***If you do not read the two sections below, you will probably be giving away points.***

***Not including your name here will cost you 5 points.***

NAME: Emiliana Geronimo

**Guidelines:**

1. All problems must be solved using Relational Algebra unless told differently. **Do not submit any SQL code. It will be ignored.**
2. Do not assume files/tables will not change in size in the future unless stated so.
3. Please state any assumptions you are making. Assumptions are required to be “real-world”, reasonable, and accurate.
4. The only permissible Relational algebra syntax is that shown in the textbook
   1. The format for the relational algebra functions is attached on the final page. ***USE ONLY THESE FUNCTIONS PLUS THE AGGREGATE FUNCTIONS INCLUDED IN THE TEXT.***
5. Problem outputs (when requested) can be based on the snapshot of the table presented in the problem unless stated differently.
6. Certainly, if you have questions, you are welcome to call or text me (908-418-6078) or send an email (bforman13@gmail.com).

**Submission Requirements:**

1. This assignment is due Wednesday November 22, 2023, at Midnight.
2. The assignment must be submitted via Canvas.
   1. Submit one document only, unzipped.
   2. It must be readable, and it is your responsibility to confirm this. If I cannot read it, you run the risk of getting a zero.
   3. Handwritten problems will not be accepted unless permission is granted by ME.
3. All work must be your own. The only person you may discuss the assignment with is me (Professor Forman)
   1. You may **NOT** discuss problems with any other student.
   2. You may **NOT** get answers from sites such as Chegg or Homework Hero or any other online site.
   3. Anything not mentioned, that constitutes “***not doing your own work”*** will be considered cheating.
   4. Violation of these requirements will result in a grade of 0.

**Problems 1 - 4**

Using the snapshot of the Yankees Team table, shown below, create relational algebra programs for problems 1- 4. **Assume you are looking at a representative snapshot of the table which has MANY more records.**

**Schema:**

**YankeesID = identifier of Yankees player**

**FirstName = first name of player**

**FamilyName = family name of player**

**Position = Position played**

**Age = Age of player**

**Table = YankeesTeam(PK = YankeesID, Position)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **YankeesID** | **FirstName** | **FamilyName** | **Position** | **Age** |
| 1 | Martin | Billy | 2B | 93 |
| 2 | Derek | Jeter | SS | 44 |
| 3 | Babe | Ruth | LF | 120 |
| 4 | Lou | Gehrig | 1B | 115 |
| 5 | Joe | DiMaggio | CF | 105 |
| 6 | Clete | Boyer | 3B | 83 |
| 7 | Mantle | Mickey | CF | 92 |
| 8 | Yogi | Berra | C | 100 |
| 9 | Roger | Maris | RF | 81 |
| 7 | Mantle | Mickey | 1B | 92 |

**Problem 1 (10 points)**

List only the YankeesID, FirstName, and FamilyName for those Yankees who are older than 100. Show the output as a table based on the snapshot of the YankeesTeam table shown above. **Show your code plus the output.**

**Temp <-**

**Result <-**

**Print(Result)**

**Result:**

|  |  |  |
| --- | --- | --- |
| **YankeesID** | **FirstName** | **FamilyName** |
| 3 | Babe | Ruth |
| 4 | Lou | Gehrig |
| 5 | Joe | DiMaggio |

I used the select function to filter for players over 100 years old, then used the project function to show the columns requested.

**Problem 2 (10 points):**

The Yankees are looking for a new Manager and the only candidates are those shown in the YankeesTeam table. The new manager **cannot** have a position of RF or CF. In addition, the Manager cannot be over the age of 90 and the Manager cannot be under the age of 40.

In a table based on the snapshot shown above, list those players that are eligible to become Manager. The column headings should be the items shown exactly as presented in Bold Face: **YID (=YankeesID), FirstName** (=FirstName), **LastName** (=FamilyName), Age. **Show your code plus the output.**

**Final:**

|  |  |  |
| --- | --- | --- |
| **YID** | **FirstName** | **LastName** |
| 2 | Derek | Jeter |
| 6 | Clete | Boyer |

In this problem, I first used the select function to only show the players that do not have a CF or RF position and named it temp. After that, I used the select function to look for ages between 40 and 90 for only on the Temp table and named it temp2. In temp2 table, I projected the YankeesID, FirstName, and FamilyName and named it result then changed the column names to YID, FirstName, and LastName.

**Problem 3 (10 points):**

Write a relational algebra set of commands to compute the count and average age for Yankees, 100 years old or less by position played. Show the results. For this problem, **base your results on the snapshot of the table**. Remember to show column headings as defined by the relational algebra aggregate functions. **Show your code plus the output.**

**Temp2:**

|  |  |  |
| --- | --- | --- |
| **Position** | **Count\_YankeesID** | **Average\_Age** |
| 2B | 1 | 93 |
| SS | 1 | 44 |
| 1B | 1 | 92 |
| CF | 1 | 92 |
| 3B | 1 | 83 |
| C | 1 | 100 |
| RF | 1 | 81 |

I first used the select function to find all rows that had players with ages greater than or equal to 100 and named the table Temp. Then on the Temp table, I used the aggregation function to count players by position and found the average age and named the table “Temp2”.

**Problem 4 (15 Points)**

Only certain players are eligible for the Yankees Annual World Series Bash. These players must have a position of CF plus one other position. Write a relational algebra query to display all the players who are eligible for the Bash. Show the results of the query in tabular form based on the data from the snapshot shown above. *Use the headings listed below for the table format.*

**Sample output format:**

|  |  |  |  |
| --- | --- | --- | --- |
| **FirstN** | **LastN** | **Pos1** | **Pos2** |
| **Xxxx** | **Yyyyy** | **ZZ** | **VV** |

Write your code so that it is not limited to what is shown in the table snapshot:

1. Do not assume that the table is limited to just the records shown.
2. Show code and output.
3. You can assume that the only positions played for the whole table are those listed in the snapshot. You cannot assume that there will be no new players added to the table, but if there are, you can assume they will play one of the positions listed.
4. Base your output on the snapshot provided.
5. The output must show both positions in the same row. You can assume that an individual will appear at most twice in the table.
6. **DO NOT USE THE JOIN COMMAND ANYWHERE IN THIS PROBLEM – Using the JOIN command will cost 8 points.**

**Final:**

|  |  |  |  |
| --- | --- | --- | --- |
| **FirstN** | **LastN** | **Pos1** | **Pos2** |
| Mantle | Mickey | CF | 1B |

In this problem, I first created 2 tables that only projected the first name, family name, position, and yankeesid and renamed the columns. I named the tables “temp” and “temp2”. I then took the cartesian product of temp and temp2 that gives me all possible combos of the tuples and named that table Temp3. Then, in the temp3 table, I selected rows that shared the same DIDs so we could look at one person at a time and labed it “temp4” table. In the Temp4 table, I then used the select function to find rows that had Pos1 = CF and Pos2 ≠CF and named the table temp5. This gives me the list of candidates that satisfy the select function. After, I only projected the FirstN, LastN, Pos1, and Pos2 of that table.

**For Problem 5, Problem 6, Problem 7, and Problem 8, use the table provided below. Assume that the table MAY change in the future unless told differently.**

***Problems 5, 6, 7, and 8 do NOT have the same solution.***

Table = Comedy (PK = DID, ComedyAward)

|  |  |  |  |
| --- | --- | --- | --- |
| DID | Fname | Lname | ComedyAward |
| 4864 | Lucy | Ricardo | 74 |
| 4957 | Joey | Tribiani | 37 |
| 4771 | Rachel | Greene | 74 |
| 7279 | Phoebe | Boufee | 74 |
| 8437 | Monica | Geller | 63 |
| 9279 | Frasier | Crane | 47 |
| 9279 | Frasier | Crane | 74 |
| 1418 | Daphne | Moon | 86 |
| 4864 | Lucy | Ricardo | 47 |

**Problem 5 (15 points):**

Write a relational algebra expression to list the individuals with ANY ComedyAward that the Frasier Crane shown in the table has won. Display DID, Fname, Lname, ComedyAward for that individual. **Exclude the Frasier Crane record from the output.** Show the results based on the snapshot of the table. You can assume DID uniquely identifies the individual. **Assume that Frasier Crane might get other awards in the future. Your code must reflect this. Show your code plus the output.**

**Assume that the table may change in the future.**

**Final:**

|  |  |  |  |
| --- | --- | --- | --- |
| **DID** | **Fname** | **Lname** | **ComedyAward** |
| 4864 | Lucy | Ricardo | 74 |
| 4771 | Rachel | Greene | 74 |
| 7279 | Phoebe | Boufee | 74 |
| 4864 | Lucy | Ricardo | 47 |

In this problem, I changed the name of columns of Comedy and saved that edit to a table called temp. I then took the cartesian product of temp and Comedy tables. I used the select function on the product to get rows that had DID1 = 9279 which is Fraiser Crane. After, I used select again to have ComedyAward = ComedyAward1, which means that Fraiser’s award number has to match theirs. After, I projected the DID, Fname, Lname, and ComedyAward and used the select function to exclude Fraiser Crane in the results.

**Problem 6 (10 points)**

Student Ray Romano was given the following assignment:

Write a relational algebra expression to list the DIDs with ALL of the ComedyAwards that the Frasier Crane shown in the table has won. Display DID. Show the actual results based on the snapshot of the table that was given.

Note that this is different than Problem 5 in the following way. Problem 5 basically requires that the programmer display any ComedyAward that Frasier Crane has. This means display any DID that has either ComedyAward 47, 74, or others that Frasier picks up in the future. In this problem, Ray is required to display any Comedy characters that have all the ComedyAwards that Frasier has, which means he must have both 47 and 74, plus any other awards that Frasier might receive in the future.

Ray submitted the following relational algebra statement. Show the output of this statement.

A(CA) 🡨 Project <ComedyAward> (Select <ComedyAward=47 AND ComedyAward = 74>(Comedy))

**Ray’s Output:**

|  |
| --- |
| **ComedyAward** |

Ray’s output would not show anything in the result. The select function in relational algebra looks for its conditions being met within the row or tuple. In this case, he used the select function to look for a ComedyAward 47 **AND** 74. This means that the row must meet both conditions to be able to be included in the table. Since the table only shows one ComedyAward at a time, then it cannot satisfy the select condition, therefore there will be no output.

**Problem 7 (10 points)**

Write a relational algebra expression to solve the problem Ray Romano was given in Problem 6 which is re-stated directly below.

Write a relational algebra expression to list the DIDs with ALL of the ComedyAwards that the Frasier Crane shown in the table has won. Display DID. Show the actual results based on the snapshot of the table that was given. **Show your code plus the output.**

**You MUST use the division function. Assume the table will increase in size. Show the output based on the snapshot of the table shown above**

**Result:**

|  |
| --- |
| **DID** |
| 9279 |
| 4864 |

In this problem, I first projected only the DID and ComedyAward and named it first. Then, I specified which rows I need which was DID = 9279. This shows the Comedy Awards that Fraiser Crane has won. After I only projected the DID column and named that table DID. I then divided the “first” table with DID to give me the final table. The final table would have both Fraiser Crane and Lucy Ricardo’s DID.

**Problem 8 (10 points)**

**Write a relational algebra expression to solve Problem 6.**

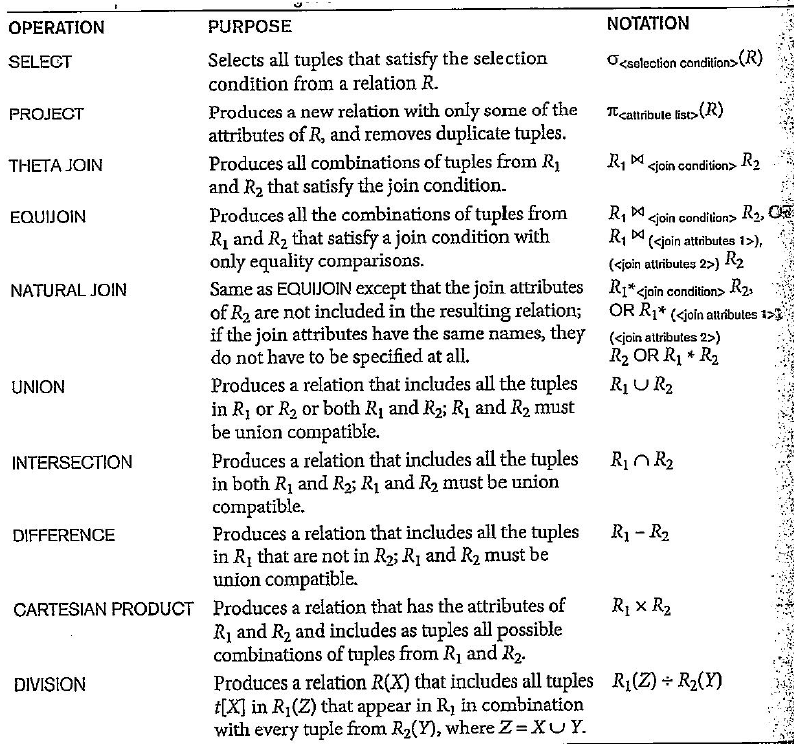
Write a relational algebra expression to list the DIDs with ALL of the ComedyAwards that the Frasier Crane shown in the table has won. Display DID. Show the actual results based on the snapshot of the table that was given.

**You MAY NOT use the division function. You MAY assume the table will *not* change in any way.**

**Result:**

|  |
| --- |
| **DID** |
| 9279 |
| 4864 |

In this problem, I used the select function to select rows that had a ComedyAward value of 74 and then projected the DID and named it temp. Then I created another table and did the same but found rows with a ComedyAward value of 47. After, I used intersection operation to grab the DIDs that were in both columns. That would leave 9279 (Fraiser Crane) and 4864 (Lucy Ricardo).



**You may also use the CROSS function and the applicable aggregate functions, which are not listed here.**