

PIZZA SALES PORTFOLIO SQL PROJECT





HELLO !

I have utilized SQL queries to solve the questions that were related to pizza sales.

QUESTION 1: RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED

```
SELECT COUNT(order_id) as total_orders FROM orders
```



Result Grid	
	total_orders
▶	21350



QUESTION 2: CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALE

```
SELECT round(sum(order_details.quantity * pizzas.price),2) as Total_Revenue  
FROM order_details  
inner join pizzas on order_details.pizza_id = pizzas.pizza_id
```



Result Grid	
	Total_Revenue
▶	817860.05



QUESTION 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
```

Result Grid | Filter Row

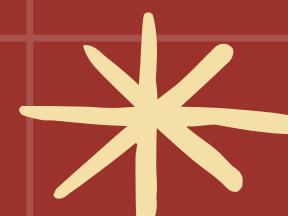
	name	price
▶	The Greek Pizza	35.95



QUESTION 4: IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

```
SELECT  
    pizzas.size, COUNT(order_details.order_details_id)  
FROM  
    order_details  
    JOIN  
    pizzas ON pizzas.pizza_id = order_details.pizza_id  
GROUP BY pizzas.size  
order by count(order_details.order_details_id) desc limit 1
```

	size	COUNT(order_details.order_details_id)
▶	L	18526



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

SELECT

 pizza_types.name, SUM(order_details.quantity)

FROM

 pizza_types

 JOIN

 pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id

 JOIN

 order_details ON pizzas.pizza_id = order_details.pizza_id

GROUP BY pizza_types.name

ORDER BY SUM(order_details.quantity) DESC

LIMIT 5



Result Grid | Filter Rows: Export:

	name	sum(order_details.quantity)
▶	The Classic Deluxe Pizza	2453
	The Barbecue Chicken Pizza	2432
	The Hawaiian Pizza	2422
	The Pepperoni Pizza	2418
	The Thai Chicken Pizza	2371



QUESTION 6: JOIN THE NECESSARY TABLES TO FIND THE TOTAL QUANTITY OF EACH PIZZA ORDERED

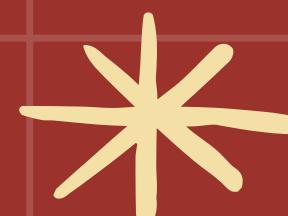
```
SELECT  
    pizza_types.name,  
    SUM(order_details.quantity) AS Total_quantity  
FROM  
    pizza_types  
        JOIN  
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id  
        JOIN  
    order_details ON order_details.pizza_id = pizzas.pizza_id  
GROUP BY pizza_types.name  
ORDER BY Total_quantity
```

	name	Total_quantity
▶	The Brie Carre Pizza	490
	The Mediterranean Pizza	934
	The Calabrese Pizza	937
	The Spinach Supreme Pizza	950
	The Soppressata Pizza	961
	The Spinach Pesto Pizza	970
	The Chicken Pesto Pizza	973
	The Italian Vegetables Pizza	981
	The Chicken Alfredo Pizza	987
	The Green Garden Pizza	997
	The Pepperoni, Mushroom,...	1359
	The Five Cheese Pizza	1409
	The Greek Pizza	1420
	The Italian Capocollo Pizza	1438
	The Pepper Salami Pizza	1446
	The Spinach and Feta Pizza	1446
	The Prosciutto and Arugula...	1457

QUESTION 7: DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY

```
SELECT  
    HOUR(time), COUNT(order_id)  
FROM  
    orders  
GROUP BY HOUR(time)  
ORDER BY COUNT(order_id)
```

	hour(time)	count(order_id)
▶	9	1
	10	8
	23	28
	22	663
	21	1198
	11	1231
	15	1468
	14	1472
	20	1642
	16	1920
	19	2009
	17	2336
	18	2399
	13	2455
	12	2520

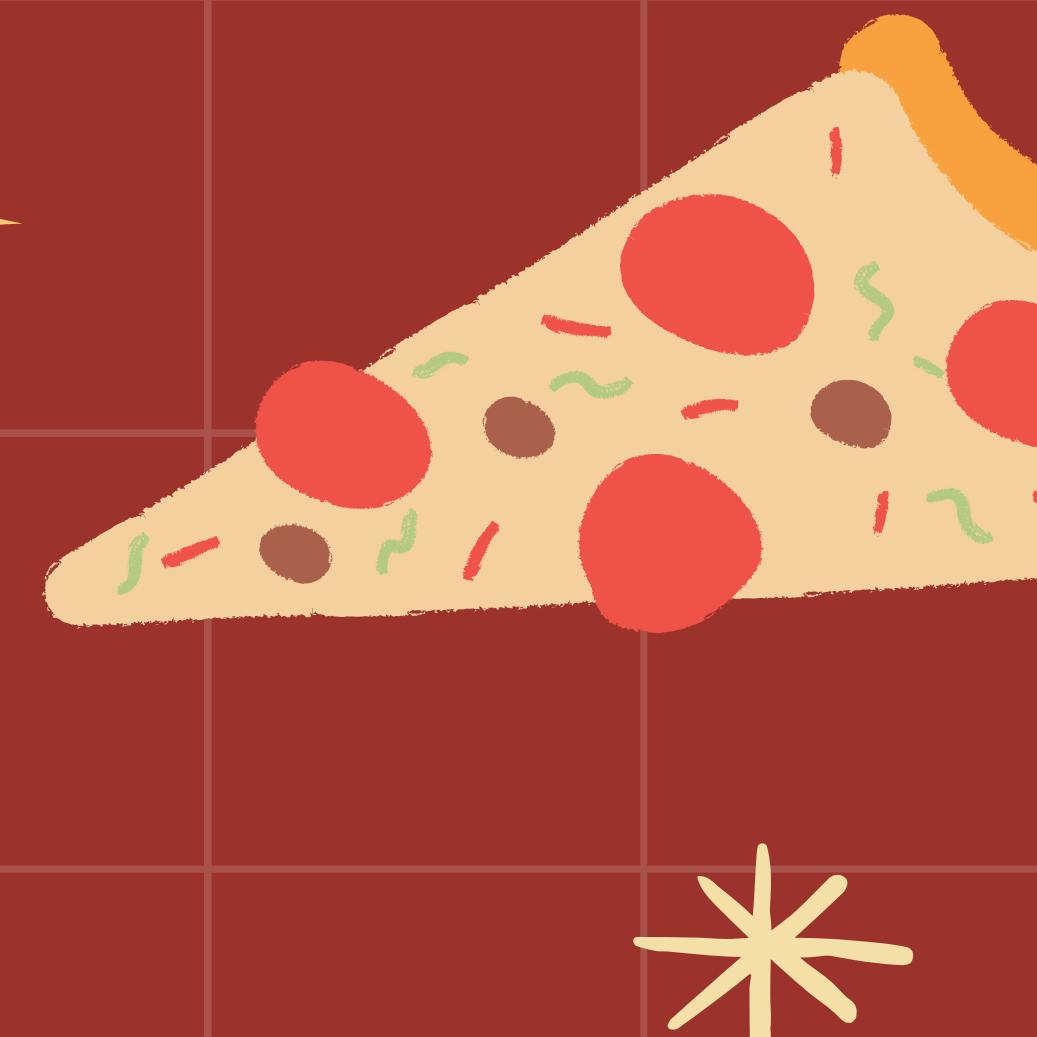


QUESTION 8: JOIN RELEVANT TABLES TO FIND THE CATEGORY-WISE DISTRIBUTION OF PIZZAS

```
SELECT  
    pizza_types.category, COUNT(pizza_types.name)  
FROM  
    pizza_types  
GROUP BY pizza_types.category
```

Result Grid | Filter Rows:

	category	count(pizza_types.name)
▶	Chicken	6
▶	Classic	8
▶	Supreme	9
▶	Veggie	9



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

SELECT

```
ROUND(AVG(quantity), 0) as avg_pizzas_perday
```

FROM

```
(SELECT
```

```
orders.date, SUM(order_details.quantity) AS quantity
```

FROM

```
orders
```

```
JOIN order_details ON order_details.order_id = orders.order_id
```

```
GROUP BY orders.date) AS order_quantity
```



	avg_pizzas_perday
▶	138



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

```
SELECT  
    pizza_types.name,  
    SUM(pizzas.price * order_details.quantity) 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.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

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

SELECT

```
    pizza_types.category,  
    ROUND(SUM(pizzas.price * order_details.quantity) / (SELECT  
        ROUND(SUM(order_details.quantity * pizzas.price),  
            2) AS total_sales  
    ) * 100,  
        2) AS Percentage_contribution  
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
```



Result Grid | Filter Rows:

	category	Percentage_contribution
▶	Classic	26.91
	Veggie	23.68
	Supreme	25.46
	Chicken	23.96



QUESTION 12: ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME



```
select date, sum(revenue) over(order by date) as cum_revenue from
(select orders.date, sum(order_details.quantity*pizzas.price) as revenue from orders
join order_details on
order_details.order_id=orders.order_id
join pizzas on pizzas.pizza_id=order_details.pizza_id
group by orders.date) as sales
```



	date	cum_revenue
▶	2015-01-01	2713.850000000004
	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	23990.35000000002
	2015-01-11	25862.65
	2015-01-12	27781.7
	2015-01-13	29831.30000000003
	2015-01-14	32358.70000000004
	2015-01-15	34343.50000000001
	2015-01-16	36937.65000000001
	2015-01-17	39001.75000000001

QUESTION 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, round(sum(pizzas.price*order_details.quantity),2) 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) as a) as b
where rn <=3
```



Result Grid | Filter Rows: E

	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	32265.7
	Veggie	The Mexicana Pizza	26780.75
	Veggie	The Five Cheese Pizza	26066.5

A festive illustration set against a red background with a white grid. In the center, the words "THANK YOU" are written in large, bold, white capital letters. To the left, a person with dark curly hair and a green sweater holds a slice of pizza. To the right, another person with glasses and a green sweater holds a small wrapped gift. Above them, a reindeer with a yellow and orange patterned collar and a bell hangs from its neck. The reindeer has large, expressive eyes and a small smile. The background is decorated with yellow stars and a large yellow starburst at the bottom.

THANK YOU