

DELICIOUS PIZZA SALES ANALYSIS USING SQL

[Home](#)

[About](#)

[Contact](#)





[Home](#)

[About](#)

[Contact](#)

NAME:-LIPAN PATRA

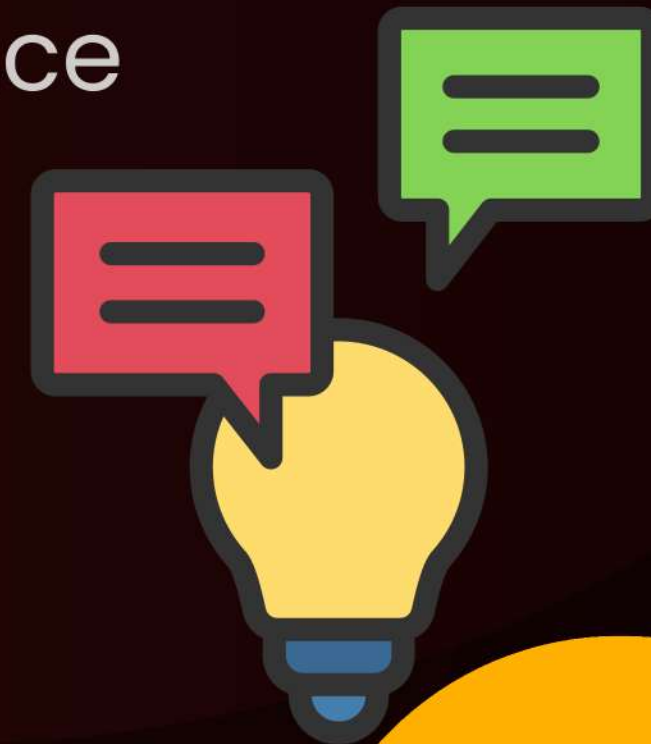


PROJECT OVERVIEW

Analyze sales data of a pizza store chain (like Delicious) to understand customer preferences, revenue patterns, and performance insights.”

[About](#)

[Contact](#)



\$30

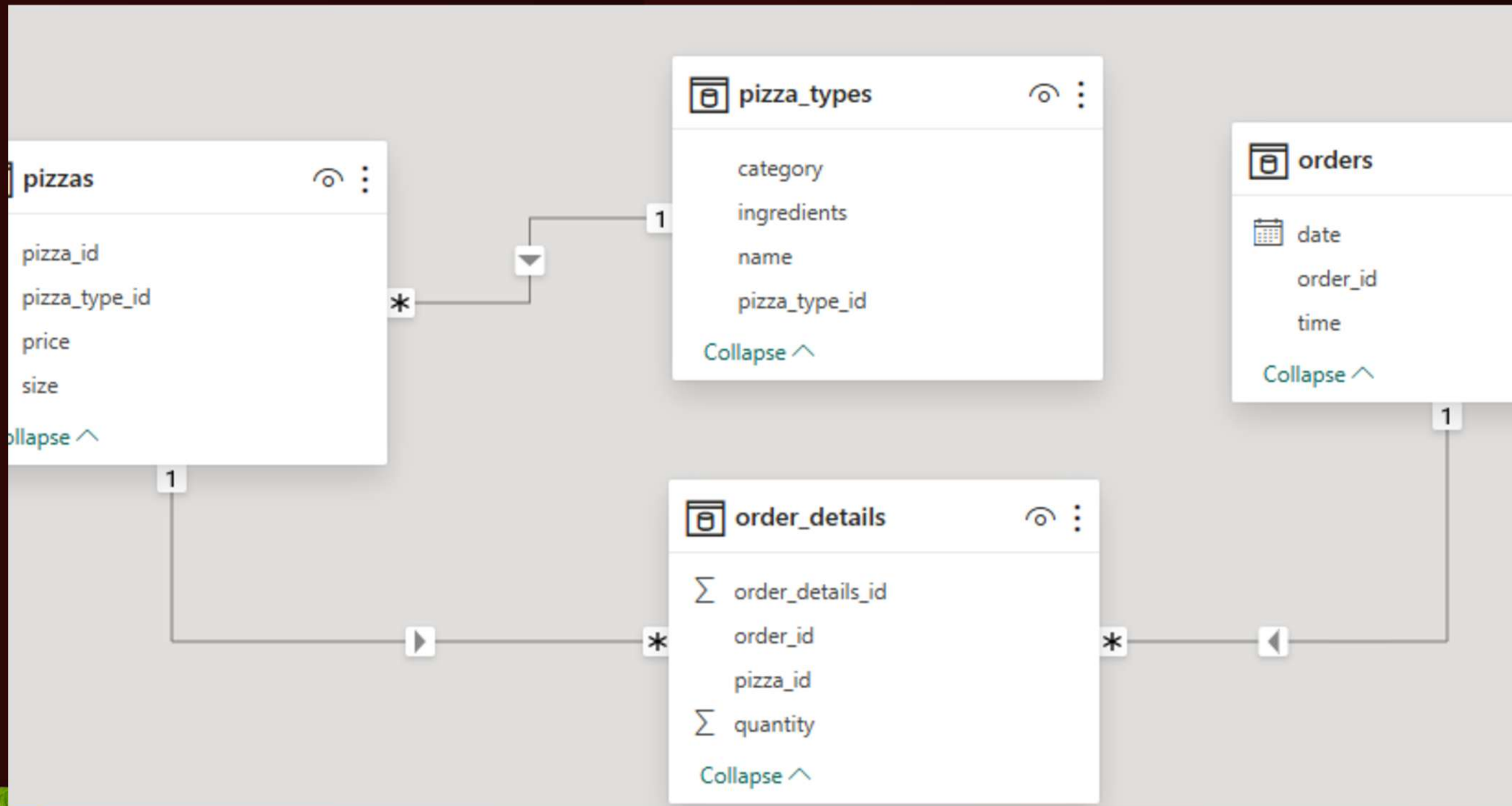
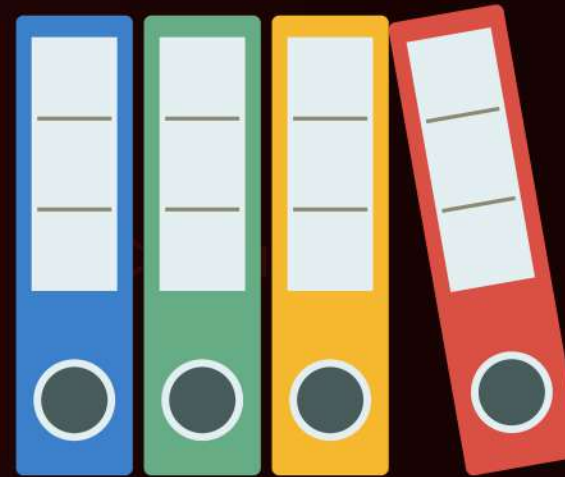


OBJECTIVES



The objective of this project is to analyze pizza sales data using SQL to gain insights into business performance and customer preferences. It focuses on identifying total orders, revenue trends, top-selling pizzas, and popular sizes and categories. By examining sales distribution over time and revenue contribution by pizza types, the project aims to support data-driven decisions to improve sales, marketing, and inventory strategies for the pizza store.

DATASET DESCRIPTION

[Home](#)[About](#)



[Home](#)

[About](#)

[Contact](#)

SQL CONCEPTS USED

- JOIN, GROUP BY, ORDER BY, having
- WINDOW FUNCTIONS (like RANK() or ROW_NUMBER())
- CTE (Common Table Expressions)
- DATE functions (e.g., MONTH(), DAY())

BUSINESS QUESTIONS

[About](#)[Contact](#)

1-RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED

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

total_order
21350

BUSINESS QUESTIONS

[About](#)[Contact](#)

CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

```
select sum(order_details.quantity * pizzaa.price) as total_sales  
from order_details  
join pizzaa  
on order_details.pizza_id = pizzaa.pizza_id;
```

total_sales

817860.05083847

BUSINESS QUESTIONS

[About](#)[Contact](#)

IDENTIFY THE HIGHEST-PRICED PIZZA

```
SELECT p.price, pt.name  
FROM pizzaa p  
JOIN pizza_types pt  
  ON p.pizza_type_id = pt.pizza_type_id  
WHERE p.price = (SELECT MAX(price) FROM pizzaa);
```

size	total_orders
L	18526

BUSINESS QUESTIONS

[About](#)[Contact](#)

IDENTIFY THE MOST COMMON PIZZA QUANTITY ORDERED.

```
select quantity ,count(order_details_id) as total_number  
from order_details  
group by quantity  
order by total_number desc;
```

quantity	total_number
1	47693

BUSINESS QUESTIONS

[About](#)[Contact](#)

IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED

```
SELECT p.size, COUNT(od.order_details_id) AS total_orders
FROM order_details od
JOIN pizzaa p
  ON p.pizza_id = od.pizza_id
GROUP BY p.size
ORDER BY total_orders DESC;
```

size	total_orders
L	18526

BUSINESS QUESTIONS

[About](#)[Contact](#)

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

```
select top 5 t.name, sum(od.quantity) as total_quantity
from pizzaa a
join pizza_types t
on t.pizza_type_id=a.pizza_type_id
join order_details od
on od.pizza_id=a.pizza_id
group by t.name
order by total_quantity desc ;
```

name	total_quantity
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

BUSINESS QUESTIONS

[About](#)[Contact](#)

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

```
select  t.category, sum(od.quantity) as total_quantity
from pizzaa a
join pizza_types t
on t.pizza_type_id=a.pizza_type_id
join order_details od
on od.pizza_id=a.pizza_id
group by t.category
order by total_quantity desc ;
```

category	total_quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

BUSINESS QUESTIONS

[About](#)[Contact](#)

DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

```
SELECT DATEPART(HOUR, time) AS order_hour,  
       COUNT(order_id) AS total_orders  
FROM orders  
GROUP BY DATEPART(HOUR, time)  
ORDER BY order_hour desc;
```

order_hour	total_orders
23	28
22	663
21	1198
20	1642
19	2009
18	2399
17	2336
16	1920
15	1468
14	1472
13	2455
12	2520
11	1231
10	8

BUSINESS QUESTIONS

[About](#)[Contact](#)

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

```
select category, count(name) as total_type_of_piza  
from pizza_types  
group by category;
```

category	total_type_of_piza
Chicken	6
Classic	8
Supreme	9
Veggie	9

BUSINESS QUESTIONS

[About](#)[Contact](#)

CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH CATEGORY TO TOTAL REVENUE.

```
select t.category, (sum(p.price*od.quantity) /  
(select sum(order_details.quantity * pizzaa.price) as total_sales  
from order_details  
join pizzaa  
on order_details.pizza_id= pizzaa.pizza_id))*100 as revenue  
from pizza_types t  
join pizzaa p  
on t.pizza_type_id=p.pizza_type_id  
join order_details od  
on od.pizza_id=p.pizza_id  
group by t.category  
order by revenue desc;
```

category	revenue
Classic	26.9059602306976
Supreme	25.4563112111462
Chicken	23.9551375322885
Veggie	23.6825910258677

BUSINESS QUESTIONS

[About](#)[Contact](#)

ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

```
SELECT
    date,
    daily_revenue,
    SUM(daily_revenue) OVER (ORDER BY date) AS cumulative_revenue
FROM (
    SELECT
        o.date,
        SUM(p.price * od.quantity) AS daily_revenue
    FROM orders o
    JOIN order_details od
        ON o.order_id = od.order_id
    JOIN pizzaa p
        ON od.pizza_id = p.pizza_id
    GROUP BY o.date
) AS daily
ORDER BY date;
```

date	daily_revenue	cumulative_revenue
2015-01-01	2713.85000228882	2713.85000228882
2015-01-02	2731.90000152588	5445.7500038147
2015-01-03	2662.40000343323	8108.15000724792
2015-01-04	1755.45000076294	9863.60000801086
2015-01-05	2065.95000076294	11929.5500087738

BUSINESS QUESTIONS

[About](#)[Contact](#)

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

```
WITH DailyTotals AS (  
    SELECT o.date, SUM(od.quantity) AS total_pizzas  
    FROM orders o  
    JOIN order_details od  
        ON o.order_id = od.order_id  
    GROUP BY o.date  
)  
SELECT AVG(total_pizzas) AS avg_pizzas_per_day  
FROM DailyTotals;
```

avg_pizzas_per_day

138

BUSINESS QUESTIONS

[About](#)[Contact](#)

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

```
select category,pizza_type_id,daily_revenue,rank_c
from
  (select SUM(p.price * od.quantity) AS daily_revenue,t.category,t.pizza_type_id,
    rank() over(PARTITION BY category ORDER BY SUM(p.price * od.quantity)DESC) as rank_c
  from pizzaa p
  join order_details od
  on p.pizza_id=od.pizza_id
  join pizza_types t
  on t.pizza_type_id=p.pizza_type_id
  group by t.category,t.pizza_type_id
  )as ab
where rank_c <=3
```

category	pizza_type_id	daily_revenue	rank_c
Chicken	thai_ckn	43434.25	1
Chicken	bbq_ckn	42768	2
Classic	classic_dlx	38180.5	1
Classic	hawaiian	32273.25	2
Supreme	spicy_ital	34831.25	1
Supreme	ital_supr	33476.75	2
Veggie	four_cheese	32265.7010040283	1
Veggie	mexicana	26780.75	2



[Home](#)

[About](#)

[Contact](#)

TOOLS & TECHNOLOGIES USED

- SQL Server
- Excel and Power BI for modeling
- Kaggle for dataset





[Home](#)

[About](#)

[Contact](#)



THANK YOU



FOR ATTENTION .

