MALAD KANDIVALI EDUCATION SOCIETY'S

**NAGINDAS KHANDWALA COLLEGE OF COMMERCE,
ARTS & MANAGEMENT STUDIES & SHANTABEN NAGINDAS
KHANDWALA COLLEGE OF SCIENCE
MALAD [W], MUMBAI – 64
(AUTONOMOUS)**

(Reaccredited 'A' Grade by NAAC)
(AFFILIATED TO UNIVERSITY OF MUMBAI)
(ISO 9001:2015)

CERTIFICATE

Name: Mr. Aman Gupta

Roll No: 17 **Programme**: BSc IT **Semester**: II

This is certified to be a bonafide record of practical works done by the above student in the college laboratory for the course **Database Management Systems I** (Course Code: **2023UISPR**) for the partial fulfillment of Second Semester of BSc IT during the academic year 2020-2021.

The journal work is the original study work that has been duly approved in the year 2020-2021 by the undersigned.

_____ _____

External Examiner **Subject-In-Charge**

(Ms.Sweety Garg)

Date of Examination: (College Stamp)

Name: Aman Gupta **Roll No: 17**

Sr. No.	DATE	TITLE	SIGN
1.		Study of Data Definition Language Statement	
2.		Study of Data Manipulation Language Statement	
3.		Study of SELECT Statement.	
4.		Draw ER diagram for given scenario/project/case study	
5.		Study of various type of JOINS	

6.		Study of different functions	
7.		Study of various types of SET OPERATORS	
8.		Study of various types of views	
9.		Study of subqueries with all its clauses	
10.		Study of Transaction (Commit/ Rollback), Locks	
11.		Implementing deadlocks	

Practical 1: Study of Data Definition Language Statement

A) Write the query for the following

1) Create the following tables and include the necessary constraints NOT NULL, DEFAULT, CHECK, PRIMARY KEY, UNIQUE.

a) Student (sid, sname, gender, dob, remark, marks, class, email)

```
create table student(sid int constraint pksid primary key,sname varchar(10) not null,dob date,gender varchar(10) not null,remark varchar(5),mark int check(mark>12),class varchar(6) default'BSCIT',email varchar(7) not null);

desc student;
```

TABLE STUDENT

Column	Null?	Type
SID	NOT NULL	NUMBER
SNAME	NOT NULL	VARCHAR2(10)
DOB	-	DATE
GENDER	NOT NULL	VARCHAR2(10)
REMARK	-	VARCHAR2(5)
MARK	-	NUMBER
CLASS	-	VARCHAR2(6)
EMAIL	NOT NULL	VARCHAR2(7)

b) Course (cid, cname, credits)

create table course(cid int primary key,cname varchar(12) not null,credits int);

desc course;

TABLE COURSE

Column	Null?	Type
CID	NOT NULL	NUMBER
CNAME	NOT NULL	VARCHAR2(12)
CREDITS	-	NUMBER

2) Alter the structure of the Course table

a) Modify datatype of cname.

alter table course

modify cname varchar(12);

desc course;

```
Table altered.


TABLE COURSE
```

Column	Null?	Type
SID	NOT NULL	NUMBER
CNAME	NOT NULL	VARCHAR2(12)
CREDITS	-	NUMBER

b) Add a column coursehours with minimum course hours greater than 45.

alter table course

add coursehours int check(coursehour>45);

desc course;

```
Table altered.


TABLE COURSE
```

Column	Null?	Type
CID	NOT NULL	NUMBER
CNAME	NOT NULL	VARCHAR2(12)
CREDITS	-	NUMBER
COURSEHOURS	-	NUMBER

c) Add a column cdesc

alter table course

add cdesc varchar(10);

desc course;

Table altered.

TABLE COURSE

Column	Null?	Type
CID	NOT NULL	NUMBER
CNAME	NOT NULL	VARCHAR2(12)
CREDITS	-	NUMBER
COURSEHOURS	-	NUMBER
CDESC	-	VARCHAR2(10)

Download CSV

3) Alter the structure of Student Table

a) Add column age with minimum age as 17.

```
alter table student
add age int check(age>16);
desc student;
```

TABLE STUDENT

Column	Null?	Type
SID	NOT NULL	NUMBER
SNAME	NOT NULL	VARCHAR2(10)
DOB	-	DATE
GENDER	NOT NULL	VARCHAR2(10)
REMARK	-	VARCHAR2(5)
MARK	-	NUMBER
CLASS	-	VARCHAR2(6)
EMAIL	NOT NULL	VARCHAR2(7)
AGE	-	NUMBER

b) Delete the column dob

```
Table altered.
```

```
TABLE STUDENT
```

Column	Null?	Type
SID	NOT NULL	NUMBER
SNAME	NOT NULL	VARCHAR2(10)
GENDER	NOT NULL	VARCHAR2(10)
REMARK	-	VARCHAR2(5)
MARK	-	NUMBER
CLASS	-	VARCHAR2(6)
EMAIL	NOT NULL	VARCHAR2(7)
AGE	-	NUMBER

c) Add a column phoneno.

```
Table altered.
```

```
TABLE STUDENT
```

Column	Null?	Type
SID	NOT NULL	NUMBER
SNAME	NOT NULL	VARCHAR2(10)
GENDER	NOT NULL	VARCHAR2(10)
REMARK	-	VARCHAR2(5)
MARK	-	NUMBER
CLASS	-	VARCHAR2(6)
EMAIL	NOT NULL	VARCHAR2(7)
AGE	-	NUMBER
PHONENO	-	NUMBER

d) Rename phoneno to contactno

TABLE STUDENT

Column	Null?	Type
SID	NOT NULL	NUMBER
SNAME	NOT NULL	VARCHAR2(10)
GENDER	NOT NULL	VARCHAR2(10)
REMARK	-	VARCHAR2(5)
MARK	-	NUMBER
CLASS	-	VARCHAR2(6)
EMAIL	NOT NULL	VARCHAR2(7)
AGE	-	NUMBER
CONTACTNO	-	NUMBER

4) Rename Student table as Student_details.

alter table student

rename to student_details;

desc student_details;

Table altered.

TABLE STUDENT_DETAILS

Column	Null?	Type
SID	NOT NULL	NUMBER
SNAME	NOT NULL	VARCHAR2(10)
GENDER	NOT NULL	VARCHAR2(10)
REMARK	-	VARCHAR2(5)
MARK	-	NUMBER
CLASS	-	VARCHAR2(6)
EMAIL	NOT NULL	VARCHAR2(7)
AGE	-	NUMBER
PHONENO	-	NUMBER

5) Describe the structure of both the tables.

desc student_details;

```
Table altered.
```

```
TABLE STUDENT_DETAILS
```

Column	Null?	Type
SID	NOT NULL	NUMBER
SNAME	NOT NULL	VARCHAR2(10)
GENDER	NOT NULL	VARCHAR2(10)
REMARK	-	VARCHAR2(5)
MARK	-	NUMBER
CLASS	-	VARCHAR2(6)
EMAIL	NOT NULL	VARCHAR2(7)
AGE	-	NUMBER
PHONENO	-	NUMBER

6) Drop the table Student_details and Course.

alter table student_details;

drop table student_details;

```
Table dropped.
```

B) 1. Create a table EMPLOYEE with following attributes and specific data types and constraints required (Emp_no, E_name, E_address, E_ph_no, Dept_no, Dept_name, Job_id, Salary)

create table EMPLOYEE(Emp_no int primary key,E_name varchar(10) not null,E_address varchar(20) not null,E_ph_no int,Dept_no int,Dept_name varchar(12),Job_id char,Salary int default'15000');

desc EMPLOYEE;

TABLE EMPLOYEE

Column	Null?	Type
EMP_NO	NOT NULL	NUMBER
E_NAME	NOT NULL	VARCHAR2(10)
E_ADDRESS	NOT NULL	VARCHAR2(20)
E_PH_NO	-	NUMBER
DEPT_NO	-	NUMBER
DEPT_NAME	-	VARCHAR2(12)
JOB_ID	-	CHAR(1)
SALARY	-	NUMBER

2. Add a new column HIREDATE to the existing relation.

```
alter table EMPLOYEE
add HIREDATE date;
desc EMPLOYEE;
```

Table altered.

TABLE EMPLOYEE

Column	Null?	Type
EMP_NO	NOT NULL	NUMBER
E_NAME	NOT NULL	VARCHAR2(10)
E_ADDRESS	NOT NULL	VARCHAR2(20)
E_PH_NO	-	NUMBER
DEPT_NO	-	NUMBER
DEPT_NAME	-	VARCHAR2(12)
JOB_ID	-	CHAR(1)
SALARY	-	NUMBER
HIREDATE	-	DATE

3. Change the datatype of JOB_ID from char to varchar2.

alter table EMPLOYEE

modify job_id varchar2;

desc EMPLOYEE;

Table altered.

TABLE EMPLOYEE

Column	Null?	Type
EMP_NO	NOT NULL	NUMBER
E_NAME	NOT NULL	VARCHAR2(10)
E_ADDRESS	NOT NULL	VARCHAR2(20)
E_PH_NO	-	NUMBER
DEPT_NO	-	NUMBER
DEPT_NAME	-	VARCHAR2(12)
JOB_ID	-	CHAR(1)
SALARY	-	NUMBER
HIREDATE	-	DATE

4. Change the name of column/field Emp_no to E_no.

alter table EMPLOYEE

rename column Emp_no to E_no;

desc EMPLOYEE;

Table altered.

TABLE EMPLOYEE

Column	Null?	Type
EMP_NO	NOT NULL	NUMBER
E_NAME	NOT NULL	VARCHAR2(10)
E_ADDRESS	NOT NULL	VARCHAR2(20)
E_PH_NO	-	NUMBER
DEPT_NO	-	NUMBER
DEPT_NAME	-	VARCHAR2(12)
JOB_ID	-	CHAR(1)
SALARY	-	NUMBER
HIREDATE	-	DATE

5. Modify the column width of the job field of emp table.

alter table EMPLOYEE

modify job_id varchar(30);

desc EMPLOYEE;

TABLE EMPLOYEE

Column	Null?	Type
E_NO	NOT NULL	NUMBER
E_NAME	NOT NULL	VARCHAR2(10)
E_ADDRESS	NOT NULL	VARCHAR2(20)
E_PH_NO	-	NUMBER
DEPT_NO	-	NUMBER
DEPT_NAME	-	VARCHAR2(12)
JOB_ID	-	VARCHAR2(30)
SALARY	-	NUMBER
HIREDATE	-	DATE

C**) Create the following tables with specified attributes and constraints Department Table**:

Department_Id varchar2(20) primary key, Department_Name varchar2(25) with required data.

create table Department(Department_Id varchar2(10) primary key, Department_Name varchar2(15) not null);

desc Department;

Table created.

TABLE DEPARTMENT

Column	Null?	Type
DEPARTMENT_ID	NOT NULL	VARCHAR2(10)
DEPARTMENT_NAME	NOT NULL	VARCHAR2(15)

Instructor Table: Instructor_id varchar2(20) primary key, Department_Id varchar2(20) Foreign key, Last_Name varchar2(25), First_Name varchar2(200) must have value, Telephone varchar2(20) must be unique, gender char (1) must be either ‘F’or ‘M’, city varchar (10) default value must be ‘MUMBAI’.

create table Instructor(Instructor_id varchar2(20) primary key,Department_Id varchar2(10),foreign key(Department_id) references Department(Department_id),Last_Name varchar2(12),First_Name varchar2(10) not null,Telephone varchar2(10) unique,gender char(1) check(gender in ('M''F')),city varchar(10) default'MUMBAI');

desc Instructor;

TABLE INSTRUCTOR

Column	Null?	Type
INSTRUCTOR_ID	NOT NULL	VARCHAR2(20)
DEPARTMENT_ID	-	VARCHAR2(10)
LAST_NAME	-	VARCHAR2(12)
FIRST_NAME	NOT NULL	VARCHAR2(10)
TELEPHONE	-	VARCHAR2(10)
GENDER	-	CHAR(1)
CITY	-	VARCHAR2(10)

B) **Create the following described below:**

Table Name: EMP

Column	Data Type	Length	Precision	Scale	Primary Key	Nullable
EMPNO	Int	-	-	-	Yes	-
ENAME	Varchar2	10	-	-	-	No
JOB	Varchar2	9	-	-	-	✓
MGR	Int	-	-	-	-	✓
HIREDATE	Date	-	-	-	-	✓
SAL	Number	-	7	2	-	✓
COMM	Int	-	-	-	-	✓
DEPTNO	Int	-	-	-	-	✓

create table Emp_Aman(Empno int primary key,Ename varchar(10) not null,job varchar(9),mgr int,hiredate date,sal number(7,2),comm int,DEPTNO int);

desc Emp_Aman;

TABLE EMP_AMAN

Column	Null?	Type
EMPNO	NOT NULL	NUMBER
ENAME	NOT NULL	VARCHAR2(10)
JOB	-	VARCHAR2(9)
MGR	-	NUMBER
HIREDATE	-	DATE
SAL	-	NUMBER(7,2)
COMM	-	NUMBER
DEPTNO	-	NUMBER

Download CSV

8 rows selected.

Table Name: DEPT

Column	Data Type	Length	Precision	Scale	Primary Key	Nullable
DEPTNO	Int	-	-	-	Yes	-
DNAME	Varchar2	14	-	-	-	No
LOC	Varchar2	13	-	-	-	✓

create table Dept_Aman1(DEPTNO int primary key not null,DAME varchar(14) not null,LOC varchar(13));

desc Dept_Aman1;

Table created.

Table created.

TABLE DEPT_AMAN1

Column	Null?	Type
DEPTNO	NOT NULL	NUMBER
DAME	NOT NULL	VARCHAR2(14)
LOC	-	VARCHAR2(13)

Download CSV

3 rows selected.

Practical 2: Study of Data Manipulation Language Statement

A) Insert the following records in above created table

EMP TABLE

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7839	KING	PRESIDENT		17-Nov-81	5000		10
7698	BLAKE	MANAGER	7839	01-May-81	2850		30
7782	CLARK	MANAGER	7839	09-Jun-81	2450		10
7566	JONES	MANAGER	7839	02-Apr-81	2975		20
7788	SCOTT	ANALYST	7566	19-Apr-87	3000		20
7902	FORD	ANALYST	7566	03-Dec-81	3000		20
7369	SMITH	CLERK	7902	17-Dec-80	800		20
7499	ALLEN	SALESMAN	7698	20-Feb-81	1600	300	30
7521	WARD	SALESMAN	7698	22-Feb-81	1250	500	30
7654	MARTIN	SALESMAN	7698	28-Sep-81	1250	1400	30
7844	TURNER	SALESMAN	7698	08-Sep-81	1500	0	30
7876	ADAMS	CLERK	7788	23-May-87	1100		20
7900	JAMES	CLERK	7698	03-Dec-81	950		30
7934	MILLER	CLERK	7782	23-Jan-82	1300		10

create table Emp_Aman(Empno int primary key,Ename varchar(10) not null,job varchar(9),mgr int,HIREDATE date,SAL number(7,2),comm int,deptno int);

desc Emp_ Aman;

Table created.

TABLE EMP_AMAN

Column	Null?	Type
EMPO	NOT NULL	NUMBER
ENAME	NOT NULL	VARCHAR2(10)
JOB	-	VARCHAR2(9)
MGR	-	NUMBER
HIREDATE	-	DATE
SAL	-	NUMBER(7,2)
COMM	-	NUMBER
DEPTNO	-	NUMBER

Download CSV

8 rows selected.

insert into Emp_Aman values(7839,'KING','PRESIDENT','','17-Nov-81',5000,'',10);

1 row(s) inserted.

insert into Emp_Aman values(7698,'BLAKE','MANAGER',7839,'01-May-81',2850,'',30);

1 row(s) inserted.

insert into Emp_Aman values(7782,'CLARK','MANAGER',7839,'09-JUN-81',2450,'',10);

1 row(s) inserted.

insert into Emp_Aman values(7566,'JONES','MANAGER',7839,'02-Apr-81',2975,'',20);

1 row(s) inserted.

insert into Emp_Aman values(7788,'SCOTT','ANALYST',7566,'19-APRIL-81',3000,'',20);

1 row(s) inserted.

insert into Emp_Aman values(7902,'FORD','ANALYST',7566,'03-Dec-81',3000,'',20);

1 row(s) inserted.

insert into Emp_Aman values(7369,'SMITH','CLERK',7902,'17-Dec-80',800,'',20);

1 row(s) inserted.

insert into Emp_Aman values(7521,'WARD','SALESMAN',7698,'22-Feb-81',1250,500,30);

```
1 row(s) inserted.
```

insert into Emp_Aman values(7654,'MARTIN','SALESMAN',7698,'28-Sep-81',1250,1400,30);

```
1 row(s) inserted.
```

insert into Emp_Aman values(7844,'TURNER','SALESMAN',7698,'08-Sep-81',1500,0,30);

```
1 row(s) inserted.
```

insert into Emp_Aman values(7876,'ADAMS','CLERK',7788,'23-May-87',1100,'',20);

```
1 row(s) inserted.
```

insert into Emp_Aman values(7900,'JAMES','CLERK',7682,'03-Dec-81',950,'',30);

```
1 row(s) inserted.
```

insert into Emp_Aman values(7934,'MILLER','CLERK',7782,'23-Jan-82',1300,'',10);

```
1 row(s) inserted.
```

select * from Emp_Aman;

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7698	BLAKE	MANAGER	7839	01-MAY-81	2850	-	30
7788	SCOTT	ANALYST	7566	19-APR-81	3000	-	20
7369	SMITH	CLERK	7902	17-DEC-80	800	-	20
7876	ADAMS	CLERK	7788	23-MAY-87	1100	-	20
7839	KING	PRESIDENT	-	17-NOV-81	5000	-	10
7566	JONES	MANAGER	7839	02-APR-81	2975	-	20
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30
7902	FORD	ANALYST	7566	03-DEC-81	3000	-	20
7900	JAMES	CLERK	7682	03-DEC-81	950	-	30
7782	CLARK	MANAGER	7839	09-JUN-81	2450	-	10
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30
7844	TURNER	SALESMAN	7698	08-SEP-81	1500	0	30
7934	MILLER	CLERK	7782	23-JAN-82	1300	-	10

DEPT TABLE

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

insert into Dept_Aman1 values(10,'ACCOUNTING','NEWYORK');

```
1 row(s) inserted.
```

insert into Dept_Aman1 values(20,'RESEARCH','DALLAS');

```
1 row(s) inserted.
```

insert into Dept_Aman1 values(30,'SALES','CHICAGO');

```
1 row(s) inserted.
```

insert into Dept_Aman1 values(40,'OPERATIONS','BOSTON');

```
1 row(s) inserted.
```

DEPTNO	DAME	LOC
10	ACCOUNTING	NEWYORK
30	SALES	CHICAGO
20	RESEARCH	DALLAS
40	OPERATIONS	BOSTON

B) Update and Delete Queries

1)Update the salary of employees working as CLERK by 500.
UPDATE Emp_Aman SET SAL=SAL+500 WHERE JOB='CLERK';

Select * from Emp_Aman;

```
4 row(s) updated.
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7698	BLAKE	MANAGER	7839	01-MAY-81	2850	-	30
7788	SCOTT	ANALYST	7566	19-APR-81	3000	-	20
7369	SMITH	CLERK	7902	17-DEC-80	1300	-	20
7876	ADAMS	CLERK	7788	23-MAY-87	1600	-	20
7839	KING	PRESIDENT	-	17-NOV-81	5000	-	10
7566	JONES	MANAGER	7839	02-APR-81	2975	-	20
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30
7902	FORD	ANALYST	7566	03-DEC-81	3000	-	20
7900	JAMES	CLERK	7682	03-DEC-81	1450	-	30
7782	CLARK	MANAGER	7839	09-JUN-81	2450	-	10
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30
7844	TURNER	SALESMAN	7698	08-SEP-81	1500	0	30
7934	MILLER	CLERK	7782	23-JAN-82	1800	-	10

2)Update the manager of James as CLARK.

```
UPDATE Emp_Aman SET MGR=7782 WHERE ENAME='JAMES';
```

```
1 row(s) updated.
```

```
select * from Emp_Aman;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7698	BLAKE	MANAGER	7839	01-MAY-81	2850	-	30
7788	SCOTT	ANALYST	7566	19-APR-81	3000	-	20
7369	SMITH	CLERK	7902	17-DEC-80	1800	-	20
7876	ADAMS	CLERK	7788	23-MAY-87	2100	-	20
7839	KING	PRESIDENT	-	17-NOV-81	5000	-	10
7566	JONES	MANAGER	7839	02-APR-81	2975	-	20
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30
7902	FORD	ANALYST	7566	03-DEC-81	3000	-	20
7900	JAMES	CLERK	7782	03-DEC-81	1950	-	30
7782	CLARK	MANAGER	7839	09-JUN-81	2450	-	10
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30
7844	TURNER	SALESMAN	7698	08-SEP-81	1500	0	30
7934	MILLER	MANAGER	7782	23-JAN-82	2300	-	10

3)Change the role of Miller as MANAGER.

UPDATE Emp_Aman SET JOB='MANAGER'WHERE ENAME='MILLER';

```
1 row(s) updated.
```

select * from Emp_Aman;

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7698	BLAKE	MANAGER	7839	01-MAY-81	2850	-	30
7788	SCOTT	ANALYST	7566	19-APR-81	3000	-	20
7369	SMITH	CLERK	7902	17-DEC-80	1800	-	20
7876	ADAMS	CLERK	7788	23-MAY-87	2100	-	20
7839	KING	PRESIDENT	-	17-NOV-81	5000	-	10
7566	JONES	MANAGER	7839	02-APR-81	2975	-	20
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30
7902	FORD	ANALYST	7566	03-DEC-81	3000	-	20
7900	JAMES	CLERK	7782	03-DEC-81	1950	-	30
7782	CLARK	MANAGER	7839	09-JUN-81	2450	-	10
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30
7844	TURNER	SALESMAN	7698	08-SEP-81	1500	0	30
7934	MILLER	MANAGER	7782	23-JAN-82	2300	-	10

4)Delete the records of Manager

Delete from Emp_Aman WHERE JOB='MANAGER';

select * from Emp_Aman;

```
4 row(s) deleted.
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7788	SCOTT	ANALYST	7566	19-APR-81	3000	-	20
7369	SMITH	CLERK	7902	17-DEC-80	1800	-	20
7876	ADAMS	CLERK	7788	23-MAY-87	2100	-	20
7839	KING	PRESIDENT	-	17-NOV-81	5000	-	10
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30
7902	FORD	ANALYST	7566	03-DEC-81	3000	-	20
7900	JAMES	CLERK	7782	03-DEC-81	1950	-	30
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30
7844	TURNER	SALESMAN	7698	08-SEP-81	1500	0	30

5)Delete the records when salary is greater than 1000.

Delete from Emp_Aman WHERE SAL>1000;

select * from Emp_Aman;

```
9 row(s) deleted.

no data found
```

- create table Emp_Aman(Empno int primary key,Ename varchar(10) not null,job varchar(9),mgr int,HIREDATE date,SAL number(7,2),comm int,deptno int,foreign key(DEPTNO) references DEPT_Aman(DEPTNO));

  desc Emp_Aman;

Table created.

TABLE EMP_AMAN

Column	Null?	Type
EMPO	NOT NULL	NUMBER
ENAME	NOT NULL	VARCHAR2(10)
JOB	-	VARCHAR2(9)
MGR	-	NUMBER
HIREDATE	-	DATE
SAL	-	NUMBER(7,2)
COMM	-	NUMBER
DEPTNO	-	NUMBER

Download CSV

8 rows selected.

PRACTICAL 3

A) Using emp table, perform the following queries:

1) Display the details of all employees.

select * from Emp_Aman;

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7839	KING	PRESIDENT	-	17-NOV-81	5000	-	10
7698	BLAKE	MANAGER	7839	01-MAY-81	2850	-	30
7782	CLARK	MANAGER	7839	09-JUN-81	2450	-	10
7566	JONES	MANAGER	7839	02-APR-81	2975	-	20
7788	SCOTT	ANALYST	7566	19-APR-81	3000	-	20
7902	FORD	ANALYST	7566	03-DEC-81	3000	-	20
7369	SMITH	CLERK	7902	17-DEC-80	800	-	20
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30
7844	TURNER	SALESMAN	7698	08-SEP-81	1500	0	30
7876	ADAMS	CLERK	7788	23-MAY-87	1100	-	20
7900	JAMES	CLERK	7682	03-DEC-81	950	-	30
7934	MILLER	CLERK	7782	23-JAN-82	1300	-	10

Download CSV

2) Display the name and job for all employees.

select ename,job from Emp_Aman;

ENAME	JOB
KING	PRESIDENT
BLAKE	MANAGER
CLARK	MANAGER
JONES	MANAGER
SCOTT	ANALYST
FORD	ANALYST
SMITH	CLERK
WARD	SALESMAN
MARTIN	SALESMAN
TURNER	SALESMAN
ADAMS	CLERK
JAMES	CLERK
MILLER	CLERK

3) Display name and salary for all employees.

select ename,sal from Emp_Aman;

ENAME	SAL
KING	5000
BLAKE	2850
CLARK	2450
JONES	2975
SCOTT	3000
FORD	3000
SMITH	800
WARD	1250
MARTIN	1250
TURNER	1500
ADAMS	1100
JAMES	950
MILLER	1300

4) Display the details of all employees who are earning salary greater than 2000.

```
select * from Emp_Aman
where sal>2000;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7839	KING	PRESIDENT	-	17-NOV-81	5000	-	10
7698	BLAKE	MANAGER	7839	01-MAY-81	2850	-	30
7782	CLARK	MANAGER	7839	09-JUN-81	2450	-	10
7566	JONES	MANAGER	7839	02-APR-81	2975	-	20
7788	SCOTT	ANALYST	7566	19-APR-81	3000	-	20
7902	FORD	ANALYST	7566	03-DEC-81	3000	-	20

Download CSV

5) Display the details of all employees who are working as Manager.

```
select * from Emp_Aman
where job='MANAGER';
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7698	BLAKE	MANAGER	7839	01-MAY-81	2850	-	30
7782	CLARK	MANAGER	7839	09-JUN-81	2450	-	10
7566	JONES	MANAGER	7839	02-APR-81	2975	-	20

Download CSV

6) Display the names of all employees who are working in department number 10.

```
select ename from Emp_Aman
where deptno=10;
```

ENAME
KING
CLARK
MILLER

7) Display the names of all employees working as clerk and drawing a salary more than 3000.

```
select ename from Emp_Aman
where job='clerk' or sal>3000;
```

ENAME
KING

8) Display employee number and names for employees who earn commission.

```
select empno,ename from Emp_Aman
where comm is not null;
```

EMPNO	ENAME
7521	WARD
7654	MARTIN
7844	TURNER

9) Display names of employees who do not earn any commission.

```
select ename from Emp_Aman
where comm is null;
```

ENAME
KING
BLAKE
CLARK
JONES
SCOTT
FORD
SMITH
ADAMS
JAMES
MILLER

10) Display the names of employees who are working as clerk, salesman or analyst and drawing a salary more than 2000.

```
select ename from Emp_Aman
where job='CLERK' or job='SALESMAN' or job='ANALYST' or sal>2000;
```

ENAME
KING
BLAKE
CLARK
JONES
SCOTT
FORD
SMITH
WARD
MARTIN
TURNER
ADAMS
JAMES
MILLER

11) Display the names of employees who are working as clerk, salesman or analyst.

```
select ename from Emp_Aman
where job in ('CLERK','SALESMAMN','ANALYST');
```

ENAME
SCOTT
FORD
SMITH
ADAMS
JAMES
MILLER

Download CSV

12) Display the names of employees working in department number 10 or 20 or 30.

```
select ename from Emp_Aman
where deptno in (10,20,30);
```

ENAME
KING
BLAKE
CLARK
JONES
SCOTT
FORD
SMITH
WARD
MARTIN
TURNER
ADAMS
JAMES
MILLER

13) Display the details of employees whose salary lies in the range of 1000 and 2000.

```
select ename from Emp_Aman
where sal between 1000 and 2000;
```

ENAME
WARD
MARTIN
TURNER
ADAMS
MILLER

14) List the employees in the ascending order of their salaries.

```
select ename from Emp_Aman
order by ename asc;
```

ENAME
ADAMS
BLAKE
CLARK
FORD
JAMES
JONES
KING
MARTIN
MILLER
SCOTT
SMITH
TURNER
WARD

15) List the Empno, Ename, Sal of all emps working for Mgr 7369.

```
select Empno,ename,sal
 from Emp_Aman
 where mgr=7369;
```

```
no data found
```

16) List the employees who are either ‘CLERK’ or ‘ANALYST’ in the Desc order.

```
 select ename from Emp_Aman
 where job in ('CLERK','ANALYST')
 order by ename desc;
```

ENAME
SMITH
SCOTT
MILLER
JAMES
FORD
ADAMS

17) List the employees who are working in Deptno 10 or 20.

```
 select ename from Emp_Aman
 where deptno in (10,20);
```

ENAME
KING
CLARK
JONES
SCOTT
FORD
SMITH
ADAMS
MILLER

18) List the employees whose name have a character set ‘ll’ together.

```
select ename from Emp_Aman
where ename like '%LL%';
```

ENAME
MILLER

19) List the employees in ascending order of their names.

```
select * from Emp_Aman
order by ename asc;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7876	ADAMS	CLERK	7788	23-MAY-87	1100	-	20
7698	BLAKE	MANAGER	7839	01-MAY-81	2850	-	30
7782	CLARK	MANAGER	7839	09-JUN-81	2450	-	10
7902	FORD	ANALYST	7566	03-DEC-81	3000	-	20
7900	JAMES	CLERK	7682	03-DEC-81	950	-	30
7566	JONES	MANAGER	7839	02-APR-81	2975	-	20
7839	KING	PRESIDENT	-	17-NOV-81	5000	-	10
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30
7934	MILLER	CLERK	7782	23-JAN-82	1300	-	10
7788	SCOTT	ANALYST	7566	19-APR-81	3000	-	20
7369	SMITH	CLERK	7902	17-DEC-80	800	-	20
7844	TURNER	SALESMAN	7698	08-SEP-81	1500	0	30
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30

20) List the employees in descending order of their names.

```
select * from Emp_Aman
order by ename desc;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30
7844	TURNER	SALESMAN	7698	08-SEP-81	1500	0	30
7369	SMITH	CLERK	7902	17-DEC-80	800	-	20
7788	SCOTT	ANALYST	7566	19-APR-81	3000	-	20
7934	MILLER	CLERK	7782	23-JAN-82	1300	-	10
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30
7839	KING	PRESIDENT	-	17-NOV-81	5000	-	10
7566	JONES	MANAGER	7839	02-APR-81	2975	-	20
7900	JAMES	CLERK	7682	03-DEC-81	950	-	30
7902	FORD	ANALYST	7566	03-DEC-81	3000	-	20
7782	CLARK	MANAGER	7839	09-JUN-81	2450	-	10
7698	BLAKE	MANAGER	7839	01-MAY-81	2850	-	30
7876	ADAMS	CLERK	7788	23-MAY-87	1100	-	20

21) List the employees who do not belong to Deptno 20.

```
select ename from Emp_Aman
where deptno in (10,30);
```

ENAME
KING
BLAKE
CLARK
WARD
MARTIN
TURNER
JAMES
MILLER

22) List all the employees except PRESIDENT and MANAGER.

```
select ename from Emp_ Aman
where job not in ('PRESIDENT','MANAGER');
```

ENAME
SCOTT
FORD
SMITH
WARD
MARTIN
TURNER
ADAMS
JAMES
MILLER

23) List the employees whose name starts with A.

```
select ename from Emp_Aman
where ename like 'A%';
```

ENAME
ADAMS

24) List all the Clerks of Deptno 20.

```
select ename from Emp_Aman
where deptno=20 and job='CLERK';
```

ENAME
SMITH
ADAMS

25) List the employees whose names ends with S.

```
select ename from Emp_Aman
where ename like '%S';
```

ENAME
JONES
ADAMS
JAMES

26) List the employees who has name of exactly 4 characters.

```
select * from Emp_Aman
where ename like '____';
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7839	KING	PRESIDENT	-	17-NOV-81	5000	-	10
7902	FORD	ANALYST	7566	03-DEC-81	3000	-	20
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30

Download CSV

27) List the names of the employees who are working as MANAGER in department 10.

```
select ename from Emp_Aman
 where job='MANAGER' and deptno=10;
```

ENAME
CLARK

28) List the total salary of employees working as ANALYST.

```
 select sum(sal) from Emp_Aman
 where job='ANALYST';
```

SUM(SAL)
6000

29) List the minimum, maximum and average salary of the employees.

```
 select avg(sal),min(sal),max(sal) from Emp_Aman;
```

AVG(SAL)	MIN(SAL)	MAX(SAL)
2109.615384615384615384615384615385	800	5000

30) List the total number of employees working in department 10.

```
 select count(*) from Emp_Aman
where deptno=10;
```

COUNT(*)
3

B) Answer the following queries:

1) Display the total salary of employees department wise.

```
select deptno,sum(sal) from Emp_Aman
group by deptno;
```

DEPTNO	SUM(SAL)
30	7800
10	8750
20	10875

Download CSV

2) Display the total salary of employees job wise in ascending order of job.

```
select job,sum(sal) from Emp_Aman
group by job
order by job asc;
```

JOB	SUM(SAL)
ANALYST	6000
CLERK	4150
MANAGER	8275
PRESIDENT	5000
SALESMAN	4000

3) Display the total number of employees with specific job.

```
select job,count(*) from Emp_Aman
group by job;
```

JOB	COUNT(*)
ANALYST	2
CLERK	4
SALESMAN	3
MANAGER	3
PRESIDENT	1

4) Display the total number of employees working in each department.

```
select deptno,count(*) from Emp_Aman
group by deptno;
```

DEPTNO	COUNT(*)
30	5
10	3
20	5

Download CSV

5) Display the total salary of employees specific to job and department in ascending order of job.

```
select job,deptno,sum(sal) from Emp_Aman
group by job,deptno
order by job asc;
```

JOB	DEPTNO	SUM(SAL)
ANALYST	20	6000
CLERK	10	1300
CLERK	20	1900
CLERK	30	950
MANAGER	10	2450
MANAGER	20	2975
MANAGER	30	2850
PRESIDENT	10	5000
SALESMAN	30	4000

Download CSV

6) Display the total salary of the employees specific to job when employee count is greater than 1.

```
select job,sum(sal),count(*) from Emp_Aman
group by job
having count(job)>1;
```

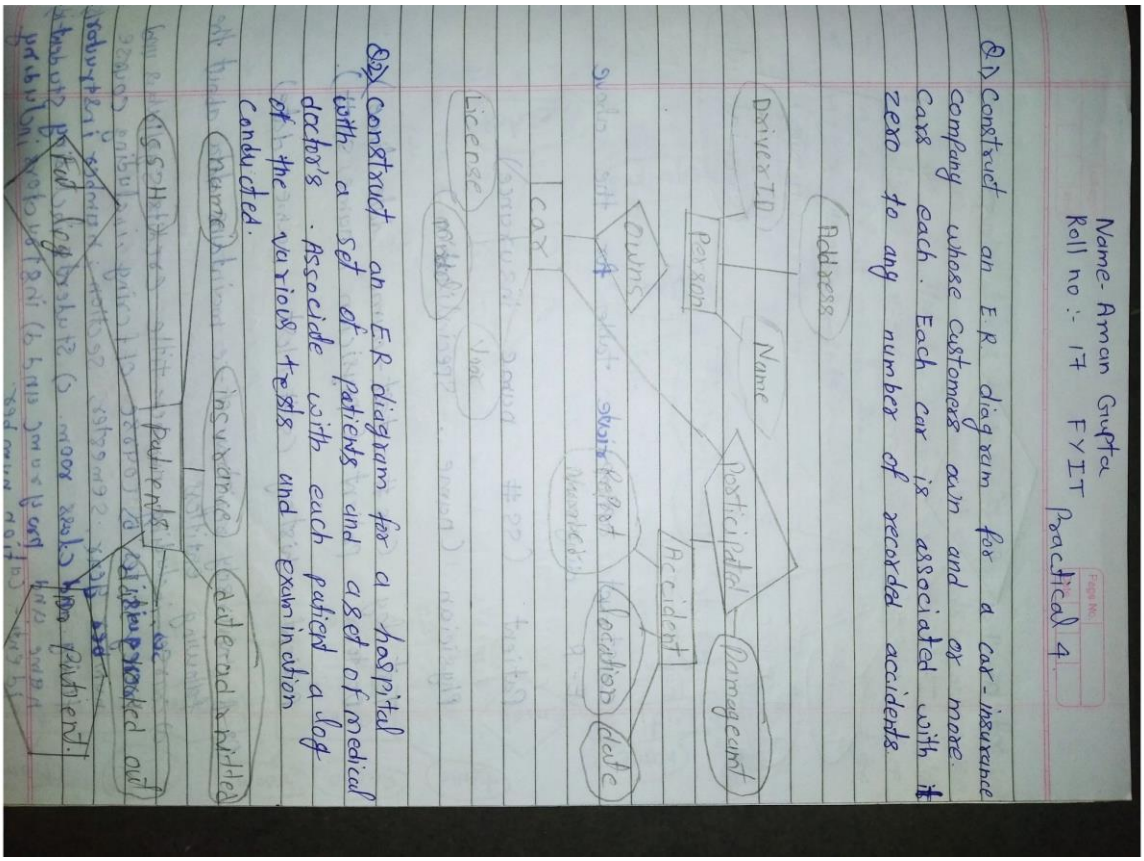
JOB	SUM(SAL)	COUNT(*)
ANALYST	6000	2
CLERK	4150	4
SALESMAN	4000	3
MANAGER	8275	3

7) Display unique jobs of employees.

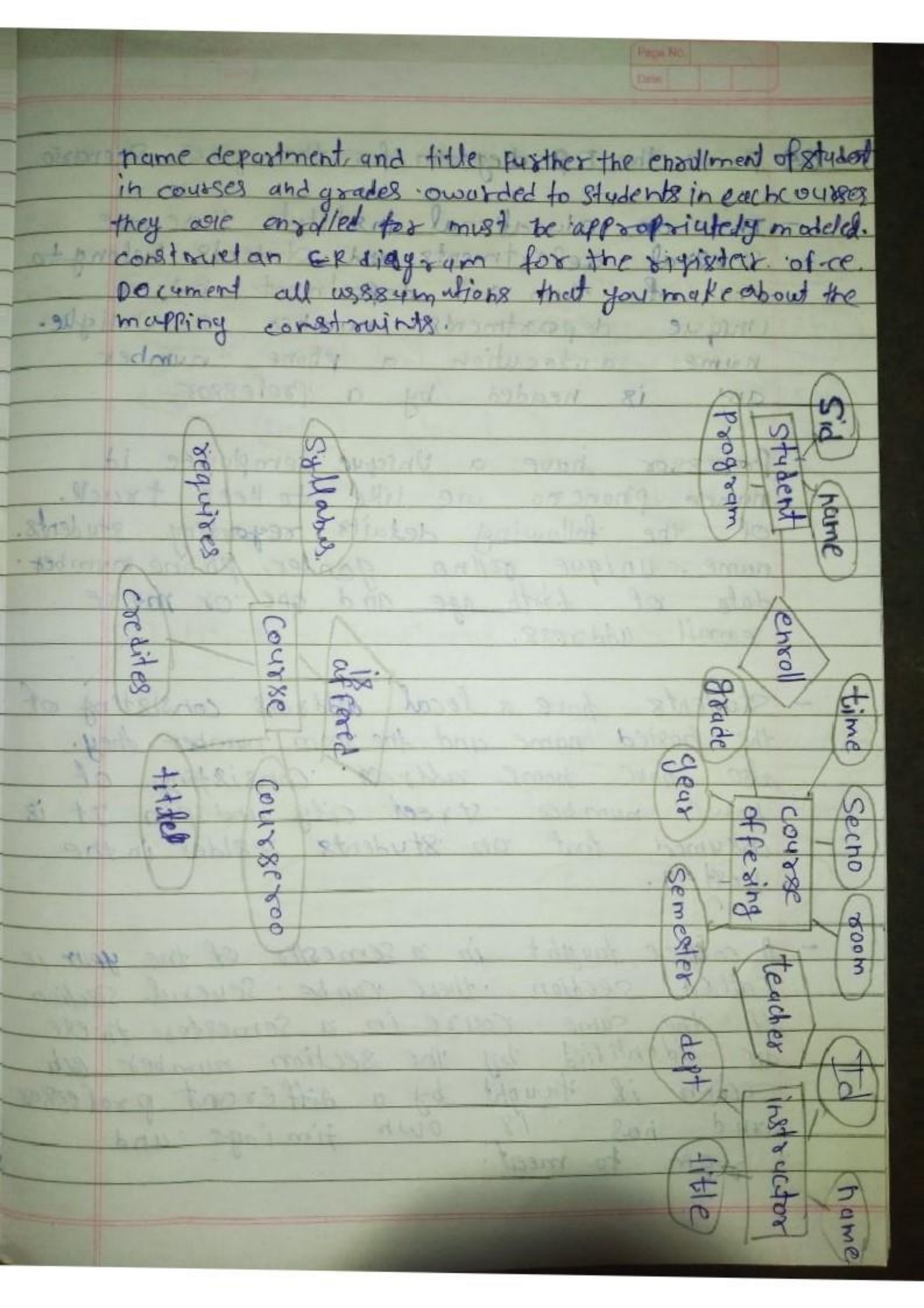
select distinct job from Emp_Aman;

JOB
ANALYST
CLERK
SALESMAN
MANAGER
PRESIDENT

PRACTICAL 4



name department and title further the enrollment of student in courses and grades awarded to students in each courses they are enrolled for must be appropriately modeled. construct an ER diagram for the registrar office. Document all assumptions that you make about the mapping constraints.



Q.4) Draw the E-R diegram for the gives Scenario.

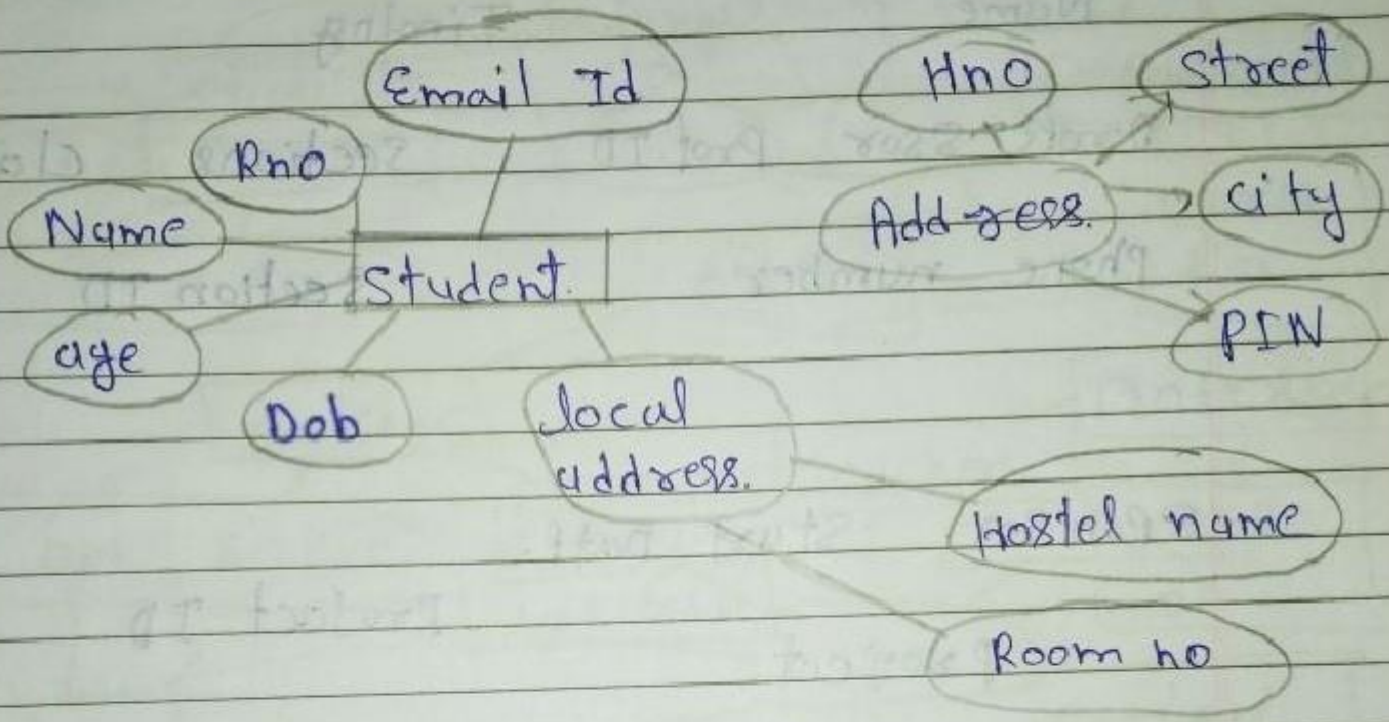
- In an educational institute there are several departments and students belong to one of them each department has a unique departments number a unique name a location a phone number and is headed by a professor.

- Professor have a Unique employee id name phone no we like to keep track. of the following details regarding students. name. unique rollno gender, phone number data of birth age and one or more email address.

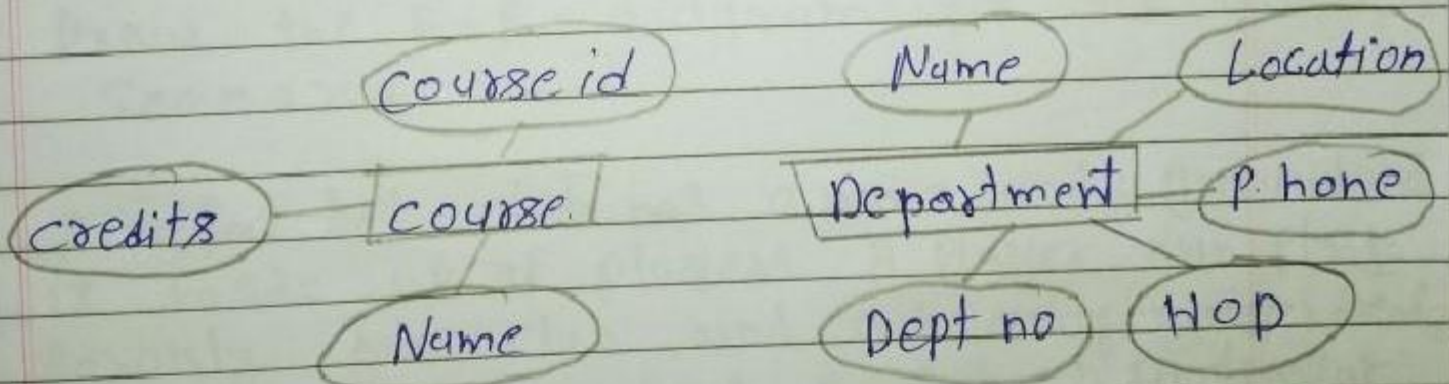
- Students have a local address consisting of the hosted name and the room number they also have home address consisting of house number street city and pin It is assumed that an students resides in the hostels.

- A course taught in a semester of the year is called section there can be several section of the same coure in a semester these are identified by the section number each section is taught by a different professor and has its own timings and room to meet.

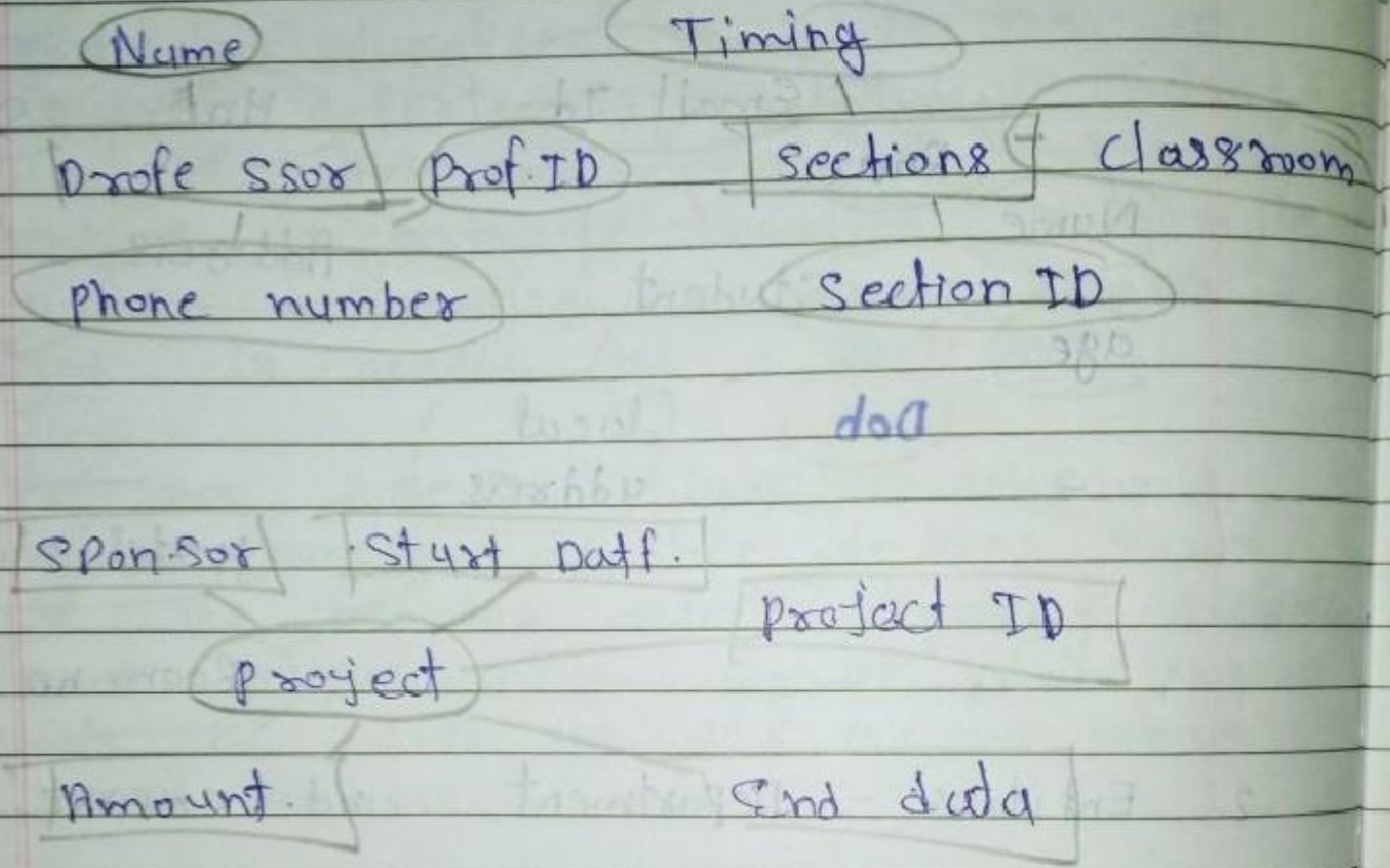
1. Entities - Student



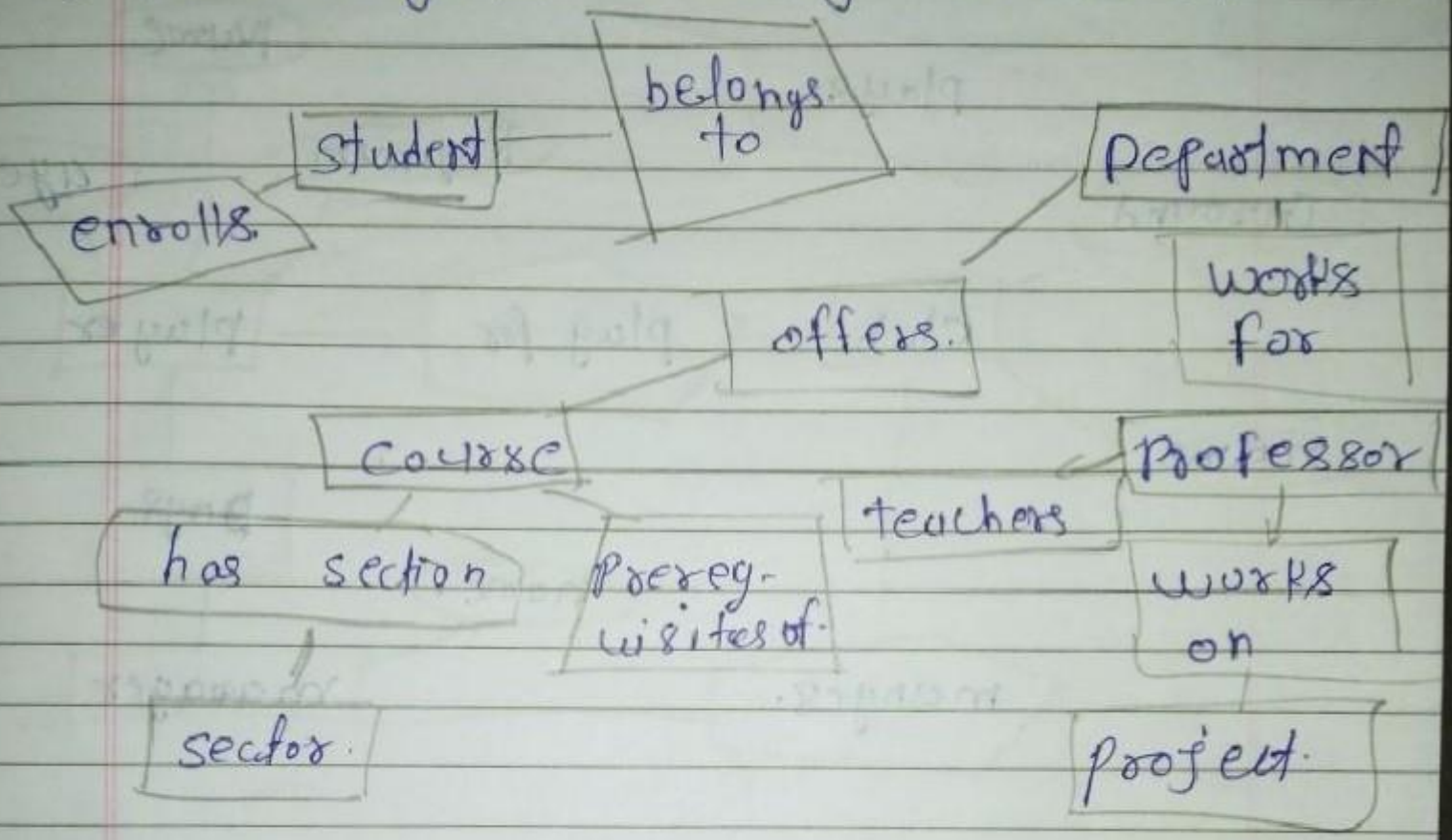
2. Entities - Department and course.



3) Entities = professor, Project and Sections.

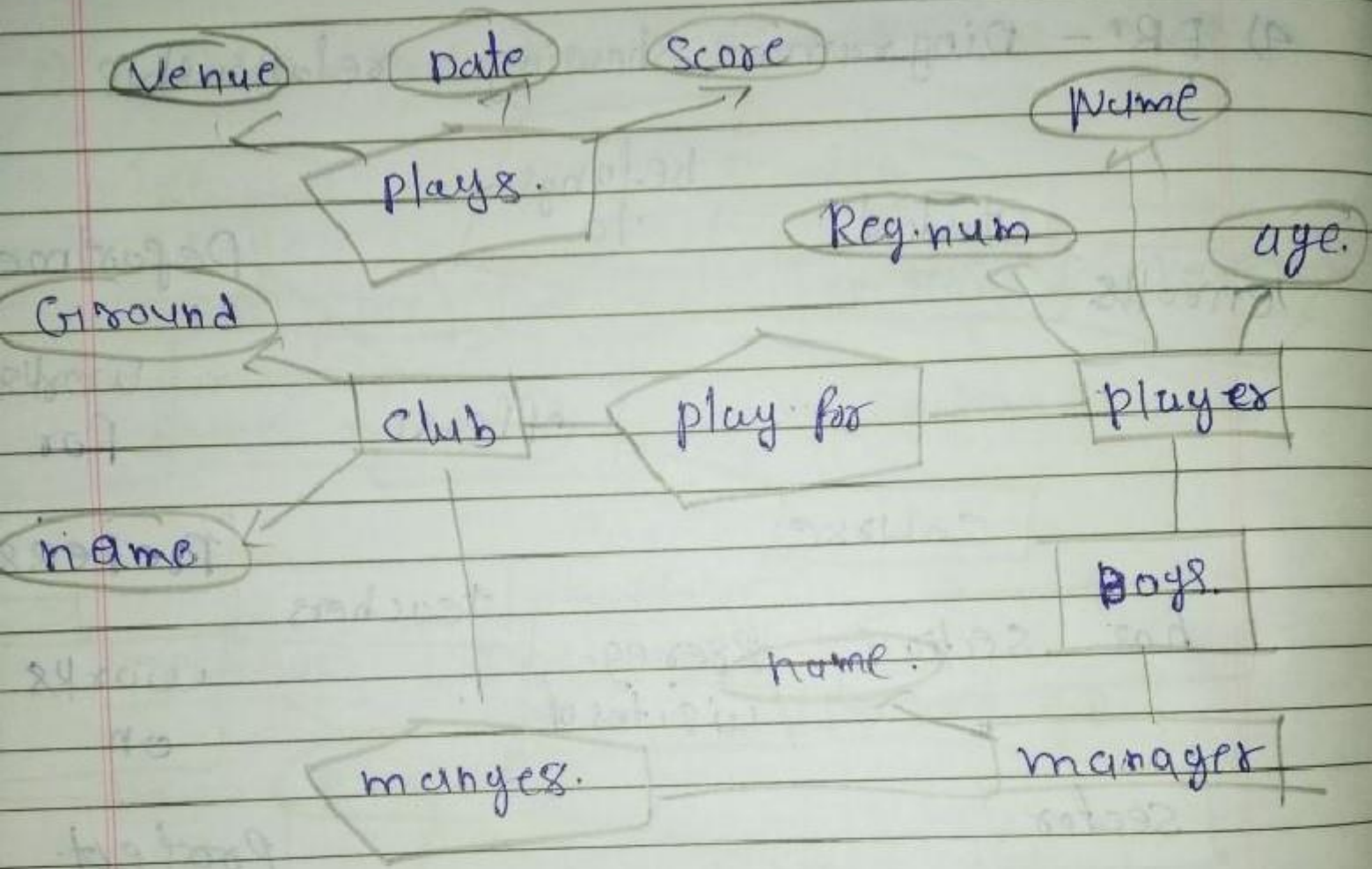


4) ER - Diagram Showing relationship



Q.5) Draw the E-R diagram for the given Seenario

- A foot boll club has a name and a ground and is made up of players. A Player can play. fo only one club and a manager represents by his name. manages a club. A footballer has a registration number name and age. A club manager also bys Player Each club. plays against each other club in the league and matches have a date venue and score.


Venue
Date
Score
plays.
Name
Reg.num
age.
Ground
club
play for
player
name
Boys.
name.
manages.
manager

Practial No .5 (JOIN)

Join: JOIN means "to combine two or more tables". JOIN clause is used to combine the records from two or more tables in a database.

Inner Join: the INNER JOIN to fetch rows that have matching values in both the tables that we are joining.

Syntax:

1. SELECT columns
2. FROM table1
3. INNER JOIN table2
4. ON table1.column = table2.column;

select ENAME,DAME,LOC,Sal from

Emp_Aman inner join Dept_Aman1

on Emp_Aman.DEPTNO=Dept_Aman1.DEPTNO;

ENAME	DAME	LOC	SAL
KING	ACCOUNTING	NEWYORK	5000
BLAKE	SALES	CHICAGO	2850
CLARK	ACCOUNTING	NEWYORK	2450
JONES	RESEARCH	DALLAS	2975
SCOTT	RESEARCH	DALLAS	3000
FORD	RESEARCH	DALLAS	3000
SMITH	RESEARCH	DALLAS	800
WARD	SALES	CHICAGO	1250
MARTIN	SALES	CHICAGO	1250
TURNER	SALES	CHICAGO	1500
ADAMS	RESEARCH	DALLAS	1100
JAMES	SALES	CHICAGO	950
MILLER	ACCOUNTING	NEWYORK	1300

Natural Join: Natural join can only be performed if there is a common attribute (column) between the relations. The name and type of the attribute must be same.

Syntax:

1. SELECT *
2. FROM table1 Natural join table2;

select * from Emp_ Aman natural join Dept_ Aman 1 order by deptno;

DEPTNO	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DAME	LOC
10	7782	CLARK	MANAGER	7839	09-JUN-81	2450	-	ACCOUNTING	NEWYORK
10	7934	MILLER	CLERK	7782	23-JAN-82	1300	-	ACCOUNTING	NEWYORK
10	7839	KING	PRESIDENT	-	17-NOV-81	5000	-	ACCOUNTING	NEWYORK
20	7902	FORD	ANALYST	7566	03-DEC-81	3000	-	RESEARCH	DALLAS
20	7788	SCOTT	ANALYST	7566	19-APR-81	3000	-	RESEARCH	DALLAS
20	7566	JONES	MANAGER	7839	02-APR-81	2975	-	RESEARCH	DALLAS
20	7369	SMITH	CLERK	7902	17-DEC-80	800	-	RESEARCH	DALLAS
20	7876	ADAMS	CLERK	7788	23-MAY-87	1100	-	RESEARCH	DALLAS
30	7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	SALES	CHICAGO
30	7844	TURNER	SALESMAN	7698	08-SEP-81	1500	0	SALES	CHICAGO
30	7900	JAMES	CLERK	7682	03-DEC-81	950	-	SALES	CHICAGO
30	7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	SALES	CHICAGO
30	7698	BLAKE	MANAGER	7839	01-MAY-81	2850	-	SALES	CHICAGO

Equi Join: When a theta join uses only equivalence condition, it becomes an equi join.

Syntax:

3. SELECT column list 4. FROM table1, table2....
5. WHERE table1.column_name =
6. table2.column_name;

select empno,ename,sal,Emp_ Aman.deptno,Dept_ Aman 1.deptno,dame

from Emp_ Aman,Dept_ Aman 1

where Emp_ Aman.deptno=Dept_ Aman 1.deptno;

EMPNO	ENAME	SAL	DEPTNO	DEPTNO	DAME
7839	KING	5000	10	10	ACCOUNTING
7698	BLAKE	2850	30	30	SALES
7782	CLARK	2450	10	10	ACCOUNTING
7566	JONES	2975	20	20	RESEARCH
7788	SCOTT	3000	20	20	RESEARCH
7902	FORD	3000	20	20	RESEARCH
7369	SMITH	800	20	20	RESEARCH
7521	WARD	1250	30	30	SALES
7654	MARTIN	1250	30	30	SALES
7844	TURNER	1500	30	30	SALES
7876	ADAMS	1100	20	20	RESEARCH
7900	JAMES	950	30	30	SALES
7934	MILLER	1300	10	10	ACCOUNTING

Outer Join: In an outer join, along with tuples that satisfy the matching criteria, we also include some or all tuples that do not match the criteria.

Right join: RIGHT JOIN returns all the values from the values from the rows of right table and the matched values from the left table. If there is no matching in both tables, it will return NULL.

Syntax:

1. SELECT table1.column1, table1.column2, table2.column1....
2. FROM table1
3. RIGHT JOIN table2
4. ON table1.matching_column = table2.matching_column;

select empno,ename,sal,Emp_ Aman.deptno,Dept_ Aman

1.deptno,dame from Emp_ Aman right outer join Dept_ Aman 1 on

Emp_ Aman.deptno=Dept_ Aman 1.deptno;

EMPNO	ENAME	SAL	DEPTNO	DEPTNO	DAME
7839	KING	5000	10	10	ACCOUNTING
7698	BLAKE	2850	30	30	SALES
7782	CLARK	2450	10	10	ACCOUNTING
7566	JONES	2975	20	20	RESEARCH
7788	SCOTT	3000	20	20	RESEARCH
7902	FORD	3000	20	20	RESEARCH
7369	SMITH	800	20	20	RESEARCH
7521	WARD	1250	30	30	SALES
7654	MARTIN	1250	30	30	SALES
7844	TURNER	1500	30	30	SALES
7876	ADAMS	1100	20	20	RESEARCH
7900	JAMES	950	30	30	SALES
7934	MILLER	1300	10	10	ACCOUNTING
-	-	-	-	40	OPERATIONS

Left join: left join returns all the values from left table and the matching values from the right table. If there is no matching join value, it will return NULL.

Syntax:

1. SELECT table1.column1, table1.column2, table2.column1....
2. FROM table1
3. LEFT JOIN table2
4. ON table1.matching_column = table2.matching_column;

select empno,ename,sal,Emp_ Aman.deptno,Dept_ Aman

1.deptno,dame from Dept_ Aman 1 right outer join Emp_ Aman on

Emp_ Aman.deptno=Dept_ Aman 1.deptno;

EMPNO	ENAME	SAL	DEPTNO	DEPTNO	DAME
7839	KING	5000	10	10	ACCOUNTING
7782	CLARK	2450	10	10	ACCOUNTING
7934	MILLER	1300	10	10	ACCOUNTING
7566	JONES	2975	20	20	RESEARCH
7788	SCOTT	3000	20	20	RESEARCH
7902	FORD	3000	20	20	RESEARCH
7369	SMITH	800	20	20	RESEARCH
7876	ADAMS	1100	20	20	RESEARCH
7698	BLAKE	2850	30	30	SALES
7521	WARD	1250	30	30	SALES
7654	MARTIN	1250	30	30	SALES
7844	TURNER	1500	30	30	SALES
7900	JAMES	950	30	30	SALES

Full join: Full outer join is like a left or right join except that it contains all rows from both tables.

Syntax:

1. SELECT table1.column1, table1.column2, table2.column1,
2. FROM table1
3. FULL JOIN table2
4. ON table1.matching_column = table2.matching_column;

select ename,job,sal,mgr,dame,Dept_ Aman

1.deptno from Emp_ Aman right outer join Dept_

Aman 1 on Emp_ Aman.deptno=Dept_ Aman

1.deptno;

ENAME	JOB	SAL	MGR	DAME	DEPTNO
KING	PRESIDENT	5000	-	ACCOUNTING	10
BLAKE	MANAGER	2850	7839	SALES	30
CLARK	MANAGER	2450	7839	ACCOUNTING	10
JONES	MANAGER	2975	7839	RESEARCH	20
SCOTT	ANALYST	3000	7566	RESEARCH	20
FORD	ANALYST	3000	7566	RESEARCH	20
SMITH	CLERK	800	7902	RESEARCH	20
WARD	SALESMAN	1250	7698	SALES	30
MARTIN	SALESMAN	1250	7698	SALES	30
TURNER	SALESMAN	1500	7698	SALES	30
ADAMS	CLERK	1100	7788	RESEARCH	20
JAMES	CLERK	950	7682	SALES	30
MILLER	CLERK	1300	7782	ACCOUNTING	10
-	-	-	-	OPERATIONS	40

Cross join: CROSS JOIN specifies that all rows from first table join with all of the rows of second table.

Syntax:

1. SELECT *
2. FROM table1
3. CROSS JOIN table2;

select * from Emp_Aman cross join Dept_Aman;

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO	DEPTNO	DAME	LOC
7839	KING	PRESIDENT	-	17-NOV-81	5000	-	10	10	ACCOUNTING	NEWYORK
7698	BLAKE	MANAGER	7839	01-MAY-81	2850	-	30	10	ACCOUNTING	NEWYORK
7782	CLARK	MANAGER	7839	09-JUN-81	2450	-	10	10	ACCOUNTING	NEWYORK
7566	JONES	MANAGER	7839	02-APR-81	2975	-	20	10	ACCOUNTING	NEWYORK
7788	SCOTT	ANALYST	7566	19-APR-81	3000	-	20	10	ACCOUNTING	NEWYORK
7902	FORD	ANALYST	7566	03-DEC-81	3000	-	20	10	ACCOUNTING	NEWYORK
7369	SMITH	CLERK	7902	17-DEC-80	800	-	20	10	ACCOUNTING	NEWYORK
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30	10	ACCOUNTING	NEWYORK
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30	10	ACCOUNTING	NEWYORK
7844	TURNER	SALESMAN	7698	08-SEP-81	1500	0	30	10	ACCOUNTING	NEWYORK
7876	ADAMS	CLERK	7788	23-MAY-87	1100	-	20	10	ACCOUNTING	NEWYORK
7900	JAMES	CLERK	7682	03-DEC-81	950	-	30	10	ACCOUNTING	NEWYORK
7934	MILLER	CLERK	7782	23-JAN-82	1300	-	10	10	ACCOUNTING	NEWYORK
7839	KING	PRESIDENT	-	17-NOV-81	5000	-	10	20	RESEARCH	DALLAS
7698	BLAKE	MANAGER	7839	01-MAY-81	2850	-	30	20	RESEARCH	DALLAS
7782	CLARK	MANAGER	7839	09-JUN-81	2450	-	10	20	RESEARCH	DALLAS
7566	JONES	MANAGER	7839	02-APR-81	2975	-	20	20	RESEARCH	DALLAS
7788	SCOTT	ANALYST	7566	19-APR-81	3000	-	20	20	RESEARCH	DALLAS
7902	FORD	ANALYST	7566	03-DEC-81	3000	-	20	20	RESEARCH	DALLAS
7369	SMITH	CLERK	7902	17-DEC-80	800	-	20	20	RESEARCH	DALLAS
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30	20	RESEARCH	DALLAS
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30	20	RESEARCH	DALLAS
7844	TURNER	SALESMAN	7698	08-SEP-81	1500	0	30	20	RESEARCH	DALLAS
7876	ADAMS	CLERK	7788	23-MAY-87	1100	-	20	20	RESEARCH	DALLAS
7900	JAMES	CLERK	7682	03-DEC-81	950	-	30	20	RESEARCH	DALLAS
7934	MILLER	CLERK	7782	23-JAN-82	1300	-	10	20	RESEARCH	DALLAS
7839	KING	PRESIDENT	-	17-NOV-81	5000	-	10	30	SALES	CHICAGO
7698	BLAKE	MANAGER	7839	01-MAY-81	2850	-	30	30	SALES	CHICAGO

7566	JONES	MANAGER	7839	02-APR-81	2975	-	20	30	SALES	CHICAGO
7788	SCOTT	ANALYST	7566	19-APR-81	3000	-	20	30	SALES	CHICAGO
7902	FORD	ANALYST	7566	03-DEC-81	3000	-	20	30	SALES	CHICAGO
7369	SMITH	CLERK	7902	17-DEC-80	800	-	20	30	SALES	CHICAGO
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30	30	SALES	CHICAGO
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30	30	SALES	CHICAGO
7844	TURNER	SALESMAN	7698	08-SEP-81	1500	0	30	30	SALES	CHICAGO
7876	ADAMS	CLERK	7788	23-MAY-87	1100	-	20	30	SALES	CHICAGO
7900	JAMES	CLERK	7682	03-DEC-81	950	-	30	30	SALES	CHICAGO
7934	MILLER	CLERK	7782	23-JAN-82	1300	-	10	30	SALES	CHICAGO
7839	KING	PRESIDENT	-	17-NOV-81	5000	-	10	40	OPERATIONS	BOSTON
7698	BLAKE	MANAGER	7839	01-MAY-81	2850	-	30	40	OPERATIONS	BOSTON
7782	CLARK	MANAGER	7839	09-JUN-81	2450	-	10	40	OPERATIONS	BOSTON
7566	JONES	MANAGER	7839	02-APR-81	2975	-	20	40	OPERATIONS	BOSTON
7788	SCOTT	ANALYST	7566	19-APR-81	3000	-	20	40	OPERATIONS	BOSTON

7902	FORD	ANALYST	7566	03-DEC-81	3000	-	20	40	OPERATIONS	BOSTON
7369	SMITH	CLERK	7902	17-DEC-80	800	-	20	40	OPERATIONS	BOSTON
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30	40	OPERATIONS	BOSTON
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30	40	OPERATIONS	BOSTON
7844	TURNER	SALESMAN	7698	08-SEP-81	1500	0	30	40	OPERATIONS	BOSTON
7876	ADAMS	CLERK	7788	23-MAY-87	1100	-	20	40	OPERATIONS	BOSTON

Download CSV

Self-join: A self-join is a join in which a table is joined with itself. To join a table itself means that each row of the table is combined with itself and with every other row of the table.

Syntax:

1. SELECT a. column_name, b.column_name...
2. FROM table1 a, table1 b
3. WHERE a.common_filed = b.common_field;

select e2.ename employee,e1.ename manager from Emp_ Aman e1,Emp_ Aman e2 where e1.empno=e2.mgr;

EMPLOYEE	MANAGER
BLAKE	KING
CLARK	KING
JONES	KING
SCOTT	JONES
FORD	JONES
SMITH	FORD
WARD	BLAKE
MARTIN	BLAKE
TURNER	BLAKE
ADAMS	SCOTT
MILLER	CLARK

PRACTICAL NO:- 6

A) Numeric function :-

1) Absolute :-

select abs(-9) from dual;

ABS(-6)
6

Download CSV

select abs(6) from dual;

ABS(6)
6

Download CSV

2) **Ceil :-**

select ceil(14) from dual;

CEIL(14)
14

Download CSV

select ceil(12.69) from dual;

CEIL(12.69)
13

Download CSV

3) **Floor:-**

select floor(12.17) from dual;

FLOOR(12.17)
12

Download CSV

select floor(12) from dual;

FLOOR(12)
12

4) **Round**

select round(12.18) from dual;

ROUND(12.18)
12

Download CSV

select round(12.56) from dual;

ROUND(12.56)
13

Download CSV

5) **Remainder:-**

select remainder(13,3) from dual;

REMAINDER(13,3)
1

6) **Square root:-**

select sqrt(25)from dual;

SQRT(25)
5

Download CSV

7) **Modules :-**

select mod(25,4) from dual;

MOD(25,4)
1

8) **Power :-**

select power(6,6) from dual;

POWER(6,6)
46656

Download CSV

9) **Trunc :-**

select trunc(1.2765,2) from dual;

TRUNC(1.2765,2)
1.27

Download CSV

10) **Exponential :-**

select exp(2) from dual;

EXP(2)
7.389056098930650227230427460575007813 2

Download CSV

11) **Log :-**

select log(10,100) from dual;

LOG(10,100)
2

B) Character function :-

1) **Lower :-**

select lower('WELCOME') from dual;

LOWER('WELCOME')
welcome

Download CSV

2) **Upper :-**

select upper('welcome') from dual;

UPPER('WELCOME')
WELCOME

3) **Initcap:-**

select initcap('Have a good day') from dual;

INITCAP('HAVEAGOODDAY')
Have A Good Day

4) **Length** :-

select length('Have a good day') from dual;

LENGTH('HAVEAGOODDAY')
15

5) **Substr :-**

select substr('Have a nice day',4,10) from dual;

SUBSTR('HAVEANICEDAY',4,10)
e a nice d

Download CSV

6) **Concat :-**

select concat('Have a',' nice day') from dual;

CONCAT('HAVEA','NICEDAY')
Have a nice day

Download CSV

7) **Instr :-**

select instr('Welcome to my world','my') from dual;

INSTR('WELCOMETOMYWORLD','MY')
12

Download CSV

8) **Trim :-**

select trim(leading '1' from '1123424') from dual;

TRIM(LEADING'1'FROM'1123424')
23424

select trim(trailing '4' from '1123424') from dual;

TRIM(TRAILING'4'FROM'1123424')
112342

select rtrim('12345432167', '7') from dual;

RTRIM('12345432167','7')
1234543216

select ltrim('1123352' ,'1') from dual;

LTRIM('1123352','1')
23352

Download CSV

9) **Translate :-**

select translate('12931893451234561','14','$%') from dual;

TRANSLATE('12931893451234561','14','$%')
$293$893%5$23%56$

10) **Replace:-**

select replace('12931893451234561','31','$') from dual;

REPLACE('12931893451234561','31','$')
129$893451234561

11) **Rpad:-**

select rpad('wel',10,'$')from dual;

RPAD('WEL',10,'$')
wel$$$$$$$

12) **Lpad:-**

select lpad('wel',7,'$')from dual;

LPAD('WEL',7,'$')
$$$$wel

Download CSV

C) **Date :-**

1) **Sysdate:-**

select sysdate from dual;

SYSDATE
22-FEB-21

2) **Next_Day:-**

select next_day(sysdate,'monday') from dual;

NEXT_DAY(SYSDATE,'MONDAY')
01-MAR-21

3) **Last_Day:-**

select last_day(sysdate) from dual;

LAST_DAY(SYSDATE)
28-FEB-21

4) **Add_months:-**

select add_months(sysdate,-1) from dual;

ADD_MONTHS(SYSDATE,-1)
22-JAN-21

5) **Months_Between:-**

select months_between('22-dec-21',sysdate) from dual;

MONTHS_BETWEEN('22-DEC-21',SYSDATE)
10

Download CSV

select months_between(sysdate,'31-dec-2021') from dual;

MONTHS_BETWEEN(SYSDATE,'31-DEC-2021')
-10.270219534050179211469534050179211469 53

6) **Systimestamp:-**

select systimestamp from dual;

SYSTIMESTAMP
22-FEB-21 02.52.40.145146 PM +00:00

7) **Current_date:-**

select current_date from dual;

CURRENT_DATE
22-FEB-21

Download CSV

Practical 7: Study of various types of SET OPERATORS

Suppose that a Product table contains two attributes, PROD_CODE and VEND_CODE. The values for the PROD_CODE are: ABC, DEF, GHI and JKL. These are matched by the following values for the VEND_CODE: 125, 124, 124 and 123, respectively (e.g., PROD_CODE value ABC corresponds to VEND_CODE value

125). The Vendor table contains a single attribute, VEND_CODE, with values 123, 124, 125 and 126. (The

VEND_CODE attribute in the Product table is a foreign key to the VEND_CODE in the Vendor table.)

create table vendor(VEND_CODE int primary key);

```
Table created.
```

create table product(PROD_CODE varchar(3),VEND_CODE int,foreign key(VEND_CODE)references vendor(VEND_CODE));

```
Table created.
```

insert into vendor values('123');

```
1 row(s) inserted.
```

insert into vendor values('124');

```
1 row(s) inserted.
```

insert into vendor values('125');

```
1 row(s) inserted.
```

insert into vendor values('126');

```
1 row(s) inserted.
```

insert into product values('ABC',125);

```
1 row(s) inserted.
```

insert into product values('DEF',124);

```
1 row(s) inserted.
```

insert into product values('GHI',124);

```
1 row(s) inserted.
```

insert into product values('JKL',123);

```
1 row(s) inserted.
```

Given the information, what would be the query output for the following? Show values.

select * from vendor;

VEND_CODE
123
124
125
126

select * from product;

PROD_CODE	VEND_CODE
ABC	125
DEF	124
GHI	124
JKL	123

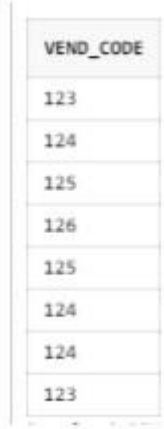
a) A UNION query based on these two tables
select VEND_CODE from vendor
union

select VEND_CODE from product;

VEND_CODE
123
124
125
126

b) A UNION ALL query based on these two tables

```
select VEND_CODE from vendor
union all select
VEND_CODE from product;
```



VEND_CODE
123
124
125
126
125
124
124
123

c) An INTERSECT query based on these two tables

```
select VEND_CODE from vendor
intersect
select VEND_CODE from product;
```

VEND_CODE
123
124
125

d) A MINUS query based on these two tables

```
select VEND_CODE from vendor
minus
select VEND_CODE from product;
```

VEND_CODE
126

PRACTICAL 8: Study of various types of views

Considering Emp and Dept table, perform the following:

EMP Table:

select * from Emp_Aman;

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7839	KING	PRESIDENT	-	17-NOV-81	5000	-	10
7698	BLAKE	MANAGER	7839	01-MAY-81	2850	-	30
7782	CLARK	MANAGER	7839	09-JUN-81	2450	-	10
7566	JONES	MANAGER	7839	02-APR-81	2975	-	20
7788	SCOTT	ANALYST	7566	19-APR-81	3000	-	20
7902	FORD	ANALYST	7566	03-DEC-81	3000	-	20
7369	SMITH	CLERK	7902	17-DEC-80	800	-	20
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30
7844	TURNER	SALESMAN	7698	08-SEP-81	1500	0	30
7876	ADAMS	CLERK	7788	23-MAY-87	1100	-	20
7900	JAMES	CLERK	7682	03-DEC-81	950	-	30
7934	MILLER	CLERK	7782	23-JAN-82	1300	-	10

DEPT Table:

Select * from Dept_Aman1;

DEPTNO	DAME	LOC
10	ACCOUNTING	NEWYORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

Download CSV

1. Create a view named emp_hor with the job titled as ‘ANALYST’.

```
create view Emp_A
as
select * from Emp_Aman
where job='ANALYST';
```

```
View created.
```

```
select * from Emp_A;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7788	SCOTT	ANALYST	7566	19-APR-81	3000	-	20
7902	FORD	ANALYST	7566	03-DEC-81	3000	-	20

Download CSV

2. Create a view named vwemp specifying name of employees, job and their salary.

```
create view vwemp
as
select ENAME,JOB,SAL from Emp_Aman;
```

```
View created.
```

```
select * from vwemp;
```

ENAME	JOB	SAL
KING	PRESIDENT	5000
BLAKE	MANAGER	2850
CLARK	MANAGER	2450
JONES	MANAGER	2975
SCOTT	ANALYST	3000
FORD	ANALYST	3000
SMITH	CLERK	800
WARD	SALESMAN	1250
MARTIN	SALESMAN	1250
TURNER	SALESMAN	1500
ADAMS	CLERK	1100
JAMES	CLERK	950
MILLER	CLERK	1300

3. Create a view displaying total salary on the basis of the jobs.

```
create view hemp(job,sal)
as
select job,sum(SAL) from Emp_Aman
group by job;
```

```
View created.
```

```
select * from hemp;
```

JOB	SAL
ANALYST	6000
CLERK	4150
SALESMAN	4000
MANAGER	8275
PRESIDENT	5000

Download CSV

4. Create a view with contains name of employee, dept and the location of the employees.

```
create view AG
as
select ENAME,Emp_Aman.deptno,loc
from Emp_Aman inner join Dept_Aman1
on Emp_Aman.deptno=Dept_Aman1.deptno;
```

```
View created.
```

```
select * from AG;
```

ENAME	DEPTNO	LOC
KING	10	NEWYORK
BLAKE	30	CHICAGO
CLARK	10	NEWYORK
JONES	20	DALLAS
SCOTT	20	DALLAS
FORD	20	DALLAS
SMITH	20	DALLAS
WARD	30	CHICAGO
MARTIN	30	CHICAGO
TURNER	30	CHICAGO
ADAMS	20	DALLAS
JAMES	30	CHICAGO
MILLER	10	NEWYORK

5. Create a view to display the name of the employees with their salary and job who belongs to department 20.

```
create view Aman12
as
select ENAME,SAL,JOB from Emp_Aman
where deptno=20;
```

```
View created.
```

```
select * from Aman12;
```

ENAME	SAL	JOB
JONES	2975	MANAGER
SCOTT	3000	ANALYST
FORD	3000	ANALYST
SMITH	800	CLERK
ADAMS	1100	CLERK

Download CSV

6. Delete all the views created above

drop view Emp_A;

```
View dropped.
```

drop view vwemp;

```
View dropped.
```

drop view hemp;

```
View dropped.
```

drop view AG;

```
View dropped.
```

drop view Aman12;

```
View dropped.
```

Practical 9: Study of subqueries with all its clauses

1. Display the employee name whose salary is greater than the salary of employee 7566.

   select * from Emp_Aman where sal>(select sal from Emp_Aman where empno = 7566);

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7839	KING	PRESIDENT	-	17-NOV-81	5000	-	10
7788	SCOTT	ANALYST	7566	19-APR-81	3000	-	20
7902	FORD	ANALYST	7566	03-DEC-81	3000	-	20

Download CSV

2. Display the employee name, sal, job of the employee whose job is similar to the employee 7369.

   select ename,sal,job from Emp_Aman where job = (select job

   from Emp_Aman where empno=7369);

ENAME	SAL	JOB
SMITH	800	CLERK
ADAMS	1100	CLERK
JAMES	950	CLERK
MILLER	1300	CLERK

Download CSV

3. Display the employee name with the salary less than any salary of job type CLERK.

   select ename,sal from Emp_Aman where sal < any (select sal

   from Emp_Aman where job='CLERK');

ENAME	SAL
SMITH	800
JAMES	950
ADAMS	1100
MARTIN	1250
WARD	1250

Download CSV

4. Display the employee name, salary, department id, job id for those employees who works in the same designation as the employee works whose id is 7900.

```
select ename,sal,empno,job from Emp_Aman where job =
(select job from Emp_Aman where empno=7900);
```

ENAME	SAL	EMPNO	JOB
SMITH	800	7369	CLERK
ADAMS	1100	7876	CLERK
JAMES	950	7900	CLERK
MILLER	1300	7934	CLERK

Download CSV

5. Display the detail of department whose manager Encode='7698'.

```
select *from Emp_Aman inner join Dept_Aman1 on
Emp_Aman.deptno = Dept_Aman1.deptno where mgr in(select mgr from
Emp_Aman where mgr=7698);
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO	DEPTNO	DAME	LOC
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30	30	SALES	CHICAGO
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30	30	SALES	CHICAGO
7844	TURNER	SALESMAN	7698	08-SEP-81	1500	0	30	30	SALES	CHICAGO

Download CSV

6. Display the employees whose salary is greater than any MANAGER.

```
select *from Emp_Aman where sal> any (select sal from
Emp_Aman where job ='MANAGER');
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7839	KING	PRESIDENT	-	17-NOV-81	5000	-	10
7902	FORD	ANALYST	7566	03-DEC-81	3000	-	20
7788	SCOTT	ANALYST	7566	19-APR-81	3000	-	20
7566	JONES	MANAGER	7839	02-APR-81	2975	-	20
7698	BLAKE	MANAGER	7839	01-MAY-81	2850	-	30

Download CSV