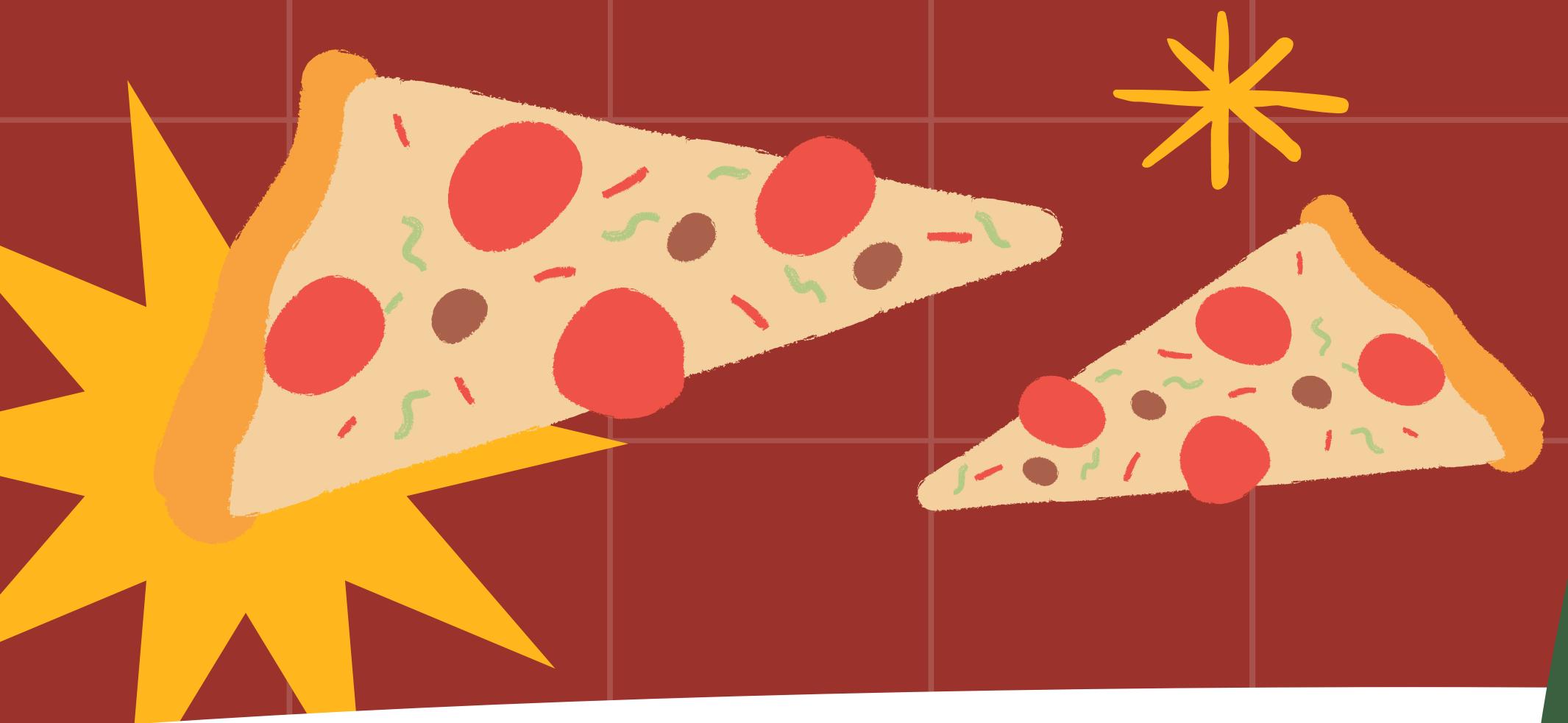


PIZZA SALES

SQL Project





LIST OF CONTENTS

Introduction

Database

List Of Tables

Questions

Code

INTRODUCTION

This project analyzes pizza sales data to identify trends and improve business strategies.

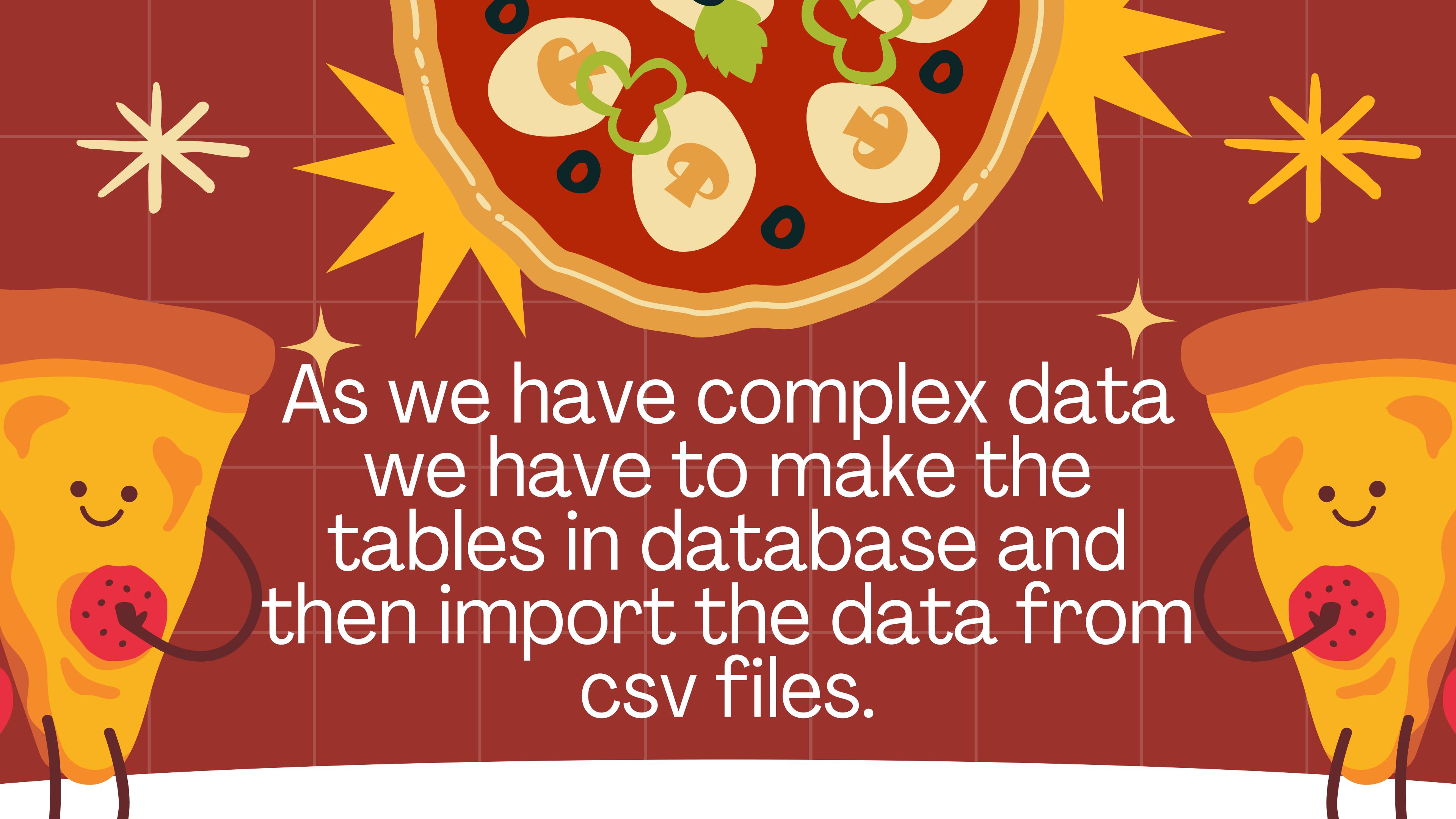
Let's start our adventure in the world of pizza!

DATABASE

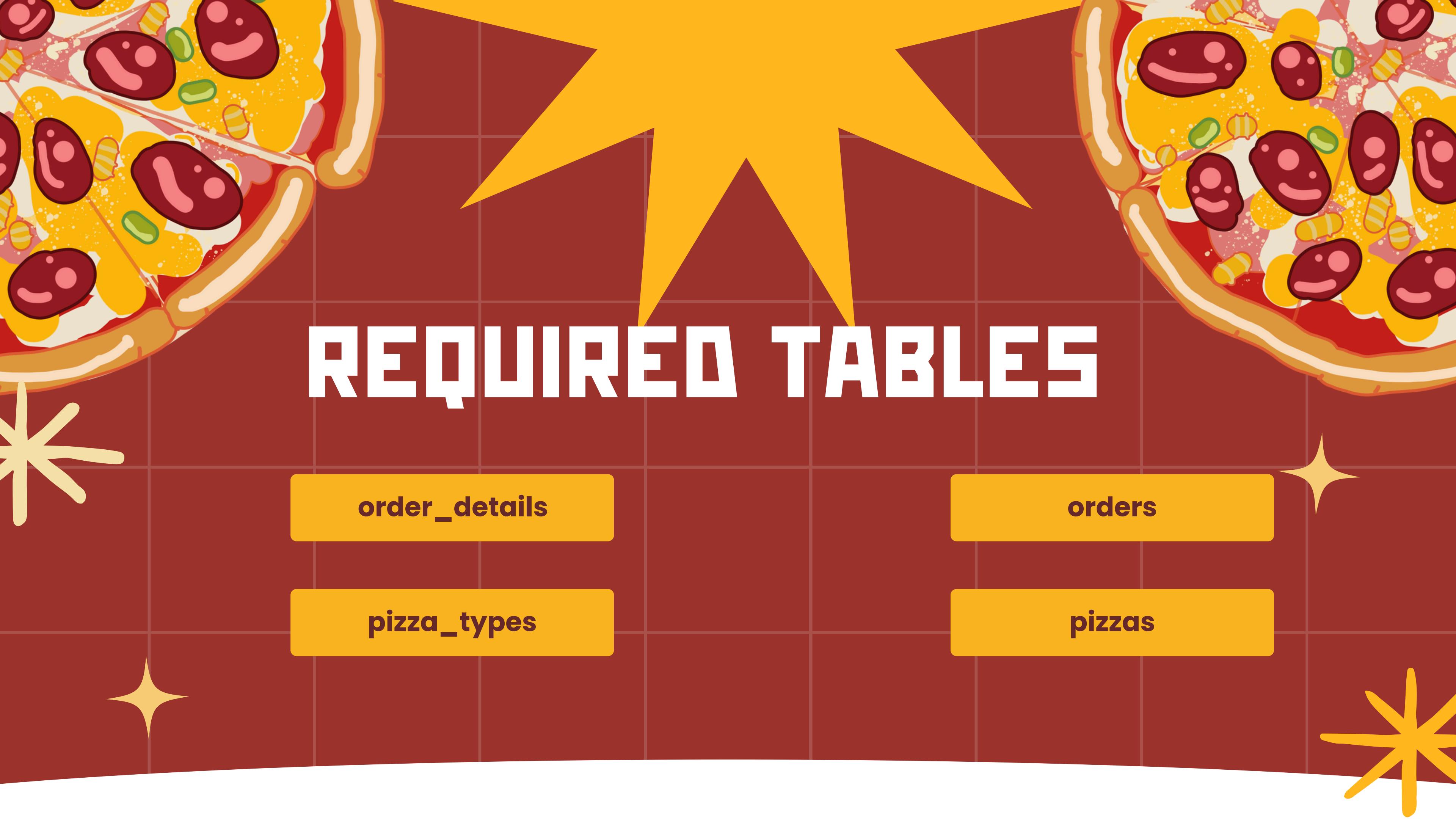
To Analyse the data we have our own database in which all the relevant data related to sales

- We will work on MySQL Workbench.
- Make sure all the required database should be there.
- Import Data file in MySQL.
- We have four tables in our Database.





As we have complex data
we have to make the
tables in database and
then import the data from
csv files.



REQUIRED TABLES

order_details

pizza_types

orders

pizzas

QUESTIONER



- Calculate the total revenue generated from the pizza's sales.
- Identify the highest price pizza.
- Identify the most common pizza size ordered.
- List the top 5 most ordered pizza types along with their quantities.
- Join the necessary tables to find the total quantity of each pizza category ordered.
- Join relevant table to find the category wise distribution of pizzas.
- Group the orders by date and calculate the avg numbers of pizzas order per day.
- Determine the top 3 most ordered pizza types based on revenue.
- Calculate the percentage contribution of each pizza types 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.

```
1 • CREATE DATABASE DOMINOZ;
2 • USE DOMINOZ;
3
4 -- BASIC
5
6 -- RETRIEVE THE TOTAL NUMBERS OF ORDERS PLACED
7
8 • SELECT COUNT(ORDER_ID) AS TOTAL_ORDERS FROM ORDERS;
9
10 -- CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES
11
12 • SELECT
13     ROUND(SUM(ORDER_DETAILS.QUANTITY * PIZZAS.PRICE),2) AS TOTAL_SALES
14     FROM
15     ORDER_DETAILS
16     LEFT JOIN
17     PIZZAS ON PIZZAS.PIZZA_ID = ORDER_DETAILS.PIZZA_ID;
```

Here we have used the two tables:

- **order_deatils**
- **pizzas**

Result Grid	
	TOTAL_SALES
▶	817860.05

```
18
19      -- IDENTIFY THE HIGHEST PRICED PIZZA
20
21 •   SELECT
22      PIZZA_TYPES.NAME, PIZZAS.PRICE
23  FROM
24      PIZZA_TYPES
25      JOIN
26      PIZZAS ON PIZZA_TYPES.PIZZA_TYPE_ID = PIZZAS.PIZZA_TYPE_ID
27  ORDER BY PIZZAS.PRICE DESC
28  LIMIT 1;
29
```

Here we have used the two tables:

- **pizza_types**
- **pizzas**

Result Grid | Filter Rows

	NAME	PRICE
▶	The Greek Pizza	35.95

```
29  
30      -- IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED  
31  
32 •   SELECT  
33      PIZZAS.SIZE,  
34      COUNT(ORDER_DETAILS.ORDER_DETAILS_ID) AS ORDER_COUNT  
35  FROM  
36      PIZZAS  
37      JOIN  
38      ORDER_DETAILS ON PIZZAS.PIZZA_ID = ORDER_DETAILS.PIZZA_ID  
39  GROUP BY PIZZAS.SIZE  
40  ORDER BY ORDER_COUNT DESC;
```

Here we have used the two tables:

- **order_deatils**
- **pizzas**

Result Grid | Filter Rows:

	SIZE	ORDER_COUNT
▶	L	18526
	M	15385
	S	14137
	XL	544
	XXL	28

```

41
42    -- LIST THE TOP 5 MOST ORDERED PIZZA TYPES ALONG WITH THEIR QUANTITIES
43
44 • SELECT
45     PIZZA_TYPES.NAME, SUM(ORDER_DETAILS.QUANTITY) AS QUANTITY
46 FROM
47     PIZZA_TYPES
48     JOIN
49     PIZZAS ON PIZZA_TYPES.PIZZA_TYPE_ID = PIZZAS.PIZZA_TYPE_ID
50     JOIN
51     ORDER_DETAILS ON ORDER_DETAILS.PIZZA_ID = PIZZAS.PIZZA_ID
52 GROUP BY PIZZA_TYPES.NAME
53 ORDER BY QUANTITY DESC
54 LIMIT 5;

```

Here we have used the three tables:

- **order_deatils**
- **pizza_types**
- **pizzas**

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

```

57
58    -- JOIN THE NECESSARY TABLES TO FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED
59
60 • SELECT
61     PIZZA_TYPES.CATEGORY,
62     SUM(ORDER_DETAILS.QUANTITY) AS QUANTITY
63 FROM
64     PIZZA_TYPES
65     JOIN
66         PIZZAS ON PIZZA_TYPES.PIZZA_TYPE_ID = PIZZAS.PIZZA_TYPE_ID
67     JOIN
68         ORDER_DETAILS ON ORDER_DETAILS.PIZZA_ID = PIZZAS.PIZZA_ID
69 GROUP BY PIZZA_TYPES.CATEGORY
70 ORDER BY QUANTITY DESC;

```

Here we have used the three tables:

- **order_deatils**
- **pizza_types**
- **pizzas**

Result Grid | Filter Rows:

	CATEGORY	QUANTITY
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

```
81  
82      -- JOIN RELEVANT TABLE TO FIND THE CATEGORY WISE DISTRIBUTION OF PIZZAS  
83  
84 •   SELECT  
85       CATEGORY, COUNT(NAME)  
86   FROM  
     •  
87       PIZZA_TYPES  
88   GROUP BY CATEGORY;
```

Here we have only single table.

Result Grid | Filter Rows:

	CATEGORY	COUNT(NAME)
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9

```
89  
90    -- GROUP THE ORDERS BY DATE AND CALCULATE THE AVG NUMBER OF PIZZAS ORDER PER DAY  
91  
92 •  SELECT  
93      ROUND(AVG(quantity), 0) AS Quantity  
94  FROM  
95  (SELECT  
96      ORDERS.ORDER_DATE, SUM(ORDER_DETAILS.QUANTITY) AS quantity  
97  FROM  
98      ORDERS  
99  JOIN ORDER_DETAILS ON ORDERS.ORDER_ID = ORDER_DETAILS.ORDER_ID  
100     GROUP BY ORDERS.ORDER_DATE) AS ORDER_QUANTITY;
```

Here we have used the two tables:

- **order_deatils**
- **orders**

Result Grid	
	Quantity
▶	138

```

101
102    -- Determine the top 3 most ordered pizza types based on revenue.
103
104 •   SELECT
105     PIZZA_TYPES.NAME,
106     SUM(ORDER_DETAILS.QUANTITY * PIZZAS.PRICE) AS REVENUE
107   FROM
108     PIZZA_TYPES
109     JOIN
110       PIZZAS ON PIZZAS.PIZZA_TYPE_ID = PIZZA_TYPES.PIZZA_TYPE_ID
111     JOIN
112       ORDER_DETAILS ON ORDER_DETAILS.PIZZA_ID = PIZZAS.PIZZA_ID
113   GROUP BY PIZZA_TYPES.NAME
114   ORDER BY REVENUE DESC
115   LIMIT 3;

```

Here we have used the three tables:

- **order_deatils**
- **pizza_types**
- **pizzas**

Result Grid | Filter Rows:

	NAME	REVENUE
▶	The Thai Chicken Pizza	43434.25
	The Barbecue Chicken Pizza	42768
	The California Chicken Pizza	41409.5

```

118
119      -- Calculate the percentage contribution of each pizza type to total revenue
120
121 •   SELECT
122     PIZZA_TYPES.CATEGORY,
123     (SUM(ORDER_DETAILS.QUANTITY * PIZZAS.PRICE) / (SELECT
124         ROUND(SUM(ORDER_DETAILS.QUANTITY * PIZZAS.PRICE),
125             2) AS TOTAL_SALES
126
127     FROM
128         ORDER_DETAILS
129         LEFT JOIN *
130             PIZZAS ON PIZZAS.PIZZA_ID = ORDER_DETAILS.PIZZA_ID)) * 100 AS REVENUE
131
132     FROM
133         PIZZA_TYPES
134         JOIN
135             PIZZAS ON PIZZA_TYPES.PIZZA_TYPE_ID = PIZZAS.PIZZA_TYPE_ID
136         JOIN
137             ORDER_DETAILS ON ORDER_DETAILS.PIZZA_ID = PIZZAS.PIZZA_ID
138
139     GROUP BY PIZZA_TYPES.CATEGORY
140
141     ORDER BY REVENUE DESC;

```

Here we have used the three tables:

- **order_deatils**
- **pizza_types**
- **pizzas**

Result Grid | Filter Rows:

	CATEGORY	REVENUE
▶	Classic	26.90596025566967
	Supreme	25.45631126009862
	Chicken	23.955137556847287
	Veggie	23.682590927384577

```

138
139    -- Analyze the cumulative revenue generated over time.
140
141 •   SELECT ORDER_DATE,
142      SUM(REVENUE) OVER (ORDER BY ORDER_DATE) AS CUM_REVENUE
143      FROM
144      (SELECT ORDERS.ORDER_DATE,
145          SUM(ORDER_DETAILS.QUANTITY * PIZZAS.PRICE) AS REVENUE
146          FROM ORDER_DETAILS JOIN PIZZAS
147          ON ORDER_DETAILS.PIZZA_ID = PIZZAS.PIZZA_ID
148          JOIN ORDERS
149          ON ORDERS.ORDER_ID = ORDER_DETAILS.ORDER_ID
150          GROUP BY ORDERS.ORDER_DATE) AS SALES;
151

```

Here we have used the three tables:

- **order_deatils**
- **orders**
- **pizzas**

Result Grid		Filter Rows:
	ORDER_DATE	CUM_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
	2015-01-14	32358.700000000004
	2015-01-15	34343.5000000001
	2015-01-16	36937.65000000001
	2015-01-17	39001.75000000001
	2015-01-18	40978.60000000006
	2015-01-19	43365.75000000001
	2015-01-20	45763.65000000001
	2015-01-21	47804.20000000001
	2015-01-22	50300.90000000001
	2015-01-23	52724.60000000006
	2015-01-24	55013.85000000006
	2015-01-25	56631.40000000001
	2015-01-26	58515.80000000001
	2015-01-27	61043.85000000001

```

151
152    -- Determine the top 3 most ordered pizza types based on revenue for each pizza category.
153
154 •  SELECT NAME, REVENUE FROM
155     (SELECT CATEGORY, NAME , REVENUE,
156      RANK() OVER(PARTITION BY CATEGORY ORDER BY REVENUE DESC) AS RN
157     FROM
158     (SELECT PIZZA_TYPES.CATEGORY, PIZZA_TYPES.NAME,
159      SUM( ( ORDER_DETAILS.QUANTITY) * PIZZAS.PRICE) AS REVENUE
160     FROM PIZZA_TYPES JOIN PIZZAS
161     ON PIZZA_TYPES.PIZZA_TYPE_ID = PIZZAS.PIZZA_TYPE_ID
162     JOIN ORDER_DETAILS
163     ON ORDER_DETAILS.PIZZA_ID = PIZZAS.PIZZA_ID
164     GROUP BY PIZZA_TYPES.CATEGORY,PIZZA_TYPES.NAME) AS A) AS B
165     WHERE RN<=3;

```

Here we have used the three tables:

- **order_deatils**
- **pizza_types**
- **pizzas**

	NAME	REVENUE
▶	The Thai Chicken Pizza	43434.25
	The Barbecue Chicken Pizza	42768
	The California Chicken Pizza	41409.5
	The Classic Deluxe Pizza	38180.5
	The Hawaiian Pizza	32273.25
	The Pepperoni Pizza	30161.75
	The Spicy Italian Pizza	34831.25
	The Italian Supreme Pizza	33476.75
	The Sicilian Pizza	30940.5
●	The Four Cheese Pizza	32265.70000000065
	The Mexicana Pizza	26780.75
	The Five Cheese Pizza	26066.5

VARIATIONS



CLASSIC



SUPREME



BBQ Chicken Pizza



VEGGIE

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