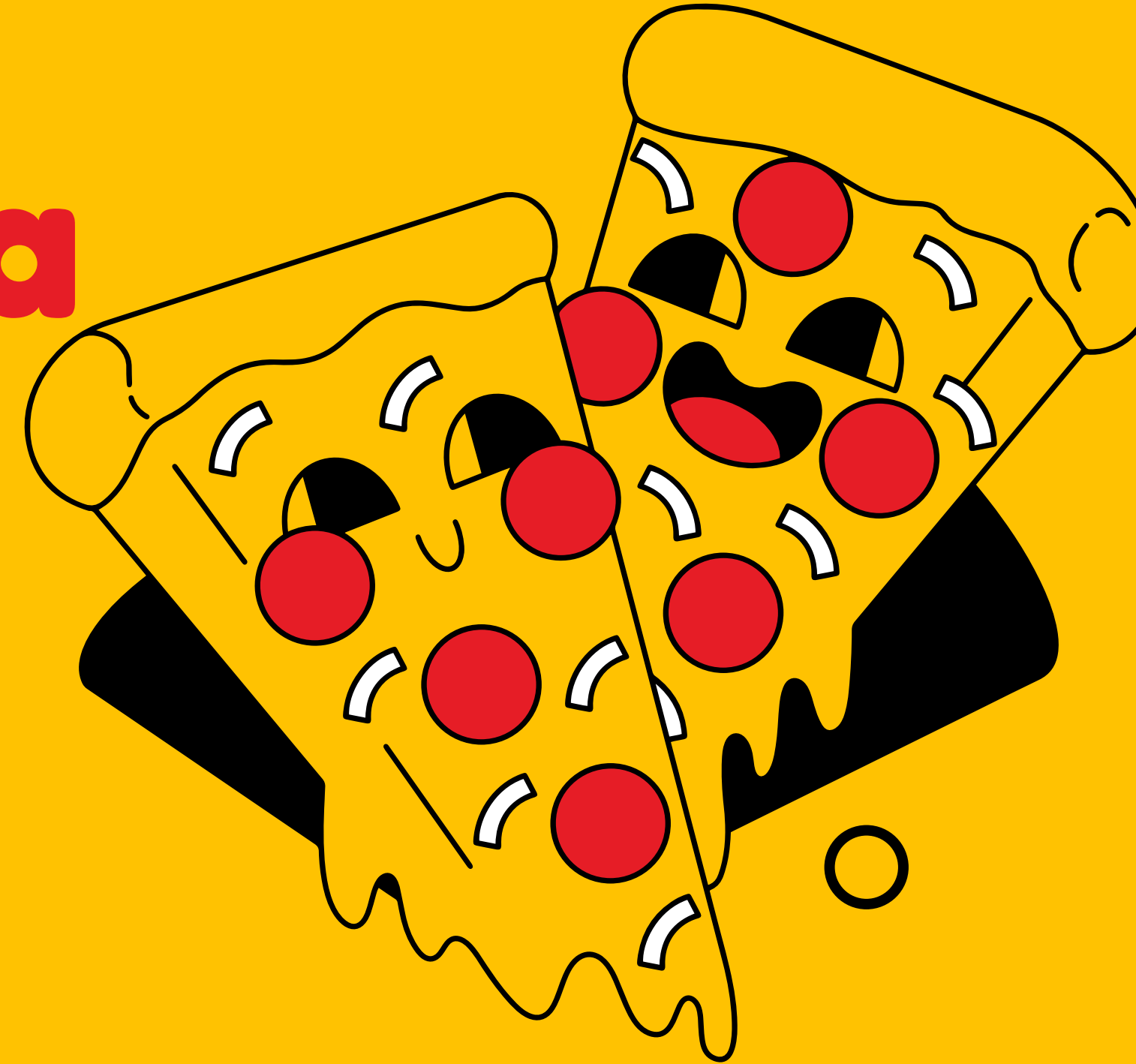


COMPREHENSIVE DATA ANALYSIS OF:

Dominos Pizza

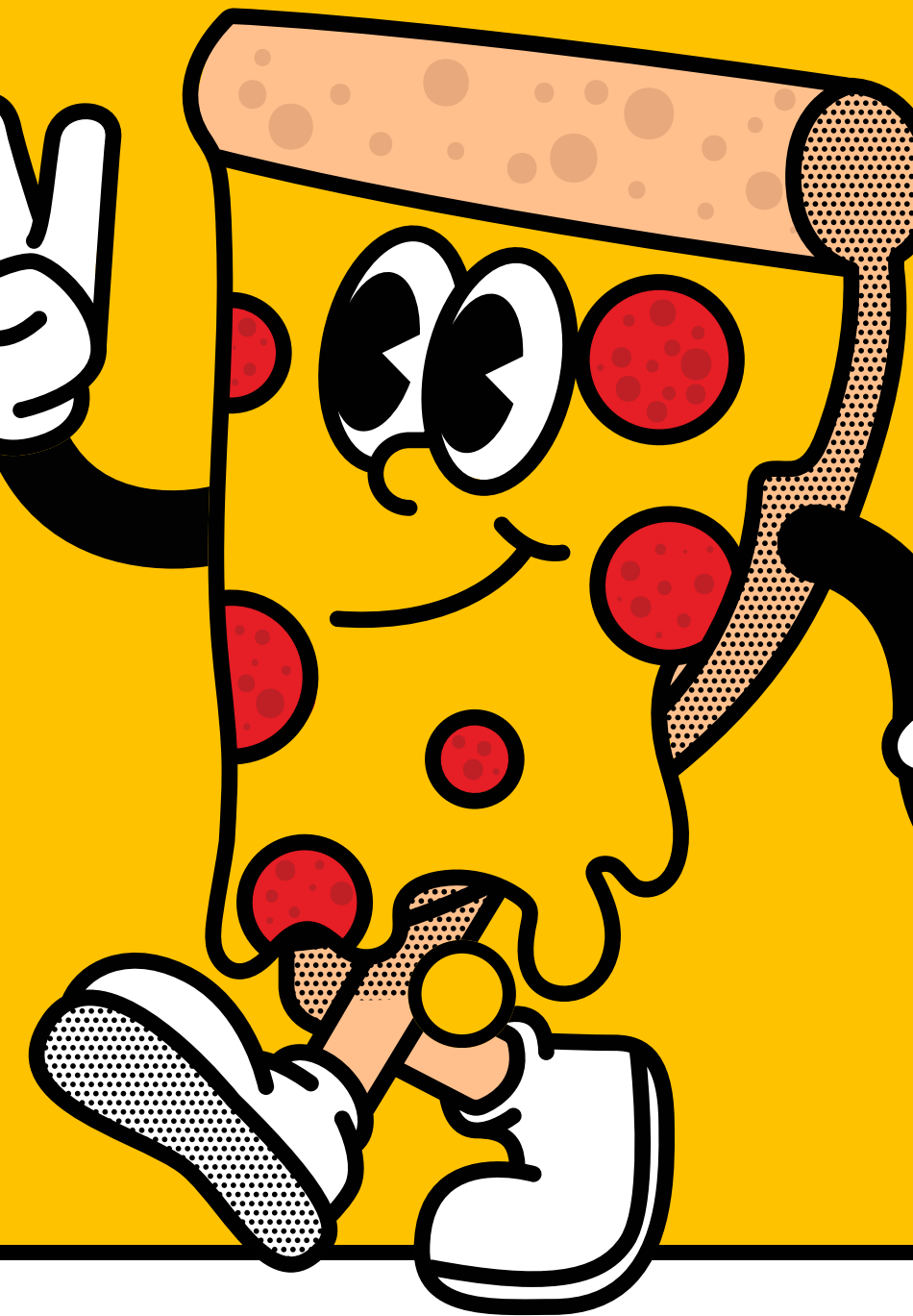
Using
MySQL



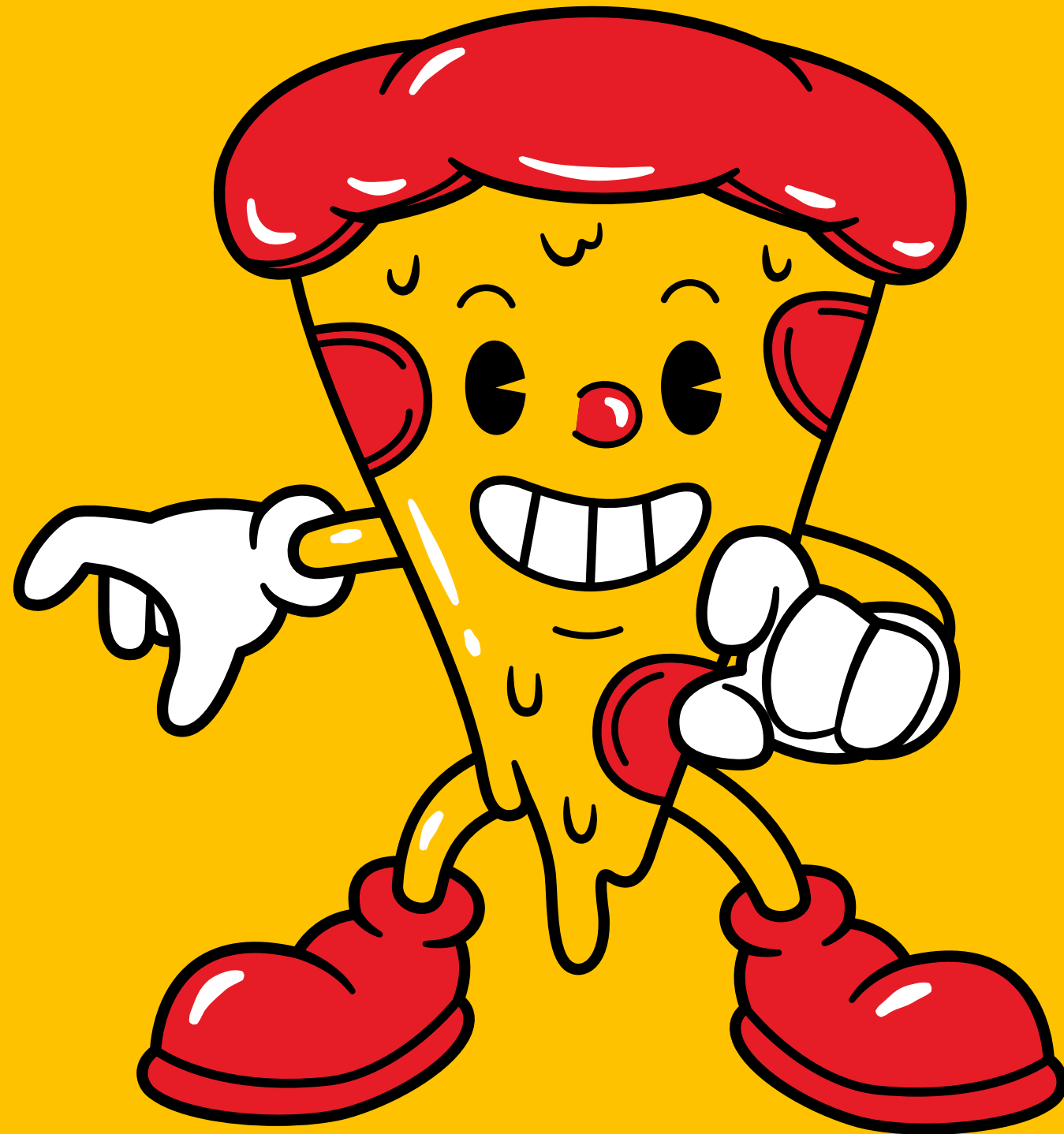
DOMINOS

PIZZA

SQL PROJECT

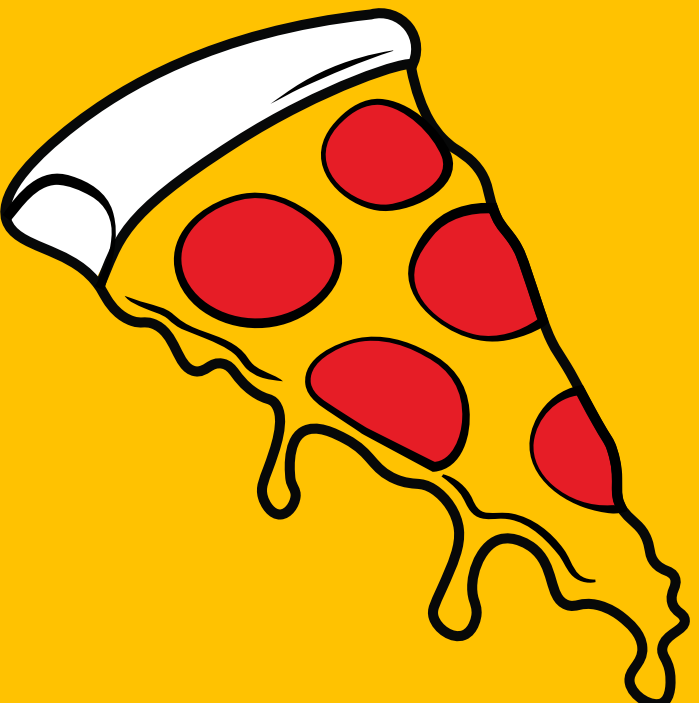
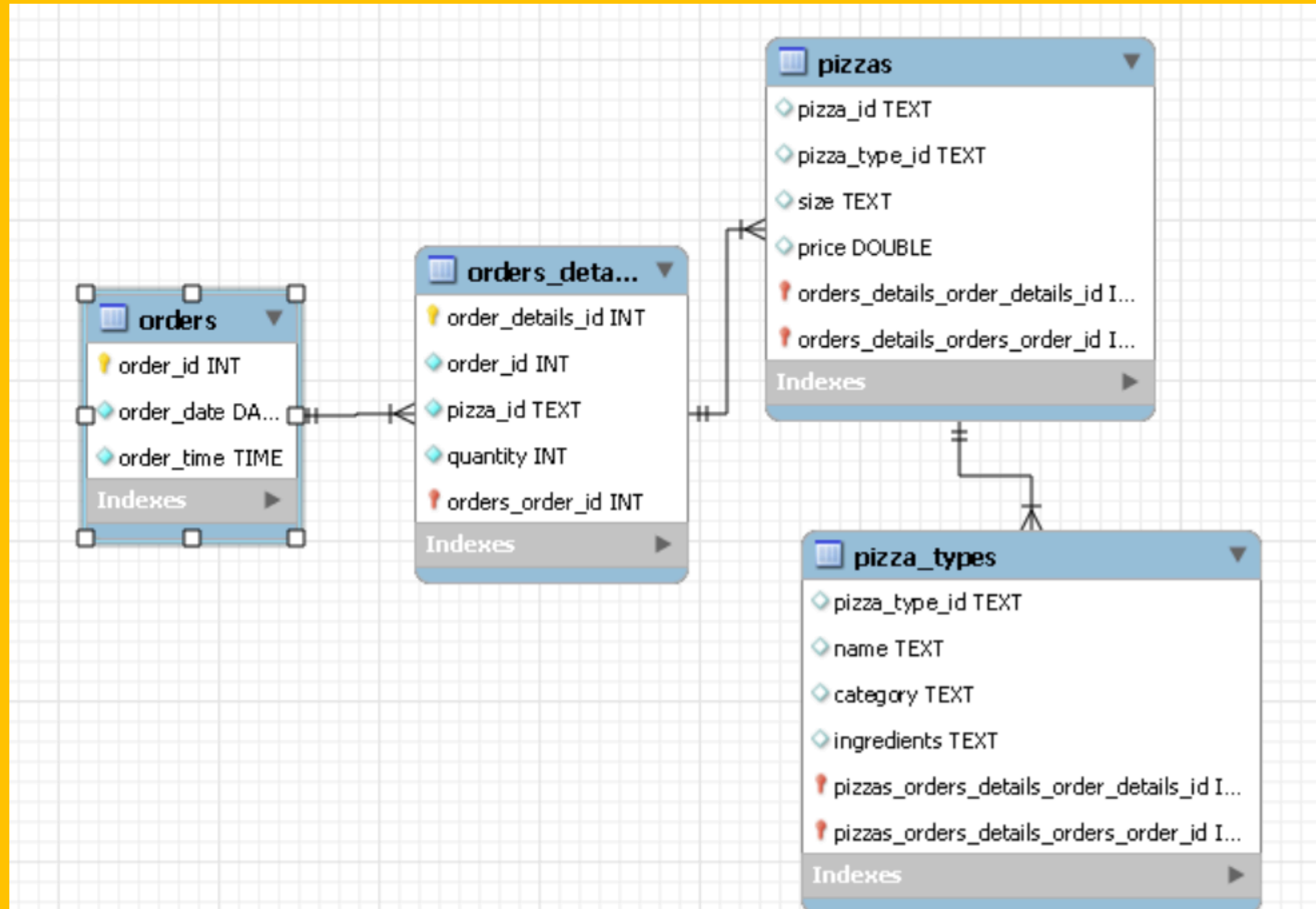


PROJECT INTRODUCTION



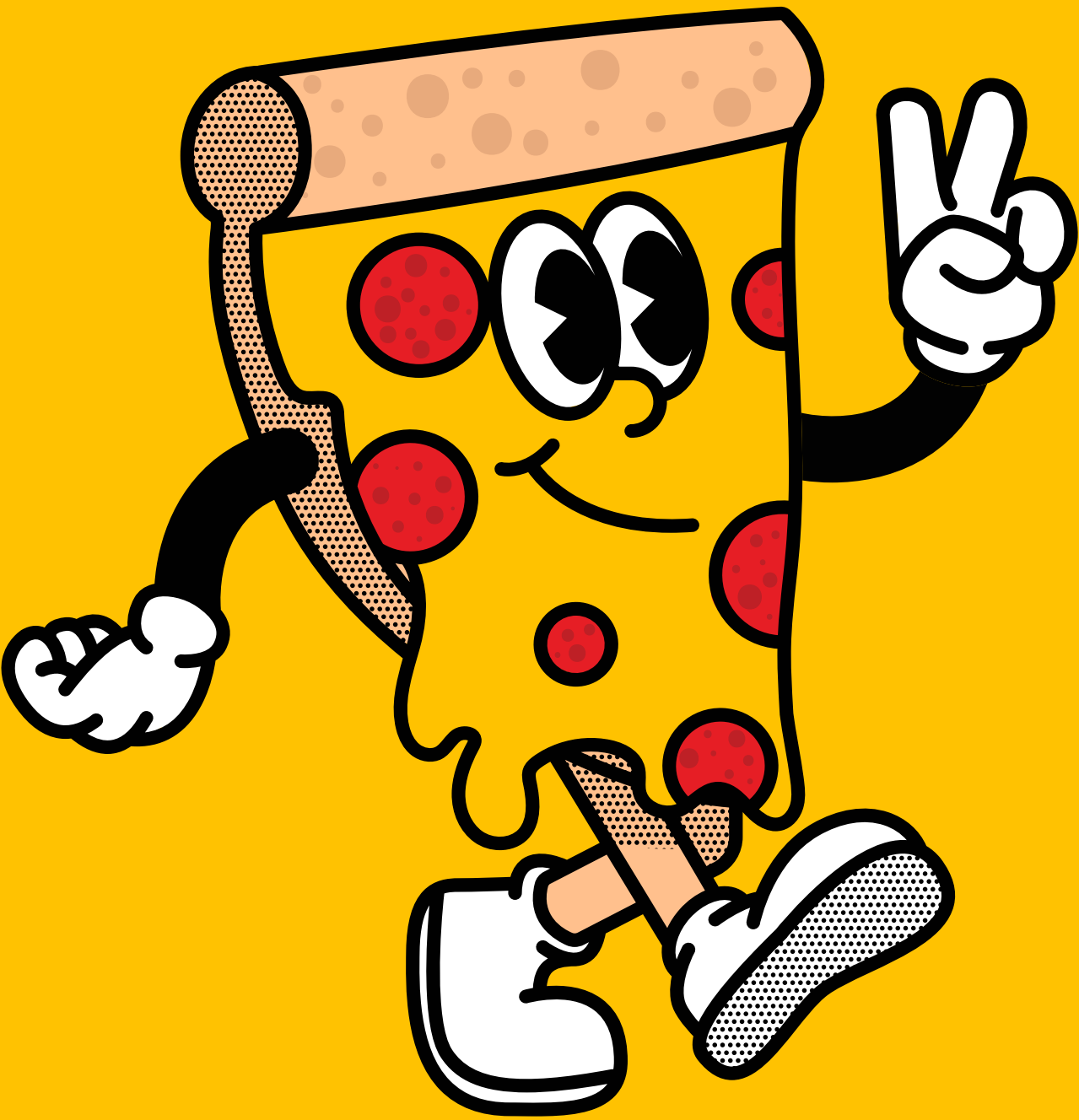
This project focuses on analyzing pizza sales data using SQL to derive valuable business insights. By querying and aggregating data from a structured database, the goal is to uncover key trends such as best-selling pizzas, revenue patterns, and customer preferences. The analysis helps identify performance metrics that can support data-driven decision-making for inventory management, marketing strategies, and operational improvements.

DATABASE SETUP [ER DIAGRAM]



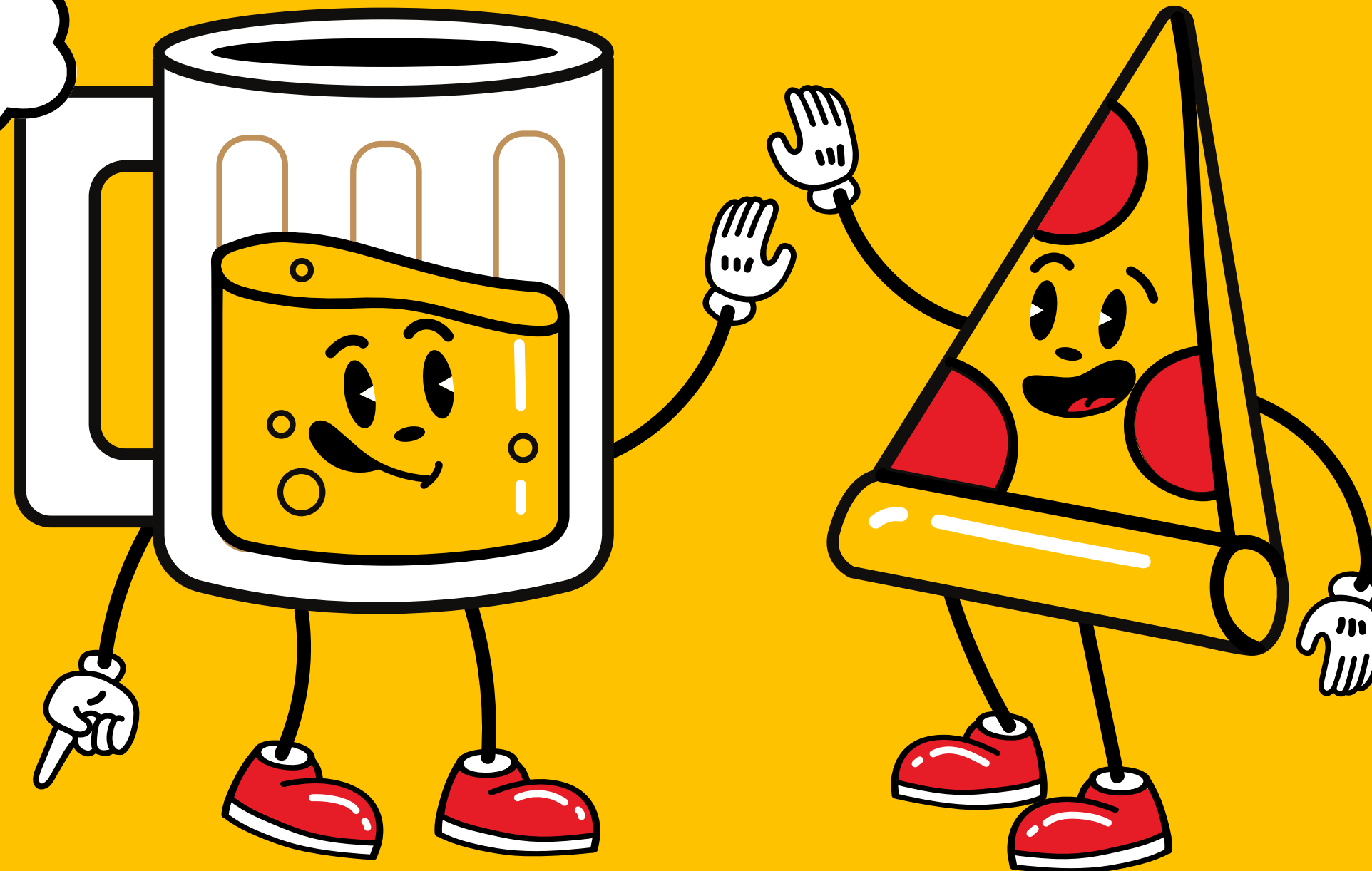
TOPICS COVERED IN THIS PROJECT:

Sheelu Singh



- Database Creation and Table Design
- Data Retrieval and Filtering (Basics SQL)
- Aggregations and Grouping
- Joining Tables
- Window Functions
- Date and Time Functions
- Conditional Logic
- Subqueries
- Ranking and Ordering
- Data Segmentation (CASE STATEMENT)

LET'S
START!



Business Problem 1

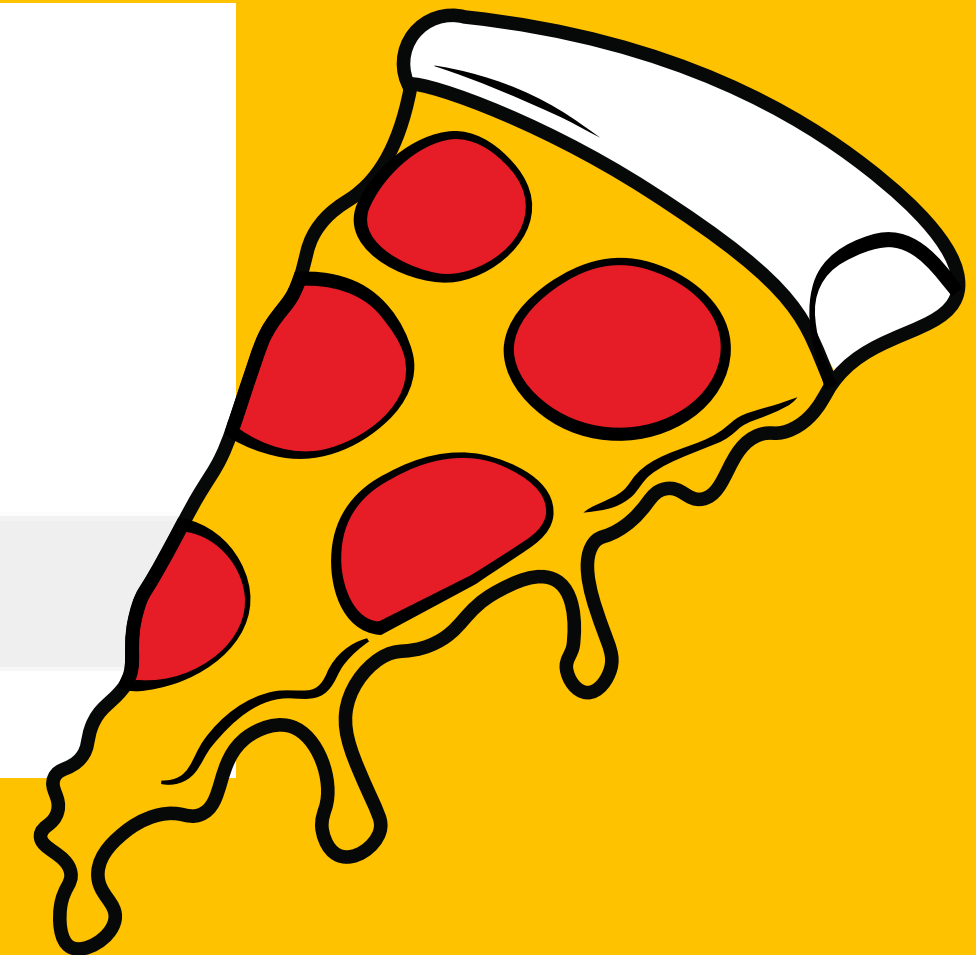
RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

```
SELECT
```

```
    COUNT(order_id) AS total_orders
```

```
FROM
```

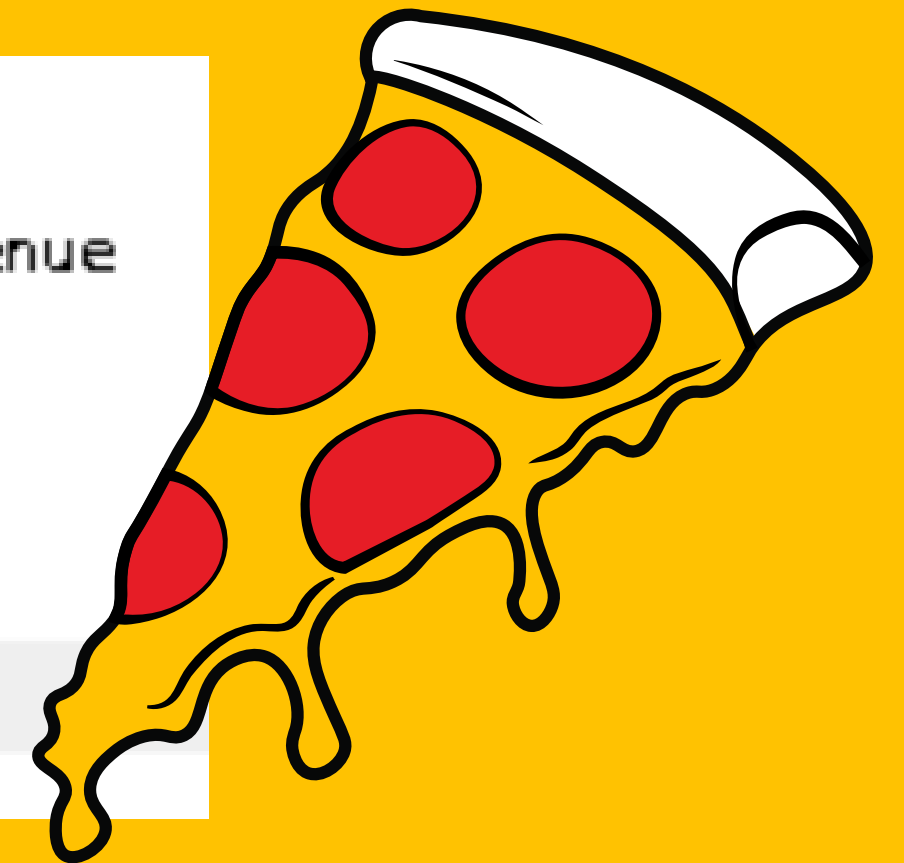
```
orders;
```



Business Problem 2

CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

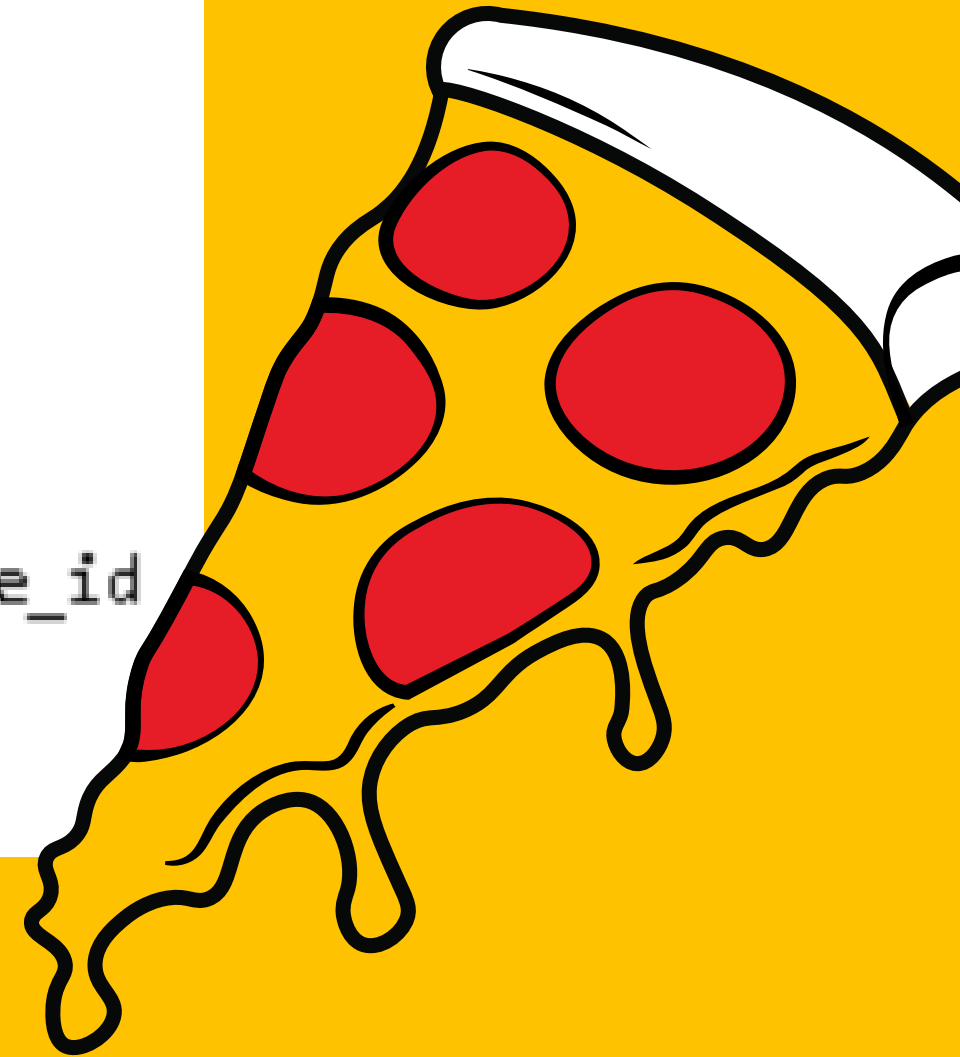
```
SELECT  
    ROUND(SUM(p.price * o.quantity), 2) AS total_revenue  
FROM  
    orders_details AS o  
    JOIN  
    pizzas AS p ON o.pizza_id = p.pizza_id;
```



Business Problem 3

IDENTIFY THE HIGHEST-PRICED PIZZA.

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



Business Problem 4

CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

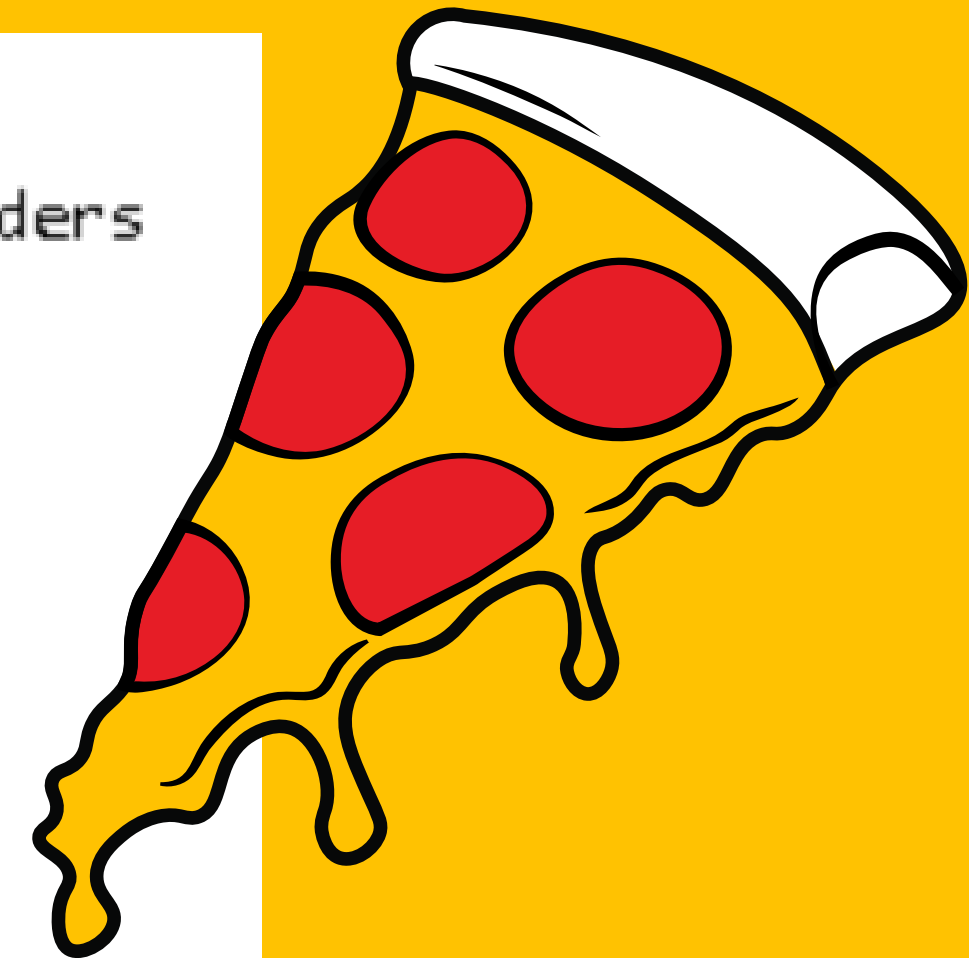
```
SELECT  
    ROUND(SUM(p.price * o.quantity), 2) AS total_revenue  
FROM  
    orders_details AS o  
    JOIN  
    pizzas AS p ON o.pizza_id = p.pizza_id;
```



Business Problem 5

IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED

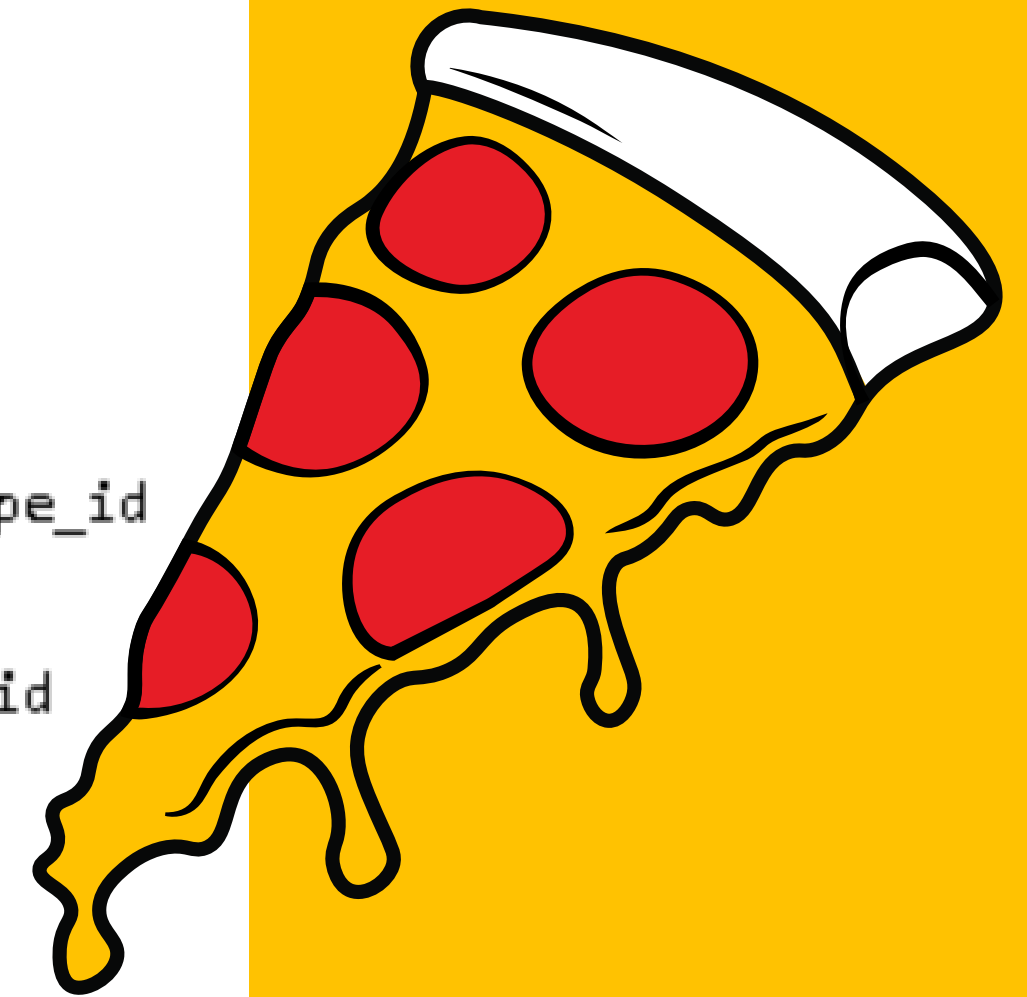
```
SELECT
    p.size, COUNT(o.order_details_id) AS total_orders
FROM
    orders_details AS o
    JOIN
    pizzas AS p ON o.pizza_id = p.pizza_id
GROUP BY p.size
ORDER BY total_orders DESC;
```



Business Problem 7

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

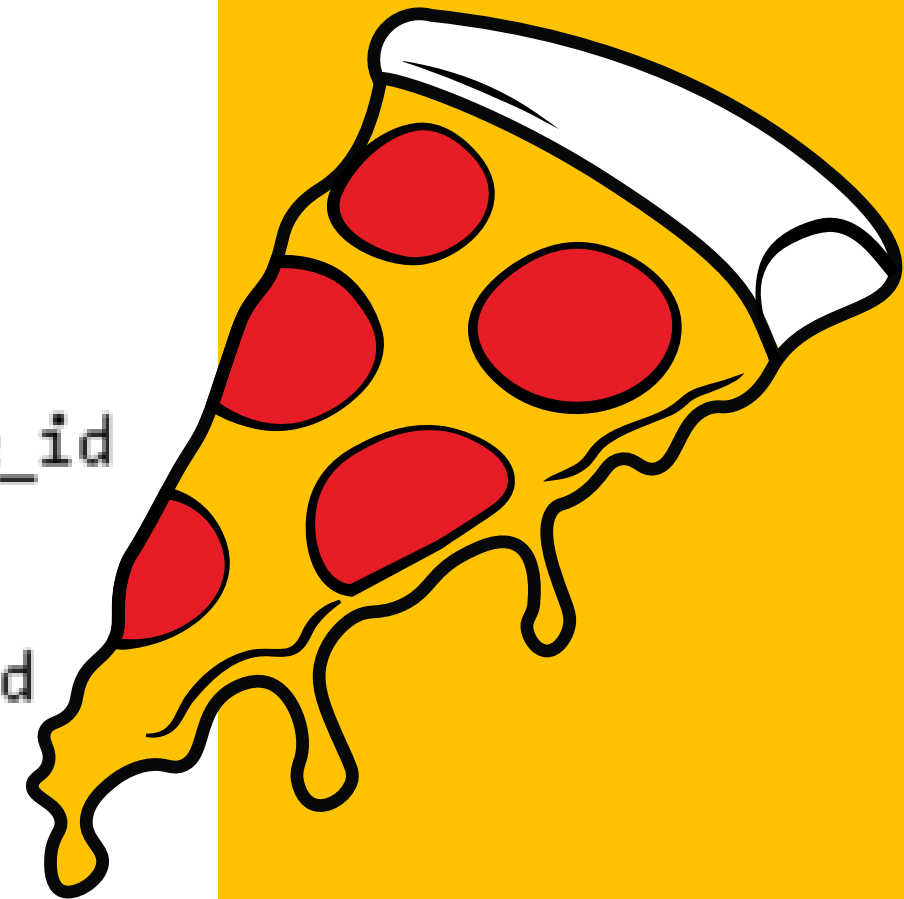
```
SELECT
    pp.name, SUM(o.quantity) AS quantities
FROM
    pizza_types AS pp
    JOIN
    pizzas AS p ON pp.pizza_type_id = p.pizza_type_id
    JOIN
    orders_details AS o ON o.pizza_id = p.pizza_id
GROUP BY pp.name
ORDER BY quantities DESC
LIMIT 5;
```



Business Problem 8

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

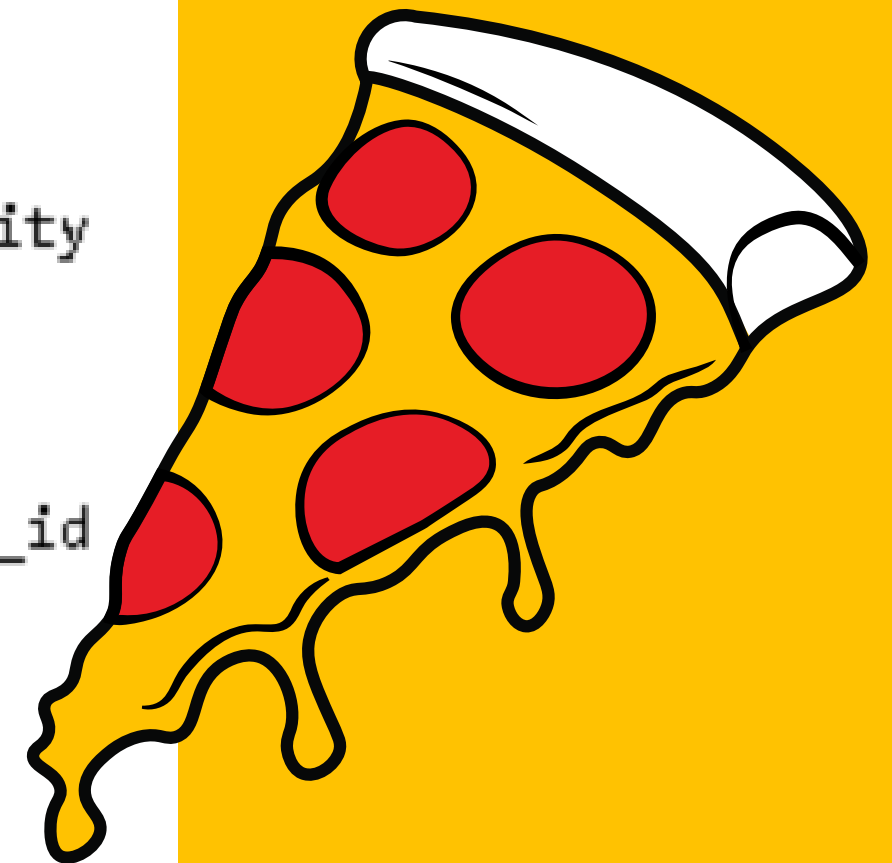
```
SELECT
    p.category, SUM(o.quantity) AS total_quantity
FROM
    pizza_types AS p
    JOIN
    pizzas AS a ON p.pizza_type_id = a.pizza_type_id
    JOIN
    orders_details AS o ON o.pizza_id = a.pizza_id
GROUP BY p.category
ORDER BY total_quantity DESC;
```



Business Problem 10

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

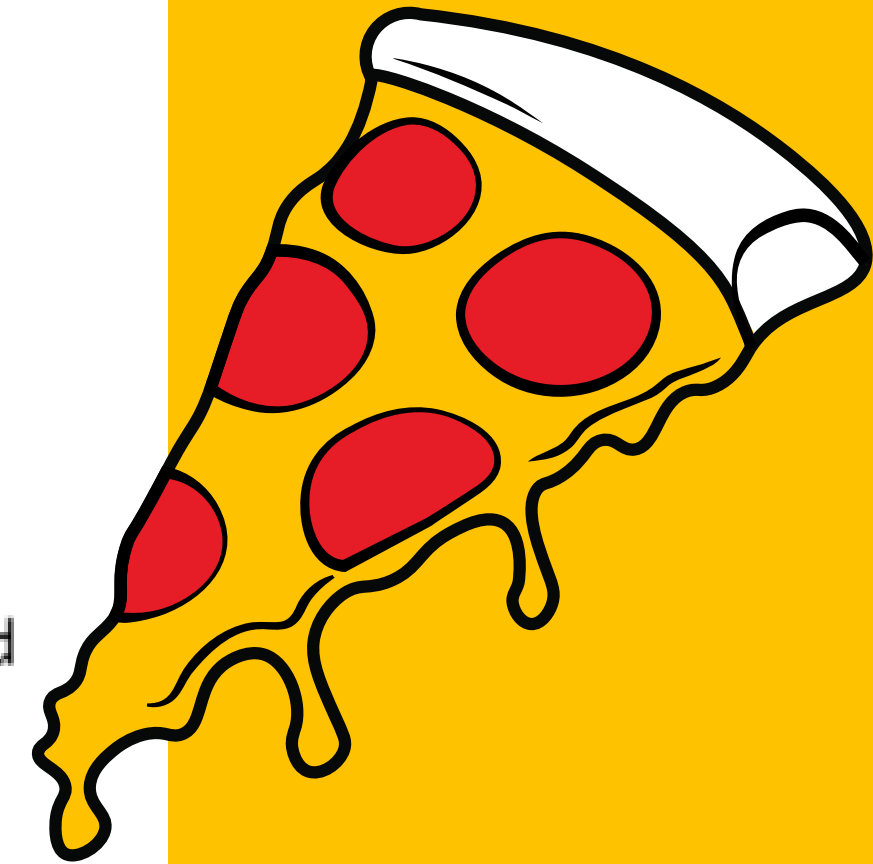
```
SELECT
    ROUND(AVG(total_quantity))
FROM
    (SELECT
        o.order_date, SUM(d.quantity) AS total_quantity
    FROM
        orders AS o
    JOIN orders_details AS d ON o.order_id = d.order_id
    GROUP BY o.order_date) AS avg_number
```



Business Problem 11

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

```
SELECT
    t.name, SUM(o.quantity * p.price) AS total_price
FROM
    pizzas AS p
    JOIN
        orders_details AS o ON p.pizza_id = o.pizza_id
    JOIN
        pizza_types AS t ON p.pizza_type_id = t.pizza_type_id
GROUP BY t.name
ORDER BY total_price desc
limit 3;
```



Business Problem 12

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

```
SELECT
  t.category,
  ROUND(SUM(o.quantity * p.price) / (SELECT
    ROUND(SUM(o.quantity * p.price), 2) AS total_sales
  FROM
    pizzas AS p
    JOIN
    orders_details AS o ON p.pizza_id = o.pizza_id) * 100,
  2) AS revenue
FROM
  orders_details AS o
  JOIN
  pizzas AS p ON o.pizza_id = p.pizza_id
  JOIN
  pizza_types AS t ON p.pizza_type_id = t.pizza_type_id
GROUP BY t.category
ORDER BY revenue DESC;
```



Business Problem 13

ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

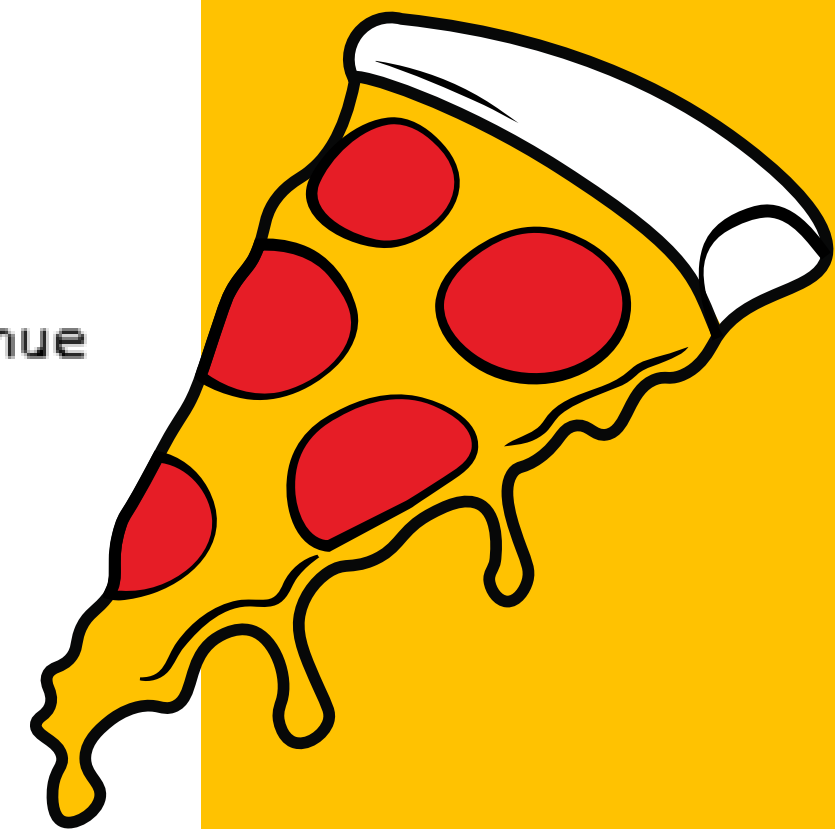
```
select order_date,  
Round(sum(revenue) over(order by order_date),2) as cum_revenue  
from  
(select o.order_date, sum(d.quantity * p.price) as revenue  
from orders as o join orders_details as d on o.order_id = d.order_id  
join pizzas as p on p.pizza_id = d.pizza_id  
group by o.order_date) as sales;
```



Business Problem 14

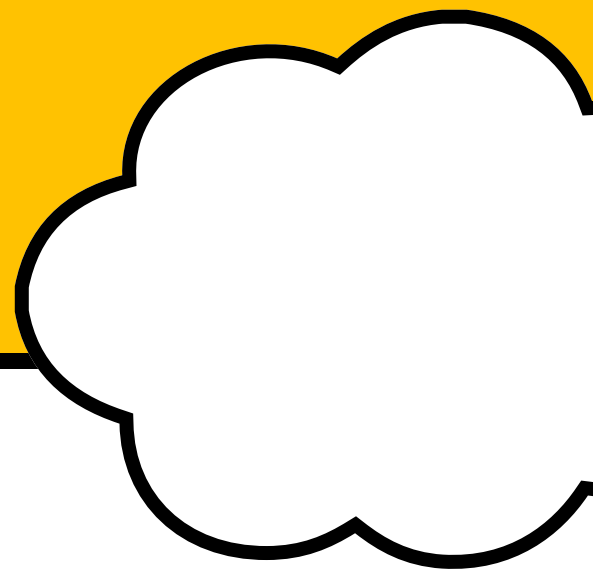
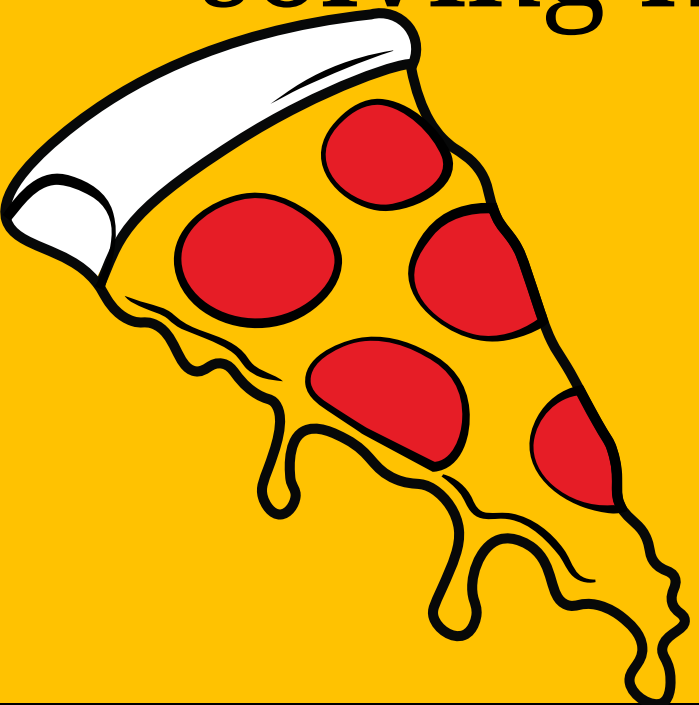
DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE FOR EACH PIZZA CATEGORY.

```
select name, category, Round(revenue, 2)
from
  (select name, category, revenue,
    rank() over(partition by category order by revenue) as rn
  from
    (select p.name, p.category, sum(o.quantity * pp.price) as revenue
    from pizza_types as p join pizzas as pp
    on p.pizza_type_id = pp.pizza_type_id
    join orders_details as o
    on o.pizza_id = pp.pizza_id
    group by p.name, p.category) as a) as b
where rn <= 3;
```



CONCLUSION

This project highlights my ability to handle complex SQL queries and provides solutions to real-world business problems in the context of a food service like Dominos. The approach taken here demonstrates a structured problem-solving methodology, data manipulation skills, and the ability to derive actionable insights from data.



THANK
YOU!

