UNIVERSITY *of* LIMERICK

OLLSCOIL LUIMNIGH

COLLEGE *of* INFORMATICS *and* ELECTRONICS
Department of Computer Science and Information Systems

Repeat Exam

Academic Year:	2006/2007	**Semester:** Spring	
Module Title:	Database Systems	**Module Code:** CS4416 and CS5122	
Exam Duration:	2½ Hours	**Total Marks:** 80	
Lecturer:	Dr. N. S. Nikolov		

Instructions to Candidates:

> **Answer ALL questions!**
> Please write **ALL** answers in the answer booklet.
> State clearly any assumptions you make.

SECTION A – SCHEMA DESIGN

Q1. (10 marks) Suppose we are to design a registrar's database to store information about students, modules, the modules students have taken, and the grades students have gotten in these modules. Modules have a number, a department, and a title, for example, "CS1001: Introduction to computing" has department = CS, number = 101, and title = "Introduction to Computing." Numbers are assigned by departments, and different departments may use the same number. Students are represented by their (unique) student ID and their name. "Enrollments" each consist of a module, a student who took that module, and the grade the student got in the module. Draw two different E/R diagrams that represent this database structure correctly.

Q2. (10 marks) Find all keys of the relation R(ABCDEFG) with functional dependencies

$$AG \rightarrow C, BE \rightarrow C, CG \rightarrow A, AF \rightarrow D, FD \rightarrow A, \text{ and } CE \rightarrow B$$

Q3. (5 marks) Which one of the following five relations is in Third normal form (3NF)?

a. R(ABCD) FD's: $ABD \rightarrow C$; $CD \rightarrow A$; $AC \rightarrow B$; $AC \rightarrow D$
b. R(ABCD) FD's: $AB \rightarrow C$; $ABD \rightarrow C$; $ABC \rightarrow D$; $AC \rightarrow D$
c. R(ABCD) FD's: $AB \rightarrow C$; $BCD \rightarrow A$; $D \rightarrow A$; $B \rightarrow C$
d. R(ABCD) FD's: $B \rightarrow C$; $AC \rightarrow D$; $ABD \rightarrow C$; $BCD \rightarrow A$
e. R(ABCD) FD's: $C \rightarrow B$; $A \rightarrow B$; $CD \rightarrow A$; $BCD \rightarrow A$

GamesA

Team	Opponent	Goals_Scored	Goals_Against
Dragons	Tigers	5	3
Carp	Swallows	4	6
Bay Stars	Giants	2	1
Marines	Hawks	5	3
Ham Fighters	Buffaloes	1	6
Lions	Golden Eagles	8	12
Tigers	Dragons	3	5
Swallows	Carp	6	4
Giants	Bay Stars	1	2
Hawks	Marines	3	5
Buffaloes	Ham Fighters	6	1
Golden Eagles	Lions	12	8

GamesB

Team	Day	Opponent	Goals_Scored
Dragons	Sunday	Swallows	4
Tigers	Sunday	Bay Stars	9
Carp	Sunday	Giants	2
Swallows	Sunday	Dragons	7
Bay Stars	Sunday	Tigers	2
Giants	Sunday	Carp	4
Dragons	Monday	Carp	6
Tigers	Monday	Bay Stars	5
Carp	Monday	Dragons	3
Swallows	Monday	Giants	0
Bay Stars	Monday	Tigers	7
Giants	Monday	Swallows	5

Figure 2

Q4. (25 marks) Consider the two relations GamesA and GamesB shown in Fig. 2. Write SQL queries which ask for:

   a. **(5 marks)** The list of all teams in GamesA which are NOT.

   b. **(6 marks)** The list teams from GamesB which scored in every game they played and the maximum number of goals scored by a team in a game.

   c. **(7 marks)** Pairs of teams from GamesA that either scored the same number of goals or had the same number of goals scored against them, or both. The output must list the teams with the first preceding the second alphabetically.

   d. **(7 marks)** The Team/Day pairs from GamesB such that the team scored the maximum number of goals for that day, and the team is a winner in GamesA.

Q5. (10 marks) Consider the following SQL trigger on relation R(a,b):

```
CREATE TRIGGER T
AFTER INSERT ON R
REFERENCING NEW ROW AS Newtuple
FOR EACH ROW
WHEN(Newtuple.a * Newtuple.b > 0)
INSERT INTO R VALUES(Newtuple.a - 1, Newtuple.b + 1);
```

When we insert a tuple into R, the trigger may cause another tuple to be inserted, which may cause yet another tuple to be inserted, and so on, until finally a tuple is inserted that does not cause the trigger to fire. Describe the behaviour of this trigger if the tuple (3,5) is inserted into R(a,b).

Consider the relation schema Sells(bar, beer, price).

Q6. (10 marks) Write a PL/SQL procedure Update(x, y, z) that adds the beer *y* to bar *x* at price z in relation Sells.

Q7. (10 marks) Write the PL/SQL procedure PriceUp(*x*) which sends a cursor through the relation Sells, and raises by 2.00 the price of each beer sold by bar *x*, if that price was initially under 5.00.

END OF EXAM