CSE 344 Section 2 Worksheet

Joins Examples

Given tables created with these commands:

```
CREATE TABLE A(a int);
INSERT INTO A VALUES (1), (2), (3), (4);
CREATE TABLE B(b int, c TEXT);
INSERT INTO B VALUES (3, 'a'), (3, 'b'), (4, 'a'), (5, 'a'), (6,
'b');
```

What's the output for each of the following:

```
SELECT * FROM A INNER JOIN B ON A.a=B.b;
SELECT * FROM A LEFT OUTER JOIN B ON A.a=B.b and B.c='b';
SELECT * FROM A LEFT OUTER JOIN B ON A.a=B.b WHERE B.c='b';
```

SQL Practice (Movie-Actor)

```
CREATE TABLE Movies (
  id int, name varchar(30), budget int,
  gross int, rating int, year int,
  PRIMARY KEY (id) );
CREATE TABLE Actors (
  id int, name varchar(30), age int,
  PRIMARY KEY (id) );
CREATE TABLE ActsIn (
  mid int, aid int,
  FOREIGN KEY (mid) REFERENCES Movies (id),
  FOREIGN KEY (aid) REFERENCES Actors (id) );
```

(a) What is the number of movies, and the average rating of all movies that the actor ”Patrick Stewart” has appeared in?

(bonus) What movies have no actors? Return movie names of those movies.
Select M.name

(b) What is the minimum age of an actor who has appeared in a movie where the gross of the movie has been over $1,000,000,000?

(c) What is the budget of each movie released in 2017 whose oldest actor is less than 30?

Self Join

Consider the following over-simplified Employee table, listing employees and their boss (if any):

```
CREATE TABLE Employees (id int not null, bossOf int);
```

Suppose all employees have an id which is not null. How would we find all distinct pairs of employees with the same boss?

Additional Movie & Director Practice

Movies and Directors (19wi section 3 #1)

```
CREATE TABLE Director (
   id      INT PRIMARY KEY,
   name    VARCHAR(75),
   country VARCHAR(75));
```

```
CREATE TABLE Movie (
   id     INT PRIMARY KEY,
   name   VARCHAR(75),
   did    INT REFERENCES Director,
   year   INT,
   budget INT);
```

Find the id and name of all directors who have directed more than 20 movies.

Additional Self Join Practice

Consider the following over-simplified Employee table

```
CREATE TABLE Employees (
  id int NOT NULL,
  bossOf int);
```

Note that all employees have an id that is not null, but they may have a null “bossOf” entry, or the bossOf entry may refer to employees already left the company. How do we find the id of all employees who are the boss of at least one other employee? Ensure that the bossOf value refers to a current employee in the Employees relation.

Consider the Employees relation {(1, NULL), (2, NULL), (5, 1), (5, 2), (5, NULL), (3, NULL)}. How many current employees is the employee with id=5 boss of? (i.e. how many employees works for employee with id=5)

Write a query that returns a relation with the id of each employee and the count of how many employees (i.e., any non-null id) they are the boss of.