School: Computer Science
Institution: University of Windsor
Term: Fall 2018
Course: 03-60-315-1 : Database Management Systems
Instructor: Dr. C. I. Ezeife
Assignment #3 Solution: Total: 50 marks
Handed Out:Thurs. Oct. 25, 2018; Due Thurs Nov. 15, 2018

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 AIRLINE RESERVATION 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 Parts of the Airline Database of Fig 5.8 of book

AIRPORT

Airport_Code	Name	City	State
YYZ	Pearson Airport	Toronto	ON
YTZ	Island Airport	Toronto	ON
YQT	Thunder Bay Airport	Thunder Bay	ON
YQG	Windsor Airport	Windsor	ON
YYC	Calgary Airport	Calgary	AB

FLIGHT

Flight_number	Airline	Weekdays
WJ250	WestJet	3
WJ261	WestJet	4
AC275	Air Canada	2
AC300	Air Canada	3
AC320	Air Canada	2
PA233	Porter Airline	5
PA280	Porter Airline	4

FLIGHT_LEG

Flight_number	Leg_number	Departure_airport_code	Scheduled_departure_time	Arrival_airport_code	Scheduled_arrival_time
WJ250	1	YQG	08.00	YYZ	09.00
WJ250	2	YYZ	12.00	YYC	17.00
AC275	1	YQT	10.00	YYZ	11.00
AC275	2	YYZ	13.00	YQG	14.00
PA233	1	YQG	16.00	YYZ	17.00
PA280	1	YYZ	11.00	YQG	12.00

LEG_INSTANCE

Flight_number	Leg_number	FDate	Num_available_seats	Airplane_id	Departure_airport_code	Departure_time	Arrival_airport_code	Arrival_time
WJ250	1	15-JAN-17	200	Boeing_777	YQG	8.00	YYZ	8.50

WJ250	2	15-JAN-17	200	Boeing_777	YYZ	12.30	YYC	16.50
AC275	1	10-JAN-17	60	Airbus_200	YQT	10.00	YYZ	11.00
AC275	2	10-JAN-17	60	Airbus_200	YYZ	13.00	YQG	14.00
PA233	1	30-JAN-17	120	Boeing_737	YQG	16.00	YYZ	17.00
PA280	1	20-JAN-17	120	Boeing_737	YYZ	11.00	YQG	12.00

FARE

Flight_number	Fare_code	Amount	Restrictions
WJ250	F1	$250	None
WJ250	F2	$350	None
AC275	F1	$275	Non-refundable
AC275	F2	$300	Non-refundable
PA233	F3	$150	None
PA280	F3	$150	None

SEAT_RESERVATION

Flight_number	Leg_number	FDate	Seat_number	Customer_name	Customer_phone
WJ250	1	15-JAN-17	20A	Mariane Mooer	519-253-3000
WJ250	2	15-JAN-17	13D	Mariane Mooer	519-253-3000
WJ250	1	15-JAN-17	6A	Tony Appa	226-253-4000
WJ250	2	15-JAN-17	7D	Tony Appa	226-253-4000
WJ250	1	15-JAN-17	6B	Karen Appa	226-253-4000
WJ250	2	15-JAN-17	7C	Karen Appa	226-253-4000
AC275	1	10-JAN-17	5C	Mark Black	226-212-5341
AC275	2	10-JAN-17	14B	Mark Black	226-212-5341
PA233	1	30-JAN-17	5A	Peter Opo	519-444-1234
PA280	1	20-JAN-17	16D	Chris Natta	226-123-5555

Fig 1.2: Schema of the Airline Reservation database of Figure 1.1

AIRPORT

Airport_Code	Name	City	State

FLIGHT

Flight_number	Airline	Weekdays

FLIGHT_LEG

Flight_number	Leg_number	Departure_airport_code	Scheduled_departure_time	Arrival_airport_code	Scheduled_arrival_time

LEG_INSTANCE

Flight_number	Leg_number	FDate	Num_available_seats	Airplane_id	Departure_airport_code	Departure_time	Arrival_airport_code	Arrival_time

FARE

Flight_number	Fare_code	Amount	Restrictions

SEAT_RESERVATION

Flight_number	Leg_number	FDate	Seat_number	Customer_name	Customer_phone

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_airlineschema.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_airlinedata.sql. (2.5 marks)

iv. To remove any inserted data and destroy all created tables in the airline reservation database, write appropriate SQL DML and DDL statements in a text file called userid_airlinedroptable.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 SQLfor the 6 tables by running your .sql file at the SQL prompt with the command: @userid_airlineschema.sql. After creating your tables successfully, you load your data with the .sql file you created in (iii) above by running @userid_airlinedata.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_airlinedroptable.sql before re-creating the schema and re-loading the data.Then, using a script file, show the contents of all 6 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 AIRPORT;   //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). FLIGHT_LEG.FLIGHT_NUMBER --> FLIGHT
FLIGHT_LEG.DEPARTURE_AIRPORT_CODE --> AIRPORT
FLIGHT_LEG.ARRIVAL_AIRPORT_CODE --> AIRPORT
LEG_INSTANCE.(FLIGHT_NUMBER,LEG_NUMBER) --> FLIGHT_LEG
LEG_INSTANCE.DEPARTURE_AIRPORT_CODE --> AIRPORT
LEG_INSTANCE.ARRIVAL_AIRPORT_CODE --> AIRPORT
LEG_INSTANCE.AIRPLANE_ID --> AIRPLANE
FARES.FLIGHT_NUMBER --> FLIGHT
SEAT_RESERVATION.(FLIGHT_NUMBER,LEG_NUMBER,DATE) --> LEG_INSTANCE

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_airlineschema.sql is:

```
CREATE TABLE AIRPORT
(Airport_Code CHAR(3) NOT NULL,
 Name VARCHAR2(25),
 City VARCHAR2(15),
 State CHAR(3),
 PRIMARY KEY(Airport_Code));

CREATE TABLE FLIGHT
(Flight_number VARCHAR2(5) NOT NULL,
 Airline VARCHAR2(15),
Weekdays NUMBER(1),
 PRIMARY KEY(Flight_number));

CREATE TABLE FLIGHT_LEG
```

```
(Flight_number VARCHAR2(5) NOT NULL,
 Leg_number NUMBER(1) NOT NULL,
 Departure_airport_code CHAR(3),
 Scheduled_departure_time NUMBER(4,2),
 Arrival_airport_code CHAR(3),
 Scheduled_arrival_time NUMBER(4,2),
 PRIMARY KEY(Flight_number, Leg_number),
 FOREIGN KEY (Flight_number) REFERENCES FLIGHT(Flight_number)
 ON DELETE CASCADE,
 FOREIGN KEY (Departure_airport_code) REFERENCES AIRPORT(Airport_Code)
 ON DELETE CASCADE,
 FOREIGN KEY (Arrival_airport_code) REFERENCES AIRPORT(Airport_Code)
 ON DELETE CASCADE);

CREATE TABLE LEG_INSTANCE
(Flight_number VARCHAR2(5) NOT NULL,
 Leg_number NUMBER(1) NOT NULL,
 FDate  DATE NOT NULL,
 Num_available_seats  NUMBER(3),
 Airplane_id VARCHAR2(15),
 Departure_airport_code CHAR(3),
 Departure_time NUMBER(4,2),
 Arrival_airport_code CHAR(3),
 Arrival_time NUMBER(4,2),
 PRIMARY KEY(Flight_number, Leg_number, FDate),
 FOREIGN KEY (Flight_number) REFERENCES FLIGHT(Flight_number)
 ON DELETE CASCADE,
 FOREIGN KEY (Departure_airport_code) REFERENCES AIRPORT(Airport_Code)
 ON DELETE CASCADE,
 FOREIGN KEY (Arrival_airport_code) REFERENCES AIRPORT(Airport_Code)
 ON DELETE CASCADE);

CREATE TABLE FARE
(Flight_number VARCHAR2(5) NOT NULL,
 Fare_code VARCHAR2(2),
 Amount NUMBER(6,2),
 Restrictions VARCHAR2(15),
 PRIMARY KEY(Flight_number,  Fare_code),
 FOREIGN KEY (Flight_number) REFERENCES FLIGHT(Flight_number)
 ON DELETE CASCADE);

CREATE TABLE SEAT_RESERVATION
(Flight_number VARCHAR2(5) NOT NULL,
 Leg_number NUMBER(1) NOT NULL,
 FDate  DATE NOT NULL,
 Seat_number VARCHAR2(3),
 Customer_name  VARCHAR2(25),
 Customer_phone VARCHAR2(12),
 PRIMARY KEY(Flight_number, Leg_number, FDate, Seat_number),
 FOREIGN KEY (Flight_number) REFERENCES FLIGHT(Flight_number)
 ON DELETE CASCADE);


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_airlinedata.sql is:

```
INSERT INTO AIRPORT
VALUES ('YYZ', 'Pearson Airport', 'Toronto', 'ON');

COMMIT;

INSERT INTO AIRPORT
VALUES ('YTZ', 'Island Airport', 'Toronto', 'ON');

COMMIT;

INSERT INTO AIRPORT
VALUES ('YQT', 'Thunder Bay Airport', 'Thunder Bay', 'ON');
COMMIT;

INSERT INTO AIRPORT
VALUES ('YQG', 'Windsor Airport', 'Windsor', 'ON');
COMMIT;

INSERT INTO AIRPORT
VALUES ('YYC', 'Calgary Airport', 'Calgary', 'AB');
COMMIT;

-- Flight begins

INSERT INTO FLIGHT
VALUES ('WJ250', 'WestJet', 3);
COMMIT;

INSERT INTO FLIGHT
VALUES ('WJ261', 'WestJet', 4);
COMMIT;

INSERT INTO FLIGHT
VALUES ('AC275', 'Air Canada', 2);
COMMIT;

INSERT INTO FLIGHT
VALUES ('AC300', 'Air Canada', 3);
COMMIT;

INSERT INTO FLIGHT
VALUES ('AC320', 'Air Canada', 2);
COMMIT;

INSERT INTO FLIGHT
VALUES ('PA233', 'Porter Airline', 5);
COMMIT;
```

```
INSERT INTO FLIGHT
VALUES ('PA280', 'Porter Airline', 4);
COMMIT;

-- FLIGHT_LEG begins

INSERT INTO FLIGHT_LEG
VALUES ('WJ250', 1, 'YQG', 08.00, 'YYZ', 09.00);
COMMIT;

INSERT INTO FLIGHT_LEG
VALUES ('WJ250', 2, 'YYZ', 12.00, 'YYC', 17.00);
COMMIT;

INSERT INTO FLIGHT_LEG
VALUES ('AC275', 1, 'YQT', 10.00, 'YYZ', 11.00);
COMMIT;

INSERT INTO FLIGHT_LEG
VALUES ('AC275', 2, 'YYZ', 13.00, 'YQG', 14.00);
COMMIT;

INSERT INTO FLIGHT_LEG
VALUES ('PA233', 1, 'YQG', 16.00, 'YYZ', 17.00);
COMMIT;

INSERT INTO FLIGHT_LEG
VALUES ('PA280', 1, 'YYZ', 11.00, 'YQG', 12.00);
COMMIT;

-- LEG_INSTANCE begins

INSERT INTO LEG_INSTANCE
VALUES ('WJ250', 1, '15-JAN-17', 200, 'Boeing_777', 'YQG', 08.00, 'YYZ', 08.50);
COMMIT;

INSERT INTO LEG_INSTANCE
VALUES ('WJ250', 2, '15-JAN-17', 200, 'Boeing_777', 'YYZ', 12.30, 'YYC', 16.50);
COMMIT;

INSERT INTO LEG_INSTANCE
VALUES ('AC275', 1, '10-JAN-17', 60, 'Airbus_200', 'YQT', 10.00, 'YYC', 11.00);
COMMIT;

INSERT INTO LEG_INSTANCE
VALUES ('AC275', 2, '10-JAN-17', 60, 'Airbus_200', 'YYZ', 13.00, 'YQG', 14.00);
COMMIT;

INSERT INTO LEG_INSTANCE
VALUES ('PA233', 1, '30-JAN-17', 120, 'Boeing_737', 'YQG', 16.00, 'YYZ', 17.00);
COMMIT;

INSERT INTO LEG_INSTANCE
```

```
VALUES ('PA233', 1, '20-JAN-17', 120, 'Boeing_737', 'YYZ', 11.00, 'YQG', 12.00);
COMMIT;

-- FARE begins

INSERT INTO FARE
VALUES ('WJ250', 'F1', 250, 'None');
COMMIT;

INSERT INTO FARE
VALUES ('WJ250', 'F2', 350, 'None');
COMMIT;

INSERT INTO FARE
VALUES ('AC275', 'F1', 275, 'Non-refundable');
COMMIT;

INSERT INTO FARE
VALUES ('AC275', 'F2', 300, 'Non-refundable');
COMMIT;

INSERT INTO FARE
VALUES ('PA233', 'F3', 150, 'None');
COMMIT;

INSERT INTO FARE
VALUES ('PA280', 'F3', 150, 'None');
COMMIT;

-- SEAT_RESERVATION begins

INSERT INTO SEAT_RESERVATION
VALUES ('WJ250', 1, '15-JAN-17', '20A', 'Mariane Mooer', '519-253-3000');
COMMIT;

INSERT INTO SEAT_RESERVATION
VALUES ('WJ250', 2, '15-JAN-17', '13D', 'Mariane Mooer', '519-253-3000');
COMMIT;

INSERT INTO SEAT_RESERVATION
VALUES ('WJ250', 1, '15-JAN-17', '6A', 'Tony Appa', '226-253-4000');
COMMIT;

INSERT INTO SEAT_RESERVATION
VALUES ('WJ250', 2, '15-JAN-17', '7D', 'Tony Appa', '226-253-4000');
COMMIT;

INSERT INTO SEAT_RESERVATION
VALUES ('WJ250', 1, '15-JAN-17', '6B', 'Karen Appa', '226-253-4000');
COMMIT;

INSERT INTO SEAT_RESERVATION
VALUES ('WJ250', 2, '15-JAN-17', '7C', 'Karen Appa', '226-253-4000');
```

```
COMMIT;

INSERT INTO SEAT_RESERVATION
VALUES ('AC275', 1, '10-JAN-17', '5C', 'Mark Black', '226-212-5341');
COMMIT;

INSERT INTO SEAT_RESERVATION
VALUES ('AC275', 2, '10-JAN-17', '14B', 'Mark Black', '226-212-5341');
COMMIT;
INSERT INTO SEAT_RESERVATION
VALUES ('PA233', 1, '30-JAN-17', '5A', 'Peter Opo', '519-444-1234');
COMMIT;

INSERT INTO SEAT_RESERVATION
VALUES ('PA280', 1, '20-JAN-17', '16D', 'Chris  Natta', '226-123-5555');
COMMIT;
```

Solution 1 (iv): (mark: 2.5)

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

```
delete from airport;

commit;

delete from flight;

commit;

delete from flight_leg;

commit;
delete from leg_instance;
commit;

delete from fare;
commit;

delete from SEAT_RESERVATION;
commit;

drop table SEAT_RESERVATION ;
drop table fare ;
drop table LEG_INSTANCE ;
drop table FLIGHT_LEG ;
drop table AIRPORT ;
drop table  FLIGHT ;

commit;
```

1 (v). (5 marks) for the script file showing correct interaction with Oracle Sqlplus creating and loading data in these 6 tables.

2. Specify the following 5 queries in SQL on the airline 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 this questions in a script file called username_assn3que2.txt. (5 marks)

(a) Retrieve the names and flight numbers of all customers who have taken a flight departing from Windsor.
(b) Retrieve the flight number and flight legs of all flights that have more than one leg.
(c) For each flight, retrieve the flight number, airline, flight date, and number of customers in the flight.
(d) Retrieve the name, seat number, and flight number of each customer in all Air Canada flights.
(e) Retrieve the names and flight numbers, departure and arrival airports of all customers who did not fly in a flight that is more than the first leg.

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

(a) Retrieve the names and flight numbers of all customers who have taken a flight departing from Windsor.

```
SELECT  S.Customer_name, S.Flight_number
   FROM SEAT_RESERVATION S, FLIGHT_LEG F, AIRPORT A
   WHERE F.Departure_airport_code = A.Airport_Code
   and A.City = 'Windsor' AND f.Flight_number = S.Flight_number;
```

Result of Query (a):
-Retrieve result follows:

```
CUSTOMER_NAME             FLIGH
------------------------- -----
Mariane Mooer             WJ250
Tony Appa                 WJ250
Karen Appa                WJ250
Peter Opo                 PA233
```

(b) Retrieve the flight number and flight legs of all flights that have more than one leg.

```
SELECT I.Flight_number, I.Leg_number, F.Airline
FROM FLIGHT_LEG I, FLIGHT F
WHERE F.Flight_number = I.Flight_number AND
  I.Leg_number > 1;
```

Result of query b given below:

```
FLIGH LEG_NUMBER AIRLINE
----- ---------- ---------------
AC275          2 Air Canada
WJ250          2 WestJet
```

(c) For each flight, retrieve the flight number, airline, flight date, and number of customers in the flight.

```
SELECT S.Flight_number, F.Airline, S.FDate, Count(Customer_name)
FROM SEAT_RESERVATION S, FLIGHT F
WHERE F.Flight_number = S.Flight_number
GROUP BY S.Flight_number,  F.Airline, S.FDate;
```

-- Result of query c is given below:

```
FLIGH AIRLINE        FDATE     COUNT(CUSTOMER_NAME)
----- --------------- --------- --------------------
PA233 Porter Airline  30-JAN-17                1
AC275 Air Canada      10-JAN-17                 2
PA280 Porter Airline  20-JAN-17                1
WJ250 WestJet        15-JAN-17                6
```

(d) Retrieve the name, seat number, and flight number of each customer in all Air Canada flights.

```
 SELECT S.Customer_name, S.Seat_number, S.Flight_number
FROM SEAT_RESERVATION S, FLIGHT F
WHERE F.Flight_number = S.Flight_number
and  F.Airline='Air Canada';
```

Result for query d is:

```
CUSTOMER_NAME           SEA FLIGH
------------------------  ---   -----
Mark Black              14B  AC275
Mark Black               5C  AC275
```

(e) Retrieve the names and flight numbers, departure and arrival airports of all customers who did not fly in a flight that is more than the first leg.
(Note that only attributes in the outer query retrievable as results of the query and thus, a join of a copy of Leg_instance table is used in the outer query to get the departure and arrival airport codes and the NOT exists is used in the nested query to find avoid tuples that flew in a flight with more than one leg).

```
SELECT S.Customer_name, S.Flight_number, K.Departure_airport_code,
K.Arrival_airport_code
FROM SEAT_RESERVATION S, LEG_INSTANCE K
WHERE S.Flight_number= K.Flight_number AND  S.Leg_number = K.Leg_number
and S.FDate = K.FDate and NOT EXISTS ( SELECT *
FROM LEG_INSTANCE I
WHERE S.Flight_number= I.Flight_number AND  S.Leg_number = I.Leg_number
and S.FDate = I.FDate and S.Leg_number > 1 );
```

Query result e is

```
CUSTOMER_NAME FLIGH    DEP ARR
------------------------ -----       ---    ---
Mark Black            AC275   YQT YYC
Peter Opo             PA233   YQG YYZ
Tony Appa             WJ250   YQG YYZ
Mariane Mooer         WJ250   YQG YYZ
Karen Appa            WJ250   YQG YYZ
Chris  Natta          PA280    YYZ  YYC
```

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 the 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 airport <'YTT', ‘Tintin Airport’, ‘Tintin’, 'ON'> in the database.

(b) Change the Customer_phone of customer ‘Tony Appa’ to 519-253-4001.

(c) Insert a new flight <'AC331,'Air Canada', 2>.

(d) Delete all reservation records for the customer whose name is 'Mariane Mooer '.

Solution 3 (i): (5 marks)

```
(a) Insert a new airport <'YTT', ‘Tintin Airport’, ‘Tintin’, 'ON'> in the database.
(a) INSERT INTO AIRPORT
VALUES ('YTT', ‘Tintin Airport’, ‘Tintin’, 'ON');

(b) Change the Customer_phone of customer ‘Tony Appa’ to 519-253-4001
(b) UPDATE SEAT_RESERVATION
SET Customer_phone = ‘519-253-4001’
 WHERE Name=' Tony Appa ';

(c) Insert a new flight <'AC331,'Air Canada', 2>.
(c) INSERT INTO FLIGHT
VALUES ('AC331,'Air Canada', 2);

(d) Delete all reservation records for the customer whose name is 'Mariane Mooer '.
(d) DELETE FROM SEAT_RESERVATION
WHERE Name=''Mariane Mooer ';
```

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, 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 customer names and flight number of all 2-leg flyers (customers who have flights with 2 legs in all their flights).
(b) Retrieve the customer names and flight number of all customers who do not have any 2-leg flight in any of their flights.

Solution 4 (i): (2.5 marks)

(a) Retrieve the customer names and flight number of all 2-leg flyers (customers who have flights with 2 legs in all their flights).

```
(a) SELECT  S.Customer_name, S.Flight_number
FROM SEAT_RESERVATION  S
WHERE NOT EXISTS ( SELECT *
FROM SEAT_RESERVATION  T
WHERE T._Flight_number = S.Flight_number AND  T._Leg_number =S. Leg_number  AND NOT(T.
Leg_number=2));
```

(b) Retrieve the customer names and flight number of all customers who do not have any 2-leg flight in any of their flights.

(b)

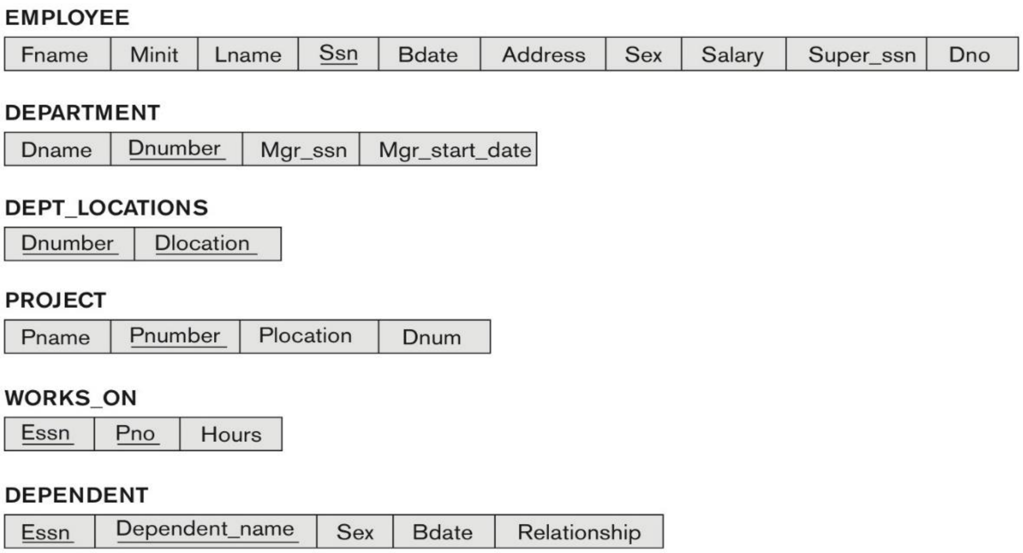
```
SELECT  S.Customer_name, S.Flight_number
FROM SEAT_RESERVATION  S
WHERE NOT EXISTS ( SELECT *
FROM SEAT_RESERVATION  T
WHERE T._Flight_number = S.Flight_number AND T._Leg_number=2);
```

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



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

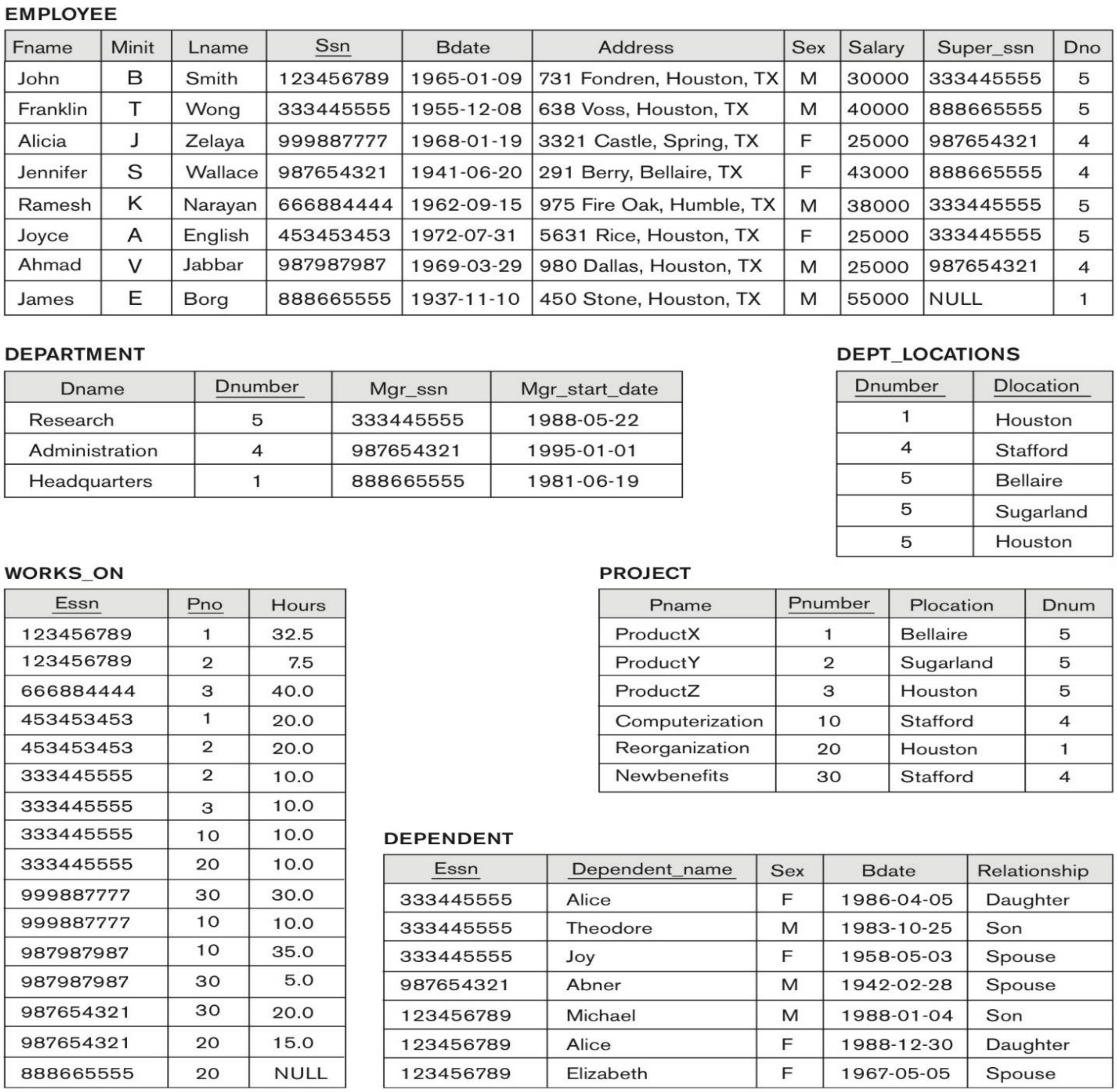
Solution 5: (5 marks)

```
a)   SELECT 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.



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)

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

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

Result (empty):

```
LNAME FNAME
```

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

$Borg\_emps \leftarrow (\text{ EMPLOYEE } \bowtie_{Super\_ssn = Ssn} (Borg\_ssn))$

$Result \leftarrow \pi_{Lname,Fname} (Borg\_emps)$

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)

$PROJ\_HOURS (Pno, Tot\_Hrs) \leftarrow Pno \ \mathfrak{I}_{Sum\ Hours} (\text{WORKS\_ON})$

$RESULT \leftarrow \pi_{Pname,Tot\_hrs} (PROJ\_HOURS \bowtie_{Pno = Pnumber} (\text{PROJECT}))$

Result:

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

(iv)

PROJ_EMPS(PNO,SSN) <-- $\pi_{pno,\ Essn}$ (WORKS_ON)

ALL_PROJS(PNO) <-- $\pi_{PNUMBER}$ (PROJECT)

EMPS_ALL_PROJS <-- PROJ_EMPS -:- ALLPROJS /* DIVISION operation */

RESULT <-- $\pi_{LNAME,FNAME}$ (EMPLOYEE $\bowtie$ EMP_ALL_PROJS) /*natural join on ssn*/

Result (empty):
LNAME FNAME

(v) RESULT(MAX_F_SAL) <-- ← $\Im_{\text{maximum salary}}$ (($\sigma_{\text{SEX = 'F'}}$ EMPLOYEE))

Result:
max_F_SAL
43000