

Day 1:

1:

Q1. a) Create a table 'STUDENT' with following structures:-

<u>Col. Name</u>	<u>Type</u>	<u>Width</u>
ROLL	NUMBER	2
NAME	VARCHAR2	15
EXAMDATE	DATE	

b) Add a primary key constraint on column 'NAME'.

c) Drop the primary key of the table 'STUDENT'.

d) Add a primary key on col. 'ROLL' in table 'STUDENT'.

e) Insert following data:-

<u>ROLL</u>	<u>NAME</u>	<u>EXAMDATE</u>
11	Sourav	01-JUN-10
12	Kamal	01-JUN-10
13	Rahul	01-JUN-10
14	Sovan	01-DEC-10
15	Shyamal	01-DEC-10

a)

```
CREATE TABLE STUDENT (ROLL NUMBER(2), NAME VARCHAR2(15), EXAMDATE DATE);
```

b)

```
ALTER TABLE STUDENT  
ADD CONSTRAINT pk_1 PRIMARY KEY (NAME);
```

c)

```
ALTER TABLE STUDENT  
DROP CONSTRAINT pk_1;
```

d)

```
ALTER TABLE STUDENT  
ADD CONSTRAINT pk_1 PRIMARY KEY (ROLL);
```

e)

```
INSERT INTO STUDENT VALUES (11, 'SOURAV', '01-JUN-10');  
INSERT INTO STUDENT VALUES (12, 'KAMAL', '01-JUN-10');  
INSERT INTO STUDENT VALUES (13, 'RAHUL', '01-JUN-10');  
INSERT INTO STUDENT VALUES (14, 'SOVAN', '01-DEC-10');  
INSERT INTO STUDENT VALUES (15, 'SHYAMAL', '01-DEC-10');
```

```
SELECT * FROM STUDENT;
```

ROLL	NAME	EXAMDATE
11	SOURAV	2010-06-01T00:00:00Z
12	KAMAL	2010-06-01T00:00:00Z
13	RAHUL	2010-06-01T00:00:00Z
14	SOVAN	2010-12-01T00:00:00Z
15	SHYAMAL	2010-12-01T00:00:00Z

2:

Q2. a) Create a table 'MARKS' with following structures:-

Col. Name	Type	Width
ROLL	NUMBER	2
MATH	NUMBER	2
ENG	NUMBER	3

b) Add a foreign key constraint on column 'ROLL' in 'MARKS' table referencing column 'ROLL' in table 'STUDENT' and name the constraint as FK_ROLL.

c) Try to insert following data:-

<11,90,80>

d) Change width of 'MATH' column to 3.

e) Add a check constraint on 'ENG' column so that permissible value for 'ENG' attribute lies between 0 and 50 and name the constraint as CHK_ENG.

f) Try to insert following data:-

<11,90,80>

g) Now insert following data:

ROLL	MATH	ENG
11	90	45
12	45	46
13	70	30
14	90	20
15	45	46

h) Add a new column 'TOTAL' in table 'MARKS'. The data type is number and width is 3.

i) Update column 'TOTAL' in 'MARKS' table with proper data.

a)

```
CREATE TABLE MARKS (ROLL NUMBER(2), MATH NUMBER(2), ENG NUMBER(3));
```

b)

```
ALTER TABLE MARKS
ADD CONSTRAINT FK_ROLL FOREIGN KEY (ROLL)
REFERENCES STUDENT;
```

c)

```
INSERT INTO MARKS VALUES(20, 90, 80);
```

ORA-02291: integrity constraint (USER_4_13F38.FK_ROLL) violated - parent key not found

d)

e)

h)

ALTER TABLE MARKS
ADD TOTAL NUMBER(3);

i)

UPDATE MARKS
SET TOTAL = MATH + ENG;

DAY2:

3:

Write queries using SQL.

- a) Display data from table 'STUDENT' with column heading ROLL_NO, STD_NAME.
- b) List students having name starting with letter 'S'.
- c) List students where second character of name is 'a'.
- d) Display EXAMDATE in 'DD/MM/YYYY' format.
- e) Display NAME, MATH, ENG and PER of all students. Assume, total marks of math are 100 and eng is 50.
- f) Display names of all students who are getting above 65 of math.
- g) Display names of students getting marks in eng between 20 and 40.
- h) Display name of the student, who get the same marks (math) as that of 'Shyamal'.

a)

SELECT ROLL "ROLL_NO", NAME "STD_NAME" FROM STUDENT;

ROLL_NO	STD_NAME
11	SOURAV
12	KAMAL
13	RAHUL
14	SOVAN
15	SHYAMAL

b)

```
SELECT * FROM STUDENT  
WHERE NAME LIKE 'S%';
```

ROLL	NAME	EXAMDATE
11	SOURAV	2010-06-01T00:00:00Z
14	SOVAN	2010-12-01T00:00:00Z
15	SHYAMAL	2010-12-01T00:00:00Z

c)

```
SELECT * FROM STUDENT  
WHERE NAME LIKE '_A%';
```

ROLL	NAME	EXAMDATE
12	KAMAL	2010-06-01T00:00:00Z
13	RAHUL	2010-06-01T00:00:00Z

d)

```
SELECT TO_CHAR(EXAMDATE, 'DD/MM/YYYY') "EXAM_DATE" FROM STUDENT;
```

EXAM_DATE
01/06/2010
01/06/2010
01/06/2010
01/12/2010
01/12/2010

e)

```
SELECT NAME, MATH, ENG, ROUND(((TOTAL/150)*100), 2) "PER"  
FROM STUDENT S, MARKS M
```

WHERE S.ROLL = M.ROLL;

NAME	MATH	ENG	PER
SOURAV	90	45	90
KAMAL	45	46	60.67
RAHUL	70	30	66.67
SOVAN	90	20	73.33
SHYAMAL	45	46	60.67

f)

SELECT DISTINCT(NAME) "NAME" FROM STUDENT S, MARKS M
WHERE S.ROLL = M.ROLL
AND MATH > 65;

NAME
RAHUL
SOURAV
SOVAN

g)

SELECT DISTINCT(NAME) "NAME" FROM STUDENT S, MARKS M
WHERE S.ROLL = M.ROLL
AND ENG BETWEEN 20 AND 40;

NAME
RAHUL
SOVAN

h)

```
SELECT DISTINCT(NAME) FROM STUDENT S, MARKS M
WHERE S.ROLL = M.ROLL
AND MATH = (SELECT DISTINCT(MATH) FROM STUDENT S, MARKS M
            WHERE S.ROLL = M.ROLL
            AND S.NAME = 'SHYAMAL')
AND NAME <> 'SHYAMAL';
```

OR

```
SELECT DISTINCT(NAME) FROM STUDENT S, MARKS M
WHERE S.ROLL = M.ROLL
AND MATH IN (SELECT MATH FROM STUDENT S, MARKS M
            WHERE S.ROLL = M.ROLL
            AND S.NAME = 'SHYAMAL')
AND NAME <> 'SHYAMAL';
```

NAME
KAMAL

[4:](#)

Q4 Create following tables:-

HOTEL (HOTEL_NO, NAME, ADDRESS)

ROOM(ROOM_NO, HOTEL_NO, TYPE, PRICE)

BOOKING(HOTEL_NO, GUEST_NO, DATE_FROM, DATE_TO, ROOM_NO)

GUEST(GUEST_NO, NAME, ADDRESS)

Where **HOTEL** contains hotel details and HOTEL_NO is the Primary Key.

ROOM contains room details for each hotel and **(HOTEL_NO,ROOM_NO)** forms the

Primary key. **BOOKING** contains details of the bookings and the Primary Key comprises

(HOTEL_NO, GUEST_NO, DATE_FORM) and **GUEST** contains guest details and

GUEST_NO is the Primary key and mention the Foreign Key constraints.

Day 3:-

```
CREATE TABLE HOTEL (HOTEL_NO NUMBER(2), NAME VARCHAR2(15), ADDRESS
VARCHAR2(30));
```



```
CREATE TABLE ROOM (ROOM_NO NUMBER(2), HOTEL_NO NUMBER(2), TYPE  
VARCHAR2(15), PRICE NUMBER(4));  
CREATE TABLE BOOKING (HOTEL_NO NUMBER(2), GUEST_NO NUMBER(2),  
DATE_FROM DATE, DATE_TO DATE, ROOM_NO NUMBER(2));  
CREATE TABLE GUEST (GUEST_NO NUMBER(2), NAME VARCHAR2(15), ADDRESS  
VARCHAR2(30));
```

```
ALTER TABLE HOTEL  
ADD PRIMARY KEY (HOTEL_NO);
```

```
ALTER TABLE ROOM  
ADD PRIMARY KEY (HOTEL_NO, ROOM_NO);
```

```
ALTER TABLE BOOKING  
ADD PRIMARY KEY (HOTEL_NO, GUEST_NO, DATE_FROM);
```

```
ALTER TABLE GUEST  
ADD PRIMARY KEY (GUEST_NO);
```

```
ALTER TABLE ROOM  
ADD CONSTRAINT FK_R_H  
FOREIGN KEY (HOTEL_NO) REFERENCES HOTEL(HOTEL_NO);
```

```
ALTER TABLE BOOKING  
ADD CONSTRAINT FK_B_R  
FOREIGN KEY (HOTEL_NO, ROOM_NO) REFERENCES ROOM(HOTEL_NO, ROOM_NO);
```

```
ALTER TABLE BOOKING  
ADD CONSTRAINT FK_B_G  
FOREIGN KEY (GUEST_NO) REFERENCES GUEST(GUEST_NO);
```

```
INSERT INTO HOTEL VALUES (10, 'Violet', 'Mumbai');  
INSERT INTO HOTEL VALUES (11, 'Evergarden', 'Kolkata');  
INSERT INTO HOTEL VALUES (12, 'Manor', 'New Delhi');  
INSERT INTO HOTEL VALUES (13, 'Moana', 'Kashmir');  
INSERT INTO HOTEL VALUES (14, 'Land Mark', 'Goa');
```

```
INSERT INTO ROOM VALUES (21, 10, 'Single', 500);  
INSERT INTO ROOM VALUES (22, 10, 'Family', 1000);
```

```
INSERT INTO ROOM VALUES (23, 10, 'Double', 800);
INSERT INTO ROOM VALUES (24, 11, 'Family', 900);
INSERT INTO ROOM VALUES (25, 11, 'Single', 400);
INSERT INTO ROOM VALUES (26, 11, 'Family', 900);
INSERT INTO ROOM VALUES (27, 12, 'Single', 500);
INSERT INTO ROOM VALUES (28, 12, 'Family', 1000);
INSERT INTO ROOM VALUES (29, 12, 'Double', 750);
INSERT INTO ROOM VALUES (31, 13, 'Family', 765);
INSERT INTO ROOM VALUES (32, 13, 'Single', 400);
INSERT INTO ROOM VALUES (33, 13, 'Double', 655);
INSERT INTO ROOM VALUES (34, 14, 'Family', 1500);
INSERT INTO ROOM VALUES (35, 14, 'Single', 600);
INSERT INTO ROOM VALUES (36, 14, 'Double', 900);
```

```
INSERT INTO GUEST VALUES (99, 'Ramesh', 'Kolkata');
INSERT INTO GUEST VALUES (98, 'Shuresh', 'New Delhi');
INSERT INTO GUEST VALUES (97, 'Akash', 'Punjab');
INSERT INTO GUEST VALUES (96, 'Karan', 'New Delhi');
INSERT INTO GUEST VALUES (95, 'Vivek', 'Kashmir');
```

```
INSERT INTO BOOKING VALUES (10, 99, '01-MAR-2015', '31-MAR-2015', 21);
INSERT INTO BOOKING VALUES (10, 99, '01-APR-2015', '30-APR-2015', 22);
INSERT INTO BOOKING VALUES (11, 97, '01-MAY-2015', '31-MAY-2015', 24);
INSERT INTO BOOKING VALUES (11, 97, '01-MAR-2015', '31-MAR-2015', 25);
INSERT INTO BOOKING VALUES (12, 96, '01-FEB-2015', '31-MAR-2015', 27);
INSERT INTO BOOKING VALUES (12, 96, '02-FEB-2015', '31-MAR-2015', 29);
INSERT INTO BOOKING VALUES (13, 98, '01-AUG-2015', '31-AUG-2015', 31);
INSERT INTO BOOKING VALUES (13, 98, '01-OCT-2015', '', 32);
INSERT INTO BOOKING VALUES (14, 95, '01-DEC-2015', '31-DEC-2015', 34);
INSERT INTO BOOKING VALUES (14, 95, '01-NOV-2015', '', 36);
```

```
SELECT * FROM HOTEL;
```

HOTEL_NO	NAME	ADDRESS
10	Violet	Mumbai
11	Evergarden	Kolkata
12	Manor	New Delhi
13	Moana	Kashmir
14	Land Mark	Goa

SELECT * FROM ROOM;

ROOM_NO	HOTEL_NO	TYPE	PRICE
21	10	Single	500
22	10	Family	1000
23	10	Double	800
24	11	Family	900
25	11	Single	400
26	11	Family	900
27	12	Single	500
28	12	Family	1000
29	12	Double	750
31	13	Family	765
32	13	Single	400
33	13	Double	655
34	14	Family	1500
35	14	Single	600
36	14	Double	900

SELECT * FROM GUEST;

GUEST_NO	NAME	ADDRESS
99	Ramesh	Kolkata
98	Shuresh	New Delhi
97	Akash	Punjab
96	Karan	New Delhi
95	Vivek	Kashmir

SELECT * FROM BOOKING;

HOTEL_NO	GUEST_NO	DATE_FROM	DATE_TO	ROOM_NO
10	99	2015-03-01T00:00:00Z	2015-03-31T00:00:00Z	21
10	99	2015-04-01T00:00:00Z	2015-04-30T00:00:00Z	22
11	97	2015-05-01T00:00:00Z	2015-05-31T00:00:00Z	24
11	97	2015-03-01T00:00:00Z	2015-03-31T00:00:00Z	25
12	96	2015-02-01T00:00:00Z	2015-03-31T00:00:00Z	27
12	96	2015-02-02T00:00:00Z	2015-03-31T00:00:00Z	29
13	98	2015-08-01T00:00:00Z	2015-08-31T00:00:00Z	31
13	98	2015-10-01T00:00:00Z	(null)	32
14	95	2015-12-01T00:00:00Z	2015-12-31T00:00:00Z	34
14	95	2015-11-01T00:00:00Z	(null)	36

Day 3:

5:

Q5. Write queries using SQL.

- i. List full details of hotels in Mumbai
- ii. List the name and addresses of all guests in New Delhi, alphabetically ordered by the name.
- iii. List all double or family rooms with a price below Rs. 800 per day, in ascending ordered.
- iv. List the bookings for which no date_to has been specified.
- v. What is the total daily revenue from all the double room?
- vi. How many different guests have made booking for august, 2015
- vii. List the price and type of all rooms at the hotel Land Mark.
- viii. What is the total income from booking for the hotel Manor today.

i)

```

SELECT H.NAME, TYPE, PRICE, DATE_FROM, DATE_TO, H.ADDRESS, G.NAME,
G.ADDRESS
FROM HOTEL H, ROOM R, BOOKING B, GUEST G
WHERE H.HOTEL_NO = R.HOTEL_NO
AND (R.ROOM_NO = B.ROOM_NO (+)
AND R.HOTEL_NO = B.HOTEL_NO (+)
AND B.GUEST_NO = G.GUEST_NO (+))
AND H.ADDRESS = 'Mumbai';

```

NAME	TYPE	PRICE	DATE_FROM	DATE_TO	ADDRESS	NAME	ADDRESS
Violet	Single	500	2015-03-01T00:00:00Z	2015-03-31T00:00:00Z	Mumbai	Ramesh	Kolkata
Violet	Family	1000	2015-04-01T00:00:00Z	2015-04-30T00:00:00Z	Mumbai	Ramesh	Kolkata
Violet	Double	800	(null)	(null)	Mumbai	(null)	(null)

ii)

```
SELECT G.NAME, H.ADDRESS FROM HOTEL H, GUEST G, BOOKING B
WHERE G.ADDRESS = 'New Delhi'
AND (B.GUEST_NO = G.GUEST_NO (+)
AND B.HOTEL_NO = H.HOTEL_NO (+))
ORDER BY G.NAME;
```

NAME	ADDRESS
Karan	New Delhi
Karan	New Delhi
Shuresh	Kashmir
Shuresh	Kashmir

iii)

```
SELECT * FROM ROOM
WHERE (TYPE = 'Double' OR TYPE = 'Family')
AND PRICE < 800;
```

ROOM_NO	HOTEL_NO	TYPE	PRICE
29	12	Double	750
31	13	Family	765
33	13	Double	655

iv)

```
SELECT * FROM BOOKING  
WHERE DATE_TO IS NULL;
```

HOTEL_NO	GUEST_NO	DATE_FROM	DATE_TO	ROOM_NO
13	98	2015-10-01T00:00:00Z	(null)	32
14	95	2015-11-01T00:00:00Z	(null)	36

v)

```
SELECT SUM(PRICE) FROM ROOM  
WHERE TYPE = 'Double';
```

SUM(PRICE)
3105

vi)

```
SELECT COUNT(DISTINCT GUEST_NO) "August Booking Count" FROM BOOKING  
WHERE TO_CHAR(DATE_FROM,'MM')='08' AND  
TO_CHAR(DATE_FROM,'YYYY')='2015';
```

August Booking Count
1

vii)

```
SELECT PRICE, TYPE FROM ROOM R, HOTEL H  
WHERE H.NAME = 'Land Mark'  
AND H.HOTEL_NO = R.HOTEL_NO;
```

PRICE	TYPE
1500	Family
600	Single
900	Double

viii)

```
SELECT SUM(PRICE) "Manor Today's Price" FROM ROOM R, BOOKING B, HOTEL H
WHERE H.NAME = 'Manor'
AND H.HOTEL_NO = R.HOTEL_NO
AND B.ROOM_NO (+) = R.ROOM_NO
AND TO_CHAR(CURRENT_DATE, 'DD-MM-YY') BETWEEN TO_CHAR(DATE_FROM,
'DD-MM-YY') AND TO_CHAR(DATE_TO, 'DD-MM-YY');
```

Manor Today's Price
1250
✓ Record Count: 1; Execution Time: 3ms + View Execution Plan ➔ link

Day 4:

1:

- Q1.**
- Create tables for following functional Dependencies –
 $eno \rightarrow \{ename, address\}$
 $pno \rightarrow \{pname, plocation\}$
 $\{eno, pno\} \rightarrow hours$

plocation must be among MUMBAI, KOLKATA, CHENNAI, and DELHI
 - Mention primary key, foreign key and CHECK constraints.
 - i. Insert following data for EMP:-

<u>ENO</u>	<u>ENAME</u>	<u>ADDRESS</u>
1	Swarnali	MUMBAI
2	Deboshree	MUMBAI
3	Moumita	KOLKATA
4	Piyali	CHENNAI
5	Surupa	DELHI

- ii. Insert following data for Proj:-

<u>PNO</u>	<u>PNAME</u>	<u>PLOCATION</u>
101	BANKING	DELHI
102	LIBRARY	MUMBAI
103	RAILWAY_BOOKING	KOLKATA
104	PF_AUTOMATION	CHENNAI
105	SHARE_ANALYZER	DELHI

- iii. Insert following data for EmpProj:-

<u>ENO</u>	<u>PNO</u>	<u>HOURS</u>
1	101	10
2	103	12
3	104	19
3	105	29
5	102	6

- List the name of employees who are working on more than one project.
- Q2. a)** Create a table PHONE_BOOK. The fields of the table are NAME, PHONE, ADDRESS.
- Insert at least 6 entries into the table.

a) and b):


```
CREATE TABLE EMP_15 (ENO NUMBER(2), ENAME VARCHAR2(15), ADDRESS  
VARCHAR2(15));  
CREATE TABLE PROJ_15 (PNO NUMBER(3), PNAME VARCHAR2(15), PLOCATION  
VARCHAR2(15));  
CREATE TABLE EMP_PROJ_15 (ENO NUMBER(1), PNO NUMBER(3), HOURS  
NUMBER(2));
```

```
ALTER TABLE EMP_15  
ADD PRIMARY KEY (ENO);
```

```
ALTER TABLE PROJ_15  
ADD PRIMARY KEY (PNO)  
ADD CONSTRAINT check_plocation  
CHECK (PLOCATION IN ('MUMBAI', 'KOLKATA', 'CHENNAI', 'DELHI'));
```

```
ALTER TABLE EMP_PROJ_15  
ADD CONSTRAINT fk_Proj FOREIGN KEY (PNO) REFERENCES PROJ_15(PNO);
```

```
ALTER TABLE EMP_PROJ_15  
ADD CONSTRAINT fk_Emp FOREIGN KEY (ENO) REFERENCES EMP_15(ENO);
```

c)

i)

```
INSERT INTO EMP_15 VALUES (1, 'Swarnali', 'MUMBAI');  
INSERT INTO EMP_15 VALUES (2, 'Deboshree', 'MUMBAI');  
INSERT INTO EMP_15 VALUES (3, 'Moumita', 'KOLKATA');  
INSERT INTO EMP_15 VALUES (4, 'Piyali', 'CHENNAI');  
INSERT INTO EMP_15 VALUES (5, 'Surupa', 'DELHI');
```

ii)

```
INSERT INTO PROJ_15 VALUES (101, 'BANKING', 'DELHI');  
INSERT INTO PROJ_15 VALUES (102, 'LIBRARY', 'MUMBAI');  
INSERT INTO PROJ_15 VALUES (103, 'RAILWAY_BOOKING', 'KOLKATA');  
INSERT INTO PROJ_15 VALUES (104, 'PF_AUTOMATION', 'CHENNAI');  
INSERT INTO PROJ_15 VALUES (105, 'SHARE_ANALYZER', 'DELHI');
```

iii)

```
INSERT INTO EMP_PROJ_15 VALUES (1, 101, 10);  
INSERT INTO EMP_PROJ_15 VALUES (2, 103, 12);
```

```
INSERT INTO EMP_PROJ_15 VALUES (3, 104, 19);
INSERT INTO EMP_PROJ_15 VALUES (3, 105, 29);
INSERT INTO EMP_PROJ_15 VALUES (5, 102, 6);
```

```
SELECT * FROM EMP_15;
```

ENO	ENAME	ADDRESS
1	Swarnali	MUMBAI
2	Deboshree	MUMBAI
3	Moumita	KOLKATA
4	Piyali	CHENNAI
5	Surupa	DELHI

```
SELECT * FROM PROJ_15;
```

PNO	PNAME	PLOCATION
101	BANKING	DELHI
102	LIBRARY	MUMBAI
103	RAILWAY_BOOKING	KOLKATA
104	PF_AUTOMATION	CHENNAI
105	SHARE_ANALYZER	DELHI

```
SELECT * FROM EMP_PROJ_15;
```

ENO	PNO	HOURS
1	101	10
2	103	12
3	104	19
3	105	29
5	102	6

d)

```
SELECT ENAME FROM EMP_15
WHERE ENO IN (SELECT ENO FROM EMP_PROJ_15
              GROUP BY ENO
              HAVING COUNT(*) > 1);
```

ENAME
Moumita

2:

- ... who are working on more than one project.
- Q2. a)** Create a table PHONE_BOOK. The fields of the table are NAME, ADDRESS, PHONE_NO.
b) Insert at least 6 entries into the table of which there are two pairs of duplicate entries.
c) Delete duplicate rows from the table.
d) Write a query to select first two rows from the table.
e) Write a query to select last two rows from the table.
f) Truncate the table PHONE_BOOK.

a)

```
CREATE TABLE PHONE_BOOK_15 (NAME VARCHAR2(15), ADDRESS VARCHAR2(15),
PHONE_NO NUMBER(10));
```

b)

```
INSERT INTO PHONE_BOOK_15 VALUES('Ramesh', 'Delhi', 8926174521);
INSERT INTO PHONE_BOOK_15 VALUES('Karan', 'Kolkata', 9264721856);
```

```

INSERT INTO PHONE_BOOK_15 VALUES('Suresh', 'Mumbai', 7395618462);
INSERT INTO PHONE_BOOK_15 VALUES('Ramesh', 'Delhi', 8926174521);
INSERT INTO PHONE_BOOK_15 VALUES('Tirup', 'Goa', 8452916402);
INSERT INTO PHONE_BOOK_15 VALUES('Suresh', 'Mumbai', 7395618462);

```

```

SELECT * FROM PHONE_BOOK_15;

```

NAME	ADDRESS	PHONE_NO
Ramesh	Delhi	8926174521
Karan	Kolkata	9264721856
Suresh	Mumbai	7395618462
Ramesh	Delhi	8926174521
Tirup	Goa	8452916402
Suresh	Mumbai	7395618462

c)

```

DELETE FROM PHONE_BOOK_15
WHERE ROWID NOT IN
(SELECT MAX(ROWID) FROM PHONE_BOOK_15 GROUP BY NAME);

```

```

SELECT * FROM PHONE_BOOK_15;

```

NAME	ADDRESS	PHONE_NO
Karan	Kolkata	9264721856
Ramesh	Delhi	8926174521
Tirup	Goa	8452916402
Suresh	Mumbai	7395618462

d)

```
SELECT * FROM PHONE_BOOK_15  
WHERE ROWNUM <=2;
```

NAME	ADDRESS	PHONE_NO
Karan	Kolkata	9264721856
Ramesh	Delhi	8926174521

e)

```
SELECT * FROM PHONE_BOOK_15  
MINUS  
SELECT * FROM PHONE_BOOK_15  
WHERE ROWNUM <= (SELECT (COUNT(*) - 2) FROM PHONE_BOOK_15);
```

NAME	ADDRESS	PHONE_NO
Suresh	Mumbai	7395618462
Tirup	Goa	8452916402

DAY 5:
3:

Q3. a) Create a table employee and insert following data into the table.

EMPNO	EMPNAME	MANAGER NO.	SALARY
E1	Amal	--	30000
E2	Bimal	E1	25000
E3	Kamal	E1	20000
E4	Nirmal	E2	15000
E5	Shymal	E2	21000
E6	Parimal	E3	10000

b) Retrieve the names of the employees and the names of their respective managers from the employee table.

c) Retrieve the name of the employee who is earning second maximum salary.

d) Retrieve the name of the employee who is earning nth highest salary.

e) Retrieve the names of employees whose salary is greater than the salary of all the employees whose manager no. is E1.

f) Get the details of all employees whose salary is lesser than the average salary of the employee.

a)

```
CREATE TABLE EMPLOYEE (EMPNO VARCHAR2(2), EMPNAME VARCHAR2(10),  
MANAGER_NO VARCHAR2(2), SALARY NUMBER(5));
```

```
INSERT INTO EMPLOYEE VALUES ('E1', 'AMAL', '', 30000);  
INSERT INTO EMPLOYEE VALUES ('E2', 'BIMAL', 'E1', 25000);  
INSERT INTO EMPLOYEE VALUES ('E3', 'KAMAL', 'E1', 20000);  
INSERT INTO EMPLOYEE VALUES ('E4', 'NIRMAL', 'E2', 15000);  
INSERT INTO EMPLOYEE VALUES ('E5', 'SHYMAL', 'E2', 21000);  
INSERT INTO EMPLOYEE VALUES ('E6', 'PARIMAL', 'E3', 10000);
```

```
SQL> SELECT * FROM EMPLOYEE;  
  
EM EMPNAME      MA      SALARY  
--  -
```

E1	AMAL		30000
E2	BIMAL	E1	25000
E3	KAMAL	E1	20000
E4	NIRMAL	E2	15000
E5	SHYMAL	E2	21000
E6	PARIMAL	E3	10000

b)

```
SQL> SELECT E.EMPNAME "EMPLOYEE", M.EMPNAME "MANAGER"
2 FROM EMPLOYEE E, EMPLOYEE M
3 WHERE E.MANAGER_NO = M.EMPNO;
```

EMPLOYEE	MANAGER
KAMAL	AMAL
BIMAL	AMAL
SHYMAL	BIMAL
NIRMAL	BIMAL
PARIMAL	KAMAL

c)

```
SQL> SELECT EMPNAME FROM EMPLOYEE E
2 WHERE (SELECT COUNT(SALARY) FROM EMPLOYEE WHERE SALARY >= E.SALARY) = 2;
```

EMPNAME
BIMAL

d)

```
SQL> SELECT EMPNAME FROM EMPLOYEE E
2 WHERE (SELECT COUNT(SALARY) FROM EMPLOYEE WHERE SALARY >= E.SALARY) = &n;
Enter value for n: 3
old 2: WHERE (SELECT COUNT(SALARY) FROM EMPLOYEE WHERE SALARY >= E.SALARY) = &n
new 2: WHERE (SELECT COUNT(SALARY) FROM EMPLOYEE WHERE SALARY >= E.SALARY) = 3
```

EMPNAME
SHYMAL

e)

```
SQL> SELECT EMPNAME FROM EMPLOYEE E
2 WHERE SALARY > ALL (SELECT SALARY FROM EMPLOYEE
3 WHERE MANAGER_NO = 'E1');
```

EMPNAME
AMAL

f)

```
SQL> SELECT * FROM EMPLOYEE
2 WHERE SALARY < (SELECT AVG(SALARY) FROM EMPLOYEE);
```

EM	EMPNAME	MA	SALARY
E3	KAMAL	E1	20000
E4	NIRMAL	E2	15000
E6	PARIMAL	E3	10000

4:

Q4. a) Create a table account and insert following data into the table Account.

ACCOUNT NO.	BRANCH NAME	AMOUNT
A1	Kolkata	50000
A2	Howrah	40000
A3	Howrah	40000
A4	Kolkata	20000
A5	Durgapur	30000

b) Create a view that will show branch name and total amount of that branch. The name of view will be acc_view.

c) Select the branch names having total amount greater than 50,000

i) Using account1 view

ii) Without using view.

a)


```
1 CREATE TABLE ACCOUNT (ACCOUNT_NO VARCHAR2(2), BRANCH_NAME VARCHAR2(10), AMOUNT NUMBER (5));
```

✓

```
1 INSERT INTO ACCOUNT VALUES('A1', 'KOLKATA', 50000);
```

1 rows affected

```
1 INSERT INTO ACCOUNT VALUES('A2', 'HOWRAH', 40000);
```

1 rows affected

```
1 INSERT INTO ACCOUNT VALUES('A3', 'HOWRAH', 40000);
```

1 rows affected

```
1 INSERT INTO ACCOUNT VALUES('A4', 'KOLKATA', 20000);
```

1 rows affected

```
1 INSERT INTO ACCOUNT VALUES('A5', 'DURGAPUR', 30000);
```

1 rows affected

```
1 SELECT * FROM ACCOUNT;
```

ACCOUNT_NO	BRANCH_NAME	AMOUNT
A1	KOLKATA	50000
A2	HOWRAH	40000
A3	HOWRAH	40000
A4	KOLKATA	20000
A5	DURGAPUR	30000

5 rows (showing 1 to 5)

b)

```
1 CREATE VIEW ACC_VIEW AS
2 SELECT BRANCH_NAME, SUM(AMOUNT) "TOTAL_AMOUNT" FROM ACCOUNT
3 GROUP BY BRANCH_NAME;
```

✓

```
1 SELECT * FROM ACC_VIEW;
```

BRANCH_NAME	TOTAL_AMOUNT
DURGAPUR	30000
KOLKATA	70000
HOWRAH	80000

3 rows (showing 1 to 3)

c)

```
1 SELECT BRANCH_NAME FROM ACC_VIEW
2 WHERE TOTAL_AMOUNT > 50000;
```

BRANCH_NAME
KOLKATA
HOWRAH

2 rows (showing 1 to 2)

```
1 SELECT BRANCH_NAME FROM ACCOUNT
2 GROUP BY BRANCH_NAME
3 HAVING SUM(AMOUNT) > 50000;
```

BRANCH_NAME
KOLKATA
HOWRAH

2 rows (showing 1 to 2)

5:

Q5. a) Create a table Marks and insert following data into the table.

STUDENT NAME	SUBJECT NAME	MARKS
Amit	DBMS	80
Amit	OS	70
Bimal	DBMS	70
Bimal	OS	70

b) Retrieve the name of the students who are getting marks in DBMS above 75 but who are getting marks in OS less than 75.

c) Write a query to retrieve student names from the marks table and output will look like:-

Mr. A
Mr. A
Mr. B
Mr. B

a)

```
1 CREATE TABLE MARKS (STUDENT_NAME VARCHAR2(10), SUBJECT_NAME VARCHAR2(10), MARKS NUMBER(3));
```

✓

```
1 INSERT INTO MARKS VALUES('AMIT', 'DBMS', 80);
```

1 rows affected

```
1 INSERT INTO MARKS VALUES('AMIT', 'OS', 70);
```

1 rows affected

```
1 INSERT INTO MARKS VALUES('BIMAL', 'DBMS', 70);
```

1 rows affected

```
1 INSERT INTO MARKS VALUES('BIMAL', 'OS', 70);
```

1 rows affected

```
1 SELECT * FROM MARKS;
```

STUDENT_NAME	SUBJECT_NAME	MARKS
AMIT	DBMS	80
AMIT	OS	70
BIMAL	DBMS	70
BIMAL	OS	70

4 rows (showing 1 to 4)

b)

```
1 SELECT STUDENT_NAME FROM MARKS
2 WHERE SUBJECT_NAME = 'DBMS' AND MARKS > 75
3 INTERSECT
4 SELECT STUDENT_NAME FROM MARKS
5 WHERE SUBJECT_NAME = 'OS' AND MARKS < 75;
```

STUDENT_NAME
AMIT

1 rows (showing 1 to 1)

c)

```
1 SELECT 'Mr. ' || SUBSTR(STUDENT_NAME, 1, 1) AS NAME FROM MARKS;
```

NAME
Mr. A
Mr. A
Mr. B
Mr. B

4 rows (showing 1 to 4)