

School: Computer Science
Institution: University of Windsor
Term: Fall 2021
Course: Comp-3150 (03-60-315-1) : Database Management Systems
Instructor: Dr. C. I. Ezeife
Assignment #3 Solution: Total: 50 marks
Handed Out: Thurs. Oct. 28, 2021; **Due** Thurs. Nov. 18, 2021

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://blackboard.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.

Student Signature

Student Name (please print)

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 vaccination database shown in Figure 1.1, with schema shown in Figure 1.2. Note that here, each person gets their vaccine dose 1 and dose 2 in a different centre.

(Total for que 1 is 15 marks)

Fig 1.1: An Example Database State of the Vaccination Database

Person

<u>Ssn</u>	Name	Age	jobtype
10	Jobe Bata	65	nurse
20	Monica Kap	80	retiree
30	Peter Good	22	retailer
40	Kate Lee	47	teacher
50	Ted Tam	50	doctor

Vaccinates

<u>Ssn</u>	<u>Cntid</u>
10	1
10	4
20	1
30	2
40	3
50	4
50	3

Administers

<u>Ssn</u>	<u>Cntid</u>	<u>Vname</u>	vacdate	dose	vactime
10	1	Pfizer	02-apr-21	1	13.30
10	4	Pfizer	12-jun-21	2	12.30
20	1	Astrazeneca	04-mar-21	1	9.00
30	2	Moderna	12-may-21	1	11.00
40	3	Pfizer	20-apr-21	1	15.30
50	4	Astrazeneca	20-apr-21	1	10.30
50	3	Pfizer	23-jun-21	2	14.00

Centre

<u>Cntid</u>	Cntname	city	budget	managerid
1	DownTn	Windsor	600000	10
2	St Clair	Windsor	400000	10
3	WFCU	Windsor	900000	50
4	Other	Windsor	600000	50

Vaccine

<u>VNAME</u>	FORMULA	MADEBY
Astrazeneca	Adenoviruses	Astrazeneca
Johnson and Johnson	viral vector	Johnson and Johnson
Moderna	mRNA	Moderna
Pfizer	mRNA	Pfizer

PCompany

<u>CNAME</u>	CPHONE
Astrazeneca	NULL
Johnson and Johnson	1(732) 524-0400
Moderna	1(866)663-3762
Pfizer	1 (877) 633-2001
US FDA	1 (888) 463-6332

Sells

<u>VNAME</u>	<u>CNAME</u>	PRICE
Astrazeneca	Astrazeneca	5.00
Johnson and Johnson	Johnson and Johnson	10.00
Moderna	Moderna	32.00
Pfizer	Pfizer	20.00

**

Fig 1.2: Schema of Vaccination database of Figure 1.1

Person

<u>Ssn</u>	Name	Age	jobtype
------------	------	-----	---------

Vaccinates

<u>Ssn</u>	<u>Cntid</u>
------------	--------------

Administers

<u>Ssn</u>	<u>Cntid</u>	<u>Vname</u>	vacdate	dose	vactime
------------	--------------	--------------	---------	------	---------

Centre

<u>Cntid</u>	Cntname	city	budget	managerid
--------------	---------	------	--------	-----------

Vaccine

<u>VNAME</u>	FORMULA	MADEBY
--------------	---------	--------

PCompany

<u>CNAME</u>	CPHONE
--------------	--------

Sells

<u>VNAME</u>	<u>CNAME</u>	PRICE
--------------	--------------	-------

- 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_vaccinateschema.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_vaccinatedata.sql`. (2.5 marks)
- iv. To remove any inserted data and destroy all created tables in the vaccinate database, write appropriate SQL DML and DDL statements in a text file called `userid_vaccinatedroptable.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 7 tables by running your .sql file at the SQL prompt with the command:
`@userid_vaccinateschema.sql`. After creating your tables successfully, you load your data with the .sql file you created in (iii) above by running `@userid_vaccinatedata.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_vaccinatedroptable.sql` before re-creating the schema and re-loading the data. Then, using a script file, show the contents of all 7 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 Person; //repeat this instruction for each table
sqlplus> exit //to exit sqlplus
exit // to exit and create script file
```

****Now attach the saved log of your session that is in `username_assn3que1.txt` with an inclusion in this script file of all the 7 .sql files in questions (ii), (iii) and (iv) or the attachment of those files as your solution.**

Solution 1 (i) (mark: 2.5)

We will write a referential integrity constraint as $R.A \rightarrow S$ (or $R.(X) \rightarrow T$) 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).

Vaccinates.SSN \rightarrow PERSON
 Vaccinates.Cntid \rightarrow CENTRE
 Administers.SSN \rightarrow PERSON
 Administers.Cntid \rightarrow CENTRE
 Administers.Vname \rightarrow VACCINE
 Sells.Vname \rightarrow VACCINE
 Sells.Cname \rightarrow PCompany
 VACCINE.Madeby \rightarrow PCompany (GA, if this is missing, do not penalize)

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_VACCINESCHEMA.SQL IS:

```
CREATE TABLE PCOMPANY (
CNAME   VARCHAR2(20) NOT NULL,
CPHONE  VARCHAR2(15),
PRIMARY KEY(CNAME));

CREATE TABLE PERSON(
SSN      NUMBER(9) NOT NULL,
NAME     VARCHAR2(20),
AGE      NUMBER(2),
JOBTYP  VARCHAR2(20),
PRIMARY KEY (SSN));

CREATE TABLE CENTRE (
CNTID    NUMBER(3) NOT NULL,
CNTNAME  VARCHAR2(20),
CITY     VARCHAR2(20),
BUDGET   NUMBER(8),
MANAGERID NUMBER(2),
PRIMARY  KEY(CNTID));

CREATE TABLE VACCINE (
VNAME    VARCHAR2(20) NOT NULL,
FORMULA  VARCHAR2(15),
MADEBY   VARCHAR2(20),
PRIMARY KEY(VNAME),
FOREIGN KEY(MADEBY) REFERENCES PCOMPANY(CNAME));

CREATE TABLE VACCINATES (
SSN      NUMBER(9) NOT NULL,
CNTID    NUMBER(3) NOT NULL,
PRIMARY KEY(SSN, CNTID),
FOREIGN KEY(SSN) REFERENCES PERSON(SSN),
FOREIGN KEY(CNTID) REFERENCES CENTRE(CNTID));

CREATE TABLE SELLS (
VNAME    VARCHAR2(20) NOT NULL,
CNAME    VARCHAR2(20) NOT NULL,
PRICE    NUMBER(6,2),
PRIMARY KEY(VNAME, CNAME),
FOREIGN KEY(VNAME) REFERENCES VACCINE(VNAME),
FOREIGN KEY(CNAME) REFERENCES PCOMPANY(CNAME));
```

```

CREATE TABLE ADMINISTERS (
SSN      NUMBER(9) NOT NULL,
CNTID    NUMBER(3) NOT NULL,
VNAME    VARCHAR2(20) NOT NULL,
VACDATE  DATE,
DOSE     SMALLINT,
VACTIME  NUMBER(4,2),
PRIMARY KEY(SSN, CNTID, VNAME),
FOREIGN KEY(SSN) REFERENCES PERSON(SSN),
FOREIGN KEY(CNTID) REFERENCES CENTRE(CNTID),
FOREIGN KEY(VNAME) REFERENCES VACCINE(VNAME));

COMMIT;

```

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_vaccinedata.sql is:

```

--First insert tuples into PCompany as:
INSERT INTO PCOMPANY VALUES ('Astrazeneca', NULL);
COMMIT;
INSERT INTO PCOMPANY VALUES ('Johnson and Johnson', '1(732) 524-0400');
COMMIT;
INSERT INTO PCOMPANY VALUES ('Moderna', '1(866) 663-3762');
COMMIT;
INSERT INTO PCOMPANY VALUES ('Pfizer', '1(877) 633-2001');
COMMIT;
INSERT INTO PCOMPANY VALUES ('US FDA', '1(888) 463-6332');
COMMIT;

INSERT INTO VACCINE VALUES ('Astrazeneca', 'Adenoviruses', 'Astrazeneca');
COMMIT;
INSERT INTO VACCINE VALUES ('Johnson and Johnson', 'viral vector', 'Johnson and Johnson');
COMMIT;
INSERT INTO VACCINE VALUES ('Moderna', 'mRNA', 'Moderna');
COMMIT;
INSERT INTO VACCINE VALUES ('Pfizer', 'mRNA', 'Pfizer');
COMMIT;

INSERT INTO PERSON VALUES (10, 'Jobe Bata', 65, 'nurse');
COMMIT;
INSERT INTO PERSON VALUES (20, 'Monica Kap', 80, 'retiree');
COMMIT;
INSERT INTO PERSON VALUES (30, 'Peter Good', 22, 'retailer');
COMMIT;

```

```

INSERT INTO PERSON VALUES (40, 'Kate Lee', 47, 'teacher');
COMMIT;
INSERT INTO PERSON VALUES (50, 'Ted Tam', 50, 'doctor');
COMMIT;

INSERT INTO CENTRE VALUES (1, 'DownTn', 'Windsor', 600000, 10);
COMMIT;
INSERT INTO CENTRE VALUES (2, 'St Clair', 'Windsor', 400000, 10);
COMMIT;
INSERT INTO CENTRE VALUES (3, 'WFCU', 'Windsor', 900000, 50);
COMMIT;
INSERT INTO CENTRE VALUES (4, 'Other', 'Windsor', 600000, 50);
COMMIT;

INSERT INTO SELLS VALUES ('Astrazeneca', 'Astrazeneca', 5.00);
COMMIT;
INSERT INTO SELLS VALUES ('Johnson and Johnson', 'Johnson and
Johnson', 10.00);
COMMIT;
INSERT INTO SELLS VALUES ('Moderna', 'Moderna', 32.00);
COMMIT;
INSERT INTO SELLS VALUES ('Pfizer', 'Pfizer', 20.00);
COMMIT;

INSERT INTO VACCINATES VALUES (10, 1);
COMMIT;
INSERT INTO VACCINATES VALUES (10, 4);
COMMIT;
INSERT INTO VACCINATES VALUES (20, 1);
COMMIT;
INSERT INTO VACCINATES VALUES (30, 2);
COMMIT;
INSERT INTO VACCINATES VALUES (40, 3);
COMMIT;
INSERT INTO VACCINATES VALUES (50 ,4);
COMMIT;
INSERT INTO VACCINATES VALUES (50, 3);
COMMIT;

INSERT INTO ADMINISTERS VALUES (10, 1, 'Pfizer', '02-apr-21', 1,
13.30);
COMMIT;
INSERT INTO ADMINISTERS VALUES (10, 4, 'Pfizer', '12-jun-21', 2,
12.30);
COMMIT;
INSERT INTO ADMINISTERS VALUES (20, 1, 'Astrazeneca', '04-mar-21', 1,
9.00);
COMMIT;
INSERT INTO ADMINISTERS VALUES (30, 2, 'Moderna', '12-may-21', 1,
11.00);
COMMIT;

```



```
INSERT INTO ADMINISTERS VALUES (40, 3, 'Pfizer', '20-apr-21', 1,  
15.30);  
COMMIT;  
INSERT INTO ADMINISTERS VALUES (50, 4, 'Astrazeneca', '20-apr-21', 1,  
10.30);  
COMMIT;  
INSERT INTO ADMINISTERS VALUES (50, 3, 'Pfizer', '23-jun-21', 2,  
14.00);  
COMMIT;
```

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_vaccinedroptable.sql` is:

```
DELETE FROM ADMINISTERS;  
DELETE FROM SELLS;  
DELETE FROM VACCINATES;  
DELETE FROM VACCINE;  
DELETE FROM CENTRE;  
DELETE FROM PERSON;  
DELETE FROM PCOMPANY;  
COMMIT;
```

```
DROP TABLE ADMINISTERS;  
DROP TABLE SELLS;  
DROP TABLE VACCINATES;  
DROP TABLE VACCINE;  
DROP TABLE CENTRE;  
DROP TABLE PERSON;  
DROP TABLE PCOMPANY;  
COMMIT;
```

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 Vaccinate 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 names and phone numbers of all pharmaceutical companies that made Pfizer vaccine.
- (b) Retrieve the Vname, formula and pharmaceutical company of all vaccines that cost more than \$5.
- (c) For each pharmaceutical company, retrieve its name, its phone number, and number of vaccines it sells.
- (d) Retrieve the vaccine name, the selling pharmaceutical company name and phone number, and price of each vaccine.
- (e) Retrieve the vaccine name, formula and the selling pharmaceutical company name of all vacciness that do not have a dose 2 taken by a person.

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

```
SELECT PC.CNAME, PC.CPHONE  
FROM PCOMPANY PC, SELLS S, VACCINE V  
WHERE V.VNAME = 'Pfizer'
```

```
AND V.VNAME = S.VNAME
AND PC.CNAME = S.CNAME;
```

```
or
```

```
SELECT PC.CNAME, PC.CPHONE
FROM PCOMPANY PC, VACCINE V
WHERE PC.CNAME = V.MADEBY
AND V.VNAME = 'Pfizer';
```

```
/*
CNAME                CPHONE
-----
Pfizer                1-877-633-2001
*/
```

```
-- (b) Retrieve the Vname, formula and pharmaceutical company of all
vaccines that cost more than $5.
```

```
SELECT V.VNAME, V.FORMULA, S.CNAME
FROM SELLS S, VACCINE V
WHERE S.PRICE > 5
AND S.VNAME = V.VNAME;
```

```
/*
VNAME                FORMULA                CNAME
-----
Johnson and Johnson  viral vector    Johnson and Johnson
Moderna              mRNA           Moderna
Pfizer               mRNA           Pfizer
*/
```

```
-- (c ) For each pharmaceutical company, retrieve its name, its phone
number, and number of vaccines it sells
```

```
SELECT PC.CNAME, PC.CPHONE, count(S.CNAME)
FROM PCOMPANY PC, SELLS S
WHERE PC.CNAME = S.CNAME
GROUP BY PC.CNAME, PC.CPHONE;
```

```
/*
CNAME                CPHONE                COUNT(S.CNAME)
-----
Astrazeneca                                1
Johnson and Johnson  1(732) 524-0400          1
Moderna              1(866) 663-3762          1
Pfizer               1(877) 633-2001          1
*/
```

```
-- (d) Retrieve the vaccine name, the selling pharmacy name and phone number, and price of each vaccine.
```

```
SELECT S.VNAME, S.CNAME, PC.CPHONE, S.PRICE
FROM PCOMPANY PC, SELLS S, VACCINE V
WHERE S.CNAME = PC.CNAME
AND V.VNAME = S.VNAME;
```

```
/*
VNAME                CNAME                CPHONE                PRICE
-----
Astrazeneca          Astrazeneca                5
Johnson and Johnson Johnson and Johnson 1(732) 524-0400      10
Moderna              Moderna              1(866) 663-3762      32
Pfizer              Pfizer              1(877) 633-2001      20
*/
```

```
-- (e) Retrieve the vaccine name, formula, and the selling pharmacy name of all vacciness that do not have a dose 2 taken by a person.
```

```
SELECT V.VNAME, V.FORMULA, S.CNAME
FROM VACCINE V, SELLS S
WHERE V.VNAME = S.VNAME
AND S.VNAME IN (
    SELECT A.VNAME
    FROM ADMINISTERS A
    GROUP BY A.VNAME
    HAVING max(A.DOSE) !=2
);
```

```
/*
VNAME                FORMULA                CNAME
-----
Astrazeneca          Adenoviruses          Astrazeneca
Moderna              mRNA                  Moderna
*/
```

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 centre <5, 'Devonshire', 'Windsor', '1000000', 10> in the database.

(b) Change the budget of centre with Cntname 'DownTn' to 700000

(c) Insert a new person <60, 'Mary Tama', 21, 'student'>

(d) Delete all Sells records for pharmaceutical company whose name is 'US FDA'.

Solution 3 (i): (5 marks)

```
-- 3 (a) Insert a new centre <5, 'Devonshire', 'Windsor', '1000000', 10> in the
database.

INSERT INTO CENTRE values (5, 'Devonshire', 'Windsor', '1000000', 10);

-- 1 row created.

-- 3 (b) Change the budget of centre with Cntname 'DownTn' to 700000.

UPDATE CENTRE SET BUDGET = 700000 WHERE CNTNAME = 'DownTn';

-- 1 row updated.

-- 3. (c) Insert a new person <60, 'Mary Tama', 21, 'student'>
INSERT INTO PERSON VALUES (60, 'Mary Tama', 21, 'student');

-- 1 row created.

-- 3 (d) Delete all Sells records for pharmaceutical company whose
name is 'US FDA'.

DELETE FROM SELLS WHERE CNAME = 'US FDA';

-- 0 rows deleted.
```

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 vaccine name and manufacturing company name of all vaccines that have been taken in at least 2 doses.

(b) Retrieve the vaccine name and manufacturing company name of all vaccines that have not been taken in at least 2 doses.

Solution 4 (i): (2.5 marks)

-- 4 (a) Retrieve the vaccine name and manufacturing company name of all vaccines that have been taken in at least 2 doses.

```
SELECT V.VNAME, V.MADEBY
FROM VACCINE V
WHERE EXISTS(
    SELECT A.VNAME
    FROM ADMINISTERS A
    WHERE V.VNAME = A.VNAME
    HAVING MAX(DOSE) >=2
    GROUP BY A.VNAME
);
```

Or the query:

```
SELECT V.VNAME, V.MADEBY
FROM VACCINE V
WHERE EXISTS(
    SELECT A.VNAME
    FROM ADMINISTERS A
    WHERE V.VNAME = A.VNAME
    and Dose >= 2
);
```

```
/*
VNAME                MADEBY
-----
Pfizer                Pfizer
*/
```

-- 4 (b) Retrieve the vaccine name and manufacturing company name of all vaccines that have not been taken in at least 2 doses.

```
SELECT V.VNAME, V.MADEBY
FROM VACCINE V
WHERE NOT EXISTS(
    SELECT A.VNAME
    FROM ADMINISTERS A
    WHERE V.VNAME = A.VNAME
    HAVING MAX(DOSE) >=2
    GROUP BY A.VNAME
);
```

```
VNAME                MADEBY
-----
Astrazeneca          Astrazeneca
Johnson and Johnson Johnson and Johnson
Moderna              Moderna
```

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)

- Retrieve the names of all projects that have the maximum number of hours worked on them per week.
- 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.
- 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	<u>Ssn</u>	Bdate	Address	Sex	Salary	Super_ssn	Dno
-------	-------	-------	------------	-------	---------	-----	--------	-----------	-----

DEPARTMENT

Dname	<u>Dnumber</u>	Mgr_ssn	Mgr_start_date
-------	----------------	---------	----------------

DEPT_LOCATIONS

<u>Dnumber</u>	<u>Dlocation</u>
----------------	------------------

PROJECT

Pname	<u>Pnumber</u>	Plocation	Dnum
-------	----------------	-----------	------

WORKS_ON

<u>Essn</u>	<u>Pno</u>	Hours
-------------	------------	-------

DEPENDENT

<u>Essn</u>	<u>Dependent_name</u>	Sex	Bdate	Relationship
-------------	-----------------------	-----	-------	--------------

Solution 5: (5 marks)

- SELECT PNAME FROM PROJECT WHERE PNUMBER IN
(SELECT PNO FROM WORKS_ON WHERE HOURS IN
(SELECT MAX(HOURS) FROM WORKS_ON));
- SELECT HOURS FROM WORKS_ON WHERE HOURS >= ANY
(SELECT AVG(HOURS) FROM WORKS_ON);
- SELECT PNO, HOURS FROM WORKS_ON WHERE HOURS >= 20 +
(SELECT MIN(HOURS) FROM WORKS_ON);

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 , \Join , 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. We use the symbol **S** for SELECT, **P** for PROJECT, **J** for EQUIJOIN, ***** for NATURAL JOIN, and **f** for FUNCTION.

(i)

$$E \leftarrow (\text{EMPLOYEE} \bowtie_{\text{Ssn}=\text{Essn} \text{ and } \text{Fname}=\text{Dependent_name}} (\text{DEPENDENT}))$$

$$R \leftarrow \pi_{\text{Lname}, \text{Fname}} (E)$$

Result (empty):

LNAME FNAME

(ii) $\text{Borg_ssn} \leftarrow \pi_{\text{Ssn}} (\sigma_{\text{Fname}='James' \text{ and } \text{Lname}='Borg'} (\text{EMPLOYEE}))$

$\text{Borg_emps} \leftarrow (\text{EMPLOYEE} \bowtie_{\text{Super_ssn}=\text{Ssn}} (\text{Borg_ssn}))$

$\text{Result} \leftarrow \pi_{\text{Lname}, \text{Fname}} (\text{Borg_emps})$

Result:

Select e.fname, e.lname

From employee e, employee m

Where e.super_ssn = m.ssn and m.fname = 'James' and m.lname='Borg';

FNAME	LNAME
Franklin	Wong
Jennifer	Wallace

(iii)

$\text{PROJ_HOURS (Pno, Tot_Hrs)} \leftarrow \text{Pno} \bowtie_{\text{Sum Hours}} (\text{WORKS_ON})$
 $\text{RESULT} \leftarrow \pi_{\text{Pname, Tot_hrs}} (\text{PROJ_HOURS} \bowtie_{\text{Pno = Pnumber}} (\text{PROJECT}))$

Result:

PNAME	TOT_HRS
ProductX	52.5
ProductY	37.5
ProductZ	50.0
Computerization	55.0
Reorganization	25.0
Newbenefits	55.0

(iv)

$\text{PROJ_EMPS(PNO,SSN)} \leftarrow \pi_{\text{pno, Essn}} (\text{WORKS_ON})$
 $\text{ALL_PROJS(PNO)} \leftarrow \pi_{\text{PNUMBER}} (\text{PROJECT})$
 $\text{EMPS_ALL_PROJS} \leftarrow \text{PROJ_EMPS} \div \text{ALLPROJS} \quad /* \text{DIVISION operation} */$
 $\text{RESULT} \leftarrow \pi_{\text{LNAME, FNAME}} (\text{EMPLOYEE} \bowtie \text{EMP_ALL_PROJS}) \quad /* \text{natural join on ssn} */$

Result (empty):

LNAME	FNAME
-------	-------

(v) $\text{RESULT(MAX_F_SAL)} \leftarrow \mathcal{F}_{\text{maximum salary}} ((\sigma_{\text{SEX} = 'F'} \text{EMPLOYEE}))$

Or

(v) $\rho \text{ RESULT(MAX-F-SAL)} (\mathcal{F}_{\text{maximum salary}} (\sigma_{\text{SEX} = 'F'} \text{EMPLOYEE}))$

Result:

max_F_SAL
43000