### PIZZA\_SALES SQL PROJECT

---- QUERIES & RESULTS

Q.1 Retrieve the total number of orders placed.

SELECT COUNT(ORDER\_ID) AS TOTAL\_ORDERS FROM ORDERS;



# Q.2 Calculate the total revenue generated from pizza sales.

### QUERY

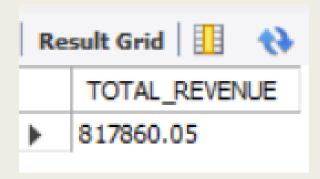
```
SELECT

ROUND(SUM(O.QUANTITY*P.PRICE),2) AS TOTAL_REVENUE

FROM ORDER_DETAILS O

JOIN PIZZAS P

ON O.PIZZA_ID=P.PIZZA_ID;
```



### Q.3 Identify the highest-priced pizza.

**QUERY** 

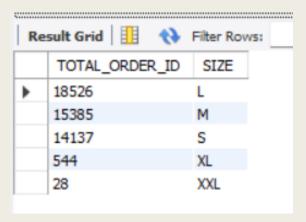
```
SELECT T.NAME , P.PRICE
FROM PIZZA_TYPES T
JOIN PIZZAS P
ON T.PIZZA_TYPE_ID = P.PIZZA_TYPE_ID
ORDER BY P.PRICE DESC LIMIT 1;
```



## Q.4 Identify the most common pizza size ordered.

#### **QUERY**

```
SELECT COUNT(0.ORDER_DETAILS_ID) AS TOTAL_ORDER_ID ,P.SIZE
FROM ORDER_DETAILS 0
JOIN PIZZAS P
ON 0.PIZZA_ID = P.PIZZA_ID GROUP BY SIZE ORDER BY TOTAL_ORDER_ID DESC;
```



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

**QUERY** 

```
FROM ORDER_DETAILS O

JOIN PIZZAS P

JOIN PIZZA_TYPES T

ON O.PIZZA_ID = P.PIZZA_ID AND P.PIZZA_TYPE_ID = T.PIZZA_TYPE_ID

GROUP BY NAME ORDER BY TOTAL_QTY DESC LIMIT 5;
```



## Q.6 Join the necessary tables to find the total quantity of each pizza category ordered.

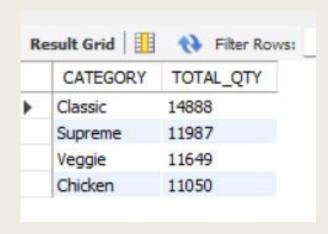
**QUERY** 

```
SELECT T. CATEGORY, SUM(QUANTITY) AS TOTAL_QTY
FROM ORDER_DETAILS O

JOIN PIZZAS P

JOIN PIZZA_TYPES T
ON O.PIZZA_ID = P.PIZZA_ID AND P.PIZZA_TYPE_ID = T.PIZZA_TYPE_ID

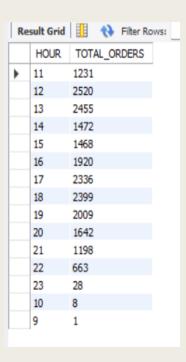
GROUP BY CATEGORY ORDER BY TOTAL_QTY DESC;
```



# Q.7 Determine the distribution of orders by hour of the day.

### **QUERY**

SELECT hour(ORDER\_TIME) AS HOUR, COUNT(ORDER\_ID) AS TOTAL\_ORDERS
FROM ORDERS GROUP BY hour(ORDER\_TIME);



Q.8 Join relevant tables to find the category-wise distribution of pizzas.

**QUERY** 

**RESULT** 

SELECT CATEGORY, COUNT(NAME) FROM PIZZA\_TYPES GROUP BY CATEGORY;

Result Grid						
	CATEGORY	COUNT(NAME)				
<b>-</b>	Chicken	6				
	Classic	8				
	Supreme	9				
	Veggie	9				

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

#### **QUERY**

```
SELECT ROUND(AVG(QUANTITY),0) as avg_pizzas_order_perday FROM

(SELECT O.ORDER_DATE , SUM(QUANTITY) AS QUANTITY

FROM ORDERS O

JOIN ORDER_DETAILS D

ON O.ORDER_ID = D.ORDER_ID GROUP BY O.ORDER_DATE) AS ORDER_QUANTITY;
```

# Q.10 Determine the top 3 most ordered pizza types based on revenue.

#### **QUERY**

```
SELECT T.NAME, SUM(QUANTITY*PRICE) AS REVENUE

FROM PIZZA_TYPES T

JOIN PIZZAS P

JOIN ORDER_DETAILS O

ON T.PIZZA_TYPE_ID = P.PIZZA_TYPE_ID AND P.PIZZA_ID = O.PIZZA_ID

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



## Q.11 Calculate the percentage contribution of each pizza type to total revenue.

#### **QUERY**

```
SELECT T.CATEGORY ,ROUND(SUM(O.QUANTITY*P.PRICE) /
(SELECT

ROUND(SUM(O.QUANTITY*P.PRICE),2) AS TOTAL_REVENUE

FROM ORDER_DETAILS O

JOIN PIZZAS P

ON O.PIZZA_ID=P.PIZZA_ID)*100,2) AS REVENUE

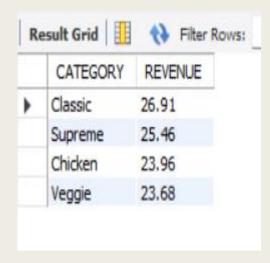
FROM PIZZA_TYPES T

JOIN PIZZAS P

JOIN ORDER_DETAILS O

ON T.PIZZA_TYPE_ID = P.PIZZA_TYPE_ID AND P.PIZZA_ID = O.PIZZA_ID

GROUP BY CATEGORY ORDER BY REVENUE DESC;
```



## Q.12 Analyze the cumulative revenue generated over time.

#### **QUERY**

```
SELECT ORDER_DATE,

SUM(REVENUE) OVER(ORDER BY ORDER_DATE) AS CUM_REVENUE

FROM

(SELECT O. ORDER_DATE , ROUND(SUM(QUANTITY*PRICE),2) AS REVENUE

FROM ORDERS O

JOIN ORDER_DETAILS D

JOIN PIZZAS P

ON O.ORDER_ID = D.ORDER_ID AND D.PIZZA_ID = P.PIZZA_ID

GROUP BY ORDER_DATE) AS SALES;
```

Result Grid					
	ORDER_DATE	CUM_REVENUE			
•	2015-01-01 00:00:00	2713.85			
	2015-01-02 00:00:00	5445.75			
	2015-01-03 00:00:00	8108.15			
	2015-01-04 00:00:00	9863.6			
	2015-01-05 00:00:00	11929.55			
	2015-01-06 00:00:00	14358.5			
	2015-01-07 00:00:00	16560.7			
	2015-01-08 00:00:00	19399.05			
	2015-01-06 00:00:00 2015-01-07 00:00:00	14358.5 16560.7			

## Q.13 Determine the top 3 most ordered pizza types based on revenue for each pizza category.

#### **QUERY**

```
SELECT CATEGORY,NAME, REVENUE,

RANK() OVER(PARTITION BY CATEGORY ORDER BY REVENUE DESC) AS RN

FROM

(SELECT T.CATEGORY,T.NAME, SUM(QUANTITY*PRICE) AS REVENUE

FROM PIZZA_TYPES T

JOIN PIZZAS P

JOIN ORDER_DETAILS O

ON T.PIZZA_TYPE_ID = P.PIZZA_TYPE_ID AND P.PIZZA_ID = O.PIZZA_ID

GROUP BY T.CATEGORY,T.NAME) SALES;
```

	esult Grid	♦ Filter Rows:	Export:	Wrap Cell	
	CATEGORY	NAME	REVENUE	RN	
Þ	Chicken	The Thai Chicken Pizza	43434.25	1	
	Chicken	The Barbecue Chicken Pizza	42768	2	
	Chicken	The California Ch The Californ	ia Chicken Pizza	3	
	Chicken	The Southwest Chicken Pizza	34705.75	4	
	Chicken	The Chicken Alfredo Pizza	16900.25	5	
	Chicken	The Chicken Pesto Pizza	16701.75	6	
	Classic	The Classic Deluxe Pizza	38180.5	1	
	Classic	The Hawaiian Pizza	32273.25	2	
	Classic	The Pepperoni Pizza	30161.75	3	
	Classic	The Greek Pizza	28454.100000000013	4	
	Classic	The Italian Capocollo Pizza	25094	5	
	Classic	The Napolitana Pizza	24087	6	
	Classic	The Big Meat Pizza	22968	7	
	Classic	The Pepperoni, Mushroom,	18834.5	8	
	Supreme	The Spicy Italian Pizza	34831.25	1	
	Supreme	The Italian Supreme Pizza	33476.75	2	
	Supreme	The Sicilian Pizza	30940.5	3	
	Supreme	The Pepper Salami Pizza	25529	4	

### THANK YOU

-- YASWANTH.P