

Student number: \_\_\_\_\_ Signature: \_\_\_\_\_

**UNIVERSITY OF VICTORIA**  
**Faculty of Engineering**

**Department of Computer Science**

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

**Midterm**  
**18 June 2003**

**Duration: 75 minutes**

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

This examination paper consists of **9** pages and **5** 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 (10)	
2 (10)	
3 (8)	
4 (13)	
5 (9)	
<b>Total (50)</b>	

For this exam, consider the following schema and instances of the relations. Feel free to remove this page from the exam.

Attributes with an empty value should be assumed to be NULL.

Students(**sid**:integer, name:string, login:string, age: integer, gpa: real)

<i>sid</i>	<i>name</i>	<i>login</i>	<i>age</i>	<i>gpa</i>
53666	Mary Elizabeth	me@cs	18	7.4
53668	Quintin	quintin@ee	18	7.8
53650	Pierre	pierre@math	19	7.4
53831	Cathy	cathy@music	11	8
53832	Reuven	reuven@music	12	

Courses(cid: string, cname: string, credits: integer)

<i>cid</i>	<i>cname</i>	<i>credits</i>
SENG265	Introduction to Software Engineering	3
CSC370	Database Systems	3
CSC360	Introduction to Operating Systems	3
CSC320	Foundations in Computer Science	3

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

<i>sid</i>	<i>cid</i>	<i>grade</i>
53666	SENG265	98
53666	CSC370	78
53668	CSC370	91
53832	SENG265	

## 1. Database Concepts

(a) [4] Give 4 advantages of using a DBMS to manage data. Be brief.

(b) [2] Explain why physical data independence is important. Be Brief.

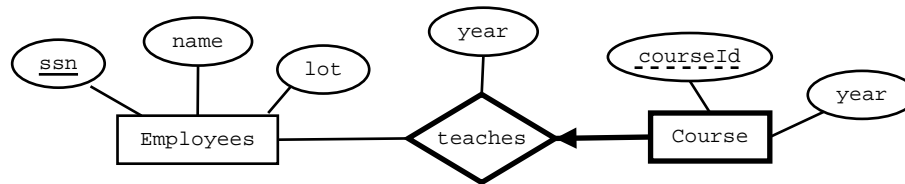
(c) [3] List 3 of the tasks a DBA is responsible for.

(d) [1] Who invented the relational model for databases?

## 2. Database Design

- (a) [5] Draw the Entity Relationship diagram for the following database: A university database contains information about professors (employeeNumber, name and salary) and courses (courseName, term, year, section, name, and department). Professors can be uniquely identified by their employeeNumber, while courses can be uniquely identified by their courseName, term, year and section. Professors teach courses, but there is the restriction that a given course can only be taught by at most one professor.

- (b) [5] For the following diagram, provide CREATE TABLE statements for the entities Employees, Courses and for the relation teaches. Choose appropriate data types for each field and include any referential integrity constraints.



### 3. Relational Model

(a) [2] What is the arity, cardinality and degree of the relation Enrolled in page 2?

Give relational algebra expressions for the following questions using the schema in page 2.

(a) [3] The cname of the courses for which the student with name 'Jones' has a grade bigger than 90.

(b) [3] The cname of each course that does not have students enrolled.

#### 4. Writing SQL queries

Answer the following questions using the schema in page 2.

(a) [3] Provide a `SELECT` statement that retrieves for every student enrolled in a course, their name and the name of the course (`cname`) in which he/she is enrolled.

(b) [3] Provide a `SELECT` statement that retrieves the name of each course (`cname`) that does not have students enrolled.

(c) [3] Provide a `SELECT` statement that retrieves every student id (`sid`) and the total number of credits they have completed (total number of credits of the courses for which they have received a grade).

(d) [4] Provide a `SELECT` statement that groups students who are at least 18 years old by their `age`. For each of these groups compute their average `gpa` and order the result by `age` (in ascending order).

## 5. Interpreting SQL queries

Based on the instances of page 2, what is the answer to the following queries?

(a) [3]

```
SELECT count(gpa) as count, max(gpa) as Maximum,  
       min(gpa) as Minimum  
FROM Students  
WHERE sid IN (SELECT sid FROM Enrolled)
```

(b) [3]

```
SELECT * FROM  
       Enrolled NATURAL RIGHT OUTER JOIN Courses
```



(c) [3]

```
SELECT S.sid, AVG(grade) as Average
  FROM Students S, Enrolled E
  WHERE S.age <= 18 and S.sid = E.sid
GROUP BY S.sid
HAVING count(grade) > 0
```

**End of examination**  
**Total pages: 9**  
**Total marks: 50**