SQL PROJECT

ANALYSE PIZZA HUT SALES

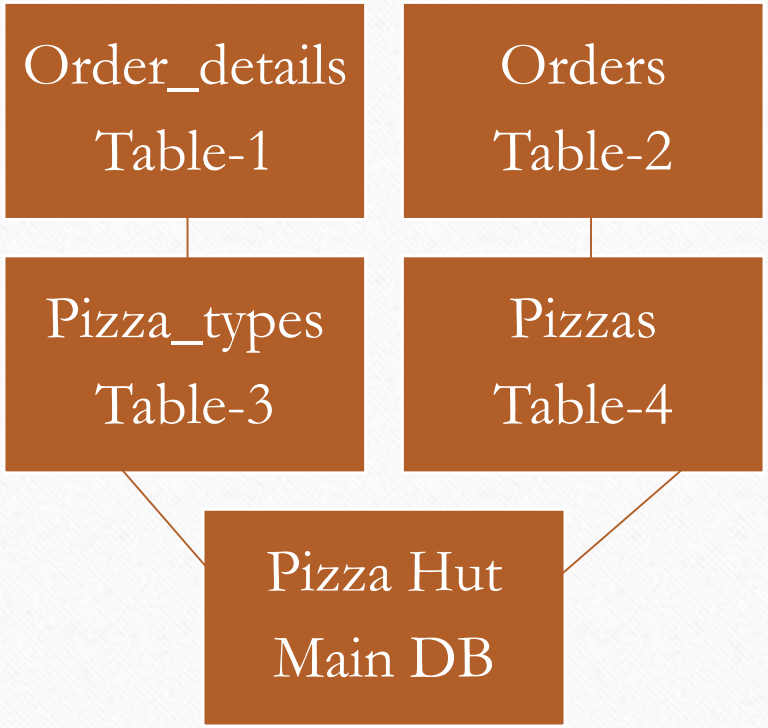
By - Sumit

HELLO I'M
SUMIT KUMAR SWAIN

"In this project, I will analyze the sales data of Pizza Hut to address their key business questions and challenges. Using SQL, I will generate insights by solving various sales-related problems and providing data-driven solutions. Below, I have listed all the queries along with their respective outputs."

By - Sumit

THE DATA IN OUR DATABASE



By - Sumit

FIRST WE HAVE TO UNDERSTAND OUR DATABASE

- So for understand the database I fetch all tables and then we can see all the data which are available for analysis.
- To fetch all the data of a table, query is –

```
SELECT * FROM order_details;
SELECT * FROM orders;
SELECT * FROM pizza_types;
SELECT * FROM pizzas;
```

- Now I fetch all the tables to understand the data I have in PIZZA HUT database.

By - Sumit

- Now for analyse the PIZZA HUT sales database I have solve some problems.
- Basic:

1. Retrieve the total number of orders placed.
2. Calculate the total revenue generated from pizza sales.
3. Identify the highest-priced pizza.
4. Identify the most common pizza size ordered.
5. List the top 5 most ordered pizza types along with their quantities.

- Intermediate:

1. Join the necessary tables to find the total quantity of each pizza category ordered.
2. Determine the distribution of orders by hour of the day.
3. Join relevant tables to find the category-wise distribution of pizzas.
4. Group the orders by date and calculate the average number of pizzas ordered per day.
5. Determine the top 3 most ordered pizza types based on revenue.

- Advanced:

1. Calculate the percentage contribution of each pizza type to total revenue.
2. Analyze the cumulative revenue generated over time.
3. Determine the top 3 most ordered pizza types based on revenue for each pizza category.

By - Sumit

RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED

- To retrieve the total number of orders placed I will use the orders table.
- So find the total number I used count function.
- The query -
- **SELECT COUNT(Order_id) AS Total_Ordered_Placed FROM Orders;**

	Total_Ordered_Placed
▶	21350

By - Sumit

Calculate The Total Revenue Generated From Pizza Sales

- To calculate the total revenue generated from pizza sales I used two tables here pizzas & order_details.
- And use ‘joins’ because my output takes data from two table.
- While joining two tables my common column is pizza_id.
- To find the total sales revenue we give query –
- **SELECT SUM(Order_details.quantity * pizzas.price) AS Total_Sales_Revenue**
  FROM Order_details
  JOIN pizzas
  ON Order_details.pizza_id = pizzas.pizza_id ;

Total_Sales_Revenue
817860.049999993

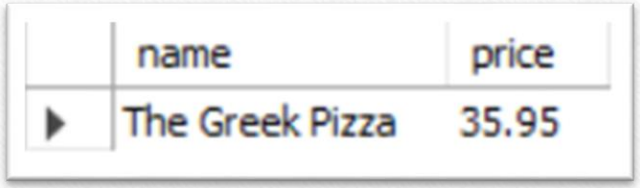
- Now the result is comes in decimal number if you want to see that in only 2 decimal places we use round function.
- Query - **SELECT ROUND (SUM (Order_details.quantity * pizzas.price), 2) AS Total_Sales_Revenue**
  FROM Order_details
  JOIN pizzas
  ON Order_details.pizza_id = pizzas.pizza_id ;

Total_Sales_Revenue
817860.05

By - Sumit

IDENTIFY THE HIGHEST-PRICED PIZZA

- To find the highest priced pizza we used two tables i.e. Pizza_types for names and Pizzas table for price.
- Now our query is –
- **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;
- Or you can use this query also –
- **SELECT pizza_types.name, pizzas.price**
  FROM pizza_types
  JOIN pizzas
  ON pizza_types.pizza_type_id = pizzas.pizza_type_id
  WHERE pizzas.price >= (select max(price) from pizzas);



	name	price
▶	The Greek Pizza	35.95

By - Sumit

IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED

- Now I have to find the most common pizza size ordered.
- So for this we used pizzas table for size and order_details table for quantity.
- And used count function to quantity for calculate the total number of orders.
- The query is –
- **SELECT pizzas.size, (count(order_details.order_details_id)) AS Total_Orders**
  FROM pizzas
  JOIN order_details
  ON pizzas.pizza_id = order_details.pizza_id
  GROUP BY pizzas.size
  ORDER BY Total_Orders desc LIMIT 1;

	size	Total_Orders
▶	L	18526

By - Sumit

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

- To calculate the top 5 most ordered pizza types with their quantities.
- We used 2 tables here order_details for quantity and pizza_types for name.
- But these 2 tables can't join because they don't have any common column so here we use another table for joining purpose i.e. pizzas table.
- So the query is –
- **SELECT pizza_types.name, SUM(order_details.quantity) AS Total_quantity**
  FROM pizzas
  JOIN pizza_types
  ON pizzas.pizza_type_id = pizza_types.pizza_type_id
  JOIN order_details
  ON pizzas.pizza_id = order_details.pizza_id
  GROUP BY pizza_types.name
  ORDER BY Total_quantity DESC LIMIT 5 ;

name	Total_quantity
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

By - Sumit

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

- To calculate the total quantity by each pizza category.
- We used two tables pizza_types for category and order_details for quantity.
- But there is no common column for joins the tables so we use another table pizzas for join purpose.
- The final query is –
- **SELECT pizza_types.category, (SUM(order_details.quantity)) AS Total_quantity**

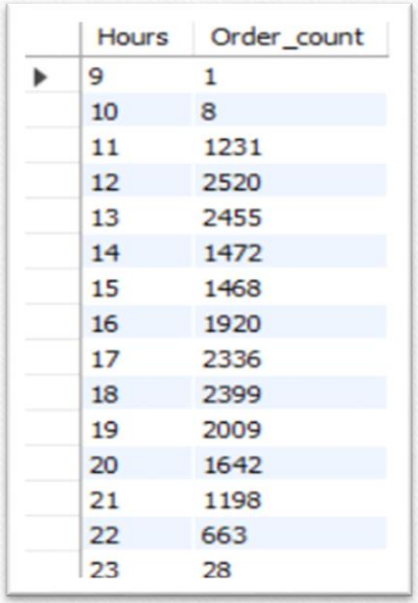
```
FROM pizzas
JOIN pizza_types
ON pizzas.pizza_type_id = pizza_types.pizza_type_id
JOIN order_details
ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizza_types.category
ORDER BY Total_quantity DESC;
```

category	Total_quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

By - Sumit

DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY

- To determine the distribution of orders by hour of a day.
- We use orders table and convert the time into hours then find total orders by each hour.
- The query is –
- **SELECT (hour(order_time)) AS Hours, (count(order_id)) AS Order_count**
  FROM orders
  GROUP BY hour(order_time)
  ORDER BY Hours;



Hours	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

By - Sumit

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

- Now find the category wise distribution of pizzas.
- Means find no. of pizza types category wise.
- The query is –
- **SELECT category, COUNT(pizza_type_id)**
  FROM pizza_types
  GROUP BY category;

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

By - Sumit

GROUP THE ORDERS BY DATE AND CALCULATE THE AVERAGE NUMBER OF PIZZAS ORDERED PER DAY

- Now we group the orders by date then calculate the average no. of pizzas ordered per day.
- So here we use Subqueries to find all the things.
- So our first query is to find the group the orders by date.
- And subquery is find the average number of pizzas ordered per day.
- So the query is –

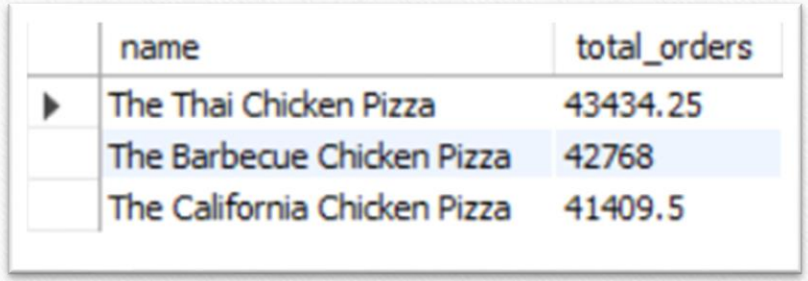
```
SELECT (ROUND(AVG(Total_quantity),2)) AS Average_quantity_ordered_perday
FROM
(SELECT orders.order_date, (SUM(order_details.quantity)) AS Total_quantity
FROM orders
JOIN order_details
ON orders.order_id = order_details.order_id
GROUP BY orders.order_date) AS Quantity;
```

Average_quantity_ordered_perday
138.47

By - Sumit

DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE

- Determine the top 3 ordered pizza types by revenue.
- So first query is to find the revenue
- Then order by desc & limit to top 3.
- So the query is –
- **SELECT pizza_types.name, (SUM(order_details.quantity * pizzas.price)) AS total_orders**
  FROM pizzas
  JOIN pizza_types
  ON pizzas.pizza_type_id = pizza_types.pizza_type_id
  JOIN order_details
  ON pizzas.pizza_id = order_details.pizza_id
  GROUP BY pizza_types.name
  ORDER BY total_orders DESC LIMIT 3;



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

By - Sumit

CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE

- To calculate the percentage of each pizza types to total revenue.
- Here we join 3 tables order_details, pizzas, pizza_types
- For percentage contribution we calculate – (Revenue from each pizza/Revenue from total pizzas)*100
- The query is –

```
SELECT pizza_types.category,
        ROUND(SUM(order_details.quantity * pizzas.price) /
        (SELECT SUM(order_details.quantity * pizzas.price)
         FROM order_details
        JOIN pizzas ON pizzas.pizza_id = order_details.pizza_id) * 100, 2) AS Revenue_percentage
FROM pizzas
JOIN order_details 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 Revenue_percentage desc;
```

category	Revenue_percentage
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

By - Sumit

ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME

- To analyse the cumulative revenue over time.
- It means how many revenue generated or increased day by day.
- The query is –
- **SELECT order_date, sum(total_revenue) over (order by order_date) as cum_revenue**

```
FROM
(SELECT orders.order_date, SUM(order_details.quantity * pizzas.price) AS total_revenue
FROM order_details
JOIN orders
ON order_details.order_id = orders.order_id
JOIN pizzas
ON pizzas.pizza_id = order_details.pizza_id
GROUP BY orders.order_date) as date_wise_revenue;
```

By - Sumit

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

- Find the top 3 most ordered pizza types based on revenue by each pizza category.
- So here we calculate 2 things top 3 most ordered by revenue & group by each pizza category.
- The query is –

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

```
SELECT name, category, revenue
  FROM
  (SELECT name, category, revenue, rank() over(partition by category order by revenue desc) as rn_num
  FROM
  (SELECT pizza_types.name, pizza_types.category, (SUM(order_details.quantity * pizzas.price))AS revenue
  FROM pizzas
  JOIN order_details
  ON pizzas.pizza_id = order_details.pizza_id
  JOIN pizza_types
  ON pizza_types.pizza_type_id = pizzas.pizza_type_id
  GROUP BY pizza_types.name, pizza_types.category) as most_ord_pizza) as rank_wise_revenue
  WHERE rn_num <= 3;
```

By - Sumit

THE END

THANK YOU

Project Done By – Sumit Kumar Swain