PIZZA SALES ANALYSIS USING SQL

By Anshul Tripathi

HELLO

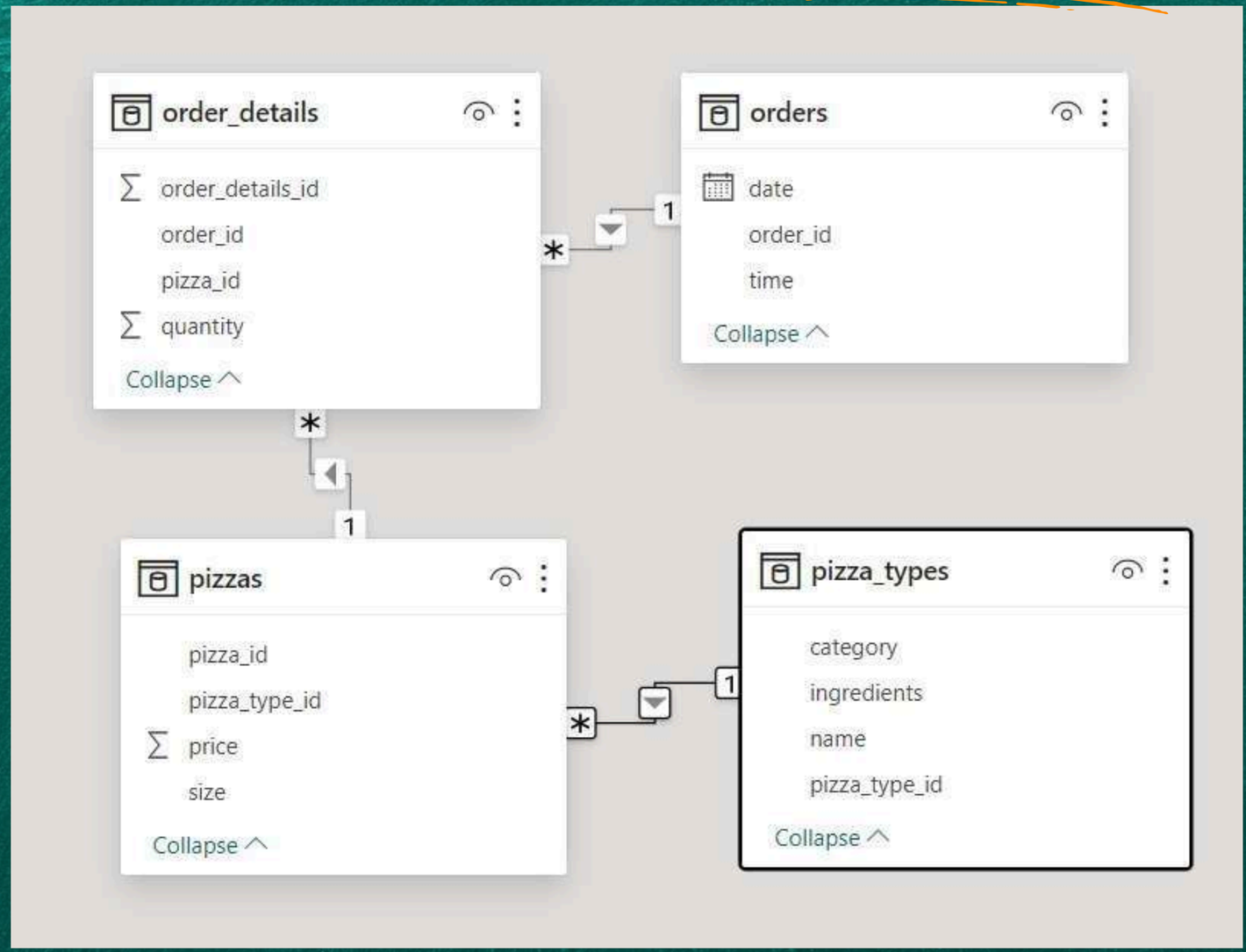
My name is Anshul Tripathi. In this project I have utilized SQL queries to solve the question those are related to the pizza sales.

OBJECTIVE

This project looks at pizza sales numbers to find out how well the restaurant is doing and what trends there are. It wants to use this information to make smarter decisions and plan better for the future.

DATA MODELING

** Data Model shows how all the tables are connencted **



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

```sql
select category,name, revenue from
(select category, name, revenue,
rank() over (partition by category order by revenue desc)as rn from
(select category, name,sum(c.quantity *b.price) as revenue
     from pizza_types as a
    join pizzas as b
    on a.pizza_type_id = b.pizza_type_id
    join order_details as c
    on b.pizza_id = c.pizza_id
    group by category,name) as a) as b
    where rn <=3;
```

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
Supreme	The Italian Supreme Pizza	33476.75
Supreme	The Sicilian Pizza	30940.5
Veggie	The Four Cheese Pizza	32265.7000000006
Veggie	The Mexicana Pizza	26780.75

2. ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

```sql
select order_date, sum(revenue)  over (order by order_date) as Cum_revenue
from
(select order_date,
sum(quantity*price) as Revenue from orders as a
join order_details as b
on a.order_id = b.order_id
join
pizzas as c
on b.pizza_id = c.pizza_id
group by order_date) as sales
```

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
2015-01-09	21526.4
2015-01-10	23990.350000000002
2015-01-11	25862.65
2015-01-12	27781.7
2015-01-13	29831.300000000003
2015-01-14	32358.700000000004
2015-01-15	34343.50000000001
2015-01-16	36937.65000000001
2015-01-17	39001.75000000001
2015-01-18	40978.600000000006

3. CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE.

```sql
select category, (sum(c.quantity *b.price)/ (SELECT
    ROUND(SUM((quantity * price)), 2) AS Total_revenue
FROM
    pizzas AS p
        JOIN
    order_details AS o ON p.pizza_id = o.pizza_id) )*100 as Revenue
 from pizza_types as a
join pizzas as b
on a.pizza_type_id = b.pizza_type_id
join order_details as c
on b.pizza_id = c.pizza_id
group by category
order by revenue
```

category	Revenue
Veggie	23.682590927384577
Chicken	23.955137556847287
Supreme	25.45631126009862
Classic	26.90596025566967

4. DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE..

```sql
select name, sum(c.quantity *b.price) as revenue
 from pizza_types as a
join pizzas as b
on a.pizza_type_id = b.pizza_type_id
join order_details as c
on b.pizza_id = c.pizza_id
group by name
order by revenue desc
limit 3
```

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

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

```sql
select round(avg(quantity),0) as Avg_pizza_ordered_perday
from (
select order_date as Days, sum(quantity) as quantity from orders as a
join order_details as b
on a.order_id = b.order_id
group by order_date
) as X
```

Avg_pizza_ordered_perday
138

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

```
select category, count(pizza_type_id) as Counts from pizza_types
group by category
```

category	count
Chicken	6
Classic	8
Supreme	9
Veggie	9

7. DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

```sql
SELECT
    HOUR(order_time) AS hour, COUNT(order_id) AS order_count
FROM
    orders
GROUP BY hour
order by hour;
```

hour	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

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

```sql
SELECT category, sum(quantity) as Quantity
FROM
    order_details AS a
        JOIN
    pizzas AS b ON a.pizza_id = b.pizza_id
        JOIN
    pizza_types AS c ON b.pizza_type_id = c.pizza_type_id
    group by category
    order by quantity desc
```

category	total_quantity
Classic	14888
Veggie	11649
Supreme	11987
Chicken	11050

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

```sql
SELECT
    name, SUM(quantity) AS Quantity
FROM
    order_details AS a
        JOIN
    pizzas AS b ON a.pizza_id = b.pizza_id
        JOIN
    pizza_types AS c ON b.pizza_type_id = c.pizza_type_id
GROUP BY 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

10. IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

```
SELECT
    p.size, COUNT(order_details_id) AS Counts
FROM
    pizzas AS p
        JOIN
    order_details AS o ON p.pizza_id = o.pizza_id
GROUP BY size
ORDER BY counts DESC;
```

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

11. IDENTIFY THE HIGHEST-PRICED PIZZA.

```
SELECT
    a.price, b.name
FROM
    pizzas AS a
        JOIN
    pizza_types AS b ON a.pizza_type_id = b.pizza_type_id
ORDER BY price DESC
LIMIT 1
```

name	price
The Greek Pizza	35.95

12. CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

```sql
SELECT
    ROUND(SUM((quantity * price)), 2) AS Total_revenue
FROM
    pizzas AS p
        JOIN
    order_details AS o ON p.pizza_id = o.pizza_id;
```

total_revenue
817860.05

13. RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED

```
select count(order_id) as Total_orders
 from orders;
```

	Total_orders
▶	21350




THANK YOU

By Anshul Tripathi