

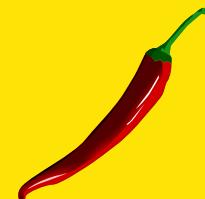
# SQL PROJECT ON PIZZA SALES

WATCH NOW



# **HELLO !**

**MY NAME IS SUSHMITA PAUL. IN THIS PROJECT I HAVE UTILIZED SQL QUERIES TO  
SOLVE QUESTIONS RELATED TO PIZZA SALES.**



**BASIC:**

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.

**INTERMEDIATE:**

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

DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

JOIN RELEVANT TABLES TO 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.

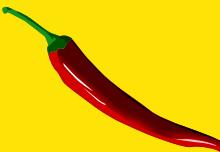
**ADVANCED:**

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.

# RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.



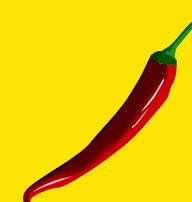
**SELECT**

COUNT(order\_id) AS total\_order

**FROM**

orders;

Result Grid	
	total_order
▶	21350



# CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

```
SELECT * FROM pizzas;  
SELECT * FROM order_details;  
  
SELECT  
    ROUND(SUM(od.quantity * p.price), 2) AS total_revenue  
FROM  
    order_details od  
    JOIN  
    pizzas p ON od.pizza_id = p.pizza_id;
```

Result Grid	
total_revenue	
817860.05	

# IDENTIFY THE HIGHEST-PRICED PIZZA.

```
SELECT * FROM pizzas;
SELECT * FROM pizza_types;

SELECT
    pt.name, p.price
FROM
    pizza_types pt
        JOIN
    pizzas p ON pt.pizza_type_id = p.pizza_type_id
ORDER BY p.price DESC
LIMIT 1;
```

Result Grid | Filter F

	name	price
▶	The Greek Pizza	35.95

# IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

```
SELECT * FROM order_details;  
SELECT * FROM pizzas;  
  
SELECT  
    p.size, COUNT(od.order_details_id) AS highest_size  
FROM  
    order_details od  
    JOIN  
    pizzas p ON od.pizza_id = p.pizza_id  
GROUP BY p.size  
ORDER BY p.size ASC;
```

	size	highest_size
▶	L	18526
	M	15385
	S	14137
	XL	544
	XXL	28

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

```
SELECT * FROM order_details;
SELECT * FROM pizza_types;
SELECT * FROM pizzas;

SELECT
    pt.name, SUM(od.quantity) AS total_quantity
FROM
    pizza_types pt
        JOIN
    pizzas p ON pt.pizza_type_id = p.pizza_type_id
        JOIN
    order_details od ON od.pizza_id = p.pizza_id
GROUP BY pt.name
ORDER BY total_quantity DESC
LIMIT 5;
```

Result Grid | Filter Rows:

	name	total_quantity
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

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



```
SELECT
    pt.category, SUM(od.quantity) AS quantity
FROM
    pizza_types pt
        JOIN
    pizzas p ON pt.pizza_type_id = p.pizza_type_id
        JOIN
    order_details od ON od.pizza_id = p.pizza_id
GROUP BY pt.category
ORDER BY quantity DESC;
```



	category	quantity
1	Classic	14888
2	Supreme	11987
3	Veggie	11649
4	Chicken	11050

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

```
SELECT  
    HOUR(order_time) AS Hour, COUNT(order_id) AS Orders  
FROM  
    orders  
GROUP BY HOUR(order_time);
```

	Hour	Orders
▶	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



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

```
SELECT  
    category, COUNT(category) AS category  
FROM  
    pizza_types  
GROUP BY category;
```

	category	category
>	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9

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

```
SELECT  
    ROUND(AVG(quantity), 0) AS avg_pizza_ordered_per_day  
FROM  
    (SELECT  
        o.order_date, SUM(od.quantity) AS quantity  
    FROM  
        orders o  
    JOIN order_details od ON o.order_id = od.order_id  
    GROUP BY o.order_date) AS order_quantity;
```

Result Grid | Filter Rows

	avg_pizza_ordered_per_day
▶	138

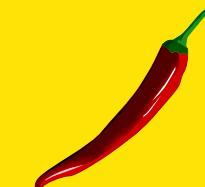
# DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE.

```
SELECT
    pt.name, SUM(od.quantity * p.price) AS revenue
FROM
    order_details od
        JOIN
    pizzas p ON od.pizza_id = p.pizza_id
        JOIN
    pizza_types pt ON pt.pizza_type_id = p.pizza_type_id
GROUP BY pt.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

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

```
SELECT
    pt.category,
    ROUND(SUM(ot.quantity * p.price) / (SELECT
        SUM(ot.quantity * p.price) AS total_sale
        FROM
            order_details ot
            JOIN
                pizzas p ON ot.pizza_id = p.pizza_id) * 100,
    2) AS percentage_revenue
FROM
    pizza_types pt
    JOIN
        pizzas p ON pt.pizza_type_id = p.pizza_type_id
    JOIN
        order_details ot ON ot.pizza_id = p.pizza_id
GROUP BY pt.category;
```



category	percentage_revenue
Classic	26.91
Veggie	23.68
Supreme	25.46
Chicken	23.96

# ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

```
SELECT order_date, SUM(revenue) OVER(ORDER BY order_date) AS cum_revenue
FROM
(SELECT o.order_date, SUM(od.quantity*p.price) AS revenue
FROM order_details od JOIN pizzas p ON od.pizza_id = p.pizza_id
JOIN orders o ON o.order_id = od.order_id GROUP BY o.order_date) AS sales;
```

# 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 pt.category, pt.name, SUM(od.quantity * p.price) AS revenue
FROM pizza_types pt JOIN pizzas p ON pt.pizza_type_id = p.pizza_type_id
JOIN order_details od ON od.pizza_id = p.pizza_id GROUP BY
pt.category, pt.name) AS a) AS b
WHERE rn <=3;
```

Result Grid | Filter Rows: Export:

	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	32265.70000000065
	Veggie	The Mexicana Pizza	26780.75
	Veggie	The Five Cheese Pizza	26066.5

