

# INFSCI 2710 Database Management, Fall 2022

## Homework 1: Relational Algebra, SQL

100 pts

Due Date: 9/28 at the end of the day. Please submit a pdf to the Canvas assignment.

Preparations: We are using the same dataset we used in the lab, please download two files (“create db.sql” and “insert table.sql”) from the lab. Run SQL queries from both files (make sure you run “create db” first).

Consider the relational database that provided in the lab material (Underlines attributes in bold are the primary keys.) **For each question in this section, you need to provide the SQL query and the screen shot of the output of that query from either MySQL Workbench or from the terminal** (make sure it is formatted properly).

classroom (**building**, **room\_number**, capacity)  
department (**dept\_name**, building, type, budget)  
instructor (**ID**, name, dept\_name, salary)  
course (**course\_id**, title, dept\_name, credits)  
section (**course\_id**, **sec\_id**, **semester**, **year**, building, room\_number, time\_slot\_id)  
teaches (**ID**, **course\_id**, **sec\_id**, **semester**, **year**)  
student (**ID**, name, dept\_name, tot\_cred)  
takes (**ID**, **course\_id**, **sec\_id**, **semester**, **year**, grade)  
advisor (**s\_ID**, **i\_ID**)  
time\_slot (**time\_slot\_id**, **day**, **start\_hr**, **start\_min**, end\_hr, end\_min)  
prereq(**course\_id**, **prereq\_id**)

**Q1 [10 pt]** Specify a **relational algebra expression** and an **SQL query** to find the names of all the instructors from Biology department

$\pi_{name}(\sigma_{dept\_name='Biology'}(instructor))$

```
SELECT name
FROM instructor
WHERE dept_name = 'Biology'
```

**Q2 [10 pt]** Specify a **relational algebra expression** and an **SQL query** to find the names of courses in Computer science department which have 3 credits

$\pi_{name}(\sigma_{dept\_name='Comp.Sci.' \wedge credits = 3}(course))$

```
SELECT title
FROM course
WHERE dept_name = 'Comp. Sci.' AND credits = 3
```

**Q3 [5 pt]** Specify a SQL query to find all course\_id and title of all courses taken by the student with ID 12345

```
SELECT T.course_id, C.title
FROM course C, takes T
WHERE T.course_id = C.course_id AND T.ID = '12345'
```

**Q4 [5 pt]** Specify a SQL query to find names of all the courses taught by computer science department in 2009 Fall.

```
SELECT S.course_id, C.title, S.semester, S.year
FROM section S, course C
WHERE S.course_id = C.course_id AND C.dept_name = 'Comp.Sci.' AND S.year = '2009' AND S.semester = 'Fall'
```

Consider another relational database that describe NBA player stats (Underlines attributes in bold are the primary keys.)

Player (**ID**, name, position, height, weight, teamID)

Team (**ID**, name, city)

Game (**gameID**, homeTeamID, awayTeamID, homeScore, awayScore)

GameStats (**playerID**, **gameID**, points, assists, rebounds)

Hints:

- (1) Two teams may play each other multiple times each season
- (2) GameStats records the performance statistics of a player within a game. A player may not play in every game, in which case it will not have its statistics recorded for that game.

**Q5[20 pt]** Write SQL DDL statements to create the above tables. Make sure that you capture the primary and foreign key constraints (if applicable), choose appropriate domain (data) type and constraints for each attribute.

```
(ID integer PRIMARY KEY,  
    name char(30),  
    position char(30),  
    height numeric(3,2),  
    weight numeric(3,2),  
    teamID char(10),  
    FOREIGN KEY(teamID) REFERENCES Team(ID))
```

```
CREATE TABLE Team  
(ID integer PRIMARY KEY,  
    name char(20),  
    city char(20))
```

```
CREATE TABLE Game  
(gameID integer PRIMARY KEY,  
    homeTeamID(10),  
    awayTeamID(10),
```

```
homeScore int,  
awayScore int,  
FOREIGN KEY(homeTeamID) REFERENCES Team(ID),  
FOREIGN KEY(awayTeamID) REFERENCES Team(ID))
```

```
CREATE TABLE GameStats  
(playerID integer,  
gameID integer,  
points int,  
assists int,  
rebounds int,  
PRIMARY KEY (playerID, gameID),  
FOREIGN KEY (playerID) REFERENCES Player(ID),  
FOREIGN KEY (gameID) REFERENCES Game(gameID))
```

**Q6 [10 pt]** Specify a SQL query to find the gameID, scores of both team of the games that are played at the Philadelphia home court

```
SELECT G.gameID, G.homeScore, G.awayScore  
FROM Game G, Team T  
WHERE Team.ID = Game.homeTeamID AND Team.city = 'Philadelphia'
```

**Q7 [10 pt]** Specify a SQL query to find points, assist and rebounds of the player whose name is “James” in all the home game he played for his team

```
SELECT GS.points, GS.assists, GS.rebounds  
FROM Gamestats GS, Game G, Player P  
WHERE P.name = 'James' AND P. teamID= G.homeTeamID AND P.ID = GS.playerID
```

**Q8[10 pt]** Specify a SQL query to find the gameID of games that Warriors won (including both home and away games).

```

SELECT G.gameID
FROM Game G, Team T
WHERE T.name = 'Warriors' AND (G.homeScore > G.awayScore AND T.ID = G.homeTeamID) OR
(G.homeScore < G.awayScore AND T.ID = G.awayTeamID)

```

**Q9[20 pt]** Consider the following relational algebra expression

$$\pi_{GameStats.playerID}(\sigma_{(GameStats.points \geq 10) \wedge (GameStats.assist \geq 10) \wedge (GameStats.rebound \geq 10)} GameStats)$$

1) How many attributes will the result have?

1

2) Write in English what question the expression is trying to answer (e.g. describe what would be the result of the expression).

Find the player id of the players that earned “triple doubles” at least once (A “triple double” is a game in which the player’s number of assists, rebounds, and points are all in the double digits)

3) Translate the expressions in 3) into SQL.

```

Select P.name
From Player P, GameStats GS
Where P. ID = GS. playerID
And GS. points ≥ 10
And GS. assists ≥ 10
And GS. rebounds ≥ 10

```