SQL PROJECT ON PIZZA SALES

WELCOME TO MY SQL PROJECT PRESENTATION

MY NAME IS GYANENDRA MAURYA

A BIT ABOUT ME

ASPIRING DATA ANALYST SKILLED IN SQL, EXCEL, POWER BI, AND PYTHON

EAGER TO SOLVE REAL BUSINESS PROBLEMS USING DATA

INTERNSHIP AT ISRO (IIRS) WITH EXPOSURE TO SATELLITE DATA FOR AGRICULTURE

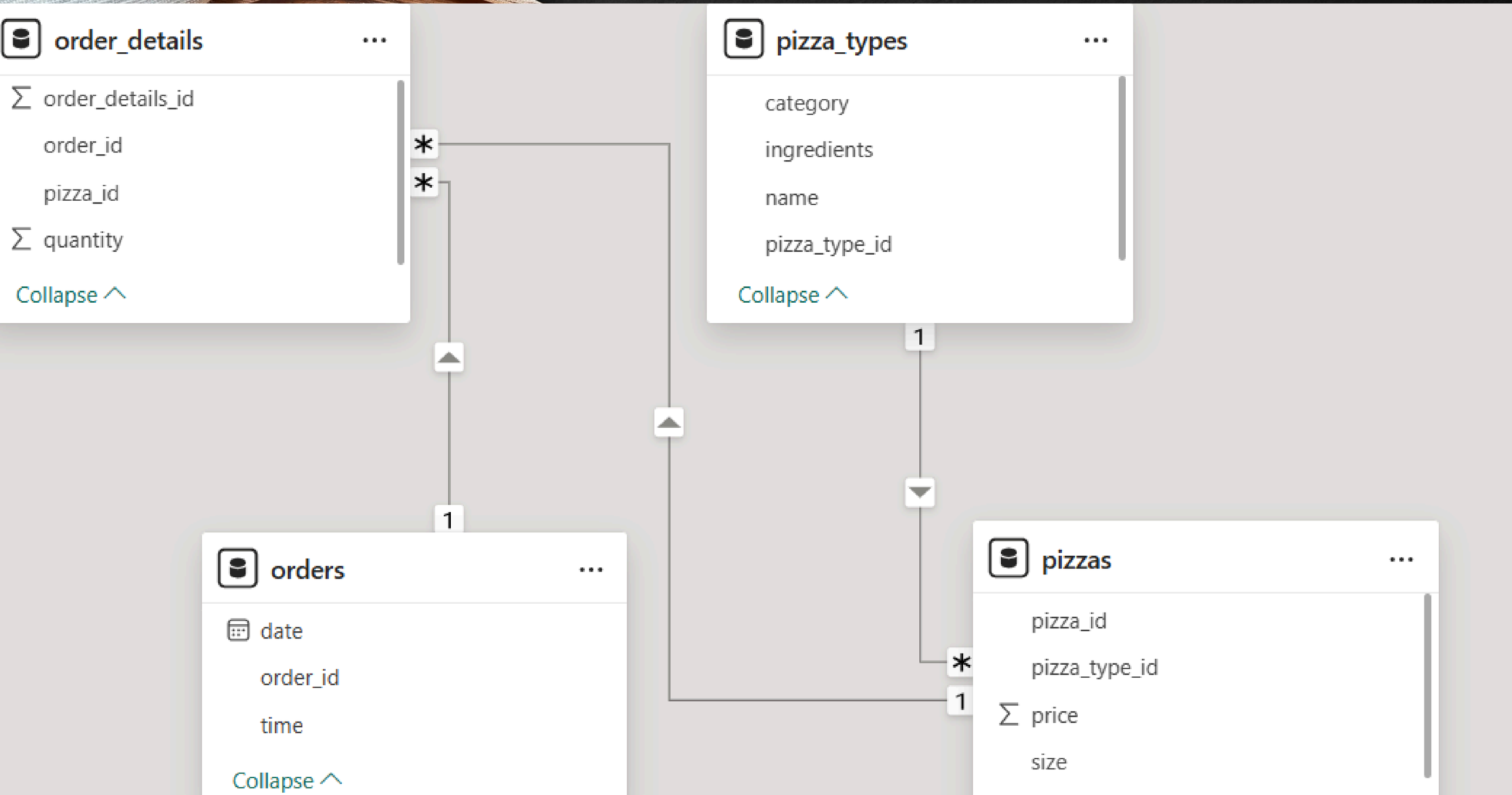
CONTINUOUSLY BUILDING HANDS-ON PROJECTS TO SHARPEN MY ANALYTICAL SKILLS

ABOUT THIS PROJECT

THIS PROJECT, I WORKED ON A PIZZA SALES DATASET TO SOLVE KEY BUSINESS QUESTIONS COMMONLY ASKED BY PIZZA COMPANIES, SUCH AS:

- WHAT ARE THE TOP-SELLING PIZZAS BY REVENUE AND QUANTITY?
- WHICH PIZZA CATEGORIES OR SIZES ARE MOST POPULAR?
- WHAT ARE THE PEAK SALES HOURS OR BEST-PERFORMING DAYS?
- HOW CAN THE COMPANY OPTIMIZE INVENTORY OR BOOST SALES?

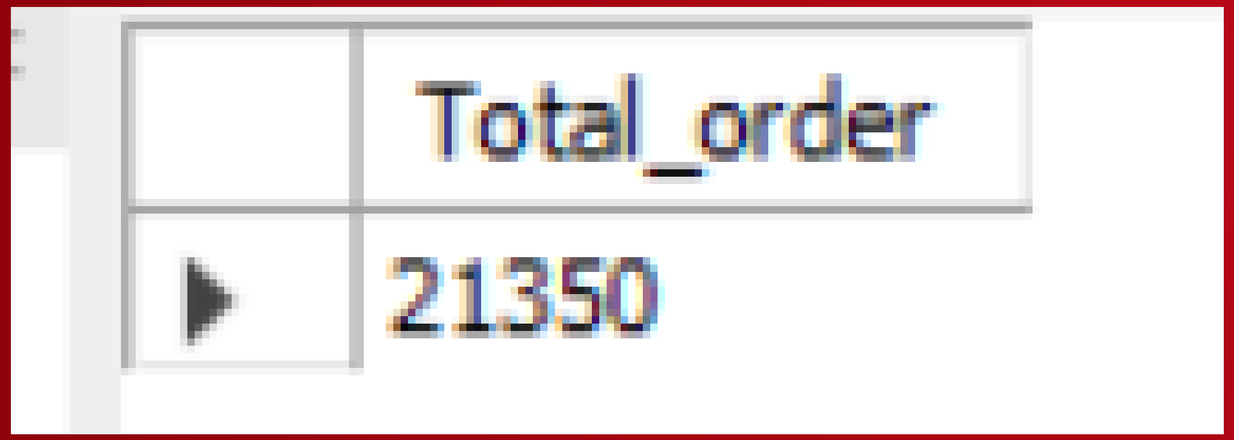
MODEL VIEW OF MY SQL DATA



MY QUERIES

1. RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED

```
SELECT
COUNT(ORDER_ID) AS TOTAL_ORDER
FROM
ORDERS;
```

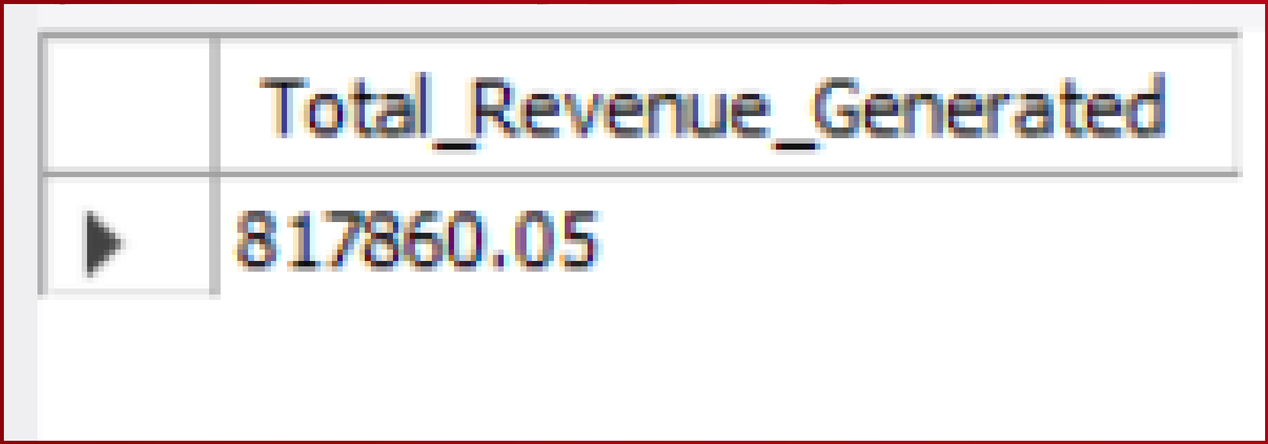


Total_order
21350

MY QUERIES

2. CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

```
SELECT
ROUND(SUM(ORDER_DETAILS.QUANTITY * PIZZAS.PRICE),
2) AS TOTAL_REVENUE_GENERATED
FROM
PIZZAS
INNER JOIN
ORDER_DETAILS ON ORDER_DETAILS.PIZZA_ID =
PIZZAS.PIZZA_ID;
```



	Total_Revenue_Generated
▶	817860.05

MY QUERIES

3.IDENTIFY THE HIGHEST-PRICED PIZZA.

```
SELECT
MAX(PRICE) AS HIGHEST_PRICE_PIZZA
FROM
PIZZAS;
```

	Highest_price_pizza
▶	35.95

MY QUERIES

4. IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

```
SELECT
    PIZZAS.SIZE AS SIZE,
    COUNT(ORDER_DETAILS.QUANTITY) AS TOTAL_COUNT
FROM
    ORDER_DETAILS
        INNER JOIN
    PIZZAS ON ORDER_DETAILS.PIZZA_ID = PIZZAS.PIZZA_ID
GROUP BY SIZE
ORDER BY TOTAL_COUNT DESC;
```

Size	Total_count
L	18526
M	15385
S	14137
XL	544
XXL	28

MY QUERIES

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

```
SELECT
    PIZZA_TYPES.NAME,
    SUM(ORDER_DETAILS.QUANTITY) AS TOTAL_QUANTITIES
FROM
    PIZZA_TYPES
        INNER JOIN
    PIZZAS ON PIZZA_TYPES.PIZZA_TYPE_ID = PIZZAS.PIZZA_TYPE_ID
        INNER JOIN
    ORDER_DETAILS ON ORDER_DETAILS.PIZZA_ID = PIZZAS.PIZZA_ID
GROUP BY NAME
ORDER BY TOTAL_QUANTITIES DESC
LIMIT 5;
```

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

MY QUERIES

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

```
SELECT
    PIZZA_TYPES.CATEGORY,
    SUM(ORDER_DETAILS.QUANTITY) AS TOTAL_QUANTITY
FROM
    PIZZA_TYPES
        JOIN
    PIZZAS ON PIZZA_TYPES.PIZZA_TYPE_ID = PIZZAS.PIZZA_TYPE_ID
        JOIN
    ORDER_DETAILS ON ORDER_DETAILS.PIZZA_ID = PIZZAS.PIZZA_ID
    GROUP BY CATEGORY
    ORDER BY TOTAL_QUANTITY DESC;
```

category	total_quantity
Classic	14888
Veggie	11649
Supreme	11987
Chicken	11050

MY QUERIES

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

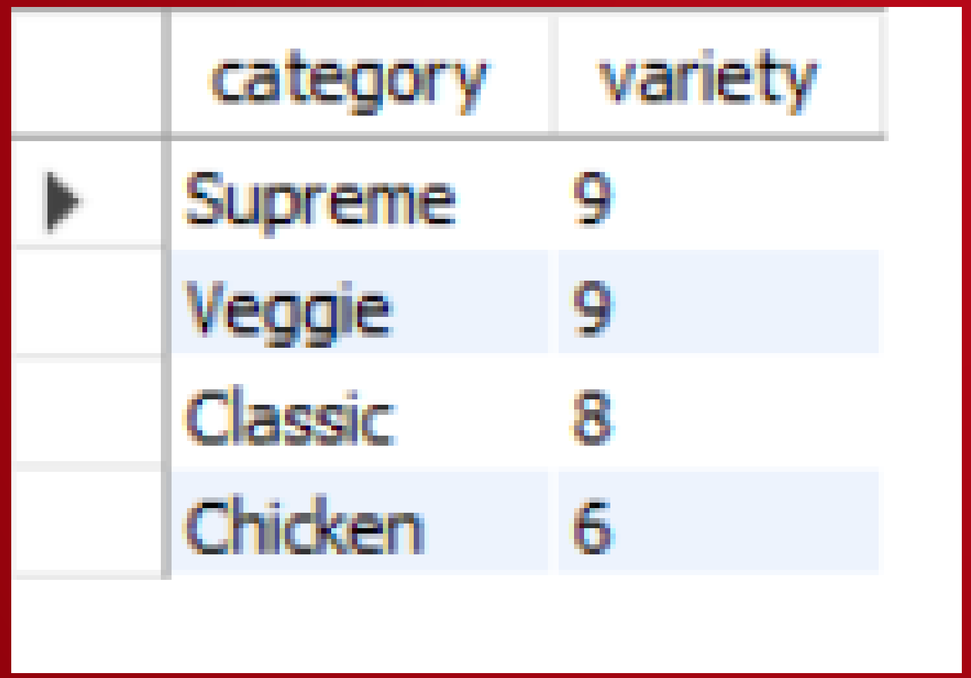
```
SELECT
HOUR(ORDER_TIME) AS ORDER_HOUR,
COUNT(ORDER_ID) AS ORDER_QUANTITY
FROM
ORDERS
GROUP BY ORDER_HOUR
ORDER BY ORDER_QUANTITY DESC;
```

order_hour	order_quantity
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
10	8
9	1

MY QUERIES

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

```
SELECT
CATEGORY, COUNT(NAME) AS VARIETY
FROM
PIZZA_TYPES
GROUP BY CATEGORY
ORDER BY VARIETY DESC;
```



category	variety
Supreme	9
Veggie	9
Classic	8
Chicken	6

MY QUERIES

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

```
SELECT
ROUND(AVG(AVG_QUANTITY), 0) AS AVERAGE
FROM
(SELECT
ORDERS.ORDER_DATE,
SUM(ORDER_DETAILS.QUANTITY) AS AVG_QUANTITY
FROM
ORDERS
JOIN ORDER_DETAILS ON ORDER_DETAILS.ORDER_ID = ORDERS.ORDER_ID
GROUP BY ORDERS.ORDER_DATE) AS ORDER_QUANTITY;
```

	average
▶	138

MY QUERIES

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

```
SELECT
    PIZZA_TYPES.NAME AS PIZZA_NAME,
    ROUND(SUM(ORDER_DETAILS.QUANTITY * PIZZAS.PRICE),
            2) AS TOTAL_REVENUE
FROM
    PIZZAS
        INNER JOIN
    PIZZA_TYPES ON PIZZA_TYPES.PIZZA_TYPE_ID = PIZZAS.PIZZA_TYPE_ID
        INNER JOIN
    ORDER_DETAILS ON PIZZAS.PIZZA_ID = ORDER_DETAILS.PIZZA_ID
GROUP BY PIZZA_NAME
ORDER BY TOTAL_REVENUE DESC
LIMIT 3;
```

pizza_name	total_revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5

MY QUERIES

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

```
SELECT
    PIZZA_TYPES.CATEGORY AS PIZZA_CATEGORY,
    ROUND((SUM(ORDER_DETAILS.QUANTITY * PIZZAS.PRICE) / (SELECT
            SUM(ORDER_DETAILS.QUANTITY * PIZZAS.PRICE) AS TOTAL_REVENUE
        FROM
            ORDER_DETAILS
                JOIN
            PIZZAS ON ORDER_DETAILS.PIZZA_ID = PIZZAS.PIZZA_ID) * 100),
        2) AS PERCENT
    FROM
        PIZZAS
            JOIN
    PIZZA_TYPES ON PIZZAS.PIZZA_TYPE_ID = PIZZA_TYPES.PIZZA_TYPE_ID
            JOIN
    ORDER_DETAILS ON ORDER_DETAILS.PIZZA_ID = PIZZAS.PIZZA_ID
    GROUP BY PIZZA_CATEGORY
    ORDER BY PERCENT DESC;
```

	pizza_category	percent
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68

MY QUERIES

12 . 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 ORDER_DETAILS.ORDER_ID = ORDERS.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
2015-01-11	25862.65
2015-01-12	27781.7
2015-01-13	29831.3
2015-01-14	32358.7
2015-01-15	34343.5
2015-01-16	36937.65
2015-01-17	39001.75
2015-01-18	40978.6
2015-01-19	43365.75
2015-01-20	45763.65
2015-01-21	47804.2
2015-01-22	50300.9
2015-01-23	52724.6

MY QUERIES

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

```
SELECT CATEGORY , NAME , REVENUE FROM (SELECT CATEGORY , NAME , REVENUE ,
RANK() OVER(PARTITION  BY CATEGORY ORDER BY REVENUE DESC) AS RN FROM
(SELECT PIZZA_TYPES.CATEGORY , PIZZA_TYPES.NAME , 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 PIZZA_TYPES.CATEGORY , PIZZA_TYPES.NAME) AS A) AS B
WHERE RN <= 3;
```

category	name	revenue
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	32265.7000
Veggie	The Mexicana Pizza	26780.75
Veggie	The Five Cheese Pizza	26066.5

MY CONTACT DETAILS

LINKEDIN.COM/IN/GYANENDRA-MAURYA-494205348

8381936236

GKMAURYA2306@GMAIL.COM

THANK YOU!