02/05/2024

Shilpa Amirishetti

cohort-9

SQL Assignment

Part 1 & 2

A blue circle with white text and a dolphin

Description automatically generated

Table of Contents

[**Assignment Part -1** 2](#_Toc165574861)

[Query 1 2](#_Toc165574862)

[Query 2 SELECT 2](#_Toc165574863)

[Task 1 3](#_Toc165574864)

[Query1 3](#_Toc165574865)

[Query2 3](#_Toc165574866)

[Task 2 4](#_Toc165574867)

[Task 3 4](#_Toc165574868)

[Task 4 5](#_Toc165574869)

[Task 5 5](#_Toc165574870)

[Task 6 6](#_Toc165574871)

[Creating an EER Diagram for sql\_store 6](#_Toc165574872)

[**Assignment Part-2** 8](#_Toc165574873)

[Task 1 8](#_Toc165574874)

[Task 2 8](#_Toc165574875)

[Task 3 9](#_Toc165574876)

[Task 4 9](#_Toc165574877)

[Task 5 10](#_Toc165574878)

[Task 6 10](#_Toc165574879)

[Task 7 11](#_Toc165574880)

[Task 8 12](#_Toc165574881)

[Task 9 12](#_Toc165574882)

[Task 10 13](#_Toc165574883)

[Task 11 13](#_Toc165574884)

[EER Diagram for world Database 14](#_Toc165574885)

# **Assignment Part -1**

## Query 1

**Input the following into the file Query**

**USE sql\_store;**

**SELECT \***

**FROM customers**

**WHERE CUSTOMER\_ID=1**

**order by first\_name**

A screenshot of a computer

Description automatically generated

Returns all the columns from customers table where CUSTOMER\_ID=1 and sorts the result based on column first\_name.

## Query 2 SELECT

**SELECT last\_name, first\_name, points, points + 10**

**FROM CUSTOMERS**

A screenshot of a computer

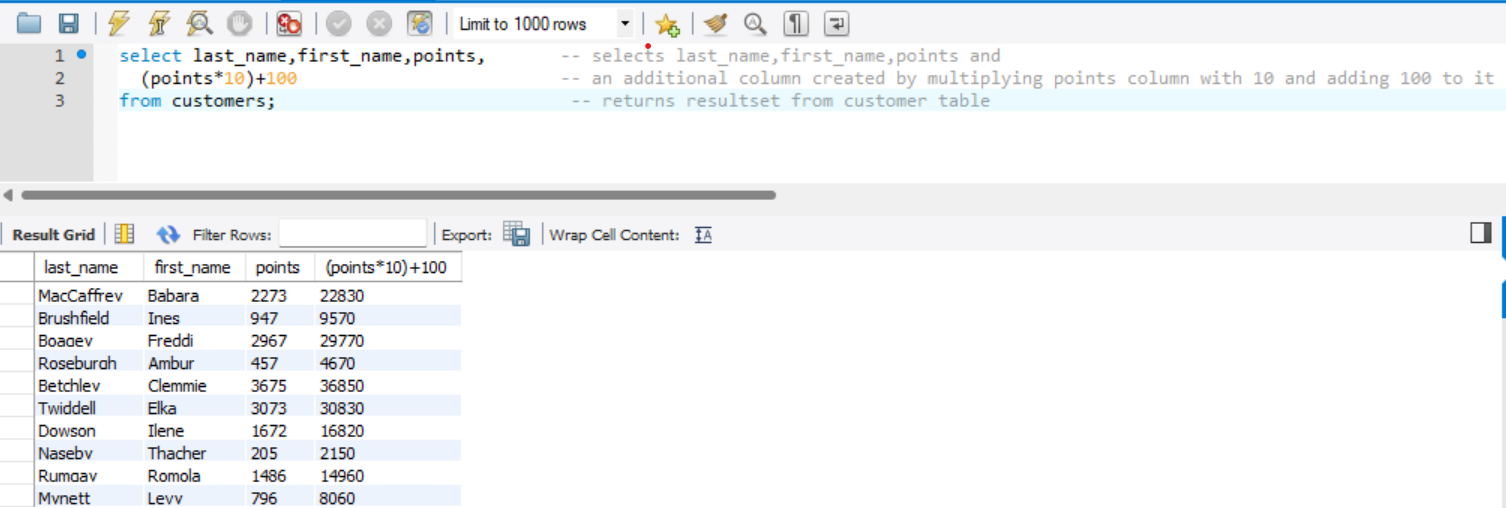
Description automatically generated

Returns columns **last\_name ,first\_name, points** and an additional column **points+10** (which is extracted by adding 10 to the points column) from Customers table.

## Task 1

### Query1

**Using the Query 2 you created change the points to reads times by 10 and plus 100. Record your results in your word document**



Returns columns **last\_name, first\_name, points** and an additional column **(points\*10)+100** which is obtained by multiplying by 10 and adding 10 to the points column from Customers table.

### Query2

**Change the Query 2 code to create a discount factor so the table now shows a discount header and changing the (point + 10) \*100**

A screenshot of a computer

Description automatically generated

Returns columns **last\_name, first\_name, points** and an creates additional column **discount\_factor** using alias which is obtained by adding 10 to the points column and multiplying it by 100 from Customers table.

## Task 2

**Write a SQL query to return all the products in our database in the result set. I want you to show columns; name, unit price, and new column called new price which is based on this expression, (unit price \* 1.1 ).**

**So what you are doing is increasing the product price of each by 10%.**

**So with the query we want all the products the original price and the new price.**

A screenshot of a computer

Description automatically generated

It returns **name,unit price** columns along with an additional column **new\_price** which is created by increasing each product unit price by 10% i.e., **new\_price**=**unit\_price**+((**unit\_price**\*10/100) from Products table.

## Task 3

**In this task create a new query to find all the customers with a birth date of >**

**'1990-01-01'**

A screenshot of a computer

Description automatically generated

Returns all the columns from customer table and filters the records based on the where condition**.** The result set contains the records of customers whose birthdate is greater than **'1990-01-01'**

## Task 4

**Select sql\_inventory.**

**Write a query to find out the name of the product with most amount in stock.**

A screenshot of a computer

Description automatically generated

To find the most amount in stock first we need to calculate the product of quantity\_in\_stock with unit\_price to get the amount of stock pertaining to each product. We are creating an alias for the product calculation as most\_amt\_in\_stock. Then we wite a subquery to find the maximum value of unit\_price as it gives the product name with maximum value and assigns it to unit\_price. The result set displays name, product\_id and most\_amt\_in\_stock columns showing the product name with most amount in stock.

## Task 5

**Select sql\_inventory.**

**Write a query to find out the name of the most expensive product.**

A screenshot of a computer

Description automatically generated

In this query we are using Max() function to find the most expensive product as MAX() function returns the maximum value from that column. Here we are selecting product\_id, name and max(quantity\_in\_stock) to be displayed from Products table and grouping the results based on product\_id. As we want only the most expensive product we are restricting the number of rows to be displayed to 1 using LIMIT 1.

## Task 6

**Select sql\_store.**

**Write a query to find out the first name, last name, address and the birthdate of the oldest customer.**

A screenshot of a computer

Description automatically generated

To find the oldest customer from Customers table we use MIN() function. The MIN() function in MYSQL is used to return the smallest value from a set of values. In this query we are selecting colums first\_name, last\_name, address and birth\_date from Customers table and writing a subquery on birth\_date using MIN(birth\_date) to find the customer with the oldest date of birth. The result set displays the customer name with the oldest birth date.

## Creating an EER Diagram for sql\_store

**Write about the relationship between tables. Primary and Foreign keys. (Few words about tables)**

A screenshot of a computer

Description automatically generated

In this EER diagram we have six different tables named Products, Customers and Orders, Order\_items, Shippers, Order\_statuses which are inter related to each other through some kind of relationship.

In the Customers table we have Customer\_id which acts a Primary key for Customers table.It has one-to-one mandatory relationship, where exactly one instance of an entity must be associated with exactly one instance of another entity associated with the Orders table.

In the Orders table the order\_id acts as the Primary Key Constraint. It has three foreign keys constraints customer\_id which is the primary key for Customers table, Status which acts as primary key for Order\_statuses table and Shipper\_id which is primary key for Shippers table.

The Orders table has one to many relationship with Customers table, Order\_statuses table and Shippers table.

In the Products table product\_id acts as the primary key and it has mandatory one-one relationship with the Order\_items table.

In the Order\_items a combination of order\_id and product\_id acts as a Primary Key and the Product\_id as a foreign Key.It has one to many relationship with Orders table and Products table.

In the Shippers table the Shipper\_id is the primary key and there is no foreign key associated to it.It has one-one mandatory relationship with Orders table.

In the Order\_statuses the order\_status\_id acts as primary key constraint and it has one to one mandatory relationship with Oders table.

# **Assignment Part-2**

## Task 1

**Using count, get the number of cities in the USA**

A screenshot of a computer

Description automatically generated

The COUNT() function is used to count the number of rows in a result set or the occurrences of a particular value in a column. We are using Count(name) and providing an alias no\_of\_cities to find different city names based on the Where condition countrycode=”USA” .It returns the count of total number of cities whose country code is USA.

## Task 2

**Find out what the population and average life expectancy for people in Argentina (ARG) is.**

A screenshot of a computer

Description automatically generated

To find the total and average we use aggregate functions sum() and avg() in MySQL. We are selecting population field and applying these aggregate functions and providing aliases as ARG\_population for total population and life\_expectancy for average life expectancy for people of Argentina.we are filtering records based on the where condition countrycode=”ARG”. The result set displays the sum of population of Argentina and the average life expectancy of Argentina.

## Task 3

**Using ORDER BY, LIMIT, what country has the highest life expectancy**

A screenshot of a computer

Description automatically generated

To find the country with highest life expectancy we are using ORDER BY clause on population field and sorting the results in decreasing order using DESC. It displays the results in descending order of population starting with the highest value first. Since we want only one row which has the highest life expectancy we are limiting the rows to 1 by using LIMIT clause which is the required result.

## Task 4

**Select 25 cities around the world that start with the letter 'F' in a single SQL query.**

A screenshot of a computer

Description automatically generated

We use LIKE operator in where clause to search for a specified pattern in a column, Here

we are using LIKE ‘F%’ to find name of the cities that start with letter ‘F’ using the where

condition. The result set displays the list of all rows whose city name starts with ‘F’ and we

are restricting the number of rows to be displayed to 25 by using the Limit clause.

## Task 5

**Create a SQL statement to display columns Id, Name, Population from the city table and limit results to first 10 rows only.**

A screenshot of a computer

Description automatically generated

LIMIT clause in MySQL is used to restrict the number of rows to be displayed in the result set. In this query we are selecting ID, Name and population fields to be displayed from the City table and restricting the number of rows to be displayed to 10 by using the Limit operator.

## Task 6

**Create a SQL statement to find only those cities from city table whose population is larger than 2000000**

A screenshot of a computer

Description automatically generated

The greater than operator is used in MySQL to compare two values. In this query we are selecting name and population columns from city table and using greater than operator to find the value of population greater than 2000000 with the help of where clause. The result will display only those rows whose population is greater than 2000000.

## Task 7

**Create a SQL statement to find all city names from city table whose name begins with “Be” prefix.**

A screenshot of a computer

Description automatically generated

We use LIKE operator in where clause to search for a specified pattern in a column, Here

we are using LIKE ‘Be%’ to find names of the city that start with letter ‘Be’ using the where

condition. The result set displays the list of all the rows whose city name starts with ‘Be’.

Task 8

**Create a SQL statement to find only those cities from city table whose population is**

**between 500000-1000000.**

A screenshot of a computer

Description automatically generated

To find the results within a varied range of values we use between operator in MySQL. In this query we are selecting name and population fields to be displayed and the between operator looks for the specified range of values, i.e from 500000 to 1000000. and filters the records using where condition.

## Task 9

**Create a SQL statement to find a city with the lowest population in the city table**

A screenshot of a computer

Description automatically generated

To find the lowest population in the city table we are using the aggregate function min() on population field. The result set contains list of all column with only one row which contains the minimum population in the City table.

## Task 10

**Create a SQL statement to display all cities from the '*city'* table sorted by *Name*in ascending order.**

A screenshot of a computer

Description automatically generated

The sql statement returns all the columns from City table and sorts the results in the ascending order of name field.

## Task 11

**Create a SQL statement to find a country with the largest population in the country table.**

A screenshot of a computer

Description automatically generated

It returns name, population columns fields from country table and write a subquery to find the largest population in the country by using aggregate function max() on population. This returns the name of the country with highest population and displays it in the resultset.

## EER Diagram for world Database

A screenshot of a computer

Description automatically generated

* **Identify the primary key in country table.**

Code acts as the primary key constraint in Country table.

* **Identify the primary key in city table.**

ID acts as the primary key in city table

* **Identify the primary key in countrylanguage table.**

Language acts as the primary key constraint in countrylanguage table.

* **Identify the foreign key in city table.**

Countrycode is the foreign key constraint in city table

* **Identify the foreign key in countrylanguage table.**

Countrycode acts as the foreign key constraint in countrylanguage table.