Student number:_____

UNIVERSITY OF VICTORIA
Faculty of Engineering
Department of Computer Science

29 May 2015

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

Duration: 50 minutes

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

This examination paper consists of **7** 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 on exam paper.

Please write your answers clearly.

For instructor's use:

	Score
1 (15)	
2 (6)	
3 (9)	
Total (30)	

Out of 26 not 30

Feel free to remove this page from the exam.

For this exam, consider the following schema and instances of the relations. Broadly speaking, **Emp** records employees of an organization, **Dept** departments of the organization, and **Works** records which employees work for a given department. **managerid** corresponds to the **eid** of the manager of a department.

This is only a sample instance of the database. Attributes with an empty value should be assumed to be NULL.

`Emp(eid: integer, ename: string, age: integer, salary: real)`

- Primary Key: **eid**

eid	*ename*	*age*	*salary*
11564812	John Williams	35	74098
15487874	Gene Edwards	51	41008
15645489	Daniel Evans	25	
51135593	Maria White	22	24998
54879887	Dorthy Lewis	33	

`Dept(did: integer, dname: string, budget: real, managerid: integer)`

- Primary Key: **did**
- Foreign Key: **managerid** references **Emp**

did	*dname*	*budget*	*managerid*
1	Hardware	100000	15645489
5	Software	200000	15487874
7	Marketing	150000	

`Works(eid: integer, did: integer, pct_time: integer)`

- Primary Key: **eid,did**
- Foreign Key: **eid** references **Emp**
- Foreign Key: **did** references **Dept**

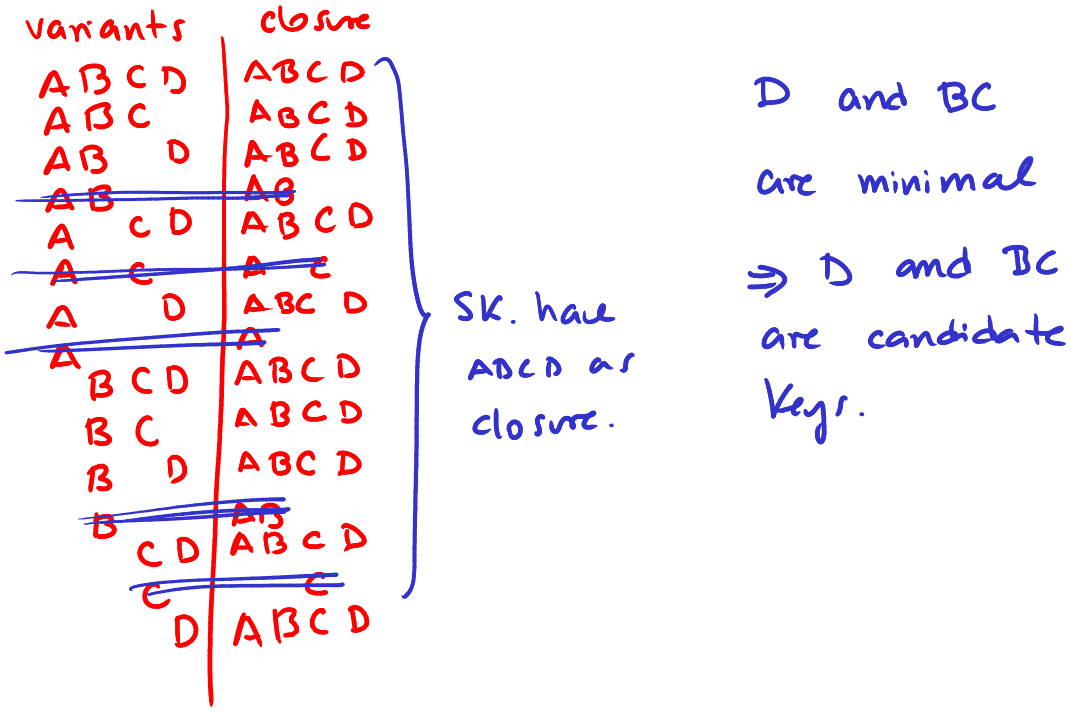
eid	*did*	*pct_time*
11564812	1	100
15487874	5	100
15645489	5	100
51135593	5	100

1. **Database Theory**

For every question, you **must** show all your work.

(a) [3] Given the relation $R(ABCD)$ with set of FDs: $B \rightarrow A$, $BC \rightarrow D$, $D \rightarrow BC$ compute all its candidate keys.

All att are in RHS, hence compute all



(b) [3] Given the relation $R(ABCD)$ with set of FDs: $A \rightarrow B$, $C \rightarrow D$, $AD \rightarrow C$ and $BC \rightarrow A$, with candidate keys AC, AD and BC.

i. is this relation in BCNF? Show all your work.

Test $A \rightarrow B$ $A^+ = AB$ A is not a SK

Hence relation is not BCNF.

ii. is this relation is 3NF? Show all your work.

$A \rightarrow B$ A is part of Key AC

$C \rightarrow D$ D is part of Key AD

$AD \rightarrow C$ C is part of Key AC and AD is a SK.

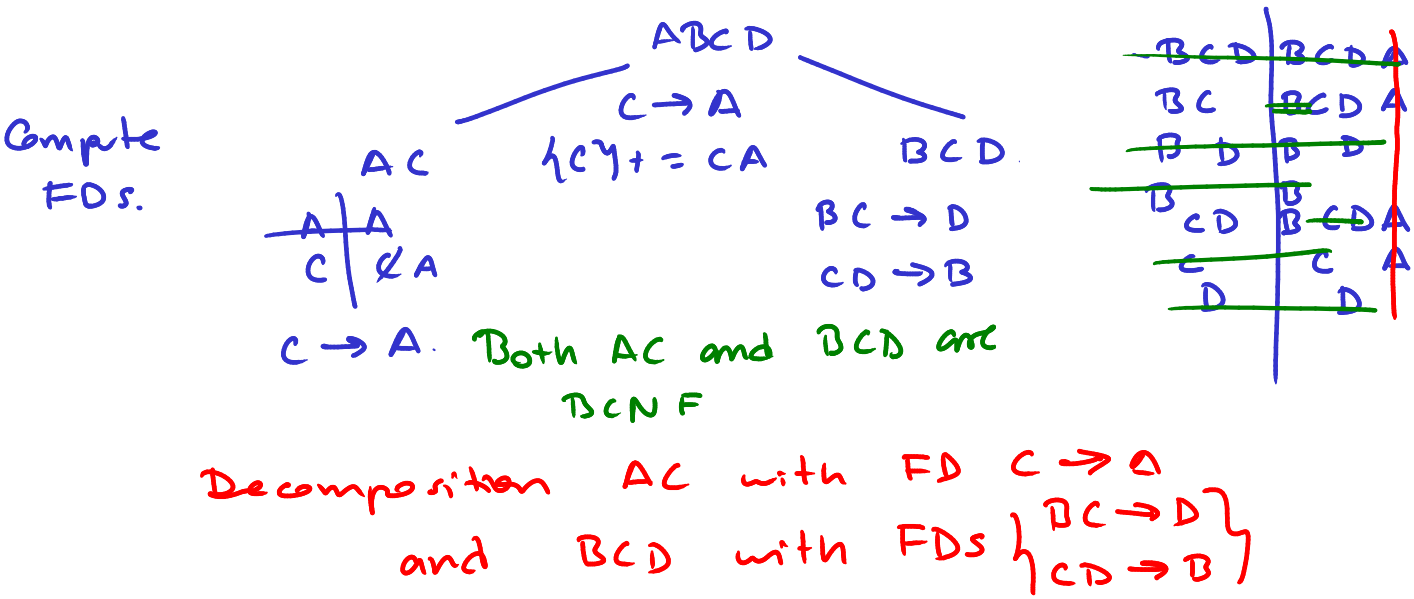
$BC \rightarrow A$ A is part of Key AC and BC is a SK.

(c) [3] Given the relation $R(ABCD)$ with set of FDs: $AB \rightarrow C$, $BC \rightarrow D$, $C \rightarrow A$, $AD \rightarrow B$

i. decompose this relation into a set of BCNF relations.

C → A breaks BCNF

Compute FDs.



Decomposition AC with FD C → A

and BCD with FDs {BC → D, CD → B}

ii. is your decomposition FD preserving? Show all your work.

No. It is not.

From {C → A, BC → D, CD → B} We can not generate AB → C

{AB}+ = {AB}

(d) [3] The relation $R(ABC)$ with set of FDs: $A \to B$, $B \to C$, $C \to A$ has been decomposed into two relations $R_1(AB)$ and $R_2(BC)$. Is this decomposition FD preserving?

1) Compute FDs of subrelations.

$R_1 = AB$

A	~~ABC~~	$A \to B$
B	~~ABC~~	$B \to A$

$R_2 = BC$

B	~~ABC~~	$B \to C$
C	~~ABC~~	$C \to A$

$\{A \to B, B \to A\} \cup \{B \to C, C \to A\}$ contains all original FDs. So Yes, it is FD preserving.

By construction these are all derived from original FDs.

(e) [3] Given the relation $R(ABCDEFGH)$ with set of FDs: $A \to C$, $AC \to D$, $E \to ADH$. Compute a minimal basis for this set of FDs.

1) $A \to C$
$AC \to D$
$E \to A$
$E \to D$
$E \to H$

2) is any attr in AC redundant in $AC \to D$
$\{A\}^+ = \{AC\}$

Yes C is redundand
$A \to D$

3) Given
$A \to C$
$A \to D$
$E \to A$
$E \to D$
$E \to H$

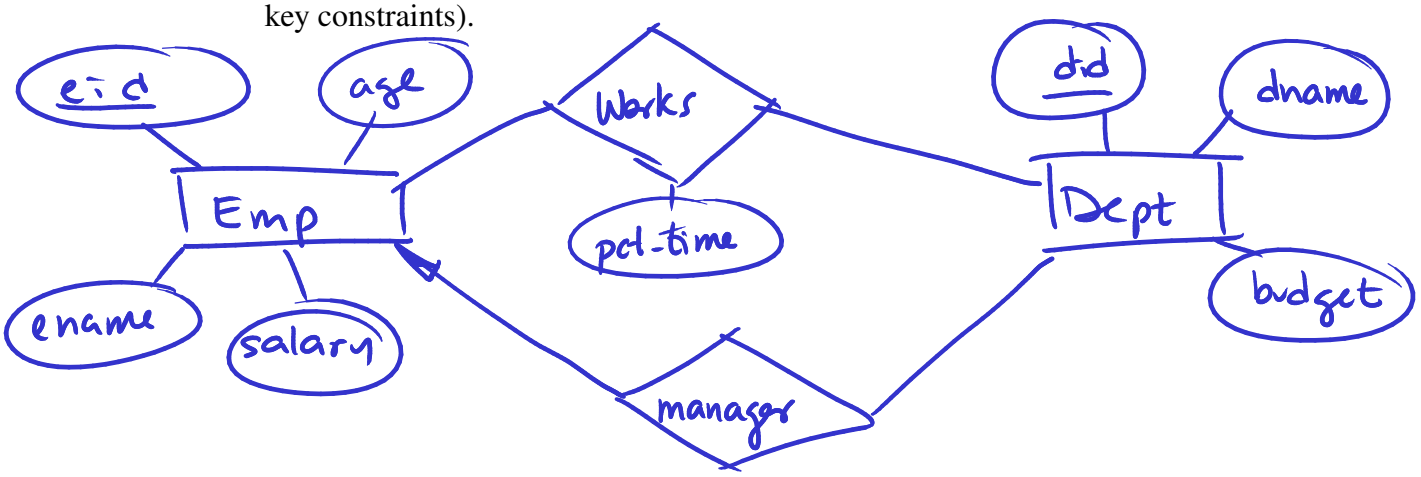
is any redundant

You had to check each. But only redundant is $E \to D$
$\{E\}^+ = EACDH$

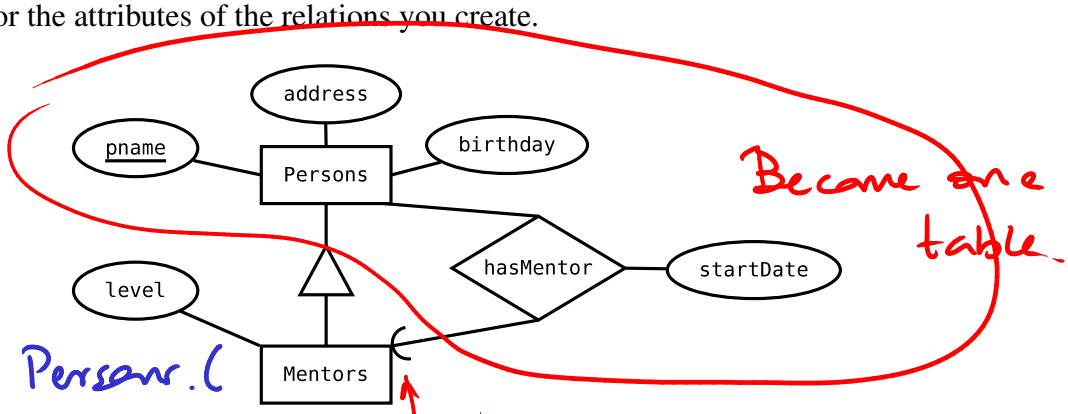
Solution $\{A \to C, A \to D, E \to A, E \to H\}$

2. Entity-Relationship diagrams

(a) [3] Given the relations in page 2 of this exam, draw their corresponding ER diagram. Note that a department can have zero or one managers. All other participation constraints are indicated in the schema of the relations (i.e. their primary keys and foreign key constraints).



(b) [3] Convert the following ER diagram into SQL relations that mimic as close as possible the entities, relationships and constraints of this diagram. You can assume any data types for the attributes of the relations you create.



Became one table.

```
CREATE TABLE Persons.(
    pname   CHAR(30),
    address CHAR(30),
    birthday DATE
    mname   CHAR(30) NOT NULL,
    startdate DATE,
    PRIMARY KEY (pname),
    FOREIGN KEY (mname) REFERENCES Mentors
);
```

mentor

```
CREATE TABLE Mentors(
    pname   CHAR(30)
    level   integer
    PRIMARY KEY (pname)
    FOREIGN KEY(pname)
     REFERENCES Persons);
```

3. **RA and SQL**

Given the database schema in page two, write queries to answer the following questions. Your solutions should include both Relational Algebra and its equivalent SQL. If the SQL does not match the Relational Algebra, the answer will be considered incorrect. You can only use the following relational algebra operations: selection, projection, renaming and set operations. You can assume $E = Emp$, $D = Dept$ and $W = Works$.

(a) [3] List the department id (**did**) and department name (**dname**) of departments that have managers with a salary (**salary**) of more than 1,000,000 (you can abbreviate this number as 1e6). Your result should have two attributes.

$\Pi_{did, dname}\, \sigma_{managerid\ IN\ (\Pi_{eid}\, \sigma_{salary>1e6}\, E)}\, D$

```
SELECT did, dname FROM Dept
WHERE managerid IN (SELECT eid FROM Emp
                    WHERE salary > 1e6);
```

(b) [3] List the **eid** of employees who are currently working for both of the departments with ids 1 and 5 (**did**). Your result should have only one attribute.

ambiguous. ok if interpreted as "either one" or "both at the same time"

Both at the same time.

$(\Pi_{eid}\, \sigma_{did=5}\, W) \cap (\Pi_{eid}\, \sigma_{did=1}\, W)$.

```
SELECT eid FROM W WHERE did = 5
INTERSECT
SELECT eid FROM W WHERE did = 1.
```

(c) [3] List the employee id (**eid**) and name (**ename**) of all employees who are not working for any department or who are working for the department with **did** equal 3. Your result should have two attributes.

$NotW = \Pi_{eid}\, E - \Pi_{eid}\, W$

$In3 = \Pi_{eid}\, \sigma_{did=3}\, W$

$\Pi_{eid, ename}\, \sigma_{eid\ IN\ (NotW \cup In3)}\, E$

```
WITH NotW AS (
    SELECT eid FROM E
    EXCEPT
    SELECT eid FROM W),
In3 AS (
    SELECT eid FROM W
    WHERE did = 3)
SELECT eid, ename FROM E
WHERE eid IN (TABLE NotW
              UNION
              TABLE In3);
```

End of examination. Total pages: 7 Total marks: 30