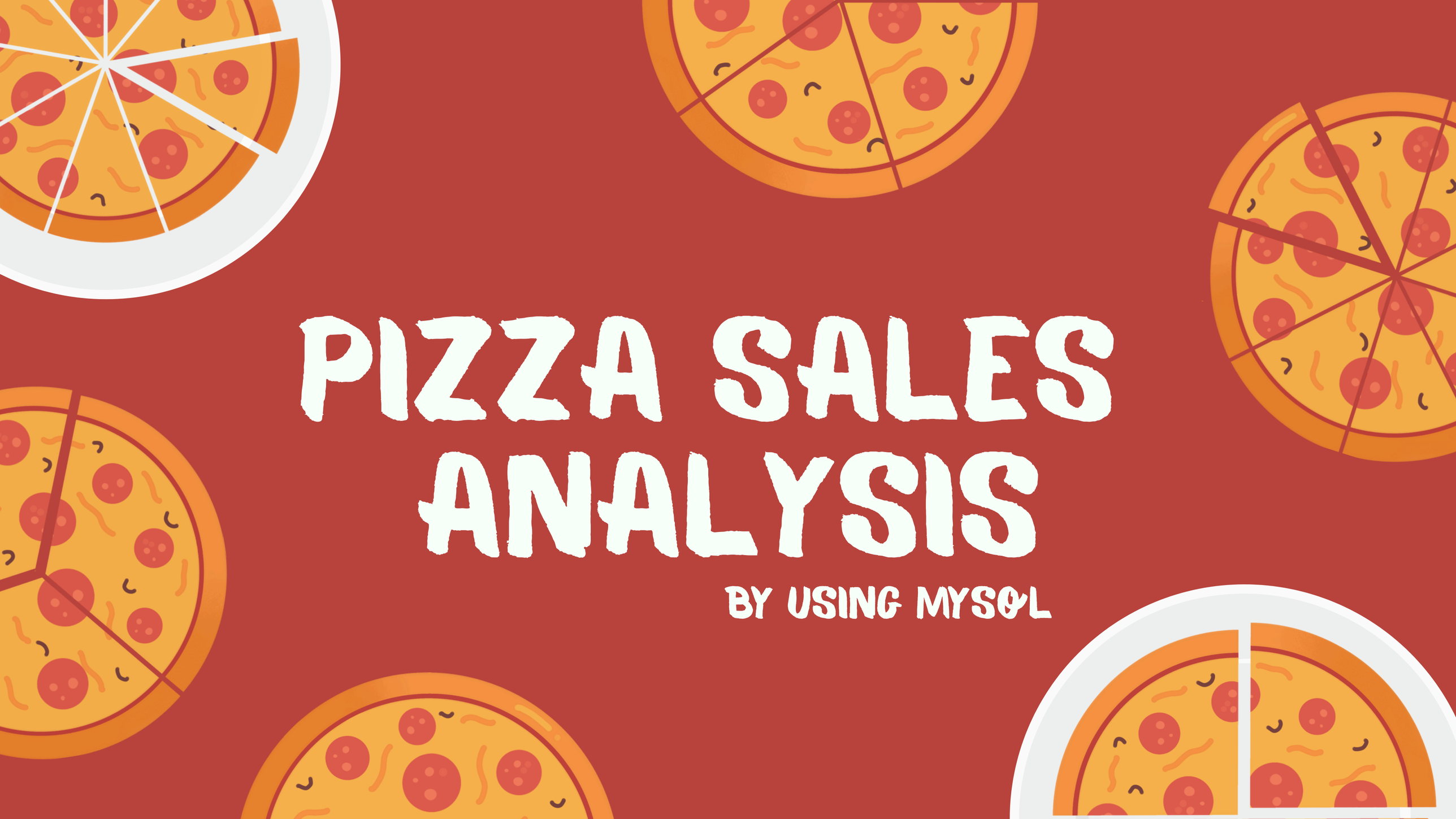

PIZZA SALES ANALYSIS
BY USING MYSQL

OBJECTIVES

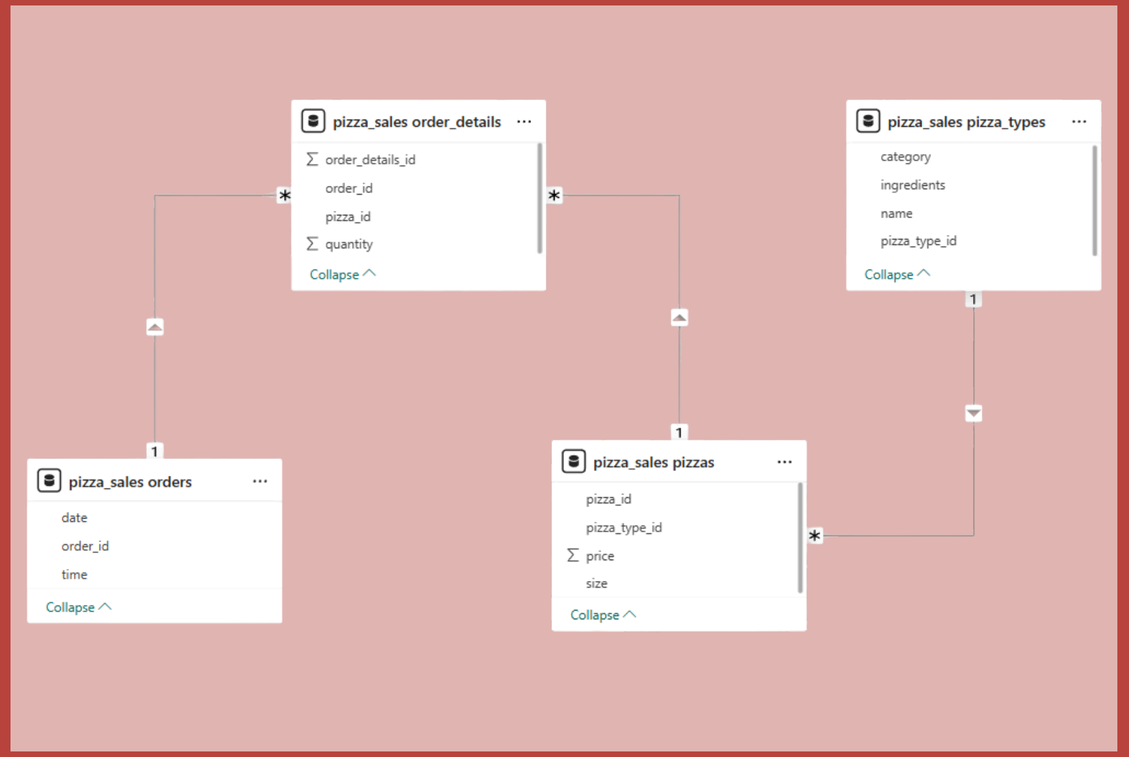
ANALYZE PIZZA SALES DATASET
USING SQL TO FIND:

1. REVENUE TRENDS
2. BEST-SELLING PIZZAS
3. PEAK ORDER TIMES
4. CATEGORY & SIZE PERFORMANCE

DATASET

TBLES USED

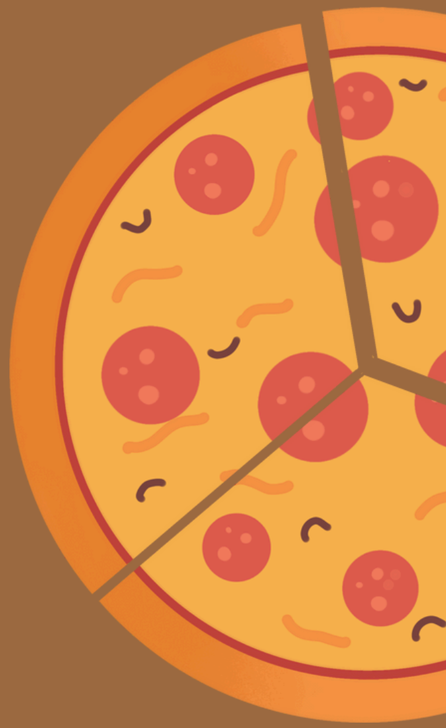
1. ORDERS → ORDER_ID, DATE, TIME
2. ORDER_DETAILS → ORDER_ID, PIZZA_ID, QUANTITY
3. PIZZAS → PIZZA_ID, PIZZA_TYPE_ID, SIZE, PRICE
4. PIZZA_TYPES → PIZZA_TYPE_ID, NAME, CATEGORY, INGREDIENTS



1. TOTAL NO OF ORDER PLACED

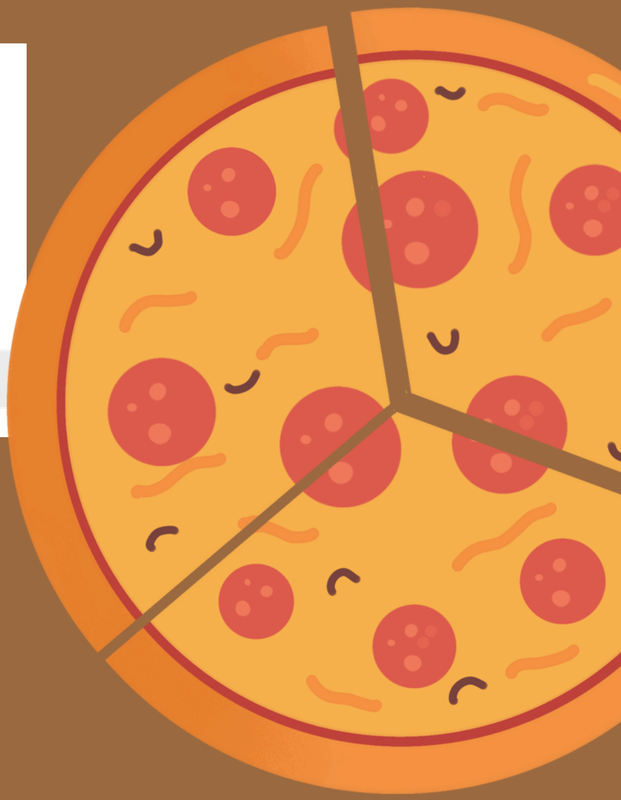
```sql
--  1.) total number of orders placed.

    SELECT COUNT(ORDER_ID)  AS TOTAL_ORDERS FROM ORDER_DETAILS;
```

TOTAL_ORDERS
48620



2. TOTAL REVENUE GENERATED FROM PIZZA SALES.

```sql
-- 2.) total revenue generated from pizza sales.

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



	TOTAL_REVENUE
▶	817860.05

3. THE HIGHEST-PRICED PIZZA.

```sql
-- 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 ;
```

	NAME	PRICE
▶	The Greek Pizza	35.95

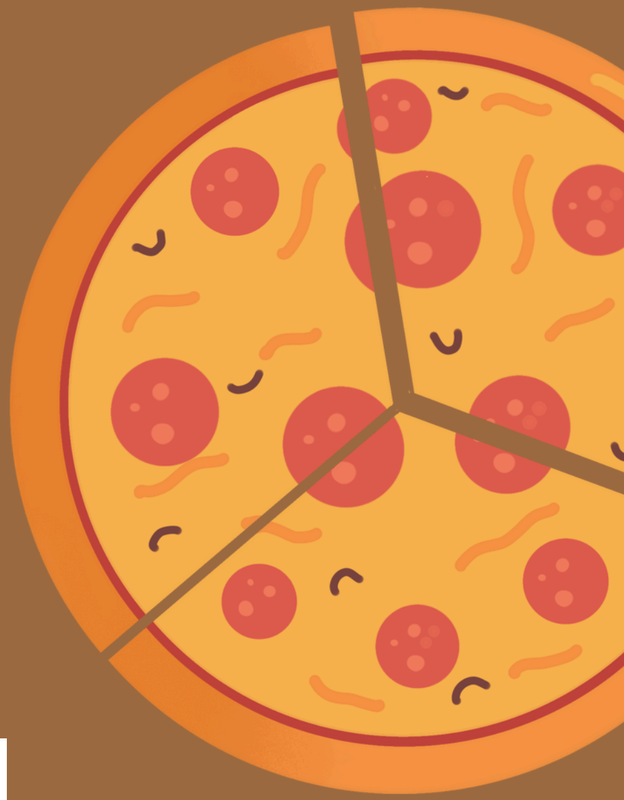
4. THE MOST COMMON PIZZA SIZE ORDERED.

```sql
--  4.) Identify the most common pizza size ordered.

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

SIZE	ORDER_COUNT
L	18526
M	15385
S	14137
XL	544
XXL	28

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

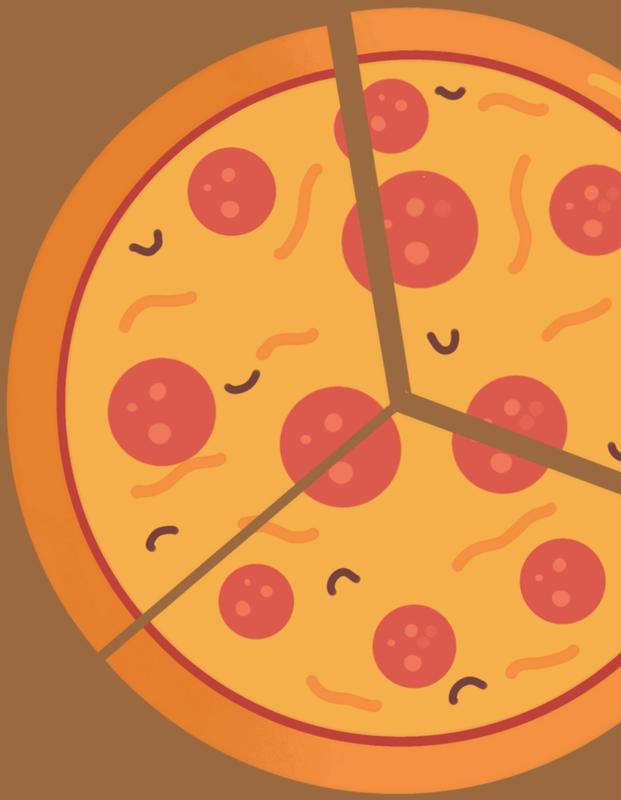
```sql
-- 5.) List the top 5 most ordered pizza types along with their quantities.

    SELECT
        PIZZA_TYPES.NAME ,
        PIZZAS.PIZZA_TYPE_ID  ,
        SUM(ORDER_DETAILS.QUANTITY) AS QUANTITY

    FROM PIZZAS JOIN ORDER_DETAILS
    ON PIZZAS.PIZZA_ID = ORDER_DETAILS.PIZZA_ID

    JOIN PIZZA_TYPES            -- 3 rd table joined.
    ON PIZZA_TYPES.PIZZA_TYPE_ID = PIZZAS.PIZZA_TYPE_ID

    GROUP BY PIZZAS.PIZZA_TYPE_ID, PIZZA_TYPES.NAME
    ORDER BY QUANTITY DESC LIMIT 5;
```



	NAME	PIZZA_TYPE_ID	QUANTITY
▶	The Classic Deluxe Pizza	classic_dlx	2453
	The Barbecue Chicken Pizza	bbq_ckn	2432
	The Hawaiian Pizza	hawaiian	2422
	The Pepperoni Pizza	pepperoni	2418
	The Thai Chicken Pizza	thai_ckn	2371

6. TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED

```sql
-- 6.) Join the necessary tables to find the total quantity of each pizza category order

    SELECT
        PIZZA_TYPES.CATEGORY ,
        SUM(ORDER_DETAILS.QUANTITY) AS QUANTITY

    FROM PIZZAS JOIN ORDER_DETAILS
    ON PIZZAS.PIZZA_ID = ORDER_DETAILS.PIZZA_ID

    JOIN PIZZA_TYPES            -- 3 rd table joined.
    ON PIZZA_TYPES.PIZZA_TYPE_ID = PIZZAS.PIZZA_TYPE_ID

    GROUP BY PIZZA_TYPES.CATEGORY
    ORDER BY QUANTITY DESC ;
```



CATEGORY	QUANTITY
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

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

```sql
SELECT
    HOUR(TIME) AS HOURS ,
    COUNT(ORDER_ID) AS ORDER_COUNT
FROM ORDERS
GROUP BY HOUR(TIME);
```

HOURS	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

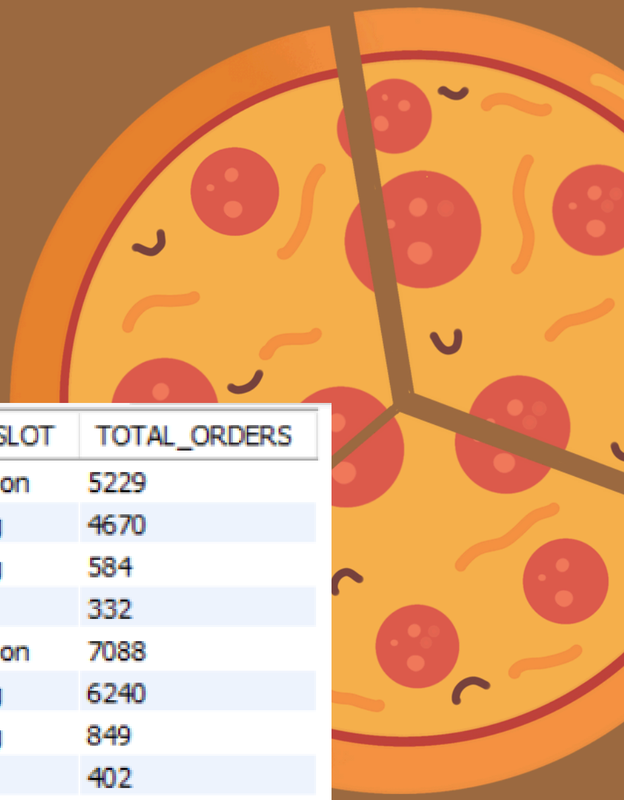
8. THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

```sql
SELECT
    HOUR(TIME) AS HOURS ,
    COUNT(ORDER_ID) AS ORDER_COUNT
FROM ORDERS
GROUP BY HOUR(TIME);
```

HOURS	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

9.EACH PIZZA CATEGORY ORDERS RECEIVE DURING DIFFERENT TIME SLOTS (MORNING, AFTERNOON, EVENING, NIGHT) IN A DAY

```sql
SELECT
     PIZZA_TYPES.CATEGORY,
     CASE
         WHEN HOUR(ORDERS.TIME) BETWEEN 6 AND 11 THEN 'Morning'
         WHEN HOUR(ORDERS.TIME) BETWEEN 12 AND 16 THEN 'Afternoon'
         WHEN HOUR(ORDERS.TIME) BETWEEN 17 AND 21 THEN 'Evening'
         ELSE 'Night'
     END AS TIME_SLOT,
 COUNT(ORDER_DETAILS.ORDER_ID) AS TOTAL_ORDERS

 FROM PIZZA_TYPES  JOIN PIZZAS
 ON PIZZAS.PIZZA_TYPE_ID = PIZZA_TYPES.PIZZA_TYPE_ID

 JOIN ORDER_DETAILS
 ON ORDER_DETAILS.PIZZA_ID = PIZZAS.PIZZA_ID

 JOIN ORDERS
 ON ORDERS.ORDER_ID = ORDER_DETAILS.ORDER_ID

 GROUP BY PIZZA_TYPES.CATEGORY , TIME_SLOT
 ORDER BY PIZZA_TYPES.CATEGORY , TOTAL_ORDERS DESC;
```



CATEGORY	TIME_SLOT	TOTAL_ORDERS
Chicken	Afternoon	5229
Chicken	Evening	4670
Chicken	Morning	584
Chicken	Night	332
Classic	Afternoon	7088
Classic	Evening	6240
Classic	Morning	849
Classic	Night	402
Supreme	Afternoon	5699
Supreme	Evening	5035
Supreme	Morning	671
Supreme	Night	372
Veggie	Afternoon	5606
Veggie	Evening	4922
Veggie	Morning	589
Veggie	Night	332

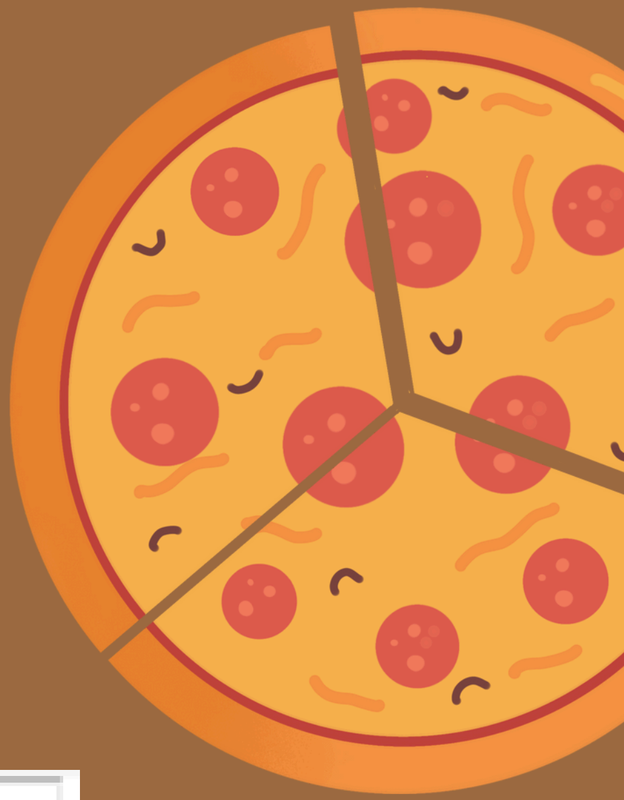
10. GROUP THE ORDERS BY DATE AND CALCULATE THE AVERAGE NUMBER OF PIZZAS ORDERED PER DAY.

```sql
SELECT ROUND(AVG(QUANTITY),0) AS AVG_PIZZAS_ORDERD_PER_DAY
FROM

(SELECT ORDERS.DATE  ,  SUM(ORDER_DETAILS.QUANTITY) AS QUANTITY
FROM ORDERS JOIN ORDER_DETAILS
ON ORDERS.ORDER_ID = ORDER_DETAILS.ORDER_ID
GROUP BY ORDERS.DATE) AS ORDER_QUANTITY ;
```

AVG_PIZZAS_ORDERD_PER_DAY
138

11. TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE.

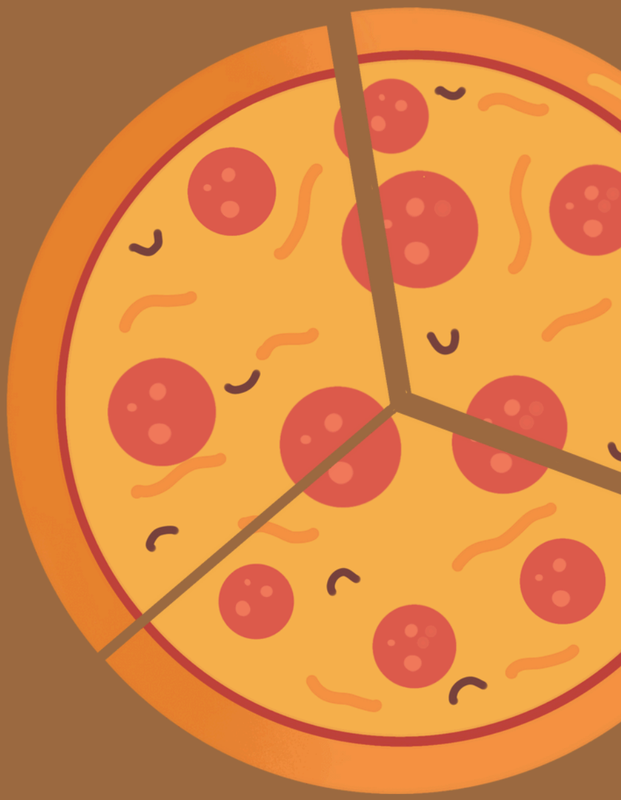
```sql
SELECT PIZZA_TYPES.NAME ,
SUM(ORDER_DETAILS.QUANTITY * PIZZAS.PRICE) AS REVENUE
FROM PIZZAS JOIN PIZZA_TYPES
ON PIZZAS.PIZZA_TYPE_ID = PIZZA_TYPES.PIZZA_TYPE_ID

JOIN ORDER_DETAILS
ON ORDER_DETAILS.PIZZA_ID = PIZZAS.PIZZA_ID

GROUP BY PIZZA_TYPES.NAME
ORDER BY REVENUE DESC LIMIT 3 ;
```



NAME	REVENUE
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5

12. THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE.

```sql
SELECT PIZZA_TYPES.CATEGORY ,
ROUND(SUM(ORDER_DETAILS.QUANTITY * PIZZAS.PRICE) / (SELECT
        ROUND(SUM(ORDER_DETAILS.QUANTITY * PIZZAS.PRICE),2) AS TOTAL_REVENUE
    FROM ORDER_DETAILS JOIN PIZZAS
    ON PIZZAS.PIZZA_ID = ORDER_DETAILS.PIZZA_ID ) * 100,2) AS REVENUE

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_TYPES.CATEGORY
ORDER BY REVENUE DESC ;
```

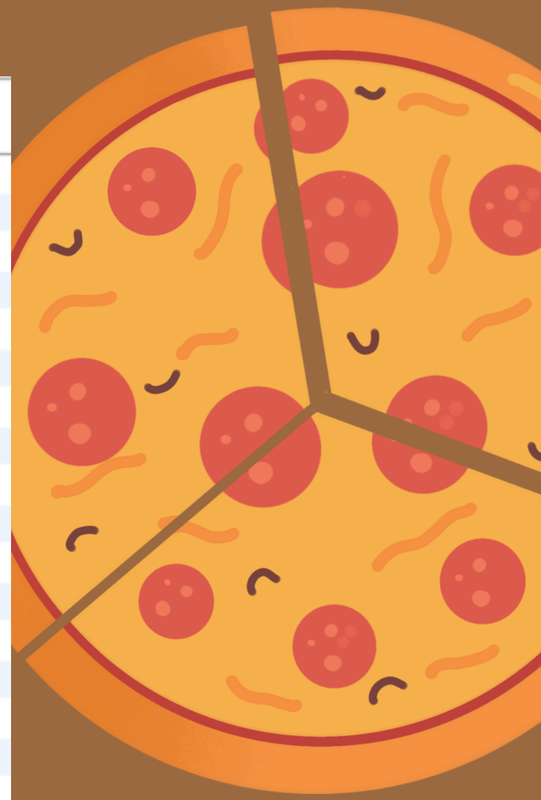


CATEGORY	REVENUE
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

13. THE CUMULATIVE REVENUE GENERATED OVER TIME.

```sql
SELECT DATE ,
SUM(REVENUE) OVER (ORDER BY DATE ) AS 'cumulative revenue'
 FROM
    (SELECT ORDERS.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.DATE ) AS SALES;
```

DATE	cumulative revenue
2015-07-24	471534.5000000003
2015-07-25	473771.7500000003
2015-07-26	475643.3000000003
2015-07-27	477815.3000000003
2015-07-28	479909.85000000027
2015-07-29	481833.10000000027
2015-07-30	484182.2500000003
2015-07-31	486277.6500000003
2015-08-01	488718.2000000003
2015-08-02	490628.3500000003
2015-08-03	492610.6000000003
2015-08-04	494700.75000000035
2015-08-05	496795.6000000003
2015-08-06	498894.8500000003
2015-08-07	501521.25000000035
2015-08-08	504237.6500000004
2015-08-09	506240.3000000004
2015-08-10	508379.7500000004
2015-08-11	510669.7500000004
2015-08-12	513035.5000000004
2015-08-13	515109.65000000043
2015-08-14	518126.2500000004
2015-08-15	520378.6000000004
2015-08-16	522517.9000000004
2015-08-17	525143.9000000004

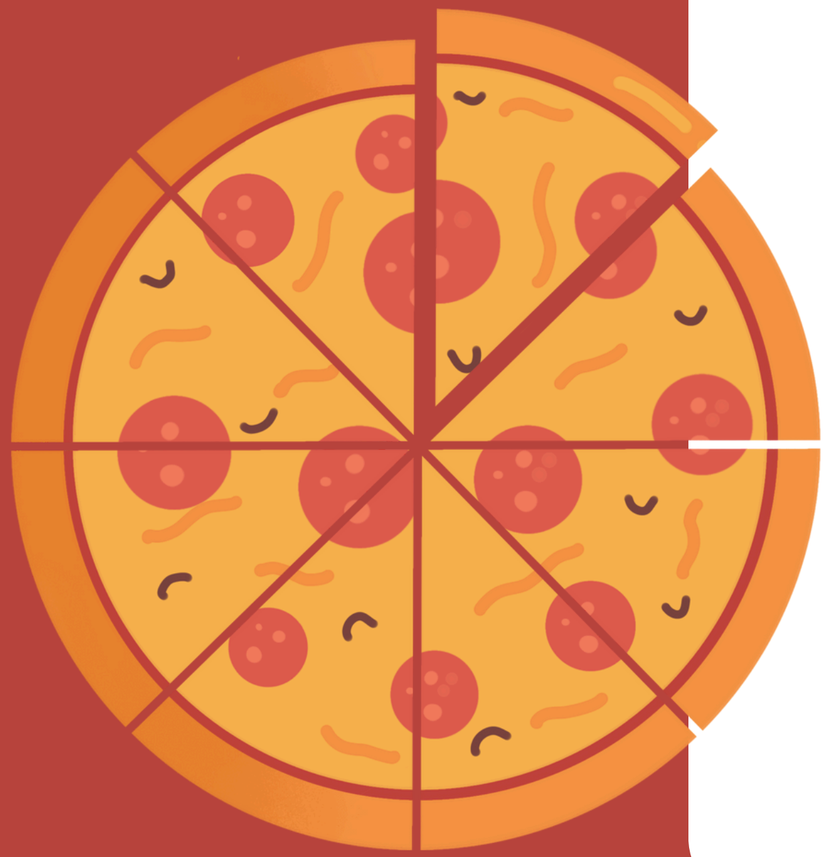


14. TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE FOR EACH PIZZA CATEGORY.

```sql
SELECT NAME,REVENUE
FROM
(WITH MY_CTE AS (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 ORDER_DETAILS.PIZZA_ID = PIZZAS.PIZZA_ID

GROUP BY PIZZA_TYPES.CATEGORY, PIZZA_TYPES.NAME)

SELECT CATEGORY , NAME, REVENUE,
RANK() OVER(PARTITION BY CATEGORY
            ORDER BY REVENUE DESC) AS RANKS
            FROM MY_CTE) AS SUB_Q
WHERE RANKS<= 3;
```

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
The Pepperoni Pizza	30161.75
The Spicy Italian Pizza	34831.25

KEY FINDINGS



1. the Classic category contributes 26.9% of total revenue, suggesting it as most stable market segment."
2. Total Orders: 48,620
3. Total Revenue: $817,860.05
4. Most Ordered Size: Large (L) – 18,526 orders
5. Highest Priced Pizza: the Greek Pizza – $35.95
6. Avg Pizzas Sold Per Day: 138
7. Peak ordering time observed around:
   12–1 PM (Lunch)
   6–8 PM (Dinner)

BUSINESS IMPACTS

- Helps identify best-selling items for promotions & combos
- Supports inventory planning using category-wise demand
- Improves staff scheduling during peak order hours
- Helps optimize menu strategy by focusing on high-revenue pizzas

CONCLUSION

"this project analyzed pizza sales data using MySQL to uncover sales trends and customer demand patterns , using SQL (Joins, Aggregations, Window Functions) to deliver actionable business insights."

THANK YOU