**School:** Computer Science

**Institution:** University of Windsor

**Term:** Winter 2019

**Course:** 03-60-315-1 : Database Management Systems

**Instructor:** Dr. C. I. Ezeife

**Assignment #**3 Solution: Total: 50 marks

**Handed Out:Thurs. Feb. 14, 2019; Due Thurs Mar. 14, 2019**

**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 Sheme** : 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 SQL**

1. Given a database state of the Windsor Music Records database shown in Figure 1.1, with schema shown in Figure 1.2.  
    (Total for que 1 is 15 marks)

Fig 1.1: An Example Database State of Windsor Music Records Database

Musician

|  |
| --- |
| SSN MNAME MADDR PHONE |
| 111111111 Majid Peters 4747 Oakwood Dr 1113331112  222222222 Steven Markel 1231 Berry Ave 2221113332  333333333 Melanie Good 423 Walker Rd 1231235555  444444444 John Doe 5431 Hefty Lane 6751231231 |

Instrument

|  |
| --- |
| INAME MUSICKEY |
| Flute C-Flat  Trumpet E-Flat  Guitar A-Minor  Drums B-Minor |

Album

|  |
| --- |
| AID ATITLE COPYRIGHTDATE FORMAT |
| 111 Recovery 2012-08-08 CD  222 Revival 2015-09-09 CD  333 Welcome 2014-04-03 CD |

Song

|  |
| --- |
| STITLE AUTHOR |
| Hello World 21 Savage Happy All 21 Savage  Love Sosa Chief Keef  Ball For Me Post Malone |

Play

|  |
| --- |
| SSN INAME |
| 111111111 Drums  111111111 Flute 111111111 Guitar  222222222 Trumpet  333333333 Guitar |

Contains

|  |
| --- |
| AID STITLE |
| 111 Hello World  111 Happy All  222 Love Sosa  333 Ball For Me |

Performs

|  |
| --- |
| SSN STITLE |
| 111111111 Hello World  222222222 Love Sosa  333333333 Ball For Me  333333333 Hello World |

Produces

|  |
| --- |
| SSN AID |
| 111111111 111  222222222 222  333333333 333 |

Fig 1.2: Schema of the Windsor Music Records database of Figure 1.1

Musician

|  |  |  |  |
| --- | --- | --- | --- |
| SSN | MName | Maddress | Phone |

Instrument

|  |  |
| --- | --- |
| Iname | Musickey |

Album

|  |  |  |  |
| --- | --- | --- | --- |
| Aid | atitle | copyrightdate | format |

Song

|  |  |
| --- | --- |
| stitle | author |

Play

|  |  |
| --- | --- |
| SSN | Iname |

Contains

|  |  |
| --- | --- |
| Aid | stitle |

Performs

|  |  |
| --- | --- |
| SSN | stitle |

ProducesS

|  |  |
| --- | --- |
| SSN | Aid |

1. List all the referential integrity constraints that should hold on the database schema?

(2.5 marks)

1. Write appropriate SQL DDL statements to define the database with the integrity constraints and store in a text file called userid\_musicschema.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)
2. To insert the data in the database tables, also write appropriate SQL DML instructions in a text file called userid\_musicdata.sql. (2.5 marks)
3. To remove any inserted data and destroy all created tables in the music database, write appropriate SQL DML and DDL statements in a text file called userid\_musicdroptable.sql to first delete all data in the tables and then drop the tables.

(2.5 marks)

1. 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 8 tables by running your .sql file at the SQL prompt with the command:

@userid\_musicschema.sql. After creating your tables successfully, you load your data with the .sql file you created in (iii) above by running @userid\_musicdata.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\_musicdroptable.sql before re-creating the schema and re-loading the data.Then, using a script file, show the contents of all 8 tables in the database by selecting \* 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 Musician; //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 3 .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).  PLAY.SSN --> MUSICIAN  PLAY.INAME --> INSTRUMENT  CONTAINS.AID --> ALBUM  CONTAINS.STITLE --> SONG  PERFORMS.SSN --> MUSICIAN  PERFORMS.STITLE --> SONG  PRODUCES.SSN --> MUSICIAN  PRODUCES.AID --> ALBUM |

**Solution 1 (ii): (mark: 2.5) (Majid, update to have correct results for this database)**

|  |
| --- |
| One possible set of CREATE TABLE statements to define the database is given below given in the file userid\_musicschema.sql is:  create table Musician(  SSN Number(9) not null,  Mname varchar2(20),  Maddr varchar2(20),  phone Number(10),  primary key(SSN));  create table Instrument(  iname varchar(20) not null,  musickey varchar(15),  primary key(iname));  create table Album(  Aid varchar(15) not null,  atitle varchar(20),  copyrightdate DATE,  format varchar(20),  primary key(aid));  create table song(  stitle varchar(30) not null,  author varchar(25),  primary key(stitle));  create table Play(  SSN Number(9) not null,  iname varchar2(20) not null,  primary key(SSN, iname),  foreign key(SSN) references musician(SSN),  foreign key(iname) references instrument(iname));  create table Contains(  Aid varchar(15) not null,  stitle varchar(30) not null,  primary key(Aid, stitle),  foreign key (aid) references album(aid),  foreign key(stitle) references song(stitle));  create table Performs(  SSN number(9) not null,  stitle varchar(30) not null,  primary key(SSN, stitle),  foreign key(SSN) references Musician(SSN),  foreign key(stitle) references song(stitle));  create table produces(SSN number(9) not null,  aid varchar(15) not null,  primary key(ssn , aid),  foreign key(ssn) references musician(ssn),  foreign key(aid) references album(aid));  COMMIT; |

**Solution 1 (iii): (mark: 2.5) (Majid, update to have correct results for this database)**

|  |
| --- |
| One possible set of INSERT INTO TABLE statements to define the database is given below given in the file userid\_musicdata.sql is:  insert into musician values(111111111, 'Majid Peters', '4747 Oakwood Dr', 1113331112);  insert into musician values(222222222, 'Steven Markel', '1231 Berry Ave', 2221113332);  insert into musician values(333333333, 'Melanie Good', '423 Walker Rd', 1231235555);  insert into musician values(444444444, 'John Doe', '5431 Hefty Lane', 6751231231);  insert into Instrument values('Flute', 'C-Flat');  insert into instrument values('Trumpet', 'E-Flat');  insert into instrument values('Guitar', 'A-Minor');  insert into instrument values('Drums', 'B-Minor');  insert into Song values('Hello World', '21 Savage');  insert into Song values('Happy All', '21 Savage');  insert into Song values('Love Sosa', 'Chief Keef');  insert into Song values('Ball For Me', 'Post Malone');  insert into album values('111', 'Recovery', '08-Jun-04', 'CD');  insert into album values('222', 'Revival', '09-Sep-09', 'CD');  insert into album values('333', 'Welcome', '03-Apr-14', 'CD');  insert into play values(111111111, 'Drums');  insert into Play values(111111111, 'Flute');  insert into Play values(111111111, 'Guitar');  insert into Play values(222222222, 'Trumpet');  insert into play values(333333333, 'Guitar');  insert into contains values('111', 'Hello World');  insert into contains values('111', 'Happy All');  insert into contains values('222', 'Love Sosa');  insert into contains values('333', 'Ball For Me');  insert into Performs values(111111111, 'Hello World');  insert into Performs values(222222222, 'Love Sosa');  insert into Performs values(333333333, 'Ball For Me');  insert into performs values(333333333, 'Hello World');  insert into produces values(111111111, 111);  insert into produces values(222222222, 222);  insert into produces values(333333333, 333);  COMMIT; |

**Solution 1 (iv): (mark: 2.5) (Majid, update to have correct results for this database)**

|  |
| --- |
| 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\_musicdroptable.sql is:  delete from contains;  delete from performs;  delete from produces;  delete from play;  delete from musician;  delete from song;  delete from instrument;  delete from album;  COMMIT;  drop table produces;  drop table performs;  drop table contains;  drop table play;  drop table song;  drop table album;  drop table instrument;  drop table musician;  COMMIT; |

1 (v). (5 marks) for the script file showing correct interaction with Oracle Sqlplus creating and loading data in these 8 tables.

**2.** Specify the following 5 queries in SQL on the Music 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 producers of songs by Chief Keef.

(b) Retrieve the copyrightdate and format of all albums that have more than one song.

(c) For each song, retrieve its title, author, album, copyrightdate, and number of performances of the song.

(d) Retrieve the title, Performing musician’s name, and album title of each song.

(e) Retrieve the instrument name, musickey and playing musician name of all instruments that are not played by more than one musician.

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

|  |
| --- |
| 1. Retrieve the names and phone numbers of all producers of songs by Chief Keef.   select M.MNAME, M.PHONE from Musician M, Produces P, Song S, Contains C where M.ssn = p.SSN  and P.AID = C.aid and C.Stitle = S.stitle and s.author = 'Chief Keef';  MNAME PHONE  -------------------- ----------  Steven Markel 2221113332  (b) (b) Retrieve the copyrightdate and format of all albums that have more than one song.  select A.copyrightdate, A.format  from Album A, Contains C where A.AID = C.AID having count(\*) > 1  group by a.copyrightdate, a.format;  COPYRIGHT FORMAT  --------- --------------------  08-JUN-04 CD  (c ) For each song, retrieve its title, author, album, copyrightdate, and number of performances of the song.  select S.stitle, s.author, a.atitle, a.copyrightdate, count(\*)  from Song S, Album A, Contains c, Performs P  where s.stitle = c.stitle and a.aid = c.aid and P.stitle = S.stitle group by  s.stitle, s.author, a.atitle, a.copyrightdate;  STITLE AUTHOR ATITLE COPYRIGHTDATE COUNT(\*)  ---------------------------------------------------------------------------------------------------------------------------  Love Sosa Chief Keef Revival 09-Sep-09 1  Ball For Me Post Malone Welcome 03-Apr-14 1  Hello World 21 Savage Recovery 08-Jun-04 2  (d) Retrieve the title, Performing musician’s name, and album title of each song.  select S.stitle, M.mname, a.atitle  from Song S, Musician M, Album A, Contains C, Performs P  where S.stitle = C.stitle and C.aid = A.aid and p.stitle = s.stitle and m.ssn = p.ssn;  STITLE MNAME ATITLE  ------------------------------ -------------------- --------------------  Hello World Majid Peters Recovery  Love Sosa Steven Markel Revival  Ball For Me Melanie Good Welcome  Hello World Melanie Good Recovery  **(e)** Retrieve the instrument name, musickey and playing musician name of all instruments that are not played by more than one musician.    select I.iname, I.musickey, M.mname from  Instrument I, Musician M, play p  Where I.iname = p.iname and M.ssn = p.ssn  and I.iname IN ( SELECT iname  FROM Play  Group by iname  Having count(\*) <= 1);  and S.FDate = I.FDate and S.Leg\_number > 1 );  Another solution for (e) is:  select T.iname, I.musickey, M.mname from  (select iname from Play having count(\*) = 1 group by iname) T, Instrument I, Musician M, Play p  where T.iname = I.iname and M.ssn = P.ssn and I.iname = p.iname;  INAME MUSICKEY MNAME  -------------------- --------------- --------------------  Flute C-Flat Majid Peters  Drums B-Minor Majid Peters  Trumpet E-Flat Steven Markel |

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 musician <555555555, ‘Tintin Arial’, ‘401 Sunset Ave’, 5192311010> in the database.

(b) Change the address of musician ‘‘John Doe’ to 500 Airport Rd.

(c) Insert a new album <444, 'Good Heart’, ’14-Feb-2017’, ‘CD’>.

(d) Delete all performance records for songs whose name is 'Hello World'.

**Solution 3 (i):** (5 marks)

|  |
| --- |
| (a) Insert a new musician <555555555, ‘Tintin Arial’, ‘401 Sunset Ave’, 5192311010> in the database.  INSERT INTO MUSICIAN VALUES(555555555, 'Tintin Arial', '401 Sunset Ave', 5192311010);  (b) Change the address of musician ‘‘John Doe’ to 500 Airport Rd.  UPDATE MUSICIAN  SET MADDR = '500 Airport Rd'  WHERE MName='John Doe';  (c) Insert a new album <444, 'Good Heart’, ’14-Feb-2017’, ‘CD’>.  INSERT INTO ALBUM VALUES (444, 'Good Heart', '14-Feb-17', 'CD');  (d) Delete all performance records for songs whose name is 'Hello World'.  DELETE FROM PERFORMS  WHERE STITLE='Hello World'; |

**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 album title and copyrightdate of all albums with 2 songs.  
(b) Retrieve the album title and copyrightdate of all albums that do not have 2 songs.

**Solution 4 (i):** (2.5 marks)

|  |
| --- |
| (a) Retrieve the album title and copyrightdate of all albums with 2 songs.  select A.atitle, a.copyrightdate  from Album A  where exists(  select c.aid from Contains c where c.aid = a.aid having count(\*) = 2 group by aid);  (b) Retrieve the album title and copyrightdate of all albums that do not have 2 songs.  select A.atitle, a.copyrightdate  from Album A  where NOT exists(  select c.aid from Contains c where c.aid = a.aid having count(\*) = 2 group by aid); |

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.

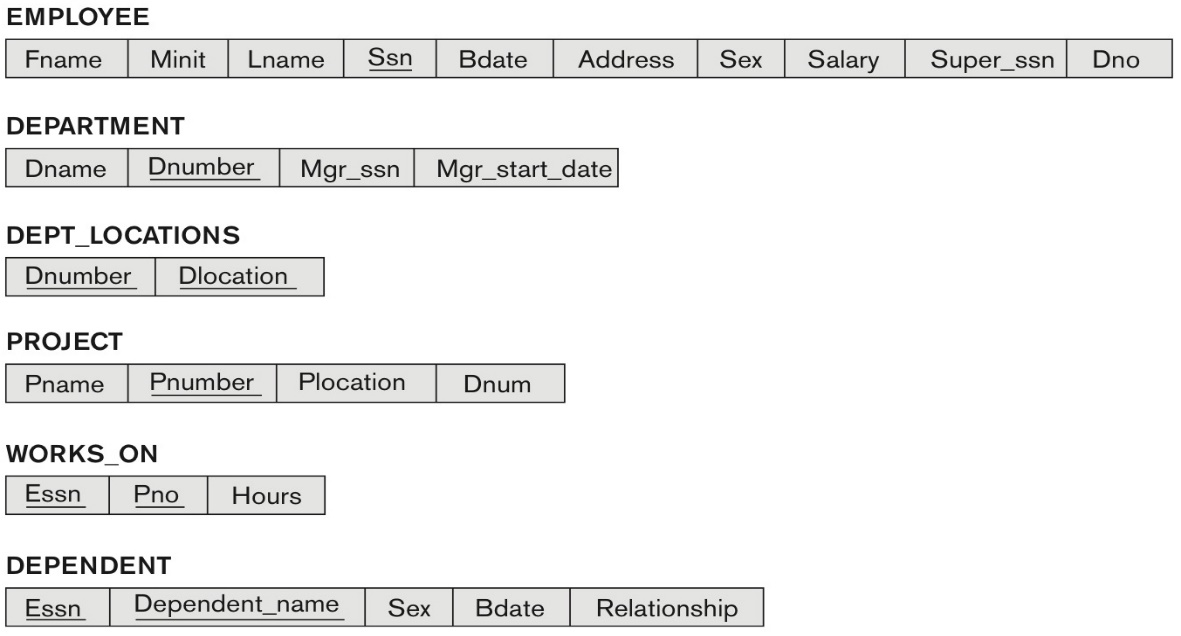
**5**. In SQL, specify the following 3 queries on the COMPANY database of Figure 5.5 using the concept of nested queries and the concepts described in chapter 7. (Total for que 5 is 5 marks)

a. Retrieve the names of all employees who work in the department that has the employee with the lowest salary among all employees.

b. Retrieve the names of all employees whose supervisor’s supervisor has '333445555' for Ssn.

c. Retrieve the names of employees who make at least $10,000 more than the employee who is paid the least in the company.

**Figure 5.5** Schema diagram for the COMPANY relational database schema.

****

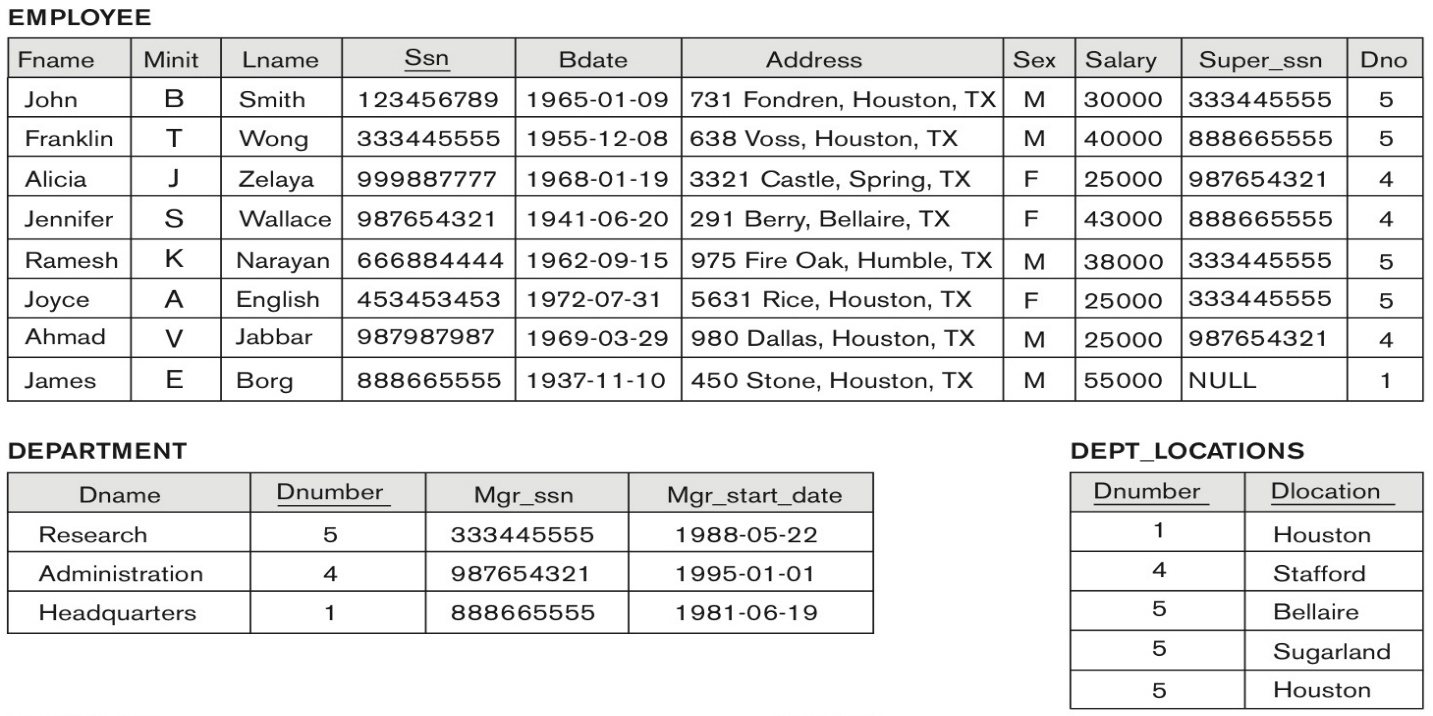
**Solution 5: (5 marks)**

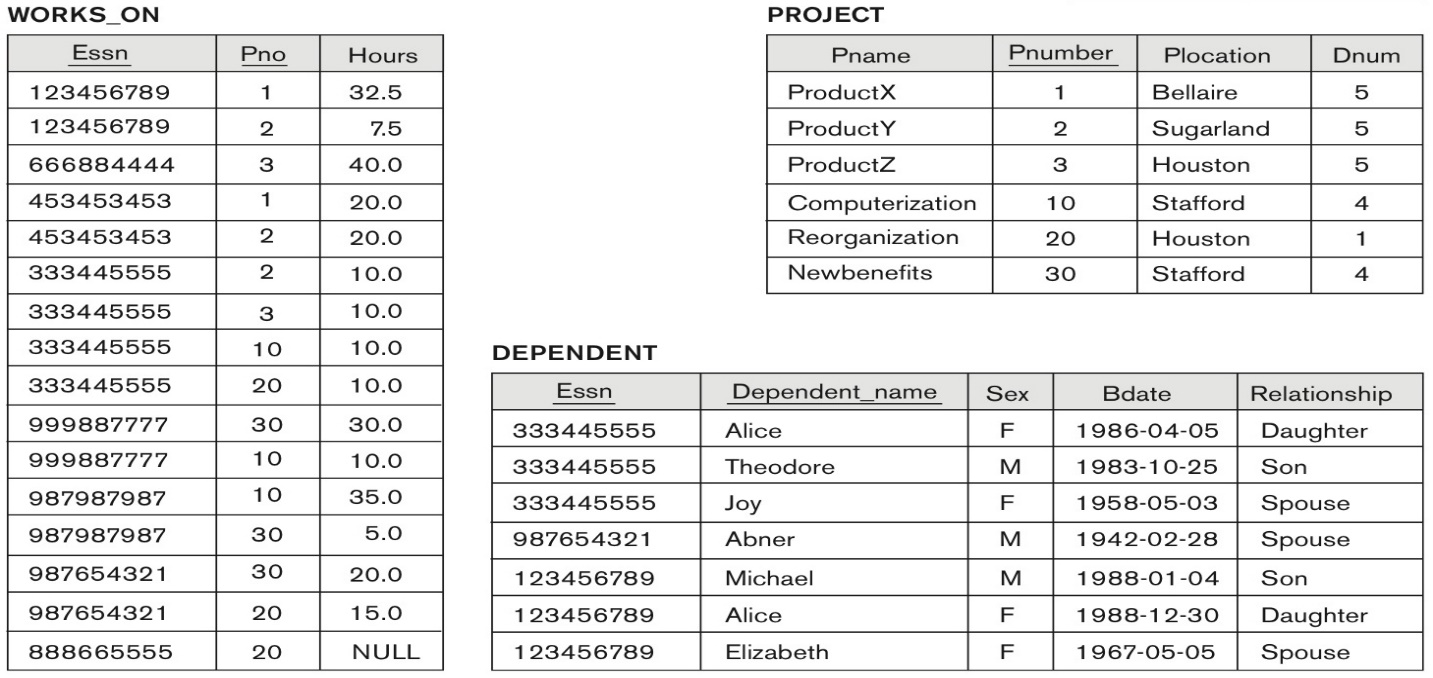
|  |
| --- |
| a) SELECT LNAME FROM EMPLOYEE WHERE DNO IN  ( SELECT DNO FROM EMPLOYEE WHERE SALARY IN  ( SELECT MIN(SALARY) FROM EMPLOYEE) );  b) SELECT LNAME FROM EMPLOYEE WHERE SUPER\_SSN IN ( SELECT SSN FROM EMPLOYEE WHERE SUPER\_SSN = ‘333445555’ );  c) SELECT LNAME FROM EMPLOYEE WHERE SALARY >= 10000 + ( SELECT MIN(SALARY) FROM EMPLOYEE); |

**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)

**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)  E 🡨 ( EMPLOYEE Ssn =Essn and Fname = Dependent\_name (DEPENDENT))  R 🡨 π Lname,Fname ( E)  Result (empty):  LNAME FNAME  (ii))Borg\_ssn 🡨 πSsn (σ Fname= ‘James’ and Lname = ‘Borg’ (EMPLOYEE))  Borg\_emps 🡨 ( EMPLOYEE Super\_ssn = Ssn (Borg\_ssn))  Result 🡨 π Lname,Fname ( 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)  Proj\_Hours (Pno, Tot\_Hrs) 🡨 Pno Sum Hours (Works\_on)  Result 🡨 π Pname,Tot\_hrs ( Proj\_Hours Pno = Pnumber (PROJECT))    Result:  PNAME TOT\_HRS  ProductX 52.5  ProductY 37.5  ProductZ 50.0  Computerization 55.0  Reorganization 25.0  Newbenefits 55.0  (iv)  PROJ\_EMPS(PNO,SSN) <-- π pno, Essn (WORKS\_ON)  ALL\_PROJS(PNO) <-- π PNUMBER (PROJECT)  EMPS\_ALL\_PROJS <-- PROJ\_EMPS -:- ALLPROJS /\* DIVISION operation \*/  RESULT <-- π LNAME,FNAME (EMPLOYEE EMP\_ALL\_PROJS) /\*natural join on ssn\*/  Result (empty):  LNAME FNAME  (v) RESULT(MAX\_F\_SAL) <-- 🡨 maximum salary ((σ SEX = ‘F’ EMPLOYEE))  Result:  max\_F\_SAL  43000 |