2428.95	2202.2	-226.75
2015-01-07	2202.2	2838.35	636.15
2015-01-08	2838.35	2127.35	-711

Insights -

BEST SELLER

THE CLASSIX DILUX PIZZA

Insights -

- we get to know that total orderes placed are **21350**
- Total revenue = **817860 USD**
- Highest Priced pizza - The Greek Pizza
- Most common size ordered - LARGE
- Top 5 pizzas -
  - The Classic Deluxe Pizza
  - The Barbecue Chicken Pizza
  - The Hawaiian Pizza
  - The Pepperoni Pizza
  - The Thai Chicken Pizza

Insights -

- TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED
  - Classic - 14888
  - Supreme - 11987
  - Veggie - 11649
  - Chicken - 11050

- Busy Hours : 12- 1 pm and 5 -7 pm
- Category -
  - Chicken
  - Classic
  - Supreme
  - Veggie

Insights -

- Average no of pizza ordered per day - **138**
- **Top 3 pizza types by revenue.**
  - The Thai Chicken Pizza
  - The Barbecue Chicken Pizza
  - The California Chicken Pizza

The Classic Deluxe Pizza is the most ordered but if we look by the revenue perspective, The Thai Chicken Pizza is on first rank

IF YOU THINK IT WAS GOOD PROJECT PLZ HIT
THE LIKE ON THIS POST ❤

THANK YOU!