

Project on:

PIZZA SALES DATA ANALYSIS

A Data-Driven Analysis of Sales Trends and Business Insights

Name :

Parmar Arpitaba M.

Date :

25-08-2025





Introduction & Project goal

Our Story Begins

Every pizza we sell creates a piece of data. This project is about bringing that data to life. We're diving into a year's worth of transactions to answer the big questions: What are we doing right? And where can we improve?

The Mission

Our goal is to be the **data detectives** of this project. We will use SQL to solve the mystery of our sales performance and translate our findings into a clear strategy for the business.



DATA & TOOLS

Key Tables/Data Points:

pizza : pizza_id,pizza_type,size,price

Pizza_type : pizza_type_id,name,category,ingredient

orders : order_id,date,time

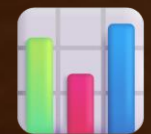
Order_detail : order_detail_id,order_id,pizza_id,quantity

Tools used:

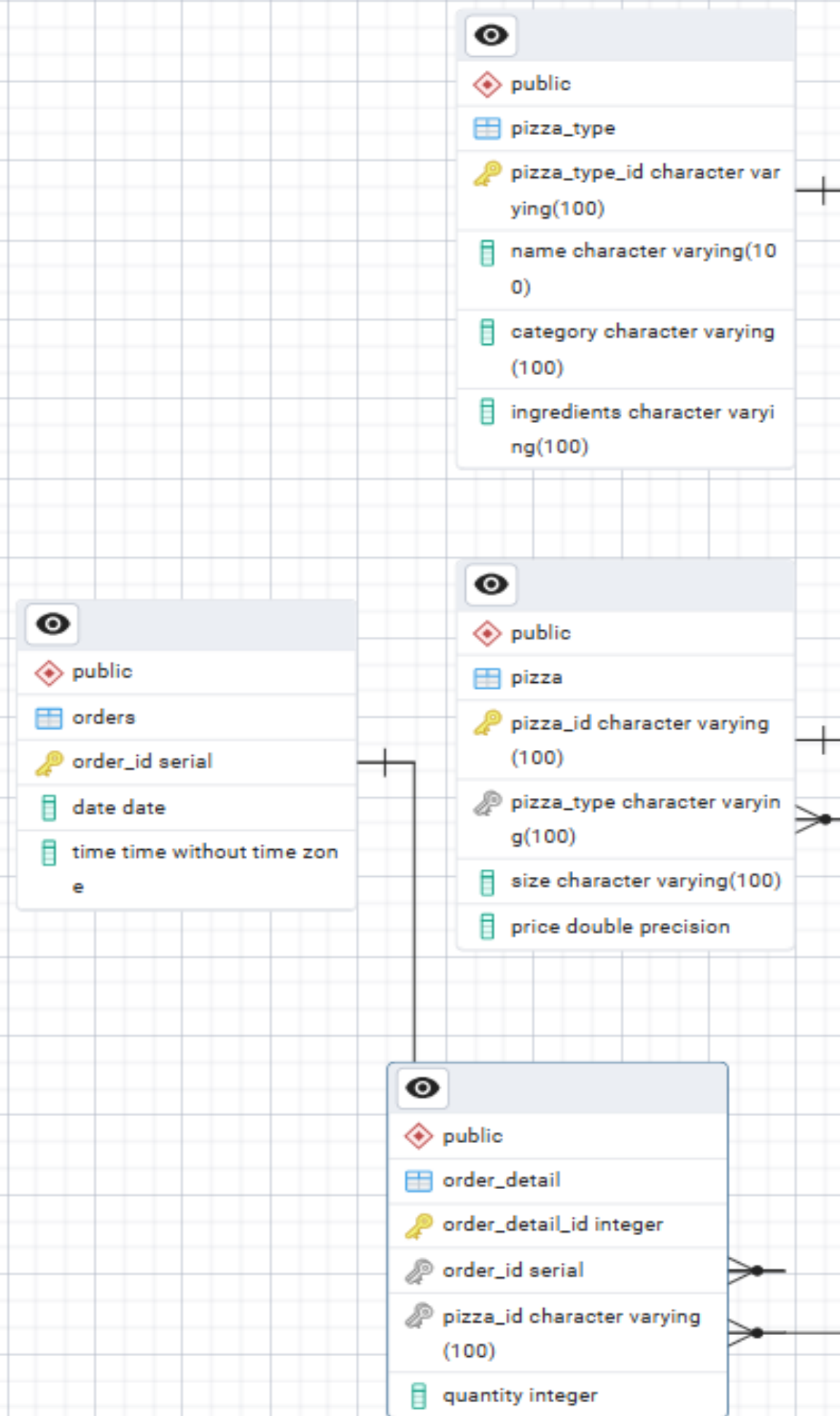
Database: PostgreSQL, MySQL, or SQLite

Query Language: SQL

[Read More](#)



E-R DIAGRAM :



Challenges

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.



Basic




1.RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED

```
--Basic
1--Retrieve the total number of orders placed.

SELECT
  COUNT (ORDER_ID)
FROM
  ORDERS
```



OUTPUT

	count 
1	21350

2. CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

2--Calculate the total revenue generated from pizza sales.

```
SELECT
SUM(P.PRICE * OD.QUANTITY) AS REVENUE
FROM
PIZZA P
JOIN ORDER_DETAIL OD ON P.PIZZA_ID = OD.PIZZA_ID
```



OUTPUT

Data Output		Messages	Notifications
	revenue double precision		
1	817860.0499999993		

3. IDENTIFY THE HIGHEST-PRICED PIZZA.

3--Identify the highest-priced pizza.

```
SELECT
    MAX(PRICE) AS HIGHEST_PRICE,
    NAME
FROM
    PIZZA P
    JOIN PIZZA_TYPE PT ON P.PIZZA_TYPE = PT.PIZZA_TYPE_ID
GROUP BY
    PT.NAME
ORDER BY
    HIGHEST_PRICE DESC
LIMIT
    1
```



OUTPUT

	highest_price double precision 🔒	name character varying (100) 🔒
1	35.95	The Greek Pizza

4. IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

4--Identify the most common pizza size ordered.

```
SELECT
  P.SIZE,
  COUNT(OD.ORDER_DETAIL_ID) AS NUM_ORDER
FROM
  PIZZA P
  JOIN ORDER_DETAIL OD ON P.PIZZA_ID = OD.PIZZA_ID
GROUP BY
  P.SIZE
ORDER BY
  NUM_ORDER DESC
```



OUTPUT

	size character varying (100) 🔒	num_order bigint 🔒
1	L	18526
2	M	15385
3	S	14137
4	XL	544
5	XXL	28

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

5--List the top 5 most ordered pizza types along with their quantities.

```
SELECT
  PT.NAME,
  SUM(OD.QUANTITY) AS TOTAL
FROM
  PIZZA P
  JOIN PIZZA_TYPE PT ON P.PIZZA_TYPE = PT.PIZZA_TYPE_ID
  JOIN ORDER_DETAIL OD ON P.PIZZA_ID = OD.PIZZA_ID
GROUP BY
  PT.NAME
ORDER BY
  TOTAL DESC
LIMIT
  5
```



OUTPUT

	name character varying (100)	total
	Graph Visual	
1	The Classic Deluxe Pizza	2453
2	The Barbecue Chicken Pizza	2432
3	The Hawaiian Pizza	2422
4	The Pepperoni Pizza	2418
5	The Thai Chicken Pizza	2371

Intermediate



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

```
6--JOIN THE NECESSARY TABLES TO FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED.  
SELECT SUM(OD.QUANTITY), PT.CATEGORY  
FROM  
  PIZZA_TYPE PT  
  JOIN PIZZA P ON PT.PIZZA_TYPE_ID = P.PIZZA_TYPE  
  JOIN ORDER_DETAIL OD ON OD.PIZZA_ID = P.PIZZA_ID  
GROUP BY  
  PT.CATEGORY
```



OUTPUT

	sum bigint	category character varying (100)
1	11987	Supreme
2	14888	Classic
3	11649	Veggie
4	11050	Chicken

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

7--Determine the distribution of orders by hour of the day.

```
SELECT
  EXTRACT(
    HOUR
    FROM
      TIME
  ) TIME_DAY,
  COUNT(ORDER_ID) AS ORDERS_DAY
FROM
  ORDERS
GROUP BY
  TIME_DAY
```



OUTPUT

Data Output			Messages	Notifica
	time_day numeric	orders_day bigint		
1	11	1231		
2	23	28		
3	18	2399		
4	19	2009		
5	15	1468		
6	9	1		
7	21	1198		
8	17	2336		

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

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

```
SELECT COUNT(NAME),
```

```
CATEGORY
```

```
FROM
```



```
PIZZA_TYPE
```

```
GROUP BY
```

```
CATEGORY
```



OUTPUT

	count bigint 	category character varying (100) 
1	9	Supreme
2	8	Classic
3	9	Veggie
4	6	Chicken


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

9--Group the orders by date and calculate the average number of pizzas ordered per day.

```
WITH
  DAILY_PIZZA AS (
    SELECT
      O.DATE,
      SUM(OD.QUANTITY) AS AVG_QUANTITY
    FROM
      ORDERS O
      JOIN ORDER_DETAIL OD ON O.ORDER_ID = OD.ORDER_ID
    GROUP BY
      O.DATE
  )
SELECT
  AVG(AVG_QUANTITY)
FROM
  DAILY_PIZZA;
```



OUTPUT

	avg numeric 
1	138.4748603351955307



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

```
10--Determine the top 3 most ordered pizza types based on revenue.

SELECT
    SUM(P.PRICE * OD.QUANTITY) AS REVENUE,
    PT.NAME
FROM
    PIZZA P
    JOIN PIZZA_TYPE PT ON P.PIZZA_TYPE = PT.PIZZA_TYPE_ID
    JOIN ORDER_DETAIL OD ON P.PIZZA_ID = OD.PIZZA_ID
GROUP BY
    PT.NAME
ORDER BY
    REVENUE DESC
LIMIT
    3
```



OUTPUT

	revenue double precision	name character
1	43434.25	The Thai Chicken Pizza
2	42768	The Barbecue Chicken Pizza
3	41409.5	The California Chicken Pizza

[Graph Visualiser](#)

Advanced



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

```
Advanced.  
11--Calculate the percentage contribution of each pizza type to total revenue.  
--this is using windows function  
  
SELECT  
  PT.CATEGORY,  
  SUM(P.PRICE * OD.QUANTITY) / SUM(SUM(P.PRICE * OD.QUANTITY)) OVER () * 100  
FROM  
  PIZZA P  
  JOIN PIZZA_TYPE PT ON P.PIZZA_TYPE = PT.PIZZA_TYPE_ID  
  JOIN ORDER_DETAIL OD ON P.PIZZA_ID = OD.PIZZA_ID  
GROUP BY  
  PT.CATEGORY
```



OUTPUT

	category character varying (100)	?column? double precision
1	Supreme	25.45631126009858
2	Classic	26.905960255669626
3	Veggie	23.68259092738454
4	Chicken	23.955137556847248

12. ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

```
12--Analyze the cumulative revenue generated over time.

WITH
  A AS (
    SELECT
      O.DATE,
      SUM(P.PRICE * OD.QUANTITY) AS DAILY_REVENUE
    FROM
      ORDER_DETAIL OD
      JOIN ORDERS O ON OD.ORDER_ID = O.ORDER_ID
      JOIN PIZZA P ON P.PIZZA_ID = OD.PIZZA_ID
    GROUP BY
      O.DATE
  )
SELECT
  DATE,
  ROUND(DAILY_REVENUE::NUMERIC, 2),
  ROUND(
    SUM(DAILY_REVENUE) OVER (
      ORDER BY
        DATE
    )::NUMERIC,
    2
  ) AS CUMULATIVE_REVENUE
FROM
  A
ORDER BY
  DATE;
```



OUTPUT

	date date	round numeric	cumulative_revenue numeric
1	2015-01-01	2713.85	2713.85
2	2015-01-02	2731.90	5445.75
3	2015-01-03	2662.40	8108.15
4	2015-01-04	1755.45	9863.60
5	2015-01-05	2065.95	11929.55
6	2015-01-06	2428.95	14358.50
7	2015-01-07	2202.20	16560.70
8	2015-01-08	2838.35	19399.05
9	2015-01-09	2127.35	21526.40
10	2015-01-10	2463.95	23990.35
11	2015-01-11	1872.30	25862.65
12	2015-01-12	1919.05	27781.70
13	2015-01-13	2049.60	29831.30
14	2015-01-14	2527.40	32358.70
15	2015-01-15	1984.80	34343.50
16	2015-01-16	2594.15	36937.65
17	2015-01-17	2064.10	39001.75

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

```
13--Determine the top 3 most ordered pizza types based on revenue for each pizza category.
WITH
ORDER_D AS (
  SELECT
    PT.CATEGORY,
    PT.NAME,
    SUM(P.PRICE * OD.QUANTITY) AS REVENUE
  FROM
    PIZZA P
    JOIN PIZZA_TYPE PT ON P.PIZZA_TYPE = PT.PIZZA_TYPE_ID
    JOIN ORDER_DETAIL OD ON OD.PIZZA_ID = P.PIZZA_ID
  GROUP BY
    PT.CATEGORY,
    PT.NAME
),
RANKED_PIZZA AS (
  SELECT
    NAME,
    CATEGORY,
    REVENUE,
    RANK() OVER (
      PARTITION BY
        CATEGORY
      ORDER BY
        REVENUE DESC
    ) AS RN
  FROM
    ORDER_D
)
SELECT
  NAME,
  CATEGORY,
  REVENUE,
  RN
FROM
  RANKED_PIZZA
WHERE
  RN <= 3
```



OUTPUT

	name	category	revenue	rn
	character varying (100)	character varying (100)	double precision	bigint
1	The Thai Chicken Pizza	Chicken	43434.25	1
2	The Barbecue Chicken Pizza	Chicken	42768	2
3	The California Chicken Piz...	Chicken	41409.5	3
4	The Classic Deluxe Pizza	Classic	38180.5	1
5	The Hawaiian Pizza	Classic	32273.25	2
6	The Pepperoni Pizza	Classic	30161.75	3
7	The Spicy Italian Pizza	Supreme	34831.25	1
8	The Italian Supreme Pizza	Supreme	33476.75	2
9	The Sicilian Pizza	Supreme	30940.5	3
10	The Four Cheese Pizza	Veggie	32265.70000000065	1

Pizza Sales Presentation

THANK YOU FOR ATTENTION

See You Next