SQL PROJECT ON PIZZA SALE

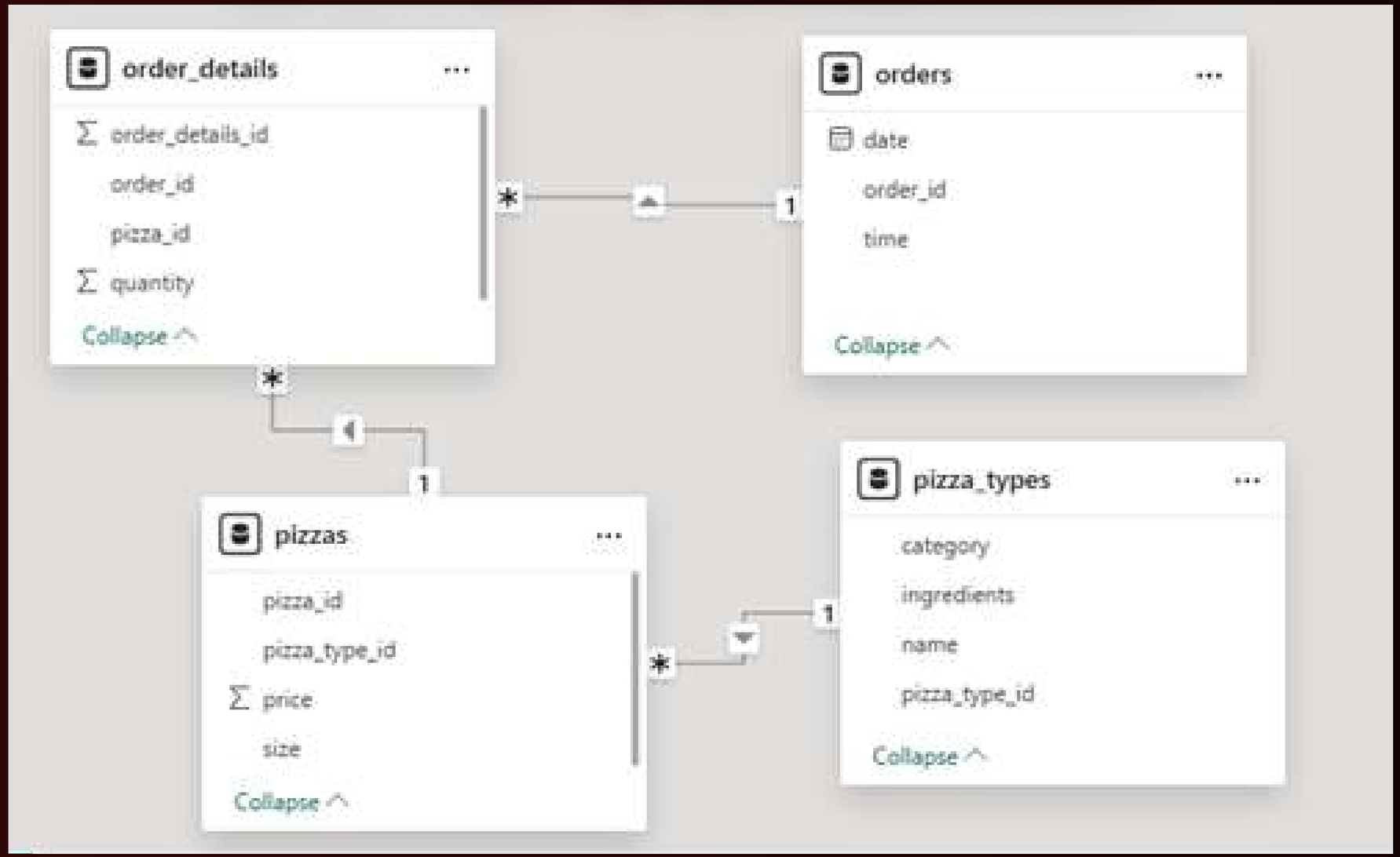
WHERE EVERY SLICE TELLS A STORY

About Contact

ABOUT THE PROJECT

Hello, my name is Rajat Gulati. In this project, I have analyzed pizza sales data using SQL to extract insights and answer key business questions.

DATA SCHEMA


order_details
order_details_id
order_id
pizza_id
quantity
Collapse
orders
date
order_id
time
Collapse
pizzas
pizza_id
pizza_type_id
price
size
Collapse
pizza_types
category
ingredients
name
pizza_type_id
Collapse
*
1

RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

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

Result Grid

	total_orders
▶	21350

CALCULATE TOTAL SALES REVENEUE GENERATED FROM PIZZAS SALES

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

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

total_sales
817860.05

IDENTIFY THE HIGHEST PRICED PIZZA.

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

	price	name
▶	35.95	The Greek Pizza

IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

```
SELECT
    pizzas.size, COUNT(order_details.quantity) as order_count
FROM
    order_details
        JOIN
    pizzas ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizzas.size
ORDER BY COUNT(order_details.quantity) DESC
LIMIT 1;
```

	size	order_count
▶	L	18526

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

```sql
SELECT
    pizza_types.name,
    COUNT(order_details.quantity) AS Number_of_Orders
FROM
    order_details
        JOIN
    pizzas 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
ORDER BY COUNT(order_details.quantity) DESC
LIMIT 5;
```

name	Number_of_Orders
The Classic Deluxe Pizza	2416
The Barbecue Chicken Pizza	2372
The Hawaiian Pizza	2370
The Pepperoni Pizza	2369
The Thai Chicken Pizza	2315

DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR PER DAY

```
2
3 •   SELECT
4         HOUR(order_time), COUNT(order_id) AS Count_of_orders
5     FROM
6         orders
7     GROUP BY HOUR(order_time);
```

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

HOUR(order_time)	Count_of_orders
11	1231
12	2520
13	2455
14	1472
15	1468

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

```
SELECT
    pizza_types.category, SUM(order_details.quantity) AS qty
FROM
    order_details
        JOIN
    pizzas 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 SUM(order_details.quantity)
;
```

category	qty
Chicken	11050
Veggie	11649
Supreme	11987
Classic	14888

FIND THE CATEGORYWISE DISTRIBUTION OF PIZZAS.

```
3    SELECT
4        category, COUNT(name)
5    FROM
6        pizza_types
7    GROUP BY category
8    ;
```

Result Grid | Filter Rows:

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(qty))
FROM
    (SELECT
        orders.order_date AS date,
            SUM(order_details.quantity) AS qty
    FROM
        order_details
    JOIN orders ON order_details.order_id = orders.order_id
    GROUP BY date
    ORDER BY qty DESC) AS order_qty;
```

	ROUND(AVG(qty))
▶	138

DETERMINE THE TOP 3 MOST ORDERED PIZZAS BASED ON REVENUE

```sql
SELECT
    pizza_types.name AS pname,
    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 pizzas.pizza_id = order_details.pizza_id
GROUP BY pname
ORDER BY revenue DESC
LIMIT 3
```

pname	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 TO THE TOTAL REVENUE

```sql
SELECT
    pizza_types.category AS cat,
   concat( round(SUM(order_details.quantity * pizzas.price) / (SELECT
            SUM(order_details.quantity * Pizzas.price)
        FROM
            pizzas
                JOIN
            order_details ON order_details.pizza_id = pizzas.pizza_id) * 100,2 ), ' ', '%') AS rev
FROM
    pizzas
        JOIN
    pizza_types ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.Pizza_id = pizzas.pizza_id
GROUP BY cat
ORDER BY rev;
```

cat	rev
Veggie	23.68 %
Chicken	23.96 %
Supreme	25.46 %
Classic	26.91 %

ANALYSE THE CUMULATIVE REVENUE GENERATED OVER TIME.

```sql
SELECT
    order_date,
    SUM(revenue) OVER (ORDER BY order_date) AS cum_revenue
FROM (
    SELECT
        orders.order_date,
        SUM(order_details.quantity * pizzas.price) AS revenue
    FROM order_details
    JOIN pizzas
        ON order_details.pizza_id = pizzas.pizza_id
    JOIN orders
        ON orders.order_id = order_details.order_id
    GROUP BY orders.order_date
) AS total_sales
ORDER BY order_date;
```

order_date	cum_revenue
2015-01-01	2713.850000000004
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 TYPES BASED ON REVENUE FOR EACH PIZZA CATEGORY

```sql
select category, name, rev, ran from (
select category, name, rev, rank() over(partition by category order by rev desc) as ran
from
(select pizza_types.category, pizza_types.name, sum((order_details.quantity) * pizzas.price) as rev
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.category, pizza_types.name ) as a
) as b where ran<=3  ;
```

category	name	rev	ran
Chicken	The Thai Chicken Pizza	43434.25	1
Chicken	The Barbecue Chicken Pizza	42768	2
Chicken	The California Chicken Pizza	41409.5	3
Classic	The Classic Deluxe Pizza	38180.5	1
Classic	The Hawaiian Pizza	32273.25	2
Classic	The Pepperoni Pizza	30161.75	3
Supreme	The Spicy Italian Pizza	34831.25	1
Supreme	The Italian Supreme Pizza	33476.75	2
Supreme	The Sicilian Pizza	30940.5	3
Veggie	The Four Cheese Pizza	32265.70000000065	1
Veggie	The Mexicana Pizza	26780.75	2
Veggie	The Five Cheese Pizza	26066.5	3

PROJECT SUMMARY

1. ANALYZED PIZZA SALES DATA USING SQL TO EXTRACT BUSINESS INSIGHTS
2. PERFORMED BASIC REPORTING ON OVERALL ORDERS AND REVENUE
3. IDENTIFIED POPULAR PIZZAS, SIZES, AND PRICING TRENDS
4. JOINED TABLES TO EXPLORE PIZZA CATEGORY PERFORMANCE AND TIME-BASED PATTERNS
5. CALCULATED AVERAGES AND TOP REVENUE-GENERATING PIZZA TYPES
6. USED ADVANCED SQL TO MEASURE REVENUE CONTRIBUTIONS AND CUMULATIVE SALES TRENDS
7. DELIVERED INSIGHTS TO SUPPORT DECISIONS ON INVENTORY, PRICING, AND CUSTOMER DEMAND

THANK YOU

FOR ATTENTION

- SQL PROJECT 2026 BY RAJAT GULATI