Assignment – 1

**Instructions**

**Q 1. Create a table with Columns - ID, First\_Name, Last\_Name, Phone\_Number. Assign following constraints without naming them in a single SQL Syntax:**

ID - PRIMARY KEY Constraint

First\_Name - CHECK Constraint (Name needs to start with A)

Last\_Name - CHECK Constraint (Last\_Name needs to have exactly 5 characters)

Phone\_Number - UNIQUE Constraint

**Q 2. Create a table with Columns - ID, First\_Name, Last\_Name, Phone\_Number. Assign following constraints along with naming them in a single SQL Syntax:**

ID - PRIMARY KEY Constraint

First\_Name - CHECK Constraint (Name needs to start with A)

Last\_Name - CHECK Constraint (Last\_Name needs to have exactly 5 characters)

Phone\_Number - UNIQUE Constraint

**Q 3. Create 2 tables Customers and Orders.**

Customers - ID, Name, Phone\_Num, Address

Orders - ID, Customer\_ID, Total\_Order\_Amount

**-- Assign Primary Key constraints to suitable columns of both tables.**

**-- Assign Foreign Key Constraint to CustomerID referencing it to suitable column of Customers table.**

**Q 4. Import Customers table from here:** [*https://drive.google.com/file/d/1W3ueqDoTbpkjd63jPDywyBlwhkM2Csgc/view?usp=sharing*](https://drive.google.com/file/d/1W3ueqDoTbpkjd63jPDywyBlwhkM2Csgc/view?usp=sharing)

Define a Composite Primary Key Constraint on FirstName, LastName and City using the Alter Command.

**Q 5. Create following table:**

Users - ID, Name, Phone, Address, City, State, Country, Email.

**--Define a named Primary Key Constraint on Name, City and State.**

**Q 6. Add Constraints to Existing Table:** Given an existing table 'Employees' with columns - Emp\_ID, Emp\_Name, Department, Salary, Email. hint - If no table given you need to first create it then solve question

* Add a CHECK constraint to ensure Salary is greater than 3000.
* Add a UNIQUE constraint on Email.

**Q 7. Create Table with Multiple Check Constraints:** Create a table 'Library\_Books' with columns - Book\_ID, Title, Genre, Publication\_Year, Price.

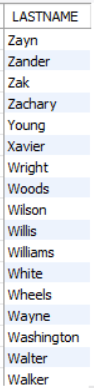
* Book\_ID - PRIMARY KEY Constraint
* Genre - CHECK constraint (must be either 'Fiction', 'Non-Fiction', 'Science', 'History')
* Publication\_Year - CHECK constraint (must be after 1900)
* Price - CHECK constraint (must be positive)

Assignment – 2

Q1. **Print the unique Last names of customers whose details are stored in the database.**

**Sort the result set in reverse alphabetical order of the Last names.**

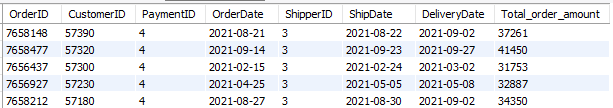
Sample Output:

[](https://cp.masaischool.com/assignments/3275/pi/1/solve)

Q2. **Print all details of Orders which were placed by Customers whose ID is a multiple of 10, payment method by Payment with ID 4, shipped by Shipper with ID 3 and which are greater than 30,000 in order value.**

**Sort the result set in descending order of Customer ID.**

**Sample Output**



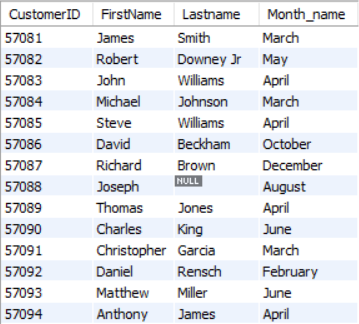
Q3. **Print the Customer First Name which comes last according to the alphabetical order.**

**Hint: Apply MAX function on the relevant column.**

Q4. **Print the CustomerID, First name, Last name, and Birth month name of each customer.**

**Sort the result in ascending order of CustomerID.**

**Sample Output:**

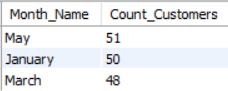


Q5. **Identify the top three months which have the highest number of Births irrespective of the year.**

**Print the Month Name followed by the count of customers.**

**Sort the result in descending order of Count.**

**Sample Output:**

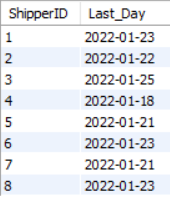


Q6. **Identify the last day of delivery for each Shipper.**

**Print Shipper ID followed by the Last Delivery Date.**

**Sort the result in ascending order of Shipper ID.**

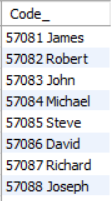
**Sample Output:**



Q7. **Print Customer Id and Customer First Name as space seperated single string.**

**Sort the result in ascending order of column values.**

**Sample Output:**

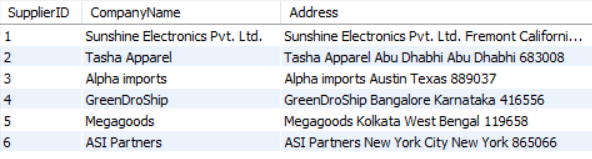


Q8. **Print SupplierID, Supplier Name and Address of the Supplier.**

**Here address refers to Company name, City, State, and Postal code as space seperated single string.**

**Sort the result in ascending order of SupplierID.**

**Sample Output:**

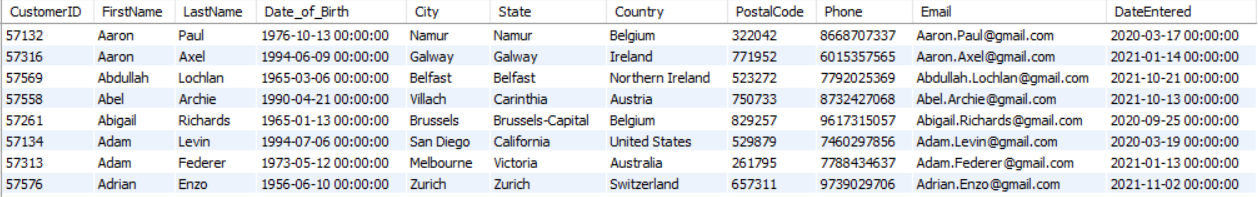


Q9. **Write a query to print all customer details.**

**Order your output in ascending order of the first 3 characters of FirstName.**

**For records with the same first 3 characters in the first name, sort them in ascending order of CustomerID.**

**Sample Output:**



Q10. **Calculate and Print the Average of Total Order Amount of all orders whose payment was made using Payment with ID 2.**

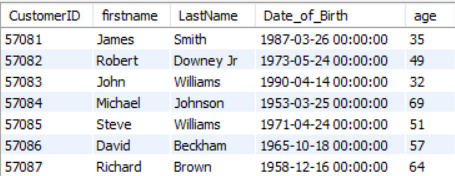
Assignment – 3

Q1. **Print Customer ID, First Name, Last Name, Date of Birth and Customer's current age.**

**Sort the result in ascending order of CustomerID.**

**Note: Use NOW() function as CP will not allow you to use GETDATE().**

**Sample Output**



Q2. **Identify the First Day of Delivery for each Shipper.**

**Print Shipper ID, Shipper Company Name followed by the First Delivery Date.**

**Sort the result in ascending order of ShipperID.**

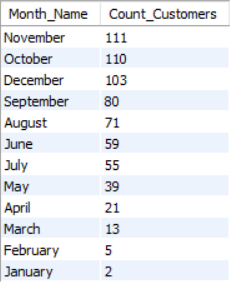
**Sample Output:**



Q3. **Print month names followed by the number of orders delivered in that particular month for all months of the year 2020.**

**Sort the result in descending order of order count.**

**Sample Output:**

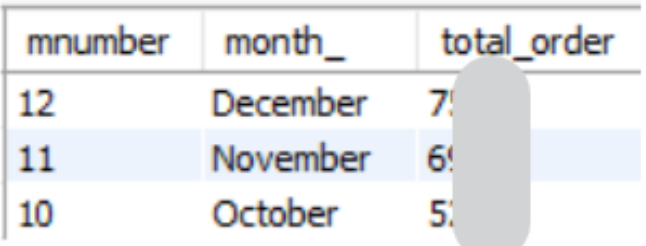


Q4. **Identify and print top three months where the most number of orders were delivered across both the years 2020 and 2021.**

**Print Month Number, Month Name, followed by the order count.**

**Sort the result in descending order of Count.**

Sample Output -

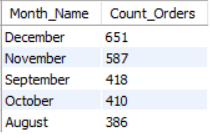


Q5. **Identify and print top 5 months where the most orders were delivered across 2021.**

**Print Month Name, followed the order count.**

**Sort the result in descending order of count.**

**Sample Output:**



Q6. **Identify and print the top 3 months where the least orders were delivered.**

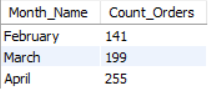
**Considering each of the months of all years as single entity.**

**For example, Jan of 2020 and 2021 will be considered as a single entity and so on**

**Print Month Name, followed by the order count.**

**Sort the result in ascending order of Count.**

**Sample Output:**



Q7. **Write a query to identify pairs of customers who belong to the same state.**

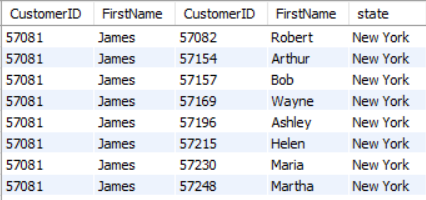
**Print First customer's id, first name, second customer's id, first Name and state.**

**Make sure you do not print a pair in which the customer ids are the same.**

**Let the first customer's id be less than the second customer's id.**

**Sort the result in ascending order of first customer's id.**

**For records with the same first customer's id, sort them in ascending order of second customer's id.**



Q8. **Calculate the total spend across orders of customers whose full name consists of the letter 'a'.**

**Print Customer ID, Full Name and Total Spend.**

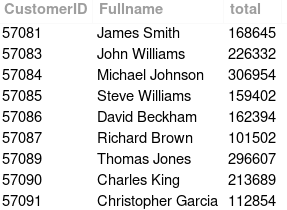
**Let the full name be combination of First Name and Last Name separated by a space.**

**Sort the result in ascending order of Customer ID.**

**If you see blank cells in the full name column that means you have concatenated a string with null.**

**For such cases just print the first name without any concatenation.**

Sample Output:



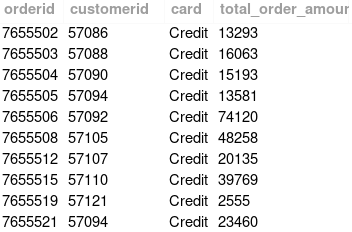
Q9. **Write a query to get the order IDs of all the orders which were paid either using Credit card or debit card.**

**Print Order ID, Customer ID, Payment type and Total Order Amount.**

**(Replace Credit card with 'Credit' and Debit card with 'Debit' - Use REPLACE Function and not CASE WHEN).**

**Sort the result in ascending order of Order ID.**

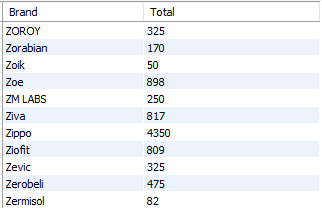
Sample Output:



Q10. **Write a query to find the total Sale price for each Brand of all products where the Quantity is an even number.**

**Sort the output in descending order of Brand.**

**Sample Output**



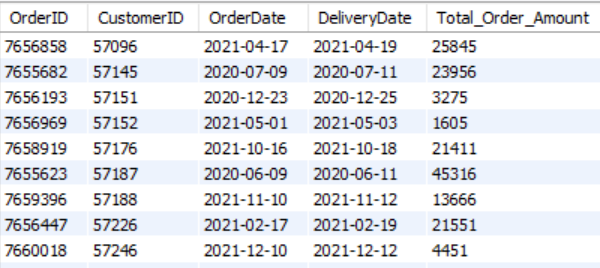
Assignment – 4

Q1. **Write a query to find the Orders which were delivered exactly in 2 days.**

**Print Order Id, Customer ID, Order Date, Delivery Date and Total Order Amount for each of these orders.**

**Order your output in terms of Customer ID asc.**

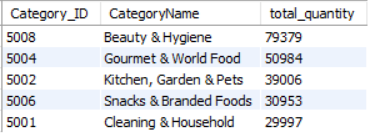
Sample Output -



Q2. **Identify the top 5 categories which had the highest quantity of products ordered.**

**Print Category ID, Category Name and Corresponding Total Quantity.**

**Sort the result in descending order of Total Quantity.**

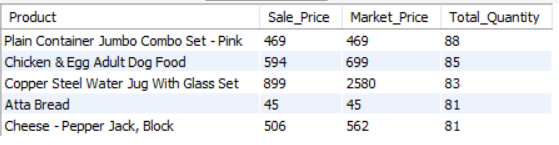


Q3. **Identify the top 5 products in terms of total quantity sold.**

**Print Product Name, Sale Price, Market Price and Total Quantity.**

**Order your output in descending order of total quantity for the same number of quantities order your output by Product.**

**Sample Output:**

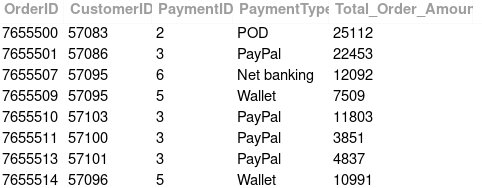


Q4. **Print Order ID, Customer ID, Payment ID, Payment Type, and Total Order Amount for all orders which were paid through by a non-card payment type.**

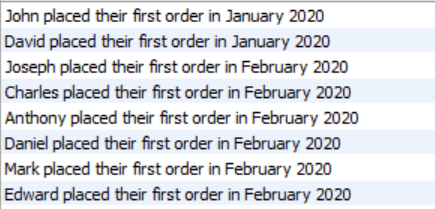
**Card payment types are 'Credit card' and 'Debit card'.**

**Sort the result in ascending order of Order ID**

Sample Output:



Q5. **For each customer print the following sentence. Consider only those customers who ordered at least once.**  
**Sort the output in ascending order on Customer Id.** **Example - James placed their first order in May 2020**



Q6. **Write a query to find the average discount percenatge for each Category.**

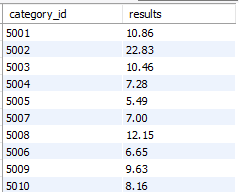
Discount % = ((Market Price - Sale Price)/Market Price)\* 100

**Print the Category ID along with average discount percentage(Round the avg discount percentage to 2 decimals).**

**Sort the output in ascending order on CategoryId.**

**HINT: Use CTE for this question.**

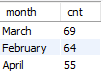
**Sample Output**



Q7. **Write a query to find the top 3 months with the highest customer acquisition.**

**Print the months in string format like January, February,etc along with the count of customers entered in that month.**

Sample Output :

[](https://imgbb.com/)

Q8. **Write a query to print all the product details with Sub category as 'All Purpose Cleaners' and 'Bakeware'.**

**Print all the columns present in Products table.**

**Order your output in ascending order of Product ID.**

**Solve the problem using CTEs.**

**Sample Output**

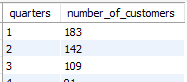


Q9. **Write a query to find the number of customers who entered each Quarter.**

**Print the Quarter and the count of customers who entered in that Quarter.**

**Order your output in ascending order of Quarter.**

**Sample Output**



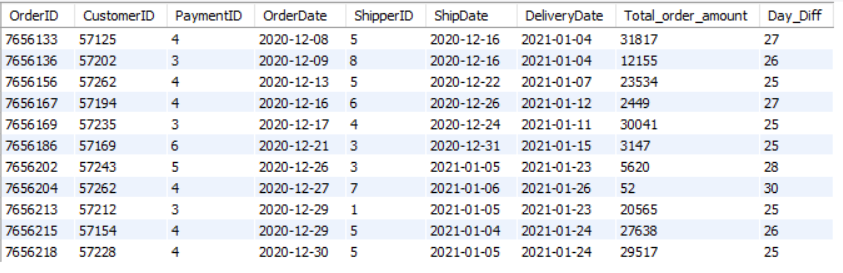
Q10. **Identify the orders which were ordered in 2020 but delivered in 2021.**

**Print all details of such orders along with the difference in days from the order date to the delivery date.**

**Filter to only print details of orders where the difference between the 2 dates in terms of days in greater than or equal to 25.**

**Sort the result in ascending order of OrderID.**

Required Result:



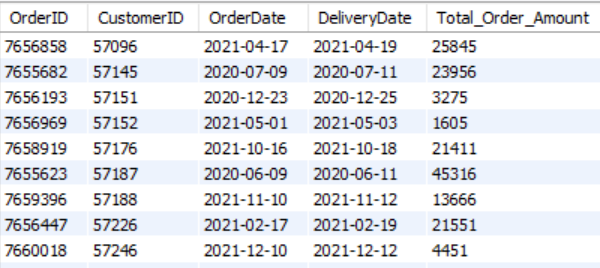
ASSIGNMENT – 5

Q1. **Write a query to find the Orders which were delivered exactly in 2 days.**

**Print Order Id, Customer ID, Order Date, Delivery Date and Total Order Amount for each of these orders.**

**Order your output in terms of Customer ID asc.**

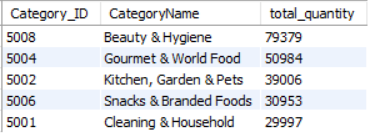
Sample Output -



Q2. **Identify the top 5 categories which had the highest quantity of products ordered.**

**Print Category ID, Category Name and Corresponding Total Quantity.**

**Sort the result in descending order of Total Quantity.**

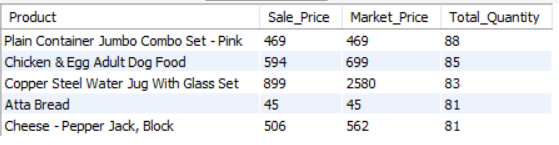


Q3. **Identify the top 5 products in terms of total quantity sold.**

**Print Product Name, Sale Price, Market Price and Total Quantity.**

**Order your output in descending order of total quantity for the same number of quantities order your output by Product.**

**Sample Output:**

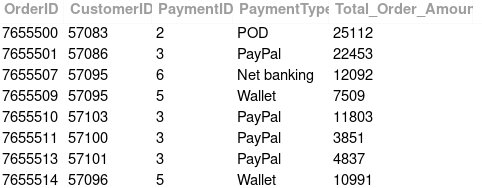


Q4. **Print Order ID, Customer ID, Payment ID, Payment Type, and Total Order Amount for all orders which were paid through by a non-card payment type.**

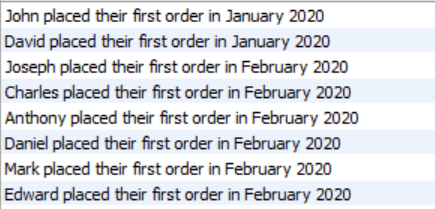
**Card payment types are 'Credit card' and 'Debit card'.**

**Sort the result in ascending order of Order ID**

Sample Output:



Q5. **For each customer print the following sentence. Consider only those customers who ordered at least once.**  
**Sort the output in ascending order on Customer Id.** **Example - James placed their first order in May 2020**



Q6. **Write a query to find the average discount percenatge for each Category.**

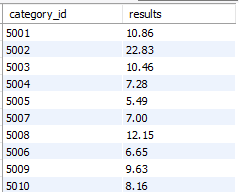
Discount % = ((Market Price - Sale Price)/Market Price)\* 100

**Print the Category ID along with average discount percentage(Round the avg discount percentage to 2 decimals).**

**Sort the output in ascending order on CategoryId.**

**HINT: Use CTE for this question.**

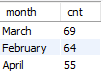
**Sample Output**



Q7. **Write a query to find the top 3 months with the highest customer acquisition.**

**Print the months in string format like January, February,etc along with the count of customers entered in that month.**

Sample Output :

[](https://imgbb.com/)

Q8. **Write a query to print all the product details with Sub category as 'All Purpose Cleaners' and 'Bakeware'.**

**Print all the columns present in Products table.**

**Order your output in ascending order of Product ID.**

**Solve the problem using CTEs.**

**Sample Output**

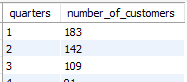


Q9. **Write a query to find the number of customers who entered each Quarter.**

**Print the Quarter and the count of customers who entered in that Quarter.**

**Order your output in ascending order of Quarter.**

**Sample Output**



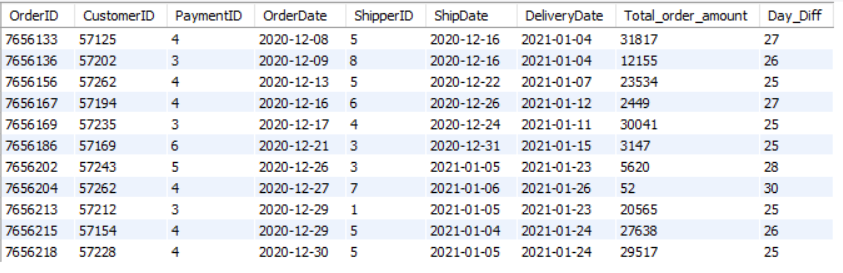
Q10. **Identify the orders which were ordered in 2020 but delivered in 2021.**

**Print all details of such orders along with the difference in days from the order date to the delivery date.**

**Filter to only print details of orders where the difference between the 2 dates in terms of days in greater than or equal to 25.**

**Sort the result in ascending order of OrderID.**

Required Result:

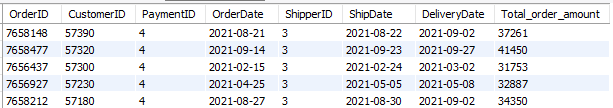


ASSIGNMENT – 5

Q1. **Print all details of Orders which were placed by Customers whose ID is a multiple of 10, payment method by Payment with ID 4, shipped by Shipper with ID 3 and which are greater than 30,000 in order value.**

**Sort the result set in descending order of Customer ID.**

**Sample Output**



Q2. **Print the Customer First Name which comes last according to the alphabetical order.**

**Hint: Apply MAX function on the relevant column.**

Q3. **Get the sum of quantity shipped by each Shipper in each quarter of each year.**

**Print Year, Quarter, ShipperID, Company Name, Quantity Shipped.**

**Order your output in ascending order of the year.**

**For records with same year - sort them in ascending order of quarter, for records with the same quarter - sort them in descending order of Total Quantity.**

**Remember to consider the shipping date when getting the Year and Quarter information.**

**Sample Output:**



Q4. **Each order contains different bunch of products.**

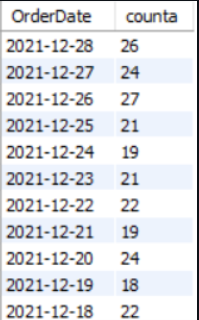
**Each of these products are supplied by different suppliers.**

**Identify the orders for which one of the suppliers is from India.**

**Finally print the count of such orders placed on each individual order date.**

**Print the final output in descending order of orderdate.**

**Sample Output:**

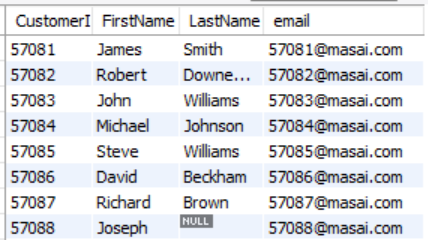
[](https://cp.masaischool.com/assignments/3285/pi/4/solve)

Q5. **Create an email of each customer as**[**customerid@masai.com**](mailto:customerid@masai.com)**. So for customer with ID 500, his ID should be created as '**[**500@masai.com**](mailto:500@masai.com)**'.**

**Print CustomerID, First Name, Last Name and the newly created email address.**

**Sort the result in ascending order of CustomerID.**

Sample Output -



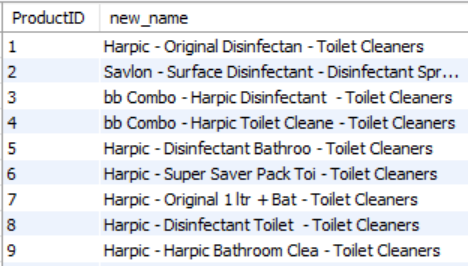
Q6. **Print ProductID and the combination of Brand Name, Product Name and the Type.**

**The Brand Name, the Product Name and the Type should be separated by a space, an hyphen, followed by another space.**

**Make sure to append only the first 20 characters of the Product Name.**

**Sort the result set in ascending order of ProductID.**

**Sample Output** -



Q7. **Count the number of Products with a null value in sub category column.**

Q8. **Calculate the average discount on products of each Brand.**

**Print the Brand name and the corresponding average discount nearest integer value.**

**(e.g. - 76.60 will be 77 and 76.10 will be 76)**

**Only print those brands which have an average discount greater than 50%. Sort the result set in alphabetical order of Brand Name.**

Q9. **Calculate the average discount percentage on products of each brand belonging to the bakeware sub category.**

**Print Brand name and their average percentage discount in nearest integer value.(e.g. - 77.80 will be 78 and 77.10 will be 77)**

**DISCOUNT = ((MARKET\_PRICE - SALE\_PRICE)/MARKET\_PRICE) X 100**

**Sort the result in desceding order of Average discount value.**

Sample Output -



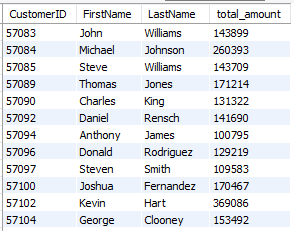
Q10. **Write a query to find out the premium customers who have ordered gross total order amount more than 100000 in the past 24 months.**

**Print Customer ID, First name,Last name and the total amount.**

**Order your output in ascending order of Customer ID.**

**(Hint - For 24 months calculation consider Today's date and time)**

**Sample Output**

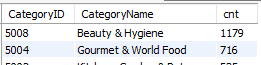


Q11. **Write a query to find out the top 3 selling categories in 2020.**

**Print Category ID, Category name and the number of orders placed.**

**Order your output in descending order of number of order.**

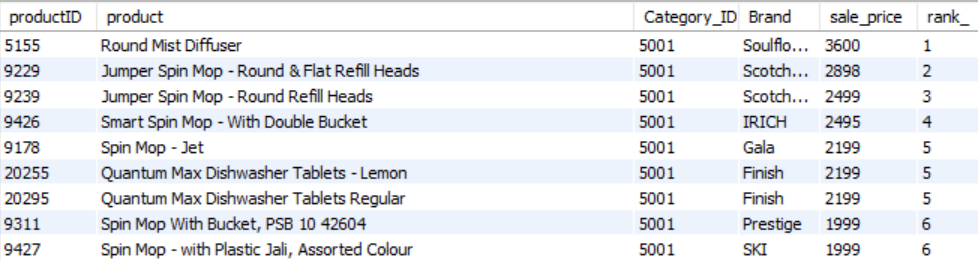
**Sample Output**



Q12. **Write a query to rank the products on the basis of highest selling price within each category.**  
**If two Products has same selling price then ranking took alphabetically their name in consideration.**  
**Prevent skipping of ranks.**

**Print ProductID, Product Name, CategoryID, Brand, Sale Price and Rank.**

Sample Output -



Q13. **Identify the products whose names consist of the word 'Baby'.**

**Then count the number of such products for each Category and Sub Category.**

**Print Category ID, Category Name, Sub Category and followed by the Count.**

**Sort the result in ascending order of Category ID. Secondary sort on the basis of alphabetical order of Sub Category.**

**Sample Output**



Q14. **Print Product Id, Product Name, Difference between Market Price and Sale Price and Discount Status.**

**Discount Status should be:**

**'High discount' -> if the Difference betweeen Market Price and Sale Price greater than 1500.**

**'Medium discount'-> if Difference betweeen Market Price and Sale Price greater than 500 and less than equal to 1500.**

**'Less discount' -> if Difference betweeen Market Price and Sale Price greater than 0 and less than equal to 500.**

**'No Discount' -> if the Difference betweeen Market Price and Sale Price is 0.**

**Sort the output in descending order of the Difference and ascending order of Product Id.**

**Note : Difference = Market Price - Sale Price**

**Sample Output**



Q15. **We want to analyze on which Weekday of Month customers ordered the most, for each month in the year 2021.**

**Print Months name, Weekday name, Number of orders ordered.**

**If a month has two days where same Number of orders were placed consider them all.**

**Sort the output in descending order of Number of orders ordered.**

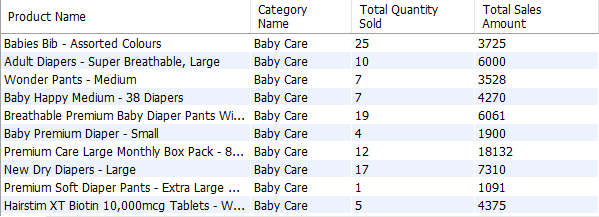
**Note : Weekday name as in Monday,Tuesday ..... and so on.**

Sample Output :



#### Q16. Write a SQL query to list each product's name, category name, total quantity sold, and total sales amount. The output columns should be Product Name, Category Name, Total Quantity Sold, and Total Sales Amount. Order the results first by Category Name in ascending order and then by Total Sales Amount in descending order.

#### Sample Result

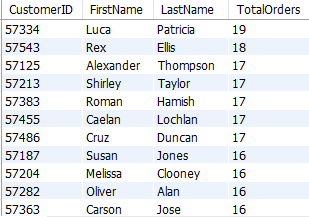
[](https://imgbb.com/)

Q17. Construct a SQL query to compile a list of customers whose number of orders exceeds the average order count across all customers in the database. The campaign management team will use this list to recognize and reward these customers for their loyalty. The output should contain the following information:

* **CustomerID**: The unique identifier for the customer.
* **FirstName**: The first name of the customer.
* **LastName**: The last name of the customer.
* **TotalOrders**: The total number of orders that the customer has placed.

Order the results first by **TotalOrders** in descending order to highlight the customers with the highest number of orders. Apply secondary sorting by **CustomerID** in ascending order to organize customers with the same number of orders.

#### Sample Result

[](https://imgbb.com/)

ASSIGNMENT – 6

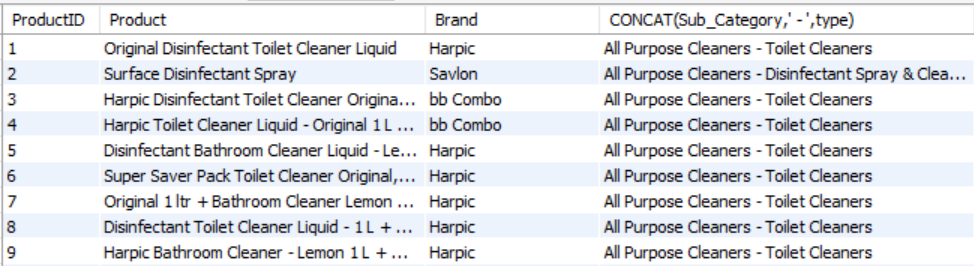
Q1. **Print Product Details along with the corresponding sub category and type as a single entity.**

**Let the Sub Category and Type be separated by a space, a hyphen, followed by another space.**

**Print ProductID, Product Name, Brand Name followed by the newly created entity.**

**Sort the result set in ascending order of ProductID.**

Sample Output -



Q2. **Write a query to identify customer IDs who have ordered only once and find out how many months back the order was placed from now.**

**Use timestampdiff to find the last ordered months ago**

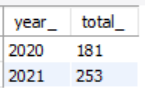
**Print customer ID along with the number of months count.**

**Order your output in ascending order of Customer ID.**

Q3. **Write a query to find the number of customers who entered each year.**

**Print the year and the count of customers who entered in that year. Order by year.**

Sample Output - (Not Actual Numbers)



Q4. **Write a query to find the Total order amount in each year during the festive season.**

**(Consider festive season to be September,October and November months).**

**Print 2x2 matrix of year and total order amount.**

**Order your query based on ascending order of Year.**

Sample Output :

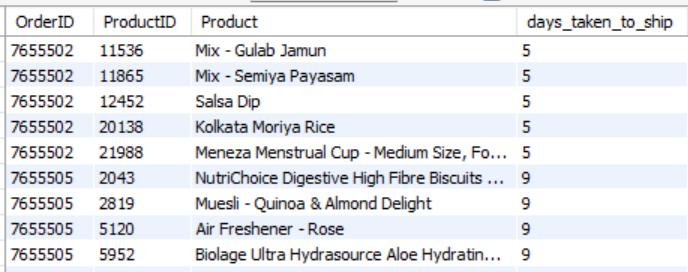
[Screenshot-2023-07-22-162636](https://imgbb.com/)

Q5. **Write a query to find all the Order IDs which took equal to or more than 5 days to ship after the order was placed.**

**Print the OrderID, ProductID,Product Name and days taken to ship.**

**Order your output in ascending order of OrderID, for records with same order ID, sort them in ascending order of ProductId.**

Sample Output -

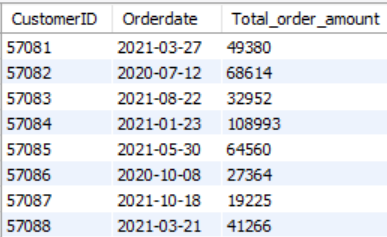


Q6. **Write a query to find the highest order amount and it's corresponding order date for each customer.**

**Print CustomerID,orderdate and total order amount.**

**Sort the result in ascending order of CustomerID.**

Sample Output -



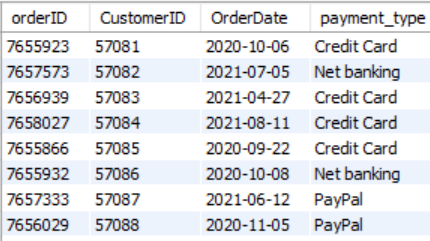
Q7. **Write a query to find the details of all orders placed by customers on their 5th Day of Ordering.**

**Print the OrderId, CustomerId, Order Date and Payment Type.**

**Avoid skipping ranks.**

**Order your output in ascending order of Customerid. For records with same CustomerID, sort them in ascending order of OrderID.**

Sample Output -



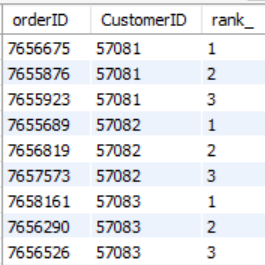
Q8. **Write a query to find the top 3 OrderIds with highest order amount for each customer.**

**Print OrderId, CustomerId and Rank.**

**Avoid skipping of ranks.**

**Order your query in ascending order of Customer Id.**

Sample Output -

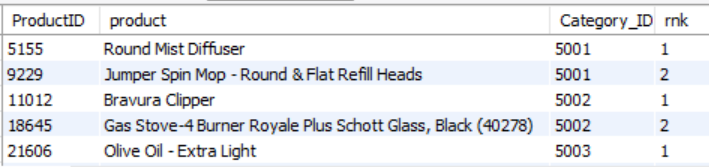


Q9. **Write a query to find the top 2 products for each category with the highest market price.**

**Print ProductId, Product name, CategoryId and Rank. Avoid skipping ranks.**

**Sort the result in ascending order of CategoryId, for records with same CategoryId, sort them in ascending order of Rank, for records with same Rank, sort them in ascending order of ProductId.**

Sample Output -

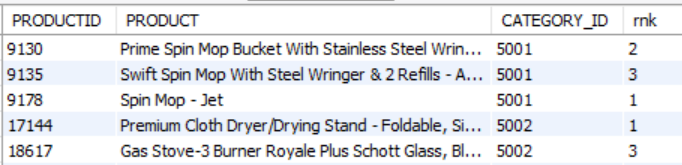


Q10. **Write a query to find the top 3 products with highest discounts (market\_Price - sale\_price) , in terms of money, for each category.**

**Print ProductId, Product Name, CategoryId and Rank. Avoid skipping ranks.**

**Sort the result in ascending order of CategoryId, for records with same CategoryId, sort them in ascending order of ProductId.**

Sample Output -



Assignment – 7

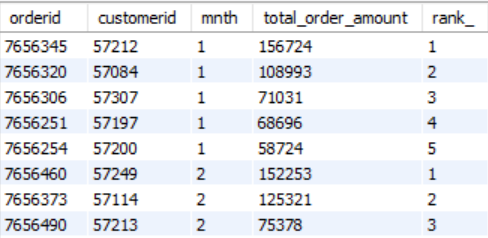
Q1. **Write a query to find the top 5 customers for each month with highest total order amount.**

**Print order id,customer id, month number, order amount and rank.**

**Avoid skipping ranks.**

**Sort the result in ascending order of month number. For records with same month number, sort them in ascending order of rank.**

Sample Output -



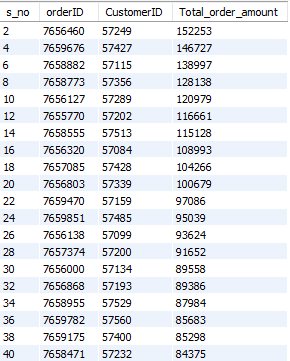
Q2. **Write a query to assign s\_no (or row\_number) to the orders table by ordering total order amount in decreasing order. For orders with same total order amount, sort them in ascending order of OrderID**

**Print only those records where s\_no is even.**

**Your output should contain the following columns: s\_no,order id, customer id, total\_order\_amount.**

**Sort the result in ascending order of s\_no.**

Sample Output -

[](https://cp.masaischool.com/assignments/3292/pi/2/solve)

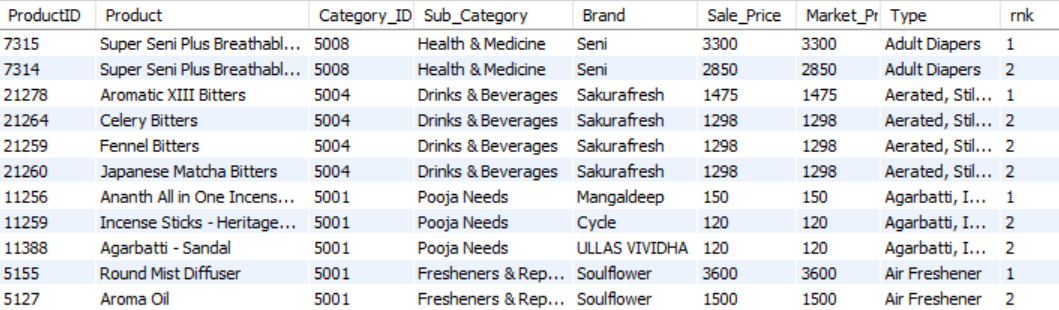
Q3. **Write a query to find the top 2 Sale prices for each type.**

**Print all the columns of products table and the rank.**

**Avoid skipping ranks.**

**Sort the result set in alphabetical order of Type. For records with the same type sort them in ascending order of rank.**

Sample Output -



Q4. **Write a query to find the top 10 customers who placed the maximum number of orders in the year 2021.**

**Print Customerid,first name and the number of orders placed.**

**Sort the result in ascending order of CustomerID.**

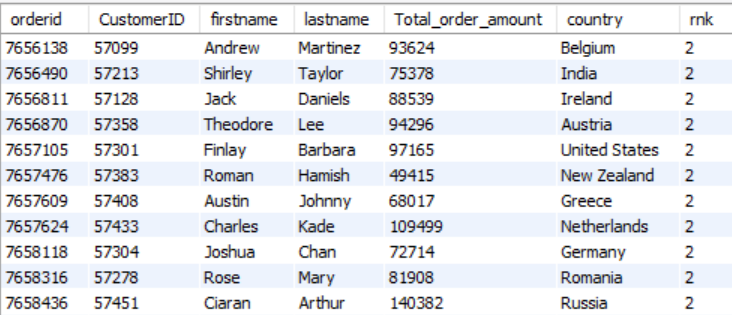
Q5. **Write a query to identify the customer from each country who placed that particular country's second most expensive order.**

**Print orderid, customerid, first name, last name, total order amount, country and rank.**

**Avoid skipping ranks.**

**Sort the result in ascending order of OrderID.**

Sample Output -



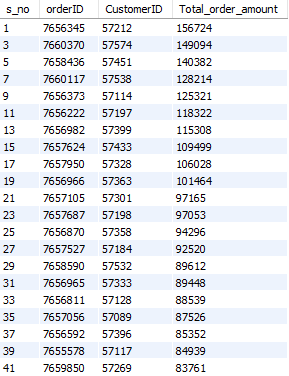
Q6. **Write a query to assign s\_no (or row\_number) to the orders table by ordering total order amount in desc. For records with same order amount, sort them in ascending order of OrderID.**

**Then, print only those records where s\_no is odd.**

**Your output should contain the following columns: s\_no,order id, customer id, total\_order\_amount.**

**Sort the result in inccreaasing order of s\_no.**

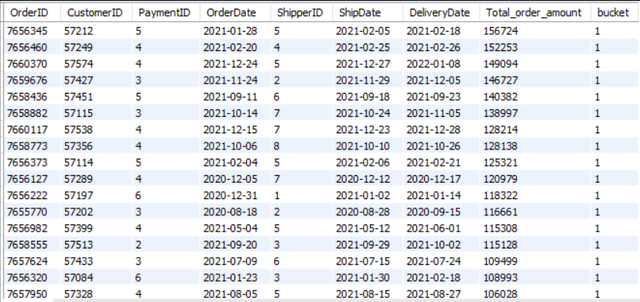
Sample Output -

[](https://cp.masaischool.com/assignments/3292/pi/6/solve)

Q7. **Write a query to divide all the orders into 30 buckets on the basis of decreasing order of total order amount.**

**Print all the records from orders table and the bucket number.**

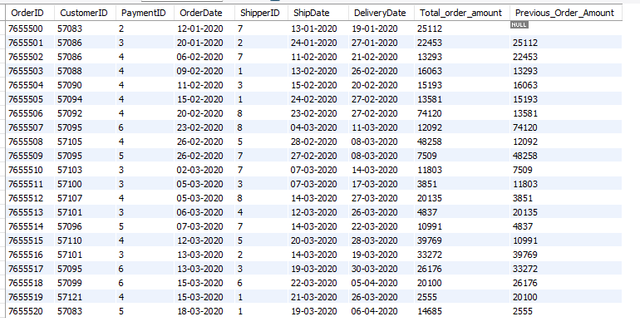
Sample Output:

[](https://cp.masaischool.com/assignments/3292/pi/7/solve)

Q8. **Write a query to find the total order amount of the previous OrderID.**

**Print all the details from orders table and the prevous order amount.**

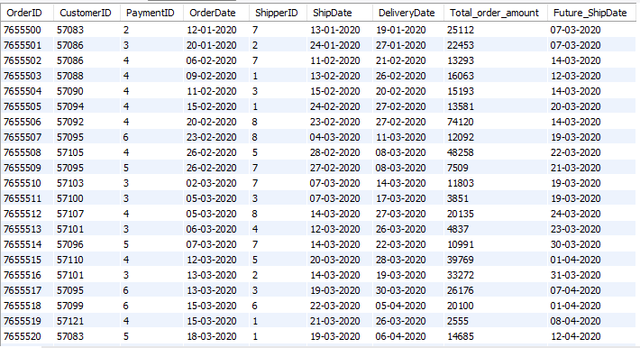
Sample Output -

[](https://cp.masaischool.com/assignments/3292/pi/8/solve)

Q9. **Write a query to find the shipdate of the 10th following OrderID.**

**Print all the details from orders table and the 10th following shipdate.**

Sample Output -

[](https://cp.masaischool.com/assignments/3292/pi/9/solve)

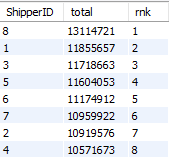
Q10. **Write a query to find the sum of total order amount for each shipper id and rank them basis the decreasing order of this sum.**

**Print shipper id, total order amount and rank.**

**Avoid skipping ranks.**

**Sort the result in ascending order of rank.**

Sample Output -

[](https://imgbb.com/)