CS145 Final Examination

Spring 2000, Prof. Widom

- Please read all instructions (including these) carefully.
- There are 13 problems on the exam, with a varying number of points for each problem and subproblem for a total of 120 points. You have 120 minutes to complete the exam. *You should look through the entire exam before getting started, in order to plan your strategy.*
- The exam is closed book and closed notes, but you may refer to your three pages of prepared notes.
- Please write your solutions in the spaces provided on the exam. Make sure your solutions are neat and clearly marked. You may use the blank areas and backs of the exam pages for scratch work. Please do not use any additional scratch paper.
- *Simplicity and clarity of solutions will count.* You may get as few as 0 points for a problem if your solution is far more complicated than necessary, or if we cannot understand your solution.

NAME: ______________________________

In accordance with both the letter and spirit of the Honor Code, I have neither given nor received assistance on this examination.

SIGNATURE: ______________________________

Problem	1	2	3	4	5	6	7	8	9	10	11	12	13	TOTAL
Max. points	5	12	9	8	8	12	14	6	12	6	9	10	9	120
Points														

1. **SQL** (5 points)

   Consider a relation `R(A integer)` and the following SQL query over `R`:

   ```
   select sum(B)
   from   R, (select distinct A, 1 as B from R) as R2
   where  R.A = R2.A and R2.A = 5
   ```

   In one sentence, state in English what this query computes. The correct answer is simple and brief.

2. **Transactions** (12 points: 3 per part)

Consider a relation `Employee(ID,salary)`, where `ID` is the key, and consider the following two concurrent transactions over this relation:

```
T1: begin transaction
    update Employee set salary = 2 * salary where ID = 25;
    update Employee set salary = 3 * salary where ID = 25;
    commit

T2: begin transaction
    update Employee set salary = salary - 20;
    commit
```

Suppose the salary of the employee with ID=25 is 100 before either transaction executes. Assume that each transaction executes to completion exactly one time, although it may need to rollback and restart one or more times in order to do so.

For each of the following scenarios, specify all possible final values for the salary of the employee with ID=25.

(a) Both transactions execute with isolation level *serializable*.

Possible final values:

(b) Both transactions execute with isolation level *repeatable read*.

Possible final values:

(c) Both transactions execute with isolation level *read committed*.

Possible final values:

(d) Both transactions execute with isolation level *read uncommitted*.

Possible final values:

3. **Authorization** (9 points)

Consider the following relations in a SQL database:

```
Advisor(name, office)            // name is a key
Student(name, GPA, advisorName)  // name is a key
```

Suppose the owner (creator) of these relations is a user named `Peche`, and `Peche` wants to grant to a user named `Marianne` the ability to read all information about all students whose advisor's office contains the string "Gates 4A" (and only those students). Specify a command or sequence of commands that achieves this goal.

4. **Recursion** (8 points)

Consider a relation `Edge(N1 integer, N2 integer)`, where a tuple (n_1,n_2) in `Edge` indicates that there is an edge from node number n_1 to node number n_2 in a directed graph. Consider the following `with` statement in SQL3.

```
with recursive Mystery as
  ( (select * from Edge)
    union
    (select Mystery.N1, Edge.N2
     from Mystery, Edge
     where Mystery.N2 = Edge.N1) )
select N1
from Mystery
where N1 = N2
```

In one sentence, state in English what is computed by the above `with` statement. The correct answer is simple and brief.

5. **Constraints** (8 points)

Consider a relation `R(A integer, B integer)` and the following constraint over `R`:

```
CREATE ASSERTION Secret CHECK (
   NOT EXISTS(select * FROM R
              WHERE A > (select AVG(B) FROM R)))
```

Can this general assertion be rewritten as one or more attribute-level or tuple-level constraints on `R`? Circle One: YES NO

If you circled YES, write the constraint(s). If you circled NO, briefly explain why not.

6. **Triggers** (12 points: 6 per part)

Consider a relation `Employee(ID,salary)`, where `ID` is the key. Suppose the following trigger is defined on this relation:

```
create trigger AutoRaise
after insert on Employee
update Employee
  set salary = salary + (select min(salary) from Employee)
```

Suppose that before the trigger is created the relation contains one tuple: (1, 50). Then after trigger creation three new tuples are inserted as the result of one statement: (2, 70), (3, 30), (4, 70).

(a) Assume the trigger above is specified as "`for each row`." After all three insertions, what are the final salaries for the four employees? Fill them in:

ID	salary
1	
2	
3	
4	

(b) Now assume the trigger above is specified as "`for each statement`." After all three insertions, what are the final salaries for the four employees? Fill them in:

ID	salary
1	
2	
3	
4	

7. **ODL and OQL** (14 points, division below)

Consider the following ODL class declarations:

```
interface Patient {
  extent Patients;
  unique attribute string name;
  attribute Struct<string street, string city> info;
  relationship Doctor doctor
    inverse Doctor::patient; }

interface Doctor {
  extent Doctors;
  unique attribute string name;
  attribute string city;
  relationship Set<Patient> patient
    inverse Patient::doctor; }
```

In part (a) you will receive 3 points for a correct answer, -2 points for an incorrect answer, and 0 points for no answer.

(a) (3 points) Which of the following describes the relationship from patients to doctors (circle one)? One-One One-Many Many-One Many-Many

(b) (3 points) Suppose the `Doctor` class is translated to a relation in the straightforward way:

```
Doctor(name, city, patient-name)
```

State in three words or less what is wrong with this relational design:

(c) (8 points) Write an OQL query that returns the name of every doctor such that all of the doctor's patients live in the same city where the doctor practices.

8. **Object-Relational SQL** (6 points: 3 per part)

Suppose you have created a new object type `Preference` in your Oracle 8 object-relational database schema, which may be a complicated type with several attributes. You decide to implement an *order method* for the object type. An order method enables you to write SQL queries that perform comparison operations (<, >, =, etc.) over `Preference` objects. State two additional features of SQL queries that are enabled by implementing an order method. That is, name two types of operations other than comparisons that you can perform on `Preference` objects or sets of `Preference` objects via SQL once an order method is defined.

Feature 1:

Feature 2:

9. **SQL and Association Rules** (12 points: 4 per part)

Consider a relation `History(saleID,itemID)`, which records "market basket" sales data as discussed in class and in your last assignment with respect to data mining.

(a) Using SQL, create a view that contains, for each `itemID`, the number of sales in which that item was purchased.

(b) Using SQL, create a second view that contains, for each pair of `itemID`'s $\langle I_1, I_2 \rangle$, the number of sales in which both items were purchased.

(c) Use your views from parts (a) and (b) to write a SQL query that finds all pairs of `itemID`'s $\langle I_1, I_2 \rangle$ such that $I_1 \rightarrow I_2$ is an association rule with *confidence* > 0.6.

10. **Data Warehousing** (6 points)

Consider a more detailed sales history relation than the one in Problem 9:

```
Sales(saleID,itemID,color,size,qty,unitPrice)
```

Suppose a user issues the following three queries, in order:

```
Query1:
  select    itemID, color, size, sum(qty*unitPrice)
  from      Sales
  group by  itemID, color, size

Query2:
  select    itemID, size, sum(qty*unitPrice)
  from      Sales
  group by  itemID, size

Query3:
  select    itemID, size, sum(qty*unitPrice)
  from      Sales
  WHERE     size < 10
  group by  itemID, size
```

In the following two problem parts, you will receive 3 points for a correct answer, -2 points for an incorrect answer, and 0 points for no answer.

(a) In moving from `Query1` to `Query2`, the user is performing a (circle one):

- drill down
- roll up
- slice
- dice
- none of the above

(b) In moving from `Query2` to `Query3`, the user is performing a (circle one):

- drill down
- roll up
- slice
- dice
- none of the above

11. **XML** (9 points: 3 per part)

Consider the following DTD for XML documents:

```
<!DOCTYPE Univ [
   <!ELEMENT UNIVERSITY (PROF*, COURSE*)>
   <!ELEMENT PROF (NAME, OFFICE+)>
   <!ATTLIST PROF Ident ID Courses IDREFS>
   <!ELEMENT NAME (#PCDATA)>
   <!ELEMENT OFFICE (#PCDATA)>
   <!ELEMENT COURSE (DEPT, (BUILDING | ROOM), TIME?)>
   <!ATTLIST COURSE Number ID Teacher IDREF>
   <!ELEMENT DEPT (#PCDATA)>
   <!ELEMENT BUILDING (#PCDATA)>
   <!ELEMENT ROOM (#PCDATA)>
   <!ELEMENT TIME (#PCDATA)>
]>
```

For each of the following three XML documents, state whether or not the document conforms to the DTD. You will receive 3 points for a correct answer, -2 points for an incorrect answer, and 0 points for no answer.

(a) The following document (circle one): Conforms Does not conform

```
<?XML VERSION="1.0" STANDALONE="no"?>
<UNIVERSITY>
   <PROF Ident="JW" Courses="CS145 CS346">
      <NAME>Widom</NAME>
      <OFFICE>Gates 422</OFFICE>
   </PROF>
   <COURSE Number="CS145" Teacher="JW">
      <DEPT>Computer Science</DEPT>
      <BUILDING>Gates</BUILDING>
   </COURSE>
   <COURSE Number="CS346" Teacher="JW">
      <DEPT>Computer Science</DEPT>
      <ROOM>B08</ROOM>
      <TIME>11:00</TIME>
   </COURSE>
</UNIVERSITY>
```

(b) The following document (circle one): Conforms Does not conform

```
<?XML VERSION="1.0" STANDALONE="no"?>
<UNIVERSITY>
   <PROF Ident="JW" Courses="CS145">
      <NAME>Widom</NAME>
      <OFFICE>Gates 422</OFFICE>
      <OFFICE>Gates 436</OFFICE>
   </PROF>
   <COURSE Number="CS145" Teacher="CS346">
      <DEPT>Computer Science</DEPT>
      <BUILDING>Gates</BUILDING>
   </COURSE>
   <COURSE Number="CS346" Teacher="JW">
      <DEPT>Computer Science</DEPT>
      <ROOM>B08</ROOM>
      <TIME>11:00</TIME>
   </COURSE>
</UNIVERSITY>
```

(c) The following document (circle one): Conforms Does not conform

```
<?XML VERSION="1.0" STANDALONE="no"?>
<UNIVERSITY>
   <PROF Ident="JW" Courses="CS145 CS346">
      <NAME>Widom</NAME>
      <OFFICE>Gates 422</OFFICE>
      <OFFICE>Gates 436</OFFICE>
   </PROF>
   <COURSE Number="CS145" Teacher="JW">
      <BUILDING>Gates</BUILDING>
      <DEPT>Computer Science</DEPT>
   </COURSE>
   <COURSE Number="CS346" Teacher="JW">
      <DEPT>Computer Science</DEPT>
      <BUILDING>Gates</BUILDING>
      <ROOM>B08</ROOM>
      <TIME>11:00</TIME>
   </COURSE>
</UNIVERSITY>
```

12. **Temporal Relational Algebra** (10 points: 5 per part)

Consider a temporal relation `Borrows(student,book,time)`, which records information about students borrowing books from a library. A student may borrow multiple books, and a student may borrow a single book multiple times. Because the special `time` attribute contains sets of intervals as discussed in class, attribute pair `<student,book>` forms a key for this relation. For each of the following two expressions of temporal relational algebra, state in English what the expression computes. In both cases the correct answer is simple and brief.

(a) $\Pi_{\texttt{student}}(\texttt{Borrows}) -$
$\Pi_{B1.s}(\sigma_{B1.s=B2.s \ \wedge \ B1.b \neq B2.b}(\rho_{B1(s,b,t)}(\texttt{Borrows}) \times \rho_{B2(s,b,t)}(\texttt{Borrows}))$

(b) $\Pi_{\texttt{end(time)}}(\sigma_{\texttt{student="John"}}(\Pi_{\texttt{student,time}}(\texttt{Borrows}))$

13. **Course Readings and Short Answer** (9 points: 3 per part)

In each of the following problem parts you will receive 3 points for a correct answer, -2 points for an incorrect answer, and 0 points for no answer.

(a) In the reading material for the lecture on database-backed web sites (two chapters from *Philip and Alex's Guide to Web Publishing*), the authors motivated concurrency control by describing the purchase of which of the following items? Circle one:

- A dog
- A new computer
- A new stove
- A rhinoceros

(b) In the reading material for the lecture on XML (*XML in 10 points*), how many points where there really? Circle one:

- 5
- 7
- 9
- 10

(c) In a highly tuned database-backed web site, *connection pooling* is performed in which of the following components? Circle one:

- Database server
- Application server
- Web server
- Client