**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 : Total: 50 marks

**Handed Out:Thurs. Oct. 28, 2021; Due: ~~Thurs~~. Tue. Nov. 23 ~~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.**

\_RAVI TRIVEDI\_ \_Ravi Trivedi\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Student Signature Student Name (please print)

\_105197609\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_23th novembre, 2021\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

|  |
| --- |
| Ssn 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

|  |
| --- |
| Ssn Cntid |
| 10 1  10 4  20 1  30 2  40 3  50 4  50 3 |

Administers

|  |
| --- |
| Ssn Cntid Vname 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

|  |
| --- |
| Cntid 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

|  |
| --- |
| Vname Formula Madeby |
| Astrazeneca Adenoviruses Astrazeneca  Johnson and Johnson viral vector Johnson and Johnson Moderna mRNA Moderna  Pfizer mRNA Pfizer |

PCompany

|  |
| --- |
| CNAME 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

|  |
| --- |
| VNAME CNAME 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

|  |
| --- |
| Ssn Name Age jobtype |

Vaccinates

|  |
| --- |
| Ssn Cntid |

Administers

|  |
| --- |
| Ssn Cntid Vname vacdate dose vactime |

Centre

|  |
| --- |
| Cntid Cntname city budget managerid |

Vaccine

|  |
| --- |
| Vname Formula Madeby |

PCompany

|  |
| --- |
| CNAME CPHONE |

Sells

|  |
| --- |
| VNAME CNAME 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)

1. 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)
2. 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)
3. 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) an (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 --> 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).  There are 9 referential integrity constraints based on the database given:   * 1. Centre.managerId 🡪 Person (Person.Ssn)   2. Vaccinates.Ssn 🡪 Person (Person.Ssn)   3. Vaccinates.Cntid 🡪 Centre (Center.Cntid)   4. Administers.Ssn 🡪 Person (Person.Ssn)   5. Administers.Cntid 🡪 Centre (Centre.Cntid)   6. Administers.Vname 🡪 Vaccine (Vaccine.VNAME)   7. Sells.Vname 🡪 Vaccine (Vaccine.VNAME)   8. Sells.Cname 🡪 Pcompany (PCompany.CName)   9. Vaccine.madeBy 🡪 Pcompany (PCompany.CName) |

**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:  SQL> CREATE TABLE PERSON(  SSN NUMBER(3) NOT NULL,  NAME VARCHAR2(15),  AGE NUMBER(3),  JOBTYPE VARCHAR2(15),  PRIMARY KEY(SSN));  SQL> CREATE TABLE CENTRE  (  CNTID NUMBER(3) NOT NULL,  CNTNAME VARCHAR2(15) NOT NULL,  CITY VARCHAR2(15),  BUDGET NUMBER(10,2),  MANAGERID NUMBER(3),  PRIMARY KEY (CNTID),  FOREIGN KEY(MANAGERID) REFERENCES PERSON(SSN));  SQL> CREATE TABLE PCOMPANY  (  CNAME VARCHAR2(15) NOT NULL,  CPHONE VARCHAR2(10),  PRIMARY KEY(CNAME));  SQL> CREATE TABLE VACCINE  (  VNAME VARCHAR2(15) NOT NULL,  FORMULA VARCHAR2(15) NOT NULL,  MADEBY VARCHAR2(15) NOT NULL,  PRIMARY KEY (VNAME),  FOREIGN KEY(MADEBY) REFERENCES PCOMPANY(CNAME));  SQL> CREATE TABLE VACCINATES  (  SSN NUMBER(3) NOT NULL,  CNTID NUMBER(3) NOT NULL,  PRIMARY KEY(SSN, CNTID),  FOREIGN KEY(SSN) REFERENCES PERSON(SSN),  FOREIGN KEY(CNTID) REFERENCES CENTRE(CNTID));  SQL> CREATE TABLE ADMINISTERS  (  SSN NUMBER(3) NOT NULL,  CNTID NUMBER(3) NOT NULL,  VNAME VARCHAR2(15) NOT NULL,  VDATE DATE,  DOSE NUMBER(1),  VACTIME NUMBER(6,2),  PRIMARY KEY(SSN, CNTID, VNAME),  FOREIGN KEY(SSN) REFERENCES PERSON(SSN),  FOREIGN KEY(CNTID) REFERENCES CENTRE(CNTID),  FOREIGN KEY(VNAME) REFERENCES VACCINE(VNAME));  SQL> CREATE TABLE SELLS  (  VNAME VARCHAR2(15) NOT NULL,  CNAME VARCHAR2(15) NOT NULL,  PRICE NUMBER(4,2) NOT NULL,  PRIMARY KEY(VNAME, CNAME),  FOREIGN KEY(VNAME) REFERENCES VACCINE(VNAME),  FOREIGN KEY(CNAME) REFERENCES PCOMPANY(CNAME)); |

**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:  INSERT INTO PERSON VALUES(10, “Jobe Bata”, 65, “nurse”);  INSERT INTO PERSON VALUES(20, “Monica Kap”, 80, “retire”);  INSERT INTO PERSON VALUES(30, “Peter Good”, 22, “retailer”);  INSERT INTO PERSON VALUES(40, “Kate Lee”, 47, “teacher”);  INSERT INTO PERSON VALUES(50, “Ted Tam”, 50, “Doctor”);  INSERT INTO CENTRE VALUES(1, “DownTn”, “Windsor”, 600000, 10);  INSERT INTO CENTRE VALUES(2, “St Clair”, “Windsor”, 400000, 10);  INSERT INTO CENTRE VALUES(3, “WFCU”, “Windsor”, 900000, 50);  INSERT INTO CENTRE VALUES(4, “Other”, “Windsor”, 600000, 50);  INSERT INTO PCOMPANY VALUES(“Astrazeneca”, NULL);  INSERT INTO PCOMPANY VALUES(“Johnson and Johnson”, “1(732) 524-0400”);  INSERT INTO PCOMPANY VALUES(“Moderna”, “1(866) 633-3762”);  INSERT INTO PCOMPANY VALUES(“Pfizer”, “1(877) 633-2001”);  INSERT INTO PCOMPANY VALUES(“US FDA”, “1(888) 463-6332”);  INSERT INTO VACCINE VALUES(“Astrazeneca”, “Adenoviruses”, “Astrazeneca”);  INSERT INTO VACCINE VALUES(“Johnson and Johnson”, “viral vector”, “Johnson and Johnson”);  INSERT INTO VACCINE VALUES(“Moderna”, “mRNA”, “Moderna”);  INSERT INTO VACCINE VALUES(“Pfizer”, “mRNA”, “Pfizer”);  INSERT INTO VACCINATES VALUES(10, 1);  INSERT INTO VACCINATES VALUES(20, 1);  INSERT INTO VACCINATES VALUES(30, 2);  INSERT INTO VACCINATES VALUES(40, 3);  INSERT INTO VACCINATES VALUES(50, 4);  INSERT INTO ADMINISTERS VALUES(10, 1, “Pfizer” , ‘02-apr-21’, 1, 13.30);  INSERT INTO ADMINISTERS VALUES(20, 4, “Astrazeneca”, ‘04-mar-21’, 1, 9.00);  INSERT INTO ADMINISTERS VALUES(30, 2, “Moderna”, ‘12-may-21’, 1, 11.00);  INSERT INTO ADMINISTERS VALUES(40, 3, “Pfizer”, ‘20-apr-21’, 1, 15.30);  INSERT INTO ADMINISTERS VALUES(50, 4, “Astrazeneca”, ‘20-apr-21’, 1, 10.30);  INSERT INTO SELLS VALUES(“Astrazeneca”, “Astrazeneca”, 5.00);  INSERT INTO SELLS VALUES(“Johnson and Johnson”, “Johnson and Johnson”, 10.00);  INSERT INTO SELLS VALUES(“Moderna”, “Moderna”, 32.00);  INSERT INTO SELLS VALUES(“Pfizer”, “Pfizer”, 20.00); |

**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\_vaccinedroptable.sql is:  delete from VACCINATES;  delete from ADMINISTERS;  delete from SELLS;  delete from VACCINE;  delete from PCOMPANY; delete from PERSON; delete from CENTRE;  commit;  drop table VACCINATES cascade constraints;  drop table ADMINISTERS cascade constraints;  drop table SELLS cascade constraints;  drop table VACCINE cascade constraints;  drop table PCOMPANY cascade constraints;  drop table PERSON cascade constraints;  drop table CENTRE cascade constraints;  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)

|  |
| --- |
| For each query in (a) to (e), show  SQL Query (executed in Sqlplus):   1. SELECT CNAME, CPHONE   FROM PCOMPANY  WHERE  CNAME = (SELECT MADEBY  FROM VACCINE  WHERE VNAME = ‘Pfizer’);  SELECT PC.CNAME, PC.CPHONE  FROM PCOMPANY PC, VACCINE V  WHERE PC.CNAME = V.MADEBY AND V.VNAME = ‘Pfizer’ ;   1. SELECT \*   FROM VACCINE V, SELLS S  WHERE V.VNAME = S.VNAME AND S.PRICE > 5 ;   1. SELECT PC.CNAME, PC.CPHONE, count (S.CNAME)   FROM PCOMPANY PC, SELLS S  WHERE PC.CNAME = S.CNAME  GROUP BY PC.CNAME, PC.PHONE;   1. SELECT S.VNAME, S.CNAME,PC.CPHONE, S.PRICE   FROM SELLS S, PCOMPANY PC, VACCINE V  WHERE PC.CNAME = S.VNAME AND V.VNAME = S.VNAME;   1. SELECT ADMINISTERS.VNAME, VACCINE.FORMULA, SELLS.CNAME   FROM VACCINE, SELLS, ADMINISTERS  WHERE ADMINISTERS.DOSE < 2 AND ADMINISTERS.VNAME = VACCINE.VNAME AND SELLS.VNAME = ADMINISTERS.VNAME;  SELECT V.VNAME, V.FORMULA, S.CNAME  FROM VACCINE V, SELLS V  WHERE S.VNAME = V.VNAME  AND S.VNAME IN (  SELECT A.VNAME  FROM ADMINISTER A  GROUP BY A.VNAME  HAVING max (A.DOSE ) != 2  );  Then, Result of the query retrieved by Sqlplus.  Result will be as below :   1. CNAME CPHONE   -------------------- ---------------  Pfizer 1(877) 633-2001   1. VNAME FORMULA MADEBY   -------------------- --------------- --------------------  Johnson and Johnson viral vector Johnson and Johnson  Moderna mRNA Moderna  Pfizer mRNA Pfizer   1. CNAME CPHONE VNAME   -------------------- --------------- --------------------  Astrazeneca Astrazeneca  Johnson and Johnson 1(732) 524-0400 Johnson and Johnson  Moderna 1(866) 633-3762 Moderna  Pfizer 1(877) 633-2001 Pfizer   1. VNAME CNAME CPHONE PRICE   -------------------- -------------------- --------------- ----------  Astrazeneca Astrazeneca 5  Johnson and Johnson Johnson and Johnson 1(732) 524-0400 10  Moderna Moderna 1(866) 633-3762 32  Pfizer Pfizer 1(877) 633-2001 20   1. VNAME FORMULA CNAME   --------------- --------------- --------------------  Astrazeneca Adenoviruses Astrazeneca  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)

|  |
| --- |
| 1. INSERT INTO CENTRE VALUES(5, 'Devonshire', 'Windsor', 1000000, 10); 2. UPDATE CENTRE   SET BUDGET = 700000  WHERE  CNTNAME = 'DownTn';   1. INSERT INTO PERSON VALUES(60, 'Mary Tama', 21, 'student'); 2. Delete   From Sells  Where cname = 'US FDA'; |

**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)

|  |
| --- |
| 1. SELECT VACCINE.VNAME, SELLS.CNAME   FROM VACCINE, SELLS  WHERE EXISTS (SELECT DOSE  FROM ADMINISTERS  WHERE DOSE = 2  AND VACCINE.VNAME = ADMINISTERS.VNAME  AND SELLS.VNAME = VACCINE.VNAME);  SELECT V.VNAME, V.MADEBY  FROM VACCINE\_V V  WHERE EXISTS(  SELECT A.VNAME  FROM ADMINISTERS\_V A  WHERE V.VNAME = A.VNAME  HAVING MAX(DOSE) >=2  GROUP BY A.VNAME );   1. SELECT VACCINE.VNAME, SELLS.CNAME   FROM VACCINE, SELLS  WHERE NOT EXISTS (SELECT DOSE  FROM ADMINISTERS  WHERE DOSE = 2  AND VACCINE.VNAME = ADMINISTERS.VNAME  AND SELLS.VNAME = VACCINE.VNAME)  AND SELLS.VNAME = VACCINE.VNAME;  SELECT V.VNAME, V.MADEBY  FROM VACCINE\_V V  WHERE NOT EXISTS(  SELECT A.VNAME  FROM ADMINISTERS\_V A  WHERE V.VNAME = A.VNAME  HAVING MAX(DOSE) >=2  GROUP BY A.VNAME  ); |

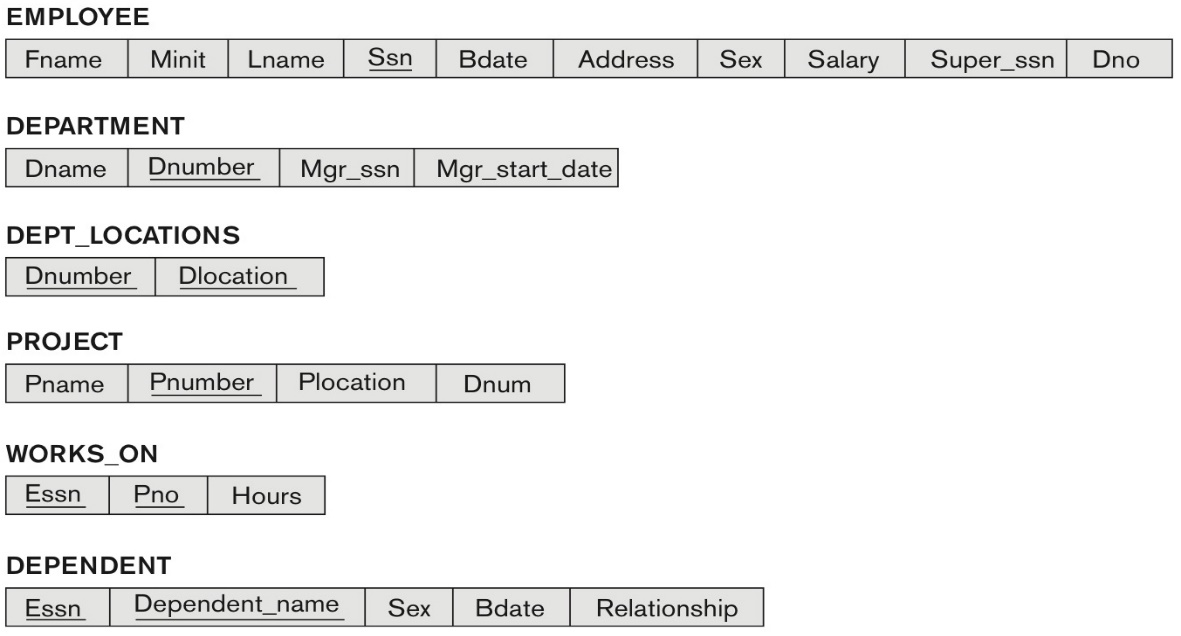
4 (ii). (2.5 marks) distributed as: 0.5 marks for the script file showing correct interaction with Oracle Sqlplus posing these queries; and 2 marks for the correctly posing the queries and retrieving correct results.

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.

1. 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.
2. 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 schema.

****

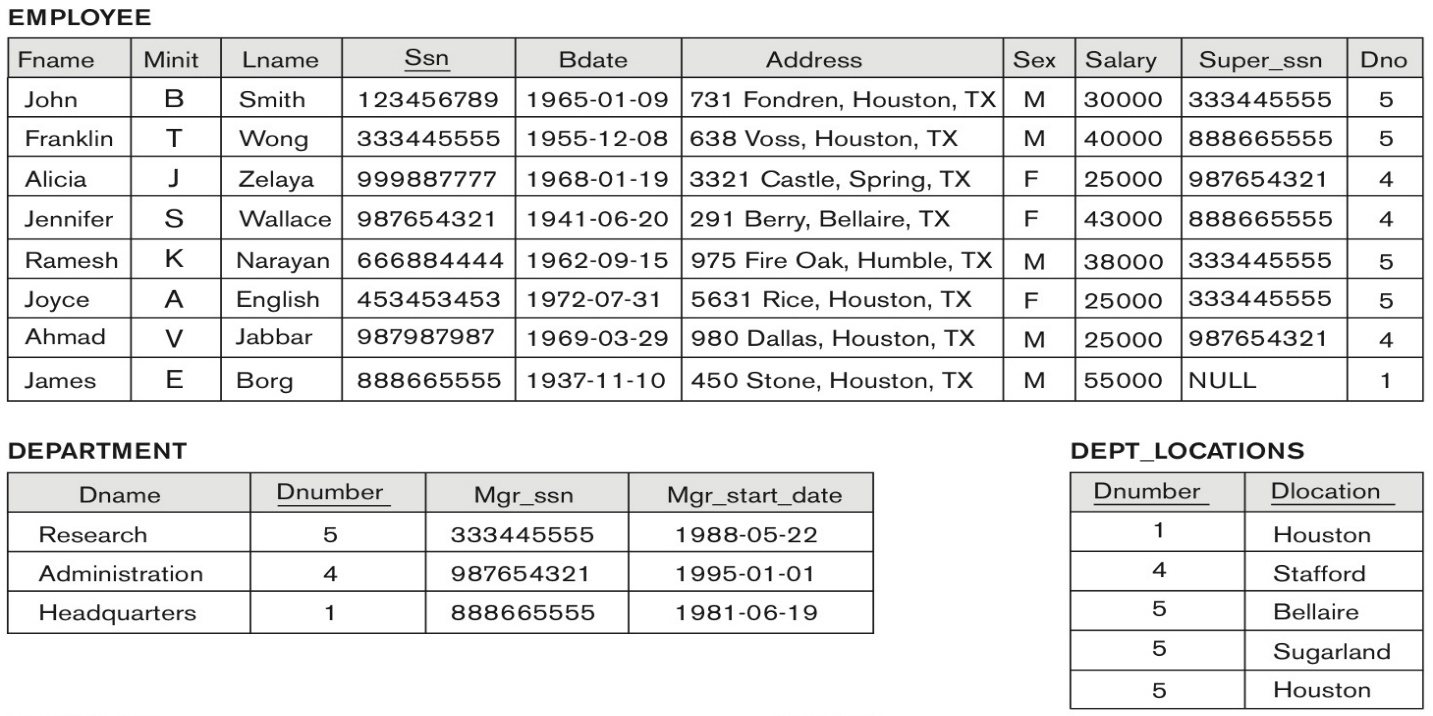
**Solution 5: (5 marks)**

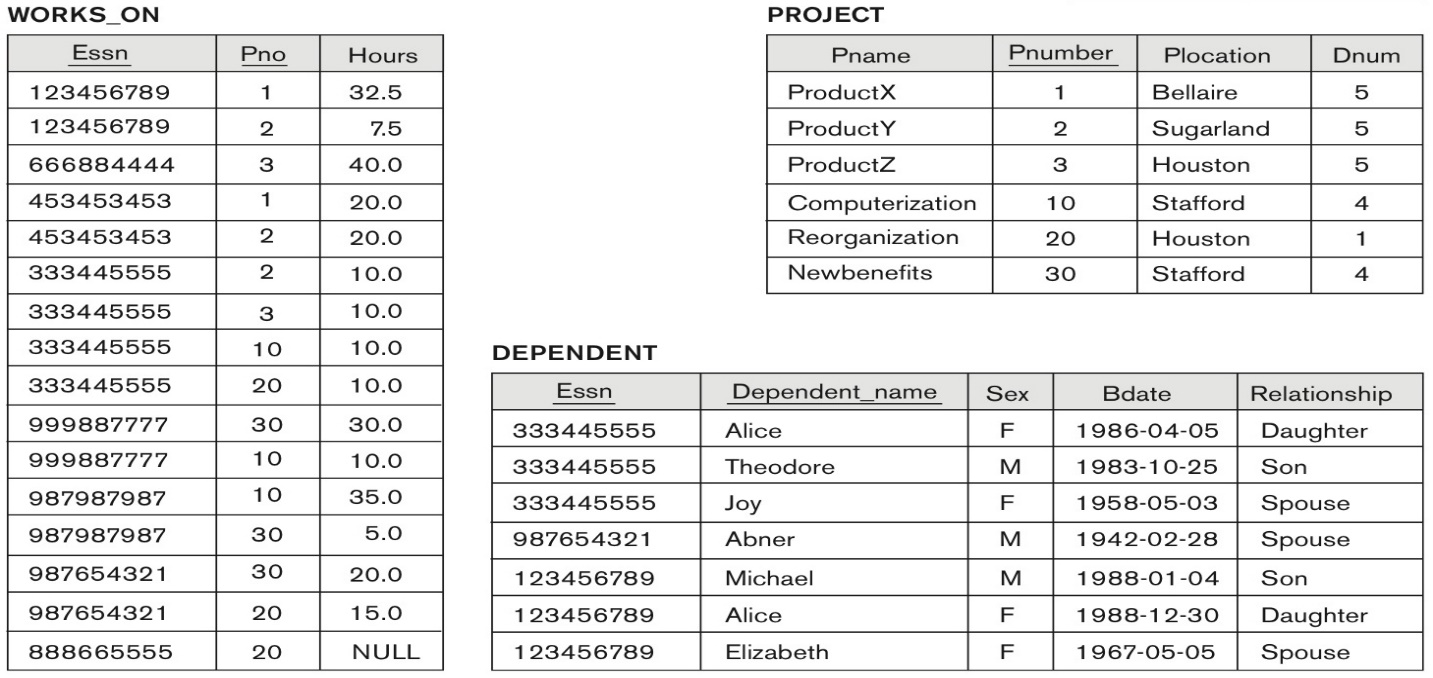
|  |
| --- |
| 1. SELECT PROJECT.PNAME, MAX(WORKS\_ON.HOURS)   FROM (WORKS\_ON JOIN PROJECT ON Pno = Pnumber)  WHERE WORKS\_ON.PNO = PROJECT.PNAME;  SELECT PNAME FROM PROJECT WHERE PNUMBER IN ( SELECT PNO FROM WORKS\_ON WHERE HOURS IN ( SELECT MAX(HOURS) FROM WORKS\_ON) );   1. SELECT HOURS   FROM WORKS\_ON  WHERE EXISTS (SELECT AVG(Hours) AS avgHours  FROM WORKS\_ON  WHERE WORKS\_ON.HOURS > avgHours);  SELECT HOURS FROM WORKS\_ON WHERE HOURS >=ANY ( SELECT AVG(HOURS) FROM WORKS\_ON );   1. SELECT PROJECT.PNAME   FROM PROJECT  WHERE EXITS (SELECT MAX(HOURS) AS maxH, MIN(HOURS) AS minH  FROM WORKS\_ON  WHERE minH + 20 <= maxH);  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 , ρ

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





(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)  Result:  (ii))  Result:  (iii)    Result:  (iv)  Result :  (v)  Result: |