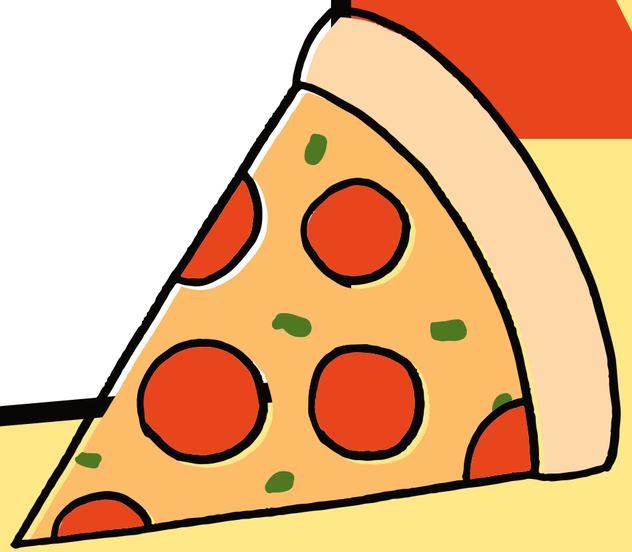
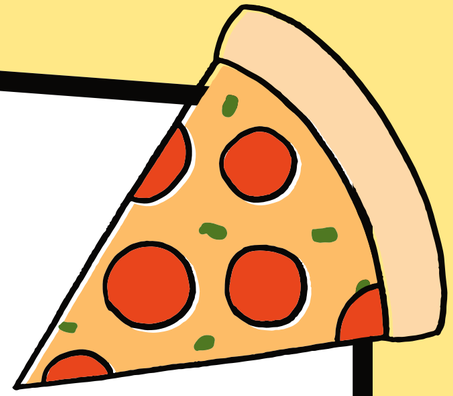
SQL Project on Pizza Sales





Hello!

Welcome to my SQL Project

My name is Sowmya Kodityala. I'm excited to share my SQL project analyzing pizza sales data.

In my SQL project analyzing pizza sales data, I delved into various aspects of the business, ranging from basic queries like retrieving the total number of orders and calculating total revenue to more advanced analyses such as determining the percentage contribution of each pizza type to total revenue and analyzing the cumulative revenue generated over time. Through this project, I gained hands-on experience in SQL querying, data analysis, and database management, which has equipped me with valuable skills for interpreting and deriving insights from complex datasets..

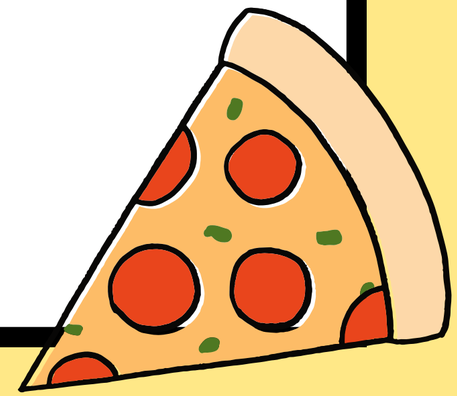
Question #1

Retrieve the total number of orders placed.

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

Result Grid

total_orders
21350



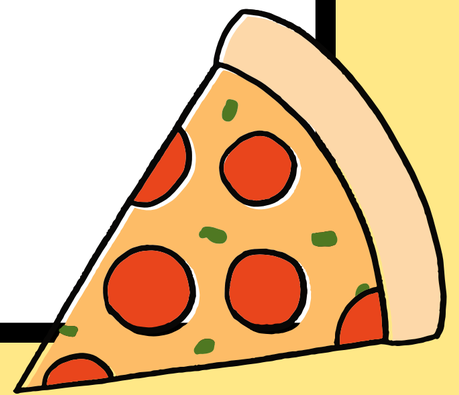
Question #2

Calculate the total revenue generated from pizza sales.

```
SELECT
    round(sum(order_details.quantity * pizzas.price),2) as Total_Revenue
FROM
    order_details
        JOIN
    pizzas ON order_details.pizza_id = pizzas.pizza_id;
```

Result Grid | Filter

	Total_Revenue
▶	817860.05



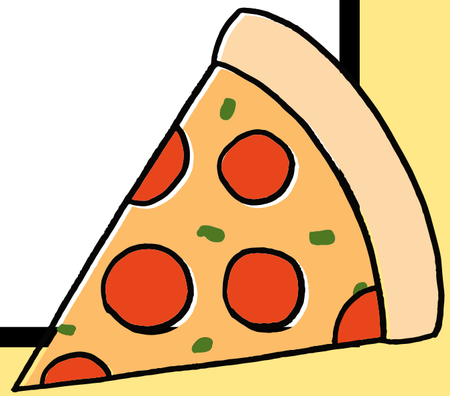
Question #3

Identify the highest-priced pizza.

```sql
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 Rows:

	name	price
▶	The Greek Pizza	35.95



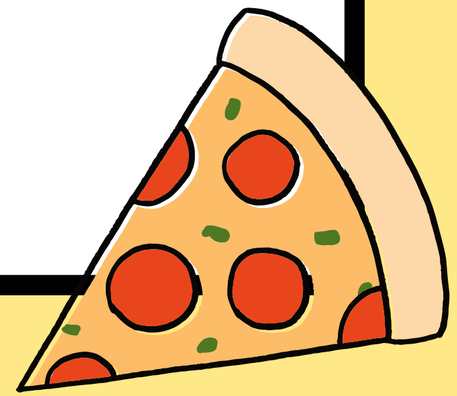
Question #4

Identify the most common pizza size ordered.

```
SELECT
    pizzas.size,
    COUNT(order_details.order_details_id) AS most_common_pizza
FROM
    pizzas
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizzas.size
ORDER BY most_common_pizza DESC;
```

Result Grid | Filter Rows:

size	most_common_pizza
L	18526
M	15385
S	14137
XL	544
XXL	28



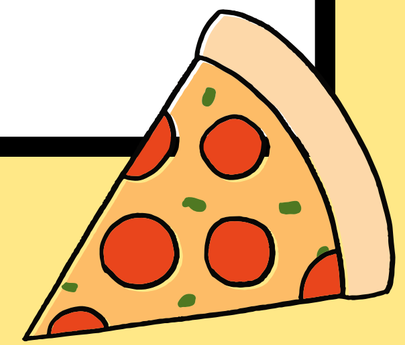
Question #5

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

```sql
SELECT
    pizza_types.name, SUM(order_details.quantity) as 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 quantity DESC
LIMIT 5;
```

Result Grid | Filter Rows:

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



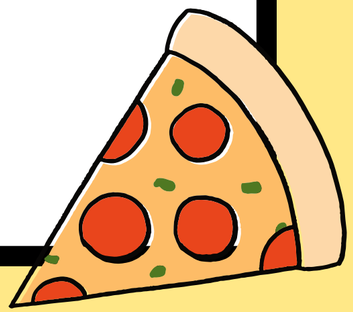
Question #6

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

```sql
SELECT
    pizza_types.category,
    SUM(order_details.quantity) AS 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.category
ORDER BY quantity DESC;
```

Result Grid | Filter

category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

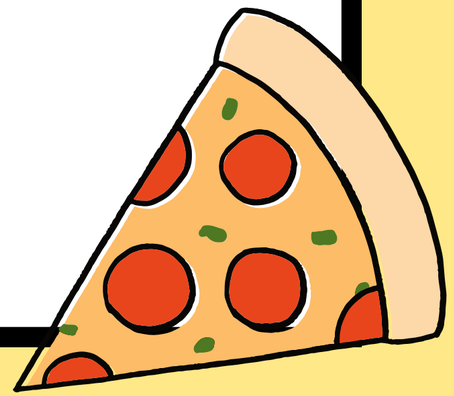


Question #7

Determine the distribution of orders by hour of the day.

```sql
SELECT
    HOUR(order_time), COUNT(order_id)
FROM
    orders
GROUP BY HOUR(order_time)
ORDER BY COUNT(order_id) DESC;
```

hour(order_time)	count(order_id)
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

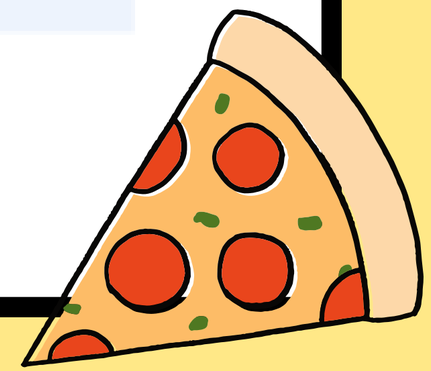


Question #8

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

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

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



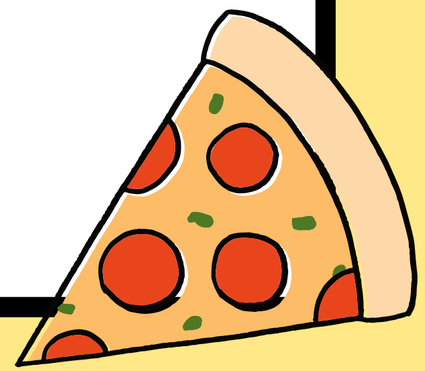
Question #9

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

```sql
SELECT
    round(AVG(quantity),0)
FROM
    (SELECT
        orders.order_date, SUM(order_details.quantity) AS quantity
    FROM
        orders
    JOIN order_details ON orders.order_id = order_details.order_id
    GROUP BY orders.order_date) as avg_quantity_per_day;
```

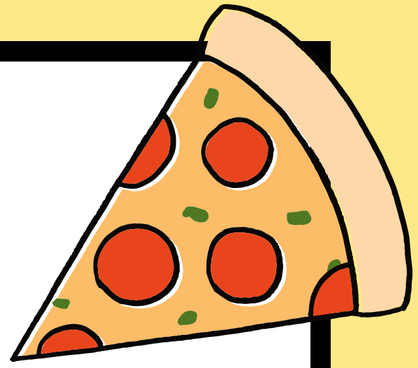
Result Grid | Filter Rows:

round(AVG(quantity),0)
138



Question #10

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

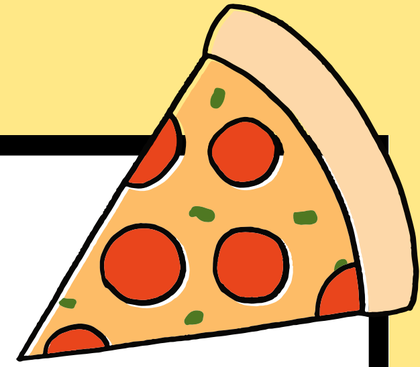


```sql
SELECT
        pizza_types.name,
        round(sum(order_details.quantity * pizzas.price),0) as most_ordered_pizza
    FROM
        pizza_types
    JOIN pizzas ON  pizzas.pizza_type_id = pizza_types.pizza_type_id
    JOIN order_details ON order_details.pizza_id = pizzas.pizza_id
    group by pizza_types.name
    order by most_ordered_pizza desc limit 3;
```

Result Grid | Filter Rows:

name	most_ordered_pizza
The Thai Chicken Pizza	43434
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41410

Question #11



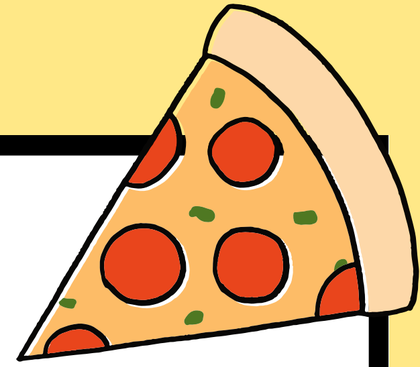
Calculate the percentage contribution of each pizza type to total revenue.

```
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 order_details.pizza_id = pizzas.pizza_id)) * 100,2) 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
order by revenue desc;
```

Result Grid | Filter Rc

category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

Question #12



Analyze the cumulative revenue generated over time.

```
select order_date , sum(revenue) over(order by order_date) as Cummulatuve_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 orders.order_id = order_details.order_id
group by orders.order_date) as sales;
```

order_date	Cummulatuve_revenue
2015-01-01	2713.8500000000004
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	23990.350000000002
2015-01-11	25862.65
2015-01-12	27781.7
2015-01-13	29831.300000000003

Question #13

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

```sql
select name, total_revenue, rn from
(select category, name, total_revenue, rank() over (partition by category order by total_revenue desc) as rn
from
(select pizza_types.category, pizza_types.name, sum(order_details.quantity * pizzas.price) as total_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 sales) as b
where rn <=3;
```

name	total_revenue	rn
The Thai Chicken Pizza	43434.25	1
The Barbecue Chicken Pizza	42768	2
The California Chicken Pizza	41409.5	3
The Classic Deluxe Pizza	38180.5	1
The Hawaiian Pizza	32273.25	2
The Pepperoni Pizza	30161.75	3
The Spicy Italian Pizza	34831.25	1
The Italian Supreme Pizza	33476.75	2
The Sicilian Pizza	30940.5	3
The Four Cheese Pizza	32265.70000000065	1
The Mexicana Pizza	26780.75	2
The Five Cheese Pizza	26066.5	3

Thank You!