

COMP 3005B

Assignment #6

Due: Dec 5

Instruction

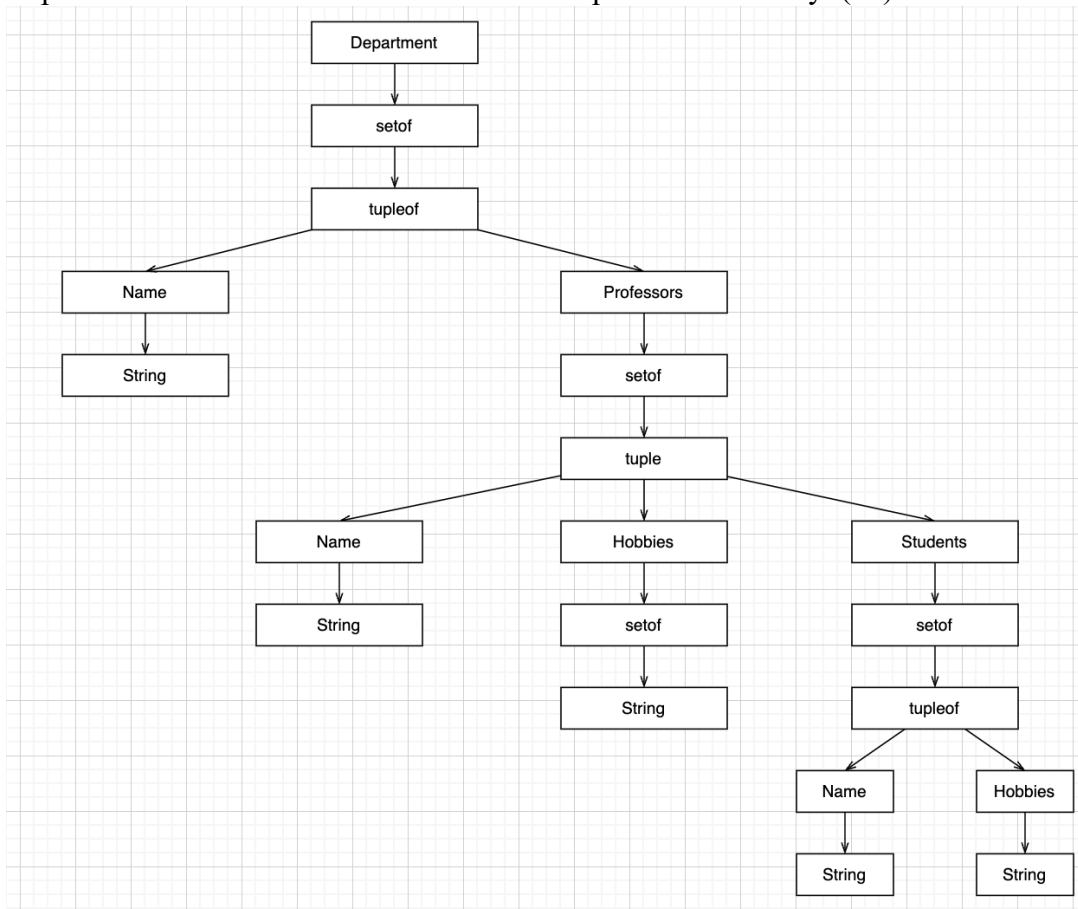
1. You should do the assignment independently. If copying is found, the case will be reported to the office of the Dean of Science immediately.
2. Do this assignment directly on this document, rename it with your last name+first name, and submit it to **brightspace**. Make sure your uploaded file can 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](#) 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
		Singing	David	Singing
			Rodelo	Reading
	Henry	Boating	Maria	Singing
				Dancing
		Fishing	Kevin	Skating
			Robin	Reading

Part 1. Nested Relational Databases (60)

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



2. 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]}

3. Use SQL to create the schema of this relation. (10)

1. create type shobbies_v as varray(5) of varchar2(10);
/
2. create type students_t as object (Name varchar2(15), Hobbies shobbies_v);
/
3. create type students_n as table of students_t;
/
4. create type phobbies_v as varray(5) of varchar2(10);
/
5. create type professors_t as object(Name varchar2(15), Hobbies phobbies_v, Students students_n);

- ```

/
6. create type professors_n as table of professors_t;
/
7. create table Department (Name varchar2(15), Professors professors_n) nested table
 professors store as professor_tb(nested table students store as students_tb());
4. 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:

```

SQL> select * from department
2 ;

NAME

PROFESSORS(NAME, HOBBIES, STUDENTS(NAME, HOBBIES))

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')))), PROFES
SORS_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'))))

```

5. Use SQL to express the following queries. The result should be a set of values or tuples but not a set of sets. (20)
  - a. 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:

```

SQL> 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;

```

| DEPARTMENT_NAME | PROFESSOR_NAME | STUDENT_NAME | STUDENT_HO | PROFESSOR_ |
|-----------------|----------------|--------------|------------|------------|
| Computer        | James          | Rodelo       | Reading    | Drawing    |
| Computer        | James          | Rodelo       | Singing    | Drawing    |
| Computer        | James          | David        | Singing    | Drawing    |
| Computer        | James          | Adams        | Yoga       | Drawing    |
| Computer        | James          | Adams        | Skating    | Drawing    |
| Computer        | James          | Rodelo       | Reading    | Singing    |
| Computer        | James          | Rodelo       | Singing    | Singing    |
| Computer        | James          | David        | Singing    | Singing    |
| Computer        | James          | Adams        | Yoga       | Singing    |
| Computer        | James          | Adams        | Skating    | Singing    |
| Computer        | Henry          | Robin        | Ski        | Boating    |
| Computer        | Henry          | Kevin        | Skating    | Boating    |
| Computer        | Henry          | Kevin        | Reading    | Boating    |
| Computer        | Henry          | Maria        | Dancing    | Boating    |
| Computer        | Henry          | Robin        | Ski        | Fishing    |
| Computer        | Henry          | Kevin        | Skating    | Fishing    |
| Computer        | Henry          | Kevin        | Reading    | Fishing    |
| Computer        | Henry          | Maria        | Dancing    | Fishing    |

18 rows selected.

- b. 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:

```

SQL> 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);

```

| 2                | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|------------------|---|---|---|---|---|---|---|----|----|
| HOBBY            |   |   |   |   |   |   |   |    |    |
| -----            |   |   |   |   |   |   |   |    |    |
| Boating          |   |   |   |   |   |   |   |    |    |
| Singing          |   |   |   |   |   |   |   |    |    |
| Ski              |   |   |   |   |   |   |   |    |    |
| Fishing          |   |   |   |   |   |   |   |    |    |
| Drawing          |   |   |   |   |   |   |   |    |    |
| Yoga             |   |   |   |   |   |   |   |    |    |
| Skating          |   |   |   |   |   |   |   |    |    |
| Dancing          |   |   |   |   |   |   |   |    |    |
| Reading          |   |   |   |   |   |   |   |    |    |
| 9 rows selected. |   |   |   |   |   |   |   |    |    |

- c. 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:

```

SQL> select p1.name,
cast(
multiset(
select distinct s 2 3 4 .name from Department d,
table(d.professors) p,
5 6 table(p.students) s where p1.name = p.name) as snames_v)
as students from
(select distinct p.name 7 8 from Department d, table(d.professors) p) p1;

NAME

STUDENTS

Henry
SNAMES_V('Maria', 'Kevin', 'Robin')
James
SNAMES_V('Adams', 'Rodelo', 'David')

```

- d. 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;
```

2. 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:

```

SQL> select * from o_person;

NAME

HOBBIES

James
O_HOBBIES_V('Drawing', 'Singing')

Adams
O_HOBBIES_V('Yoga', 'Skating')

David
O_HOBBIES_V('Singing')

NAME

HOBBIES

Rodelo
O_HOBBIES_V('Reading', 'Singing')

Henry
O_HOBBIES_V('Boating', 'Fishing')

Maria
O_HOBBIES_V('Dancing')

NAME

HOBBIES

Kevin
O_HOBBIES_V('Skating', 'Reading')

Robin
O_HOBBIES_V('Ski')

8 rows selected.

SQL> select * from o_department;

NAME

PROFESSORS(PROFESSOR, STUDENTS)

Computer
O_PROFESSORS_DEPT_N(O_PROFESSORS_DEPT_T(00002202080B8E58C26B75CB5DE060A8C0053803
C50B8E58C26B72CB5DE060A8C0053803C5, O_STUDENTS_V(00002802090B8E58C26B76CB5DE060A
8C0053803C50B8E58C26B72CB5DE060A8C0053803C50040B1E10001, 00002802090B8E58C26B77C
B5DE060A8C0053803C50B8E58C26B72CB5DE060A8C0053803C50040B1E10002, 00002802090B8E5
8C26B78CB5DE060A8C0053803C50B8E58C26B72CB5DE060A8C0053803C50040B1E10003)), O_PRO
FESSORS_DEPT_T(00002202080B8E58C26B79CB5DE060A8C0053803C50B8E58C26B72CB5DE060A8C
0053803C5, O_STUDENTS_V(00002802090B8E58C26B7ACB5DE060A8C0053803C50B8E58C26B72CB
5DE060A8C0053803C50040B1E10005, 00002802090B8E58C26B78CB5DE060A8C0053803C50B8E58
C26B72CB5DE060A8C0053803C50040B1E10006, 00002802090B8E58C26B7CCB5DE060A8C0053803
C50B8E58C26B72CB5DE060A8C0053803C50040B1E10007)))

NAME

PROFESSORS(PROFESSOR, STUDENTS)

C26B72CB5DE060A8C0053803C50040B1E10006, 00002802090B8E58C26B7CCB5DE060A8C0053803
C50B8E58C26B72CB5DE060A8C0053803C50040B1E10007)))

```

3. Use SQL to represent the following queries. (20)

a) 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:

```
SQL> SELECT DISTINCT hobby
FROM (
 SELECT COLUMN_VAL 2 3 UE AS hobby
 FROM o_person p,
 TABLE(p. 4 5 hobbies) h
 WHERE VALUE(p) IS OF (o_professors_ 6 t, o_students_t)
);
SQL> SELECT DISTINCT hobby
FROM (
 SELECT COLUMN_VAL 2 3 UE AS hobby
 FROM o_person p,
 TABLE(p. 4 5 hobbies) h
 WHERE VALUE(p) IS OF (o_professors_ 6 t, o_students_t)
);
7
HOBBY

Boating
Singing
Ski
Fishing
Drawing
Yoga
Skating
Dancing
Reading

9 rows selected.
```

- b) 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:

```
SQL> select deref(P1.professor).name as professor,
cast 2 (multiset(
select distinct deref(S.column_value).n 3 ame
from o_department D, table(D.professors) R, ta 4 ble(R.students) S
where P1.professor = R.professor 5) as names_v) as Students
from(
select ref(P) as 6 7 professor from o_person P where value(P) is of (o_professors_t))P1;
PROFESSOR
STUDENTS

James
NAMES_V('Adams', 'Rodelo', 'David')
Henry
NAMES_V('Maria', 'Kevin', 'Robin')
```

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

SQL expression:

- d) 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;
```