CS482/502 Database Management Systems I

Assignment: Relational Algebra

Assume that you are given the following relational schemas for the basketball team at NMSU.

- Player (ID: integer, Name: varchar(64), Birthday: date, Address: varchar(128), Email: varchar(32), PhoneNumber: char(10), PlayPos: varchar(16))

- Manager (ID: integer, LoginID: varchar(16), Name: varchar(64), Password: varchar(8), Birthday: date, Address: varchar(128), Email: varchar(32), PhoneNumber: char(10))

- ManagerCertificate (ManagerID: integer, CertificateId: integer, Certificate: blob)
    - Foreign key: ManagerID references Manager(ID)

- Doctor (Email: varchar(32), Name: varchar(64), PhoneNumber: char(10))

- TakeExam (PlayerID: integer, DocEmail: varchar(32), TestDate: date, TestResult: varchar(256))
    - Foreign key: PlayerID references Player(ID)
    - Foreign key: DocEmail references Doctor(Email)

- Stats (PlayerID: integer, Year: char(4), TotalPoints: integer, ASPG: integer)
    - Foreign key: PlayerID references Player(ID)

- Training (TrainingName: varchar(256), Instruction: varchar(256), TimePeriodInHour: integer)

- AssignTraining (PlayerID: integer, ManagerID: integer, TrainingName: varchar(256))
    - Foreign key: PlayerID references Player(ID)
    - Foreign key: ManagerID references Manager(ID)
    - Foreign key: TrainingName references Training(TrainingName)

- Game (GameID: integer, Date: date, Result: varchar(16), PlayingVenue: varchar(256), OpponentTeam: varchar(32))

- Play (PlayerID: integer, GameID: integer)
    - Foreign key: PlayerID references Player(ID)
    - Foreign key: GameID references Game(GameID)

1.) Show the names and ID's of all players whose play position is "center".

```
SELECT Name, ID
FROM Player
WHERE PlayPos = “center”;
```

$\Pi_{name,ID}(\sigma_{PlayPos = \text{'center'}}(Player))$

2.) Show the total points that player “Pistol Pete” has scored each year (assume there is only one Pistol Pete).

```
SELECT Stats.Year, Stats.TotalPoints
FROM Player, Stats
WHERE Player.ID = Stats.PlayerID AND Player.Name = “Pistol Pete”
GROUP BY Stats.Year;
```

$S.year\ \mathcal{G}\ SUM(S.totalPoints)\ \Pi_{S.year,\ S.totalPoints}(\sigma_{S.ID = P.playerID\ \wedge\ S.name = \text{'Pistol Pete'}}\ (\rho_P(Player)\ X\ \rho_S(Stats)))$

3.) Show the names and emails of every manager who has exactly 2 distinct certificates.

```
SELECT M.Name, M.Email
FROM Manager AS M, ManagerCertificate AS MC
WHERE M.ID = MC.ManagerID
GROUP BY M.Name
HAVING count distinct (MC.CertificateID) = 2;
```

Cannot use having with relational algebra since there is no concept of ‘HAVING’, however, some students found a method from here:
https://cs.ulb.ac.be/public/_media/teaching/infoh417/sql2alg_eng.pdf

Which would lead to an answer like this:

$\Pi_{M.name,\ M.email}(\sigma_{COUNT\text{-}DISTINCT(MC.certificateID) = 2}\mathcal{G}_{MC.name,COUNT\text{-}DISTINCT(MC.certificate)}(\sigma_{M.ID = MC.managerID}))$
$(\rho_M(Manager)\ X\ \rho_{MC}(ManagerCerteficate))))$

However, you must assume that all CertificateIDs are unique, since we cannot use distinct in RA.

4.) Show the names of every player who has played a game at "The Pit" and won (Result = "win"), in descending order of age

SELECT P.Name

FROM Player AS P, Play, Game

WHERE P.ID = Play.PlayerID AND Play.GameID = Game.GameID AND Game.Result = "win"
AND Game.PlayingVenue = "The Pit";

ORDER BY Birthday ASC

We cannot use order by in relational algebra, but this is the query without it:

$\Pi_{p.name}(\sigma_{P.ID = X.playerID \wedge G.gameID = X.gameID \wedge G.playedVenue = \text{'The Pit'} \wedge G.result = \text{'win'}} (\rho_P(Player) \times \rho_G(Game) \times \rho_x(Play)))$

5.) Show all the information of Doctors who have given exams

SELECT *

FROM Doctor, TakeExam

WHERE Doctor.Email = TakeExam.DocEmail;

$\sigma_{D.email = TE.doctorEmail}(\rho_D(Doctor) \times \rho_{TE}(TakeExam))$

6.) Find the games that players named "Pistol Pete" and "Lobo Louie" have played in, using set operators (UNION, INTERSECT, MINUS, etc…). Show the game's date, venue, and result

(SELECT G.Date, G.PlayingVenue, G.Result

FROM Player as P, Game as G, Plays

WHERE P.ID = Plays.PlayerID AND Plays.GameID = G.GameID AND P.name = "Pistol Pete")

INTERSECT

(SELECT G.Date, G.PlayingVenue, G.Result

FROM Player as P, Game as G, Plays

WHERE P.ID = Plays.PlayerID AND Plays.GameID = G.GameID AND P.name = “Lobo Louie”);

$\Pi_{G1.date,\ G1.venue,\ G1.results}$ ($\sigma_{P1.ID = X1.playID \wedge G1.gameID = X1.gameID \wedge P1.name = \text{‘Pistol Pete’}}$ (ρ_{P1}(Player) X ρ_{P1}(Game) X ρ_{X1}(Play)))

$\cap$

$\Pi_{G2.date,\ G2.venue,\ G2.results}$ ($\sigma_{P2.ID = X2.playID \wedge G2.gameID = X2.gameID \wedge P2.name = \text{‘Lobo Louie’}}$ (ρ_{P2}(Player) X ρ_{G2}(Game) X ρ_{X2}(Play)))

7.) Perform the same query as problem 6, without using set operators (UNION, INTERSECT, MINUS, etc…)

SELECT G.Date, G.PlayingVenue, G.Result

FROM Game, Player, Plays

WHERE P.name = “Pistol Pete” and G.GameID IN

(SELECT G.GameID

FROM Player as P, Game as G, Plays

WHERE P.ID = Plays.PlayerID AND Plays.GameID = G.GameID AND P.name = “Lobo Louie”);

There is no ‘IN’ in relational algebra, but it would be possible to rewrite this as a nested subquery using a temp variable

Temp $\leftarrow$ $\Pi_{G2.gameID}$($\sigma_{P2.ID = X2.playID \wedge G2.gameID = X2.gameID \wedge P2.name = \text{‘Lobo Louie’}}$ (ρ_{P2}(Player) X ρ_{G2}(Game) X ρ_{X2}(Play)))

$\Pi_{G1.date,\ G1.venue,\ G1.results}$ ($\sigma_{P1.ID = X1.playID \wedge G1.gameID = X1.gameID \wedge P1.name = \text{‘Pistol Pete’} \wedge G1.gameID = Temp.gameID}$ (ρ_{P1}(Player) X ρ_{G1}(Game) X ρ_{X1}(Play) X ρ_{E1}(Temp)))

8.) Find the Names and IDs of players who have scored more points than the average player

SELECT P.Name, P.ID

FROM Stats S, Player P

WHERE P.ID = S.PlayerID and S.TotalPoints > (SELECT AVG(TotalPoints) FROM Stats);

Temp ← $\mathcal{G}_{AVG(TotalPoints)}(Stats)$

$\Pi_{P.name,\ P.ID}\ (\sigma_{P.ID = S.playerID\ \wedge\ S.totalPoints > Temp}(\rho_{P}(Player)\ X\ \rho_{S}(Stats)))$

9.) Show all players that were born on the same day (I.E. If Bob and Joe were both born on 12/25/95, and Jim and Steve were both born on 7/4/94, show the names and the birthday they share)

SELECT P1.Name, P2.Name, Birthday

FROM Player P1, Player P2

WHERE P1.Birthday = P2.Birthday AND P1.id != P2.id;

$\Pi_{P1.name,\ P2.name,\ P1.birthday}(\sigma_{P1.birthday = P2.Birthday\ /\backslash\ P1.name\ != P2.name}(\rho_{P1}(Player)\ X\ \rho_{P2}(Player)))$

10.) Find the total number of points the Aggie basketball team scored in 2016

SELECT SUM(TotalPoints)

FROM Stats

WHERE Year = 2016;

$\mathcal{G}_{SUM(S.totalPoints)}(\sigma_{S.year = 2016}(\rho_{S}(Stats)))$