School: Computer Science
Institution: University of Windsor
Term: Fall 2023
Course: Comp-3150 (03-60-315-1) : Database Management Systems
Instructor: Dr. C. I. Ezeife
Assignment #3 : Total: 50 marks
Handed Out:Thurs. Oct. 26, 2023; Due: Thurs. Nov. 16, 2023

Objective of Assignment: To test on knowledge and use of relational database query languages SQL and relational algebra for implementing relational databases.
Scope: Assignment covers materials from Chapters 6, 7 and 8 of book discussed in class.
Electronic Assignment Submission: Done through http://brightspace.uwindsor.ca
Marking Scheme : The mark for each of the questions is indicated beside each question.
Academic Integrity Statement: Remember to submit only work that is yours and include the following confidentiality agreement and statement at the beginning of your assignment.

CONFIDENTIALITY AGREEMENT & STATEMENT OF HONESTY

I confirm that I will keep the content of this assignment/examination confidential.
I confirm that I have not received any unauthorized assistance in preparing for or doing this assignment/examination. I confirm knowing that a mark of 0 may be assigned for copied work.

__Sp_____ ___Spondon Sayeed_____
Student Signature Student Name (please print)
__110101278_____ ___2023-11-20_____
Student I.D. Number Date

Marking Scheme : The mark for each question and sub question is shown with the question below. Place your solutions in tables where possible.

For office Use only

Question	Mark
1	/15
2	/10
3	/5
4	/5
5	/5
6	/10
Total	/50

CHAPTER 6: Basic SQX

1. Given a database state of of the Farmer-growscrops-inRegions database shown in Figure 1.1, with schema shown in Figure 1.2, answer question (i) to (v) given below.

(Total for que 1 is 15 marks)

Note : Ssn, Name, Age, Regionid are the social security number, name, age and Regionid of the farmer respectively. Also, Regname, Cropbudget and managerid represent the regionid (e.g., SWO for south western Ontario), all crop budget (amount) for growing in this region (e.g., 100 for $100M), and managerid (a farmer who oversees farmer affairs in this region). Revenue from grown crop can be measured in millions of dollars as well (e.g., 0.5 means half a million). A farmer may grow crop in multiple regions.

** Note that in creating these tables in the same Oracle database space in your account, you need to give them unique names not already given to any tables in your database. For example, you may need to call table Farmer something like Farmer_asn3 if you already have a table called Farmer from your assignement 2. Rename all the other tables in a similar way.

Fig 1.1: An Example Database State of the Farmer-growscrops-inRegions Database

Farmer

Ssn	Name	Age	Regionid
10	Jobe Bata	65	1
20	Monica Kap	80	1
30	Peter Good	22	2
40	Kate Lee	47	3
50	Ted Tam	50	4

Region

Regionid	Regname	Cropbudget	managerid
1	SWO	5.00	10
2	EO	4.00	10
3	SC	9.00	50
4	SEO	6.00	50

RContact

Regionid	Fullname	RPHONE
1	Southwest Ontario	NULL
2	Eastern Ontario	1(733)524-0100
3	Southcentral Ontario	1(866)663-3161
4	Southeastern Ontario	1(877)633-2000

Growscrops

Ssn	Regionid	crop	Revenue
10	1	corn	2.50
10	1	carrots	4.00
20	1	potatoes	9.00
30	2	potatoes	11.00
40	3	peas	15.30
50	4	potatoes	10.30
50	3	peas	14.00

**

Fig 1.2: Schema of Farmer database of Figure 1.1

Farmer

Ssn	Name	Age	Regionid

Region

Regionid	Regname	Cropbudget	managerid

RContact

Regionid	Fullname	RPHONE

Growscrops

Ssn	Regionid	crop	Revenue

i. List all the referential integrity constraints that should hold on the database schema (2.5 marks)

ii. Write appropriate SQL DDL statements to define the database with the integrity constraints and store in a text file called userid_farmerschema.sql. Attach this file or also show it in your script file of (v) using more file.sql command before or after running Sqlplus. Do the same for the files in (iii) and (iv). (2.5 marks)

iii. To insert the data in the database tables, also write appropriate SQL DML instructions in a text file called userid_farmerdata.sql. (2.5 marks)

iv. To remove any inserted data and destroy all created tables in the farmer database, write appropriate SQL DML and DDL statements in a text file called userid_farmerdroptable.sql to first delete all data in the tables and then drop the tables. (2.5 marks)

v. Using Oracle Sqlplus, implement this database design by creating all the tables with the integrity constraints using the SQL DDL you defined in (ii) above. You can create all these SQL DDL for creating the 4 tables by running your .sql file at the SQL prompt with the command:
@userid_farmerschema.sql. After creating your tables successfully, you load your data with the .sql file you created in (iii) above by running @userid_farmerdata.sql. If there are errors and you need to correct them, you might want to delete the tuples and drop the tables first using the .sql file you created in (iv) above as with @userid_farmerdroptable.sql before re-creating the schema and re-loading the data. Then, using a script file, show the contents of all 4 tables in the database by running: select * from each of the tables and saving on script file called username_assn3que1.txt. You can do this using the following sequence of Unix/Linux commands after you have created the database and inserted data. (5 marks)

(Note: remember to create the entity tables with primary keys before the relationship tables that reference them through foreign key attributes. When inserting data, do the same. If you need to delete the data and tables at any time, go in the reverse order (that is, delete the tuples that reference a primary key attribute tuple in another table, before deleting the parent primary keyed tuple))

```
>script   username_assn3que1.txt
>sqlplus <username>
>password
sqlplus>  select *  from Farmer;   //repeat this instruction for each table
sqlplus> exit                        //to exit sqlplus
exit                                 // to exit and create script file
```

**Now attach the saved unix log file of your session that is in username_assn3que1.txt with an inclusion in this unix script file of all the 4 .sql files in questions (ii), (iii) an (iv) or the attachment of those files as your solution showing where you executed (that is, actually built the databases) those files in Sqlplus.

Solution 1 (i) (mark: 2.5)

We will write a referential integrity constraint as R.A --> S (or R.(X) --> T) whenever attribute A (or the set of attributes X) of relation R form a foreign key that references the primary key of relation S (or T).

Growscrops.Ssn --> Farmer.Ssn
Growscrops.Regionid --> Region.Regionid
RContact.Regionid --> Region.Regionid:
Region.managerid --> Farmer.Ssn:

Solution 1 (ii): (mark: 2.5)

ONE POSSIBLE SET OF CREATE TABLE STATEMENTS TO DEFINE THE DATABASE IS GIVEN BELOW GIVEN IN THE FILE USERID_FARMERSCHEMA.SQL AS:

```
--
CREATE TABLE Farmer (
   Ssn INT PRIMARY KEY,
   Name VARCHAR(255),
   Age INT,
   Regionid INT,
   FOREIGN KEY (Regionid) REFERENCES Region(Regionid)
);

CREATE TABLE Region (
   Regionid INT PRIMARY KEY,
   Regname VARCHAR(255),
   Cropbudget DECIMAL(5, 2),
   managerid INT,
   FOREIGN KEY (managerid) REFERENCES Farmer(Ssn)
);

CREATE TABLE RContact (
   Regionid INT PRIMARY KEY,
   Fullname VARCHAR(255),
   RPHONE VARCHAR(20),
   FOREIGN KEY (Regionid) REFERENCES Region(Regionid)
);

CREATE TABLE Growscrops (
   Ssn INT,
   Regionid INT,
   crop VARCHAR(255),
   Revenue DECIMAL(5, 2),
   PRIMARY KEY (Ssn, Regionid, crop),
   FOREIGN KEY (Ssn) REFERENCES Farmer(Ssn),
   FOREIGN KEY (Regionid) REFERENCES Region(Regionid)
);
```

Solution 1 (iii): (mark: 2.5)

One possible set of INSERT INTO TABLE statements to define the database is given below given in the file userid_farmerdata.sql as:

Scroll down a little

```
INSERT INTO Farmer VALUES (10, 'Jobe Bata', 65, 1);
INSERT INTO Farmer VALUES (20, 'Monica Kap', 80, 1);
INSERT INTO Farmer VALUES (30, 'Peter Good', 22, 2);
```

```
INSERT INTO Farmer VALUES (40, 'Kate Lee', 47, 3);
INSERT INTO Farmer VALUES (50, 'Ted Tam', 50, 4);

INSERT INTO Region VALUES (1, 'SWO', 5.00, 10);
INSERT INTO Region VALUES (2, 'EO', 4.00, 10);
INSERT INTO Region VALUES (3, 'SC', 9.00, 50);
INSERT INTO Region VALUES (4, 'SEO', 6.00, 50);

INSERT INTO RContact VALUES (1, 'Southwest Ontario', NULL);
INSERT INTO RContact VALUES (2, 'Eastern Ontario', '1(733)524-0100');
INSERT INTO RContact VALUES (3, 'Southcentral Ontario', '1(866)663 3161');
INSERT INTO RContact VALUES (4, 'Southeastern Ontario', '1(877)633-2000');

INSERT INTO Growscrops VALUES (10, 1, 'corn', 2.50);
INSERT INTO Growscrops VALUES (10, 1, 'carrots', 4.00);
INSERT INTO Growscrops VALUES (20, 1, 'potatoes', 9.00);
INSERT INTO Growscrops VALUES (30, 2, 'potatoes', 11.00);
INSERT INTO Growscrops VALUES (40, 3, 'peas', 15.30);
INSERT INTO Growscrops VALUES (50, 4, 'potatoes', 10.30);
INSERT INTO Growscrops VALUES (50, 3, 'peas', 14.00);
```

Solution 1 (iv): (mark: 2.5)

One possible set of DELETE FROM TABLE statements and DROP TABLE statements to delete data from the database and drop the tables is given below given in the file userid_farmerdroptable.sql as:

```
DELETE FROM Growscrops;
DELETE FROM RContact;
DELETE FROM Farmer;
DELETE FROM Region;

DROP TABLE Growscrops;
DROP TABLE RContact;
DROP TABLE Farmer;
DROP TABLE Region;
```

1 (v). (5 marks) for the script file showing correct interaction with Oracle Sqlplus creating and loading data in these 5 tables.

2. Specify the following 5 queries in SQL on the Farmercrop record database schema of Figure 1.1.
(Total for que 2 is 10 marks)

i. List all your 5 queries in the table below first in SQL. (5 marks)
ii. Implement the answering of your 5 queries in 2(i) using Sqlplus and the same database you created in question 1, providing your execution and answers to these questions in a script file called username_assn3que2.txt. (5 marks)

(a) Retrieve the full names and phone numbers of all regions where potatoes are grown.
(b) Retrieve the Regname, Fullname, Rphone of all regions with total revenue of more than $10M.
(c) For each region, retrieve its regionid, number of farmers growning in that region, , and number of crops it grows and total revenue in that region.
(d) Retrieve for each farmer, the farmer's name, and the farmer's crop growing region names.
(e) Retrieve the crop name, regionid and regname of all crops that are not grown in more than one region.

Solution 2 (i): Queries(5 marks) and 2(ii) Results (5 marks)

```
- -(a) Retrieve the full names and phone numbers of all regions where potatoes are grown.

SQL query is:

SELECT R.Fullname, R.RPHONE
FROM RContact R
JOIN Growscrops G ON R.Regionid = G.Regionid
WHERE G.crop = 'potatoes';

/* Query result is given below as retrieved from Oracle Sqlplus

Full Name                      RPHONE
------------------ ---------
Southwest Ontario      NULL
Estern Ontario         1(733)524-0100
Southeastern Ontaro    1(877)633-2000
*/

-- (b) Retrieve the Regname, Fullname, Rphone of all regions with total revenue of more than $10M.

SQL query is:

SELECT R.Regname, R.Fullname, R.RPHONE
FROM RContact R
JOIN Growscrops G ON R.Regionid = G.Regionid
GROUP BY R.Regname, R.Fullname, R.RPHONE
HAVING SUM(G.Revenue) > 10.0;


/* Query result is:


Regname      Fullname                    RPHONE
-------  --------------------      --------------
EO           Eastern Ontario         1(733)524-0100
SC           Southcentral Ontario    1(866)663-3161
SEO          Southeastern Ontario    1(877)633-2000
*/
```

```
-- (c) For each region, retrieve its regionid, number of farmers growning in that region, , and number of crops it grows and total revenue in that region.
SQL query is:
SELECT R.Regionid, COUNT(DISTINCT F.Ssn) AS NumFarmers, COUNT(DISTINCT G.crop) AS NumCrops, SUM(G.Revenue) AS TotalRevenue
FROM Region R
LEFT JOIN Farmer F ON R.Regionid = F.Regionid
LEFT JOIN Growscrops G ON R.Regionid = G.Regionid
GROUP BY R.Regionid;
/* Query result is:
Regionid        NumFarmers    NumCrops       TotalRevenue
--------  ----------          --------  ------------
1               2               3               15.50
2               1               1               11.00
3               2               2               29.30
4               1               1               10.30
*/

-- (d) Retrieve for each farmer, the farmer's name, and the farmer's crop growing region names.
SQL query is:
SELECT F.Name AS FarmerName, R.Regname AS CropGrowingRegion
FROM Farmer F
JOIN Region R ON F.Regionid = R.Regionid;
/*  Query result is:
FarmerName          CropGrowingRegion
Jobe Bata           SWO (Southwest ON)
Monica Kap          SWO (Southwest ON)
Peter Good          EO (Eastern ON)
Kate Lee            SC (Southcentral ON)
Ted Tam             SEO (Southeastern ON)
*/
-- (e) Retrieve the crop name, regionid and regname of all crops that are not grown in more than one region.
SQL query is:
SELECT G.crop, G.Regionid, R.Regname
FROM Growscrops G
JOIN Region R ON G.Regionid = R.Regionid
WHERE G.crop NOT IN (
   SELECT crop
   FROM Growscrops
   GROUP BY crop
   HAVING COUNT(DISTINCT Regionid) > 1
);

/*  Query result is:
Crop            Regionid        Regname
Corn            1               SWO (Southwest ON)
Carrots 1               SWO (Southwest ON)
Peas            3               SC (Southcentral ON)
Potatoes        4               SEO (Southeastern ON)
*/
```

2 (ii). (5 marks) distributed as: 2.5 marks for the script file showing correct interaction with Oracle Sqlplus posing these queries; and 2.5 marks for correctly posing the queries and retrieving correct results.

3. Write four SQL update statements to do the following updates on the database schema shown in Figure 1.2. Show the affected tables after update through script file in sqlplus and in a script file created as before and named username_assn3que3.txt. (5 marks)

(Total for que 3 is 5 marks)

(a) Insert a new crop grown by an existing farmer in a region with unknown revenue as <10, 1, 'brocolli', 'NULL'> in the database.

(b) Change the crop budget of region with regname 'SWO' to 20.

(c) Insert a new farmer <60, 'Mary Tama', 21, 2>

(d) Delete all crops grown records for farmer whose SSN is 20.

Solution 3 (i): (5 marks)

```
-- 3 (a) Insert a new  crop grown by an existing farmer in a region with unknown revenue as
 <10, 1, 'brocolli', 'NULL'> in the database.
INSERT INTO Growscrops (Ssn, Regionid, Crop, Revenue)
VALUES (10, 1, 'broccoli', NULL);
-- 3 (b) Change the crop budget of region with regname 'SWO'  to 20.
UPDATE Region
SET Cropbudget = 20
WHERE Regname = 'SWO';
-- 3. (c) Insert a new farmer <60, 'Mary Tama',  21, 2>
INSERT INTO Farmer (Ssn, Name, Age, Regionid)
VALUES (60, 'Mary Tama', 21, 2);
-- 3 (d) Delete all crops grown records for farmer whose SSN is 20..
DELETE FROM Growscrops
WHERE Ssn = 20;
```

CHAPTER 7: More SQL: Complex Queries, Triggers, Views, and Schema Modification

4. (i) Write the following 2 queries in SQL on the database schema of Figure 1.2 using EXISTS or NOT EXISTS as appropriate.

(2.5 marks)

(ii) Implement the answering of your 2 queries in 4(i) using Sqlplus and the same database you created in question 1 and modified in earlier question with updates, deletes and inserts, providing your execution and answers to this question in a script file called username_assn3que4.

(2.5 marks)

(Total for que 4 is 5 marks)

(a) Retrieve the crop name and region id of all crops that have been grown in at least 2 regions.
(b) Retrieve the crop name and region id of all crops that have not been grown in at least 2 regions.

Solution 4 (i): (2.5 marks)
GA: check student queries as my answers are with the original database. A slightly different answer with correct queries from students should be accepted to accommodate use of modified database after implementing question 3 above.

```
-- 4 (a) Retrieve the crop name and region id of all crops that have been grown in at least 2 regions.

SQL query is:
SELECT crop, Regionid
FROM Growscrops G1
WHERE EXISTS (
   SELECT 1
   FROM Growscrops G2
   WHERE G1.crop = G2.crop
   AND G1.Regionid <> G2.Regionid
);
/* Query result is:
Crops            Regionid
--------  -
potatoes         1
potatoes         2
peas             3
*/
-- 4 (b)Retrieve the crop name and region id of all crops that have not been grown in at least 2
regions.

SQL query is:
SELECT DISTINCT G1.crop, G1.Regionid
FROM Growscrops G1
WHERE NOT EXISTS (
   SELECT DISTINCT G2.crop, G2.Regionid
   FROM Growscrops G2
   WHERE G1.crop = G2.crop
   GROUP BY G2.crop
   HAVING COUNT(DISTINCT G2.Regionid) >= 2
);
/*  Query result is:
crop     Regionid
--------- --------
corn     1
carrots   1
potatoes  2
potatoes  4
peas     3
*/
```

4 (ii). (2.5 marks) distributed as: 0.5 marks for the script file showing correct interaction with Oracle Sqlplus posing these queries; and 2 marks for the correctly posing the queries and retrieving correct results.

5. In SQL, specify the following 3 queries on the COMPANY database of Figures 5.5 and 5.6 using

the concept of nested queries and the concepts described in chapter 7. Note that you can just provide the SQL of these queries without building the database or implementing through Sqlplus. (Total for que 5 is 5 marks)

a. Retrieve the names of all projects that have the maximum number of hours worked on them per week.
b. Retrieve the hours of all projects whose number of hours worked on per week are greater than the average number of hours worked on all projects.
c. Retrieve the names of projects that are worked on at least 20 hours more than the project with the least number of hours worked on it per week.

Figure 5.5 Schema diagram for the COMPANY relational database

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno

DEPARTMENT

Dname	Dnumber	Mgr_ssn	Mgr_start_date

DEPT_LOCATIONS

Dnumber	Dlocation

PROJECT

Pname	Pnumber	Plocation	Dnum

WORKS_ON

Essn	Pno	Hours

DEPENDENT

Essn	Dependent_name	Sex	Bdate	Relationship

Solution 5: (5 marks)

```
a)
SELECT Pname
FROM Project
WHERE Pnumber IN (
   SELECT Pno
   FROM Works_On
   GROUP BY Pno
   HAVING SUM(hours) = (
      SELECT MAX(total_hours)
      FROM (
         SELECT Pno, SUM(hours) AS total_hours
         FROM Works_On
         GROUP BY Pno
      ) AS ProjectHours
   )
);
```

```
b)
SELECT Pno, hours
FROM Works_On
WHERE hours > (
   SELECT AVG(total_hours)
   FROM (
      SELECT Pno, SUM(hours) AS total_hours
      FROM Works_On
      GROUP BY Pno
   ) AS ProjectHours
);

c)
SELECT Pname
FROM Project
WHERE Pnumber IN (
   SELECT Pno
   FROM Works_On
   GROUP BY Pno
   HAVING MIN(hours) + 20 <= ALL (
      SELECT hours
      FROM Works_On
      WHERE Pno = Project.Pnumber
   )
);
```

CHAPTER 8: THE RELATIONAL ALGEBRA AND RELATIONAL CALCULUS

6. Specify the following 5 queries on the COMPANY relational database schema shown in Figure 5.5, using the relational operators discussed in chapter 8. Also show the result of each query as it would apply to the database state of Figure 5.6. (Total for que 6 is 10 marks)

Some symbols for solving queries you may copy and reuse are: π, σ, $\Im$, ρ, $\bowtie$

Figure 5.6 One possible database state for the COMPANY relational database schema.

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

DEPARTMENT

Dname	Dnumber	Mgr_ssn	Mgr_start_date
Research	5	333445555	1988-05-22
Administration	4	987654321	1995-01-01
Headquarters	1	888665555	1981-06-19

DEPT_LOCATIONS

Dnumber	Dlocation
1	Houston
4	Stafford
5	Bellaire
5	Sugarland
5	Houston

WORKS_ON

Essn	Pno	Hours
123456789	1	32.5
123456789	2	7.5
666884444	3	40.0
453453453	1	20.0
453453453	2	20.0
333445555	2	10.0
333445555	3	10.0
333445555	10	10.0
333445555	20	10.0
999887777	30	30.0
999887777	10	10.0
987987987	10	35.0
987987987	30	5.0
987654321	30	20.0
987654321	20	15.0
888665555	20	NULL

PROJECT

Pname	Pnumber	Plocation	Dnum
ProductX	1	Bellaire	5
ProductY	2	Sugarland	5
ProductZ	3	Houston	5
Computerization	10	Stafford	4
Reorganization	20	Houston	1
Newbenefits	30	Stafford	4

DEPENDENT

Essn	Dependent_name	Sex	Bdate	Relationship
333445555	Alice	F	1986-04-05	Daughter
333445555	Theodore	M	1983-10-25	Son
333445555	Joy	F	1958-05-03	Spouse
987654321	Abner	M	1942-02-28	Spouse
123456789	Michael	M	1988-01-04	Son
123456789	Alice	F	1988-12-30	Daughter
123456789	Elizabeth	F	1967-05-05	Spouse

(i) List the names of employees who have a dependent with the same first name as themselves.

(ii) Find the names of employees that are directly supervised by 'James Borg'.

(iii) For each project, list the project name and the total hours per week (by all employees) spent on that project.

(iv) Retrieve the names of employees who work on every project.

(v) Retrieve the maximum salary of all female employees.

Solution 6: (10 marks)

In the relational algebra, as in other languages, it is possible to specify the same query in multiple ways. We give one possible solution for each query. Wemay also use the symbol s for SELECT, P for PROJECT, J for EQUIJOIN, * for NATURAL JOIN, and f for FUNCTION.

(i)

```
SELECT DISTINCT E.Fname
FROM EMPLOYEE E, DEPENDENT D
WHERE E.Ssn = D.Essn AND E.Fname = D.Dependent_name;
```

Result ():

```
Fname
------
Alicia
Joyce
```

(ii)

```
SELECT E.Fname, E.Lname
FROM EMPLOYEE E
WHERE E.Super_ssn = (SELECT Ssn FROM EMPLOYEE WHERE Fname = 'James' AND Lname = 'Borg');
```

Result:

```
Fname   Lname
------- --------
John    Smith
Joyce   English
Ahmad   Jabbar
```

(iii)

```
SELECT P.Pname, SUM(W.Hours) AS TotalHoursPerWeek
FROM PROJECT P, WORKS_ON W
WHERE P.Pnumber = W.Pno
GROUP BY P.Pname;
```

Result:

```
Pname          TotalHoursPerWeek
-------------- -------------------
ProductX        40.0
ProductY        10.0
ProductZ        10.0
Computerization 50.0
Reorganization  15.0
Newbenefits     55.0
```

(iv)

```
SELECT E.Fname, E.Lname
FROM EMPLOYEE E
WHERE NOT EXISTS (
   SELECT P.Pnumber
   FROM PROJECT P
   WHERE NOT EXISTS (
      SELECT W.Pno
      FROM WORKS_ON W
      WHERE W.Essn = E.Ssn AND W.Pno = P.Pnumber
   )
);
```

Result ():

```
Fname    Lname
```

```
-------- --------
Franklin Wong
Alicia   Zelaya
```

(v)

```
SELECT MAX(Salary) AS MaxSalary
FROM EMPLOYEE
WHERE Sex = 'F';
```

Result:

```
MaxSalary
----------
43000
```