**CS3431 A18 Wong**

**Assignment 3: Relational Algebra and More SQL**

Emily Hao

Due Date: T 9/11 at 11:59pm.

Late Policy: **No late submissions** because solutions will be posted immediately to allow you to study for exam 1. Maximum grade is 100 points. Submission: In PDF or Word format using the Assignment 3 Submission button.

This assignment is to be typed. The following relational algebra symbols are provided for your use in the assignment:

σ, π, ρ, γ, **δ,** ⋈, 🡨, ∩, **∪**

**Part 1: Science Fiction and Nonfiction Books**

Use the abbreviations in place of the full table names in your relational algebra and SQL answers or 5 points off. Primary keys are underlined. ***Foreign keys*** are in bold and italicized.

For problem 1, use the relations given below for tennis players and their associations and tournaments:

P 🡨 Players (playerID, fullName, handedness, birthDate, currentRank, totalEarnings, ***associationID***)

A 🡨 Association (acronym, associationName, foundingYear)

T 🡨 Tournament (tournamentName, year, startingDate, endingDate, prizeMoney)

PT 🡨 PlayersInTournament (***tournamentName***, ***year***, ***playerID***)

1. (10 points) Based on the given primary keys, specify below the foreign key relationships that exist between the tables that would make sense.
   1. Write the constraints using the following format:   
      Foreign Key (TableA.ID1, TableA.ID2) References (TableB.ID1, TableB.ID2)

Foreign Key (PT.tournamentName, Pt.year) Reference (T.tournamentName, T.year)

Foreign Key ( P.associationID) Reference (A.acronym)

Foreign Key ( Pt.player) Reference ( P.playerID)

* 1. Write named SQL constraints for the foreign keys. If a tournament is canceled then all records in referring tables should be deleted. Assume the tables with the field names already exist but without any foreign key constraints. Use ALTER commands to create the foreign key constraints.

ALTER TABLE PlayersInTournament

ADD CONSTRAINT PT\_Year

FOREIGN KEY (year)

REFERENCES Tournament(year)

ON DELETE CASCADE;

ALTER TABLE PlayersInTournament

ADD CONSTRAINT PT\_tournamentName

FOREIGN KEY (tournamentName)

REFERENCES Tournament(tournamentName)

ON DELETE CASCADE;

ALTER TABLE PlayersInTournament

ADD Constraint PT\_plater

Foreign Key (playerID)

References Players (playerID);

Alter table Players

Add constraint P\_associationID

Foreign Key (associationID)

References Association (acronym);

For problems 2 to 4, use the relations given below:

**A 🡨 Authors** (authorID, firstName, lastName, birthDate, address)

**SF 🡨 SFBooks** (ISBN, title, year, price, awardWinner, publisherName, ***authorID***) -- science fiction books

**NF 🡨 NFBooks** (ISBN, title, year, price, awardWinner, publisherName, ***authorID***) -- non-fiction books

**P 🡨 Publishers** (publisherName, city, phone)

**W 🡨 Warehouses** (code, address, city)

**S 🡨 Stocks** (warehouseCode, ISBN, numberOfBooks)

1. (20 points) For each publisher, list the publisher name and city, and the average price of the science fiction books published after the year 2000 by that publisher. For this question, use natural joins.
   1. Write the relational algebra

γpublisherName, city, avg(price) σ year > 2000 P ⨝ SF

* 1. Write the SQL code for the above, but sorted by city and then publisher names

Select publisherName, city avg(price)

From SF natural Join P

where year > 2000

group by city, publisherName

order by city, publisherName;

1. List the warehouse code and city, and the total number of books it contains for those warehouses that have more than 1000 books.
2. Write the relational algebra

π warehouseCode, city, numberOfBooks σ numberOfBooks > 1000 W ⨝ W. code = S.warehouseCode S

1. Write the SQL code

select warehouseCode, city, numberOfBooks

From W inner Join S

On S.warehouseCode = W.code

Where numberOfBooks >1000;

1. (20 points) For authors who published both science fiction and non-fiction books, list the author’s first name, last name, and how many books they wrote. For this question, use natural joins.
   1. Write the relational algebra

γ authorID, firstName, lastName, count(authorID) A⨝ SF ⨝ NF

* 1. Write the SQL code for the above using subqueries in the from/join statement, but sorted by author last name, then author first name.

select authorID, firstName, lastName, count(authorID) \* 2

from Authors natural join (select SFBooks.authorID

From SFBooks inner join NFBooks

on SFBooks.authorID = NFBooks.authorID)

group by authorID, firstName, lastName

order by lastName,firstName;

**Part 2: Relational Algebra [30 Points (5 Points each subpart)]**

Given the relations M and N:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **M** | | | |  | **N** | | |
| **A** | **B** | **C** | **D** |  | **X** | **Y** | **B** |
| 14 | 4 | Henry | David |  | 3 | Matt | 5 |
| 8 | 2 | Mary | Helen |  | 2 | Tom | 8 |
| 2 | 6 | Jane | Susan |  | 8 | Mary | 61 |
| 61 | 3 | Tom | Paul |  | 8 | Lisa | 2 |
| 2 | 5 | Lisa | Mary |  | 5 | John | 8 |
| 24 | 9 | Mark | John |  | 13 | Tom | 42 |
| 8 | 1 | Paul | Amy |  | 42 | Henry | 2 |
| 5 | 2 | Dan | Matt |  |  |  |  |

For each relational algebra statement

1. calculate the following output tables and be careful in labeling the attributes
2. write the corresponding SQL code using subqueries as appropriate
3. π B, C, D, Y as E σ A \* B > X (M ⋈N)

|  |  |  |  |
| --- | --- | --- | --- |
| B | C | D | E |
| 2 | Mary | Helen | Lisa |
| 5 | Lisa | Mary | Matt |
| 2 | Dan | Matt | Lisa |

select B, C, D, Y as E

From M natural join N

where A \* B > X;

1. γC, D, sum(A+X) AS SUM(M ⋈A=X N)

|  |  |  |
| --- | --- | --- |
| C | D | SUM |
| Lisa | Mary | 4 |
| Dan | Matt | 10 |
| Paul | Amy | 32 |
| Mary | Helen | 32 |
| Jane | Susan | 4 |

select C, D, sum(A + X) As SUM

From M inner join N

On M.A = N.X

Group by C, D;

1. πR.X,R.Y,N.B (N ⋈ N.X = R.X and N.Y = R.Y ρR (σX<5 (N) **∪** πB as X,D as Y,A as B σA<B (M)))

|  |  |  |
| --- | --- | --- |
| X | Y | B |
| 3 | Matt | 5 |
| 2 | Tom | 8 |

select R.X, R.Y, N.B

From N inner join (select B As X, D As Y, A As B

from M

where A < B

Union

Select \*

from N

where X < 5 ) R

On N.X = R.X And N.Y = R.Y ;