CS343 H1 Y Section 5101 & 2002 Third Term Test Fall 2003

December 3rd 2003

This is a **closed** book, notes, neighbors exam. You have 1 hour and 50 minutes for a total of 250 points. Write your answers in the space provided after each question. You will not be given any credits for anything written outside the page. You will not be given any credits for anything written using a pencil.

Print your First name: ______________________________

Print your Last name: ______________________________

Print your student number: ______________________________

I do not know rule If for any questions I write the following sentence "I do not know the answer" I will be automatically rewarded by a mark equal to 20% of the mark of the question.

I acknowledge and SIGN that this booklet is complete (18 pages, 5 Problems) and 1 appendix (2 pages):

Question	Points	Maximum points
1		105
2		20
3		80
4		25
5		20
Total		250

Schema

Consider the following relational schema describing an atlas (the complete schema can be found in appendix).

```
continent(name, area);
country(name, continent, population);
province(name, country, capital, population);
city(name, country, province);
```

1 Queries (105 points)

For the SQL queries your answers will be also marked in function of their efficiency. In relational algebra, you are **not** allowed to use the linear notation and renaming of *relations* used in the text. You may use renaming of attributes.

Question 1 (30 points)

Give the name of the continents with their area that contains the city whose name is 'Paris'.

Relational Algebra (10 points)

$$\Pi_{Continent.continent, Continent.area}(\sigma_{name='Paris'}(City) \bowtie (\rho_{country \leftarrow name}(Country)) \bowtie (\rho_{continent \leftarrow name}(Continent)))$$

Relational Calculus (10 points)

$$\{n, a \mid \exists na, cp, po, cna, ccp, cpr \quad Continent(n, a), Country(na, co, po), City(cna, cco, cpr) \wedge n = na \wedge co = cco \wedge cna =' Paris'\}$$

SQL (10 points)

```
SELECT cn.name, cn.area
FROM   City ci, Country co, Continent cn
WHERE  ci.name = 'Paris' AND
       ci.country = co.name AND
       co.continent = cn.name
```

Question 2 (20 points)

Give the name of the city together with their country, for the cities that do not have a value for the province attribute.

Relational Algebra (10 points)

$$\Pi_{City.name,City.country}(City \overset{\circ}{\ltimes} (\rho_{province \leftarrow name}(Province))) - \Pi_{City.name,City.country}(City \bowtie (\rho_{province \leftarrow name}(Province)))$$

SQL (10 points)

```
SELECT name, country
FROM City
WHERE province is null
```

Question 3 (10points)

We want to add a new table to represent the border of a country. This relation whose name will be `border` will have two attributes: `country_from, country_to` where `country_from` and `country_to` are valid country name. For instance to represent the border between Canada and United States the table will contain two tuples: `'Canada', 'United States'` and `'United States', 'Canada'`.

Write the correct SQL instruction to create the relation border including every necessary constraints

```
create table border(
 country_from varchar(32) not null,
 country_to varchar(32) not null,
 constraint border_pk primary key(country_from, country_to),
 constraint border_fk_from foreign key (country_from)
                                      references country,
 constraint border_fk_to foreign key (country_to) references country
 );
```

Question 4 (30 points)

Using the border table, can someone reach Brazil from Canada? The output of this query will be a relation containing at least one tuple if the answer of the question is true. For instance one can reach Mexico from Canada because Canada has a border with United States and the United States has a border with Mexico. The answer will output one tuple if the condition is satisfied.

```
with reach(country_from, country_to) as (
 select country_from, country_to
 from border
 where country_from = 'Canada'
union all
 select r.country_from, b.country_to
 from border b, reach r
 where r.country_to = 'Brazil' or r.country_to = b.country_from

)
select country_from, country_to
from reach
where country_from = 'Canada' and country_to = 'Brazil'
fetch first row only
;
```

Question 5 (15 points)

Consider the relation R(A). Write the SQL query that will return the minimum value of the attribute A of the relation R. For the SQL form you are only allow to use Select-From-Where queries without any aggregate functions.

Relational Algebra (5 points)

$$R - \Pi_{\#2}(\sigma_{\#2>\#1}(R \times R))$$

Relational Calculus (5 points)

$$\{x \mid \nexists y, R(x), R(y) \wedge y < x\}$$

SQL (5 points)

```
SELECT DISTINCT A
FROM R r1
WHERE NOT EXISTS (SELECT r2.A
                  FROM R r2
                  WHERE r2.A < r1.A
                 )
```

2 Query Optimization (20 points)

Write the optimized version Q' of the following relational algebra query Q on the relational schema R(A,B,C) and S(C,D,E):

$$Q = \Pi_{A,B}(\sigma_{B='X' \wedge C='Y'}(\sigma_{A='Z' \wedge D='T' \wedge C=E}(R \bowtie S)))$$

$$Q' = \Pi_{A,B}(\sigma_{B='X' \wedge A='Z' \wedge C='Y'}(R) \bowtie (\Pi_C(\sigma_{D='T' \wedge C=E \wedge C='Y'}(S))))$$

3 Normalization and Functional Dependencies (80 points)

Given a relation R(U, $\mathcal{F}$) where U = {A,B,C,D,E,F,G,H,I} and

$$\mathcal{F} = \{A \rightarrow BCE, B \rightarrow GI, E \rightarrow DFBG, I \rightarrow AGD, G \rightarrow D, F \rightarrow G\}$$

Question 1 (10 points)

Find the keys of the relation R. Explain why there are keys for this relation?

Keys: AH, HI, EH, BH

Question 2 (15 points)

Give the minimal closure $\mathcal{F}_{min}$ of $\mathcal{F}$

$$\mathcal{F}_{min} = \{A \rightarrow C, A \rightarrow E, B \rightarrow I, E \rightarrow F, E \rightarrow B, I \rightarrow A, G \rightarrow D, F \rightarrow G\}$$

Question 3 (20 points)

Give a lossless BCNF decomposition of R(U,$\mathcal{F}_{min}$). Is your decomposition dependencies-preserving? Justify your answer.

- R1(ACE) $\mathcal{F}_1 = \{A \rightarrow CE\}$
- R2(BI) $\mathcal{F}_2 = \{B \rightarrow I\}$
- R3(DG) $\mathcal{F}_3 = \{G \rightarrow D\}$
- R4(FG) $\mathcal{F}_4 = \{F \rightarrow G\}$
- R5(ABFH) $\mathcal{F}_5 = \emptyset$

not preversing FD: $E \rightarrow F$ losts.

Question 4 (20 points)

Give a lossless, dependencies preserving 3NF decomposition of R(U,$\mathcal{F}_{min}$).

- R1(ACE) $F_1 = \{A \rightarrow CE\}$
- R2(BI) $F_2 = \{B \rightarrow I\}$
- R3(AI) $F_3 = \{I \rightarrow A\}$
- R4(EBF) $F_4 = \{E \rightarrow BF\}$
- R5(GD) $F_5 = \{G \rightarrow D\}$
- R6(FG) $F_6 = \{F \rightarrow G\}$
- R7(AH) - one key - $F_7 = \emptyset$

Question 5 (15 points)

Proof that the algorithm given in the lecture to decompose a relational schema R(U,F) into a set of BCNF relational schema is lossless join. Your proof should not exceed 10 lines.

Algo: R(U,F) If $X \to Y \in F$ not BCNF => R1(XY) and R2(U-Y).

R1 and R2 are lossles join iff: $R1 \cap R2 \to (R1 - R2) or (R2 - R1)$

$$R1 \cap R2 = X$$

$$R1 - R2 = Y$$

$$R1 \cap R2 \to R1 - R2 = X \to Y \in F.$$

4 Multivalued Dependencies (25 points)

The following question are extracted from [Godin, R. (2003). Systèmes de Gestion de Bases de Données par l'Exemple]

Let Movies(U, $\mathcal{D}$) be a relational schema where:

- U={Title, Genre, Producer, Year, Actor, Hour, Date, thEater, Length}. From this point we will use only the capitalize letter of each attribute: U={T,G,P,Y,A,H,D,E,L}.
- $\mathcal{D} = \{TY \twoheadrightarrow A, T \rightarrow G, TY \rightarrow PL, DET \rightarrow H, HDE \rightarrow TY\}$

Question 1 (5 points)

Give without any justifications all the keys of the relation M(U, $\mathcal{D}$)

ADEH, ADET.

Question 2 (10 points)

Is this schema in 4NF? Justify your answer.

No, for instance $TY \twoheadrightarrow A$ violates the 4NF condition. (TY is not a key).

Question 3 (10 points)

Give a 4NF decomposition of the relation M(U, $\mathcal{D}$).

- R1(TYA) $\mathcal{D}_1 = \{TY \twoheadrightarrow A\}$.
- R2(TG) $\mathcal{D}_2 = \{T \rightarrow G\}$
- R3(TYPL) $\mathcal{D}_3 = \{TY \rightarrow PL\}$
- R4(TYHDE) $\mathcal{D}_4 = \{DET \rightarrow H, DEH \rightarrow TY\}$

5 Transaction (20 points)

Consider the following schedule S:

ordinal	T1	T2	T3
1			read(z)
2		read(y)	
3			write(z)
4	read(z)		
5	write(z)		
6		write (y)	
7	write(x)		
8			read(y)
9	read(x)		

Question 1 (10 points)

Is S conflict-serializable? Justify your answer.

$$\mathcal{G} = T2 \longrightarrow T3 \longrightarrow T1$$

The precedence graph is acyclic therefore S is conflict serializable.

Question 2 (10 points)

If the schedule S is conflict serializable write the equivalent serial schedule S' in term of transition only (T1,T2,T3).

S' = T2, T3, T1