**SQL Practice Questions for Superstore Dataset (MySQL Workbench)**

**Step 1: Loading the Data**

1. **Create a table to store the Superstore dataset.**
   * **Ensure the correct data types for each column.**
2. **Load the data into the table.**
   * **Use LOAD DATA INFILE or import via MySQL Workbench.**
3. **Check if all rows are loaded properly.**
   * **Example: If the dataset has 48620 data rows, COUNT(\*) should return 48620.**

select

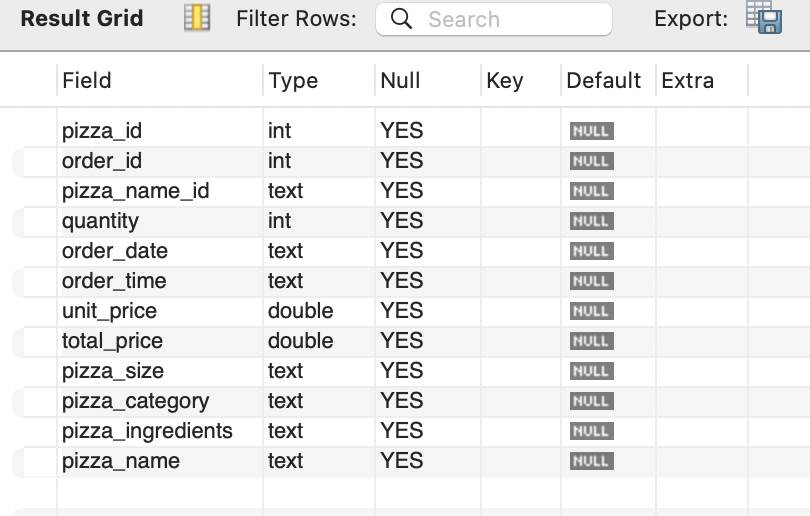
count(pizza\_id)

From pizza\_sales ;

1. **Verify the column names and data types.**

Use DESCRIBE pizza\_sales data; to confirm the structure.

**DESC pizza\_sales;**

****

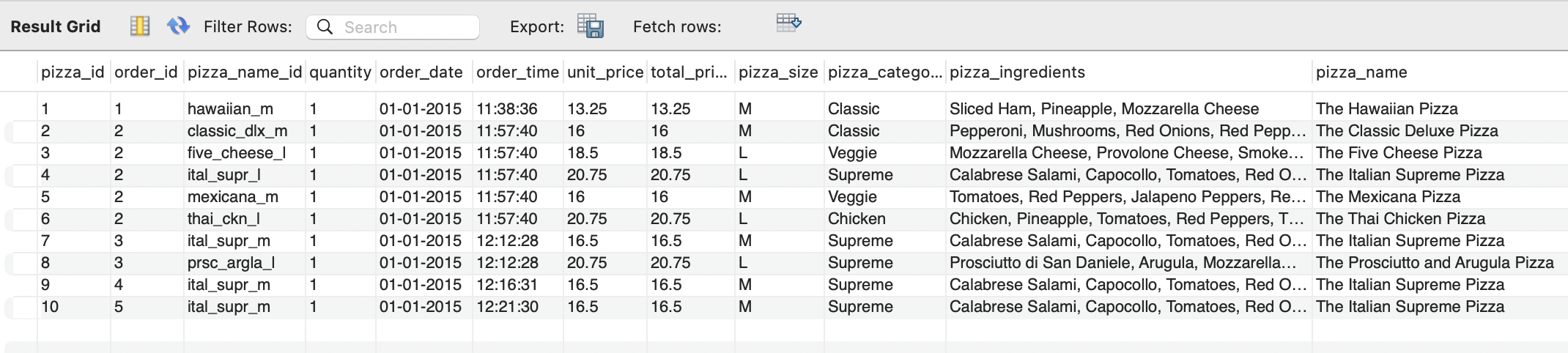
**Step 2: Basic Data Exploration**

1. **Preview the first 10 rows.**

select \*

FROM pizza\_sales

limit 10;



1. **Count the total number of unique customers.**

SELECT COUNT(DISTINCT order\_id) FROM pizza\_sales;

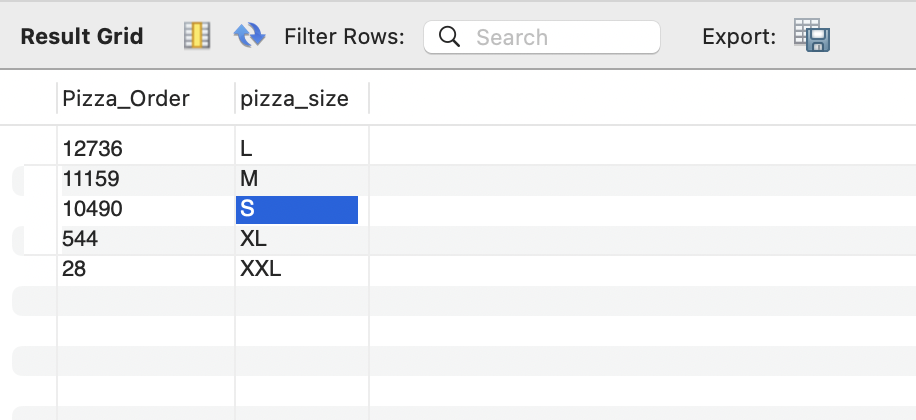
Return 21350

1. **List all unique sales by size.**

SELECT COUNT(DISTINCT order\_id) as Pizza\_Order, pizza\_size

FROM pizza\_sales

group by pizza\_size;



1. **Find the number of unique catagories in the dataset.**

**select distinct pizza\_category**

**From pizza\_sales;**

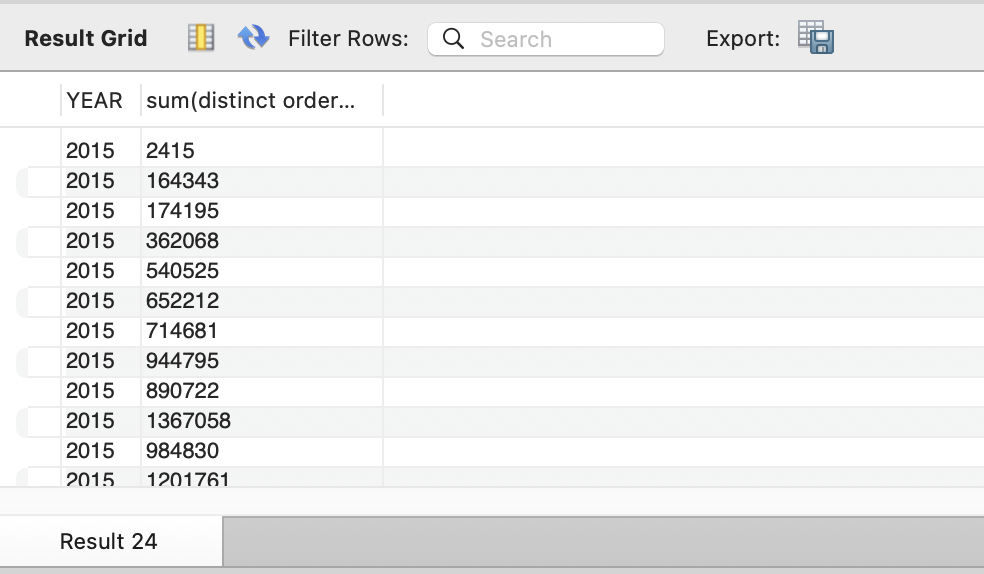
**A screenshot of a computer

AI-generated content may be incorrect.**

1. **Find the total number of orders per year.**

SELECT DISTINCT substring(order\_date, 7,4)AS YEAR, sum(distinct order\_id) FROM pizza\_sales

group by order\_date;

****

**Step 3: Filtering & Sorting Data**

1. **Find all orders placed in December 2015.**

SELECT COUNT(order\_id) AS unique\_orders\_dec

FROM pizza\_sales

WHERE MONTH(STR\_TO\_DATE(order\_date, '%d-%m-%Y')) = 12;

A screenshot of a search box

AI-generated content may be incorrect.

1. **Find the top 5 Months with the highest total sales.**

**First find all total sales of each month**

SELECT

MONTH(STR\_TO\_DATE(order\_date, '%d-%m-%Y')) AS month,

YEAR(STR\_TO\_DATE(order\_date, '%d-%m-%Y')) AS year,

SUM(total\_price) AS total\_sales

FROM pizza\_sales

GROUP BY year, month

ORDER BY year, month;

**A screenshot of a computer

AI-generated content may be incorrect.**

**Then top five**

SELECT

MONTH(STR\_TO\_DATE(order\_date, '%d-%m-%Y')) AS month,

YEAR(STR\_TO\_DATE(order\_date, '%d-%m-%Y')) AS year,

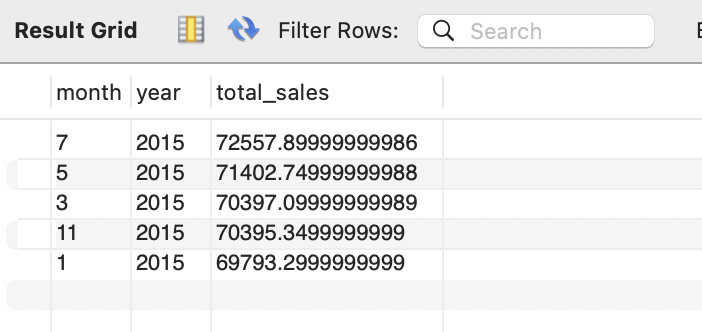
SUM(total\_price) AS total\_sales

FROM pizza\_sales

GROUP BY year, month

ORDER BY total\_sales DESC

limit 5;

****

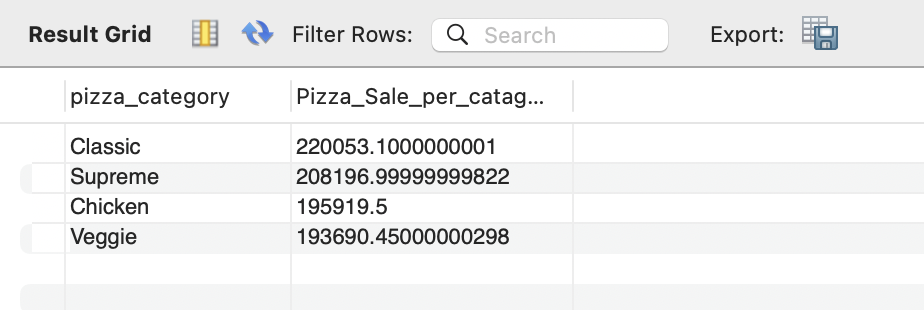
1. **Find the top 3 most profitable product categories.**

select distinct pizza\_category, sum(total\_price) As Pizza\_Sale\_per\_catagory

From pizza\_sales

group by pizza\_category

order by Pizza\_Sale\_per\_catagory Desc;



select distinct pizza\_category, sum(total\_price) As Pizza\_Sale\_per\_catagory

From pizza\_sales

group by pizza\_category

order by Pizza\_Sale\_per\_catagory Desc

limit 3;

A screenshot of a search engine

AI-generated content may be incorrect.

1. **Retrieve all orders where the discount given is more than 50%.**

**Step 4: Data Cleaning**

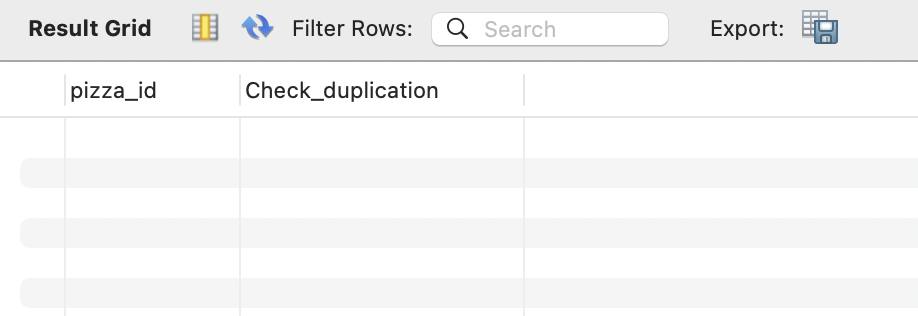
1. **Identify duplicate records.**

select pizza\_id, count(\*) As Check\_duplication

from pizza\_sales

group by pizza\_id

Having count(\*)>1 ;

****

1. **Check for missing values in important columns (order\_id and pizza\_id).**

select pizza\_id,order\_id

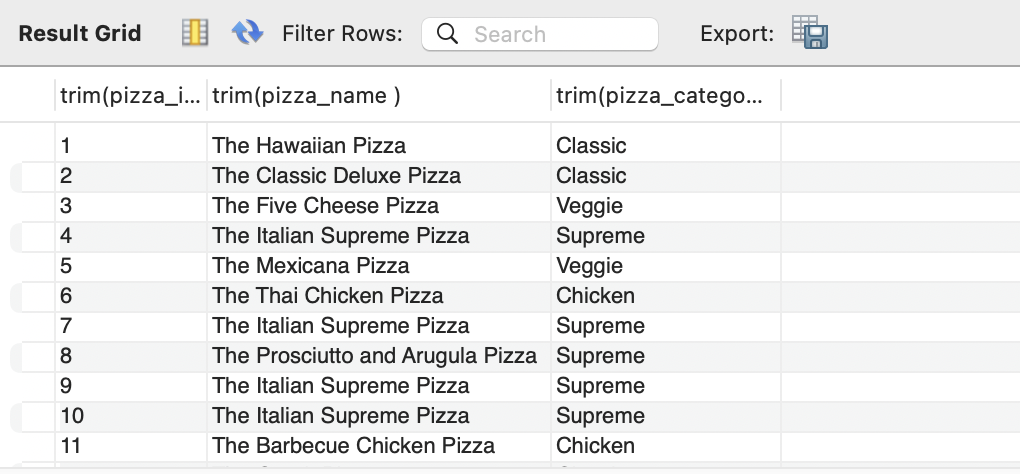
from pizza\_sales

where pizza\_id= null & order\_id=null ;

1. **Standardize the State column to remove extra spaces.**

select trim(pizza\_id), trim(pizza\_name ),trim(pizza\_category)

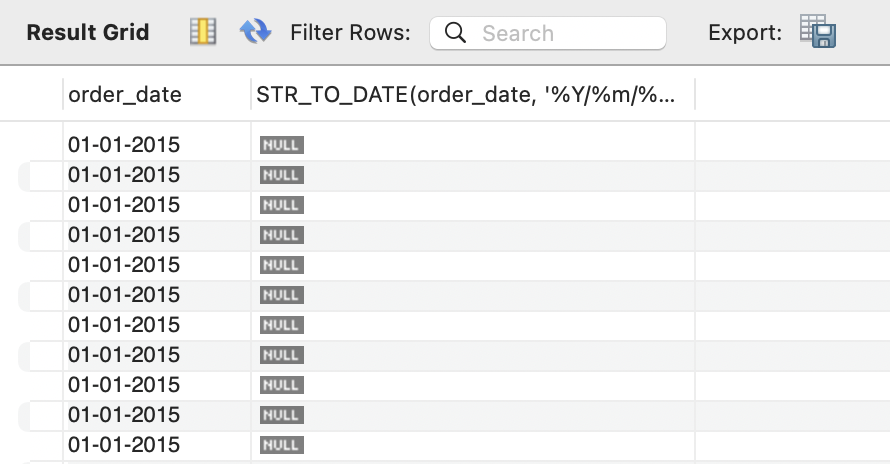
From pizza\_sales

****

1. **Convert Order Date to the correct date format (if needed).**

select order\_date, STR\_TO\_DATE(order\_date, '%d/%m/%Y')

from pizza\_sales;



1. **Ensure Profit column contains only numerical values.**

**Step 5: Aggregation & Grouping**

1. **Find total sales, total quantity, and total profit per year.**

SELECT

YEAR(STR\_TO\_DATE(order\_date, '%d-%m-%Y')) AS year,

SUM(total\_price) AS total\_sales,

Sum(quantity) As Total\_Quantity

FROM pizza\_sales

GROUP BY year;

**A screenshot of a computer

AI-generated content may be incorrect.**

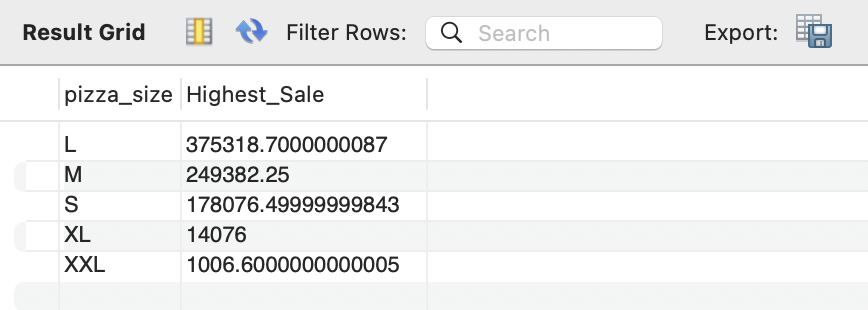
1. **Calculate the average discount per product category.**
2. **Find the pizza\_size who placed the highest number of orders.**

select pizza\_size, sum(total\_price) as Highest\_Sale

From pizza\_sales

group by pizza\_size

order by Highest\_Sale DESC;



1. **Determine the month with the highest total sales.**

SELECT

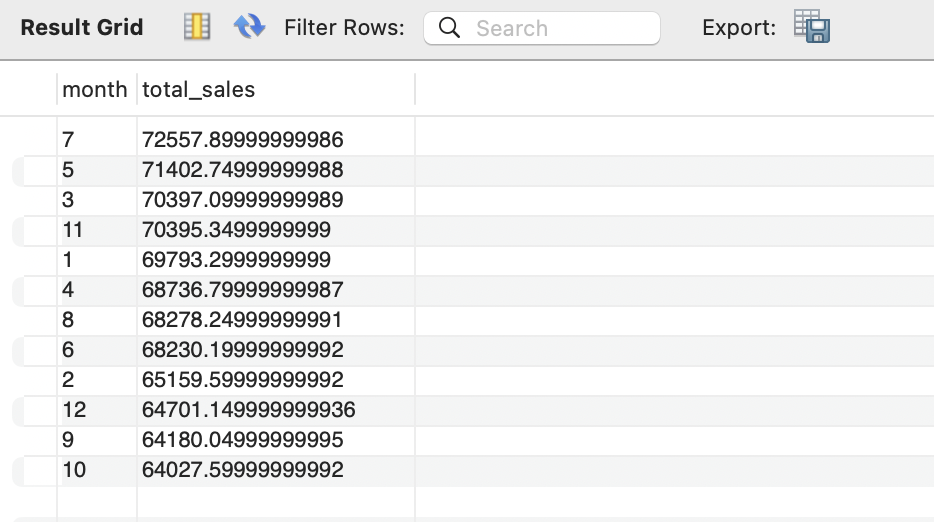
MONTH(STR\_TO\_DATE(order\_date, '%d-%m-%Y')) AS month,

SUM(total\_price) AS total\_sales

FROM pizza\_sales

GROUP BY month

ORDER BY total\_sales DESC;



**Step 6: Joins & Relationships**

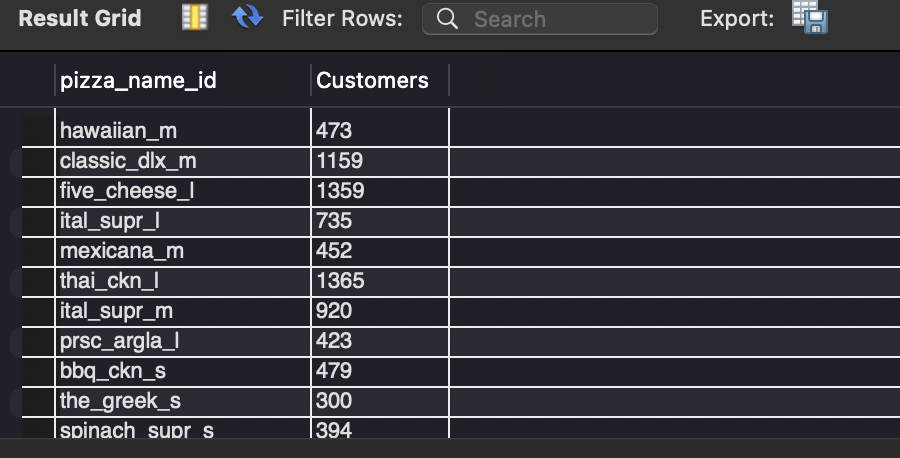
1. **Create a customers table and link it to superstore\_data using Customer\_ID.**
2. **Join customers with Pizza\_sales to get total sales per customer.**

select pizza\_name\_id, count(\*) As Customers

from pizza\_sales

group by pizza\_name\_id

Having count(\*)>1 ;



**Step 7: Advanced SQL**

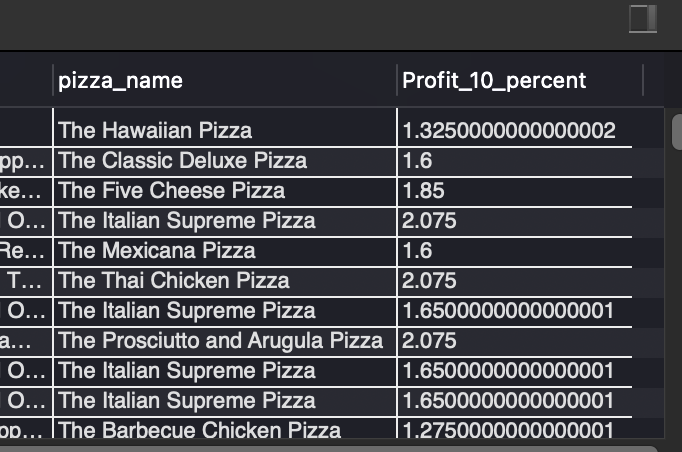
1. **Create a new column Profit Margin (%) and fill it up by the following formula:.**
   1. **(Profit/sales) \* 100**

alter Table pizza\_sales

add column Profit\_10\_percent varchar(100);

UPDATE pizza\_sales

SET profit\_10\_percent = total\_price \* 0.10;



1. **Identify the three least profitable products.**

select pizza\_name\_id, total\_price,profit\_10\_percent

from pizza\_sales

order by profit\_10\_percent DESC

limit 3;

A screenshot of a computer

AI-generated content may be incorrect.

1. **Rank customers by total sales.**
2. **Find the most frequently ordered product.**

select pizza\_name\_id, order\_id,pizza\_size

from pizza\_sales

order by pizza\_name\_id DESC

limit 10;

