EECS-3421A: TEST #2

Electrical Engineering & Computer Science

York University

Family Name: ______________________
Given Name: ______________________
Student#: ______________________
CSE Account: ______________________

- **Instructor:** Parke Godfrey
- **Exam Duration:** 75 minutes
- **Term:** Fall 2015

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 five 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 part of a relation's key are not nullable.) 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.	/10
2.	/10
3.	/10
4.	/10
5.	/10
Total	/50

1. (10 points) **Normalization.** *Whose normal?!* [Analysis]

For Questions 1a–1c, consider the relation **R** with attributes A, B, C, D, and E, and with the following set, $\mathcal{F}$, of functional dependencies:

$$\begin{array}{ll} \mathsf{AC} \mapsto \mathsf{B} & \mathsf{BC} \mapsto \mathsf{A} \\ \mathsf{AE} \mapsto \mathsf{D} & \mathsf{BD} \mapsto \mathsf{E} \\ & \mathsf{BE} \mapsto \mathsf{C} \end{array}$$

Figure 3 on page 16 is provided for reference for the definitions of the normal forms.

---

a. (3 points) What are the candidate keys of **R** given the functional dependencies in $\mathcal{F}$?

$\underline{\mathsf{BD}}$, $\underline{\mathsf{BE}}$, $\underline{\mathsf{ACD}}$, and $\underline{\mathsf{ACE}}$.

---

b. (2 points) Given $\mathcal{F}$, is **R** in BCNF?
Justify.

$\mathsf{AC} \mapsto \mathsf{B}$ breaks BCNF since AC is not a key nor superkey.
Since every attribute is prime, it is necessarily in 3NF and in 2NF.

c. (3 points) Are there additional functional dependencies (for example, $\mathsf{A} \mapsto \mathsf{D}$), over attributes A to E that could be *added* to $\mathcal{F}$—resulting, say, in the set of dependencies $\mathcal{F}'$—such that $\mathbf{R}$ with respect to $\mathcal{F}'$ would *not* be in 3NF?

If yes, show additional functional dependencies that would result in this. If not, explain why not.

Yes. We have to make it so not every attribute is prime. So we can introduce smaller candidate keys, and do it in such a way some of the previous FDs are problematic:

- *$D \mapsto ABCE$ and*
- *$E \mapsto ABCD$.*

Now $\underline{D}$ and $\underline{E}$ are the only two candidate keys; all the other attributes are not *prime. $AC \mapsto B$ now violates 3NF since B is not prime and AC is not a key nor superkey.*

---

d. (2 points) Is it always possible to decompose *losslessly* a relation that is *not* in BCNF but is in 3NF so the resulting relations are in BCNF?

Explain briefly.

Yes. We can always split a relation losslessly with respect to a BCNF-violating FD. We can do this until no BCNF violations remain.
(However, we cannot always obtain a lossless join decomposition that is also dependency preserving*. But that is another story.)*

2. (10 points) **Relational Algebra.** *That doesn't add up!* [Short Answer]

a. (2 points) Given relations $\mathbf{R}(A, B)$ and $\mathbf{S}(B, C)$, write an equivalent relational algebra formula for $\mathbf{R} \bowtie \mathbf{S}$ using only "σ", "π", "$\times$", "$\cup$", "$-$", and "ρ" (the *rename* operator). For full marks, be certain to have the correct resulting columns.

$$\pi_{A,B,C}(\sigma_{B=B2}(\mathbf{R} \bowtie \rho_{B \rightarrow B2}(\mathbf{S})))$$

b. (2 points) Given relations $\mathbf{R}(A)$ and $\mathbf{S}(A)$, write an equivalent relational algebra formula for $\mathbf{R} \cap \mathbf{S}$ using only "σ", "π", "$\times$", "$\cup$", "$-$", and "ρ" (the *rename* operator).

$$\mathbf{R} - (\mathbf{R} - \mathbf{S})$$

c. (2 points) Explain briefly what *property* the *relational algebra* has that allows it to be called an *algebra*.

Inputs to the RA operators are relational tables, and the outputs of each operator is a table. Thus, the operators are composable.

For Questions 2d and 2e, consider relations **R**(A, B), **S**(B, C), and **T**(C, A), and relational instances as in Figure 1.

R	
A	B
1	a
2	b
2	c
3	d

S	
B	C
a	a
b	a
b	b
c	d

T	
C	A
a	2
b	1
c	3
d	4

Figure 1: Tables **R**, **S**, & **T**.

d. (2 points) Show the result of $(\mathbf{R} \bowtie \mathbf{S}) \bowtie \mathbf{T}$.

$(\mathbf{R} \bowtie \mathbf{S}) \bowtie \mathbf{T}$		
A	B	C
2	b	a

e. (2 points) Show the result of $\mathbf{S} \div \pi_{C}(\sigma_{A \leq 2}(\mathbf{T}))$.

$\pi_C(\sigma_{A\leq 2}(\mathbf{T}))$
C
a
b

$\mathbf{S} \div \pi_C(\sigma_{A\leq 2}(\mathbf{T}))$
B
b

3. (10 points) **General.** *They're all C's!* [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.

In the *rare* case that you feel a clarification to your answer is needed, write a brief clarification on the side.

---

a. In a real relational database system, if you try to join (natural join) tables **R** and **S** *and* table **R** is empty (that is, it has no tuples),

A. the system reports an error.
$\boxed{\textbf{B.}}$ 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 tuple.
E. an answer set is returned; however, the results are system dependent.

For Questions 3b to 3d, let |**T**| represent the number of tuples in table **T**.

---

b. Consider the relations $\mathbf{R}(\underline{A}, B)$ and $\mathbf{S}(\underline{A}, C)$, and that **R** has a *foreign key* on A that references **S**. $|\mathbf{R} \bowtie \mathbf{S}|$ is

$\boxed{\textbf{A.}}$ $|\mathbf{R}|$
B. $|\mathbf{S}|$
C. $|\mathbf{R}| \cdot |\mathbf{S}|$
D. $\max(|\mathbf{R}|, |\mathbf{S}|)$
E. $\min(|\mathbf{R}|, |\mathbf{S}|)$

---

c. Consider the relations $\mathbf{R}(\underline{A}, \underline{B})$ and $\mathbf{S}(\underline{B}, C)$, and that **R** has a *foreign key* on B that references **S**. $|\mathbf{R} \bowtie \mathbf{S}|$ is

$\boxed{\textbf{A.}}$ $|\mathbf{R}|$
B. $|\mathbf{S}|$
C. $|\mathbf{R}| \cdot |\mathbf{S}|$
D. $\max(|\mathbf{R}|, |\mathbf{S}|)$
E. $\min(|\mathbf{R}|, |\mathbf{S}|)$

---

d. Consider the relations $\mathbf{R}(\underline{A}, \underline{B})$ and $\mathbf{S}(\underline{B}, \underline{C})$. The *largest* that $|\mathbf{R} \bowtie \mathbf{S}|$ can be is

A. $|\mathbf{R}|$
B. $|\mathbf{S}|$
$\boxed{\textbf{C.}}$ $|\mathbf{R}| \cdot |\mathbf{S}|$
D. $\max(|\mathbf{R}|, |\mathbf{S}|)$
E. $\min(|\mathbf{R}|, |\mathbf{S}|)$

---

e. We know that table **Q** has only one candidate key. From this, we know the following.

A. **Q** is in 2NF.
B. **Q** is in 2NF but is not in 3NF.
$\boxed{\textbf{C.}}$ If **Q** is in 3NF, it is also in BCNF.
D. **Q** cannot be in BCNF.
E. None of the above.

---

f. Consider the relation **T** with attributes A, B, C, D, E, F, and G and with the following functional dependencies (FDs):

$$\begin{array}{ll} \mathsf{A} \mapsto \mathsf{B} & \mathsf{BD} \mapsto \mathsf{F} \\ \mathsf{AC} \mapsto \mathsf{E} & @ \mapsto \mathsf{D} \end{array}$$

Unfortunately we do not know what '@' is. It could be any nonempty subset of **T**'s attributes. (In particular, '@' might even contain D itself, which would make "@ $\mapsto$ D" a trivial dependency.)

Which of the following must be true, regardless of what is inside '@'?

$\boxed{\textbf{A.}}$ A and G must be in any key.
B. C and D must be in all keys.
C. A and E can be in one key.
D. A can never be in a key with G.
E. F can never be in a key.

---

g. In relational algebra, the intersection operator ($\cap$) is *not* logically redundant if we only have additionally

A. join ($\bowtie$).
B. crossproduct ($\times$), select (σ), and project (π).
C. difference ($-$) and union ($\cup$).
D. crossproduct ($\times$) and difference ($-$).
$\boxed{\textbf{E.}}$ crossproduct ($\times$) and union ($\cup$).

For Questions 3h–3j, one of the choices is not like the others; that is, one of the choices could evaluate to a different answer than the others do.

Choose the one that may evaluate differently.

---

h. Consider the relations $\mathbf{R}(\underline{A}, \underline{B})$ and $\mathbf{S}(\underline{B}, \underline{C})$.

A. $\pi_{A}(\mathbf{R} \bowtie \mathbf{S})$
B. $\pi_{A}(\mathbf{R}) - (\pi_{A}(\mathbf{R}) - \pi_{A}(\mathbf{R} \bowtie \mathbf{S}))$
$\boxed{\textbf{C.}}$ $\pi_{A}(\mathbf{R}) - (\pi_{A}(\mathbf{R} - \pi_{A,B}(\mathbf{R} \bowtie \mathbf{S})))$
D. $\pi_{A}(\mathbf{R} \cap (\pi_{A}(\mathbf{R}) \times \pi_{B}(\mathbf{S})))$
E. $\pi_{A}((\mathbf{R} \times \pi_{C}(\mathbf{S})) \cap (\pi_{A}(\mathbf{R}) \times \mathbf{S}))$

---

i. Consider the schema $\mathbf{R}(\underline{A}, B, C)$, $\mathbf{S}(\underline{A}, \underline{D})$, and $\mathbf{T}(\underline{D}, B, E)$.

A. $\pi_{B}((\sigma_{C=5}(\mathbf{R}) \bowtie \sigma_{E=7}(\mathbf{T})) \bowtie \mathbf{S})$
B. $\pi_{B}(\pi_{A,B,D}(\sigma_{(C=5)\wedge(E=7)}(\mathbf{R} \bowtie \mathbf{T})) \cap (\mathbf{S} \times \pi_{B}(\mathbf{T})))$
C. $\pi_{B}((\sigma_{C=5}(\mathbf{R} \bowtie \mathbf{S})) \bowtie (\sigma_{E=7}(\mathbf{S} \bowtie \mathbf{T})))$
$\boxed{\textbf{D.}}$ $\pi_{B}(\sigma_{C=5}(\mathbf{R} \bowtie \mathbf{S})) \cap \pi_{B}(\sigma_{E=7}(\mathbf{S} \bowtie \mathbf{T}))$
E. $\pi_{B}(\pi_{A,B}(\sigma_{C=5}(\mathbf{R})) \bowtie (\mathbf{S} \bowtie \pi_{D,B}(\sigma_{E=7}(\mathbf{T}))))$

---

j. Consider the relations $\mathbf{R}(A, B)$, $\mathbf{S}(B, C)$, and $\mathbf{T}(C, A)$.

A. $\pi_{A,B}((\mathbf{R} \bowtie \mathbf{S}) \bowtie \mathbf{T})$
B. $\pi_{A,B}(\mathbf{R} \bowtie (\mathbf{T} \bowtie \mathbf{S}))$
C. $\mathbf{R} \bowtie \pi_{A,B}(\mathbf{S} \bowtie \mathbf{T})$
D. $\pi_{A,B}(\mathbf{R} \bowtie \mathbf{S}) \bowtie \pi_{A,B}(\mathbf{S} \bowtie \mathbf{T})$
$\boxed{\textbf{E.}}$ $\pi_{A,B}(\mathbf{R} \bowtie \mathbf{T}) \bowtie \pi_{A,B}(\mathbf{R} \bowtie \mathbf{S})$

---

4. (10 points) **SQL.** *Some quirky language.* [Exercise]

   For Questions 4a through 4c, consider the schema for the Canadian Quiddich Association Match database in Figure 2 on page 15. To receive full credit, do *not* source *any* table—that is, list a table in a *from* clause—in your SQL query that is *not absolutely necessary* to evaluate the query correctly.

   You may consider your query in answer to Question 4a as a *view* named **Point** for use in answering Questions 4b and 4c, if you wish.

---

a. (3 points) Report by **Match** (tournament, day, and time) each **Event** that occurred during the **Match** by its time (etime), the **Player** responsible by team, player#, and name, and its action and the points it achieved.

```
select E.tournament, E.day, E.time, E.etime
       E.team, E.player#, P.name,
       E.action, A.points
from Event E, Player P, Action A
where E.team = P.team and E.player# = P.player# -- Join of E & P
  and E.action = A.action;                      -- Join of E & A
```

b. (3 points) Report the *winning* **Team** for each **Match** by the **Team**'s name, the **Match**'s tournament, day, and time, and the total points scored by the winning **Team**. The **Team** with the most points in a **Match** is the winning **Team**. (In case of a tie for the winner, list each of the teams with the high score. For this problem, ties *are* possible.)

```
with
    Score (tournament, day, time, team, scored) as (
        select P.tournament, P.day, P.time, P.team, sum(P.points)
        from (
                select tournament, day, time, team, points
                from Points
                union
                select tournament, day, time, team, 0
                from TeamInMatch
            ) as P
        group by P.tournament, P.day, P.time, P.team),
    Best (tournament, day, time, best) as (
        select S.tournament, S.day, S.time, max(S.scored)
        from Score S
        group by S.tournament, S.day, S.time)
select S.tournament, S.day, S.time, S.team as name, S.scored
from Score S, Best B
where S.tournament = B.tournament
  and S.day = B.day
  and S.time = B.time
  and S.scored = B.best;
```

To be completely correct, we have to account for that a team may have scored no points, having no events *during a match. (I did not mark off for not accommodating this.)*

c. (4 points) For each **Player** by team, player#, name, and gender and **Position** by position, report the *number* of **Matches** (#matches) that the **Player** has played *in* that position and the *total number of points* (#points) that the **Player** playing *in* that position has scored over all **Matches**.

Only list a **Player** in a position if the **Player** has played at least one **Match** in that position (so, #matches > 0). However, do report '0' if the **Player** has not scored any points ever playing in that position.

```
with
    Made (tournament, day, time, team, player#, position, points) as (
        select E.tournament, E.day, E.time,
               E.team, E.player#, R.position, A.points
        from Roster R, Event E, Action A
        where R.tournament = E.tournament
          and R.day = E.day
          and R.time = E.time
          and E.action = A.action
        union
        select M.tournament, M.day, M.time,
               P.team, P.player#, Q.position, 0
        from Match M, Player P, Position P),
    Scored (team, player#, position, #points) as (
        select M.team, M.player#, M.position, sum(M.points)
        from Made M
        group by M.team, M.player#, M.position),
    Matches (team, player#, position, #matches) as (
        select team, player#, position, count(*)
        from Roster
        group by team, player#, position)
select M.team, M.player#, P.name, P.gender, M.position,
       M.#matches, S.#points
from Matches M, Player P, Scored S
where M.team = P.team and M.player# = P.player#
  and M.team = S.team
  and M.player# = S.player# and M.position = S.position;
```

5. (10 points) **Query Logic.** *Stop making sense!* [Analysis]

a. (2 points) Given $\mathbf{R}(\underline{A}, \underline{B})$ and $\mathbf{S}(\underline{A}, \underline{B})$, rewrite

$$\pi_{A}(\mathbf{R} - \mathbf{S})$$

as a succinct, equivalent *SQL* query.

```
select distinct A
from ( select A, B from R
       except
       select A, B from S) as Z;
```

b. (2 points) Given $\mathbf{R}(\underline{A}, \underline{B})$ and $\mathbf{S}(\underline{B}, \underline{C})$, rewrite

```
select distinct Z.A, X.C
from R Z, S X
where Z.B not in (
        select Y.B
        from S Y
        where X.C = Y.C);
```

as a succinct, equivalent *relational-algebra* expression.

$$(\pi_{A}(\mathbf{R}) \times \pi_{C}(\mathbf{S})) - \pi_{A,C}(\mathbf{R} \bowtie \mathbf{S})$$

c. (2 points) State in *plain, concise English* what the following SQL query (over the CQA schema in Figure 2 on page 15) does.

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!)

```
select A.team as teamA, A.player# as playerA,
       B.team as teamB, B.player# as playerB
from Roster A, Roster B
where A.tournament = B.tournament
  and A.day = B.day and A.time = B.time
  and A.team < B.team;
```

Players who have played against each other in a match.

d. (2 points) Given **R**(<u>A</u>, <u>B</u>, <u>C</u>), rewrite the relational-algebra expression

$$(\pi_{\mathsf{A}}(\sigma_{\mathsf{B}<5}(\mathbf{R}))) \bowtie (\pi_{\mathsf{A}}(\sigma_{\mathsf{C}\geq 5}(\mathbf{R})))$$

as an equivalent *SQL* query.

```
select distinct X.A
from R X, R Y
where X.A = Y.A
  and X.B < 5
  and Y.C >= 5;
```

e. (2 points) Consider **R**(<u>A</u>, *B*, *C*)—notice the difference from Question 5d—and the relational-algebra expression

$$(\pi_{\mathsf{A}}(\sigma_{\mathsf{B}<5}(\mathbf{R}))) \bowtie (\pi_{\mathsf{A}}(\sigma_{\mathsf{C}\geq 5}(\mathbf{R})))$$

(same as in Question 5d). Dr. Dogfurry says the relational-algebra expression above will then evaluate to the *empty* answer set, regardless of what is in table **R**!
Either explain why he is correct, *or* construct a small example of **R**—tuples in **R**—that would result in an answer.

He is wrong. Consider

R		
A	*B*	*C*
1	*4*	*6*

The query over this instance of **R** *would return* $\langle 1 \rangle$.

Extra Space

Relax. Turn in your exam. Return to the wild.

Reference *(Detach this page for convenience, if you want.)*

Schema for The Canadian Quiddich Association Match Database.

Team(name, *founded*)
Player(team, player#, *name*, *gender*, *house*, *dob*)
  FK (team) refs **Team** (name)
Position(position)
Tournament(name, *start*)
Stadium(name, *capacity*)
Match(tournament, day, time, *where*, *price*)
  FK (tournament) refs **Tournament** (name)
  FK (where) refs **Stadium** (name)
TeamInMatch(tournament, day, time, team)
  FK (tournament, day, time) refs **Match**
  FK (team) refs **Team** (name)
Roster(tournament, day, time, team, player#, *position*)
  FK (tournament, day, time, team) refs **TeamInMatch**
  FK (team, player#) refs **Player**
  FK (position) refs **Position**
  UNIQUE (tournament, day, time, team, position)
Action(action, *points*, description)
Event(tournament, day, time, team, player#, etime, *action*)
  FK (tournament, day, time, team, player#) refs **Roster**
  FK (action) refs **Action**

Figure 2: CQA Schema.

REFERENCE

The Normal Form Definitions.

1NF: Domain of each attribute is an *elementary* type; that is, not a *set* or a *record structure*.

2NF: Whenever $\mathcal{X} \mapsto A$ is a functional dependency that holds in relation **R** and $A \notin \mathcal{X}$, then either

- A is *prime*, or
- $\mathcal{X}$ is not a proper subset of any key for **R**.

3NF: Whenever $\mathcal{X} \mapsto A$ is a functional dependency that holds in relation **R** and $A \notin \mathcal{X}$, then either

- A is *prime*, or
- $\mathcal{X}$ is a key or a super-key for **R**.

BCNF: Whenever $\mathcal{X} \mapsto A$ is a functional dependency that holds in relation **R** and $A \notin \mathcal{X}$, then

- $\mathcal{X}$ is a key or a super-key for **R**.

An attribute A is called *prime* if A is in any of the candidate keys.

Figure 3: The Normal Forms.