CSE 880 Midterm Sample Exam

1. Extend the following EER diagram to include students GRE Scores, Student’s grade in a course, Student’s GPA, average grade of a course and average grade of a student in courses offered by her/his own department.

![EER Diagram](image)

2. Give an example of an object schema for a Students database with the following features. Use graphical representation for your answer. For each class show the inherited attributes:

   (a) One interface class StudentIF

   (b) Four classes A: Student, B: Employee, C: StudentEmployee and D: Person. A and B are subclasses of the interface class StudentIF as well as subclasses of D. C is a subclass of both A and B (multiple inheritance). A redefines an inherited attribute and C inherits an attribute from both A and B with the same origin.

   Answers:

   A and B must inherit attributes from D and not from IF (IF attributes cannot be inherited). Because C inherits attributes from A and B with the same origin, origin for this attribute must be D.

3. Assume the Project-Task object-database schema diagram attached at the end for this exam:
(a) Write in plain English what the following OQL statement achieves.

Select P. project_name, P.objective, P.sub_project, P.balance()
Documents: (Select D.acronym_document, D.name, D.classification
From P.Documents D)
Tasks: (Select T.date_start, T.date_end, T.description_task,
    T.participating.group_name, T.leader,
Members: (Select M.name, M.specialization, M.salary,
    M.production, M.average_salary,
    M.monthly_salary()
From T.participating.members M)
From P.workplan T)
From Project P, P.workplan W
Where EXISTS (Select *
From W
Where W.leader.name="John R")

(b) Give OQL statement for the following queries:

i. Get all project names for those projects which have project
document "database" and at least one of the tasks with par-
ticipating group leader "John R".

ii. Single valued attributes:
Get descriptions of those tasks with participating Group
leader name is John R.
Select T.description_task there is no structure here
From Tasks T
Where T.participating.leader.name="John R"
NOTE: In traditional SQL it required joins because there is
no notion of object id or pointers.

iii. Set valued attributes:
A. Select project names of those projects whose work-
plan has a task with leader "John R"
SELECT P.project_name
FROM Projects P
WHERE "John R" IS IN (SELECT T.leader.name
FROM P.work_plan T)

B. Select project names and Task descriptions of those
projects where the task leader is "John R"
Select Struct (Project_Name: P.project_name, TaskDescriptions:
Select Struct (T.description_task)
From P.work_plan T)

FROM Projects P
WHERE "John R" IS IN (SELECT UNIQUE T.leader.name
FROM P.work_plan T)

4. Answer the following questions for Oracle Object-Relational database.

(a) Define an object-relational schema, using nested tables, for the
following entity set:
Students(sid, addresses, TelNo)
sid is the student ID, addresses and TelNos are multi-valued.

(b) Consider the following SQL statement:
Select *
From THE(Select addresses
From Students S
Where S.sid=1234) A
Where A.city= "East Lansing"

   i. What does the keyword THE accomplishes?
   ii. Give the meaning of the above SQL statement in plain En-
glish.

(c) Give an SQL statement for inserting telephone number 517-333-
3333 into the nested table (tiny table) for the student with sid=1234.
INSERT INTO THE(Select TelNo from Students S where S.sid=1234)
VALUES(telNo('517-333-3333'));

5. In Object relational databases a table can be viewed as a relation or a
table of objects where each row of the table can be viewed as an object.
For example, following table books can be a relation or a table with
each row as an object:
CREATE TYPE bookType AS OBJECT(
   name varchar2(30),
publisher varchar2(30));
CREATE TABLE books of bookType;

Following two SELECT statements on defined on this table:
What does the VALUE(p) in the second SELECT do compared to * in
the first SELECT?
1) SELECT * FROM books
2) SELECT VALUE(p) FROM books p

The VALUE(p) returns rows as objects of type bookType
The 'Select *' returns rows as tuples in a regular relation.

6. Following is an Oracle example of using table employee as a relation as well as a table of objects. Note that Programmer is an Employee and Representative is an Employee:

CREATE TYPE Employee AS OBJECT (
    Name VARCHAR2(20),
    Salary NUMBER(6,2)
) NOT FINAL;
CREATE TYPE Programmer UNDER Employee (
    Language VARCHAR2(12),
    Project VARCHAR2(30)
);
CREATE TYPE Representative UNDER Employee (
    Region VARCHAR2(30)
);
CREATE TABLE employees OF Employee;

INSERT INTO employees
    VALUES (Employee('Sylvia Karsen', 3000.00));
INSERT INTO employees
    VALUES (Programmer('William Helprin', 4000.00, 'C++', 'Database'));
INSERT INTO employees
    VALUES (Representative('Akiko Yokomoto', 5000.00, 'Asia'));

(a) What will the following SELECT statement return?

    select * from employees;

<table>
<thead>
<tr>
<th>NAME</th>
<th>SALARY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sylvia Karsen</td>
<td>3000</td>
</tr>
<tr>
<td>William Helprin</td>
<td>4000</td>
</tr>
<tr>
<td>Akiko Yokomoto</td>
<td>5000</td>
</tr>
</tbody>
</table>

(b) What will the following SELECT statement return?

    select *
    from employees e
    where value(e) is of (programmer);
7. Consider the following temporal relation where valid start time and valid end time are two separate attributes of the table. Thus, valid start and end times are a part of the primary key of the table.

Prescription Table:

<table>
<thead>
<tr>
<th>Name</th>
<th>Drug</th>
<th>Physician</th>
<th>Vst</th>
<th>Vet</th>
</tr>
</thead>
<tbody>
<tr>
<td>n1</td>
<td>d1</td>
<td>p1</td>
<td>5</td>
<td>10</td>
</tr>
<tr>
<td>n1</td>
<td>d2</td>
<td>p2</td>
<td>20</td>
<td>28</td>
</tr>
<tr>
<td>n1</td>
<td>d2</td>
<td>p2</td>
<td>8</td>
<td>25</td>
</tr>
<tr>
<td>n1</td>
<td>d3</td>
<td>p2</td>
<td>30</td>
<td>35</td>
</tr>
<tr>
<td>n1</td>
<td>d1</td>
<td>p1</td>
<td>29</td>
<td>36</td>
</tr>
</tbody>
</table>

(a) Write the table as a TSQL2 table where a temporal element is a set of maximal non-overlapping intervals.

(b) Give the TSQL2 table resulting from the following TSQL2 statement. Include the temporal elements.

```
SELECT Name
FROM Prescription
```

(c) Give only the affected and the newly created tuples of the above temporal relation Prescription (resulting from question in part-a of this problem) after the following UPDATE transaction is executed.

Transaction:

```
UPDATE Prescription
SET physician TO 'P1'
VALID PERIOD '8,28'
WHERE Name='n1' AND Drug = 'd2'
```

(d) Consider the following TSQL2 statement:

```
SELECT SNAPSHOT P1.Name, VALID(P1)
FROM Prescription(Name, Drug) AS P1, P1(Drug) AS (PERIOD) P1',
Prescription(Name, Drug)AS P2, P2(Drug) AS (PERIOD) P2'
WHERE P1.Name=p2.Name & P1.Drug != P2.Drug & VALID(P1)=VALID(P2) &
CAST(VALID(P1') AS INTERVAL MONTH)> INTERVAL '6' MONTH &
CAST(VALID(P2') AS INTERVAL MONTH)> INTERVAL '6' MONTH
```

i. If PERIOD is omitted from the above statement will VALID(P1) be a set of time intervals or a single time interval. When PERIOD is omitted, will SNAPSHOT be able to create a table. Explain your answer.
ii. Describe in plain English the meaning of the above TSQL2 statement.
Schema for the above database defined in ODL (inverse relationships omitted) is given below:

class Project
(  extent Projects)
{  attribute string  project_name;
  attribute string  objective;
  relationship set<Document>  document;
  relationship set<Task>  work_plan;
  relationship <Project>  sub_project;

  number balance();
  set<Group>  participating();
};

class Task
(  extent Tasks)
{  attribute string  date_start;
  attribute string  date_end;
  attribute string  description_task;
  relationship set<Group>  participating;
  attribute number  man_year;
  relationship <Researcher>  leader;
  relationship <Task>  precedes;
};
class Group
(    extent Groups)
{    attribute string group_name;
    relationship set<Researcher> member;
    relationship <Researcher> leader;
};

class Researcher
(    extent Researchers)
{    attribute string name;
    attribute string specialization ;
    attribute number salary;
    attribute number production_bonus;
    attribute average_salary:number;
    relationship number monthly_salary();
};

class Document
(    extent Documents)
{    attribute string acronym_document;
    attribute string name;
    attribute string classification;
};

class TechnicalReport extends Document
(    extent TechnicalReports)
{    attribute string topics;
    attribute DATE start_validity;
    attribute DATE end_validity;
    relationship <TechnicalReport> amendment_to;
};

class Article extends Document
(    extent Articles)
{    attribute string publication_type;
    attribute string publication_place;
    attribute string date;
    relationship set<Researcher> author;
};