

CSE 344 Section 2 Worksheet

1. Joins Examples

Given tables created with these commands:

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

What's the output for each of the following:

```
SELECT *  
  FROM A INNER JOIN B  
    ON A.a=B.b;
```

```
SELECT *  
  FROM A RIGHT OUTER JOIN B  
    ON A.a=B.b;
```

```
SELECT *  
  FROM A LEFT OUTER JOIN B  
    ON A.a=B.b;
```

```
SELECT *  
  FROM A FULL OUTER JOIN B  
    ON A.a=B.b;
```

```
SELECT * FROM A INNER JOIN B; (Challenging question!)
```

2. Self Join

Consider the following over simplified Employee table:

```
CREATE TABLE Employees (id int, bossId int);
```

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

3. SQL Aggregates Practice

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

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?

What is the name and budget of each movie released in 2017 whose oldest actor is less than 30?

More Challenging

```
CREATE TABLE Class (  
    dept    VARCHAR(50),  
    number INT,  
    title   VARCHAR(50),  
    PRIMARY KEY (dept, number));  
  
CREATE TABLE Instructor (  
    username VARCHAR(50) PRIMARY KEY,
```

```
fname      VARCHAR(50),  
lname      VARCHAR(50),  
started_on CHAR(10));
```

```
CREATE TABLE Teaches(  
  username VARCHAR(50), -- alternately, can use the REFERENCES syntax here  
  dept      VARCHAR(50),  
  number    INT,  
  PRIMARY KEY (username, dept, number),  
  FOREIGN KEY (username) REFERENCES Instructor,  
  FOREIGN KEY (dept, number) REFERENCES Class);
```

1. For each instructor, return their username and the number of classes they teach.
2. Return the first and last name of the newest instructor(s) (who started on the latest date). Assume Instructor.started_on uses yyyy-mm-dd format. If there are multiple instructors, list all of them.
3. How many classes are currently being taught by at least one instructor?
4. Return the first name and last name of the instructors who teach the most number of classes. If there are multiple instructors, list all of them.