RDBMS ASSIGNMENTS

1. Draw E-R diagram and convert entities and relationships to relation table for a given scenario.
    a. Two assignments shall be carried out i.e., consider two different scenarios (eg. bank, college)
2. Write relational algebra queries for a given set of relations.
3. Perform the following:
    Creating Tables (With and Without Constraints),
Inserting/Updating/Deleting Records in a Table, Saving (Commit) and Undoing (rollback)
4. Perform the following:
    a. Altering a Table, Dropping/Truncating/Renaming Tables, Backing up / Restoring a Database.
5. For a given set of relation schemes, create tables and perform the following
    Simple Queries, Simple Queries with Aggregate functions, Queries with Aggregate functions (group by and having clause),
    Join Queries-Inner join and outer join
6. Queries involving- Date Functions, String Functions , Math Functions
    Subqueries- With IN clause, With EXISTS clause
7. For a given set of relation tables perform the following
    a. Creating Views (with and without check option), Dropping views, Selecting from a view

Assignment 1:

1. COLLEGE DATABASE:

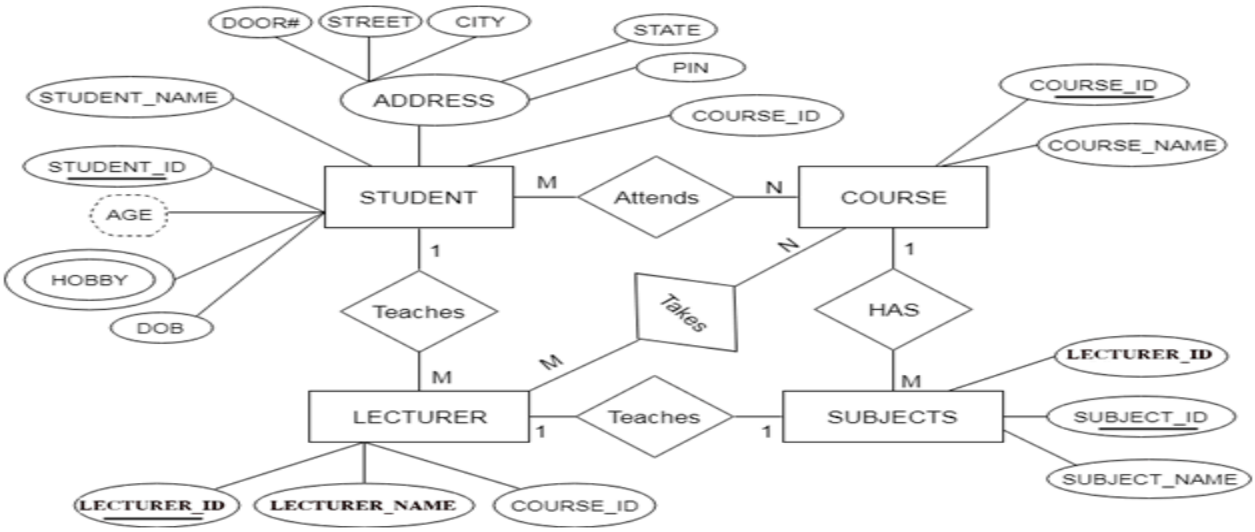
STUDENT (student_id, *student_name, address, Course_id, DOB*)
COURSE (course_id, course_name)
LECTURER (Lecturer_id, Lecturer_name,Course_name)
SUBJECTS (Lecturer_id, Subject_id,Subject_name)

2. COMPANY DATABASE:

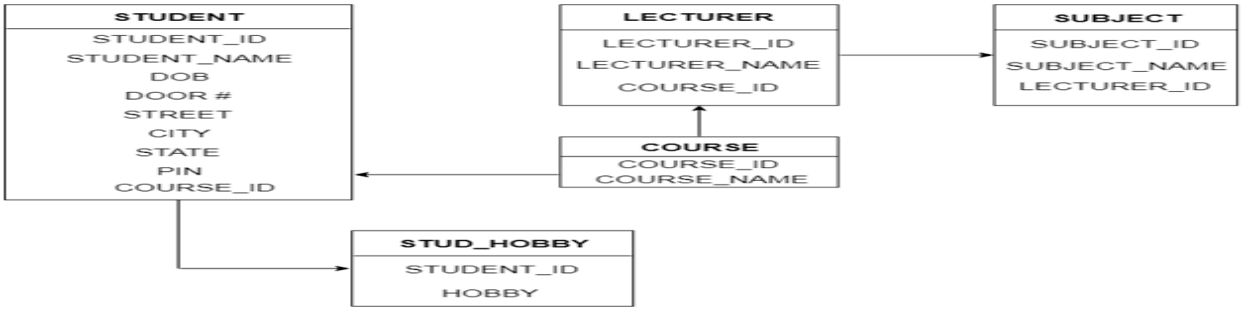
EMPLOYEE (*SSN, Name, Address, Sex, Salary, SuperSSN, DNo*)
DEPARTMENT (*DNo, DName, MgrSSN, MgrStartDate*)
DLOCATION (*DNo,DLoc*)
PROJECT (*PNo, PName, PLocation, DNo*)
WORKS_ON (*SSN, PNo, Hours*)

SOLUTION:

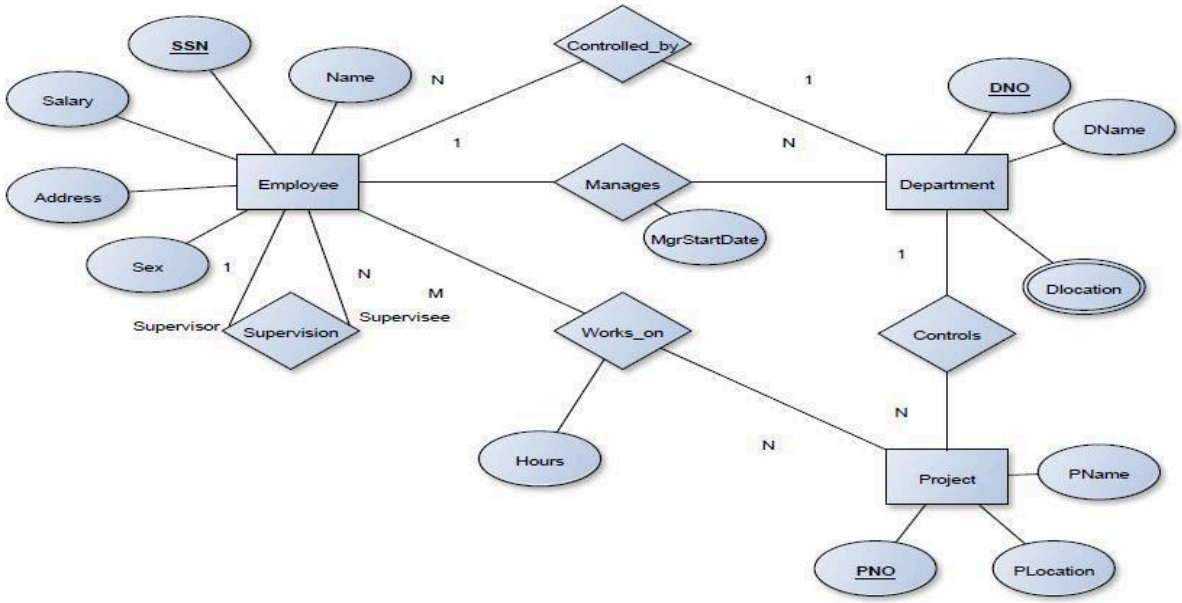
College Database: E-R Diagram



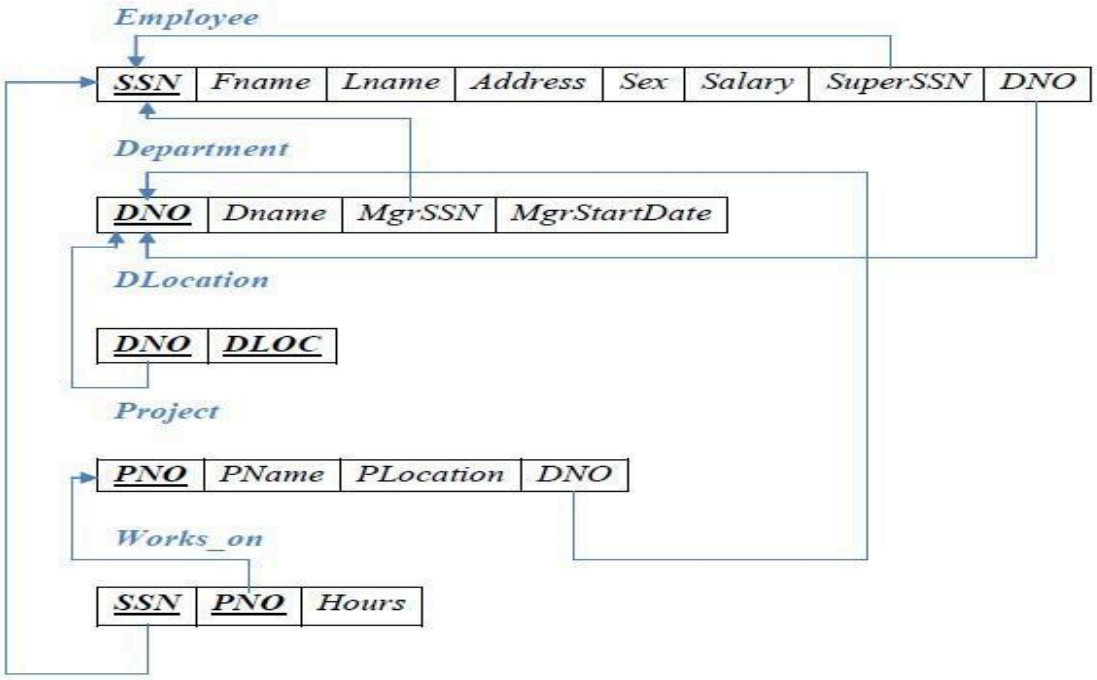
Mapping entities and relationships to relation table
(Schema Diagram)



COMPANY DATABASE E-R Diagram



Schema Diagram



Assignment 2

Write relational algebra queries for a given set of relations.

Student

ID	NAME	MARK
1	Jessy	70
2	Vishnu	75
3	Dwayne	80

. To retrieve entire details from STUDENT

σ (STUDENT)

2.To retrieve details from STUDENT where ID=2 $\sigma_{ID=2}$

(STUDENT)

Π Attribute (Relation)

1. **To retrieve ID and NAME from STUDENT.**

   Π ID, NAME (STUDENT)

Combination of Select and Project: Retrieve ID and Name from STUDENT where mark>=75

Π ID, NAME (σ Mark>=75 (STUDENT))

3. Retrieve students name either participant in arts or sports

ARTS

ID	NAME
1	A
2	B
3	C

SPORTS

ID	NAME
3	C
2	B
4	D

NAME (ARTS) ∪ *NAME (SPORTS)*

ARTS **U** SPORTS

ID	NAME
1	A
2	B
3	C
4	D

Assignment 3

Consider the Company database with following tables

Employee Table

EID	FNAME	LNAME	JOB_NAME	ADDRESS	SALARY	DNO
2201	JOHN	SMITH	PRESIDENT	BANGALORE	250000	1
2202	ANJAN	SHAH	PRESIDENT	CHENNAI	120000	2

2203	AMRIT	WARMA	MANAGER	BANGALORE	140000	3
2204	ANKITA	SHAH	MANAGER	DELHI	120000	4
2205	ASHRITA	SHARMA	ANALYST	DELHI	90000	3
2206	DEEPTI	B	ANALYST	CHENNAI	95000	3
2207	DHIRAJ	M	SALESMAN	CHENNAI	95000	2
2208	MANOJ	M	SALESMAN	BANGALORE	95000	2
2209	KRITIKA	MALHOTRA	AUDITOR	MANGALORE	90000	4
2210	SANJANA	SHARMA	CASHIER	DELHI	60000	1

Department Table

D.No	DName	DLocation
1	FINANCE	BANGALORE
2	MARKETING	BANGALORE
3	IT	BANGALORE
4	AUDIT	MYSORE

Perform the following:

1. Creating Tables (With and Without Constraints)
2. Inserting/Updating/Deleting Records in a Table
3. Saving (Commit) and Undoing (rollback)

SOLUTION:

1. Creating Tables (With and Without Constraints)

```
CREATE TABLE DEPARTMENT (
  DNO VARCHAR2 (20) PRIMARY KEY,
  DNAME VARCHAR2 (20),
  DLOCATION VARCHAR (20)
  );


CREATE TABLE EMPLOYEE (
EID NUMBER (20) PRIMARY KEY,
FNAME VARCHAR (20),
LNAME VARCHAR (20),
JOB_NAME VARCHAR (20),
ADDRESS VARCHAR (20),
SALARY NUMBER NOT NULL,
DNO REFERENCES DEPARTMENT (DNO));
```

2). **Inserting/Updating/Deleting Records in a Table,**

INSERTING RECORDS INTO DEPARTMENT TABLE:

```
SQL> INSERT INTO DEPARTMENT VALUES(1,'FINANCE','BANGALORE');
1 row created.
SQL> INSERT INTO DEPARTMENT VALUES(2,'MARKETING','BANGALORE');
1 row created.
SQL> INSERT INTO DEPARTMENT VALUES(3,'IT','BANGALORE');
1 row created.
SQL> INSERT INTO DEPARTMENT VALUES(4,'AUDIT','MYSORE');
1 row created.
```

***INSERTING RECORDS INTO EMPLOYEE TABLE*:**

```
SQL> INSERT INTO EMPLOYEE
VALUES(2201,'JOHN','SMITH','PRESIDENT','BANGALORE',250000,1);
1 row created.
SQL> INSERT INTO EMPLOYEE VALUES(2202,'ANJAN','SHAH','PRESIDENT','CHENNAI',120000,2);
1 row created.
SQL> INSERT INTO EMPLOYEE
VALUES(2203,'AMRIT','WARMA','MANAGER','BANGALORE',140000,3);
1 row created.
SQL> INSERT INTO EMPLOYEE VALUES(2204,'ANKITA','SHAH','MANAGER','DELHI',120000,4);
1 row created.
SQL> INSERT INTO EMPLOYEE VALUES(2205,'ASHRITA','SHARMA','ANALYST','DELHI',90000,3);
1 row created.
SQL> INSERT INTO EMPLOYEE VALUES (2206,'DEEPTI','B','ANALYST','CHENNAI',95000,3);
1 row created.
SQL> INSERT INTO EMPLOYEE VALUES (2207,'DHIRAJ','M','SALESMAN','CHENNAI',95000,2);
1 row created.
SQL> INSERT INTO EMPLOYEE VALUES (2208,'MANOJ','M','SALESMAN','BANGALORE',95000,2);
1 row created.
SQL> INSERT INTO EMPLOYEE VALUES
(2209,'KRITIKA','MALHOTRA','AUDITOR','MANGALORE',90000,4);
1 row created.
SQL> INSERT INTO EMPLOYEE VALUES
(2210,'SANJANA','SHARMA','CASHIER','MANGALORE',60000,1);
1 row created.
```

```
SQL> select *from department;

DNO             DNAME           DLOCATION
--------------- --------------- ---------------
1               FINANCE         BANGALORE
2               MARKETING       BANGALORE
3               IT                      BANGALORE
4               AUDIT           MYSORE
```

SQL> SELECT * FROM EMPLOYEE;

EID	FNAME	LNAME	JOB_NAME	ADDRESS	SALARY	DNO
2201	JOHN	SMITH	PRESIDENT	BANGALORE	250000	1
2202	ANJAN	SHAH	PRESIDENT	CHENNAI	120000	2
2203	AMRIT	WARMA	MANAGER	BANGALORE	140000	3
2204	ANKITA	SHAH	MANAGER	DELHI	120000	4
2205	ASHRITA	SHARMA	ANALYST	DELHI	90000	3
2206	DEEPTI	B	ANALYST	CHENNAI	95000	3
2207	DHIRAJ	M	SALESMAN	CHENNAI	95000	2
2208	MANOJ	M	SALESMAN	BANGALORE	95000	2
2209	KRITIKA	MALHOTRA	AUDITOR	MANGALORE	90000	4
2210	SANJANA	SHARMA	CASHIER	DELHI	60000	1

Demonstrate the usage of Update and Delete commands for single and multiple records.

UPDATE:

SQL query to update the employee address as 'Delhi' of employee ID '2210' from Employee Table.

```
SQL> UPDATE EMPLOYEE
  SET ADDRESS='DELHI' WHERE EID='2210';
```

DELETE:

SQL query to delete the employee ID '2208' from employee table.

```
SQL> DELETE FROM EMPLOYEE
       WHERE EID='2208';
1 row deleted.

SQL> SELECT EID FROM EMPLOYEE;

EID
--------------------
2201
2202
2203
2204
2205
2206
2207
2209
2210

9 rows selected.
```

.

SQL query to delete the employee who is from Mangalore from Employee Table.

```
SQL> DELETE FROM EMPLOYEE WHERE ADDRESS='MANGALORE';
    1 row deleted.

    SQL> SELECT ADDRESS FROM EMPLOYEE;

    ADDRESS
--------------------
    BANGALORE
    CHENNAI
    BANGALORE
    DELHI
    DELHI
    CHENNAI
    CHENNAI
    DELHI
```

3). COMMIT and ROLLBACK

Before concluding this section on Data Manipulation Language commands there are two further commands, which are very useful. Changes made to the database by INSERT, UPDATE and DELETE commands are temporary until explicitly committed. This is performed by the command:

COMMIT;

```
SQL> commit;
Commit complete.
```

On execution of this command all changes to the database made by you are made permanent and cannot be undone.

- A COMMIT is automatically executed when you exit normally from SQL*Plus. However, it does no harm to occasionally issue a COMMIT command.
- A COMMIT does not apply to any SELECT commands as there is nothing to commit.
- A COMMIT does not apply to any DDL commands (eg CREATE TABLE, CREATE INDEX, etc). These are automatically committed and cannot be rolled back.
- If you wished to rollback (ie undo) any changes made to the database since the last commit, you can issue the command:

ROLLBACK;

A group of related SQL commands that all have to complete successfully or otherwise be rolled back, is called a transaction. Part of your research for Outcome 3 includes investigating transaction processing and the implications of rollback and commit.

```
SQL> rollback;
Rollback complete.
```

Assignment 4

Perform the following:

a. Altering a Table, Dropping/Truncating/Renaming Tables, Backing up / Restoring a Database.

SQL> desc department;

Name	Null?	Type
DNO	NOT NULL	VARCHAR2(20)
DNAME		VARCHAR2(20)
DLOCATION		VARCHAR2(20)

Altering a Table:

1. **Rename The Table Department as Dept**

Solution:

SQL> alter table department rename to dept;

Table altered.

SQL> DESC DEPT;

Name	Null?	Type
DNO	NOT NULL	VARCHAR2(20)
DNAME		VARCHAR2(20)
DLOCATION		VARCHAR2(20)

2. **Add a new column PINCODE to the existing table Dept**

Solution:

SQL> ALTER TABLE DEPT ADD PINCODE NUMBER (6);

Table altered.

SQL> desc dept;

Name	Null?	Type
DNO	NOT NULL	VARCHAR2(20)
DNAME		VARCHAR2(20)
DLOCATION		VARCHAR2(20)
PINCODE		NUMBER (6)

3. **To delete a column (attribute) from Table dept.**

Solution:

```
SQL> ALTER TABLE DEPT DROP COLUMN PINCODE;

Table altered.

SQL> DESC DEPT1;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 DNO                                       NOT NULL VARCHAR2(20)
 DNAME                                              VARCHAR2(20)
 DLOCATION                                          VARCHAR2(20)
```

4. **Rename the column DNAME to DEPT_NAME IN DEPT TABLE**.

Solution:

```
SQL> ALTER TABLE DEPT RENAME COLUMN DNAME TO DEPT_NAME;

 Table altered.

 SQL> desc dept;
  Name                                      Null?    Type
----------------------------------------- -------- ----------------------------
 DNO                                       NOT NULL VARCHAR2(20)
 DEPT_NAME                                          VARCHAR2(20)
 DLOCATION                                          VARCHAR2(20)
 PINCODE                                            NUMBER(6)
```

5. **Change the datatype of column DLOCATION as CHAR with size 10 .**

Solution:

```
SQL> ALTER TABLE DEPT  MODIFY  DLOCATION CHAR(10);

Table altered.


SQL> DESC DEPT;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 DNO                                       NOT NULL VARCHAR2(20)
 DEPT_NAME                                          VARCHAR2(20)
 DLOCATION                                          CHAR (10)
 PINCODE                                            NUMBER (6)
```

6. Delete table

```
SQL> DROP TABLE DEPARTMENT;
Table dropped.
```

7. Truncate table

```
SQL> TRUNCATE TABLE DEPT;
Table truncated.
```

8. To take backup

```
SQL> Create table dept_Bkp as Select  *from dept;
Table created.
```

Assignment 5

Consider the following Employee table

EID	FNAME	LNAME	JOB_NAME	ADDRESS	SALARY	DNO
2201	JOHN	SMITH	PRESIDENT	BANGALORE	250000	1
2202	ANJAN	SHAH	PRESIDENT	CHENNAI	120000	2
2203	AMRIT	WARMA	MANAGER	BANGALORE	140000	3
2204	ANKITA	SHAH	MANAGER	DELHI	120000	4
2205	ASHRITA	SHARMA	ANALYST	DELHI	90000	3

2206	DEEPTI	B	ANALYST	CHENNAI	95000	3

Perform the following:

1. **Display all the fields of employee table**

SQL> select *from employee;

EID	FNAME	LNAME	JOB_NAME	ADDRESS	SALARY	DNO
2201	JOHN	SMITH	PRESIDENT	BANGALORE	250000	1
2202	ANJAN	SHAH	PRESIDENT	CHENNAI	120000	2
2203	AMRIT	WARMA	MANAGER	BANGALORE	140000	3
2204	ANKITA	SHAH	MANAGER	DELHI	120000	4
2205	ASHRITA	SHARMA	ANALYST	DELHI	90000	3
2206	DEEPTI	B	ANALYST	CHENNAI	95000	3

2. **Retrieve employee ID and their salary.**

SQL> select eid, salary from employee;

EID	SALARY
2201	250000
2202	120000
2203	140000
2204	120000
2205	90000
2206	95000

3. **Retrieve fname, lname, job_name from employee**

SQL> select fname, lname, job_name from employee;

FNAME	LNAME	JOB_NAME
JOHN	SMITH	PRESIDENT
ANJAN	SHAH	PRESIDENT
AMRIT	WARMA	MANAGER
ANKITA	SHAH	MANAGER
ASHRITA	SHARMA	ANALYST
DEEPTI	B	ANALYST

4. **Retrieve the fname, salary of an employee who is taking maximum(Highest) salary.**

SQL> select Fname, salary from employee
2 where salary = (select max(salary) from employee);

FNAME	SALARY
JOHN	250000

5. **Demonstrate the usage of BETWEEN, IN, LIKE, AND OR, NOT.**

SQL> select*from employee where salary between 120000 and 250000;

EID	FNAME	LNAME	JOB_NAME	ADDRESS	SALARY	DNO
2201	JOHN	SMITH	PRESIDENT	BANGALORE	250000	1
2202	ANJAN	SHAH	PRESIDENT	CHENNAI	120000	2
2203	AMRIT	WARMA	MANAGER	BANGALORE	140000	3
2204	ANKITA	SHAH	MANAGER	DELHI	120000	4

SQL> select *from employee where salary in (120000,140000);

EID	FNAME	LNAME	JOB_NAME	ADDRESS	SALARY	DNO
2203	AMRIT	WARMA	MANAGER	BANGALORE	140000	3
2204	ANKITA	SHAH	MANAGER	DELHI	120000	4

SQL> select *from employee where not(eid='2204');

EID	FNAME	LNAME	JOB_NAME	ADDRESS	SALARY	DNO
2201	JOHN	SMITH	PRESIDENT	BANGALORE	250000	1
2202	ANJAN	SHAH	PRESIDENT	CHENNAI	120000	2
2203	AMRIT	WARMA	MANAGER	BANGALORE	140000	3
2205	ASHRITA	SHARMA	ANALYST	DELHI	90000	3
2206	DEEPTI	B	ANALYST	CHENNAI	95000	3

SQL> select * from employee where fname like 'a%';

EID	FNAME	LNAME	JOB_NAME	ADDRESS	SALARY	DNO
2202	ANJAN	SHAH	PRESIDENT	CHENNAI	120000	2
2203	AMRIT	WARMA	MANAGER	BANGALORE	140000	3
2205	ASHRITA	SHARMA	ANALYST	DELHI	90000	3

SQL Queries to Demonstrate the usage of Aggregate Functions.

Retrieve average salary of all employee

SQL> select avg(salary) from employee;

```
AVG(SALARY)
---------------
 125833.333
```

Retrieve number of employees

SQL> select count (*) from employee;

```
  COUNT (*)
----------
         6
```

Retrieve the number of rows in salary field

```
SQL> select count(salary) from employee1;

COUNT(SALARY)
-------------
           6
```

Retrieve Maximum Salary from employee

```
SQL> select max(salary) from employee;

MAX(SALARY)
-----------
     250000
```

Retrieve Minimum Salary from employee

```
SQL> select min(salary) from employee;

MIN(SALARY)
-----------
      90000
```

Retrieve Minimum Salary and Minimum salary from employee

```
SQL> select max(salary),min(salary) from employee;

MAX(SALARY)          MIN(SALARY)
----------- -----------------------------------
     250000                 90000
```

To display the sum of salary

```
SQL> SELECT SUM(SALARY) FROM EMPLOYEE;

SUM(SALARY)
-----------
```

815,000

count the number of employees who are taking more than 120000 salary.

SQL> select count(salary) from employee where salary>120000;

COUNT(SALARY)

2

SQL Queries to demonstrate the usage of Group By and Order by

SQL> select *from employee order by fname asc;

EID	FNAME	LNAME	JOB_NAME	ADDRESS	SALARY	DNO
2203	AMRIT	WARMA	MANAGER	BANGALORE	140000	3
2204	ANKITA	SHAH	MANAGER	DELHI	120000	4
2205	ASHRITA	SHARMA	ANALYST	DELHI	90000	3
2206	DEEPTI	B	ANALYST	CHENNAI	95000	3
2201	JOHN	SMITH	PRESIDENT	BANGALORE	250000	1

Display name of employee in descending order

SQL> select fname from employee **order by** fname desc;

FNAME

JOHN
DEEPTI
ASHRITA
ANKITA
ANJAN
AMRIT

6 rows selected.

Retrieve total salary of employee which is greater than >120000

SQL> select fname, sum(salary) from employee
2 **group by** (fname)
3 **having** sum(salary)>120000;

FNAME SUM(SALARY)
--- ---

AMRIT 140000
JOHN 250000

SQL Queries Demonstrate the Join Queries

Write the queries to implement the joins

```
SQL> create table studentnew(
  2  rollno number(25),
  3  name varchar(25),
  4  address varchar(25),
  5  phone number,
  6  age number);
```

Table created.

```
SQL> create table stdcoursenew(
  2  courseid varchar(25),
  3  rollno number);
```

Table created.

```
SQL> insert into studentnew values(1,'arpita','bijapur',9887666,19);
1 row created.
SQL> insert into stdcoursenew values('bsc123',1);
1 row created.
SQL> insert into stdcoursenew values('bsc123',2);
1 row created
SQL> insert into stdcoursenew values('bsc123',3);
1 row created.
SQL> insert into stdcoursenew values('bsc123',4);
1 row created.
SQL> insert into stdcoursenew values('bsc125',5);
1 row created.

SQL> select * from stdcourse ;
COURSEID                  ROLLNO
------------------------- ----------
```

```
BSC123          1
BSC124          2
BSC125          4
BSC127          3
BSC127          5
```

```
SQL> select * from studentnew;

ROLLNO.  NAME      ADDRESS          PHONE.      AGE
---------- ------------------------ ------------------------- ---------
     1.          arpita          bijapur              9887666.     19

     2.          bindu           jamakhandi         78999989.    18

     4           chandrika        rabakavi            8999999.    19
```

```
   ROLLNO.    NAME    ADDRESS    PHONE .   AGE
---------- ------------------------ ------------------------- -----
     3.       deeikas         banahatti      8999989         20
     2.      srusthi          hubli          78999989   ..   20
     3 .     srusthi          hubli           78999989.       20
```

6 rows selected.

INNER JOIN

```
SQL> select stdcoursenew.courseid,studentnew.name,studentnew.age from studentnew
  2  inner join stdcoursenew
  3  on studentnew.rollno=stdcoursenew.rollno;
```

OUTPUT:

```
COURSEID             NAME                    AGE
-------------------- ----------------------- ----------
bsc123               arpita                  19
bsc123               bindu                   18
bsc123               chandrika               19
bsc123               deeikas                 20
bsc123               srusthi                 20
bsc123               srusthi                 20
6 rows selected.
```

RIGHT JOIN

```
SQL> SELECT Studentnew.NAME,Stdcoursenew.courseid
  2  FROM Studentnew
  3  RIGHT JOIN Stdcoursenew
  4  ON StdCoursenew.rollno = Studentnew.rollno;
```

OUTPUT:

```
NAME                 COURSEID
-------------------- ------------------------
arpita               bsc123
bindu                bsc123
chandrika            bsc123
deeikas              bsc123
srusthi              bsc123
srusthi              bsc123
```

FULL JOIN

```
SQL> SELECT Studentnew.NAME,Stdcoursenew.courseid
  2  FROM Studentnew
  3  FULL JOIN Stdcoursenew
  4  ON StdCoursenew.ROLLNO = Studentnew.ROLLNO;
```

OUTPUT:

```
NAME                 COURSEID
-------------------- ------------------------
arpita               bsc123
```

srusthi bsc123
bindu bsc123
srusthi bsc123
deeikas bsc123
chandrika bsc123

7 rows selected.

LEFT JOIN

```
SQL> select studentnew.name,stdcoursenew.courseid
  2  from studentnew
  3  left join stdcoursenew
  4  on stdcoursenew.rollno=studentnew.rollno;
```

OUTPUT:

```
NAME                 COURSEID
-------------------- -------------------------
arpita               bsc123
srusthi              bsc123
bindu                bsc123
srusthi              bsc123
deeikas              bsc123
chandrika.           bsc123

6 rows selected.
```

Assignment 6

Perform the String Functions, Date functions and Mathematical functions supported by Oracle

Demonstrate the usage of String Functions

```
SQL> select ascii('t') from dual;

ASCII('T')
----------
       116

SQL> select ascii('a') from dual;

ASCII('A')
----------
97

SQL> select ascii('A') from dual;

ASCII('A')
----------
65


SQL> select ascii('Z') from dual;

ASCII('Z')
----------
90

SQL> select ascii('z') from dual;

ASCII('Z')
----------
122


SQL> SELECT UPPER('BANGALORE') FROM DUAL;

SQL> select upper('Bangalore')from dual;

UPPER('BA
---------
BANGALORE

SQL> SELECT UPPER('WELCOME TO JAMKHANDI')FROM DUAL;

UPPER('WELCOMETOJAMK
--------------------
WELCOME TO JAMKHANDI
```

```
SQL> SELECT LOWER('WELCOME TO DBMS LAB') FROM DUAL;

LOWER('WELCOMETODBM
-------------------
welcome to dbms lab



SQL> SELECT REPLACE('HELLO','H','K') FROM DUAL;

REPLA
-----
KELLO

SQL> SELECT REPLACE('COMPUTER','C','K') FROM DUAL;

REPLACE(
--------
KOMPUTER


SQL> SELECT REPLACE('HELLO','L','A') FROM DUAL;

REPLA
-----
HEAAO

SQL> SELECT TRIM('A' FROM 'ANJANA') FROM DUAL;

TRIM
----
NJAN


SQL> SELECT LTRIM('ANJANA','A') FROM DUAL;

LTRIM
-----
NJANA

SQL> SELECT LTRIM('ANIL','A') FROM DUAL;

LTR
---
NIL

SQL> SELECT RTRIM('ANITA','A') FROM DUAL;

RTRI
```

```
----
ANIT

SQL> SELECT RTRIM('ANJANA','A')FROM DUAL;

RTRIM
-----
ANJAN
```

Demonstrate the usage of Date Functions

Query 1:

```
SQL> SELECT CURRENT_DATE FROM DUAL;

CURRENT_D
---------
28-JUL-22
```

Query 2:

```
SQL>  SELECT EXTRACT(YEAR FROM SYSDATE) FROM DUAL;

EXTRACT(YEARFROMSYSDATE)
------------------------
                    2022
```

Query 3:

```
SQL> SELECT EXTRACT (DAY FROM SYSDATE) FROM DUAL;

EXTRACT(DAYFROMSYSDATE)
-----------------------
                     28
```

Query 4:

```
SQL> SELECT EXTRACT(MONTH FROM SYSDATE) FROM DUAL;

EXTRACT(MONTHFROMSYSDATE)
-------------------------
                       7
```

Query 5:

```
SQL> SELECT SYSDATE FROM DUAL;
```

```
SYSDATE
---------
28-JUL-22
```

Query 6:

```
SQL> SELECT MONTHS_BETWEEN(SYSDATE,'09-JAN-2021') AS "EXPERIENCE" FROM DUAL;

EXPERIENCE
----------
19.5761331
```

Query 7:

```
SQL> SELECT ADD_MONTHS(SYSDATE,5)  FROM DUAL;

ADD_MONTH
---------
26-JAN-23
```

Query 8:

```
SQL> SELECT NEXT_DAY('28-JULY-2022','FRIDAY')  FROM DUAL;

NEXT_DAY(
---------
29-JUL-22
```

Demonstrate the usage of Mathematical Functions

```
SQL> select ABS(-100) from dual;

 ABS(-100)
----------
       100


SQL> select FLOOR(2345.78) FROM DUAL;

FLOOR(2345.78)
--------------
          2345
```

```
SQL> SELECT GREATEST(23,67,90,123,78,50) FROM DUAL;

GREATEST(23,67,90,123,78,50)
----------------------------
                         123


SQL> SELECT LEAST(34, 21,67,11,89,9) FROM DUAL;

LEAST(34,21,67,11,89,9)
-----------------------
                      9

SQL> SELECT LENGTH('RAJESHWARI') FROM DUAL;

LENGTH('RAJESHWARI')
--------------------
                  10


SQL> SELECT LENGTH(17245637) FROM DUAL;

LENGTH(17245637)
----------------
               8


SQL> SELECT SQRT(16) FROM DUAL;

  SQRT(16)
----------
         4


SQL> SELECT POWER(2,4) FROM DUAL;

POWER(2,4)
----------
        16


SQL>  SELECT power(2,10) FROM DUAL;

POWER(2,10)
-----------
       1024
```

```
SQL> SELECT ROUND(5.86) FROM DUAL;

ROUND(5.86)
-----------
          6

SQL> SELECT ROUND(1001.6) FROM DUAL;
ROUND(1001.6)
-------------
        1002



SQL> SELECT ROUND(1001.2) FROM DUAL;

ROUND(1001.2)
-------------
        1001


SQL> SELECT SIN(90) FROM DUAL;

   SIN(90)
----------
.893996664


SQL>  SELECT COS(45) FROM DUAL;

   COS(45)
----------
.525321989



SQL> SELECT MOD(4,3) FROM DUAL;

  MOD(4,3)
----------
        1

SQL> SELECT MOD(5,3) FROM DUAL;

  MOD(5,3)
----------
        2
```

Assignment 7

For a given set of relation tables

EID	ENAME	ADDRESS	SALARY	DEPTNO
1	aruna	delhi	25000	101
2	Aarushi	haryana	30000	102
3	ashrita	H.P	20000	103
4	Suprita	MP	2500	104
5	Drishti	AP	35000	105

perform the following

a. Creating Views (with and without check option),
b. Selecting from a view
c. Dropping views,

Solution:

a. Creating Views (with and without check option)

Example 1.

```
SQL> create view emps as
  2 select ename, salary
  3 from employee1;
```

View created.

Example 2.

```
SQL> create view empss as
  2 select ename, salary, address
  3  from employee1
  4  where dno=104
```

```
5     with check option ;
```

b. Selecting from a view .

Example 1:

```
SQL> select *from emps;

ENAME                SALARY
-------------------- ----------
aruna                25000
arushi               30000
ashrita              20000
Suprita               25000
Drishti              35000
```

Example 2.

```
SQL> select *from empss;

ENAME                SALARY ADDRESS
-------------------- ---------- --------------------
Suprita               25000       MP
```

c. Dropping views,

```
SQL> drop view emps;

View dropped.
```