CSE-3421M Test #2

"Queries"

Sur / Last Name:
Given / First Name:
Student ID:

- **Instructor:** Parke Godfrey
- **Exam Duration:** 75 minutes
- **Term:** Winter 2011

Answer the following questions to the best of your knowledge. Your answers may be brief, but be precise and be careful. The exam is closed-book and closed-notes. Calculators, etc., are fine to use. Write any assumptions you need to make along with your answers, whenever necessary.

There are four major questions, each with parts. Points for each question and sub-question are as indicated. In total, the test is out of 50 points.

In schemas, the underlined attributes denote a table's key. Attributes that are in *italics* are not nullable. Foreign keys are indicated by FK.

If you need additional space for an answer, just indicate clearly where you are continuing.

Marking Box	
1.	/15
2.	/10
3.	/10
4.	/15
Total	/50

1. (15 points) **SQL.** *Name that query!* Exercise

Person(p#, *name*)
Journal(name, *publisher*)
Issue(name, volume, *month*, *year*)
  FK (name) refs **Journal**
Paper(title, name, volume, abstract, *#pages*)
  FK (name, volume) refs **Issue**
Institution(name, *address*)
Affiliation(p#, name, from, until)
  FK (p#) refs **Person**
  FK (name) refs **Institution**
Author(p#, title, name, volume)
  FK (p#) refs **Person**
  FK (title, name, volume) refs **Paper**
Topic(topic)
Coverage(name, topic)
  FK (name) refs **Journal**
  FK (topic) refs **Topic**
Keyword(title, name, volume, topic)
  FK (title, name, volume) refs **Paper**
  FK (topic) refs **Topic**
Editor(p#, name, from, until)
  FK (p#) refs **Person**
  FK (name) refs **Journal**

Figure 1: Journal Schema.

The basic schema of a database for tracking academic papers published in journals is shown in Figure 1.

Several **Authors** together can write a **Paper**. The paper appears in an **Issue** of a **Journal**. A **Journal** can specialize in certain **Topics**, as indicated by **Coverage**. A **Paper** may be on certain **Topics**, as indicated by **Keyword**.

An **Author** is affiliated with an **Institution** (e.g., York) via **Affiliation**. This **Affiliation** can change over time (people change jobs), and the from and until fields indicate the duration. The from and until fields in both cases indicate a *year* (e.g., from = 1997). A NULL in until indicates the person is presently affiliated with the institution.

Additional implicit constraints on the database are that a **Journal** should have an **Editor**, and a paper should have at least one author. For **Editor**, the from and until fields are just as for **Affiliation**, and indicate *when* this person was **Editor** for the **Journal**.

Write SQL queries with respect to the Journal Schema in Figure 1.

Keep your queries *as simple as possible.* Do not employ any table in your query if it is not required. Do not use nested queries, if not necessary. (Use of the *with* clause is fine, however.)

a. (5 points) List all persons who have published a paper in a journal with "Databases" in the journal's coverage after the year 2000. Do not list a person twice.

```
select distinct P.p#, P.name
    from Person P, Author A, Issue I, Coverage C
    where P.p# = A.p#
        and A.name = I.name and A.volume = I.volume
        and I.name = C.name and 'Databases' = C.topic
        and I.year > 2000;
```

---

b. (5 points) List each person, the journal name, and the year such that the person published a paper in that journal in that year while he or she was the editor of that journal (so during that same year).

```
select distinct P.p#, P.name as pname, I.name as jname, I.year
    from Person P, Author A, Editor E, Issue I
    where P.p# = A.p#
        and P.p# = E.p# and E.name = I.name
        and A.name = I.name and A.volume = I.volume
        and E.from <= I.year
            and (E.until >= I.year or E.until is null);
```

c. (5 points) Consider a paper to be *affiliated* with an institution *if* one of the paper's authors was affiliated with the institution in the year that the paper was published.

List the institution by name with the most papers affiliated with it, along with its number of papers as total. (In the case of a tie for most, list all that tie.)

```
with
    InstAffil (name, title, journal, volume) as (
        select distinct A.name, J.title, J.name, J.volume
            from Issue J, Affiliation A, Author W
            where W.p# = A.p#
                and W.name = J.name and W.volume = J.volume
                and A.from <= J.year
                    and (A.until >= J.year or A.until is null)
    ),
    PubCount (name, total) as (
        select I.name, count(*)
            from InstAffil
    )
select I.name, I.total
    from PubCount I
    where I.total = (
            select max(total)
                from PubCount
        );
```

2. (10 points) **General.** *I rushed* $\Pi \bowtie \Sigma$. Multiple Choice

Choose *one* best answer for each of the following. Each is worth one point. There is no negative penalty for a wrong answer.

---

a. In a real relational database system, if you try to join (natural join) tables **R** and **S** *and* **R** is empty (that is, it has no tuples),
   A. the system reports an error.
   [B.] the answer set is an empty table.
   C. the answer set is the same as table **S**.
   D. the answer set consists of just one row.
   E. an answer set is returned; however, the results are system dependent.

---

b. The SQL statement "delete from R;"
   A. is guaranteed to remove all the tuples from **R**.
   [B.] may also remove tuples in tables other than **R**.
   C. may remove just some tuples from **R**.
   D. will drop table **R** from the database.
   E. will do nothing because it is missing a where clause.

---

c. In relational algebra, the join operator ($\bowtie$) is logically redundant if we have additionally
   A. intersection ($\cap$).
   [B.] crossproduct ($\times$), select (σ), and project (π).
   C. difference ($-$) and union ($\cup$).
   D. crossproduct ($\times$) and difference ($-$).
   E. crossproduct ($\times$) and union ($\cup$).

---

d. Consider the following two properties.

   I. a lossless join decomposition
   II. dependency preserving

   For any schema,
   A. there is always a BCNF refinement that is both **I** and **II**.
   [B.] there is always a BCNF refinement that is **I**, but not necessarily **II**.
   C. there is always a BCNF refinement that is **II**, but not necessarily **I**.
   D. there is never a BCNF refinement that is both **I** and **II**.
   E. there is never a BCNF refinement that is **I** or **II**.

---

e. Which is the most *expressive* query language?
   That is, are certain queries only possible to state in one of the following, but cannot be stated in the others?
   A. relational algebra
   B. domain relational calculus
   C. tuple relational calculus
   D. SQL without the aggregate operators or recursion
   [E.] They are all equally expressive.

R

A	B
1	2
3	2
5	6
7	8
9	8

S

B	C
6	2
2	4
8	1
8	3
2	5

T

A	C
7	1
1	2
9	3
5	4
3	5

Three tables: **R**, **S**, & **T**.

I

A	B	C
1	2	4
1	2	5
3	2	4
3	2	5
5	6	2
7	8	1
7	8	3
9	8	1
9	8	3

II

A	B	C
1	2	2
3	2	5
5	6	4
7	8	1
9	8	3

III

A	B	C
1	6	2
3	2	5
5	2	4
7	8	1
9	8	3

IV

A	B	C
3	2	5
7	8	1
9	8	3

V

A	B	C

Possible answer tables.

All the join operations below are natural joins.

f. What is the resulting table of $\mathsf{R} \bowtie \mathsf{S}$?

[A.] I **B. II** **C. III** **D. IV** **E. V**

g. What is the resulting table of $(\mathsf{R} \bowtie \mathsf{S}) \bowtie \mathsf{T}$?

A. I **B. II** **C. III** **[D.] IV** **E. V**

h. What is the resulting table of $\mathsf{R} \bowtie (\mathsf{S} \bowtie \mathsf{T})$?

A. I **B. II** **C. III** **[D.] IV** **E. V**

i. What is the resulting table of $\pi_{\mathsf{A,B}}(\mathsf{R} \bowtie \mathsf{S}) \bowtie \pi_{\mathsf{A,C}}(\mathsf{S} \bowtie \mathsf{T})$?

A. I **[B.] II** **C. III** **D. IV** **E. V**

j. What is the resulting table of $\pi_{\mathsf{A,B}}(\mathsf{R} \bowtie \mathsf{T}) \bowtie \pi_{\mathsf{B,C}}(\mathsf{S} \bowtie \mathsf{T})$?

[A.] I **B. II** **C. III** **D. IV** **E. V**

3. (10 points) **Relational Algebra & Calculus.** *The area under what?!* [Multiple Choice]

Choose *one* best answer for each of the following. Each is worth two points. There is no negative penalty for a wrong answer.

For Questions 3a and 3b, consider the following schema.

$\mathbf{R}(\underline{A}, B)$ FK (B) refs $\mathbf{S}$

$\mathbf{S}(A, \underline{B})$ FK (A) refs $\mathbf{R}$

(None of the attributes is nullable.)

a. (2 points) Which of the following is guaranteed to produce as many as, or more, tuples than each of the others?

A. $\mathbf{R} \bowtie \mathbf{S}$
[B.] $\mathbf{R} \times \mathbf{S}$
C. $\mathbf{R} \cap \mathbf{S}$
D. $\mathbf{R} \cup \mathbf{S}$
E. There is not enough information to answer this.

b. (2 points) Which of the following is guaranteed to produce as many as, or more, tuples than each of the others?

A. $\mathbf{R} \bowtie \mathbf{S}$
B. $\pi_{\mathsf{A}}(\mathbf{R}) \bowtie \mathbf{S}$
C. $\mathbf{R} \bowtie \pi_{\mathsf{B}}(\mathbf{S})$
[D.] $\pi_{\mathsf{A}}(\mathbf{R}) \bowtie \pi_{\mathsf{B}}(\mathbf{S})$
E. There is not enough information to answer this.

c. (2 points) Consider the schema $\mathbf{R}(\mathsf{A}, \mathsf{B})$, $\mathbf{S}(\mathsf{B}, \mathsf{C})$, and $\mathbf{T}(\mathsf{C}, \mathsf{A})$ (with no FKs).

One of these things is not like the other. That is, one of them may evaluate differently than the others. Which one?

A. $\{\langle \mathsf{A} \rangle \mid \forall \mathsf{B}(\langle \mathsf{A}, \mathsf{B} \rangle \in \mathbf{R} \wedge \forall \mathsf{C}(\langle \mathsf{B}, \mathsf{C} \rangle \in \mathbf{S} \wedge \forall \mathsf{A2}(\langle \mathsf{C}, \mathsf{A2} \rangle \in \mathbf{T} \rightarrow \mathsf{A} \neq \mathsf{A2})))\}$

[B.] $\{\langle \mathsf{A} \rangle \mid \forall \mathsf{C}(\langle \mathsf{C}, \mathsf{A} \rangle \in \mathbf{T} \wedge \forall \mathsf{B}(\langle \mathsf{B}, \mathsf{C} \rangle \in \mathbf{S} \wedge \forall \mathsf{A2}(\langle \mathsf{A2}, \mathsf{B} \rangle \in \mathbf{R} \rightarrow \mathsf{A} \neq \mathsf{A2})))\}$

C. $\{\langle \mathsf{A} \rangle \mid \forall \mathsf{B}(\langle \mathsf{A}, \mathsf{B} \rangle \in \mathbf{R} \rightarrow \forall \mathsf{C}(\langle \mathsf{C}, \mathsf{A} \rangle \notin \mathbf{T} \vee \langle \mathsf{B}, \mathsf{C} \rangle \notin \mathbf{S}))\}$

D. $\{\langle \mathsf{A} \rangle \mid \neg \exists B(\langle \mathsf{A}, \mathsf{B} \rangle \in \mathbf{T} \wedge \exists C(\langle \mathsf{B}, \mathsf{C} \rangle \in \mathbf{S} \wedge \langle \mathsf{C}, \mathsf{A} \rangle \in \mathbf{T}))\}$

E. $\{\langle \mathsf{A} \rangle \mid \neg \exists B, C(\langle \mathsf{A}, \mathsf{B} \rangle \in \mathbf{T} \wedge \langle \mathsf{B}, \mathsf{C} \rangle \in \mathbf{S} \wedge \langle \mathsf{C}, \mathsf{A} \rangle \in \mathbf{T})\}$

---

d. (2 points) Consider the relations $\mathbf{R}(A, B)$, $\mathbf{S}(B, C)$, and $\mathbf{T}(C, D)$.

One of these is not like the others. That is, one can evaluate differently than the other four. Which one?

A. $\pi_{A,D}((\mathbf{R} \bowtie \mathbf{S}) \bowtie \mathbf{T})$
B. $\pi_{A,D}((\mathbf{R} \times \mathbf{T}) \bowtie \mathbf{S})$
C. $\pi_{A,D}((\mathbf{R} \bowtie \mathbf{S}) \bowtie (\mathbf{S} \bowtie \mathbf{T}))$
[D.] $\pi_{A,D}(\pi_{A,B}(\mathbf{R} \bowtie \mathbf{S}) \bowtie \pi_{B,D}(\mathbf{S} \bowtie \mathbf{T}))$
E. $\pi_{A,D}((\mathbf{R} \times \mathbf{T}) \cap (\pi_{A}(\mathbf{R}) \times (\mathbf{S} \times \pi_{D}(\mathbf{T}))))$

---

e. (2 points) Consider the relation **Enrol** with attributes sid, cid, term, and grade which stores academic records of students. Attribute sid is a student identifier and cid is a class—a given section of a course in a given term—identifier.

Here is a query involving **Enrol**:

```
select distinct cid
    from ( select * from Enrol E1
                where not exists
                    (select *
                        from Enrol E2
                        where E2.cid = E1.cid
                              and E2.grade > E1.grade)
           ) as V
    where grade = 8;
```

Which of the following queries must return the same result as the query above?

I.
```
select distinct E1.cid
    from Enrol E1, Enrol E2
    where E1.grade = 8
          and E2.grade <= E1.grade
          and E1.sid <> E2.sid;
```

II.
```
select distinct cid
    from Enrol
    group by cid
    having max(grade) = 8;
```

A. **I** only.
[B.] **II** only.
C. Both **I** and **II**.
D. Neither **I** nor **II**.
E. There is not enough information available to determine this.

4. (15 points) **Jeopardy.** *SQL for a hundred, Alex.* Analysis

Consider the schema in Figure 1 from Question 1 again.

State in *plain, concise English* what the following query does, as Questions 1a to 1c do.

You get *zero* credit if you use database terms in your answer! (E.g., "Well, the query first *joins* two tables, taking the *projection* of..." does not count!)

a. (4 points)

```
select distinct A.name, K.topic
    from Affiliation A, Author W, Keyword K, Issue I
    where A.p# = W.p#
        and W.title = K.title and W.name = K.name and W.volume = K.volume
        and W.name = I.name and W.volume = I.volume
        and A.from <= I.year
            and (A.until is null or A.until >= I.year);
```

List each institution with the topics on which people at that institution have written papers.

b. (4 points)

```
select distinct P.p#, P.name as author, J.name as journal
    from Person P, Author A, Journal J
    where P.p# = A.p#
        and A.name = J.name
        and not exists (
            select C.topic
                from Coverage C
                where J.name = C.name
            intersect
            select K.topic
                from Keyword K
                where A.title = K.title and A.name = K.name
                    and A.volume = K.volume
        );
```

List each author with the journals in which he or she has published a paper that covers no topics that the journal does.

Consider the table **R** with attributes A, B, and C. The following functional dependencies hold on **R**:

$$A \mapsto B$$
$$C \mapsto AB$$

c. (2 points) Is the decomposition of **R** into CA and AB lossless or lossy?
Justify *in brief* your answer.

Yes. A is a key for AB, so CA and AB can be joined losslessly.

d. (5 points) The decomposition of **R** into AB and BC is lossy.
Demonstrate that this is lossy by example.
(Construct a small table for **R**, and show its projection onto AB and BC, which demonstrates this.)

```
A B C     A B     B C
----- => --- &  ---
1 2 3     1 2     2 3
3 2 1     3 2     2 1
```

But ⋈ the two, AB and BC, we get

```
A B C
-----
1 2 3
1 2 1
3 2 3
3 2 1
```

Hence, lossy!

Extra space.

EXTRA SPACE.

RELAX. BREATHE. TURN IN YOUR TEST. GO HOME.