



PIZZA SALES ANALYSIS

USING
SQL, EXCEL & POWER BI

BY ASHISH DEEP SEN

All
About
Pizza
Sales !!

ABOUT -

**THIS PROJECT IS BASED ON A PIZZA SALES DATASET
CONTAINING FOUR TABLES – ORDERS , ORDER_DETAILS,
PIZZAS, AND PIZZA_TYPES.**

THE MAIN GOAL WAS TO :

- SOLVE BUSINESS QUESTIONS USING SQL (ORDERS, REVENUE, TOP PIZZAS, BUSIEST HOURS, CATEGORIES).**
- USE EXCEL FOR INITIAL DATA CLEANING, PIVOT SUMMARIES, AND QUICK CHECKS.**
- BUILD AN INTERACTIVE POWER BI DASHBOARD TO VISUALIZE SALES INSIGHTS.**



PROBLEM AND DATASET -

GOAL - ANALYZE SALES TO ANSWER KEY BUSINESS QUESTIONS

- RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED
- CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES
- IDENTIFY THE HIGHEST-PRICED PIZZA
- IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED
- LIST THE TOP 5 MOST ORDERED PIZZA TYPES ALONG WITH THEIR QUANTITIES
- FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED
- DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY
- FIND THE CATEGORY-WISE DISTRIBUTION OF PIZZAS
- GROUP THE ORDERS BY DATE AND CALCULATE THE AVERAGE NUMBER OF PIZZAS ORDERED PER DAY
- DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE
- CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE
- ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME
- DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE FOR EACH PIZZA CATEGORY

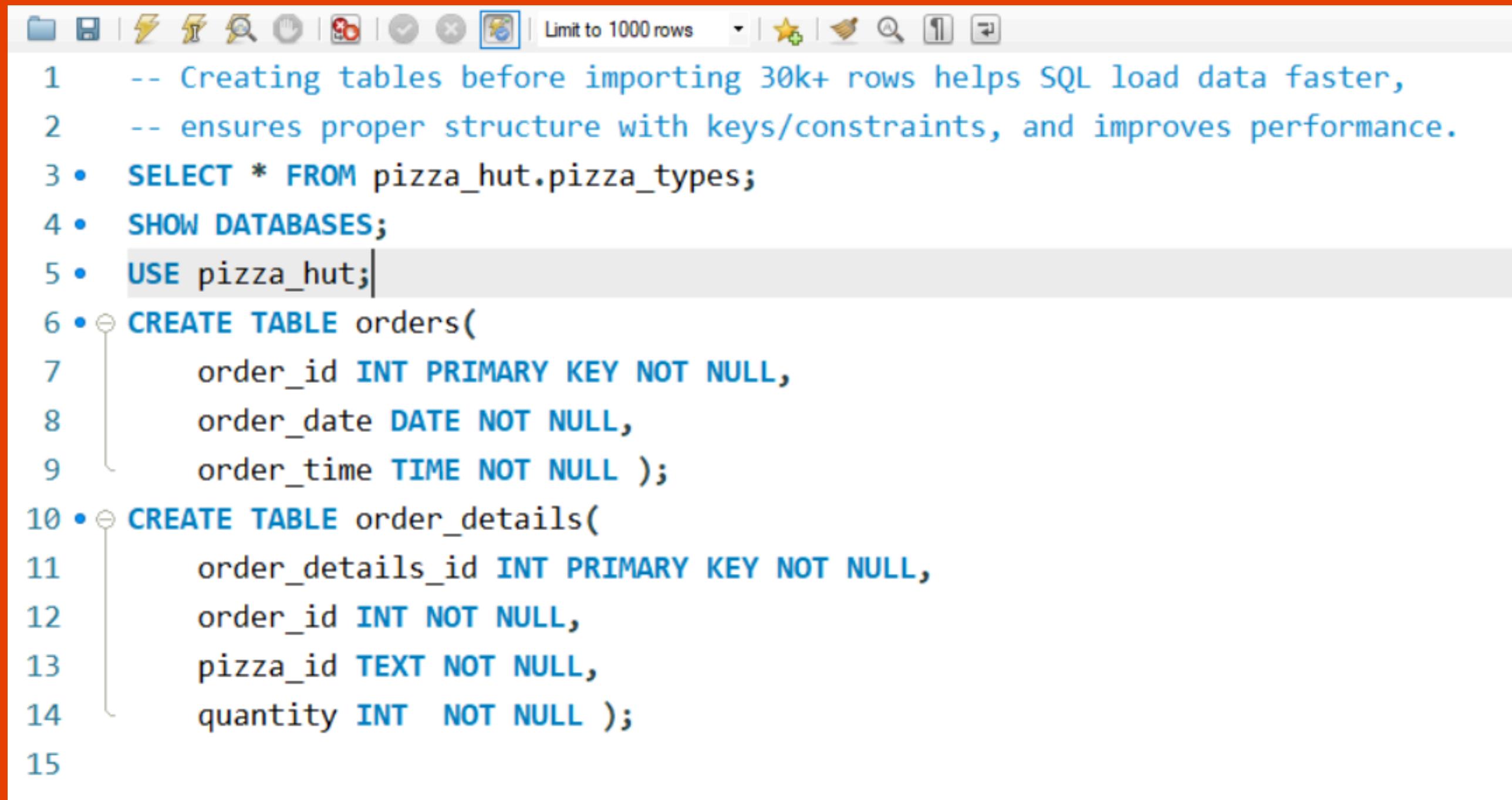
DATASET CONTAINS 4 TABLES:

- ORDERS → ORDER_ID, DATE, TIME
- ORDER DETAILS → ORDER_DETAILS_ID, ORDER_ID, PIZZA_ID, QUANTITY
- PIZZAS → PIZZA_ID, PIZZA_TYPE_ID, SIZE, PRICE
- PIZZA TYPES → PIZZA_TYPE_ID, NAME, CATEGORY, INGREDIENTS



SQL QUERIES -

-- CREATING TABLES BEFORE IMPORTING 30K+ ROWS HELPS SQL LOAD DATA FASTER, ENSURES PROPER STRUCTURE WITH KEYS/CONSTRAINTS, AND IMPROVES PERFORMANCE.



```
1 -- Creating tables before importing 30k+ rows helps SQL load data faster,
2 -- ensures proper structure with keys/constraints, and improves performance.
3 • SELECT * FROM pizza_hut.pizza_types;
4 • SHOW DATABASES;
5 • USE pizza_hut;
6 • CREATE TABLE orders(
7     order_id INT PRIMARY KEY NOT NULL,
8     order_date DATE NOT NULL,
9     order_time TIME NOT NULL );
10 • CREATE TABLE order_details(
11     order_details_id INT PRIMARY KEY NOT NULL,
12     order_id INT NOT NULL,
13     pizza_id TEXT NOT NULL,
14     quantity INT NOT NULL );
15
```

SQL QUERIES -

QUESTION NO.- 1

```
1 -- Retrieve the total number of orders placed.--  
2  
3 • SELECT * FROM orders;  
4 • SELECT COUNT(order_id) FROM orders;  
5 • SELECT  
6     COUNT(order_id) AS Total_orders  
7     FROM  
8     orders;  
9  
10
```

OUTPUT -

Result Grid	
	Total_orders
▶	21350

QUESTION NO.- 3

```
-- Identify the highest-priced pizza.  
  
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;
```

OUTPUT -

Result Grid		
	name	price
▶	The Greek Pizza	35.95

QUESTION NO.- 5

```
-- List the top 5 most ordered pizza types along with their quantities.  
SELECT  
    pizza_types.name,  
    SUM(order_details.quantity) AS order_type_quantity  
FROM pizzas  
    JOIN order_details 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.name  
ORDER BY order_type_quantity DESC  
LIMIT 7;
```

OUTPUT -

	name	order_type_quantity
▶	The Classic Deluxe Pizza	2453
	The Barbecue Chicken Pizza	2432
	The Hawaiian Pizza	2422
	The Pepperoni Pizza	2418
	The Thai Chicken Pizza	2371
	The California Chicken Pizza	2370
	The Sicilian Pizza	1938

QUESTION NO.- 2

```
1 -- Calculate the total revenue generated from pizza sales.  
2  
3 • SELECT  
4     ROUND(SUM(order_details.quantity * pizzas.price),  
5             2) AS Total_Sales  
6     FROM  
7     order_details  
8     JOIN  
9     pizzas ON pizzas.pizza_id = order_details.pizza_id;  
10  
11 -- Select & CTRL+B For BEAUTY
```

OUTPUT -

Result Grid	
	Total_Sales
▶	817860.05

QUESTION NO.- 4

```
-- Identify the most common pizza size ordered.  
  
• SELECT  
    pizzas.size, COUNT(order_details.quantity) AS Count_Com  
FROM  
    order_details  
    JOIN  
    pizzas ON pizzas.pizza_id = order_details.pizza_id  
GROUP BY pizzas.size  
ORDER BY Count_Com DESC  
LIMIT 5  
;
```

OUTPUT -

Result Grid		
	size	Count_Com
▶	L	18526
	M	15385
	S	14137
	XL	544
	XXL	28





SQL QUERIES -

QUESTION NO.- 6

```
-- List the top 5 most ordered pizza types along with their quantities.  
• SELECT  
    pizza_types.name,  
    SUM(order_details.quantity) AS order_type_quantity  
FROM pizzas  
    JOIN order_details 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.name  
ORDER BY order_type_quantity DESC  
LIMIT 5;
```

OUTPUT -

Result Grid		Filter Rows:
	category	Category_Quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

QUESTION NO.- 7

```
-- Determine the distribution of orders by hour of the day.  
SELECT  
    HOUR(order_time) AS HOUR_, COUNT(order_id) AS orders_Count  
FROM  
    orders  
GROUP BY HOUR(order_time)  
ORDER BY orders_Count DESC;  
;
```

OUTPUT -

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

QUESTION NO.- 9

```
-- Group the orders by date and calculate the average  
-- number of pizzas ordered per day.  
SELECT ROUND(AVG(Quantities),0) AS AVERAGE_PIZZAS_PERDAY  
FROM  
    (SELECT orders.order_date, SUM(order_details.quantity) AS Quantities  
    FROM order_details  
    JOIN orders ON order_details.order_id = orders.order_id  
    GROUP BY orders.order_date) AS Order_quantities;
```

OUTPUT -

Result Grid		Filter Rows:
	AVERAGE_PIZZAS_PERDAY	
▶	138	

QUESTION NO.- 8

```
-- Join relevant tables to find the  
-- category-wise distribution of pizzas.  
SELECT  
    category, COUNT(name)  
FROM  
    pizza_types  
GROUP BY category;
```

OUTPUT -

	category	COUNT(name)
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9

SQL QUERIES -

QUESTION NO.- 10

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

SELECT pizza_types.name, SUM(order_details.quantity * pizzas.price) AS REVENUE
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.name
ORDER BY REVENUE DESC
LIMIT 3;
```

OUTPUT -

	name	REVENUE
▶	The Thai Chicken Pizza	43434.25
	The Barbecue Chicken Pizza	42768
	The California Chicken Pizza	41409.5

QUESTION NO.- 11

```
-- Calculate the percentage contribution of each pizza type to total revenue.

SELECT pizza_types.category,
ROUND((SUM(order_details.quantity * pizzas.price) / (SELECT
    ROUND(SUM(order_details.quantity * pizzas.price),
    2) AS Total_Sales
    FROM order_details
    JOIN pizzas ON pizzas.pizza_id = order_details.pizza_id)) * 100,
1) AS REVENUE_PER
FROM order_details
    JOIN pizzas ON pizzas.pizza_id = order_details.pizza_id
    JOIN pizza_types ON pizzas.pizza_type_id = pizza_types.pizza_type_id
GROUP BY pizza_types.category ORDER BY REVENUE_PER DESC;
```

OUTPUT -

	category	REVENUE_PER
▶	Classic	26.9
	Supreme	25.5
	Chicken	24
	Veggie	23.7





SQL QUERIES -

QUESTION NO.- 6

OUTPUT -

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

SELECT order_date ,      ROUND(SUM(revenue) OVER (ORDER BY order_date) ,2)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 sales;
```

	order_date	cum_revenue
▶	2015-01-01	2713.85
	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	23990.35

QUESTION NO.- 9

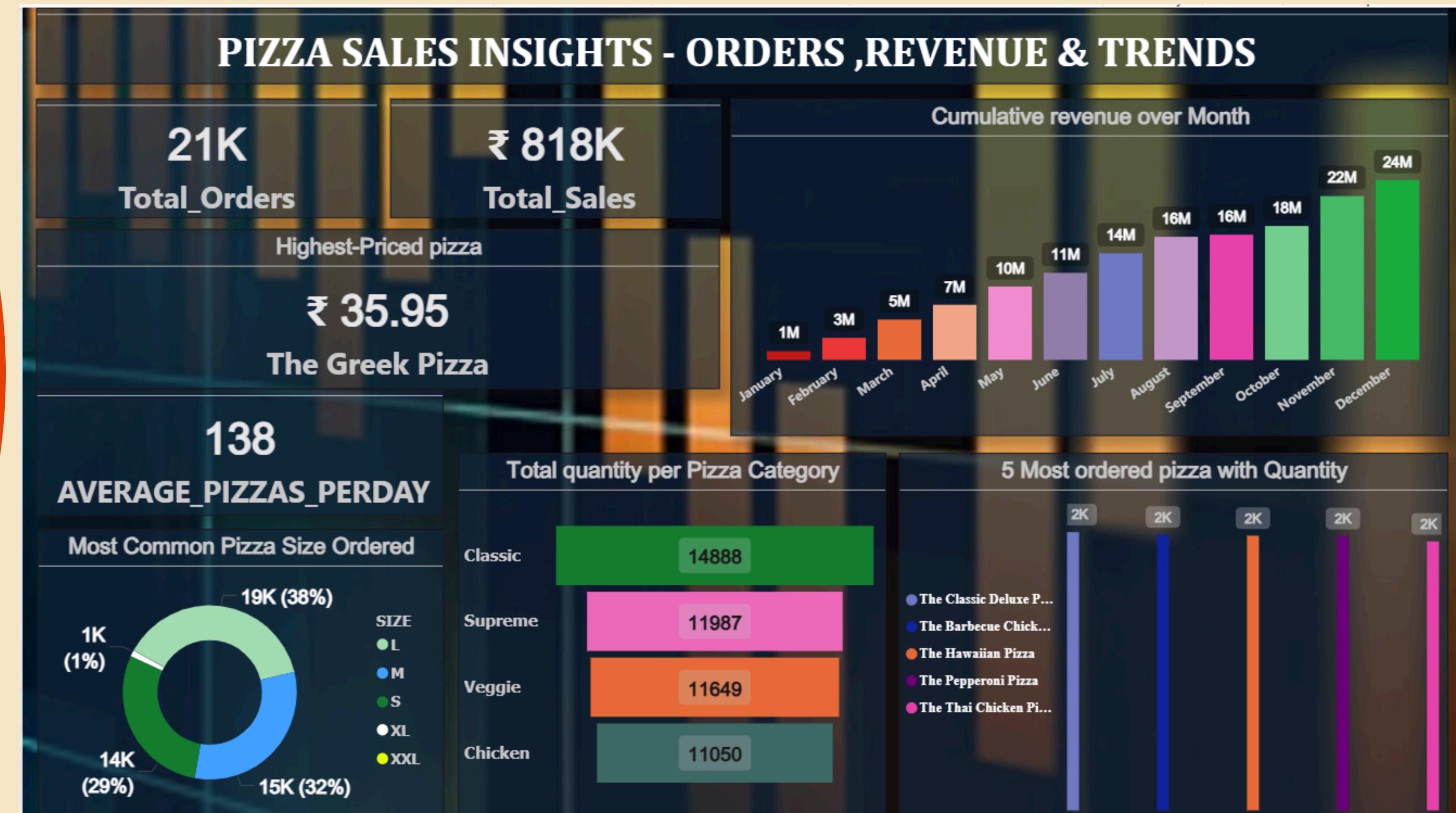
OUTPUT -

```
-- Determine the top 3 most ordered pizza types
-- based on revenue for each pizza category.

SELECT category ,name ,ROUND(Revenue,2) AS R_3_P_B_O_R
  FROM
(SELECT category ,name , Revenue ,
RANK() OVER(PARTITION BY category ORDER BY Revenue DESC) AS Rank_
  FROM
(SELECT
    pizza_types.category,
    pizza_types.name,
    SUM(order_details.quantity * pizzas.price) AS Revenue
  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 , pizza_types.name) AS a ) AS b WHERE Rank_ <=3 ;
```

	category	name	R_3_P_B_O_R
▶	Chicken	The Thai Chicken Pizza	43434.25
	Chicken	The Barbecue Chicken Pizza	42768
	Chicken	The California Chicken Pizza	41409.5
	Classic	The Classic Deluxe Pizza	38180.5
	Classic	The Hawaiian Pizza	32273.25
	Classic	The Pepperoni Pizza	30161.75
	Supreme	The Spicy Italian Pizza	34831.25
	Supreme	The Italian Supreme Pizza	33476.75
	Supreme	The Sicilian Pizza	30940.5
	Veggie	The Four Cheese Pizza	32265.7
	Veggie	The Mexicana Pizza	26780.75
	Veggie	The Five Cheese Pizza	26066.5

DASHBOARDS & INSIGHTS



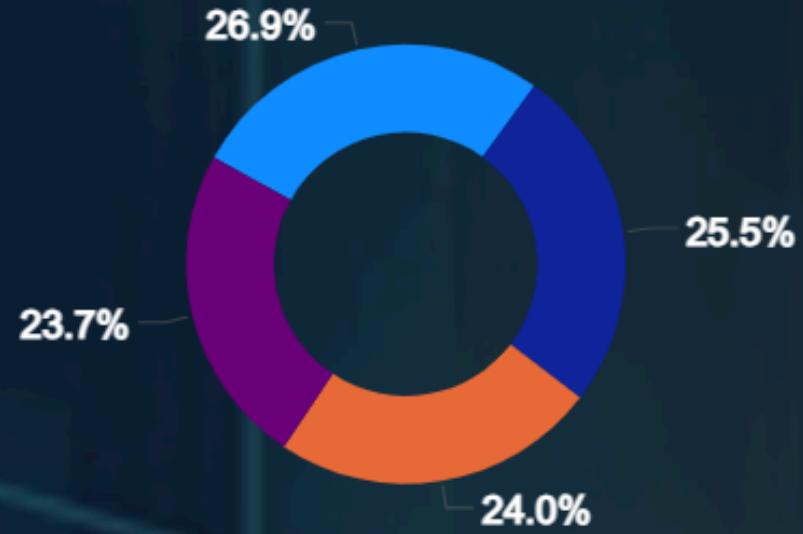
DASHBOARDS & INSIGHTS

From the hourly order distribution chart, it is clear that the busiest time for pizza orders is during the Afternoon Hours and evening hours (around 11 AM to 8 PM).

Top 3 Pizza Types By Revenue



% Contribution Per Pizza Type



Hour | Orders By Hours

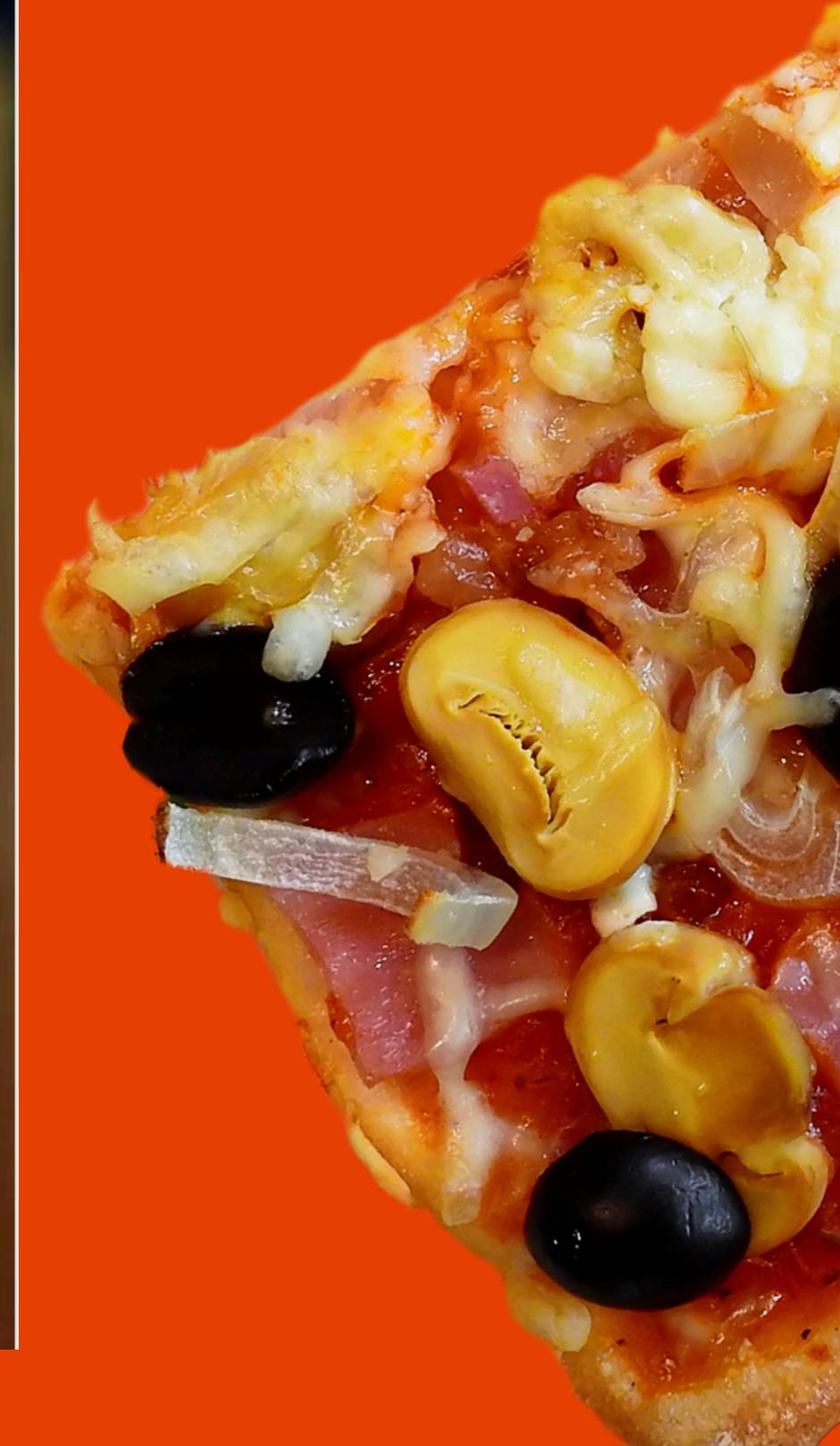
Hour	Orders By Hours
9:00	1
10:00	8
11:00	1231
12:00	2520
13:00	2455
14:00	1472
15:00	1468
16:00	1920
17:00	2336
18:00	2399
19:00	2009
20:00	1642
21:00	1198
22:00	663
23:00	28
Total	21350

CATEGORY

All

Most Ordered Pizza Types - Based on revenue

Pizza_Types	Based_On_Revenues
The Thai Chicken Pizza	₹ 43,434
The Barbecue Chicken Pizza	₹ 42,768
The California Chicken Pizza	₹ 41,410
The Classic Deluxe Pizza	₹ 38,181
The Spicy Italian Pizza	₹ 34,831
The Italian Supreme Pizza	₹ 33,477
The Hawaiian Pizza	₹ 32,273
The Four Cheese Pizza	₹ 32,266
The Sicilian Pizza	₹ 30,941
The Pepperoni Pizza	₹ 30,162
The Mexicana Pizza	₹ 26,781
The Five Cheese Pizza	₹ 26,067



CONCLUSION

- THE ANALYSIS GAVE A CLEAR PICTURE OF PIZZA SALES PERFORMANCE USING SQL, EXCEL, AND POWER BI .
- SQL HELPED ANSWER IMPORTANT BUSINESS QUESTIONS LIKE REVENUE, TOP PIZZAS, BUSIEST HOURS, AND CUSTOMER PREFERENCES.
- EXCEL SUPPORTED DATA CLEANING AND QUICK VALIDATION.
- POWER BI DASHBOARD PROVIDED INTERACTIVE VISUALS FOR BETTER DECISION-MAKING.
- KEY INSIGHTS SHOWED AFTERNOON AND EVENING HOURS (11AM –8 PM) AS BUSIEST, CLASSIC PIZZAS AS THE MOST POPULAR CATEGORY, AND A FEW PIZZAS CONTRIBUTING THE HIGHEST SHARE OF REVENUE.
- THIS PROJECT HIGHLIGHTS HOW COMBINING DATA CLEANING, SQL ANALYSIS, AND VISUALIZATION CAN TURN RAW DATA INTO VALUABLE BUSINESS INSIGHTS.



THANK YOU



[HTTPS://GITHUB.COM/ASHISHDS-09](https://github.com/ASHISHDS-09)



[HTTPS://WWW.LINKEDIN.COM/IN/ASHISH-DEEP-SEN09/](https://www.linkedin.com/in/ASHISH-DEEP-SEN09/)



ASHISHDEEP800@gmail.com



Thank you for viewing my project! Full SQL scripts, Excel file, and dashboard are available on GitHub. .