SQL Project

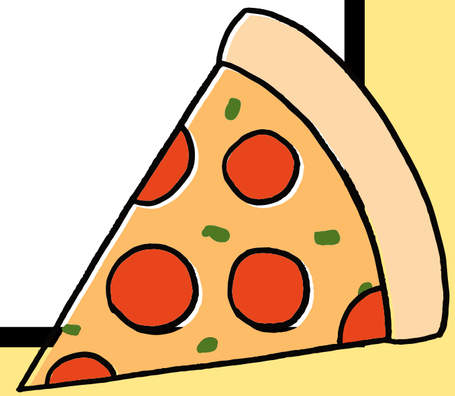
Pizza Fractions

This project aims to provide a comprehensive analysis of pizza sales data by utilizing SQL to extract meaningful insights. The analysis covers various aspects of the sales process, including the total number of orders placed and the total revenue generated. It identifies key metrics such as the highest-priced pizza and the most commonly ordered pizza size. Additionally, the project highlights the top 5 most popular pizza types based on order quantities.

Question #1

Retrieve the total number of orders placed

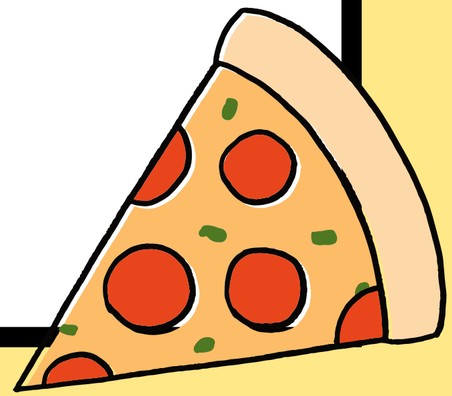
```sql
SELECT
    COUNT(order_id) AS total_orders
FROM
    orders;
```



Question #2

Calculate the total revenue generated from pizza sales.

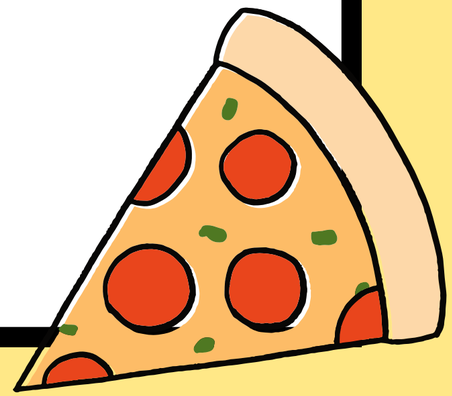
```sql
SELECT
    ROUND(SUM((od.quantity * p.price)), 2) AS total_revenue
FROM
    order_details AS od
        JOIN
    pizzas AS p ON od.pizza_id = p.pizza_id;
```



Question #3

Identify the highest-priced pizza.

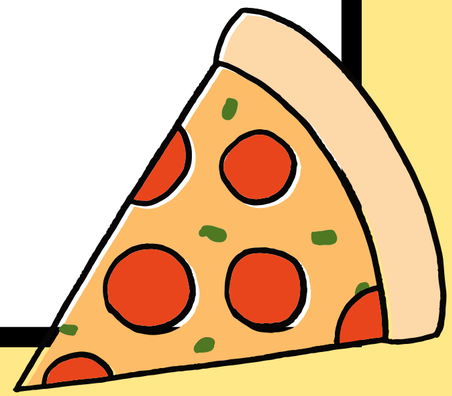
```
SELECT
    pt.name, p.price
FROM
    pizza_types AS pt
        JOIN
    pizzas AS p ON pt.pizza_type_id = p.pizza_type_id
ORDER BY p.price DESC
LIMIT 1;
```



Question #4

Identify the most common pizza size ordered.

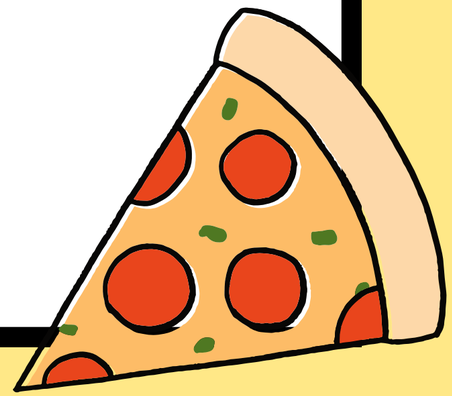
```
SELECT
   p.size, COUNT(od.order_detail_id) AS order_count
FROM
    order_details AS od
        JOIN
    pizzas AS p ON od.pizza_id = p.pizza_id
GROUP BY p.size ORDER BY order_count DESC LIMIT 1;
```



Question #5

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

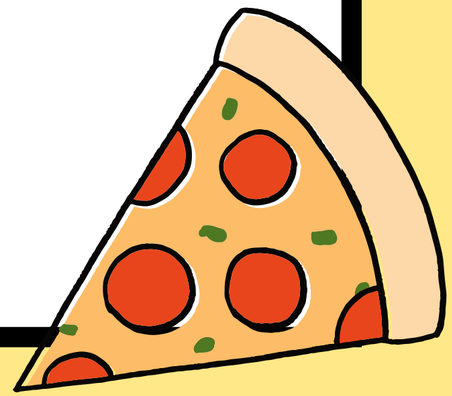
```sql
SELECT
    pt.name, SUM(od.quantity) AS total_order
FROM
    pizza_types AS pt
        JOIN
    pizzas AS p ON pt.pizza_type_id = p.pizza_type_id
        JOIN
    order_details AS od ON p.pizza_id = od.pizza_id
GROUP BY pt.name
ORDER BY total_order DESC
LIMIT 5;
```



Question #6

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

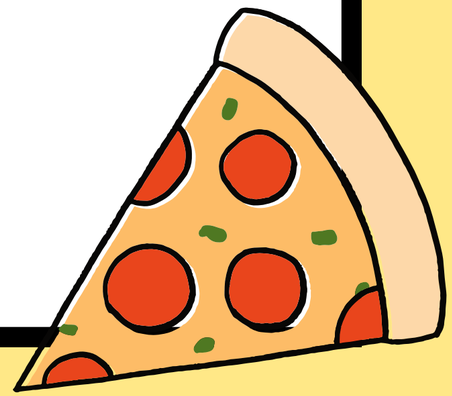
```sql
SELECT
    pt.category, SUM(od.quantity)
FROM
    pizza_types AS pt
        JOIN
    pizzas AS p ON p.pizza_type_id = pt.pizza_type_id
        JOIN
    order_details AS od ON p.pizza_id = od.pizza_id
GROUP BY pt.category;
```



Question #7

Determine the distribution of orders by hour of the day.

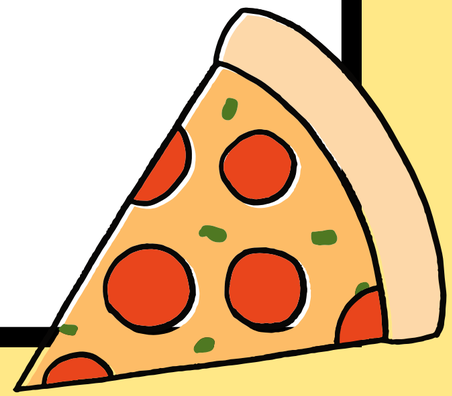
```sql
SELECT
    HOUR(order_time), COUNT(order_id)
FROM
    orders
GROUP BY HOUR(order_time);
```



Question #8

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

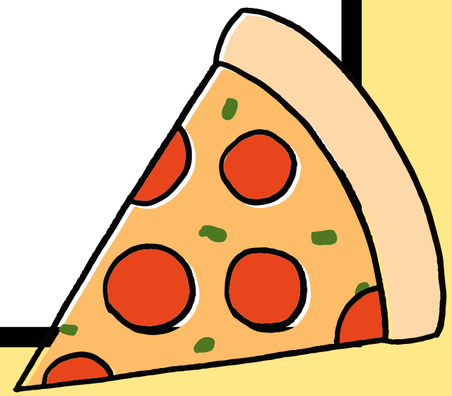
```sql
SELECT
    category, COUNT(name) AS pizza
FROM
    pizza_types
GROUP BY category;
```



Question #9

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

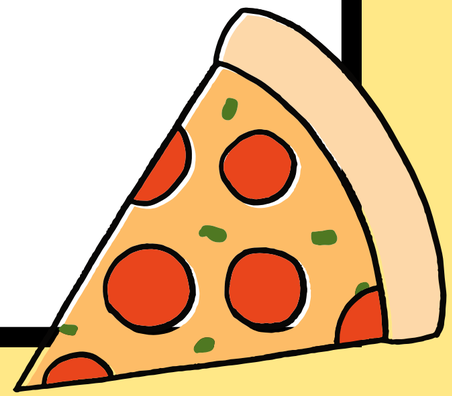
```sql
SELECT
    ROUND(AVG(QUANTITY), 0)
FROM
    (SELECT
        o.order_date, SUM(od.quantity) AS QUANTITY
    FROM
        orders AS o
    JOIN order_details AS od ON o.order_id = od.order_id
    GROUP BY o.order_date) AS order_quantity;
```



Question #10

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

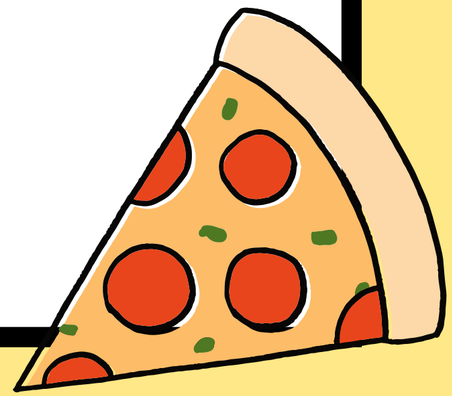
```sql
SELECT
    pt.name, ROUND(SUM(od.quantity * p.price), 2) AS revenue
FROM
    pizza_types AS pt
        JOIN
    pizzas AS p ON pt.pizza_type_id = p.pizza_type_id
        JOIN
    order_details AS od ON p.pizza_id = od.pizza_id
GROUP BY pt.name
ORDER BY revenue DESC
LIMIT 3;
```



Question #11

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

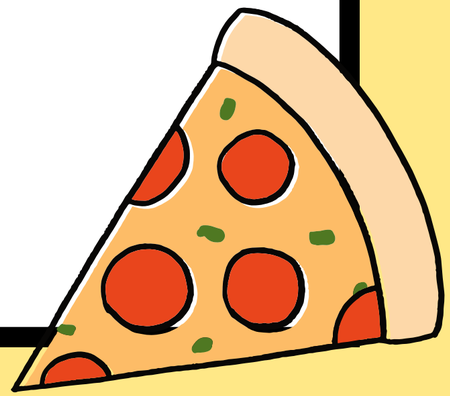
```sql
SELECT
    pt.category,ROUND((SUM(od.quantity * p.price)  / (SELECT
    SUM(od.quantity * p.price) AS total_sales
    FROM order_details AS od
    JOIN pizzas AS p
    ON p.pizza_id = od.pizza_id) )  * 100,0) AS revenue
FROM
    pizza_types AS pt
        JOIN
    pizzas AS p ON pt.pizza_type_id = p.pizza_type_id
        JOIN
    order_details AS od ON p.pizza_id = od.pizza_id
GROUP BY pt.category ORDER BY revenue DESC
;
```



Question #12

Analyze the cumulative revenue generated over time.

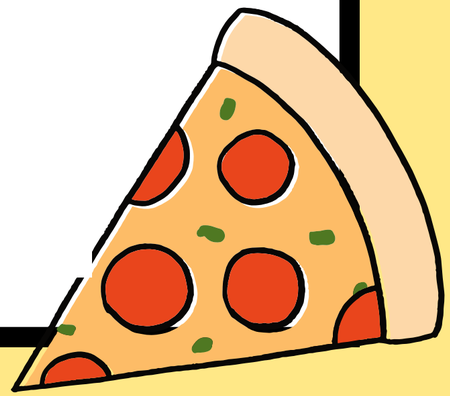
```sql
SELECT order_date , sum(revenue) OVER(order by order_date) AS cum_revenue
FROM (SELECT o.order_date, ROUND(SUM(od.quantity * p.price),0) AS revenue
FROM orders AS o
JOIN order_details AS od
ON o.order_id = od.order_id
JOIN pizzas AS p
ON od.pizza_id = p.pizza_id
GROUP BY o.order_date) AS sales ;
```



Question #13

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

```sql
SELECT name, revenue
FROM (
    SELECT category, name, revenue, RANK() OVER(PARTITION BY category ORDER BY revenue DESC) AS RN
    FROM (
        SELECT pt.category, pt.name, ROUND(SUM(od.quantity * p.price), 0) AS revenue
        FROM pizza_types AS pt
        JOIN pizzas AS p ON pt.pizza_type_id = p.pizza_type_id
        JOIN order_details AS od ON od.pizza_id = p.pizza_id
        GROUP BY pt.category,    pt.name
    ) AS sales
) AS b
WHERE RN <= 3;
```



Thank You