**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 | | Last | 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)   
   There are total 21 boxes, missing one deduct 1 mark, upto 5 marks.

Most errors are missing tupleof as shown below, deduct 2 marks for each missing one.

A diagram of a company

Description automatically generated

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

Department = {

[Name: Computer,

Professors: {

[Name: James, Hobbies: {Drawing, Singing},

Students: {[Name: Adams, Hobbies: {Yoga, Skating}],

[Name: David, Hobbies: {Singing}],

[Name: Last, Hobbies: {Reading, Singing}]}],

[Name: Henry, Hobbies: {Boating, Fishing},

Students: {[Name: Maria, Hobbies: {Dancing}],

[Name: Kevin, Hobbies: {Skating, Reading}],

[Name: Robin, Hobbies: {Ski}]}]

}

]

}

Mainly two kinds of brackets are used: { } represents a set and [ ] a tuple.

If their order is used wrong, deduct 2 marks for each wrong usage. If only one kind is used, deduct 5 marks.   
If there is no instance information, give 0.

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

If partially done, give marks based on the percent of the work done that is correct.

create type Name\_v as varray(5) of Varchar2(10);

/

create type Student\_t as object (

Name Varchar2(10),

Hobbies Name\_v);

/

create type Student\_v as varray(5) of Student\_t;

/

create type Professor\_t as object (

Name Varchar2(10),

Hobbies Name\_v,

Students Student\_v);

/

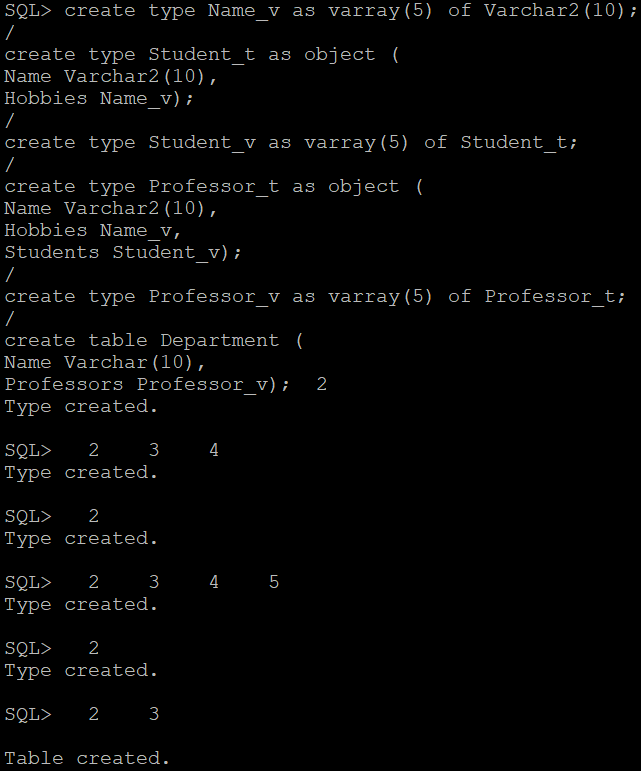
create type Professor\_v as varray(5) of Professor\_t;

/

create table Department (

Name Varchar(10),

Professors Professor\_v);



The nested table can be used instead of Professor\_v as follows:

create type Professor\_n as table of Professor\_t;

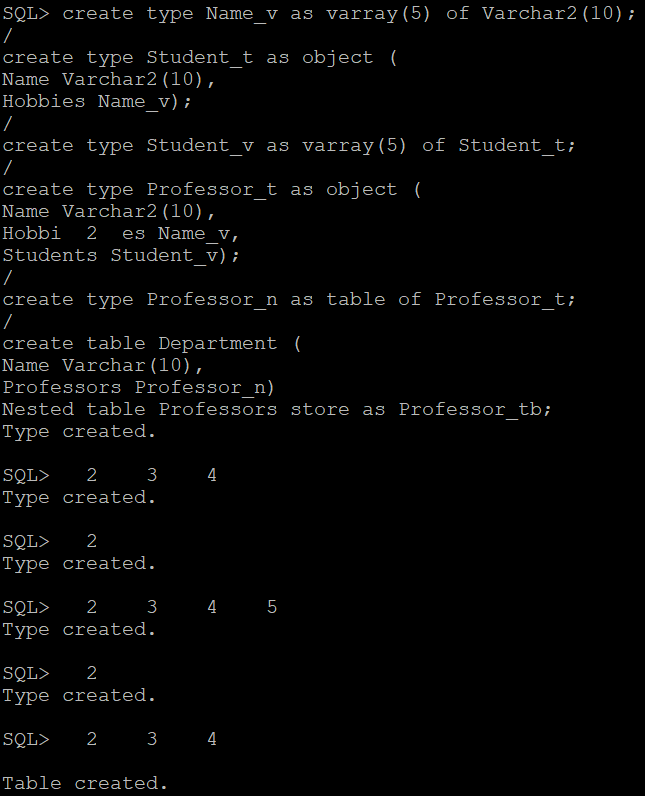
/

create table Department (

Name Varchar(10),

Professors Professor\_n)

Nested table Professors store as Professor\_tb;



If no screenshot, 0

If partially done, give marks based on the percent of the work done that is correct.

Same for all remaining questions.

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

insert into Department values (

'Computer',

Professor\_v (

Professor\_t ('James', Name\_v ('Drawing', 'Singing'), Student\_v (

Student\_t ('Adams', Name\_v ('Yoga', 'Skating')),

Student\_t ('Daving', Name\_v ('Singing')),

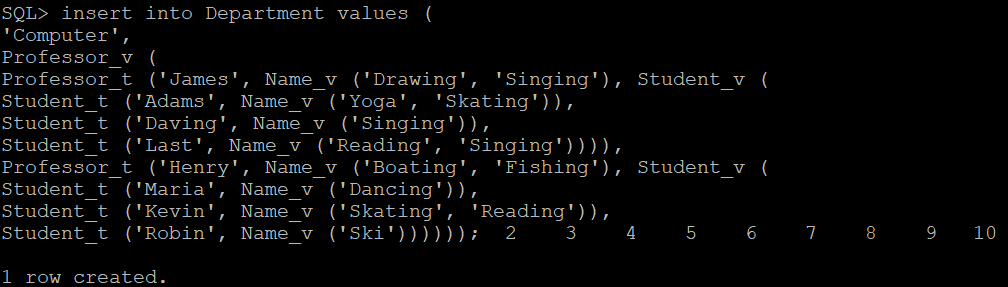
Student\_t ('Last', Name\_v ('Reading', 'Singing')))),

Professor\_t ('Henry', Name\_v ('Boating', 'Fishing'), Student\_v (

Student\_t ('Maria', Name\_v ('Dancing')),

Student\_t ('Kevin', Name\_v ('Skating', 'Reading')),

Student\_t ('Robin', Name\_v ('Ski'))))));



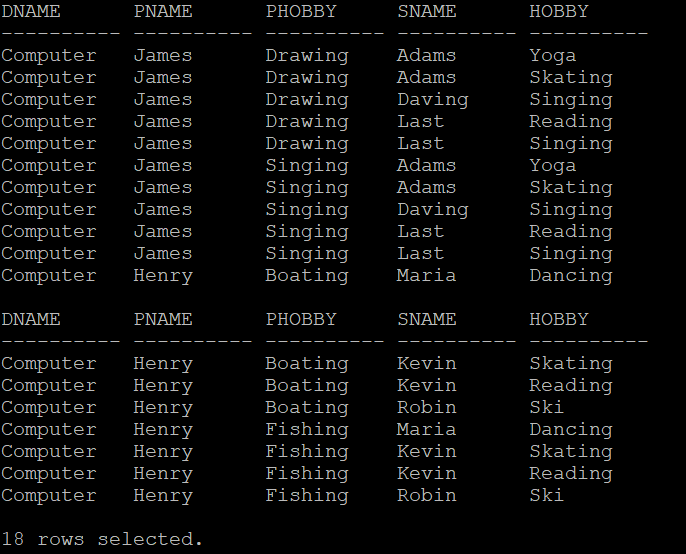
1. Use SQL to express the following queries. The result should be a set of values or tuples but not a set of sets. (20)
2. Display the nested relation as a 1NF relation.

select D.Name as Dname P.Name as Pname, H1.column\_value AS Phobby,

S.Name As Sname, H2.column\_value as Hobby

from Department D, table(D. Professors) P, table(P.Hobbies) H1, table(P.Students) S,

table(S.Hobbies) H2;



AS is not necessary.

1. List every distinct hobby

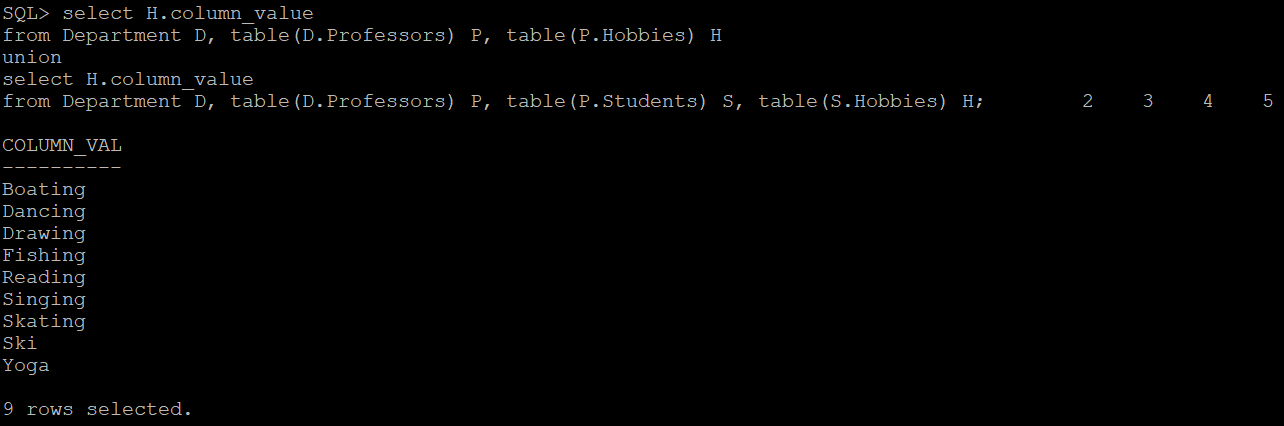
select H.column\_value

from Department D, table(D.Professors) P, table(P.Hobbies) H

union

select H.column\_value

from Department D, table(D.Professors) P, table(P.Students) S, table(S.Hobbies) H;



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

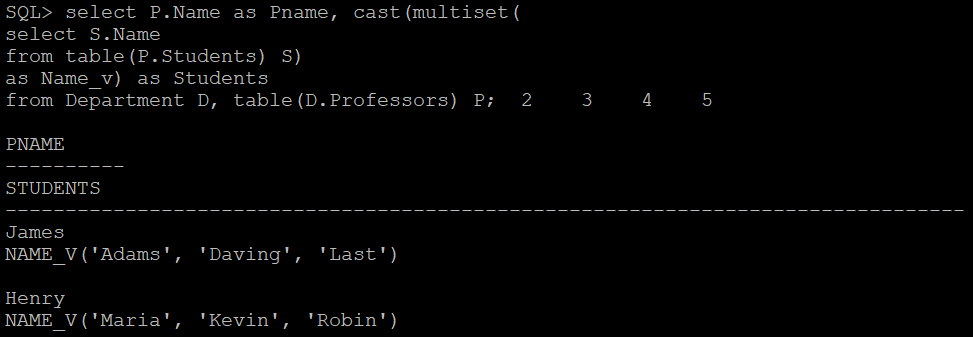
select P.Name as Pname, cast(multiset(

select S.Name

from table(P.Students) S)

as Name\_v) as Students

from Department D, table(D.Professors) P;



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

select P.Name as Pname, cast(multiset(

select S.Name

from table(P.Students) S

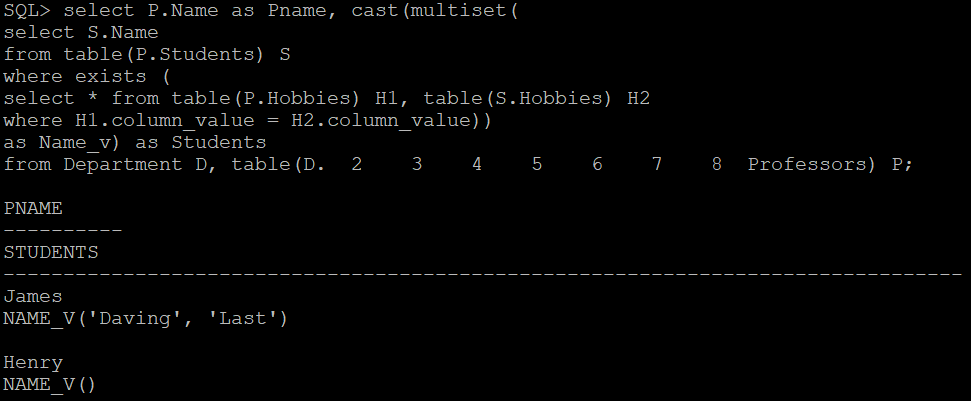
where exists (

select \* from table(P.Hobbies) H1, table(S.Hobbies) H2

where H1.column\_value = H2.column\_value))

as Name\_v) as Students

from Department D, table(D.Professors) P;



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

If no screenshot, 0

If partially done, give marks based on the percent of the work done that is correct.

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)

create type Person\_t as object (Name varchar2(10), Hobbies Name\_v) not final;

/

create type Student\_t under Person\_t ();

/

create type Student\_v as varray(5) of ref student\_t;

/

create type Professor\_t under Person\_t (Students student\_v);

/

create type Professor\_v as varray(5) of ref professor\_t;

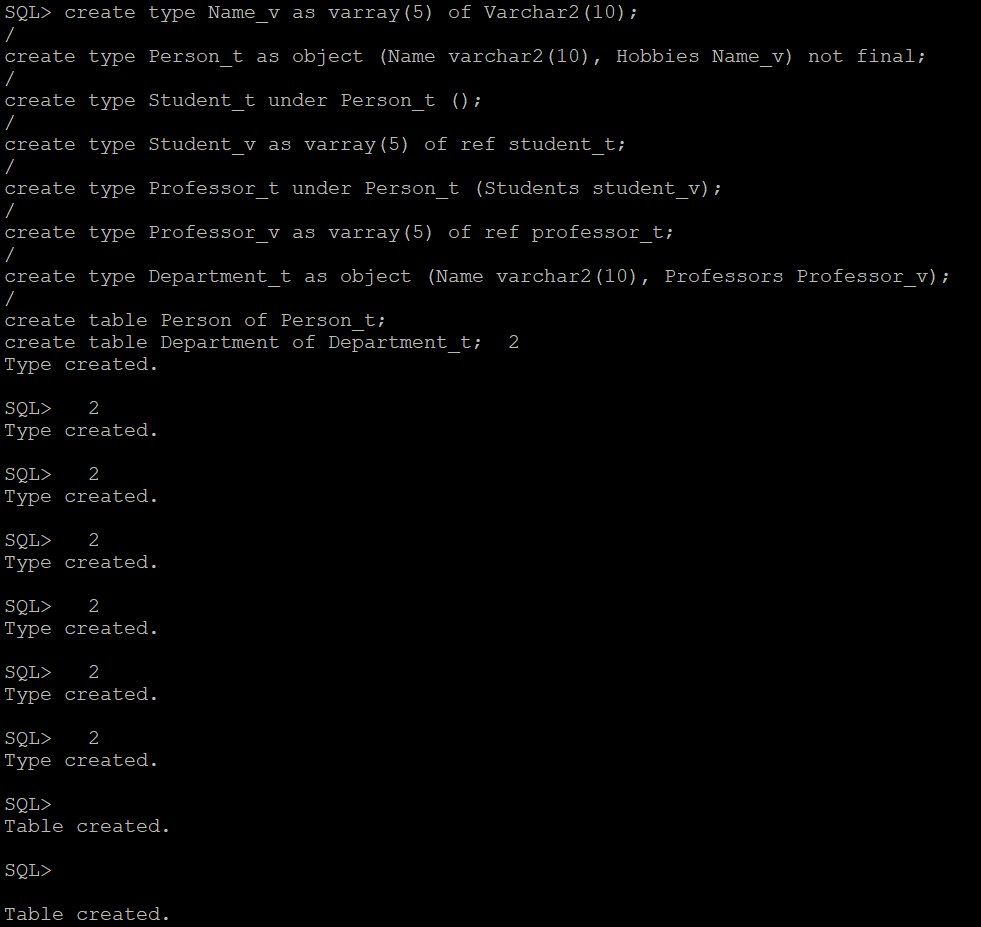
/

create type Department\_t as object (Name varchar2(10), Professors Professor\_v);

/

create table Person of Person\_t;

create table Department of Department\_t;



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

insert into Person values (Student\_t('Adams', Name\_v ('Yoga', 'Skating')));

insert into Person values (Student\_t('David', Name\_v ('Singing')));

insert into Person values (Student\_t('Last', Name\_v ('Reading', 'Singing')));

insert into Person values (Student\_t('Maria', Name\_v ('Dancing')));

insert into Person values (Student\_t('Kevin', Name\_v ('Skating', 'Reading')));

insert into Person values (Student\_t('Robin', Name\_v ('Ski')));

insert into Person values (

Professor\_t(

'James',

Name\_v ('Drawing', 'Singing'),

Student\_v(

(select treat(ref(P) as ref Student\_t) from Person P where Name ='Adams'),

(select treat(ref(P) as ref Student\_t) from Person P where Name ='David'),

(select treat(ref(P) as ref Student\_t) from Person P where Name ='Last')

)

)

);

insert into Person values (

Professor\_t(

'Henry',

Name\_v ('Boating', 'Fishing'),

Student\_v(

(select treat(ref(P) as ref Student\_t) from Person P where Name ='Maria'),

(select treat(ref(P) as ref Student\_t) from Person P where Name ='Kevin'),

(select treat(ref(P) as ref Student\_t) from Person P where Name ='Robin')

)

)

);

insert into Department values(

'Computer',

Professor\_v(

(select treat(ref(P) as ref Professor\_t) from Person P where Name ='James'),

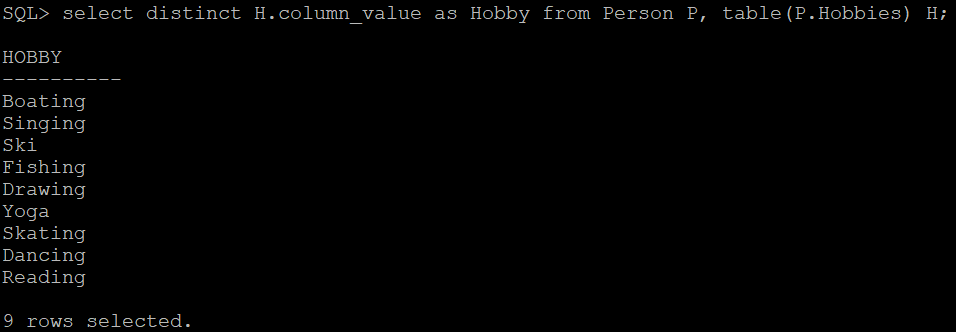
(select treat(ref(P) as ref Professor\_t) from Person P where Name ='Henry')

)

);

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

select distinct H.column\_value as Hobby from Person P, table(P.Hobbies) H;



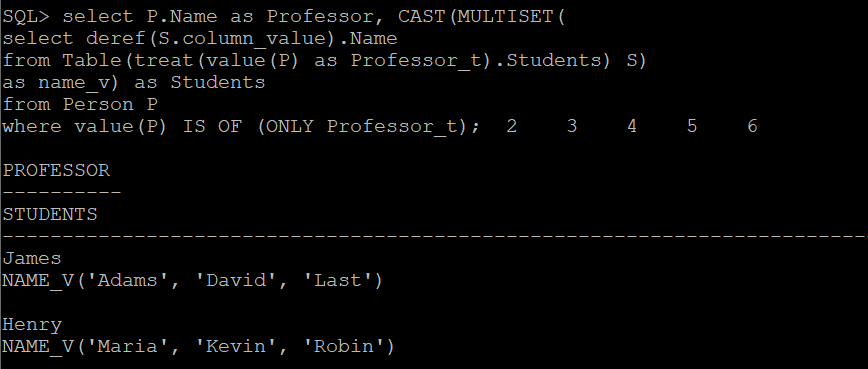
1. List every professor together with his/her students in a nested relation.

select P.Name as Professor, CAST(MULTISET(

select deref(S.column\_value).Name

from Table(treat(value(P) as Professor\_t).Students) S) as name\_v) as Students

from Person P

where value(P) IS OF (ONLY Professor\_t);

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

select P.Name as Professor, CAST(MULTISET(

select deref(S.column\_value).Name

from Table(treat(value(P) as Professor\_t).Students) S,

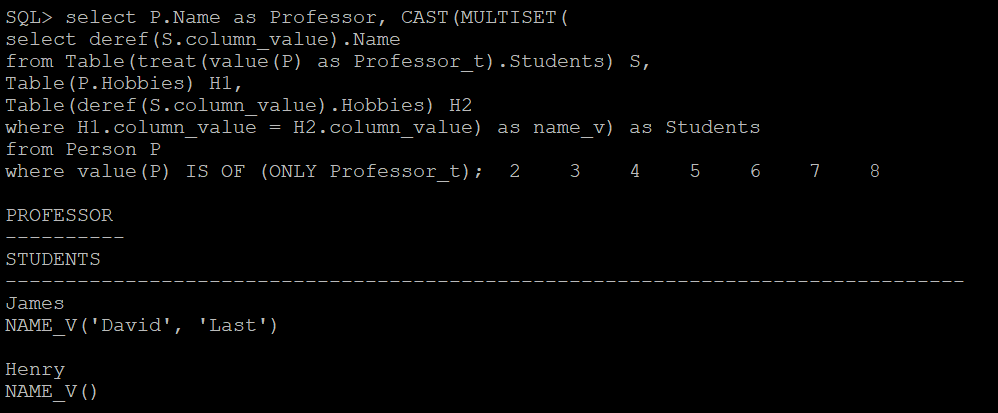
Table(P.Hobbies) H1,

Table(deref(S.column\_value).Hobbies) H2

where H1.column\_value = H2.column\_value) as name\_v) as Students

from Person P

where value(P) IS OF (ONLY Professor\_t);



1. List every department and its professors together with his/her students in a nested relation; i.e, the same as the nested table shown without hobbies for professors and students.

create type result\_t as object (

Pname Varchar2(10),

Students Name\_v);

/

create type result\_v as varray(5) of result\_t;

/

select D.Name, CAST(MULTISET(

select deref(P.column\_value).Name, CAST(MULTISET(

select deref(S.column\_value).Name

from Table(treat(value(P) as ref Professor\_t).Students) S) as name\_v) as Students

from Table(D.Professors) P) as result\_v) as Professors

from Department D;

