**Homework 3 for CS542 - Fall 2023**

**Assigned:** Wednesday September 13, 2023  
**Due:** Wednesday September 20, 2023 10am ET (via CANVAS).   
**Maximum:** 100pts.   
**Note:** This homework is to be done by each student individually.   
**Purpose of homework:** To become familiar with the foundations of relational database querying.

The assignment has to be typed, no handwritten solutions are accepted. The following relational algebra symbols are provided for your use in the assignment:

* σ, π, γ, **δ,** ⋈, 🡨, ∩, **∪**

**Problem 1: Getting Started with Oracle.**[30 pts]

This homework problem will give you a little experience with the SQLDeveloper and how to import data into the table from an external file. Follow the directions below to get started.

1. Connect to the WPI Oracle server using **SQLDeveloper**. If you don’t remember how to log in to the WPI Server with SQLDeveloper, then review the document file in the Week 1 Module called " Pre Assignment : Connecting to Oracle Instructions ". There is also a video instruction for your convenience.
2. Create a table SuperBallScores with attributes playerNum of type integer and score of type integer. For your convenience, the DDL is included below: (click on Run Script i.e. f5 button)

CREATE TABLE SuperBallScores (  
playerNum INTEGER,  
score INTEGER  
);

A screenshot of a computer

Description automatically generated

1. Verify that you see the output “Table created.” If you do not see this output after running the above DDL, you’re table was not created correctly.
2. Now we will create a script to generate random data using Python to fill the table you’ve created.
3. Set aside the SQLDeveloper window and open your Python IDE and copy the following code :-

import random

import csv

with open("sample.csv", "w", newline="") as csvfile:

csvwriter = csv.writer(csvfile)

csvwriter.writerow(["PLAYERNUM", "SCORE"])

for i in range(50000):

roll = random.randint(1, 1000)

csvwriter.writerow([i + 1, roll])

This will create a file named “sample.csv” with columns “PLAYERNUM” and “SCORE” and will contain the data that you can import.

1. Return back to your SQLDeveloper window and find the table that you created in the left-hand side column and right-click the table. Then click on Import Data. (click on refresh if you don’t see your table)

A screenshot of a computer

Description automatically generated

1. Browse the “sample.csv” file that you created using the python script and open it.

A screenshot of a computer

Description automatically generated

1. Click on “Next” until you reach the end of the step. Then click on “Finish”

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. Double click on your table and you will be able to see the data (under Data tab) that has been imported into your table.

A screenshot of a computer

Description automatically generated

Now that you have the data imported, you can run queries on the data! For each of the following SQL statements, run them, report the answer, and record how long it took to execute the statement. To show how long an SQL query takes in Oracle, first run the command “SET TIMING ON;”. All subsequent commands will report how long they take to execute.

Lastly please observe which query ran the quickest, which the slowest, and so on. Then indicate if this is what you had expected that would happen, or if you are surprised. Explain your thoughts about your observations.

* SELECT COUNT(playerNum) FROM SuperBallScores;
* SELECT playerNum, MIN(score), MAX (score) FROM SuperBallScores GROUP BY playerNum;
* SELECT T1.playerNum FROM SuperBallScores T1, SuperBallScores T2 WHERE T1.playerNum = T2.playerNum;
* UPDATE SuperBallScores SET score=score - 10;

The data is huge. Thus do NOT submit the actual log showing the loading of the data or the execution of the queries. Instead for each of the above queries simply report the performance results (numbers) you observed.

**Problem 2: Relational Algebra About Employment.** [70 pts]

Relation Emp

eid Integer,

ename Char Width 255,

age Integer,

salary Float,

PRIMARY KEY (eid);

relation Works

eid Integer,

did Integer,

pct\_time Integer,

PRIMARY KEY (eid,did),

FOREIGN KEY (eid) REFERENCES Emp (eid),

FOREIGN KEY (did) REFERENCES Dept (did);

relation Dept

did Integer,

dname Char Width 255,

budget Float,

managerid Integer,

PRIMARY KEY (did),

FOREIGN KEY (managerid) REFERENCES Emp (eid);

1. Find the names and ages of each employee who works in both the Hardware department and the Software department.

2. For each department with more than 10 full-time-equivalent employees (i.e., where the part-time and full-time employees add up to at least that many full-time employees), print the did together with the number of employees that work in that department. Note: for a full-time employee the pct\_time=100 and for a part-time employee the pct\_time<100. Full-time equivalent employees means that the sum of their pct\_time is at least 100).

3. Retrieve the name of each employee whose salary exceeds the individual budgets for all of the departments that he or she works in.

4. Find the managerids of managers who manage only departments with budgets greater than $1 million.

5. Find the enames of managers who manage the departments with the largest budgets. If multiple departments share the maximum budget, make sure to report all of the managers for each of these departments.

Write each query using relational algebra. Make sure to review the lecture notes discussed in class about the relational algebra. It covers a few more operators than what is listed in your text book, such as, special Joins, group-by (G) and aggregation functions. These operators may be helpful for this homework.