CSE 344 Section 2 Worksheet

1. 3 Valued-Logic

```
Given the table created with these commands:
     CREATE TABLE A (a int, b int);
      INSERT INTO A VALUES (1, 1), (2, 10), (3, NULL);
What is the output for each of the following:
SELECT A.a FROM A
                                         SELECT A.a FROM A
  WHERE A.b < 5;
                                           WHERE A.b >= 5;
1
                                        SELECT A.a FROM A
SELECT A.a FROM A
                                         WHERE A.b < 5 OR
  WHERE A.b != 1 AND
                                            A.b IS NULL;
    A.b != 10;
                                       1
[no output]
(3, NULL) returns unknown for (3, NULL) returns true on the both conditions due to second condition only
3-value logic.
2. 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 *
                                         SELECT *
  FROM A INNER JOIN B
                                          FROM A RIGHT OUTER JOIN B
      ON A.a=B.b;
                                             ON A.a=B.b;
alb
                                         alb
3 | 3
                                         3 | 3
4 | 4
                                         4 | 4
                                          | 5
                                          16
SELECT *
 FROM A LEFT OUTER JOIN B
                                        SELECT *
      ON A.a=B.b;
                                          FROM A FULL OUTER JOIN B
alb
                                               ON A.a=B.b;
1 |
                                         a|b
2 |
                                         11
3 | 3
                                         21
                                         3 | 3
4 | 4
                                         4 | 4
                                          15
                                          16
```

```
SELECT * FROM A INNER JOIN B; (Challenging question!)
alb
                                        Sidenote:
1 | 3
                                        sqlite3 supports neither RIGHT
1 | 4
                                        OUTER nor FULL OUTER.
1 | 5
1 | 6
                                        Right outer can be implemented with
2 | 3
                                              SELECT *
2 | 4
                                                FROM B LEFT OUTER JOIN A
2 | 5
                                                     ON A.a=B.b;
2 | 6
3 | 3
                                        Full outer can be implemented with
3 | 4
                                              SELECT *
3 | 5
                                                FROM A LEFT OUTER JOIN B
3 | 6
                                                     ON A.a=B.b
4 | 3
                                               UNION
4 | 4
                                              SELECT *
4 | 5
                                                FROM B LEFT OUTER JOIN A
4 | 6
                                                  ON A.a=B.b;
```

3. 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?

```
SELECT E1.id, E2.id
  FROM Employees AS E1, Employees AS E2
WHERE E1.id < E2.id
    AND E1.bossId = E2.bossId;</pre>
```

Sidenote: The predicate "E1.id < E2.id" could also be written as "E1.id > E2.id". We cannot use plain inequality "E1.id <> E2.id" as the predicate condition because this would lead to duplicate pairs.