UNIVERSITY OF VICTORIA
Faculty of Engineering
Department of Computer Science

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

Mid-Midterm Exam
Feb 28, 2014

Duration: 50 minutes

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

This examination paper consists of **5** pages and **3** sections. Please bring any discrepancy to the attention of an invigilator. The number in parenthesis 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 (8)	
2 (12)	
3 (10)	
Total (30)	

For this exam, consider the following schema of a simple university database. It includes information about instructors, students, and the courses offered. Feel free to remove this page from the exam.

- The **Students** table contains the id of the student (**sid**), his/her name (**sname**), age (in years), and gpa.

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

  - Key: **sid**

- The **Instructors** table contains information about instructors of the courses: their id (**iid**), name (**iname**) and department they belong to (**dept**). An instruct can teach many different courses.

```
Instructors(iid: string, iname: string, dept: string)
```

  - Key: **iid**

- The **Courses** table contains information about courses: their id (**cid**), their name (**cname**), the department that offers it (**dept**), the id of its instructor (**iid**), and the maximum number of students who can take it (**maxenrol**). Every **iid** in this table is also found in the table **Instructors**.

```
Courses(cid: string, cname: string,
        dept: string, iid: string,
        maxenrol:integer
)
```

  - Key: **cid**

- The table **Enrolled** contains what students are registered to which courses, and the grade they receive (NULL if they have not received one yet). A student can only register once to any given course, but he/she can register to as many courses as necessary. Neither **sid** nor **cid** can be NULL. Every **sid** in this table is also found in the table **Students**, and every **cid** in this table is also found in the table **Courses**.

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

  - Key: (**sid,cid**)

1. Functional Dependencies

(a) [2] Assume a relation $R(T, C, M)$.

- T corresponds to the name of the Theater.
- C corresponds to the name of the City
- M corresponds to the name of the Movie.
- The name of the Theater is unique across all Cities. $T \to C$
- There are several Theaters per City. Explicitly $C \not\to T$
- We only show a given Movie in one Theater per City. $MC \to T$

What are the functional dependencies that apply to this relation?

So fds are $T \to C$, $MC \to T$

(b) [2] Given the functional dependencies: $A \not\to B$, $CH \to A$, $B \not\to E$, $BD \not\to C$, $EG \to H$, $DE \not\to F$, is it possible to generate $ADE \to CH$? Why?

We can generate ADE iff $\{ADE\}^+$ contains CH.

$\{ADE\}^+ = \{ADEBCF\}$ so no, it can not be generated

(c) [4] Consider relation R(A,B,C,D) with functional dependencies: $D \to C$,$CB \to A$,$D \to A$,$AB \to D$. Compute all its candidate keys.

B is never in right hand side, so any key will contain B

Superkeys:

✓	B	A	C	D	BACD
✓	B	A	C		BACD
✓	B	A		D	BADC
✓	B	A			BADC
✓	B		C	D	BCDA
✓	B		C		BCAD
✓	B			D	BDCA
	B				B

BA, BC and BD are candidate keys

2. Normalization

(a) [2] Given the relation $R(ABC)$ with functional dependencies $A \to C$ and $C \to B$. Is the decomposition into relations AC and BC lossless join? Explain.

To be lossless join $\exists$ $AC \cap BC \to A$ or $AC \cap BC \to B$

Since $C \to B$ is given, this Decomp. is lossless join.

(b) [2] Is the previous decomposition FD preserving? Why?

AC has FDs. $A \to C$ and BC has FDs. $C \to B$.

So yes. It is FD preserving.

(c) [2] Assume R is a relation with two or more attributes, and that it has one non-trivial functional dependency. Is R **always** BCNF? Explain.

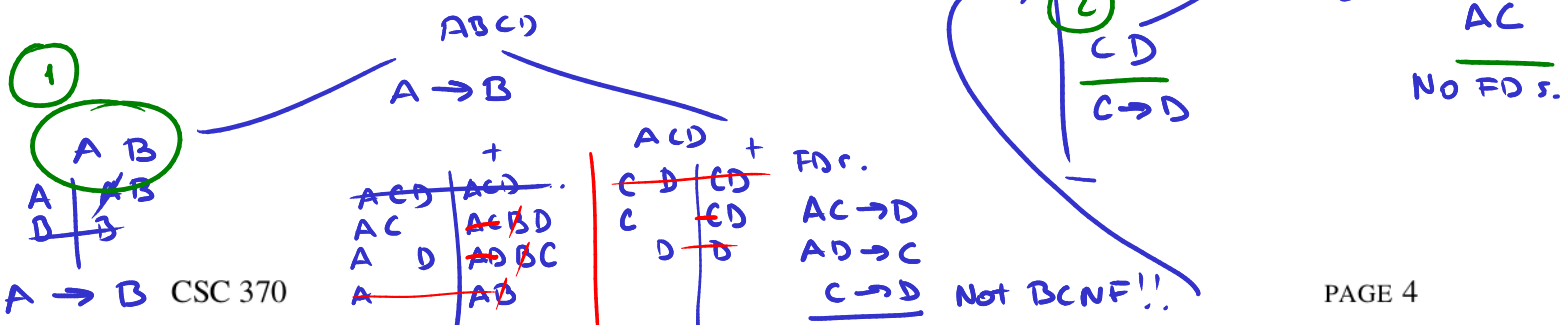
if the FD $A_1 \ldots A_n \to B_1 \ldots B_k$ and it is not trivial than none of $B_1 \ldots B_k$ are in $A_1 \ldots A_n$

It would be BCNF iff $A_1 \ldots A_n \cup B_1 \ldots B_n = R$.

otherwise it is NOT. So no, it is not always BCNF

(d) [6] Consider the relation R(A,B,C,D) with functional dependencies: $A \to B$, $C \to D$, $AD \to C$, $BC \to A$. This table is not BCNF. Decompose this relation into a set of BCNF relations that are functional dependency preserving.

$\{A\}^+ = \{AB\}$ $A \to B$ NOT BCNF



Decomposition into AB, CD, AC loses $BC \to A$ and $AD \to C$

$A \to B$ $C \to D$ NO FDs add ABC and ADC

Optional : we can combine AB & ABC and ADC and CD

⇒ Final Decomp: ABC , ADC and AC with NO FDs.
BC → A, A → B; AD → C, C → D

3. Relational Algebra and SQL

For each of the following questions, provide a relational algebra expression to answer them, and its equivalent SQL query:

(a) [2] What is the average **age** of the students who are taking at least one course? Result should have only one column (and one tuple). Hint. Make sure you average each student's age only once.

$\gamma_{avg(age)}(\sigma_{sid \text{ IN } (\pi_{sid} E)} S)$

```
select avg(age) from students
where sid IN (select sid from Enrolled)
```

(b) [4] For every instructor that is teaching exactly two courses, list the **iid** of the instructor, their name **iname** and the course **cid** they are teaching. There are going to be two tuples for each instructor, one for each course they teach. For instance, your result should look something like this (three columns).

iid	iname	cid
342	M. Zastre	Seng 365
342	M. Zastre	CSC 360
123	D. German	CSC 370
~~456~~ 123	D. German	CSC 225

$A = \pi_{iid} \sigma_{c=2} \gamma_{iid, count(*) \to c} C$

$\pi_{iid, iname, cid}$ C ⋈ A ⋈ I

```
WITH A AS ( select iid from C group by iid
            having count(*) = 2)
select iid, iname, cid from C NATURAL
JOIN A NATURAL JOIN I;
```

(c) [4] List the **sId** and **sname** of the students who are enrolled in the fewest courses. Your result should include three columns: **sid**, **sname** and total number of courses. Make sure you consider students who might not be taking any course (in that case they are enrolled to zero courses).

$T = \gamma_{sid, sname, count(cid) \to c}$ E ⟕$_R$ S

sid → sname, so ok.

$M = \gamma_{min(c) \to c} T$

$\pi_{sid, sname}$ T ⋈ M.

```
WITH T AS (SELECT sid, sname,
           count(cid) as c
           from E NATURAL RIGHT JOIN S)
WITH M AS( select min(c) as c
           from T)
select sid, sname from
T NATURAL JOIN M;
```

End of examination
Total pages: 5
Total marks: 30