

Assignment : Demonstration of creating database and table

♦ CREATE TABLE :

```
SQL> CREATE TABLE STUD1  
(RNO NUMBER(3),  
FNAME VARCHAR(10),  
LNAME VARCHAR(10),  
MOBILE NUMBER(10),  
CITY VARCHAR(12),  
COURSE VARCHAR(10));
```

Table created.

```
SQL> DESC STUD1;
```

Name	Null?	Type
RNO		NUMBER(3)
FNAME		VARCHAR2(10)
LNAME		VARCHAR2(10)
MOBILE		NUMBER(10)
CITY		VARCHAR2(12)
COURSE		VARCHAR2(10)

♦ Insert Values / Records in Table -

```
SQL> INSERT INTO STUD1 VALUES (01, 'OM', 'BHAVSAR', 9977885566, 'NASHIK', 'BCA');
```

1 row created.

```
SQL> INSERT INTO STUD1 (RNO, FNAME, LNAME, MOBILE, CITY, COURSE)  
VALUES (02, 'SAI', 'PATIL', 9922665588, 'SHIRPUR', 'BBA');
```

1 row created.

```
SQL> SELECT * FROM STUD1;
```

RNO	FNAME	LNAME	MOBILE	CITY	COURSE
1	OM	BHAVSAR	9977885566	NASHIK	BCA
2	SAI	PATIL	9922665588	SHIRPUR	BBA

♦ Use of Truncate Command (All records will be deleted) :

```
SQL> TRUNCATE TABLE STUD1; // Table truncated.
```

♦ Used of Drop Command : (Delete Table) :

```
SQL> DROP TABLE STUDENT; // Table dropped.
```

Assignment : Defining different types of database constraint. UNIQUE, DEFAULT, CHECK and NOT NULL etc.

• UNIQUE, NOT NULL and DEFAULT Constraints –

```
SQL> CREATE TABLE SYBCA  
(RNO NUMBER(3) UNIQUE,  
FNAME VARCHAR(12) NOT NULL,  
LNAME VARCHAR(10),  
MARK NUMBER(3) CHECK(MARK<=100),  
CITY VARCHAR(12) DEFAULT 'SHIRPUR')
```

```
SQL> INSERT INTO SYBCA (RNO, FNAME, LNAME, MARK, CITY)  
VALUES (01, 'OM', 'JOSHI', 80, 'NASHIK');
```

```
SQL> SELECT * FROM SYBCA;
```

RNO	FNAME	LNAME	MARK	CITY
1	OM	BHAVSAR	88	NASHIK
2	SAI	PATIL	65	SHIRPUR

• NOT NULL Constraints –

```
SQL> CREATE TABLE STUD100  
(RNO NUMBER(3),  
FNAME VARCHAR(10) NOT NULL,  
CITY VARCHAR(10));
```

• UNIQUE Constraints –

```
SQL> CREATE TABLE STUD100  
(RNO NUMBER(3) UNIQUE,  
FNAME VARCHAR(10));
```

• CHECK Constraints –

```
SQL> CREATE TABLE EMP  
(EMPID NUMBER(3),  
EMPNAME VARCHAR(10),  
SALARY NUMBER(5) CHECK(SALARY>=1000),  
LOCATION VARCHAR(10));
```

• DEFAULT Constraints -

SQL> CREATE TABLE STUD101

(RNO NUMBER(3)

FNAME VARCHAR(12)

CITY VARCHAR(12) DEFAULT 'SHIRPUR'

SQL> INSERT INTO STUD101 (RNO, FNAME, CITY) VALUES (03, 'SAI', 'NASHIK');

1 row created.

SQL> SELECT * FROM STUD101;

RNO	FNAME	CITY
1	SHIV	MUMBAI
3	SAI	NASHIK

SQL> INSERT INTO STUD101 (RNO, FNAME) VALUES (04, 'SHIV');

1 row created.

SQL> SELECT * FROM STUD101;

RNO	FNAME	CITY
1	SHIV	MUMBAI
3	SAI	NASHIK
4	SHIV	SHIRPUR

Assignment : Demonstrate PRIMARY KEY, FOREIGN KEY Relationship.

♦ PRIMARY KEY -

```
SQL> CREATE TABLE TCS  
2 (ENO NUMBER(3) PRIMARY KEY,  
3 ENAME VARCHAR(10));
```

```
SQL> DESC TCS;
```

Name	Null?	Type
ENO	NOT NULL	NUMBER(3)
ENAME		VARCHAR2(10)

♦ FOREIGN KEY -

```
SQL> CREATE TABLE TCSNEW  
(EMPNO NUMBER(3),  
CITY VARCHAR(10),  
FOREIGN KEY(EMPNO) REFERENCES TCS(ENO));
```

Table created.

```
SQL> INSERT INTO TCS (ENO, ENAME) VALUES (01, 'OM');
```

1 row created.

```
SQL> SELECT * FROM TCS;
```

ENO	ENAME
1	OM
2	SAI
3	RAM

```
SQL> INSERT INTO TCSNEW (EMPNO, CITY) VALUES (01, 'SHIRPUR');
```

1 row created.

```
SQL> SELECT * FROM TCSNEW;
```

EMPNO	CITY
1	SHIRPUR
2	MUMBAI

```
SQL> SELECT * FROM TCS TCSNEW;
```

ENO	ENAME
1	OM
2	SAI
3	RAM

Assignment : Demonstrate to INSERT, UPDATE, and DELETE Records in Table.

♦ Create Table –

```
SQL> CREATE TABLE RCPEMP  
(EMP_ID NUMBER(3),  
ENAME VARCHAR(12),  
LOCATION VARCHAR(12),  
DOJ DATE,  
SALARY NUMBER(5));
```

Table created.

♦ Insert Records / Values into Table –

```
SQL> INSERT INTO RCPEMP (EMP_ID, ENAME, LOCATION, DOJ, SALARY)  
VALUES (101, 'JAYANT', 'PUNE', '07-FEB-2023', 20000);
```

1 row created.

```
SQL> INSERT INTO RCPEMP VALUES(410, 'AAKASH', 'MUMBAI', '25-OCT-2018', 25000);  
SQL> INSERT INTO RCPEMP VALUES(815, 'KRISHNA', 'BANGLORE', '15-MAR-2016', 22000);  
SQL> INSERT INTO RCPEMP VALUES(165, 'KOMAL', 'SHIRPUR', '20-SEP-2019', 24000);
```

♦ Display all records of Table –

```
SQL> SELECT * FROM RCPEMP;
```

EMP_ID	ENAME	LOCATION	DOJ	SALARY
101	JAYANT	PUNE	07-FEB-23	20000
410	AAKASH	MUMBAI	25-OCT-18	25000
815	KRISHNA	BANGLORE	15-MAR-16	22000
165	KOMAL	SHIRPUR	20-SEP-19	24000

♦ Display Column wise Records –

```
SQL> SELECT EMP_ID, ENAME, SALARY FROM RCPEMP;
```

EMP_ID	ENAME	SALARY
101	JAYANT	20000
410	AAKASH	25000
815	KRISHNA	22000
165	KOMAL	24000

• Display all records of Table –

SQL> SELECT * FROM RCPEMP;

EMP_ID	ENAME	LOCATION	DOJ	SALARY
101	JAYANT	PUNE	07-FEB-23	20000
410	AAKASH	MUMBAI	25-OCT-18	25000
815	KRISHNA	BANGLORE	15-MAR-16	22000
165	KOMAL	SHIRPUR	20-SEP-19	24000

• UPDATE RECORDS –

Syntax –

UPDATE table_name

SET field1 = new-value1, field2 = new-value2,

[WHERE CLAUSE]

SQL> UPDATE RCPEMP

SET LOCATION='HYDRABAD'

WHERE EMP_ID=410;

1 row updated.

SQL> SELECT * FROM RCPEMP;

EMP_ID	ENAME	LOCATION	DOJ	SALARY
101	JAYANT	PUNE	07-FEB-23	20000
410	AAKASH	HYDRABAD	25-OCT-18	25000
815	KRISHNA	BANGLORE	15-MAR-16	22000
165	KOMAL	SHIRPUR	20-SEP-19	24000

SQL> UPDATE RCPEMP

SET SALARY=30000

WHERE EMP_ID=101;

SQL> SELECT * FROM RCPEMP;

EMP_ID	ENAME	LOCATION	DOJ	SALARY
101	JAYANT	PUNE	07-FEB-23	30000
410	AAKASH	HYDRABAD	25-OCT-18	25000
815	KRISHNA	BANGLORE	15-MAR-16	22000
165	KOMAL	SHIRPUR	20-SEP-19	24000

SQL> UPDATE RCPEMP

SET DOJ='18-NOV-2019'

WHERE EMP_ID=165;

1 row updated.

SQL> SELECT * FROM RCPEMP;

EMP_ID	ENAME	LOCATION	DOJ	SALARY
101	JAYANT	PUNE	07-FEB-23	30000
410	AAKASH	HYDRABAD	25-OCT-18	25000
815	KRISHNA	BANGLORE	15-MAR-16	22000
165	KOMAL	SHIRPUR	18-NOV-19	24000

SQL> UPDATE RCPEMP
SET SALARY=98000
WHERE ENAME='JAYANT';

1 row updated.

SQL> SELECT * FROM RCPEMP;

EMP_ID	ENAME	LOCATION	DOJ	SALARY
101	JAYANT	PUNE	07-FEB-23	98000
410	AAKASH	HYDRABAD	25-OCT-18	25000
815	KRISHNA	BANGLORE	15-MAR-16	22000
165	KOMAL	SHIRPUR	18-NOV-19	24000

Some Other Examples -

SQL> UPDATE IMRDEMP
SET DESIGNATION='CEO', LOCATION='SINGAPUR'
WHERE EMPID=101;

SQL> UPDATE IMRDEMP
SET ENAME='JAYASHREE', DESIGNATION='IT', DOJ='01-JAN-2021', LOCATION='DUBAI'
WHERE EMPID=410;

SQL> UPDATE IMRDEMP
SET EMPID=110
WHERE ENAME='ROHIT';

♦ Delete All Records from table -

♦ Use of Truncate Command (All records will be deleted) :

SQL> TRUNCATE TABLE RCPEMP; // Table truncated.

Assignment : Demonstrate to Alter Table.

Existing Table -

Name	Type
EMPID	NUMBER(3)
ENAME	VARCHAR2(15)
DESIGNATION	VARCHAR2(15)
DOJ	DATE
LOCATION	VARCHAR2(15)

ALTER COMMAND – Add New Column in an existing table

SYNTAX -

```
ALTER TABLE table_name  
ADD column_name datatype;
```

EXAMPLE -

```
SQL> ALTER TABLE IMRDEMP  
ADD SALARY NUMBER(5);
```

Table altered.

```
SQL> DESC IMRDEMP;
```

Name	Type
EMPID	NUMBER(3)
ENAME	VARCHAR2(15)
DESIGNATION	VARCHAR2(15)
DOJ	DATE
LOCATION	VARCHAR2(15)
SALARY	NUMBER(5)

ALTER COMMAND - DROP COLUMN in an existing table

SYNTAX -

```
ALTER TABLE table_name  
DROP COLUMN column_name;
```

EXAMPLE -

```
SQL> ALTER TABLE IMRDEMP  
DROP COLUMN SALARY;
```

Table altered.

```
SQL> DESC IMRDEMP;
```


Name	Type
EMPID	NUMBER(3)
ENAME	VARCHAR2(15)
DESIGNATION	VARCHAR2(15)
DOJ	DATE
LOCATION	VARCHAR2(15)

ALTER TABLE - RENAME COLUMN

ALTER TABLE *table_name*
 RENAME COLUMN *old_name* to *new_name*;

se

SQL> ALTER TABLE IMRDEMP
 RENAME COLUMN LOCATION TO CITY;

Table altered.

SQL> DESC IMRDEMP;

Name	Null?	Type
EMPID		NUMBER(3)
ENAME	NOT NULL	VARCHAR2(15)
DESIGNATION		VARCHAR2(10)
DOJ		DATE
CITY		VARCHAR2(15)

ALTER COMMAND - CHANGE DATA TYPE

SYNTAX -

ALTER TABLE *table_name*
 MODIFY *column_name* *datatype*;

Example -

SQL> ALTER TABLE IMRDEMP
 MODIFY DESIGNATION VARCHAR(10);

Table altered.

SQL> DESC IMRDEMP;

Name	Type
EMPID	NUMBER(3)
ENAME	VARCHAR2(15)
DESIGNATION	VARCHAR2(10)
DOJ	DATE
LOCATION	VARCHAR2(15)

ALTER COMMAND - Add NOT NULL constraint

SYNTAX -
ALTER TABLE *table_name*
MODIFY *column_name* *datatype* NOT NULL;

Example -
SQL> ALTER TABLE IMRDEMP
MODIFY ENAME VARCHAR(15) NOT NULL;

Table altered.

Name	Null?	Type
EMPID		NUMBER(3)
ENAME	NOT NULL	VARCHAR2(15)
DESIGNATION		VARCHAR2(10)
DOJ		DATE
LOCATION		VARCHAR2(15)

ALTER COMMAND - ADD UNIQUE CONSTRAINT

SYNTAX -
ALTER TABLE *table_name*
MODIFY *column_name* *datatype* UNIQUE;

Example -
SQL> ALTER TABLE IMRDEMP
MODIFY DESIGNATION VARCHAR(10) UNIQUE;

Table altered.

ALTER COMMAND - ADD CHECK CONSTRAINT

SYNTAX -
ALTER TABLE *table_name*
MODIFY *column_name* *datatype* CHECK (CONDITION);

SYNTAX -
SQL> ALTER TABLE IMRDEMP
MODIFY SALARY NUMBER(5) CHECK(SALARY >= 1000);

ALTER COMMAND - ADD PRIMARY KEY

SYNTAX -
ALTER TABLE *table_name*
MODIFY *column_name* *datatype* PRIMARY KEY;

SYNTAX -
SQL> ALTER TABLE IMRDEMP
MODIFY EMPID NUMBER(3) PRIMARY KEY;

Assignment : Write a query to demonstrate WHERE CLAUSE.

Consider this table –

SQL> SELECT * FROM RCPEMP;

EMP_ID	ENAME	LOCATION	DOJ	SALARY
101	JAYANT	PUNE	07-FEB-23	20000
410	AAKASH	MUMBAI	25-OCT-18	25000
815	KRISHNA	BANGLORE	15-MAR-16	22000
165	KOMAL	SHIRPUR	20-SEP-19	24000

◆ Display specific Records –

SQL> SELECT * FROM RCPEMP
WHERE EMP_ID=165;

EMP_ID	ENAME	LOCATION	DOJ	SALARY
165	KOMAL	SHIRPUR	20-SEP-19	24000

SQL> SELECT ENAME, SALARY FROM
RCPEMP WHERE EMP_ID=165;

ENAME	SALARY
KOMAL	24000

SQL> SELECT EMP_ID, ENAME, LOCATION, SALARY FROM RCPEMP
WHERE LOCATION='MUMBAI';

EMP_ID	ENAME	LOCATION	SALARY
410	AAKASH	MUMBAI	25000

SQL> SELECT * FROM RCPEMP
WHERE LOCATION='BANGLORE';

EMP_ID	ENAME	LOCATION	DOJ	SALARY
815	KRISHNA	BANGLORE	15-MAR-16	22000

SQL> SELECT * FROM RCPEMP
WHERE SALARY>20000;

EMP_ID	ENAME	LOCATION	DOJ	SALARY
410	AAKASH	MUMBAI	25-OCT-18	25000
815	KRISHNA	BANGLORE	15-MAR-16	22000
165	KOMAL	SHIRPUR	20-SEP-19	24000

// Display all records having ENAME that start with "K".

```
SQL> SELECT * FROM RCPEMP  
WHERE ENAME LIKE 'K%';
```

EMP_ID	ENAME	LOCATION	DOJ	SALARY
815	KRISHNA	BANGLORE	15-MAR-16	22000
165	KOMAL	SHIRPUR	18-NOV-19	24000

// Name of all Captain Having 'A' as Second Letter

```
SQL> SELECT * FROM RCPEMP  
WHERE CAPTAIN LIKE '_A%';
```

ENAME

JAYANT
AAKASH

// Display location in the name "____UN____".

```
SQL> SELECT * FROM RCPEMP  
2 WHERE LOCATION LIKE '%_UN_%';
```

EMP_ID	ENAME	LOCATION	DOJ	SALARY
101	JAYANT	PUNE	07-FEB-23	20000

// Display records having EMP_ID must between 100 & 200.

```
SQL> SELECT * FROM RCPEMP  
WHERE EMP_ID BETWEEN 100 AND 200;
```

EMP_ID	ENAME	LOCATION	DOJ	SALARY
101	JAYANT	PUNE	07-FEB-23	20000
165	KOMAL	SHIRPUR	20-SEP-19	24000

// Display records having salary less than 25000.

```
SQL> SELECT * FROM RCPEMP  
WHERE SALARY <= 25000;
```

EMP_ID	ENAME	LOCATION	DOJ	SALARY
410	AAKASH	HYDRABAD	25-OCT-18	25000
815	KRISHNA	BANGLORE	15-MAR-16	22000
165	KOMAL	SHIRPUR	18-NOV-19	24000

// Display records having salary greater than 25000.

```
SQL> SELECT * FROM RCPEMP  
WHERE SALARY >= 25000;
```

EMP_ID	ENAME	LOCATION	DOJ	SALARY
101	JAYANT	PUNE	07-FEB-23	98000
410	AAKASH	HYDRABAD	25-OCT-18	25000

// Display records having salary greater than 20000 AND LOCATION is PUNE.

```
SQL> SELECT * FROM RCPEMP  
WHERE CITY='PUNE' AND SALARY>=20000;
```

// Display records having salary greater than 96000 OR LOCATION is PUNE.

```
SQL> SELECT * FROM RCPEMP  
WHERE CITY='PUNE' OR SALARY>=96000;
```

♦ Delete Records from table –

```
SQL> DELETE FROM RCPEMP  
WHERE EMPID=101;
```

1 row deleted.

```
SQL> SELECT * FROM RCPEMP;
```

EMP_ID	ENAME	LOCATION	DOJ	SALARY
410	AAKASH	MUMBAI	25-OCT-18	25000
815	KRISHNA	BANGLORE	15-MAR-16	22000
165	KOMAL	SHIRPUR	20-SEP-19	24000

```
SQL> DELETE FROM RCPEMP  
WHERE ENAME='KOMAL';
```

1 row deleted.

```
SQL> SELECT * FROM RCPEMP;
```

EMP_ID	ENAME	LOCATION	DOJ	SALARY
410	AAKASH	MUMBAI	25-OCT-18	25000
815	KRISHNA	BANGLORE	15-MAR-16	22000

♦ Delete All Records from table –

♦ Use of Truncate Command (All records will be deleted) :

```
SQL> TRUNCATE TABLE RCPEMP; // Table truncated.
```


Assignment : Demonstrate ORDER BY Clause.

ORDER BY –

```
SQL> CREATE TABLE BCA20  
(STUD_ID NUMBER(3),  
FNAME VARCHAR(10),  
DOB DATE,  
CITY VARCHAR(10));
```

Table created.

```
SQL> INSERT INTO BCA20 (STUD_ID, FNAME, DOB, CITY) VALUES (01, 'om', '01-jan-2020', 'shirpur');
```

Syntax –

```
SELECT column-list  
FROM table_name  
[WHERE condition]  
[ORDER BY column1, column2, .. columnN] [ASC | DESC];
```

```
SQL> SELECT * FROM BCA20;
```

STUD_ID	FNAME	DOB	CITY
1	om	01-JAN-20	NAGPUR
2	SAI	02-FEB-19	MUMBAI
3	SHIVAM	03-MAR-18	DELHI

```
SQL> SELECT * FROM BCA20 ORDER BY CITY;
```

(Arrange city in ascending Order)

STUD_ID	FNAME	DOB	CITY
3	SHIVAM	03-MAR-18	DELHI
2	SAI	02-FEB-19	MUMBAI
1	om	01-JAN-20	NAGPUR

```
SQL> SELECT * FROM BCA20 ORDER BY stud_id DESC;
```

(Arrange stud_id in descending order)

STUD_ID	FNAME	DOB	CITY
3	SHIVAM	03-MAR-18	DELHI
2	SAI	02-FEB-19	MUMBAI
1	OM	01-JAN-20	NAGPUR

```
SQL> CREATE TABLE IMRDEMP
(EMPID NUMBER(3),
ENAME VARCHAR(15),
DESIGNATION VARCHAR(15),
DOJ DATE,
LOCATION VARCHAR(15));
```

Table created.

```
SQL> INSERT INTO IMRDEMP VALUES(101, 'ROHIT', 'MANAGER', '01-JAN-2018', 'PUNE');
```

1 row created.

```
SQL> SELECT * FROM IMRDEMP;
```

EMPID	ENAME	DESIGNATION	DOJ	LOCATION
101	ROHIT	MANAGER	01-JAN-18	PUNE
402	KARAN	ENGINEER	07-FEB-22	NAGPUR
201	SNEHA	CA	18-DEC-21	HYDRABAD
410	SAKSHI	HRM	25-OCT-12	BANGLORE

```
SQL> SELECT * FROM IMRDEMP ORDER BY DOJ;
(After Order By / in Descending order)
```

EMPID	ENAME	DESIGNATION	DOJ	LOCATION
110	ROHIT	CEO	01-JAN-18	SINGAPUR
410	JAYASHREE	IT	01-JAN-21	DUBAI
201	SNEHA	CA	18-DEC-21	HYDRABAD
402	KARAN	ENGINEER	07-FEB-22	NAGPUR

```
SQL> SELECT * FROM IMRDEMP ORDER BY LOCATION DESC;
```

EMPID	ENAME	DESIGNATION	DOJ	LOCATION
110	ROHIT	CEO	01-JAN-18	SINGAPUR
402	KARAN	ENGINEER	07-FEB-22	NAGPUR
201	SNEHA	CA	18-DEC-21	HYDRABAD
410	JAYASHREE	IT	01-JAN-21	DUBAI

```
SQL> SELECT * FROM IMRDEMP ORDER BY LOCATION;
(By default Ascending Order)
```

EMPID	ENAME	DESIGNATION	DOJ	LOCATION
410	JAYASHREE	IT	01-JAN-21	DUBAI
201	SNEHA	CA	18-DEC-21	HYDRABAD
402	KARAN	ENGINEER	07-FEB-22	NAGPUR
110	ROHIT	CEO	01-JAN-18	SINGAPUR

Assignment : Demonstrate GROUP BY Clause.

GROUP BY CLAUSE –

```
SQL> INSERT INTO BCAEMP VALUES (4, 'SHIV', 'SHIRPUR', 65000, 101, 'BCA');
SQL> INSERT INTO BCAEMP VALUES (5, 'DEV', 'PUNE', 72000, 301, 'BMS');
SQL> INSERT INTO BCAEMP VALUES (6, 'GANESH', 'DHULE', 90000, 101, 'BCA');
```

```
SQL> SELECT * FROM BCAEMP;
```

EMP_ID	ENAME	CITY	SALARY	DEPTID	DEPTNAME
1	OM	SHIRPUR	80000	101	BCA
2	SAI	PUNE	60000	201	BBA
3	RAM	NASHIK	70000	301	BMS
4	SHIV	SHIRPUR	65000	101	BCA
5	DEV	PUNE	72000	301	BMS
6	GANESH	DHULE	90000	101	BCA

```
SQL> SELECT COUNT(DEPTNAME) FROM BCAEMP
GROUP BY DEPTNAME;
```

```
COUNT(DEPTNAME)
```

3
2
1

```
SQL> SELECT DEPTNAME, COUNT(*) FROM BCAEMP
GROUP BY DEPTNAME;
```

DEPTNAME	COUNT(*)
BCA	3
BMS	2
BBA	1

```
SQL> SELECT DEPTNAME, MAX(SALARY) FROM BCAEMP
GROUP BY DEPTNAME;
```

DEPTNAME	MAX(SALARY)
BCA	90000
BMS	72000
BBA	60000

SQL> SELECT DEPTNAME, MIN(SALARY) FROM BCAEMP
GROUP BY DEPTNAME;

DEPTNAME	MIN(SALARY)
BCA	65000
BMS	70000
BBA	60000

SQL> SELECT DEPTNAME, SUM(SALARY) FROM BCAEMP
GROUP BY DEPTNAME;

DEPTNAME	SUM(SALARY)
BCA	235000
BMS	142000
BBA	60000

SQL> SELECT DEPTNAME, SUM(SALARY), AVG(SALARY) FROM BCAEMP
GROUP BY DEPTNAME;

DEPTNAME	SUM(SALARY)	AVG(SALARY)
BCA	235000	78333.3333
BMS	142000	71000
BBA	60000	60000

SQL> SELECT DEPTNAME, COUNT(*) FROM BCAEMP
WHERE SALARY >= 65000
GROUP BY DEPTNAME;

DEPTNAME	COUNT(*)
BCA	3
BMS	2

SQL> SELECT DEPTNAME, COUNT(SALARY) FROM BCAEMP
WHERE SALARY >= 65000
GROUP BY DEPTNAME;

DEPTNAME	COUNT(SALARY)
BCA	3
BMS	2

Assignment : Demonstrate HAVING Clause.

SQL> select * from India;

ID	NAME	CITY	STATE
1	Ram	Shirpur	Maharashtra
2	Rina	Surat	Gujrat
3	Gita	Panji	Goa
4	Noor	Amritsar	Punjab
5	Sham	Dhule	Maharashtra
6	Aabha	Rajkot	Gujrat
7	Milkha	Patiala	Punjab
8	Tina	Jalgaon	Maharashtra
9	Disha	Ahmedabad	Gujrat
10	Arya	Nashik	Maharashtra

10 rows selected.

SQL> SELECT COUNT(ID), STATE FROM INDIA
GROUP BY STATE
HAVING COUNT(ID)>3;

COUNT(ID)	STATE
4	Maharashtra

SQL> SELECT COUNT(ID), STATE
FROM INDIA
GROUP BY STATE
HAVING COUNT(ID)=1;

COUNT(ID)	STATE
1	Goa

SQL> select count(ID), State
from India
group by State
having count(ID)<=3;

COUNT(ID)	STATE
1	Goa
2	Punjab
3	Gujrat

Assignment : Demonstrate Aggregate Function.

AGGREGATE FUNCTIONS –

```
SQL> CREATE TABLE RCSTAFF  
(EID NUMBER(3),  
ENAME VARCHAR(10),  
DESIGNATION VARCHAR(6),  
GSALARY NUMBER(5),  
PF NUMBER(4),  
ALLOWANCES NUMBER(4));
```

Table created.

```
SQL> DESC RCSTAFF;
```

Name	Null?	Type
EID		NUMBER(3)
ENAME		VARCHAR2(10)
DESIGNATION		VARCHAR2(6)
GSALARY		NUMBER(5)
PF		NUMBER(4)
ALLOWANCES		NUMBER(4)

```
SQL> INSERT INTO RCSTAFF VALUES(01, 'OM', 'CEO', 50000, 5000, 8000);
```

```
SQL> INSERT INTO RCSTAFF VALUES(02, 'SAI', 'MD', 60000, 6000, 9000);
```

```
SQL> INSERT INTO RCSTAFF VALUES(03, 'SHIV', 'RM', 40000, 4000, 6000);
```

```
SQL> INSERT INTO RCSTAFF VALUES(04, 'ROCKY', 'CLERK', 10000, 1000, 2000);
```

```
SQL> SELECT * FROM RCSTAFF;
```

EID	ENAME	DESIGN	GSALARY	PF	ALLOWANCES
1	OM	CEO	50000	5000	8000
2	SAI	MD	60000	6000	9000
3	SHIV	RM	40000	4000	6000
4	ROCKY	CLERK	10000	1000	2000

AVERAGE -

```
SQL> SELECT AVG(GSALARY) FROM RCSTAFF;
```

```
AVG(GSALARY)
```

40000

SQL> SELECT AVG(GSALARY) "AVERAGE" FROM RCSTAFF;

AVERAGE

40000

SUM -

SQL> SELECT SUM(PF) FROM RCSTAFF;

SUM(PF)

16000

SQL> SELECT SUM(PF) "TOTAL PF AMOUNT" FROM RCSTAFF;

TOTAL PF AMOUNT

16000

MAXIMUM -

SQL> SELECT MAX(GSALARY) FROM RCSTAFF;

MAX(GSALARY)

60000

MINIMUM -

SQL> SELECT MIN(GSALARY) FROM RCSTAFF;

MIN(GSALARY)

10000

COUNT -

SQL> SELECT COUNT(EID) FROM RCSTAFF;

COUNT(EID)

4

Assignment No. : Demonstrate String Function.

SQL> select ename, length(ename) from rcpstaff;

ENAME	LENGTH(ENAME)
SHIV	4

SQL> select eid, concat(eid, ename) from rcpstaff;

EID	CONCAT(EID,ENAME)
101	101SHIV

SQL> select lower(ename) from rcpstaff;

LOWER(E
Shiv

SQL> select upper(ename) "employee name" from rcpstaff;

employee
SHIV

SQL> select upper(ename) "emp" from rcpstaff;

emp
SHIV

SQL> SELECT TRIM(' RCPIMRD ') FROM DUAL;

TRIM
RCPIMRD

SQL> SELECT ENAME, TRIM(ENAME) FROM RC;

ENAME	TRIM(ENAME)
OM	OM
SAI	SAI
RAM	RAM
SHIV	SHIV
DEEPAK	DEEPAK

```
SQL> SELECT LPAD(ENAME, 7, '*') FROM RCPSTAFF;
```

```
LPAD(EN
```

```
***SHIV
```

```
*****OM
```

```
SQL> SELECT LPAD('ABC', 7, '*') FROM DUAL;
```

```
LPAD
```

```
****ABC
```

```
SQL> SELECT INSTR('HELLO THIS IS IMRD COLLEGE', 'IMRD') FROM DUAL;
```

```
INSTR('HELLOTHISISIMRDCOLLEGE','IMRD')
```

```
15
```

```
SQL> SELECT ASCII('X') FROM DUAL;
```

```
ASCII('X')
```

```
88
```

```
SQL> SELECT ENAME, ASCII(ENAME) FROM RC;
```

ENAME	ASCII(ENAME)
OM	79
SAI	83
RAM	82
SHIV	83
DEEPAK	68

```
SQL> SELECT ENAME, REVERSE(ENAME) FROM RC;
```

ENAME	REVERSE(ENAME)
OM	MO
SAI	IAS
RAM	MAR
SHIV	VIHS
DEEPAK	KAPEED

Assignment - Demonstrate DATE Functions.

SQL> SELECT CURRENT_DATE FROM dual;

CURRENT_D

03-MAR-23

SQL> SELECT MONTHS_BETWEEN(DATE '2017-03-31', DATE '2017-02-28') MONTH_DIFF FROM DUAL;

MONTH_DIFF

1

SQL> SELECT

LAST_DAY(DATE '2000-02-01') LAST_DAY_OF_FEB_2000,
LAST_DAY(DATE '2016-02-01') LAST_DAY_OF_FEB_2016,
LAST_DAY(DATE '2017-02-01') LAST_DAY_OF_FEB_2017
FROM dual;

LAST_DAY_	LAST_DAY_	LAST_DAY_
29-FEB-00	29-FEB-16	28-FEB-17

SQL> SELECT LAST_DAY(SYSDATE) FROM dual;

LAST_DAY(SYSDATE)

28

Assignment - Demonstrate JOINS & NESTED queries.

NESTED query:-

```
SQL> create table custmer  
( custid int primary key,  
  custname varchar(20),  
  address varchar(20) );
```

Table created.

```
SQL> create table order6  
( oid int primary key,  
  onum int,  
  custid int,  
  foreign key(custid) references custmer(custid)  
);
```

Table created.

```
SQL> insert into custmer values(1,'vaishali','shirpur');
```

```
SQL> insert into custmer values(2,'vaishu','dhule');
```

```
SQL> insert into custmer values(3,'ram','dhule');
```

```
SQL> select * from custmer;
```

CUSTID	CUSTNAME	ADDRESS
1	vaishali	shirpur
2	vaishu	dhule
3	ram	dhule

```
SQL> select * from order6;
```

OID	ONUM	CUSTID
1	5	2
2	2	3

```
SQL> select * from custmer  
WHERE custid IN (select custid from order6);
```

CUSTID	CUSTNAME	ADDRESS
2	vaishu	dhule
3	ram	dhule

```
SQL> select * from custmer  
WHERE custid NOT IN (select custid from order6);
```

CUSTID	CUSTNAME	ADDRESS
1	vaishali	shirpur

JOINS queries:-

```
SQL> create table emp8  
(eid int primary key,  
  ename varchar(20) );
```

Table created.

```
SQL> insert into emp8 values(1, 'ram'); 1 row created.  
SQL> insert into emp8 values(2, 'sham'); 1 row created.  
SQL> insert into emp8 values(3, 'mohan'); 1 row created.  
SQL> insert into emp8 values(4, 'sai'); 1 row created.  
SQL> insert into emp8 values(5, 'gita'); 1 row created.
```

```
SQL> select * from emp8;
```

EID	ENAME
1	ram
2	sham
3	mohan
4	sai
5	gita

```
SQL> create table emp9  
( address varchar(20),  
  eid int,  
  foreign key(eid) references emp8(eid));
```

Table created.

```
SQL> insert into emp9 values('shirpur', 1); 1 row created.  
SQL> insert into emp9 values('dhule', 2); 1 row created.  
SQL> insert into emp9 values('pune', 4); 1 row created.
```

```
SQL> select * from emp9;
```

ADDRESS	EID
shirpur	1
dhule	2
pune	4

```
SQL> select emp8.eid, emp8.ename, emp9.address from emp8  
LEFT JOIN emp9 ON emp8.eid=emp9.eid;
```

EID	ENAME	ADDRESS
1	ram	shirpur
2	sham	dhule
4	sai	pune
5	gita	
3	mohan	

SQL> select emp8.eid, emp8.ename, emp9.address from emp8
RIGHT JOIN emp9 ON emp8.eid=emp9.eid;

EID	ENAME	ADDRESS
1	ram	shirpur
2	sham	dhule
4	sai	pune

SQL> select emp8.eid, emp8.ename, emp9.address from emp8
FULL JOIN emp9 ON emp8.eid=emp9.eid;

EID	ENAME	ADDRESS
1	ram	shirpur
2	sham	dhule
3	mohan	
4	sai	pune
5	gita	

SQL> select ename, address from emp8
CROSS JOIN emp9

ENAME	ADDRESS
ram	shirpur
sham	shirpur
mohan	shirpur
sai	shirpur
gita	shirpur
ram	dhule
sham	dhule
mohan	dhule
sai	dhule
gita	dhule
ram	pune

ENAME	ADDRESS
sham	pune
mohan	pune
sai	pune
gita	pune

15 rows selected.