SQL Project on pizza sales

-By Pravat Kumar Gupta

it's Pizza Time

EAT HEALTHY FOOD TO LIVE HEALTHY

+91-9681818427

Hello!

My name is Pravat Kumar Gupta. In this project I have utilised SQL queries to solve questions that were related to Pizza Sales

FRESHLY BAKED, DELIVERED TO YOUR DOOR.

QUESTIONS:

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.

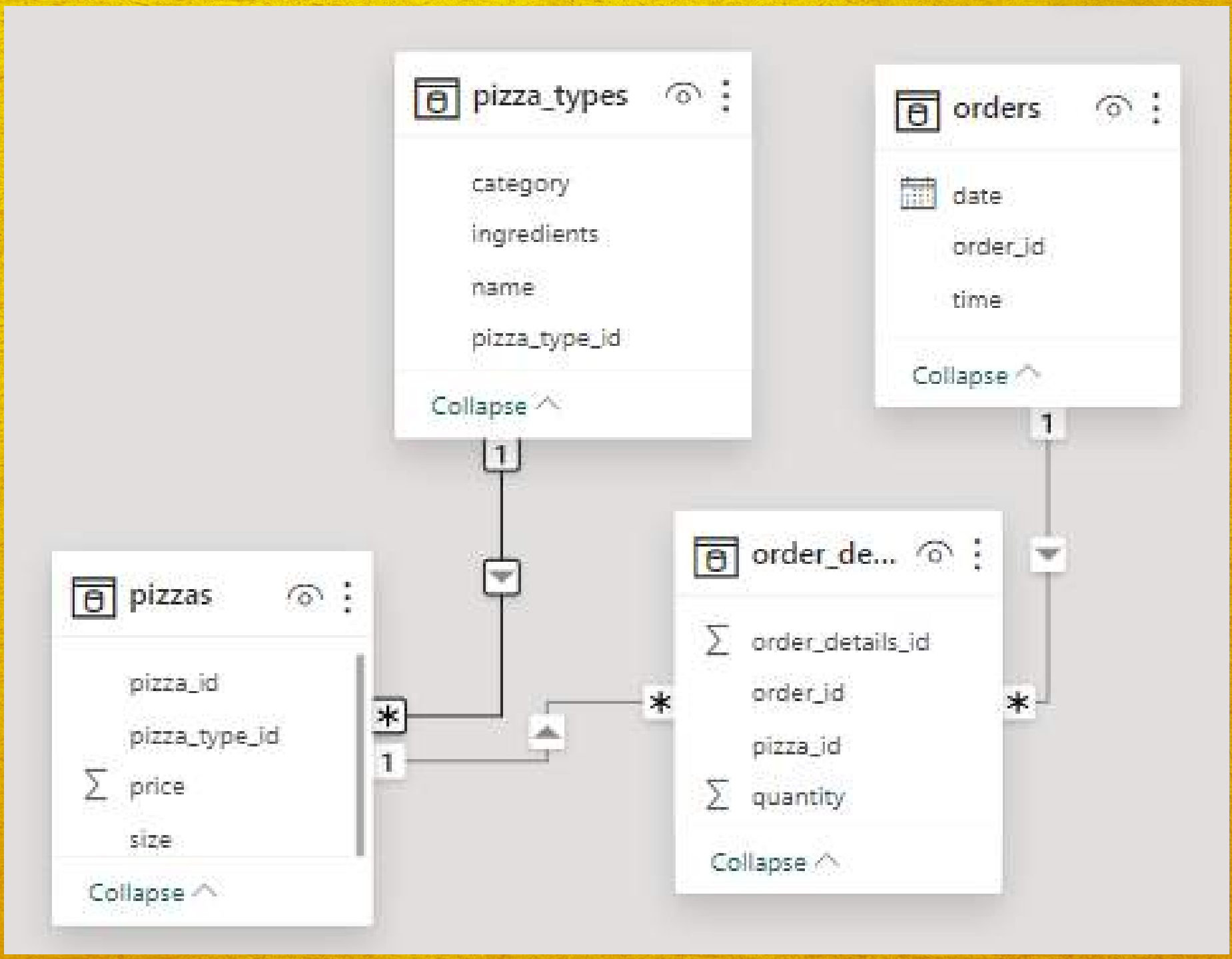
INTERMEDIATE:

6. JOIN THE NECESSARY TABLES TO FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED.
7. DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.
8. JOIN RELEVANT TABLES TO FIND THE CATEGORY-WISE DISTRIBUTION OF PIZZAS.
9. GROUP THE ORDERS BY DATE AND CALCULATE THE AVERAGE NUMBER OF PIZZAS ORDERED PER DAY.
10. DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE.

ADVANCED:

11. CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE.
12. ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.
13. DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE FOR EACH PIZZA CATEGORY.

MODEL VIEW


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

Q1. RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

```
SELECT
    COUNT(order_id) AS Total_Orders
FROM
    orders;
```

Result Grid

	Total_Orders
▶	21350

Q2. CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

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

Result Grid

	total_revenue
▶	817860.05

Q3. IDENTIFY THE HIGHEST-PRICED PIZZA.

```sql
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;
```

Result Grid | Filter Rows:

name	price
The Greek Pizza	35.95

Q4. IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

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

Result Grid

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

Q5. LIST THE TOP 5 MOST ORDERED PIZZA TYPES ALONG WITH THEIR QUANTITIES.

```
SELECT pt.name, SUM(od.quantity) AS quantity
FROM pizza_types pt
  JOIN pizzas p ON pt.pizza_type_id = p.pizza_type_id
    JOIN orders_details od ON od.pizza_id = p.pizza_id
GROUP BY pt.name ORDER BY quantity DESC LIMIT 5;
```

Result Grid | Filter 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

Q6. JOIN THE NECESSARY TABLES TO FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED.

```
SELECT pt.category, SUM(od.quantity) AS quantity
FROM pizza_types pt
  JOIN pizzas p ON pt.pizza_type_id = p.pizza_type_id
    JOIN orders_details od ON od.pizza_id = p.pizza_id
GROUP BY pt.category
ORDER BY quantity DESC;
```

Result Grid

category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

Q7. 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_time);
```

Result Grid | Filt

hour	order_count
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

Q8. JOIN RELEVANT TABLES TO FIND THE CATEGORY-WISE DISTRIBUTION OF PIZZAS.

```
SELECT
    Category, COUNT(name) AS Total_No
FROM
    pizza_types
GROUP BY Category;
```

Result Grid | Filt

Category	Total_No
Chicken	6
Classic	8
Supreme	9
Veggie	9

Q9. GROUP THE ORDERS BY DATE AND CALCULATE THE AVERAGE NUMBER OF PIZZAS ORDERED PER DAY.

```sql
SELECT
    ROUND(AVG(quantity), 0) AS Avg_quantity
FROM
    (SELECT
        o.order_date, SUM(od.quantity) AS quantity
    FROM
        orders o
    JOIN orders_details od ON o.order_id = od.order_id
    GROUP BY o.order_date) AS order_quantity;
```

Result Grid

	Avg_quantity
	138

Q10. DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE.

```sql
SELECT
    pt.name, SUM(od.quantity * p.price) AS revenue
FROM pizza_types pt
    JOIN pizzas p ON p.pizza_type_id = pt.pizza_type_id
        JOIN orders_details od ON od.pizza_id = p.pizza_id
GROUP BY pt.name
ORDER BY revenue DESC
LIMIT 3;
```

Result Grid | Filter Rows:

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

Q11. CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE.

```sql
SELECT pt.category, ROUND(SUM(od.quantity * p.price) /
 (SELECT ROUND(SUM(od.quantity * p.price), 2) AS total_sales
FROM orders_details od
    JOIN pizzas p ON p.pizza_id = od.pizza_id) * 100, 2) AS revenue
FROM pizza_types pt JOIN pizzas p ON pt.pizza_type_id = p.pizza_type_id
    JOIN orders_details od ON od.pizza_id = p.pizza_id
GROUP BY pt.category ORDER BY revenue DESC;
```

Result Grid

category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

Q12. ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

```
SELECT order_date, SUM(revenue)
 OVER(ORDER BY order_date) AS cumulative_revenue
FROM (SELECT o.order_date,
SUM(od.quantity * p.price) AS revenue
      FROM orders_details od
        JOIN pizzas p ON od.pizza_id = p.pizza_id
          JOIN orders o ON o.order_id = od.order_id
            GROUP BY o.order_date) AS sales;
```

Result Grid | Filter Rows:

order_date	cumulative_revenue
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	23990.35000000000
2015-01-11	25862.65
2015-01-12	27781.7
2015-01-13	29831.30000000000
2015-01-14	32358.70000000000
2015-01-15	34343.50000000001
2015-01-16	36937.65000000001
2015-01-17	39001.75000000001
2015-01-18	40978.60000000000
2015-01-19	43365.75000000001

Q13. DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE FOR EACH PIZZA CATEGORY.

```
SELECT name, revenue FROM
(SELECT category, name, revenue,
  RANK() OVER(PARTITION BY category ORDER BY revenue DESC) AS rnk
  FROM
 (SELECT pt.category, pt.name, SUM((od.quantity) * p.price) AS revenue
  FROM pizza_types pt JOIN pizzas p
   ON pt.pizza_type_id = p.pizza_type_id
    JOIN orders_details od ON od.pizza_id = p.pizza_id
     GROUP BY pt.category, pt.name) AS a) AS b
WHERE rnk <= 3;
```

Result Grid | Filter Rows:

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

CONCLUSION

As this is my first project so got little excited working on this project. Based on the SQL analysis conducted on the pizza sales data, several key findings and insights have been revealed. These insights include identifying the best-selling pizza toppings, total revenue generated, peak sales hours, highest-priced pizza, popular pizza sizes, and customer spending patterns.

By doing so, Stackholders can make informed decisions and foster business growth within the competitive pizza industry.

GET IN TOUCH WITH US

+91-9681818427

pkg.me21@gmail.com

Kolkata, West Bengal

WWW.LINKEDIN.COM/IN/PRAVATGUPTA


Thank
you!
pkg.me21@gmail.com