Assignment-2

CS482/502 Database Management Systems I **Spring 2020**

Assignment: Relational Algebra + SQL

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";
```

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

3.) Show the names of every player who has played a game at "The Pit" and won (Result = "win")

```
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";
```

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

4.) 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”);

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

$\Pi^{G2.date,\ G2.venue,\ G2.results}\ (\sigma_{P2.ID = X2.playID\ \wedge\ G2.gameID = X2.gameID\ \wedge\ P2.name = \text{'Lobo Louie'}}\ (\rho_{P2}(Player)\ X\ \rho_{G2}(Game)\ X\ \rho_{X2}(Play)))$

5.) 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);

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

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

Assume that you are given the following relational schemas.

- members (memb_no int(3), name varchar(64))
- books (isbn int(6), title varchar(64), authors varchar(128), publisher varchar(128))
- borrowed (memb_no int(3), isbn int(6))

Write an SQL Query for each of the following.

1.) **Show the names of members who borrowed books with title “Math”.**

```
SELECT m.name
  FROM members m, books b, borrowed br
  WHERE m.memb_no = br.memb_no and br.isbn = b.isbn and b.title = "Math";
```

2.) **Show the details of members whose name does not start with ‘J’.**

```
SELECT *
FROM members
WHERE name not like "j%";
```

3.) **Find the number of books borrowed by each member and show them in descending order.**

```
SELECT m.memb_no, count(m.memb_no)
FROM members m, borrowed br
WHERE m.memb_no = br.memb_no
GROUP BY m.memb_no
ORDER BY count(m.memb_no) DESC;
```

4.) **Show the details of members whose name contains ‘A’.**

```
SELECT *
FROM members
WHERE name like "%a%";
```

5.) **Find the distinct publisher name of the book which has been borrowed by “Sam”.**

```
SELECT DISTINCT(b.publisher)
FROM members m, books b, borrowed br
WHERE m.memb_no = br.memb_no and br.isbn = b.isbn and m.name = "Sam";
```