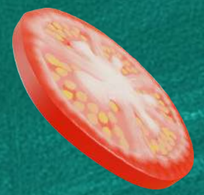
PIZZA SALES ANALYSIS USING SQL

By Nilesh Jhalani



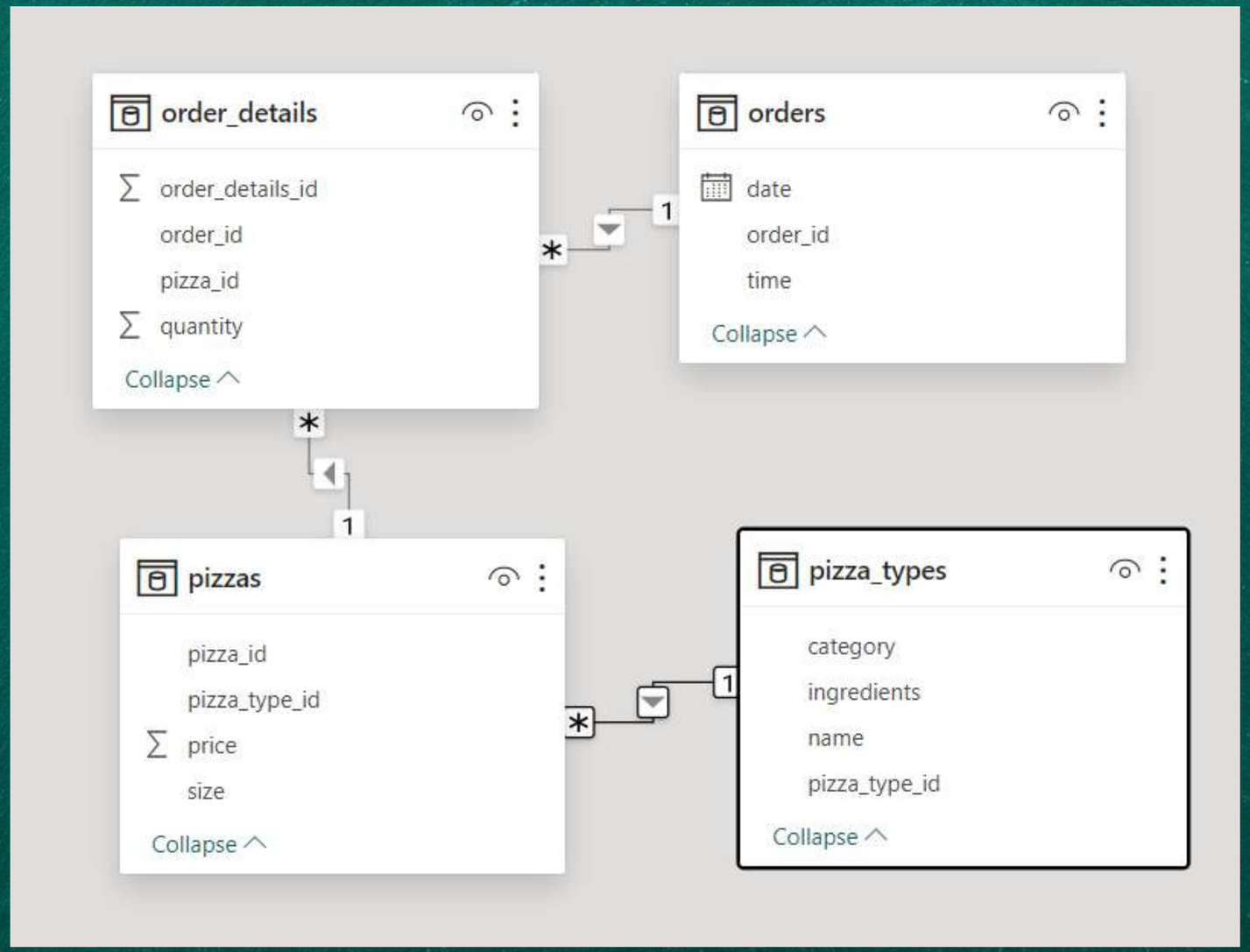
HELLO

My name is Nilesh Jhalani. In this product 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 MODEL VIEW



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 ) as  rn
from
(select pzt.category , pzt.name , sum(pz.price*od.quantity) as revenue from pizza_types as pzt
join pizzas as pz
on pz.pizza_type_id = pzt.pizza_type_id
join orders_details as od
on od.Pizza_id = pz.pizza_id
group by pzt.category , pzt.name) as a) as b
where rn<=3;
```

category	name	revenue
Chicken	The Chicken Pesto Pizza	16701.75
Chicken	The Chicken Alfredo Pizza	16900.25
Chicken	The Southwest Chicken Pizza	34705.75
Classic	The Pepperoni, Mushroom, and Peppers Pizza	18834.5
Classic	The Big Meat Pizza	22968
Classic	The Napolitana Pizza	24087
Supreme	The Brie Carre Pizza	11588.4999999999
Supreme	The Spinach Supreme Pizza	15277.75
Supreme	The Calabrese Pizza	15934.25
Veggie	The Green Garden Pizza	13955.75
Veggie	The Mediterranean Pizza	15360.5
Veggie	The Spinach Pesto Pizza	15596

2. ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

```sql
select date , sum(revenue)  over ( order by date) as cum_revenue from
(select o.order_date as date , sum(pz.price* od.quantity) as Revenue from orders_details as od
join pizzas as pz
on pz.pizza_id = od.pizza_id
join orders as o
on od.order_id = o.order_id
group by 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
    pzt.category,
    ROUND(SUM(pz.price * od.quantity) / (SELECT
                    ROUND(SUM(pz.price * od.quantity), 2)
                FROM
                    pizzas AS pz
                        JOIN
                    orders_details AS od ON pz.pizza_id = od.Pizza_id) * 100,
            2) AS revenue
FROM
    pizza_types AS pzt
        JOIN
    pizzas AS pz ON pzt.pizza_type_id = pz.pizza_type_id
        JOIN
    orders_details AS od ON od.Pizza_id = pz.pizza_id
GROUP BY pzt.category;
```

category	revenue
Classic	26.91
Veggie	23.68
Supreme	25.46
Chicken	23.96

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

```sql
SELECT
    pzt.name,
    ROUND(SUM(pz.price * od.quantity), 2) AS total_sales
FROM
    pizza_types AS pzt
        JOIN
    pizzas AS pz ON pz.pizza_type_id = pzt.pizza_type_id
        JOIN
    orders_details AS od ON pz.pizza_id = od.Pizza_id
GROUP BY pzt.name
ORDER BY total_sales 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 Average_order
FROM
    (SELECT
        o.order_date AS date, SUM(od.quantity) AS quantity
    FROM
        orders AS o
    JOIN orders_details AS od ON o.order_id = od.order_id
    GROUP BY date) AS order_quantity;
```

Average_order
138

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

```sql
SELECT
    category, COUNT(name) as count
FROM
    pizza_types AS pzt
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
    pzt.category, SUM(od.quantity) as total_quantity
FROM
    pizza_types AS pzt
        JOIN
    pizzas AS pz ON pz.pizza_type_id = pzt.pizza_type_id
        JOIN
    orders_details AS od ON pz.pizza_id = od.Pizza_id
GROUP BY pzt.category;
```

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
    pzt.name, SUM(od.quantity) AS max_order
FROM
    pizza_types AS pzt
        JOIN
    pizzas AS pz ON pz.pizza_type_id = pzt.pizza_type_id
        JOIN
    orders_details AS od ON od.Pizza_id = pz.pizza_id
GROUP BY pzt.name
ORDER BY max_order DESC
LIMIT 5;
```

name	max_order
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.

```sql
SELECT
    pz.size, COUNT(od.order_details_id) AS Total_order
FROM
    pizzas AS pz
        JOIN
    orders_details AS od ON od.Pizza_id = pz.pizza_id
GROUP BY pz.size
ORDER BY total_order DESC;
```

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

11. IDENTIFY THE HIGHEST-PRICED PIZZA.

```sql
SELECT
    pzt.name, pz.price
FROM
    pizzas AS pz
        JOIN
    pizza_types AS pzt ON pzt.pizza_type_id = pz.pizza_type_id
ORDER BY pz.price DESC
LIMIT 1;
```

name	price
The Greek Pizza	35.95




12. CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

```sql
SELECT
    ROUND(SUM(pz.price * od.quantity), 2) as total_revenue
FROM
    pizzas AS pz
        JOIN
    orders_details AS od ON pz.pizza_id = od.Pizza_id;
```

total_revenue
817860.05




13. RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED

```sql
SELECT
    COUNT(order_id) AS total_orders
FROM
    orders;
```

total_orders
21350

THANK YOU

PORTFOLIO - CLICK HERE