IMPORTANT

- You should have 6 non-empty pages, including this cover
- No aids allowed - only calculator and one 8.5 x 11 page (both sides)

For your information:

- 80 minutes duration (1:30-2:50pm)
- Graded out of 100 points
- Numbers in [square brackets] indicate points.
- Please raise your hand, for clarifications.

<table>
<thead>
<tr>
<th>LAST NAME (pls PRINT)</th>
<th>First Name (pls PRINT)</th>
<th>‘andrew’ login (pls PRINT)</th>
</tr>
</thead>
</table>

Contents

Q1. E-R diagrams [15 pts] 2
Q2. Query Languages [40 pts] 2
Q3. Query Languages - SQL [15 pts] 5
Q4. Linear Hashing [20 pts] 5
Q5. Sorting [10 pts] 6

Revision: October 13, 2009
Q1.   **E-R diagrams [15 pts]**

Consider the E-R diagram of Figure 1. Entity set \( E_1 \) has attributes \( E_1(A, b, c) \), with primary key \( A \), and the weak entity set \( E_2 \) has attributes \( E_2(D, e) \) with partial key \( D \). And \( R \) is the identifying relationship.

![E-R diagram](image)

Figure 1: E-R diagram: \( E_2 \) is a weak entity, with \( R \) being the *identifying relationship*

Consider the SQL *create table* commands in Figure 2; modify/complete them, to create a schema that obeys the constraints of the E-R diagram.

1. (2 pts) How many tables do you need?
   
   **ANSWER:** Two, \( E_1 \) and \( E_2 \) (2 min)

2. (13 pts) Give the schema for each one, complete with constraints
   
   **ANSWER:** \( E_1 \) is fine as is; \( E_2 \) has primary key \( (A,D) \); foreign key \( (A) \) references \( E_1 \); on delete cascade; on update cascade; no need for table \( R \)

   To graders: 1 pt for \( E_1 \), 3+3+3+3 for the rest (10 min)

**Specifications:**

- Use *as few tables as possible*
- Try to reject as few operations as possible, i.e., avoid on delete no action
- Do *not* use check statements; but list the constraints that your schema can not satisfy, if any.

Q2.   **Query Languages [40 pts]**

Consider the table \( \text{LIKES} \) in Figure 3, which records information about a social network of who likes whom. Consider the queries below, some in relational algebra, some in relational domain calculus, and some SQL; and also consider the English statements A-M below. What does each query do, in English? Pair up each query with the appropriate statement - if no statement matches, please write down your own.

1. (5 pts)
   
   ```sql
   select idol from LIKES where fan='John'
   ```

   **ANSWER:** \( B \) (1 min)
create table E1 (A char(20), b char(20), c char(20), ...  
    primary key (A) ...  
    foreign key ... references ...  
    ...  
)
create table E2 ( D char(20), e char (20), ...  
    primary key ....  
    foreign key ....  
    on delete cascade  
    on update ...  
    ....  
)
create table R ( ... , ...  
    primary key ...  
    foreign key ...  
    on delete ...  
)
Figure 2: Possibly wrong, half-finished SQL schema

2. ([5 pts])
\[ \pi_{L2.idol} \left[ \sigma_{L1.fan=\text{'John'}}(\rho_{L1(LIKES)} \land_{L1.idol=L2.fan} \rho_{L2(LIKES)}) \right] \]
\text{Answer: None - it finds the second-level idols of 'John', that is, the idols of his idols}  
\hspace{1cm} (2 \text{ minutes})

3. ([5 pts])
\{ <a> | <a, 'John'> \in LIKES \}
\text{Answer: A - fans of John}  
\hspace{1cm} (1 \text{ minute})

4. ([5 pts])
\{ <a> | <a, 'John'> \in LIKES \land <a, 'Peter'> \in LIKES \}
\text{Answer: None - gives people who like both John and Peter}  
\hspace{1cm} (2 \text{ minutes})

5. ([5 pts])
\{ <a> | \forall b (<'John'>, b) \in LIKES \Rightarrow <a, b) \in LIKES \} (4 \text{ minutes})
\text{Answer: 'I': it gives all the people that like the same idols like 'John', and maybe more.}  

6. ([5 pts])
\{ <a, b> | <a, b) \in LIKES \land <b, a) \notin LIKES \}
\text{Answer: None - Description: gives the pairs of people with one-way liking only}  
\hspace{1cm} (2 \text{ minutes})

7. ([5 pts])
select L1.fan, L1.idol  
from LIKES L1, LIKES L2
where L1.fan = L2.idol and
L1.idol = L2.fan

**ANSWER:** M (reciprocated/mutual liking) (3 min)

8. ([5 pts]) Which of the statements A-M can *not* be expressed in relational algebra? List them all, or say “none”

**ANSWER:** I, J: RA can not handle recursion. (3 min)

Here is the list of statements:

- A) all the fans of 'John' (like 'Mary', in Figure 3)
- B) all idols of 'John' (like 'Alice', 'Mary' and 'Peter', in the same Figure)
- C) pairs of fans that have at least one idol in common
- D) all the idols that both 'John' and 'Peter' like
- E) pairs of idols that have at least one fan in common.
- F) all the triangles: that is, triplets of people , like ('Mary' → 'John' → 'Peter' → 'Mary') in Figure 3:
- G) people that dominate 'John' with respect to fans, that is, persons that have all the fans of 'John', and maybe more.
- H) people that are 'fan'-dominated by 'John', that is, persons that have a subset of the fans of 'John' (or exactly the same fans).
- I) people that 'idol'-dominate 'John', that is, they have all the idols of 'John', and maybe more.
- J) 'like-minded': all the people that have exactly the same idols as 'John'
- K) all the followers of 'John', where we define as John’s “followers” the people who like him, and their fans, and their fans of fans, and so on.
- L) all the people that have 'John' as a follower (= fan, or fan of fan, etc)
- M) all pairs of reciprocated liking, that is, all pairs that mutually like each other, such as 'Mary' and 'John' in Figure 3.

<table>
<thead>
<tr>
<th>LIKEs</th>
<th>fan</th>
<th>idol</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mary</td>
<td>Alice</td>
<td></td>
</tr>
<tr>
<td>Mary</td>
<td>John</td>
<td></td>
</tr>
<tr>
<td>John</td>
<td>Alice</td>
<td></td>
</tr>
<tr>
<td>John</td>
<td>Mary</td>
<td></td>
</tr>
<tr>
<td>John</td>
<td>Peter</td>
<td></td>
</tr>
<tr>
<td>Peter</td>
<td>Mary</td>
<td></td>
</tr>
<tr>
<td>Sam</td>
<td>Winnie</td>
<td></td>
</tr>
<tr>
<td>Tom</td>
<td>Winnie</td>
<td></td>
</tr>
</tbody>
</table>

... ... ...

**Figure 3:** Table 'LIKEs', recording who-likes-whom
Q3. Query Languages - SQL [15 pts]

Again consider the table LIKES in Figure 3, and assume that there are no other tuples in it except for the ones shown. For each of the following queries, give the output results, when applied on the instance of the LIKES table.

1. ([5 pts])
   select idol, count(*)
   from LIKES
   group by idol
   order by idol asc
   ANSWER: (Alice, 2) , (John, 1), (Mary, 2), (Peter, 1), (Winnie, 2) (3 min)

2. ([10 pts])
   select L1.fan, L2.fan, L1.idol
   from LIKES L1, LIKES L2
   where L1.idol = L2.idol
   and L1.fan < L2.fan
   order by L1.fan asc, L2.fan asc, L1.idol asc
   ANSWER: (John, Mary, Alice), (John, Peter, Mary) , (Sam, Tom, Winnie) (3 min)

Q4. Linear Hashing [20 pts]

Consider a hash table that operates under linear hashing. When it started, the initial hashing function was \( h(x) = x \mod 4 \); the hash table had \( B=4 \) buckets (0,1,2,3), and the split pointer was \( s=0 \).

1. After some splits, the table has \( N=6 \) buckets, numbered 0,1,...,5.
   • [4 pts] Where is the split pointer \( s \)?
     ANSWER: \( s=2 \)
   • [3 pts] How many hashing functions are active?
     ANSWER: 2
   • [3 pts] Which one(s)?
     ANSWER: mod4, and mod8

2. After some other splits, the table has \( N=32 \) buckets (numbered 0,1,..., 31)
   • [4 pts] Where is the split pointer \( s \)?
     ANSWER: 0 - we have completed an expansion
   • [3 pts] How many hashing functions are active?
     ANSWER: one only
   • [3 pts] Which one(s)?
     ANSWER: \( h(x) = x \mod 32 \)
Q5. Sorting [10 pts]

We have $B=200$ buffers in memory, and we want to use them in external merge sort, to sort files. Let $P$ be the number of pages in the target file.

1. [2 pts] How many pages $P$ are in the largest file we can sort with one pass?
   
   \begin{align*}
   \text{ANSWER: } P &= B = 200 \text{ (1 min)}
   \end{align*}

2. [3 pts] With 2 passes?
   
   \begin{align*}
   \text{ANSWER: } B \cdot (B - 1) &= 200 \cdot 199 = 39,800 \text{ (2 min)}
   \end{align*}

3. [5 pts] With 3 passes?
   
   \begin{align*}
   \text{ANSWER: } B \cdot (B - 1) \cdot (B - 1) &= 7,920,200 \text{ (2 min)}
   \end{align*}

Again, we want a single number each time. Explanations are optional, and will only be used to your benefit, that is, to give partial credit, if the numerical answer is incorrect.

END OF EXAM QUESTIONS

GOOD LUCK!

ANSWER:

\begin{align*}
\text{Total number of points: } 100; \quad \text{section weights: } 200 \quad \text{total count of minutes: } 41
\end{align*}