**COMP 3005B  
Assignment #6   
Due: Dec 5**

**Instruction**

1. You should do the assignment independently. If copying is found, the case will be reported to the office of the Dean of Science immediately.
2. Do this assignment directly on this document, rename it with your last name+first name, and submit it to **brightspace**. Make sure your uploaded file can be opened and is correct. No submission will be accepted after the deadline no matter what reason.
3. Replace Last in the table below with your own last name. If your last name is not shown correctly in the result, you will get a 0 mark for the assignment.
4. You need to use [Oracle VM](https://git.scs.carleton.ca/downloads/CourseVirtualMachines/2022F-2023W/COMP3005-W23.ova) to do this assignment and take proper screenshots of execution results for each question. If there is no screenshot, you will get 0 for the question.
5. All questions for this assignment are based on the following nested relation.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Department** |  | |  | |
| Name | | Professors | | | | | |
| Name | | Hobbies | | Students | |
| Name | Hobbies |
| Computer | | James | | Drawing | | Adams | Yoga |
| Skating |
|  | | David | Singing |
| Singing | | Rodelo | Reading |
|  | | Singing |
| Henry | | Boating | | Maria | Dancing |
| Kevin | Skating |
| Fishing | | Reading |
| Robin | Ski |

**Part 1. Nested Relational Databases (60)**

1. Represent the schema of the relation as a composition hierarchy. (10)

A diagram of a group of people

Description automatically generated

1. Represent the instance of the relation as a set of tuples. (10)

{[Computer, James, Drawing, Adams, Yoga], [Computer, James, Drawing, Adams, Skating], [Computer, James, Singing, David, Singing], [Computer, James, Singing, Rodelo, Reading], [Computer, James, Singing, Rodelo, Singing], [Computer, Henry, Boating, Maria, Dancing], [Computer, Henry, Boating, Maria, Dancing], [Computer, Henry, Boating, Kevin, Skating], [Computer, Henry, Fishing, Kevin, Reading], [Computer, Henry, Fishing, Robin, Ski]}

1. Use SQL to create the schema of this relation. (10)
2. create type shobbies\_v as varray(5) of varchar2(10);

/

1. create type students\_t as object (Name varchar2(15), Hobbies shobbies\_v);

/

1. create type students\_n as table of students\_t;

/

1. create type phobbies\_v as varray(5) of varchar2(10);

/

1. create type professors\_t as object(Name varchar2(15), Hobbies phobbies\_v, Students students\_n);

/

1. create type professors\_n as table of professors\_t;

/

1. create table Department (Name varchar2(15), Professors professors\_n) nested table professors store as professor\_tb(nested table students store as students\_tb());

1. Use SQL to population this relation with the information given in the table. (10)

SQL statement:

insert into Department values('Computer',

professors\_n(

professors\_t('James', phobbies\_v('Drawing', 'Singing'),

students\_n(

students\_t('Adams',

shobbies\_v('Yoga', 'Skating')),

students\_t('David',

shobbies\_v('Singing')),

students\_t('Rodelo',

shobbies\_v('Reading', 'Singing')))),

professors\_t('Henry', phobbies\_v('Boating', 'Fishing'),

students\_n(

students\_t('Maria',

shobbies\_v('Dancing')),

students\_t('Kevin',

shobbies\_v('Skating', 'Reading')),

students\_t('Robin',

shobbies\_v('Ski')) ))));

Result:

A computer screen shot of text

Description automatically generated

1. Use SQL to express the following queries. The result should be a set of values or tuples but not a set of sets. (20)
2. Display the nested relation as a 1NF relation.

SQL expression:

select d.Name as Department\_Name,

p.Name as Professor\_Name,

ph.Name as Student\_Name,

sh.column\_value as Student\_Hobby,

shh.column\_value as Professor\_Hobby

from Department d,

table(d.Professors) p,

table(p.Students) ph,

table(ph.hobbies) sh,

table(p.hobbies) shh;

Result:

A screenshot of a computer

Description automatically generated

1. List every distinct hobby

SQL expression:

select distinct Hobby from(

select column\_value as Hobby

from Department d,

table(d.Professors) p,

table(p.Hobbies) ph

union all

select column\_value as Hobby

from Department d,

table(d.Professors) p,

table(p.Students) s,

table(s.Hobbies) sh );

Result:

A screenshot of a computer program

Description automatically generated

1. List every professor together with his/her students in a nested relation.

SQL expressions:

create type snames\_v as varray(5) of varchar(10);

/

select p1.name,

cast(

multiset(

select distinct s.name from Department d,

table(d.professors) p,

table(p.students) s where p1.name = p.name) as snames\_v)

as students from

(select distinct p.name from Department d, table(d.professors) p) p1;

Result:

A computer screen shot of a computer code

Description automatically generated

1. List every professor together with his/her students with a common hobby in a nested relation.

**Part 2 Object Relational Databases (40)**

1. Use SQL to create an object-relational database with the information in the above nested tables by properly defining types and subtypes. Your database should just have two object tables Department and Person where Person is a substitutable table that contains both professors and students. You need to use the ID of the persons for their relationships in the Department relation. (10)

SQL expression:

* Creating persons table:

create type o\_hobbies\_v as varray(5) of varchar2(10);

/

create type o\_person\_t as object(name varchar2(10), hobbies o\_hobbies\_v) not final;

/

create type o\_professors\_t under o\_person\_t();

/

create type o\_students\_t under o\_person\_t();

/

* Creating department table:

create type o\_students\_v as varray(5) of ref o\_students\_t;

/

create type o\_professors\_dept\_t as object(Professor ref o\_professors\_t, Students o\_students\_v);

/

create type o\_professors\_dept\_n as table of o\_professors\_dept\_t;

/

create type o\_department\_t as object(Name varchar2(10), Professors o\_professors\_dept\_n);

/

create table o\_person of o\_person\_t;

create table o\_department of o\_department\_t nested table professors store as professors\_tb;

1. Use SQL to populate this database with the information in the above relation. (10)

SQL expressions to populate o\_person:

insert into o\_person values (o\_professors\_t ('James', o\_hobbies\_v ('Drawing', 'Singing')));

insert into o\_person values (o\_students\_t ('Adams', o\_hobbies\_v ('Yoga', 'Skating')));

insert into o\_person values (o\_students\_t ('David', o\_hobbies\_v ('Singing')));

insert into o\_person values (o\_students\_t ('Rodelo', o\_hobbies\_v ('Reading', 'Singing')));

insert into o\_person values (o\_professors\_t ('Henry', o\_hobbies\_v ('Boating', 'Fishing')));

insert into o\_person values (o\_students\_t ('Maria', o\_hobbies\_v ('Dancing')));

insert into o\_person values (o\_students\_t ('Kevin', o\_hobbies\_v ('Skating', 'Reading')));

insert into o\_person values (o\_students\_t ('Robin', o\_hobbies\_v ('Ski')));

SQL expression to populate o\_department:

insert into o\_department values(

'Computer',

o\_professors\_dept\_n(

o\_professors\_dept\_t(

(select treat(ref(P) as ref o\_professors\_t)

from o\_person P where value(P) is of (o\_professors\_t) and P.Name = 'James'),

o\_students\_v(

(select treat(ref(S) as ref o\_students\_t)

from o\_person S where value(S) is of (o\_students\_t) and S.Name = 'Adams'),

(select treat(ref(S) as ref o\_students\_t)

from o\_person S where value(S) is of (o\_students\_t) and S.Name = 'David'),

(select treat(ref(S) as ref o\_students\_t)

from o\_person S where value(S) is of (o\_students\_t) and S.Name = 'Rodelo'))

),

o\_professors\_dept\_t(

(select treat(ref(P) as ref o\_professors\_t)

from o\_person P where value(P) is of (o\_professors\_t) and P.Name = 'Henry'),

o\_students\_v(

(select treat(ref(S) as ref o\_students\_t)

from o\_person S where value(S) is of (o\_students\_t) and S.Name = 'Maria'),

(select treat(ref(S) as ref o\_students\_t)

from o\_person S where value(S) is of (o\_students\_t) and S.Name = 'Kevin'),

(select treat(ref(S) as ref o\_students\_t)

from o\_person S where value(S) is of (o\_students\_t) and S.Name = 'Robin'))

)));

Screenshots:

A screenshot of a computer program

Description automatically generated

A computer screen shot of a program

Description automatically generated

1. Use SQL to represent the following queries. (20)
2. List every distinct hobby.

SQL expression:

Select distinct hobby from (select column\_value as hobby from o\_person p, table(p.hobbies) h where value(p) is of (o\_professors\_t, o\_students\_t));

Result:

A screenshot of a computer program

Description automatically generated

1. List every professor together with his/her students in a nested relation.

SQL expression:

create type names\_v as varray(5) of varchar(10);

/

select deref(P1.professor).name as professor,

cast(multiset(

select distinct deref(S.column\_value).name

from o\_department D,

table(D.professors) R,

table(R.students) S

where P1.professor = R.professor) as names\_v) as Students

from(

select ref(P) as professor from o\_person P where value(P) is of (o\_professors\_t))P1;

Result:

A computer screen shot of a program

Description automatically generated

1. List every professor together with his/her students with a common hobby in a nested relation.

SQL expression:

1. List every department and its professors together with his/her students in a nested relation.

select d.name as departmentName,

cast(multiset(

select deref(P1.professor).name as professor,

cast(multiset(

select distinct deref(S.column\_value).name

from o\_department D,

table(D.professors) R,

table(R.students) S

where P1.professor = R.professor) as names\_v) as Students

from(

select ref(P) as professor from o\_person P where value(P) is of (o\_professors\_t))P1

)as professors\_n) as professors

from o\_department d;