Pizza Hut

MySQL

Pizza Sales Project

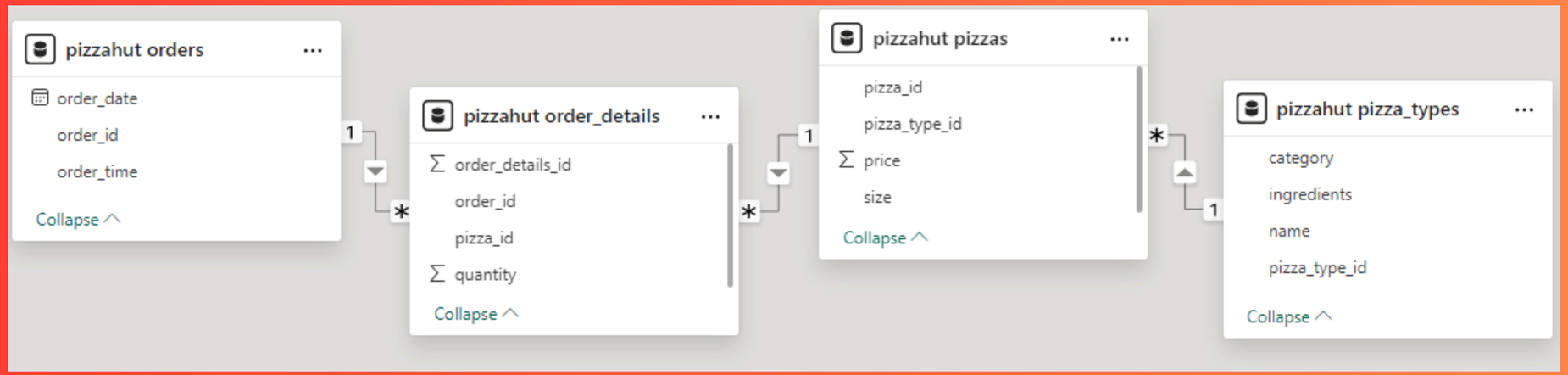
By: Mohammed Hibban

INTRODUCTION

Hi, I am Mohammed Hibban and in this project I have utilised SQL queries to solve questions that were related to Pizza sales.

Database and Schema

In this database we have 4 tables as follows



Retrieve total number of orders placed

```
23    SELECT
24        COUNT(order_id) AS total_order
25    FROM
26        orders;
27
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

total_order
21350

Result 4

Calculate the total revenue generated from Pizza sales

```sql
-- Calculate the total revenue generated from pizza sales.

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 | Filter Rows: | Export: | Wrap Cell Content:

total_sales
817860.05

Identify the highest priced Pizzas

```
41 •  SELECT
42        pizza_types.name, pizzas.price
43    FROM
44        pizza_types
45            JOIN
46        pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
47    ORDER BY pizzas.price DESC
48    LIMIT 1;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

name	price
The Greek Pizza	35.95

Identify the most common Pizza siza ordered

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

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

size	order_count
L	18526
M	15385
S	14137
XL	544
XXL	28

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;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

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

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

```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 order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY quantity;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

category	quantity
Chicken	11050
Veggie	11649
Supreme	11987
Classic	14888

Determine the distribution of orders by hour of the day.

```
93  SELECT
94      HOUR(order_time), COUNT(order_id)
95  FROM
96      orders
97  GROUP BY HOUR(order_time);
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

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

Join the relevant tables to find out the category wise distribution of Pizzas

```
100 •   SELECT
101         category, COUNT(name)
102     FROM
103         pizza_types
104     GROUP BY category;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

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

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

```sql
SELECT
    ROUND(AVG(quantity), 2) as Avg_pizza_ordered_per_day
FROM
    (SELECT
        orders.order_date, SUM(order_details.quantity) AS quantity
    FROM
        orders
    JOIN order_details ON orders.order_id = order_details.order_id
    GROUP BY orders.order_date) AS order_quantity;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Avg_pizza_ordered_per_day
138.47

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

```sql
SELECT
    pizza_types.name,
    SUM(order_details.quantity * pizzas.price) AS revenue
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 revenue DESC
LIMIT 3;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

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

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

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

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

Analyse the cumulative revenue generated overtime.

```
152 •  select order_date,sum(revenue)over(order by order_date)
153    from
154    (select orders.order_date,
155    sum(order_details.quantity*pizzas.price)as revenue
156    from order_details join pizzas
157    on order_details.pizza_id=pizzas.pizza_id
158    join orders
159    on orders.order_id=order_details.order_id
160    group by orders.order_date)as sales;
161
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

order_date	sum(revenue)over(order by order_date)
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

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

```sql
select 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(order_details.quantity*pizzas.price)as revenue
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 a)as b
where rn<=3;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

name	revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5
The Classic Deluxe Pizza	38180.5
The Hawaiian Pizza	32273.25
The Pepperoni Pizza	30161.75
The Spicy Italian Pizza	34831.25
The Italian Supreme Pizza	33476.75
The Sicilian Pizza	30940.5
The Four Cheese Pizza	32265.70000000065

Thank You