CMPUT 291 - Midterm Exam (20%)

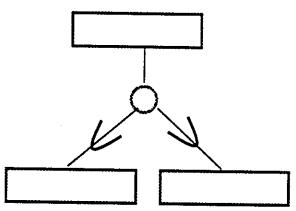
Instructor: *E. Elmallah*

Questions: 3
Time: 70 minutes
Closed Book
- ***No questions during exam time.***
- ***If you are unsure, write down your assumptions.***

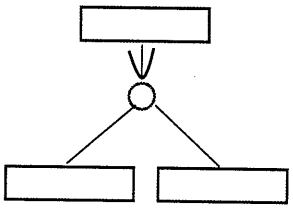
Date: March 6, 2001
Total Pages: 2 + an appendix
Total Marks: 80
Calculators: allowed

Question 1: [27 marks]

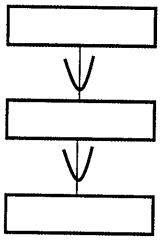
For each of the following diagrams indicate whether it represents a valid E-R diagram, or not. If the diagram is invalid explain the reason. If the diagram is valid, redraw the diagram and fill in the empty boxes, diamonds, and ellipses with suitable data. Briefly explain your answer.



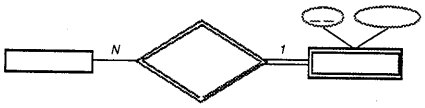
(a)



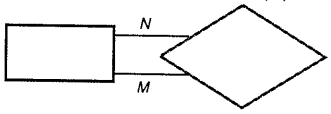
(b)



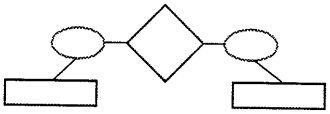
(c)



(d)



(e)



(f)

Question 2: [13 marks]

Draw an E-R diagram of an example specialization where the subclasses overlap. Label all types and attributes. Briefly explain the context that you are modelling, and the suitability of using overlapping specialization to model the context. Devise a database schema that implements your E-R diagram. Identify all primary keys.

Question 3: [40 marks] Consider the following simple university database schema for storing information of students taking courses offered by departments in various faculties.

student (s_id, s_last, s_first, s_deptName, s_gpa)
course (c_number, c_session, c_instr, c_deptName, c_size, c_maxSize)
takes (t_sid, t_cNumber)
department (d_deptName, d_facultyName)

Attribute names have intuitive meaning. For example, `s_id` (also `t_sid`) refers to the student identification number, `c_number` (also `t_cNumber`) is the course number, `s_last` is the student's last name, `c_session` is the date of offering a course, `c_instr` is the instructor's name, `c_size` is the current enrollment, and `c_maxSize` is the maximum allowed enrollment. Refer to the attached Appendix for the `create` commands, and an instance of the database. Answer the following questions using the given instance.

1. Give a possible sequence of `create` statements to create the database. You don't need to copy the detailed statements given in the Appendix, just outline a possible order for creating the tables.

2. Give a possible sequence of `drop` statements to delete all tables from the database. Use as few `drop` commands as possible.

3. Write in a tabular form the answer to the following query (refer to the given instance):

```
select  d_deptName, c_number, c_instr
from    department, course
where   d_deptName = c_deptName and
        c_instr in (select  c_instr
                    from    course
                    where   c_size >= 60);
```

4. Write in a tabular form the answer to the following query:

```
select  s_id, s_last
from    student
where   not exists (select  *
                    from    takes, course
                    where   s_id = t_sid and
                            t_cNumber= c_number and
                            c_session = '9907');
```

5. Write in a tabular form the answer to the following query:

```
select    d_facultyName, d_deptName, count(*)
from      student, department
where     s_deptName = d_deptName
group by  d_facultyName, d_deptName
having    max(s_gpa) >= 7.7;
```

6. Write in a tabular form the answer to the following **sequence** of statements.

```
delete from course
where       c_number like 'ENG%';
select * from takes;
```

7. Write an SQL query that outputs a table of course numbers for all courses involving a person whose last name is 'Smith', either as a student taking the course, or as an instructor teaching the course.

8. Write an SQL query that outputs a department name, and its associated faculty, only if the department offers at least 10 courses where the student enrollment `c_size` in each course is at least 90% of the maximum allowed enrollment `c_maxSize`.

```
create table student (
        s_id            number,
        s_last          char(10),
        s_first         char(10),
        s_deptName      char(10),
        s_gpa           number,
        primary key (s_id),
        foreign key (s_deptName) references department);


create table course (
        c_number        char(10),
        c_session       char(10),
        c_instr         char(10),
        c_deptName      char(10),
        c_size          number,
        c_maxSize       number,
        primary key (c_number),
        foreign key (c_deptName) references department);


create table takes (
        t_sid           number,
        t_cNumber       char(10),
        primary key (t_sid, t_cNumber),
        foreign key (t_sid) references student on delete cascade,
        foreign key (t_cNumber) references course on delete cascade);


create table department (
        d_deptName      char(10),
        d_facultyName   char(10),
        primary key (d_deptName));
```

```
SQL> select * from department;

D_DEPTNAME D_FACULTYN
---------- ----------
CS         Science
Math       Science
English    Arts
History    Arts
Forestry   Forestry


SQL> select * from course;

C_NUMBER   C_SESSION  C_INSTR    C_DEPTNAME      C_SIZE  C_MAXSIZE
---------- ---------- ---------- ---------- ---------- ----------
CMPUT291   9901       Bunt       CS                 50         70
CMPUT304   9908       Karp       CS                 50         70
MATH200    9704       Syslo      Math               80        300
MATH300    9605       Cantor     Math               40         70
ENG100     9904       Murray     English            70        100
HIST100    9405       Smith      History            40        200
FOREST100  9907       Rose       Forestry           40         60


SQL> select * from student;

      S_ID S_LAST     S_FIRST    S_DEPTNAME       S_GPA
---------- ---------- ---------- ---------- ----------
         1 Talor      Jason      CS                8.7
         2 Maher      William    Math              7.5
         3 Wong       Andrew     English           8.1
         4 Smith      Alan       Forestry          8.3
         5 Pike       Ronald     CS                7.8

SQL> select * from takes;

     T_SID T_CNUMBER
---------- ----------
         1 CMPUT291
         1 CMPUT304
         1 ENG100
         2 CMPUT304
         2 MATH200
         2 HIST100
         3 MATH300
         3 ENG100
         3 FOREST100
         4 MATH300
         4 ENG100
         4 FOREST100
```