9/25/2023

Nisanth Mahalingam

JUSTIT DATA TECHNICIAN TRAINEE

Assignment 2

SQL

Contents

[Task 1 0](#_Toc147346107)

[Part 1 0](#_Toc147346108)

[Part 2 0](#_Toc147346109)

[Task 2 1](#_Toc147346110)

[Task 3 2](#_Toc147346111)

[Task 4 2](#_Toc147346112)

[Task 5 2](#_Toc147346113)

[Task 6 3](#_Toc147346114)

[Task 7 3](#_Toc147346115)

[Task 8 3](#_Toc147346116)

[Task 9 4](#_Toc147346117)

[Challenge 1 4](#_Toc147346118)

[Challenge 2 4](#_Toc147346119)

[Challenge 3 5](#_Toc147346120)

[Creating an Enhanced Entity-Relationship Diagram 6](#_Toc147346121)

[Assignment Part 2 – World db 7](#_Toc147346122)

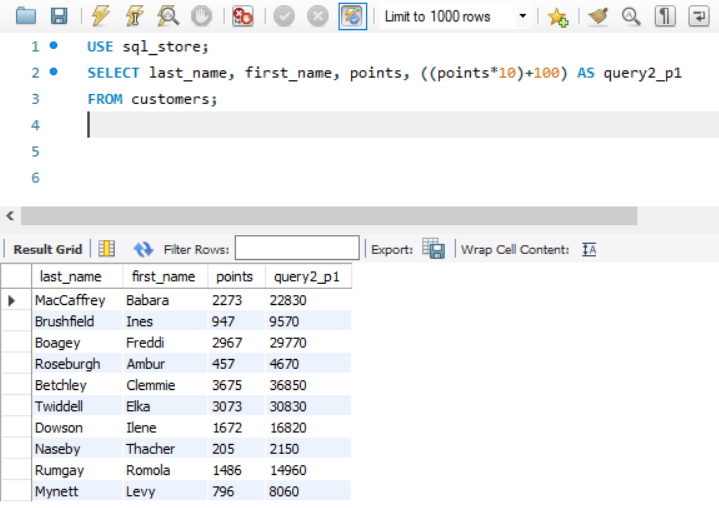
[Full MySQL Script 8](#_Toc147346123)

[Reflection 10](#_Toc147346124)

# Task 1

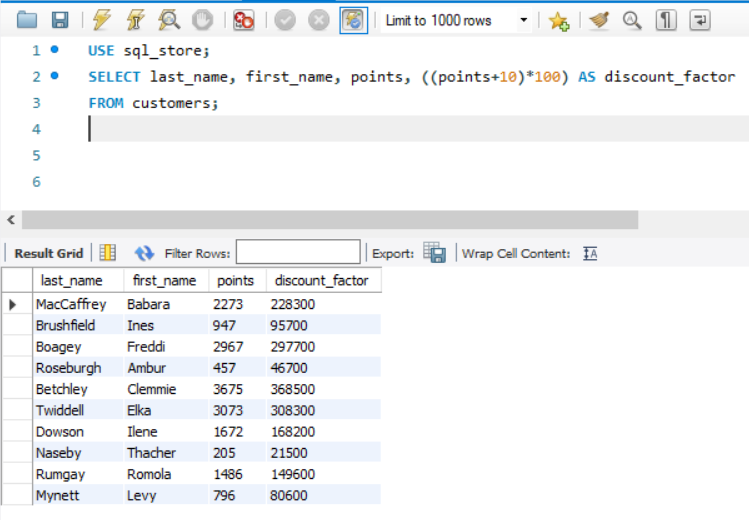
## Part 1

Using the Query 2 you created change the points to reads times by 10 and plus 100.



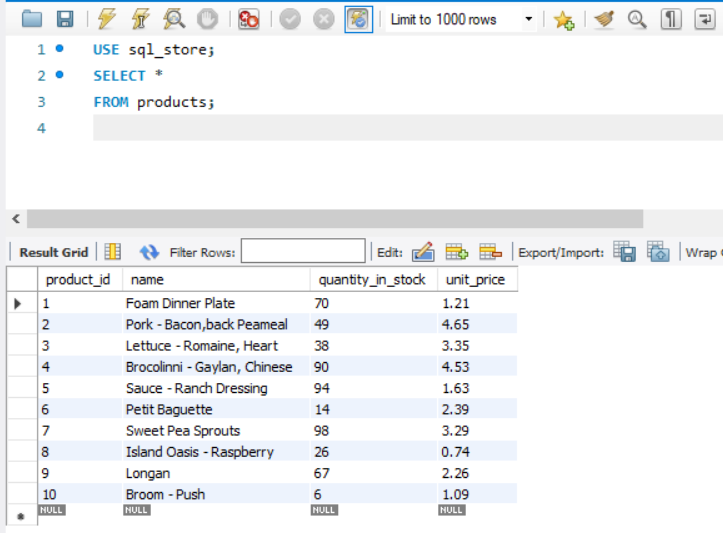
## Part 2

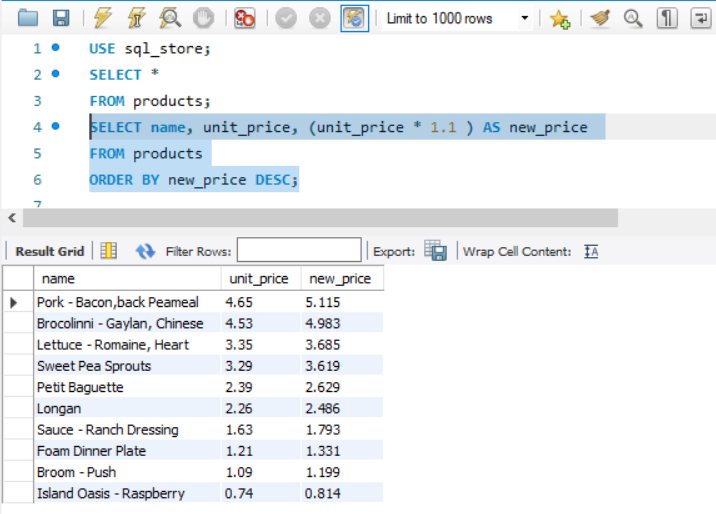
Create a discount factor so the table now shows a discount header and changing the (point + 10) \*100



# Task 2

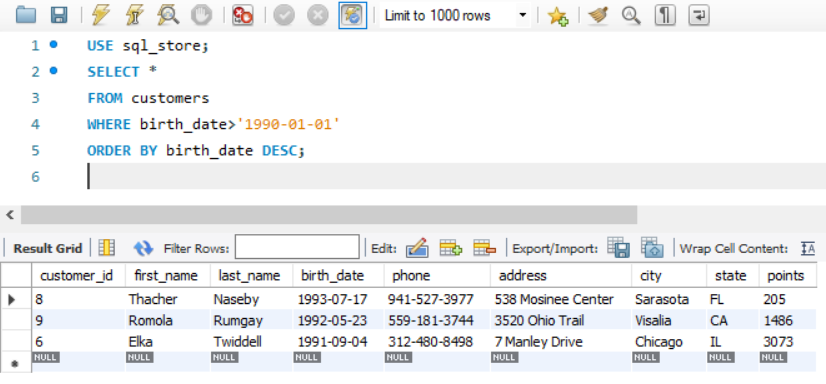
Return all the products in our database in the result set. I want you to show columns; name, unit price, and new column called new price which is based on increasing the product price of each by 10%.





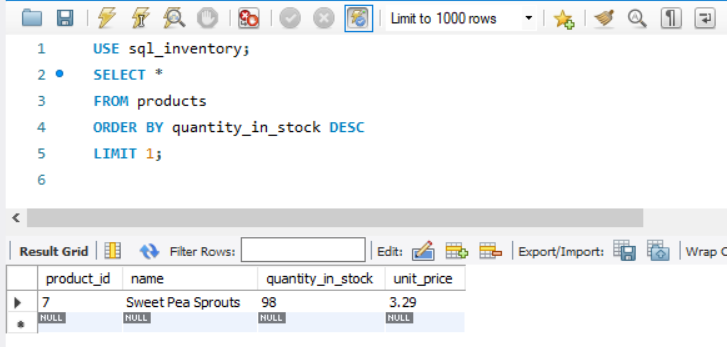
# Task 3

Find all the customers with a birth date of > '1990-01-01'



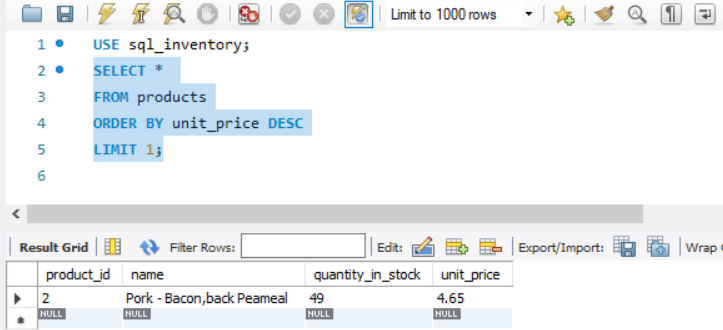
# Task 4

Select sql\_inventory and write a query to find out the name of the product with most amount in stock.



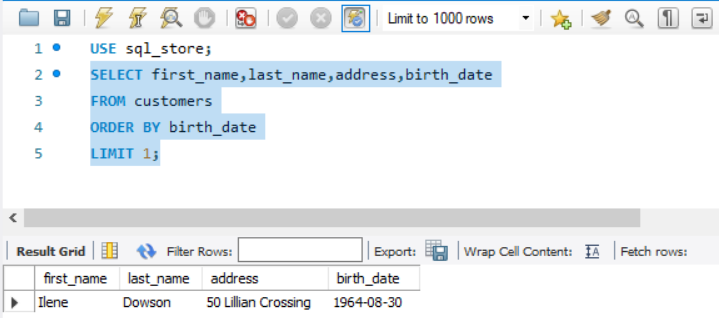
# Task 5

Select sql\_inventory and write a query to find out the name of the most expensive product.

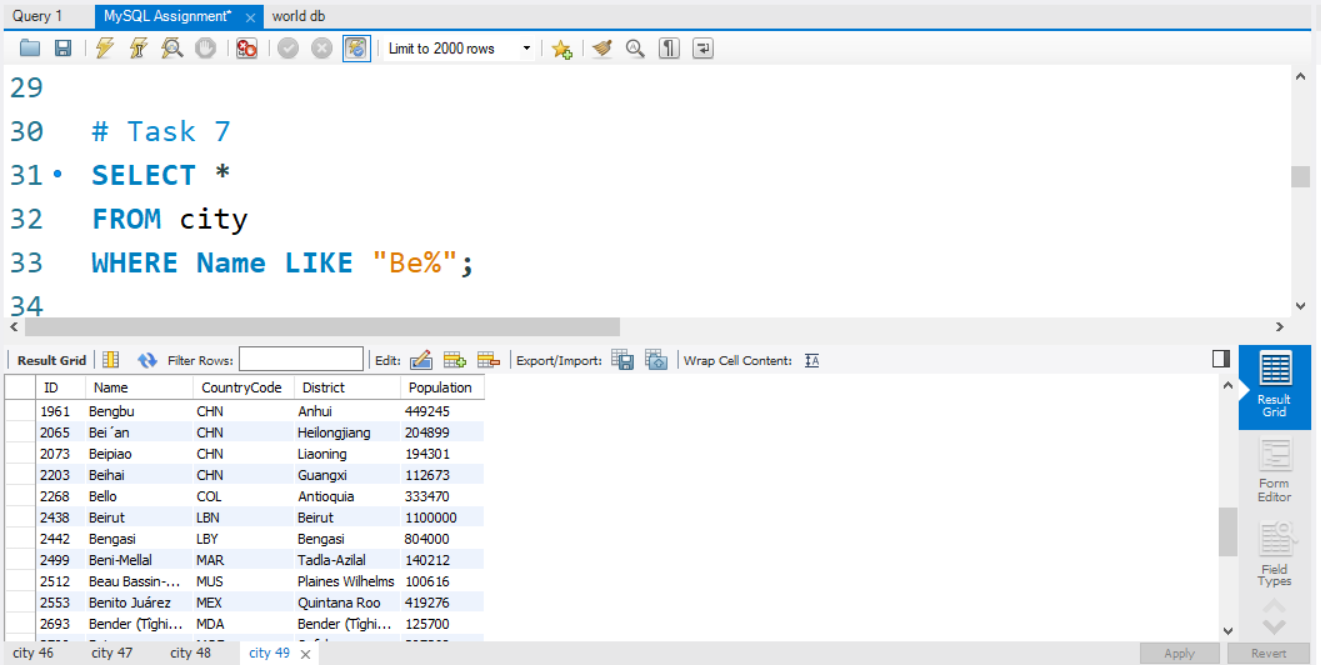


# Task 6

Select sql\_store and write a query to find out the first name, last name, address and the birthdate of the oldest customer.

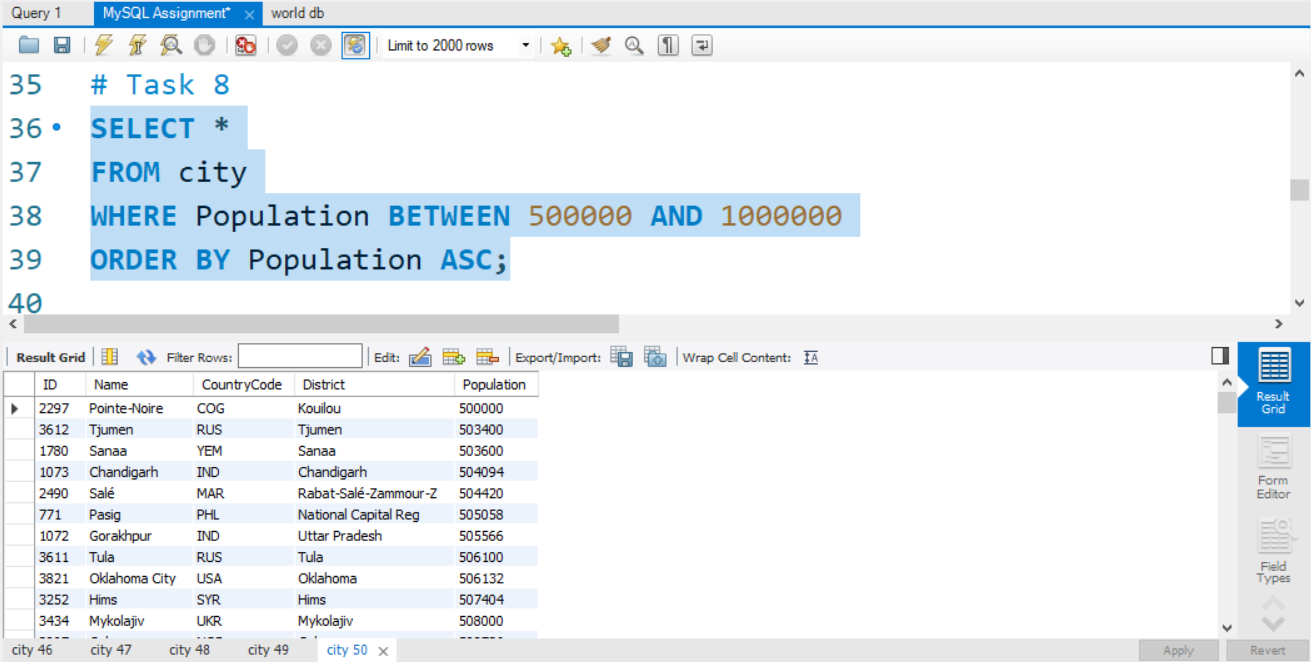


# Task 7

Select cities that start with “Be”.

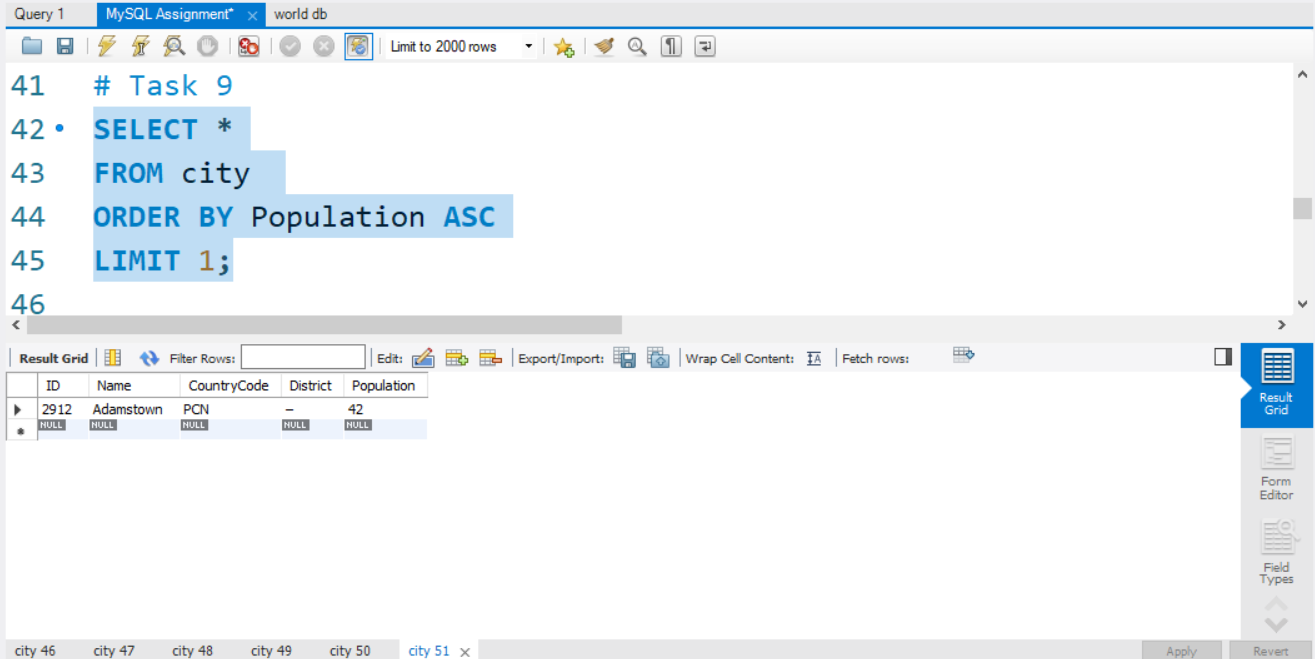
# Task 8

Select the cities where population is between 500,000 and 1,000,000.

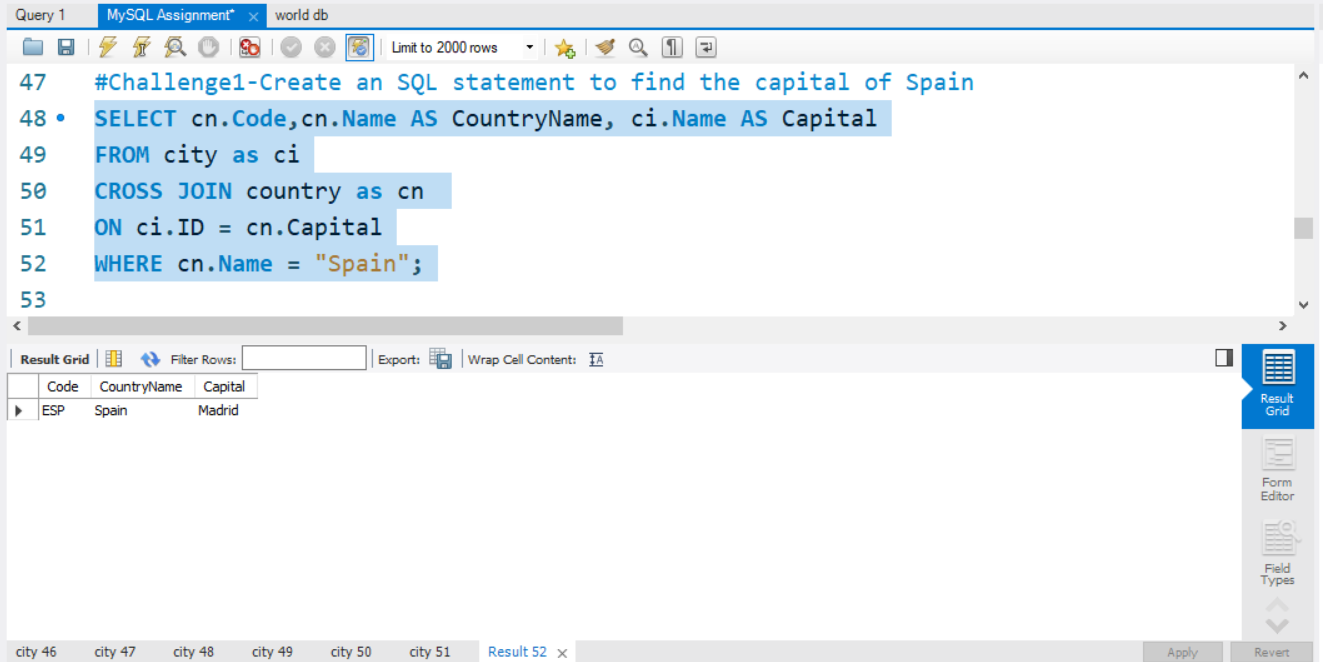


# Task 9

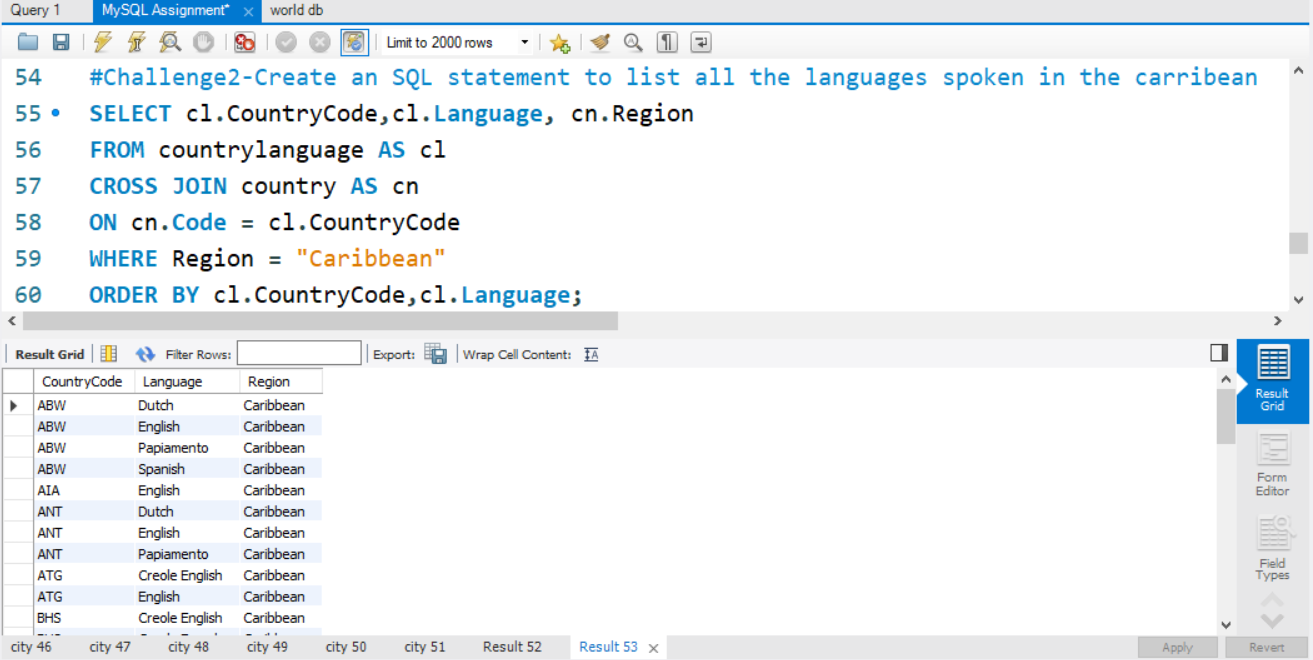
Select the city with the lowest population



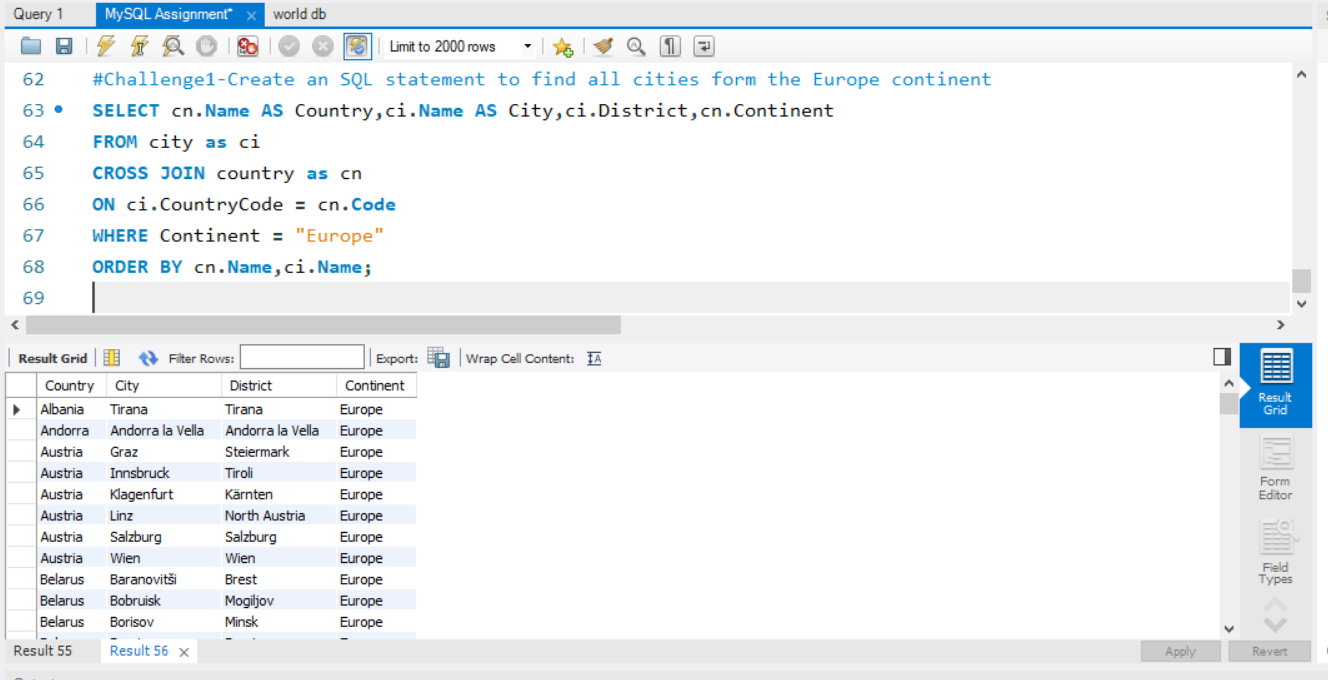
# Challenge 1



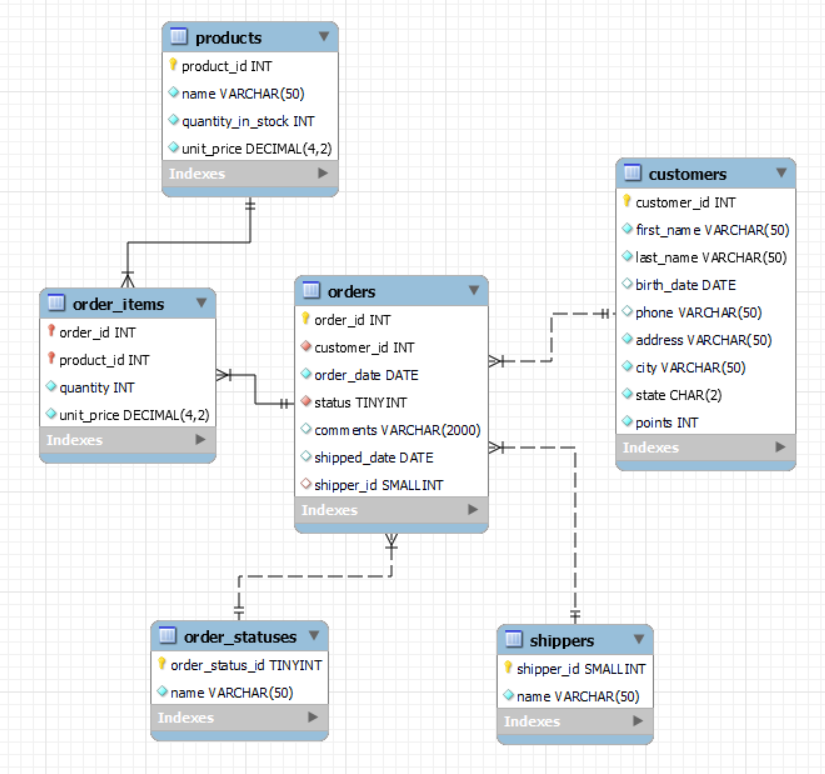
# Challenge 2



# Challenge 3

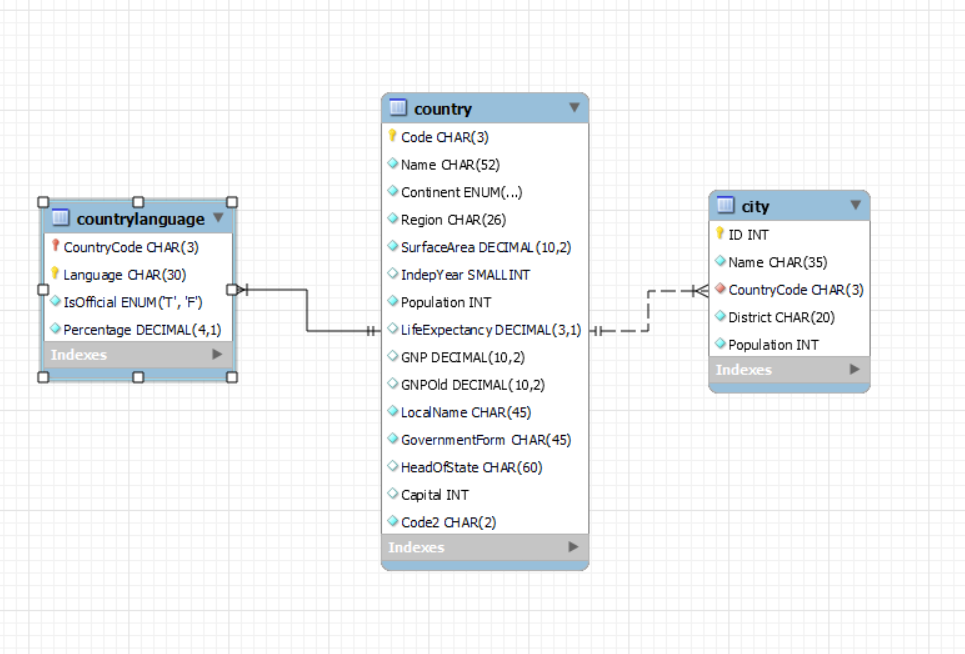


# Creating an Enhanced Entity-Relationship Diagram



A snowflake schema has been adopted for sql\_store’s schema with the orders table being the central fact table. This schema wouldn’t qualify for a star schema as the products table doesn’t have a direct relationship with the orders table developing a hierarchical structure. The primary key for the orders table is order\_id and it consists of foreign keys. The order\_items table deosn’t necessarily have a specific primary key but is made up of composite keys that give rows a unique identifier depending on the different combination of columns, which in this case is products and orders.

# Assignment Part 2 – World db



1. Primary Key in country table is Code CHAR(3)
2. Primary Key in city table is ID INT
3. Primary Key in countrylanguage table is Language CHAR(30)
4. Foreign Key in city table is CountryCode CHAR(3)
5. Foreign Key in countrylanguage table is CountryCode CHAR(3)

# Full MySQL Script

**#Task 1 – Get the number of cities in the USA**

USE world;

SELECT COUNT(DISTINCT(Name))

FROM city

WHERE CountryCode = 'USA';

**#Task 2 - Find out what the population and life expectancy for people in Argentina (ARG) is**

SELECT Code,Name, Population,LifeExpectancy

FROM country

WHERE Name="Argentina";

**#Task 3 - What country has the highest life expectancy?**

SELECT \*

FROM country

ORDER BY LifeExpectancy DESC

LIMIT 1;

**# Task 4 - Select 25 cities around the world that start with the letter 'F' in a single SQL query**

SELECT \*

FROM city

WHERE Name LIKE "F%"

LIMIT 25;

**# Task 5 - Display columns Id, Name, Population from the city table and limit results to first 10 rows only.**

SELECT ID, Name, Population

FROM city

LIMIT 10;

**# Task 6 - Find only those cities from city table whose population is larger than 2000000.**

SELECT \*

FROM city

WHERE Population>2000000

ORDER BY Population;

**# Task 7 - Find all city names from city table whose name begins with “Be” prefix**

SELECT \*

FROM city

WHERE Name LIKE "Be%";

**# Task 8 - Find only those cities from city table whose population is between 500000-1000000.**

SELECT \*

FROM city

WHERE Population BETWEEN 500000 AND 1000000

ORDER BY Population ASC;

**# Task 9 - Find a city with the lowest population in the city table.**

SELECT \*

FROM city

ORDER BY Population ASC

LIMIT 1;

**#Challenge1-Create an SQL statement to find the capital of Spain**

SELECT cn.Code,cn.Name AS CountryName, ci.Name AS Capital

FROM city as ci

CROSS JOIN country as cn

ON ci.ID = cn.Capital

WHERE cn.Name = "Spain";

**#Challenge2-Create an SQL statement to list all the languages spoken in the Caribbean**

SELECT cl.CountryCode,cl.Language, cn.Region

FROM countrylanguage AS cl

CROSS JOIN country AS cn

ON cn.Code = cl.CountryCode

WHERE Region = "Caribbean"

ORDER BY cl.CountryCode,cl.Language;

**#Challenge3-Create an SQL statement to find all cities form the Europe continent**

SELECT cn.Name as Country,ci.Name as City,ci.District,cn.Continent

FROM city as ci

CROSS JOIN country as cn

ON ci.CountryCode = cn.Code

WHERE Continent = "Europe"

ORDER BY cn.Name,ci.Name;

# Reflection

I really enjoyed doing this assignment as it allowed me to both explore the MySQL but also how to make ERR Diagrams and understand their concepts as well as strengthening my understanding of database schema. This exercise helped me realise that it is important to identify the type of SQL used on the database system being used.