Practice Midterm Q2

Group 46
Question 2. (12 pts)

- Consider the following schema.

  CREATE TABLE T (C INT PRIMARY KEY, D INT);
  CREATE TABLE S (B INT PRIMARY KEY, C INT REFERENCES T(C) ON DELETE SET NULL);
  CREATE TABLE R (A INT PRIMARY KEY, B INT REFERENCES S(B) ON DELETE CASCADE);

- Q2a. (4 pts) Suppose the current contents of R, S, and T are as follows.
  
  R(A,B) has {(10,10) and (20,20)}
  S(B,C) has {(10,10) and (20,10)}
  T(C,D) has {(10,10) and (20,10)}

After executing the command: DELETE FROM S;
What tuples will each R and S contain? (For example: no tuples in R, {(10,null),(null,null)} in S)

- Delete from S will delete all of S. Since R references S on delete cascade, all of R will be deleted as well.

- So no tuples in R or S.
SQL

SQL> delete from s;

2 rows deleted.

SQL> select * from r,s;

no rows selected
Q2b

• Consider the following schema.
CREATE TABLE T (C INT PRIMARY KEY, D INT);
CREATE TABLE S (B INT PRIMARY KEY, C INT REFERENCES T(C) ON DELETE SET NULL);
CREATE TABLE R (A INT PRIMARY KEY, B INT REFERENCES S(B) ON DELETE CASCADE);

• Q2b. (4 pts)
Suppose the tables R, S, and T contain r, s, and t tuples respectively. Let n be the number of
tuples in the result of the equi-join query: SELECT * FROM R,S,T WHERE R.B = S.B AND S.C=T.C;
What is the most restrictive (smallest) yet correct upper bond on n?
(A) t    (B) r    (C) min(r,s,t)    (D) r+s+t    (E) r*s*t where * is integer multiplication

• Having the foreign keys apply to primary keys on other tables means that
the largest relation in an equi-join is R. Drawing the ERD will help illustrate this.
• The correct answer is B. r.
Q2b

- The foreign key in S refers to the primary key in T, and these two keys are joined together in the equi-join. One value in T.c may be represented many times in S.c, but the reverse is not true; T.c is a primary key and must be unique. Every value in S.c can appear at most once in T.c. Thus, when the equi-join is performed, rows from T may be duplicated in order to match up with rows from S, which means that “t” is too small of an upper bound. This part of the equi-join will contain at most the number of rows in S, as each row in S can match up to at most one row in T.

- The same principle applies to the foreign key R.b, which applies to the primary key S.b. When the equi-join is performed, rows from S may be duplicated in order to match up with rows from R, so “s” is also too small of an upper bound. This part of the equi-join will contain at most the number of rows in R, as each row in R can match up to at most one row in S. Thus, our final answer is “r”.

SQL

```
SQL> select * from r,s,t
    2  where r.b = s.b
    3  and s.c = t.c;

    A  B  B  C  C  D
    ------- ------- ------- ------- ------- ------- -------
      10  10  10  10  10  10  10
      20  20  20  10  10  10  10
```
Q2c.

- Consider the following schema.
  
  CREATE TABLE T (C INT PRIMARY KEY, D INT);
  CREATE TABLE S (B INT PRIMARY KEY, C INT REFERENCES T(C) ON DELETE SET NULL);
  CREATE TABLE R (A INT PRIMARY KEY, B INT REFERENCES S(B) ON DELETE CASCADE);

- Q2c. (4 pts)

Suppose the tables R, S, and T contain r, s, and t tuples respectively. Let n be the number of tuples in the result of the outer join query: SELECT * FROM (R OUTER JOIN S ON R.B = S.B) OUTER JOIN T ON (S.C = T.C)

What is the most restrictive (smallest) yet correct upper bound on n?

(A) t  (B) r  (C) \text{min}(r,s,t)  (D) r+s+t  (E) r \times s \times t \text{ where } \times \text{ is integer multiplication}

- The outer join will include any unmatched tuples that were excluded from the equi-join in the previous question. This join cannot return anything more than all the tuples in every table, however.

- So the upper bound must be D. r+s+t.
SQL

```sql
SQL> select * from
    2  (r full outer join s
    3  on r.b = s.b)
    4  full outer join t
    5  on s.c = t.c;
```

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>B</th>
<th>C</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>20</td>
<td>20</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>20</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>