B561
Final Exam

Thursday, December 16, 2:45-4:45 pm

This exam comprises 3 pages. Ensure you hand in answers to **9** questions.

Part I: Query Formulation

Recall the schema for the database maintained by the *BeeSure* honey farm:

WORKS_IN(*bee*, *hive*)
HEAD_OF(*queen*, *hive*)

You may wish to abbreviate WORKS_IN by W and HEAD_OF by H in order to save writing.

1. Formulate the following queries in the *Relational Algebra* (use only the 5 basic operations, plus $\bowtie$ if you wish):
   (a) Find those queens that head hive 'h1' or 'h2'.
   (b) Find those bees working in hives headed by the queen bee 'Isabella'.
   (c) Find those queen bees that head exactly one hive.
2. Formulate the following queries in the *Tuple Calculus*:
   (a) Find those queens that head hives that 'b1' works in.
   (b) Find those bees working in hives headed by the queen bee 'Sheba' who do *not* work alongside the bee 'b2'.
3. Formulate the following queries in SQL. Conform to *either* the SQL '92 or the Oracle syntax (but not a mixture of both).
   (a) Find those bees working in hives 'h1' or 'h2'.
   (b) Find those queen bees that supervise hives having exactly 5 worker bees. You may wish to use aggregate functions.
   (c) Find those queen bees that supervise the most worker bees (in terms of the sum total of *distinct* bees in hives that they head). You may wish to use aggregate functions.
4. Translate your tuple calculus query 2(b) (above) into the RA, *using the translation algorithm outlined in class.* For **bonus points**, you may wish to optimize your result RA expression, using the optimization technique given in class.

Part II: Query Optimization

5. Transform the following SQL query into an *optimized* RA expression, using the method given in class. Assume the context of the *BeeSure* database (as in part I).

```
SELECT H1.queen, H2.queen
FROM   HEAD_OF H1, HEAD_OF H2, WORKS_IN W1
WHERE  W1.hive = H1.hive AND W1.hive = H2.hive AND W1.bee = ‘b1’
       AND NOT EXISTS ( SELECT *
                        FROM WORKS_IN W
                        WHERE W.hive=H1.hive AND W.hive=H2.hive
                              AND W.bee = ‘b2’)
```

   Hint: You will need to transform the `NOT EXISTS` clause into an appropriate set difference expression in your initial RA expression.

6. Consider the relation schemas $R(A, B, C)$, $S(A, B)$, and $T(B, C)$.
   (a) Show that $R \cap (S \bowtie T) \subseteq (\Pi_{A,B}(R) \cap S) \bowtie (\Pi_{B,C}(R) \cap T)$.
   (b) Given the additional information that the functional dependency $AB \rightarrow C$ holds in R, show that $R \cap (S \bowtie T) = (\Pi_{A,B}(R) \cap S) \bowtie (\Pi_{B,C}(R) \cap T)$.

Part III: Query Processing

7. Show that the complexity of performing a merge-sort join of relations R and S is $O(|R|log(|R|) + |S|log(|S|) + |R \bowtie S|)$.

Part IV: Concurrency Control & Recovery

8. The following four terms describe properties of schedules.

serial	conflict serializable (as in class)
recoverable	strict

   (a) Rank the four properties in terms of *strength*. A property P_1 is *stronger* than a property P_2 in case the fact that schedule S satisfies P_1 implies schedule S also satisfies P_2 (in symbols, $P_1 \Rightarrow P_2$).

(b) A schedule, S_1 is given below. State which of the four properties hold of S_1. For only the *strongest* such properties, argue *why* they hold of S_1.

Time	T_1	T_2
1	read(x)	
2	read(y)	
3		read(y)
4	write(x)	
5		write(x)
6	write(y)	
7	commit	
8		commit

(c) A schedule, S_2 is given below. Is S_2 serializable? Argue why or why not.

Time	T_1	T_2	T_3	T_4
1		rlock(x)		
2			rlock(x)	
3		wlock(y)		
4		unlock(x)		
5			wlock(x)	
6		unlock(y)		
7	rlock(y)			
8			unlock(x)	
9				rlock(y)
10	rlock(x)			
11				unlock(y)
12	wlock(z)			
13	unlock(x)			
14				wlock(x)
15				unlock(x)
16	unlock(y)			
17	unlock(z)			

9. Two transactions are *not interleaved* in a schedule if every operation of one transaction precedes every operation of the other. Prove that every schedule, S, having the following property is serializable: If $p_i, q_j \in S$ and p_i, q_j are conflicting operations of transactions T_i and T_j (respectively), then T_i and T_j are not interleaved in S.