CSC365 Spring 2019

Assignment 2&3

Relational Algebra & SQL

100 points total

Due: May 23 @ midnight

Purpose

In this assignment you are going to practice translating queries expressed in English to queries in **Relational Algebra** and then to queries in **SQL**.

The relational algebra is very important because 1) it provides a formal foundation for relational model operations, 2) it is used as a basis for implementing and optimizing queries in the query processing and optimization modules that are integral parts of relational database management systems, and 3) some of its concepts are incorporated into the SQL.

Therefore, even though this may be the last time you use the Relational Algebra to write queries unless you move on to advanced database courses, I strongly believe that doing this assignment will help you understand how SQL queries are interpreted by DBMS and why SQL queries produce results you see.

Preface

Consider the example from Assignment 1 about soccer players and teams. Assume you have the following tables. You can find actual create table statements appended at the end of this file.

Team(id, name, shortName, abbr)

Player(id, team_id, shirt_num, first, last, position, height, weight, birth_country, birthdate)

Game(id, date, home_team_id, away_team_id, score_home, score_away)

Event(id, game_id, type, count, player_id)

PART 1: Relational Algebra

Write the solution to the following queries in **relational algebra** (10 points per query). Go over the slides and make sure you follow proper syntax. A common mistake is to project an attribute that doesn't exist. Feel free to introduce as many intermediate relations as you like using the linear notation. For example, in linear notation you can create an intermediate relation which stores intermediate results of an expression of relational algebra like this:

$P(id, name, team\_id) := \pi_{id, name, team\_id} (Player \bowtie_{id=player\_id} (\sigma_{type='yellow\ card'} (Event)))$

Then you can use P in subsequent expressions:

$P \bowtie_{team\_id = Team.id} Team$

For this part, you can hand write your solutions on paper and turn that in directly to me or scan the paper and submit the file to polylearn. If you want, you can write expression tree of relational algebra, instead.

1. Which soccer player has scored the most total goals? You want to create an intermediate relation that stores the number of goals scored by each soccer player. Also you probably want to create an intermediate relation that stores the max number of goals from the previous intermediate relation.
2. Which soccer player has scored the most goals in a single game? You want to create an intermediate relation that stores the maximum number of goals scored in a single game.
3. Which team has scored the most total goals? You can create intermediate relations for the total number of goals for the home teams and the total number of goals for the away teams. Aggregate the data by creating a relation that stores the total number of goals for each team.
4. Which team has scored the most goals in a single game? You can create intermediate relations for goals scored for home teams (team name and goals), away teams, and for union of two relations. You want to create another intermediate relation that stores just the number of most goals scored.
5. Which team did not win a single game? Again, you can create as many intermediate relations as you want. For example, you might want to create an intermediate relation for storing teams who won games as either the home team or as the away team.

PART 2: SQL

Write the solution to the following queries (exactly the same queries as in the PART 1) in **SQL** (10 points per query).

For this part, I need you to write SQL queries in a text file and save it as assignment2.sql and submit it to polylearn.

1. Which soccer player has scored the most total goals?
2. Which soccer player has scored the most goals in a single game?
3. Which team has scored the most total goals?
4. Which team has scored the most goals in a single game?
5. Which team did not win a single game?

Appendix: Create Table Statements

```
CREATE TABLE IF NOT EXISTS Team (
  id INTEGER NOT NULL,
  name VARCHAR(50),
  shortName VARCHAR(20),
  abbr CHAR(3),
  PRIMARY KEY (id)
);
CREATE TABLE IF NOT EXISTS Player (
  id INTEGER NOT NULL,
  team_id INTEGER NOT NULL,
  shirt_num INTEGER DEFAULT NULL,
  first VARCHAR(50) NOT NULL,
  last VARCHAR(50) NOT NULL,
  position VARCHAR(50) DEFAULT NULL,
  height INTEGER DEFAULT NULL,
  weight INTEGER DEFAULT NULL,
  birth_country VARCHAR(50),
  birthdate DATE DEFAULT NULL,
  PRIMARY KEY (id),
  FOREIGN KEY (team_id) REFERENCES Team (id)
);
```

```
CREATE TABLE IF NOT EXISTS Game (
  id INTEGER NOT NULL,
  date DATE,
  home_team_id INTEGER,
  away_team_id INTEGER,
  score_home INTEGER,
  score_away INTEGER,
  PRIMARY KEY (id),
  FOREIGN KEY (home_team_id) REFERENCES Team (id),
  FOREIGN KEY (away_team_id) REFERENCES Team (id)
);
CREATE TABLE IF NOT EXISTS Event (
  id INTEGER NOT NULL,
  game_id INTEGER NOT NULL,
  type VARCHAR(50) NOT NULL,
  count INTEGER,
  player_id INTEGER,
  PRIMARY KEY (id, game_id),
  FOREIGN KEY (game_id) REFERENCES Game (id),
  FOREIGN KEY (player_id) REFERENCES Player (id)
);
```