

Database Management Systems

(COP 5725)

Fall 2021

Instructor: Dr. Markus Schneider

TA: Kyuseo Park

Homework 3

Name:	
UFID:	
Email Address:	

Pledge (Must be signed according to UF Honor Code)

On my honor, I have neither given nor received unauthorized aid in doing this assignment.

Signature

For scoring use only:

	Maximum	Received
Exercise 1	85	
Exercise 2	15	
Total	100	

Exercise 1 (SQL Queries) [85 points]

We are given a geostatistical database about countries, continents, rivers, etc. The following information is available in Canvas together with this homework assignment for download:

- An ER diagram of the geostatistical database in PDF format (*HW3Ex1-geostatistical-database-ER-diagram.pdf*).
- An informal description of the database schema in PDF format (*HW3Ex1-geostatistical-database-schema-explanation.pdf*).
- A text file that contains *create table* commands to create the database schema (*HW3Ex1-geostatistical-database-schema.sql*).
- A text file that contains *insert* commands for about 47,800 tuples to fill the database tables (*HW3Ex1-geostatistical-database-input-data.sql*).
- A text file that contains *drop table* commands to delete the database schema and the data in the database (*HW3Ex1-geostatistical-database-drop-tables.sql*).

In a first step, use the CISE Oracle DBMS and the Oracle SQL Developer software to create the database schema and fill the database with data. This will also help you learn about the system environment for your group project. In particular, the use of MySQL, PostgreSQL, and other database systems is not allowed.

In a second step, look at the database schema in the file *HW3Ex1-geostatistical-database-schema.sql*. From lines 38 to 52 you will find the following lines:

```
ALTER TABLE Country
  ADD CONSTRAINT FK_CountryREFCity
  FOREIGN KEY (Code, Capital, Province)
  REFERENCES City(Country, Name, Province)
  INITIALLY DEFERRED DEFERRABLE;

ALTER TABLE City
  ADD CONSTRAINT FK_CityREFProvince
  FOREIGN KEY (Country, Province)
  REFERENCES Province(Country, Name)
  INITIALLY DEFERRED DEFERRABLE;

ALTER TABLE Province
  ADD CONSTRAINT FK_ProvinceREFCountry
  FOREIGN KEY (Country)
  REFERENCES Country(Code)
  INITIALLY DEFERRED DEFERRABLE;

ALTER TABLE Province
  ADD CONSTRAINT FK_ProvinceREFCity
  FOREIGN KEY (Capital, Country, CapProv)
  REFERENCES City(Name, Country, Province)
  INITIALLY DEFERRED DEFERRABLE;
```

Your task is to explore this scenario by using the Internet. The keywords `INITIALLY DEFERRED DEFERRABLE` are non-standard SQL. They are supported by several database systems such as Oracle and PostgreSQL. Answer the following questions:

1. [4 points] What is the meaning of these keywords?
2. [6 points] Why is the action indicated by the keyword `INITIALLY DEFERRED DEFERRABLE` needed in the scenario above? What is the problem? How is the problem solved?

In a third step, write SQL queries for the colloquial queries below and **show the results by providing screenshots for both your SQL queries and query results**. The screenshots must be embedded into the PDF file that contains your solutions to this whole assignment. In order to increase readability, the SQL queries should be written in a structured manner, all SQL keywords should be fully capitalized, and the table and attribute names should be written in the same way as in the schema file.

1. [1 point] Find the names of countries where agriculture takes more than 50% of its gross domestic product (GPD).
2. [3 points] List the top five countries that will have the largest population after five years. [Assume that the population in five years is equal to the population this year * $(1 + \text{growth rate})^5$. The population growth in the database schema is in percentage and should be divided by 100. Use the new attributes *Country*, *Population after 5 years*, and *Rank* for the resulting table schema.
3. [4 points] Find the country *c1* that *used to* have the maximum number *n1* of countries/areas depending on it. Further, find the country *c2* that *now* has the maximum number *n2* of countries/areas depending on it. Output *c1*, *n1*, *c2*, *n2*, and the difference between *n1* and *n2*.
4. [4 points] List the country names that have more than 4 different kinds of religion and at least one religion takes more than 80%.
5. [3 points] Compute the total length of the border that China shares with its neighboring countries.
6. [4 points] Find the top five popular religions and the numbers of their believers in the world.
7. [3 points] Find the names of the lakes in the United States with an elevation that is above the average elevation of all lakes world-wide.
8. [4 points] Find the largest population density (population/area) of provinces that have mountains of the “volcano” type. Output the province name, mountain name, and the population density.
9. [3 points] Find the provinces that are located on more than 2 islands and whose country’s GDP is greater than 1000000.
10. [3 points] Find the two longest rivers that flow through at least one lake and that finally flow into the Atlantic Ocean. Output the name and the length of the rivers.
11. [4 points] Determine the names of countries that have more than three rivers and that have lakes next to more than three provinces.
12. [4 points] Find the names of those countries that are bounded by the largest lake.

13. [2 points] Find the height of the highest mountain for each continent.
14. [3 points] Find the countries whose depth of the deepest sea is less than the elevation of the highest mountain. Display the country name, depth of its deepest sea, and the elevation of the highest mountain.
15. [4 points] Find the northernmost cities of each continent (except Asia). Display the names of these cities and their continent. List cities that are northern of other cities in the result table first.
16. [1 point] Find all countries whose capitals have positive latitudes and less than 10000 inhabitants.
17. [4 points] Find what is larger. Is it the sum of the areas of the 10 largest countries (attribute *top10*) or the sum of the areas of the remaining countries (attribute *rest_world*)? What is their difference (attribute *difference*)? Display the values for the attributes *top10*, *rest_world*, and *difference*.
18. [2 points] Find all countries that cross continental boundaries.
19. [2 points] Display each island in Africa and its area if the area is larger than 1000 square kilometers. The output should be in descending order of the size of the areas.
20. [3 points] List the names and GDPs of those countries that are members of the NATO and more than 5 percent of their population are Muslims.
21. [1 point] Find names of rivers which cross at least 12 provinces in the same country.
22. [2 points] Find the name and length of the longest river on the American continent.
23. [3 points] Find the provinces that have the largest number of islands in the world. Output the country code, the province, and the number of islands.
24. [3 points] List the 10 country names (attribute "Country Name") with the highest population density (attribute "Population Density") as well as the percentage of the world population (attribute "Percentage") each one contains.
25. [5 points] List the names of organizations that have only Asian countries as members.

Exercise 2 (QBE) [15 points]

Consider the following table schemas (primary keys are underlined):

Company (cID, name, location)

Flight (cID, fnumber, departure, arrival, price:integer, numberOfSeats:integer)

Book (bID, cID, fnumber, customerID, seat, date)

Customer (customerID, name, address)

Draw Query-By-Example (QBE) tables for the following queries.

1. [3 points] Find the names of customers who have booked flights at every company in the US.
2. [2 points] Find the names and addresses of customers who never booked a flight.
3. [2 points] Find the names of customers who booked the flight to New York more than once.
4. [3 points] Find the names of the companies that have the biggest (in terms of the number of seats) airplane.
5. [2 points] Find the names of customers who booked flights to Boston and Seattle.
6. [3 points] Insert tuples into a new table CustomerCheck that stores the names and addresses of customers along with their flight numbers who booked a flight from New York to LA on 09/30/21.