Student number:_____

UNIVERSITY OF VICTORIA
Faculty of Engineering
Department of Computer Science

Oct 2, 2014

CSC 370 (Database Systems)
Instructor: Daniel M. German

Duration: 60 minutes

This is a closed-book exam. You are only allowed one sheet of paper, handwritten, letter sized.

This examination paper consists of **5** pages and **3** questions. Please bring any discrepancy to the attention of an invigilator. The number in brackets at the start of each question is the number of points the question is worth.

Answer all questions.

Please write your answers clearly.

For instructor's use:

	Score
1 (4)	
2 (32)	
3 (4)	
Total (40)	

For this exam, consider the following schema and instances of the relations. Feel free to remove this page from the exam. The instances of the tables are for illustration purposes only.

```
Students(sid: integer, sname: string,
         age: integer, gpa: real)
```

- Primary Key: **sid**

sid	*sname*	*age*	*gpa*
1	Storey	24	7.5
2	Damian	21	8.2
3	Hoffman	21	5.9

```
Courses(cid: string, cname: string,
        department: string)
```

- Primary Key: **cid**

cid	*cname*	*department*
CSC-370	Managing my phone book	CS
HIST-320	History of geeks	HIST
CSC-450	Social Networks	CS
CHEM-100	Baking doughnuts	CHEM

```
Enrolled(sid: integer, cid: string,
         grade: integer)
```

- Primary Key: (**sid,cid**)
- Foreign Key: **sid** references **Students**
- Foreign Key: **cid** references **Courses**

sid	*cid*	*grade*
1	CSC-370	70
2	CSC-370	
3	CSC-370	
1	CSC-450	59
3	CSC-450	100
2	HIST-320	92

1. **Relational Model**

(a) [4] Write a CREATE TABLE for the relation *Enrolled*. Make sure you add all the necessary constraints. Choose adequate data types for the attributes of the relation.

```
CREATE TABLE Enrolled (
    sid    char(10),
    cid    char(10),
    grade  integer,
    primary key (sid, cid),
    foreign key (sid) references students,
    foreign key (cid) references Courses
);
```

2. **Writing queries**

Write queries to answer the following questions, both in relational algebra and SQL. The relational algebra should match the SQL.

(a) [4] Find the **sid, sname** of all the students who are younger than 19 years old. Result should have two columns.

$\Pi_{sid, sname}\, \sigma_{year<19}\, S$

```
SELECT sid, sname FROM students
WHERE year < 19
```

(b) [4] Find the **sid** of the students who are not enrolled in any course. Result should have one column.

$\Pi_{sid}\, S - \Pi_{sid}\, E$

```
SELECT sid FROM students EXCEPT
SELECT sid FROM Enrolled;
```

(c) [4] Find the **sid, sname** of every student who has GPA > 8.0 and who is enrolled in at least one course. Result should have two columns.

$\Pi_{sid, sname}\, \sigma_{gpa>8.0}\, (S \bowtie E)$

```
SELECT sid, sname FROM
students NATURAL JOIN Enrolled;
```

(d) [4] Find the **sname** and **cid** for every student who is registered in a course for which he/she has not received a grade. Result should have 2 columns.

$\Pi_{sname, cid}\ \sigma_{grade\ is\ NULL}\ S \bowtie E$

```
SELECT sname, cid FROM
   students NATURAL JOIN Enrolled
WHERE grade IS NULL;
```

(e) [4] Find the **sid** and **sname** of all the students who are enrolled in the course *'History of Geeks'*.

$\Pi_{sid, sname}\ \sigma_{cname='History\ of\ Geeks'}\ S \bowtie E \bowtie C$

```
SELECT sid, sname FROM
    students NATURAL JOIN enrolled
                NATURAL courses
WHERE cname = 'History of Geeks'
```

(f) [4] Find the **sname** of every student who is enrolled in both (**cid**) 'CSC-370' and 'CSC-450'. Result should have two columns.

$\Pi_{sname}\ (\Pi_{sid}\ \sigma_{cid='csc-370'}\ E \cap \Pi_{sid}\ \sigma_{cid='csc-450'}\ E\ ) \bowtie S$

```
SELECT sname FROM (SELECT sid FROM
   Enrolled WHERE cid = 'CSC370' INTERSECT
    SELECT sid FROM Enrolled WHERE cid = 'csc450')
AS C1 NATURAL JOIN S;
```

(g) [4] What is the difference of the GPA of student 'Damian' with respect to the GPA of student 'Storey'. REsult should have one column and one attribute. In other words, compute GPA of Damian minus GPA of 'Storey'. using sname is valid too

$D = \Pi_{gpa \to gpad}\ \sigma_{sid='Damian'}\ S$

$T = \Pi_{gpa \to gpas}\ \sigma_{sid='storey'}\ S$

$\Pi_{gpad - gpas}\ D \times T$

```
WITH D AS(Select gpa as gpad from students
          WHERE sid = 'Damian'),
WITH T AS(Select gpa as gpas from students
          WHERE sid = 'Storey')
SELECT gpad - gpas FROM D, T;
```

(h) [4] Find the **sid** of students who are enrolled in at least two courses (you cannot use `GROUP BY`).

E1 = E

$\pi_{E.sid}\ \sigma_{E1.sid = E.sid \text{ and } E1.cid <> E.cid}\ (E1 \times E)$

```
SELECT E.sid FROM Enrolled as E1, E
WHERE E1.sid = E.sid and
      E1.cid <> E.cid;
```

3. Security

(a) [4] All tables in our database are owned by user Alice. User Bob needs to execute the following command. What are the minimal privileges he needs from Alice to be able to execute it:

```
DELETE FROM Enrolled
WHERE sid IN (SELECT sid from STUDENTS
              WHERE sname = 'Charlie Quitter');
```

Enrolled: Delete
Select sid

Students: Select sid
Select sname

End of examination
Total pages: 5
Total marks: 40