Section 3 Worksheet Solutions

Part 1: Movies and Directors

```
CREATE TABLE Movie (
      movie_name VARCHAR(75),
      movie_id INT,
      director_id INT,
      year_released INT,
      budget INT,
      PRIMARY KEY(movie_id),
      FOREIGN KEY(director_id) REFERENCES Director(director_id)
);

CREATE TABLE Director (
      director_id INT,
      director_name VARCHAR(75),
      director_country VARCHAR(75),
      PRIMARY KEY(director_id)
);
```

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

```
SELECT d.director_id, d.director_name
FROM Director d, Movie m
WHERE d.director_id = m.director_id
GROUP BY d.director_name, d.director_id
HAVING COUNT(*) > 20;
```

2. For each director, find the corresponding movie that has the highest budget.

```
-- This question is about finding witnesses.
WITH MovieMaxBudget AS
     ( SELECT m.director_id AS director_id, max(m.budget) AS max_budget
       FROM Movie m
       GROUP BY m.director_id )
SELECT d.director_name, m.movie_name
FROM Movie m, Director d, MovieMaxBudget mmb
WHERE m.director_id = d.director_id AND d.director_id = mmb.director_id
AND mmb.max_budget = m.budget;
```

Part 2: Classes and Instructors

```
CREATE TABLE Class (
      dept VARCHAR(6),
      number INTEGER,
      title VARCHAR(75),
      PRIMARY KEY (dept, number)
);

CREATE TABLE Instructor (
      username VARCHAR(8),
      fname VARCHAR(50),
      lname VARCHAR(50),
      started_on CHAR(10),
      PRIMARY KEY (username)
);

CREATE TABLE Teaches (
      username VARCHAR(8),
      dept VARCHAR(6),
      number INTEGER,
      PRIMARY KEY (username, dept, number),
      FOREIGN KEY (username) REFERENCES Instructor(username),
      FOREIGN KEY (dept, number) REFERENCES Class(dept, number)
);
```

1. How many classes are being taught by at least one instructor?

```
-- Solution 1
SELECT COUNT(DISTINCT number) AS class_count
FROM Teaches;

/* General case: we'll solve using subqueries and grouping - first, we group
the Teaches table by department and number in a subquery, then we count the
number of groups in the top-level query.

Note that we don't care what the subquery tuples are, only how many
tuples/groups there are, so we return dummy tuples containing only the
constant 1. */

-- Solution 2
SELECT COUNT(*) AS class_count
FROM   ( SELECT 1
         FROM Teaches
         GROUP BY dept, number ) x ;
```

2. Which instructors teach more than 1 class? Give the username, first name, and last name of these instructors.

```
-- With grouping (no subquery)
SELECT i.username, i.fname, i.lname
FROM Instructor i, Teaches t
WHERE i.username = t.username
GROUP BY i.username, i.fname, i.lname
HAVING COUNT(*) > 1;

-- Without grouping (uses subquery)
SELECT i.username, i.fname, i.lname
FROM Instructor i
WHERE 1 < ( SELECT COUNT(*)
            FROM Teaches t
            WHERE t.username = i.username );

/* We refer to this kind of subquery as a correlated subquery. A correlated
subquery uses values from the outer query. */
```

3. Which CSE courses do neither Dr. Levy (‘levy’) nor Dr. Wetherall (‘djw’) teach? Give the department, number, and title of these courses.

```
-- This query is wrong. Why?
SELECT c.dept, c.number, c.title
FROM Class c, Teaches t
WHERE c.dept = 'CSE' AND c.dept = t.dept AND c.number = t.number AND
t.username NOT IN ('levy','djw');

/* The trick to this problem is that more than one instructor may teach a
single course. For example, Dr. Levy might teach CSE 311, but another
instructor may teach the course as well (e.g. Adam Blank). With this example,
the above query would incorrectly return CSE 311. */

-- Correct solution
SELECT *
FROM Class c
WHERE c.dept = 'CSE'
AND c.number NOT IN ( SELECT c.number
                      FROM Class c, Teaches t
                      WHERE c.dept = t.dept AND c.number = t.number AND
                      t.username IN ('levy', 'djw') );
```