



**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./Ms. _AKSHAT CHUDASAMA_____

Roll No: 13_____ Programme: BSc IT/CS 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/CS 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: AKSHAT CHUDASAMA

Roll No: 13

Sr. No.	DATE	TITLE	SIGN
1.	28/1/21	Study of Data Definition Language Statement	
2.	4/2/21	Study of Data Manipulation Language Statement	
3.	12/2/21	Study of SELECT Statement.	
4.	18/2/21	Draw ER diagram for given scenario/project/case study	
5.	4/3/21	Study of various type of JOINS	
6.	18/3/21	Study of different functions	
7.	18/3/21	Study of various types of SET OPERATORS	
8.	25/3/21	Study of various types of views	
9.	25/3/21	Study of subqueries with all its clauses	
10.	8/3/21	Study of Transaction (Commit/ Rollback), Locks	
11.	8/3/21	Implementing deadlocks	

PRECTICAL 1 :-
AKSHAT CHUDASAMA – 13 FYCS

Write the query for the following.

1. Create the following table and include the necessary constraints NOT NULL, DEFAULT, CHECK, PRIMARY KEY, UNIQUE.
 - a. Student (sId,sname,gender,dob,marks,class,email)

```
SQL> create table student(sid int primary key, sname varchar(10) not null, gender varchar(10) not null, dob date not null, marks int check(marks>50), class varchar(10) default 'FYCS', emailid varchar(10));
create table student(sid int primary key, sname varchar(10) not null, gender varchar(10) not null, dob date not null, marks int check(marks>50), class varchar(10) default 'FYCS', emailid varchar(10))
*
```

ERROR at line 1:
ORA-00955: name is already used by an existing object

```
SQL> desc student
```

Name	Null?	Type
SID	NOT NULL	NUMBER(38)
SNAME	NOT NULL	VARCHAR2(10)
DOB		DATE
MARKS		NUMBER(38)
CLASS		VARCHAR2(10)
EMAILID		VARCHAR2(10)

```
SQL>
```

- b. course(cId,cname,credits)

```
SQL> create table course(cid int primary key,cname varchar(10) not null,credits int not null);
```

Table created.

```
SQL> desc course
```

Name	Null?	Type
CID	NOT NULL	NUMBER(38)
CNAME	NOT NULL	VARCHAR2(10)
CREDITS	NOT NULL	NUMBER(38)

```
SQL>
```

2) Alter the structure of the course table

- c. Modify data type of cname

```
SQL> alter table course
2 modify cname varchar(20);
```

Table altered.

```
SQL> desc course
```

Name	Null?	Type
CID	NOT NULL	NUMBER(38)
CNAME	NOT NULL	VARCHAR2(20)
CREDITS	NOT NULL	NUMBER(38)

- d. Add a column coursehours with minimum course hours greater than 45.

```
SQL> alter table course
2 add coursehours int check(coursehours>45);
```

Table altered.

```
SQL> desc course
```

Name	Null?	Type
CID	NOT NULL	NUMBER(38)
CNAME	NOT NULL	VARCHAR2(20)
CREDITS	NOT NULL	NUMBER(38)
COURSEHOURS		NUMBER(38)

- e. Add a column cdesc

```
SQL> alter table course
2 add cdesc varchar(10);
```

Table altered.

```
SQL> desc course
```

Name	Null?	Type
CID	NOT NULL	NUMBER(38)
CNAME	NOT NULL	VARCHAR2(20)
CREDITS	NOT NULL	NUMBER(38)
COURSEHOURS		NUMBER(38)
CDESC		VARCHAR2(10)

3) Alter the structure of the student table

- f. Add column age with minimum age as 17

```
SQL> alter table student
2 add age int check(age>17);
```

Table altered.

```
SQL> desc student
```

Name	Null?	Type
SID	NOT NULL	NUMBER(38)
SNAME	NOT NULL	VARCHAR2(10)
DOB		DATE
MARKS		NUMBER(38)
CLASS		VARCHAR2(10)
EMAILID		VARCHAR2(10)
AGE		NUMBER(38)

g. Delete column dob

```
SQL> alter table student
2 drop column dob;
```

Table altered.

```
SQL> desc student
```

Name	Null?	Type
-----	-----	-----
SID	NOT NULL	NUMBER(38)
SNAME	NOT NULL	VARCHAR2(10)
MARKS		NUMBER(38)
CLASS		VARCHAR2(10)
EMAILID		VARCHAR2(10)
AGE		NUMBER(38)

h. Add a column phoneno

```
SQL> alter table student
2 add phoneno int;
```

Table altered.

```
SQL> desc student
```

Name	Null?	Type
-----	-----	-----
SID	NOT NULL	NUMBER(38)
SNAME	NOT NULL	VARCHAR2(10)
MARKS		NUMBER(38)
CLASS		VARCHAR2(10)
EMAILID		VARCHAR2(10)
AGE		NUMBER(38)
PHONENO		NUMBER(38)

i. Rename phoneno to contactno

```
SQL> alter table student
2 rename column phoneno to contactno;
```

Table altered.

```
SQL> desc student
```

Name	Null?	Type
-----	-----	-----
SID	NOT NULL	NUMBER(38)
SNAME	NOT NULL	VARCHAR2(10)
MARKS		NUMBER(38)
CLASS		VARCHAR2(10)
EMAILID		VARCHAR2(10)
AGE		NUMBER(38)
CONTACTNO		NUMBER(38)

4) Rename student table as Student_details

```
SQL> alter table student  
2 rename to student_details;
```

Table altered.

```
SQL> desc student_details
```

Name	Null?	Type
SID	NOT NULL	NUMBER(38)
SNAME	NOT NULL	VARCHAR2(10)
MARKS		NUMBER(38)
CLASS		VARCHAR2(10)
EMAILID		VARCHAR2(10)
AGE		NUMBER(38)
CONTACTNO		NUMBER(38)

6) Drop the table student_details and course.

```
SQL> drop table course;
```

Table dropped.

```
SQL> drop table student_details;
```

Table dropped.

```
SQL> desc course
```

ERROR:

ORA-04043: object course does not exist

```
SQL> desc student_details
```

ERROR:

ORA-04043: object student_details does not exist

- 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)

```
SQL> create table employee(Emp_no int primary key, E_name varchar(10) not null, E_address varchar(20),E_ph_no int, dept_
on int not null, Dept_name varchar(10), Job_id int, salary int);

Table created.

SQL> desc employee

```

Name	Null?	Type
EMP_NO	NOT NULL	NUMBER(38)
E_NAME	NOT NULL	VARCHAR2(10)
E_ADDRESS		VARCHAR2(20)
E_PH_NO		NUMBER(38)
DEPT_ON	NOT NULL	NUMBER(38)
DEPT_NAME		VARCHAR2(10)
JOB_ID		NUMBER(38)
SALARY		NUMBER(38)

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

```
SQL> alter table employee
2 add hiredate date;

Table altered.

SQL> desc employee

```

Name	Null?	Type
EMP_NO	NOT NULL	NUMBER(38)
E_NAME	NOT NULL	VARCHAR2(10)
E_ADDRESS		VARCHAR2(20)
E_PH_NO		NUMBER(38)
DEPT_ON	NOT NULL	NUMBER(38)
DEPT_NAME		VARCHAR2(10)
JOB_ID		NUMBER(38)
SALARY		NUMBER(38)
HIREDATE		DATE

alter

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

```
SQL> alter table employee
2 modify Job_id varchar(20);

Table altered.

SQL> desc employee

```

Name	Null?	Type
EMP_NO	NOT NULL	NUMBER(38)
E_NAME	NOT NULL	VARCHAR2(10)
E_ADDRESS		VARCHAR2(20)
E_PH_NO		NUMBER(38)
DEPT_ON	NOT NULL	NUMBER(38)
DEPT_NAME		VARCHAR2(10)
JOB_ID		VARCHAR2(20)
SALARY		NUMBER(38)
HIREDATE		DATE

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

```
SQL> alter table employee
  2  rename column Emp_no to E_on;
```

Table altered.

```
SQL> desc employee
```

Name	Null?	Type
E_ON	NOT NULL	NUMBER(38)
E_NAME	NOT NULL	VARCHAR2(10)
E_ADDRESS		VARCHAR2(20)
E_PH_NO		NUMBER(38)
DEPT_ON	NOT NULL	NUMBER(38)
DEPT_NAME		VARCHAR2(10)
JOB_ID		VARCHAR2(20)
SALARY		NUMBER(38)
HIREDATE		DATE

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

```
SQL> alter table employee
  2  modify Job_id varchar(10);
```

Table altered.

```
SQL> desc employee
```

Name	Null?	Type
E_ON	NOT NULL	NUMBER(38)
E_NAME	NOT NULL	VARCHAR2(10)
E_ADDRESS		VARCHAR2(20)
E_PH_NO		NUMBER(38)
DEPT_ON	NOT NULL	NUMBER(38)
DEPT_NAME		VARCHAR2(10)
JOB_ID		VARCHAR2(10)
SALARY		NUMBER(38)
HIREDATE		DATE

C. Create the following tables with specified attributes and constraints

1. Department Table: Department_Id varchar2(20) primary key, Department_Name varchar2(25) with required data.

```
SQL> create table Department(Department_Id varchar(20) primary key, Department_Name varchar(25));
Table created.

SQL> alter table Department
  2 modify Department_Name varchar(25) not null;
Table altered.

SQL> desc Department
Name                                     Null?      Type
-----
DEPARTMENT_ID                          NOT NULL   VARCHAR2(20)
DEPARTMENT_NAME                         NOT NULL   VARCHAR2(25)
```

2. 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'.

```
SQL> create table Instructor(Instructor_id varchar(20) primary key ,Department_Id varchar(20) references Department(Department_Id),Last_name varchar(20),First_name varchar(200) not null, Telephone varchar(20) unique,gender char(1) check(gender='F'or gender='M'),city varchar(10) default 'MUMBAI');
```

Table created.

```
SQL> desc Instructor
```

Name	Null?	Type
INSTRUCTOR_ID	NOT NULL	VARCHAR2(20)
DEPARTMENT_ID		VARCHAR2(20)
LAST_NAME		VARCHAR2(20)
FIRST_NAME	NOT NULL	VARCHAR2(200)
TELEPHONE		VARCHAR2(20)
GENDER		CHAR(1)
CITY		VARCHAR2(10)

D) 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	-	-	-	-	✓

Table Name: DEPT

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

```
SQL> create table akshat_DEPT (Dept_no int primary key, Dname varchar(14) not null, loc varchar(13));
Table created.

SQL> desc akshat_DEPT
Name                               Null?    Type
-----
DEPT_NO                            NOT NULL NUMBER(38)
DNAME                              NOT NULL VARCHAR2(14)
LOC                                VARCHAR2(13)

SQL> create table akshat_EMP(EMP_no int primary key, Ename varchar(10) not null, Job varchar(9), MGR int, Hiredate date, SAL
decimal (7,2), Comm int, Dept_no int references AKSHAT_DEPT(Dept_no));
Table created.

SQL> desc akshat_EMP
Name                               Null?    Type
-----
EMP_NO                             NOT NULL NUMBER(38)
ENAME                              NOT NULL VARCHAR2(10)
JOB                                VARCHAR2(9)
MGR                                NUMBER(38)
HIREDATE                           DATE
SAL                                NUMBER(7,2)
COMM                               NUMBER(38)
DEPT_NO                            NUMBER(38)
```

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

DEPT TABLE

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

```
SQL> insert into akshat_DEPT values(10,'ACCOUNTING','NEW YORK');  
1 row created.
```

```
SQL> insert into akshat_DEPT values(20,'RESEARCH','DALLAS');  
1 row created.
```

```
SQL> insert into akshat_DEPT values(30,'SALES','CHICAGO');  
1 row created.
```

```
SQL> insert into akshat_DEPT values(40,'OPERATIONS','BOSTION');  
1 row created.
```

```
SQL> select * form akshat_DEPT;  
select * form akshat_DEPT  
      *
```

```
ERROR at line 1:  
ORA-00923: FROM keyword not found where expected
```

```
SQL> select * from akshat_DEPT;
```

DEPT_NO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTION

```
SQL> insert into akshat_EMP values(7934,'MILLER','CLERK',7782,'23-JAN-82',1300,NULL,10);
```

```
1 row created.
```

```
SQL> select * from akshat_EMP;
```

EMP_NO	ENAME	JOB	MGR	HIREDATE	SAL	COMM
7839	KING	PRESIDENT		17-NOV-81	5000	
7698	BLAKE	MANAGER	7839	01-MAY-81	2850	
7782	CLARK	MANAGER	7839	09-JUN-81	2450	

EMP_NO	ENAME	JOB	MGR	HIREDATE	SAL	COMM
7566	JONES	MANAGER	7839	02-APR-81	2975	
7788	SCOTT	ANALYST	7566	19-APR-87	3000	
7902	FORD	ANALYST	7566	03-DEC-81	3000	

EMP_NO	ENAME	JOB	MGR	HIREDATE	SAL	COMM
7369	SMITH	CLERK	7902	17-DEC-80	800	
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500

EMP_NO	ENAME	JOB	MGR	HIREDATE	SAL	COMM
DEPT_NO						
7654 30	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400
7844 30	TURNER	SALESMAN	7698	08-SEP-81	1500	0
7876 20	ADAMS	CLERK	7788	23-MAY-87	1100	
EMP_NO	ENAME	JOB	MGR	HIREDATE	SAL	COMM
DEPT_NO						
7900 30	JAMES	CLERK	7698	03-DEC-81	950	
7934 10	MILLER	CLERK	7782	23-JAN-82	1300	

14 rows selected.

B) Update and Delete Queries

1) Update the salary of employees working as CLERK by 500.

```
SQL> update akshat_EMP
2  set SAL=SAL+500
3  where Job='CLERK';

4 rows updated.
```

EMP_NO	ENAME	JOB	MGR	HIREDATE	SAL	COMM
7369	SMITH	CLERK	7902	17-DEC-80	1300	
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400
7844	TURNER	SALESMAN	7698	08-SEP-81	1500	0
7876	ADAMS	CLERK	7788	23-MAY-87	1600	
7900	JAMES	CLERK	7698	03-DEC-81	1450	
7934	MILLER	CLERK	7782	23-JAN-82	1800	

14 rows selected.

2) Update the manager of James as CLARK.

```
SQL> update akshat_EMP
2 set job='CLEARK'
3 where Ename='JAMES';
```

1 row updated.

SQL>

EMP_NO	ENAME	JOB	MGR	HIREDATE	SAL	COMM
DEPT_NO						
7900	JAMES	CLEAR	7698	03-DEC-81	1450	
30						

3) Change the role of Miller as MANAGER.

```
SQL> update akshat_EMP
2 set Job='MANAGER'
3 where Ename='MILLER';
```

1 row updated.

7934	MILLER	MANAGER	7782	23-JAN-82	1800	
10						

4) Delete the records of Manager

```
SQL> delete from akshat_EMP
2 where Job='MANAGER';
```

4 rows deleted.


```
SQL> select * from akshat_EMP;
```

EMP_NO	ENAME	JOB	MGR	HIREDATE	SAL	COMM

DEPT_NO						

7839 10	KING	PRESIDENT		17-NOV-81	5000	
7788 20	SCOTT	ANALYST	7566	19-APR-87	3000	
7902 20	FORD	ANALYST	7566	03-DEC-81	3000	
EMP_NO	ENAME	JOB	MGR	HIREDATE	SAL	COMM

DEPT_NO						

7369 20	SMITH	CLERK	7902	17-DEC-80	1300	
7499 30	ALLEN	SALESMAN	7698	20-FEB-81	1600	300
7521 30	WARD	SALESMAN	7698	22-FEB-81	1250	500
EMP_NO	ENAME	JOB	MGR	HIREDATE	SAL	COMM

DEPT_NO						

7654 30	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400
7844 30	TURNER	SALESMAN	7698	08-SEP-81	1500	0
7876 20	ADAMS	CLERK	7788	23-MAY-87	1600	
EMP_NO	ENAME	JOB	MGR	HIREDATE	SAL	COMM

DEPT_NO						

7900 30	JAMES	CLEARC	7698	03-DEC-81	1450	

```
10 rows selected.
```

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

```
SQL> delete from akshat_emp  
2 where SAL>1000;  
  
10 rows deleted.  
  
SQL> select * from akshat_EMP;  
  
no rows selected
```

Name: AKSHAT CHUDASAMA

Roll no: 13

Class: FYCS

PRACTICAL 3

A) Using emp table, perform the following queries:

1) Display the details of all employees.

```
SQL> set linesize 10000
SQL> set pagesize 10000
SQL> select * from akshat_EMP;
```

EMP_NO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPT_NO
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

14 rows selected.

2) Display the name and job for all employees.

```
SQL> select Ename,Job from akshat_EMP;
```

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

14 rows selected.

3) Display name and salary for all employees.

```
SQL> select Ename,SAL from akshat_EMP;
```

ENAME	SAL
KING	5000
BLAKE	2850
CLARK	2450
JONES	2975
SCOTT	3000
FORD	3000
SMITH	800
ALLEN	1600
WARD	1250
MARTIN	1250
TURNER	1500
ADAMS	1100
JAMES	950
MILLER	1300

14 rows selected.

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

```
SQL> select * from akshat_EMP
2  where SAL>2000;
```

EMP_NO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPT_NO
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

6 rows selected.

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

```
SQL> select * from akshat_EMP
2  where Job='MANAGER';
```

EMP_NO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPT_NO
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

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

```
SQL> select * from akshat_EMP
2  where Dept_no=10;
```

EMP_NO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPT_NO
7839	KING	PRESIDENT		17-NOV-81	5000		10
7782	CLARK	MANAGER	7839	09-JUN-81	2450		10
7934	MILLER	CLERK	7782	23-JAN-82	1300		10

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

```
SQL> select * from akshat_EMP
2  where Job='CLERK' and SAL>3000;

no rows selected
```

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

```
SQL> select Emp_no,Ename,comm from akshat_EMP
2  where comm>0;
```

EMP_NO	ENAME	COMM
7499	ALLEN	300
7521	WARD	500
7654	MARTIN	1400

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

```
SQL> select Emp_no,Ename,comm from akshat_EMP
2  where comm is null;
```

EMP_NO	ENAME	COMM
7839	KING	
7698	BLAKE	
7782	CLARK	
7566	JONES	
7788	SCOTT	
7902	FORD	
7369	SMITH	
7876	ADAMS	
7900	JAMES	
7934	MILLER	

10 rows selected.

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

```
SQL> select Ename from akshat_EMP
  2  where Job in('CLERK','SALESMAN','ANALYST')and SAL>2000;

ENAME
-----
SCOTT
FORD
```

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

```
SQL> select Ename from akshat_EMP
  2  where Job in('CLERK','SALESMAN','ANALYST');

ENAME
-----
SCOTT
FORD
SMITH
ALLEN
WARD
MARTIN
TURNER
ADAMS
JAMES
MILLER

10 rows selected.
```

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

```
SQL> select Ename from akshat_EMP
2  where Dept_no in(10,20,30);

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

14 rows selected.
```

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

```
SQL> select * from akshat_EMP
2  where SAL between 1000 and 2000;

  EMP_NO ENAME      JOB          MGR HIREDATE          SAL      COMM      DEPT_NO
-----
  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         0          20
  7934 MILLER     CLERK         7782 23-JAN-82          1300         0          10

6 rows selected.
```

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

```
SQL> select * from akshat_EMP
2 order by SAL ASC;
```

EMP_NO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPT_NO
7369	SMITH	CLERK	7902	17-DEC-80	800		20
7900	JAMES	CLERK	7698	03-DEC-81	950		30
7876	ADAMS	CLERK	7788	23-MAY-87	1100		20
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30
7934	MILLER	CLERK	7782	23-JAN-82	1300		10
7844	TURNER	SALESMAN	7698	08-SEP-81	1500	0	30
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30
7782	CLARK	MANAGER	7839	09-JUN-81	2450		10
7698	BLAKE	MANAGER	7839	01-MAY-81	2850		30
7566	JONES	MANAGER	7839	02-APR-81	2975		20
7902	FORD	ANALYST	7566	03-DEC-81	3000		20
7788	SCOTT	ANALYST	7566	19-APR-87	3000		20
7839	KING	PRESIDENT		17-NOV-81	5000		10

14 rows selected.

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

```
SQL> select Emp_no,Ename,SAL from akshat_EMP
2 where MGR=7369;
```

no rows selected

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

```
SQL> select * from akshat_EMP
2 where Job='CLERK' or Job='ANALYST'
3 order by Job desc;
```

EMP_NO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPT_NO
7369	SMITH	CLERK	7902	17-DEC-80	800		20
7900	JAMES	CLERK	7698	03-DEC-81	950		30
7934	MILLER	CLERK	7782	23-JAN-82	1300		10
7876	ADAMS	CLERK	7788	23-MAY-87	1100		20
7902	FORD	ANALYST	7566	03-DEC-81	3000		20
7788	SCOTT	ANALYST	7566	19-APR-87	3000		20

6 rows selected.

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

```
SQL> select * from akshat_EMP
2  where Dept_no in(10,20);
```

EMP_NO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPT_NO
7839	KING	PRESIDENT		17-NOV-81	5000		10
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
7876	ADAMS	CLERK	7788	23-MAY-87	1100		20
7934	MILLER	CLERK	7782	23-JAN-82	1300		10

8 rows selected.

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

```
SQL> select * from akshat_EMP
2  where Ename like '%LL%';
```

EMP_NO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPT_NO
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30
7934	MILLER	CLERK	7782	23-JAN-82	1300		10

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

```
SQL> select * from akshat_EMP
2  order by Ename ASC;
```

EMP_NO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPT_NO
7876	ADAMS	CLERK	7788	23-MAY-87	1100		20
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30
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	7698	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-87	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

14 rows selected.

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

```
SQL> select * from akshat_EMP
2  order by Ename DESC;
```

EMP_NO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPT_NO
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-87	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	7698	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
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30
7876	ADAMS	CLERK	7788	23-MAY-87	1100		20

14 rows selected.

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

```
SQL> select * from akshat_EMP
2  where Dept_no not in 20;
```

EMP_NO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPT_NO
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
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
7900	JAMES	CLERK	7698	03-DEC-81	950		30
7934	MILLER	CLERK	7782	23-JAN-82	1300		10

9 rows selected.

22) List all the employees except PRESIDENT and MANAGER.

```
SQL> select * from akshat_EMP
2  where Job not in( 'PRESIDENT','MANAGER');
```

EMP_NO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPT_NO
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

10 rows selected.

23) List the employees whose name starts with A.

```
SQL> select * from akshat_EMP
2  where Ename like 'A%';
```

EMP_NO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPT_NO
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30
7876	ADAMS	CLERK	7788	23-MAY-87	1100		20

24) List all the Clerks of Deptno 20.

```
SQL> select * from akshat_EMP
2  where Job='CLERK' and Dept_no=20;
```

EMP_NO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPT_NO
7369	SMITH	CLERK	7902	17-DEC-80	800		20
7876	ADAMS	CLERK	7788	23-MAY-87	1100		20

25) List the employees whose names ends with S.

```
SQL> select * from akshat_EMP
2  where Ename like '%S';
```

EMP_NO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPT_NO
7566	JONES	MANAGER	7839	02-APR-81	2975		20
7876	ADAMS	CLERK	7788	23-MAY-87	1100		20
7900	JAMES	CLERK	7698	03-DEC-81	950		30

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

```
SQL> select * from akshat_EMP
2  where Ename like '____';
```

EMP_NO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPT_NO
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

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

```
SQL> select * from akshat_EMP
2  where Job='MANAGER' and Dept_no=10;
```

EMP_NO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPT_NO
7782	CLARK	MANAGER	7839	09-JUN-81	2450		10

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

```
SQL> select sum(SAL)
  2  from akshat_EMP
  3  where Job='ANALYST';

SUM(SAL)
-----
      6000
```

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

```
SQL> select MIN(SAL),MAX(SAL),AVG(SAL) from akshat_EMP;

MIN(SAL)  MAX(SAL)  AVG(SAL)
-----
      800      5000 2073.21429
```

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

```
SQL> select Dept_no, count(*)
  2  from akshat_EMP
  3  group by Dept_no;

DEPT_NO  COUNT(*)
-----
      30         6
      20         5
      10         3
```

B) Answer the following queries:

1) Display the total salary of employees department wise.

```
SQL> select Dept_no, sum(SAL) from akshat_EMP
2 group by Dept_no;
```

DEPT_NO	SUM(SAL)
30	9400
20	10875
10	8750

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

```
SQL> select Job, sum(SAL)
2 from akshat_EMP
3 group by Job
4 order by Job ASC;
```

JOB	SUM(SAL)
ANALYST	6000
CLERK	4150
MANAGER	8275
PRESIDENT	5000
SALESMAN	5600

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

```
SQL> select Job, count(*)
2 from akshat_EMP
3 group by Job;
```

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

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

```
SQL> select Dept_no, count(*)
2 from akshat_EMP
3 group by Dept_no;
```

DEPT_NO	COUNT(*)
30	6
20	5
10	3

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

```
SQL> select Job,Dept_no,sum(SAL)
2  from akshat_EMP
3  group by Job,Dept_no
4  order by Job;
```

JOB	DEPT_NO	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	5600

9 rows selected.

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

```
SQL> select sum(SAL),count(Job) from akshat_EMP
2  group by Job
3  having count(Job)>1;
```

SUM(SAL)	COUNT(JOB)
4150	4
5600	4
8275	3
6000	2

7) Display unique jobs of employees.

```
SQL> select distinct Job
2  from akshat_EMP;
```

JOB

CLERK
SALESMAN
PRESIDENT
MANAGER
ANALYST

Name: AKSHAT CHUDASAMA

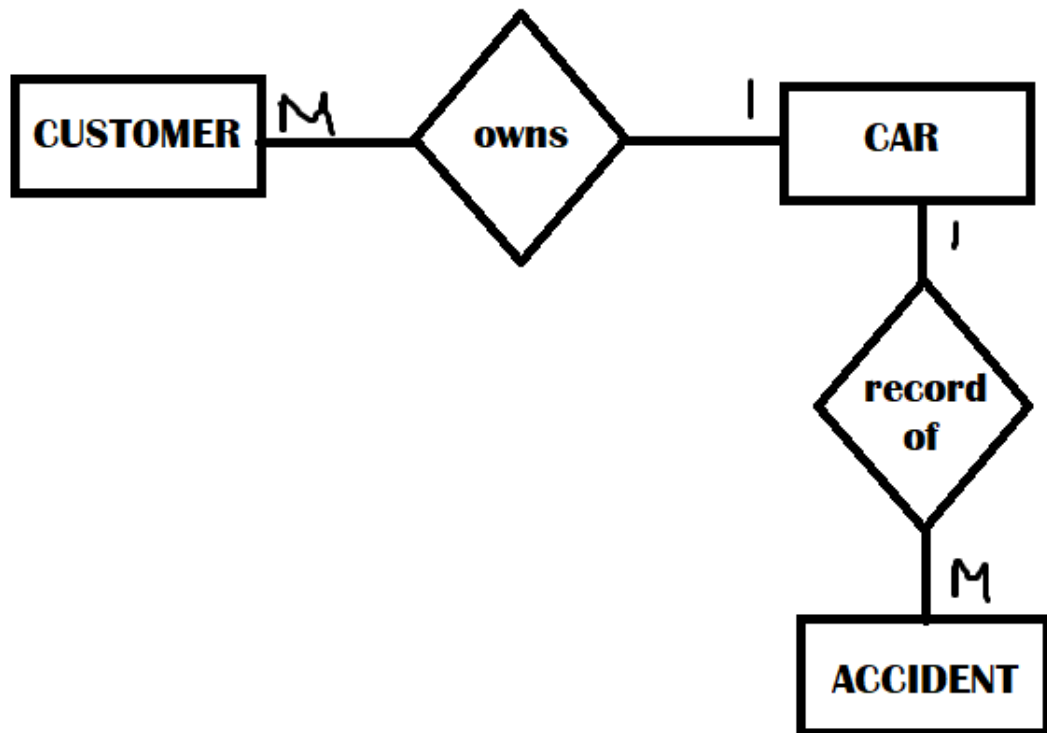
Roll no: 13

FYCS

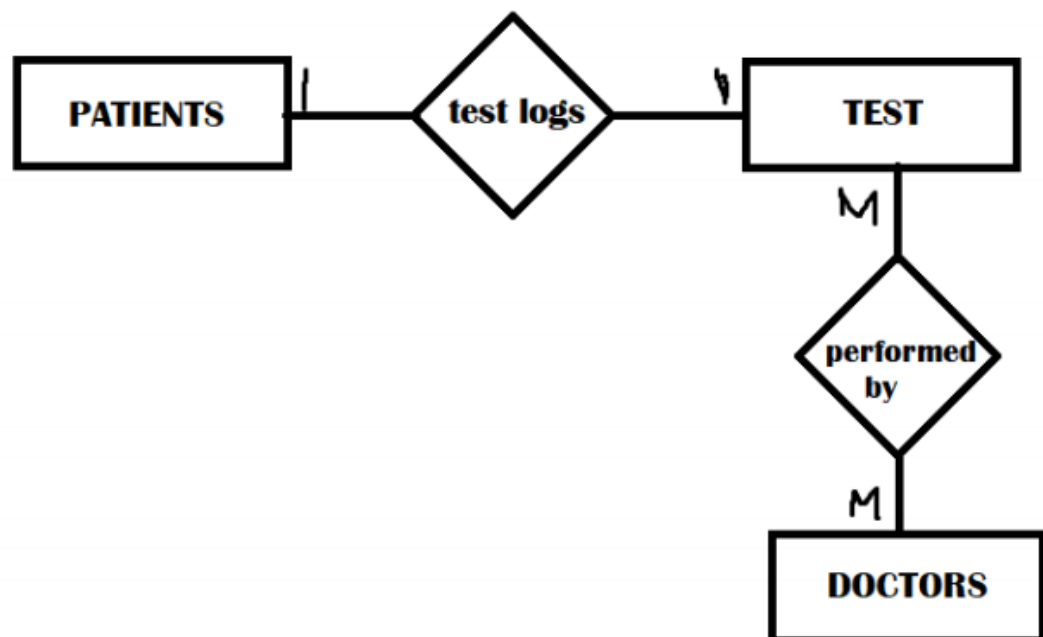
PRACTICAL 4

Questions on ER diagram

1. Construct an E-R diagram for a car-insurance company whose customers own one or more cars each. Each car has associated with it zero to any number of recorded accidents.



2. Construct an E-R diagram for a hospital with a set of patients and a set of medical doctors. Associate with each patient a log of the various tests and examinations conducted.



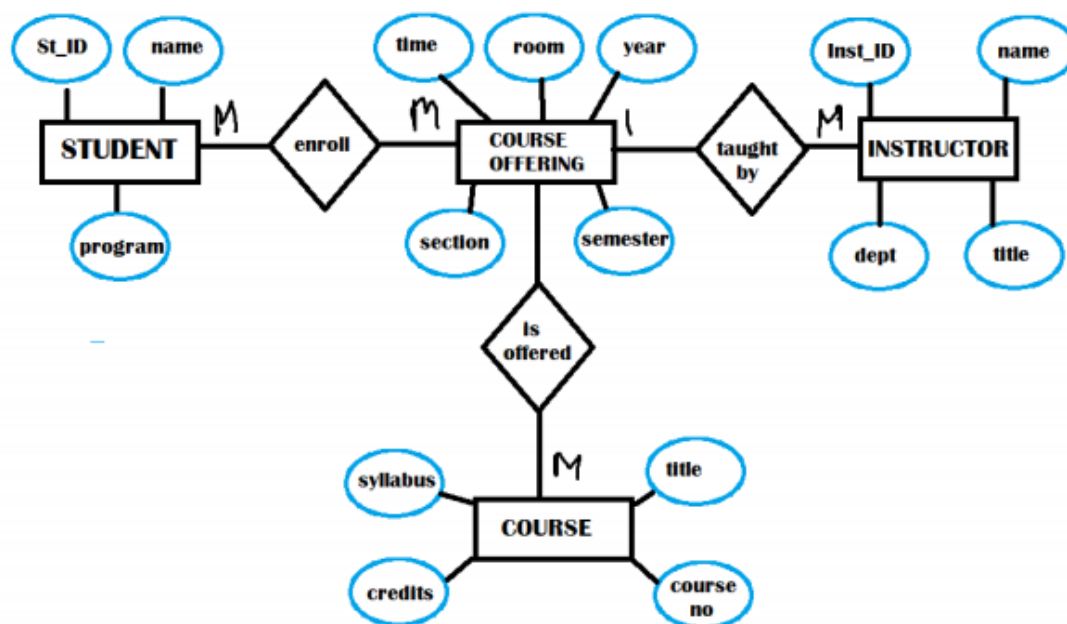
3. A university registrar.s of_ce maintains data about the following entities:

(a) courses, including number, title, credits, syllabus, and prerequisites;

(b) course offerings, including course number, year, semester, section number, instructor(s), timings, and classroom;

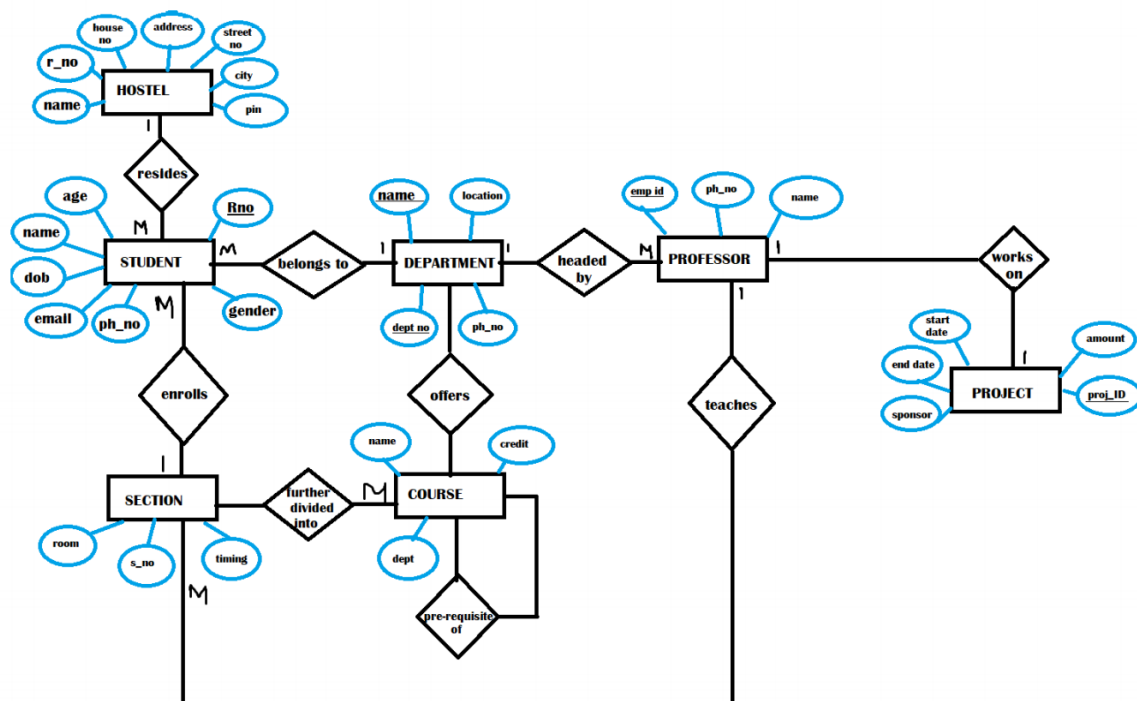
(c) students, including student-id, name, and program; and

(d) instructors, including identi_cation number, name, department, and title. Further, the enrollment of students in courses and grades awarded to students in each course they are enrolled for must be appropriately modeled. Construct an E-R diagram for the registrar.s of_ce. Document all assumptions that you make about the mapping constraints.

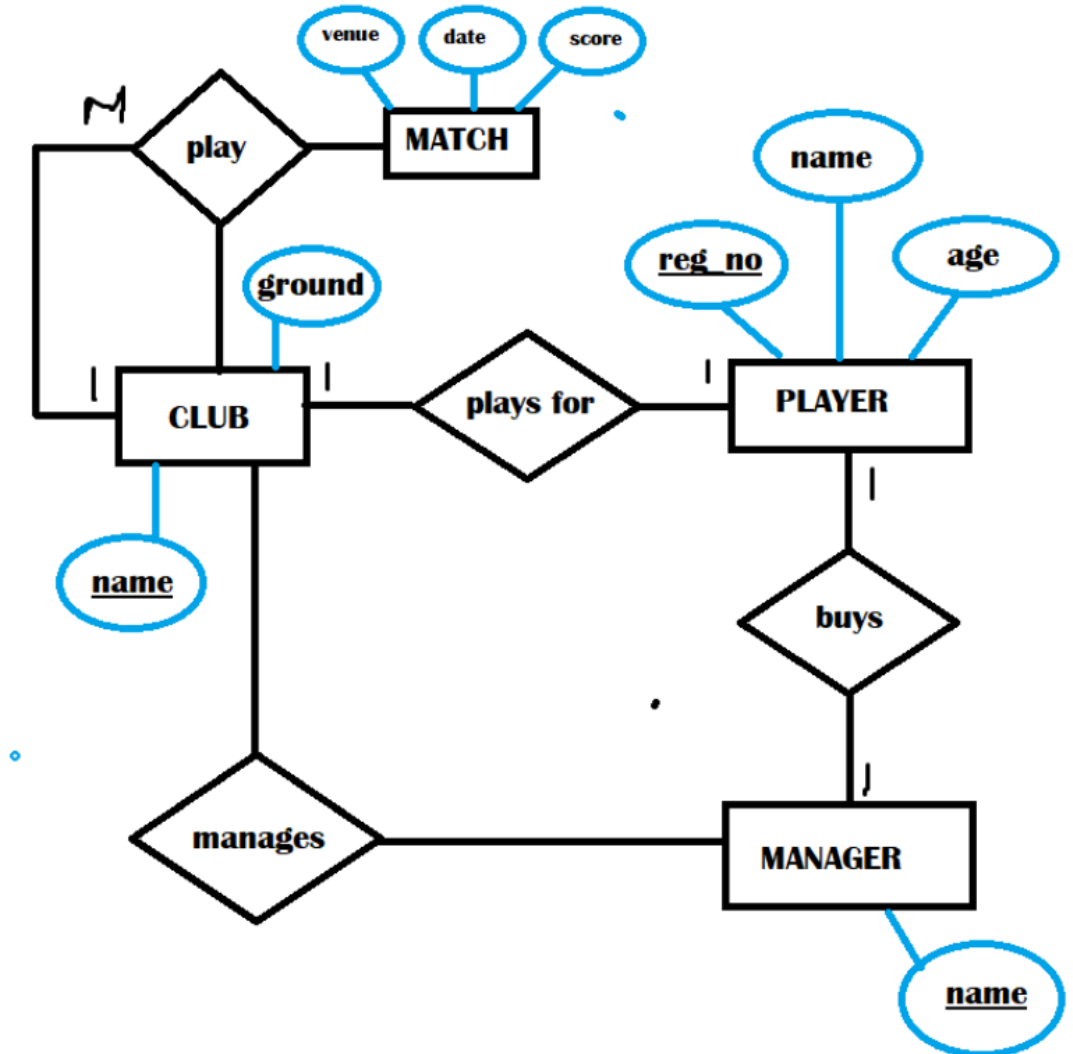


4. Draw the ER diagram for the given scenario

- In an educational institute, there are several departments and students belong to one of them. Each department has a unique department number, a unique name, a location, a phone number and is headed by a professor.
- Professors have a unique employee id, name, phoneno. We like to keep track of the following details regarding students: name, unique roll no, gender, phone number, date of birth, age and one or more email addresses.
- Students have a local address consisting of the hostel name and the room number. They also have home address consisting of house number, street, city and pin. It is assumed that all students reside in the hostels.
- A course taught in a semester of the year is called a section. There can be several sections of the same course in a semester; these are identified by the section number. Each section is taught by a different professor and has its own timings and a room to meet.
- Students enroll for several sections in a semester. Each course has a name, number of credits and the department that offers it. A course may have other courses as pre-requisites i.e courses to be completed before it can be enrolled in.
- Professors also undertake research projects. These are sponsored by funding agencies and have a specific start date, end date and amount of money given. More than one professor can be involved in a project. Also a professor may be simultaneously working on several projects. A project has a unique projectid.



- 5 .Draw the ER diagram for the given scenario “A football club has a name and a ground and is made up of players. A player can play for only one club and a manager, represented by his name, manages a club. A footballer has a registration number, name and age. A club manager also buys players. Each club play against each other club in the league and matches have a date, venue and score.



Name: AKSHAT CHUDASAMA

Roll No: 13

FYCS

Practical No:5

JOINS

INNER:

- The INNER JOIN keyword selects all rows from both the tables as long as the condition satisfies.
- SYNTAX SELECT table1.column1,table1.column2,table2.column1,.... FROM table1 INNER JOIN table2 ON table1.matching_column = table2.matching_column;

```
SQL> select Ename,SAL,Job,Dname,Loc
  2  from akshat_EMP inner join akshat_Dept
  3  on akshat_EMP.Dept_no=akshat_Dept.Dept_no;
```

ENAME	SAL	JOB	DNAME	LOC
KING	5000	PRESIDENT	ACCOUNTING	NEW YORK
BLAKE	2850	MANAGER	SALES	CHICAGO
CLARK	2450	MANAGER	ACCOUNTING	NEW YORK
JONES	2975	MANAGER	RESEARCH	DALLAS
SCOTT	3000	ANALYST	RESEARCH	DALLAS
FORD	3000	ANALYST	RESEARCH	DALLAS
SMITH	800	CLERK	RESEARCH	DALLAS
ALLEN	1600	SALESMAN	SALES	CHICAGO
WARD	1250	SALESMAN	SALES	CHICAGO
MARTIN	1250	SALESMAN	SALES	CHICAGO
TURNER	1500	SALESMAN	SALES	CHICAGO
ADAMS	1100	CLERK	RESEARCH	DALLAS
JAMES	950	CLERK	SALES	CHICAGO
MILLER	1300	CLERK	ACCOUNTING	NEW YORK

14 rows selected.

NATURAL:

- A natural join is a type of equi join which occurs implicitly by comparing all the same names columns in both tables. The join result has only one column for each pair of equally named columns.
- SYNTAX Select * From table1 natural join table2;

```
SQL> select * from akshat_EMP natural join akshat_Dept;
```

DEPT_NO	EMP_NO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DNAME	LOC
10	7839	KING	PRESIDENT		17-NOV-81	5000		ACCOUNTING	NEW YORK
30	7698	BLAKE	MANAGER	7839	01-MAY-81	2850		SALES	CHICAGO
10	7782	CLARK	MANAGER	7839	09-JUN-81	2450		ACCOUNTING	NEW YORK
20	7566	JONES	MANAGER	7839	02-APR-81	2975		RESEARCH	DALLAS
20	7788	SCOTT	ANALYST	7566	19-APR-87	3000		RESEARCH	DALLAS
20	7902	FORD	ANALYST	7566	03-DEC-81	3000		RESEARCH	DALLAS
20	7369	SMITH	CLERK	7902	17-DEC-80	800		RESEARCH	DALLAS
30	7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	SALES	CHICAGO
30	7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	SALES	CHICAGO
30	7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	SALES	CHICAGO
30	7844	TURNER	SALESMAN	7698	08-SEP-81	1500	0	SALES	CHICAGO
20	7876	ADAMS	CLERK	7788	23-MAY-87	1100		RESEARCH	DALLAS
30	7900	JAMES	CLERK	7698	03-DEC-81	950		SALES	CHICAGO
10	7934	MILLER	CLERK	7782	23-JAN-82	1300		ACCOUNTING	NEW YORK

14 rows selected.

OUTER:

- In an outer join, unmatched rows in one or both tables can be returned. Right: returns only unmatched rows from the right table
- SYNTAX SELECT table1.column1,table1.column2,table2.column1,.... FROM table1 RIGHT [OUTER] JOIN table2 ON table1.matching_column = table2.matching_column;

```
SQL> select EMP_no,Ename,SAL,akshat_EMP.Dept_no,akshat_Dept.Dept_no,Dname
2  from akshat_EMP right outer join akshat_Dept
3  on akshat_EMP.Dept_no=akshat_Dept.Dept_no;
```

EMP_NO	ENAME	SAL	DEPT_NO	DEPT_NO	DNAME
7782	CLARK	2450	10	10	ACCOUNTING
7934	MILLER	1300	10	10	ACCOUNTING
7839	KING	5000	10	10	ACCOUNTING
7902	FORD	3000	20	20	RESEARCH
7788	SCOTT	3000	20	20	RESEARCH
7566	JONES	2975	20	20	RESEARCH
7369	SMITH	800	20	20	RESEARCH
7876	ADAMS	1100	20	20	RESEARCH
7521	WARD	1250	30	30	SALES
7654	MARTIN	1250	30	30	SALES
7844	TURNER	1500	30	30	SALES
7900	JAMES	950	30	30	SALES
7499	ALLEN	1600	30	30	SALES
7698	BLAKE	2850	30	30	SALES
				40	OPERATIONS

15 rows selected.

Left: returns only unmatched rows from the left table.

- SYNTAX SELECT table1.column1,table1.column2,table2.column1,.... FROM table1 LEFT [OUTER] JOIN table2 ON table1.matching_column = table2.matching_column;

```
SQL> select EMP_no,ENAME,SAL,akshat_EMP.Dept_no,akshat_Dept.Dept_no,Dname
 2  from akshat_EMP left outer join akshat_Dept
 3  on akshat_EMP.Dept_no=akshat_Dept.Dept_no;
```

EMP_NO	ENAME	SAL	DEPT_NO	DEPT_NO	DNAME
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
7499	ALLEN	1600	30	30	SALES
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

14 rows selected.

Full: returns unmatched rows from both tables

- SYNTAX SELECT table1.column1,table1.column2,table2.column1,.... FROM table1 FULL [OUTER] JOIN table2 ON table1.matching_column = table2.matching_column;

```
SQL> select EMP_no,Ename,SAL,akshat_EMP.Dept_no,akshat_Dept.Dept_no,Dname
2  from akshat_EMP full outer join akshat_Dept
3  on akshat_EMP.Dept_no=akshat_Dept.Dept_no;
```

EMP_NO	ENAME	SAL	DEPT_NO	DEPT_NO	DNAME
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
7499	ALLEN	1600	30	30	SALES
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

15 rows selected.

CROSS:

- The CARTESIAN JOIN is also known as CROSS JOIN.
- In a CARTESIAN JOIN there is a join for each row of one table to every row of another table. This usually happens when the matching column or WHERE condition is not specified.
- SYNTAX SELECT * FROM TABLE1, TABLE2;

```
SQL> select * from akshat_EMP, akshat_Dept;
```

EMP_NO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPT_NO	DEPT_NO	DNAME	LOC
7839	KING	PRESIDENT		17-NOV-81	5000		10	20	RESEARCH	DALLAS
7839	KING	PRESIDENT		17-NOV-81	5000		10	30	SALES	CHICAGO
7839	KING	PRESIDENT		17-NOV-81	5000		10	10	ACCOUNTING	NEW YORK
7839	KING	PRESIDENT		17-NOV-81	5000		10	40	OPERATIONS	BOSTON
7698	BLAKE	MANAGER	7839	01-MAY-81	2850		30	20	RESEARCH	DALLAS
7698	BLAKE	MANAGER	7839	01-MAY-81	2850		30	30	SALES	CHICAGO
7698	BLAKE	MANAGER	7839	01-MAY-81	2850		30	10	ACCOUNTING	NEW YORK
7698	BLAKE	MANAGER	7839	01-MAY-81	2850		30	40	OPERATIONS	BOSTON
7782	CLARK	MANAGER	7839	09-JUN-81	2450		10	20	RESEARCH	DALLAS
7782	CLARK	MANAGER	7839	09-JUN-81	2450		10	30	SALES	CHICAGO
7782	CLARK	MANAGER	7839	09-JUN-81	2450		10	10	ACCOUNTING	NEW YORK
7782	CLARK	MANAGER	7839	09-JUN-81	2450		10	40	OPERATIONS	BOSTON
7566	JONES	MANAGER	7839	02-APR-81	2975		20	20	RESEARCH	DALLAS
7566	JONES	MANAGER	7839	02-APR-81	2975		20	30	SALES	CHICAGO
7566	JONES	MANAGER	7839	02-APR-81	2975		20	10	ACCOUNTING	NEW YORK
7566	JONES	MANAGER	7839	02-APR-81	2975		20	40	OPERATIONS	BOSTON
7788	SCOTT	ANALYST	7566	19-APR-87	3000		20	20	RESEARCH	DALLAS
7788	SCOTT	ANALYST	7566	19-APR-87	3000		20	30	SALES	CHICAGO
7788	SCOTT	ANALYST	7566	19-APR-87	3000		20	10	ACCOUNTING	NEW YORK
7788	SCOTT	ANALYST	7566	19-APR-87	3000		20	40	OPERATIONS	BOSTON
7902	FORD	ANALYST	7566	03-DEC-81	3000		20	20	RESEARCH	DALLAS
7902	FORD	ANALYST	7566	03-DEC-81	3000		20	30	SALES	CHICAGO
7902	FORD	ANALYST	7566	03-DEC-81	3000		20	10	ACCOUNTING	NEW YORK
7902	FORD	ANALYST	7566	03-DEC-81	3000		20	40	OPERATIONS	BOSTON
7369	SMITH	CLERK	7902	17-DEC-80	800		20	20	RESEARCH	DALLAS
7369	SMITH	CLERK	7902	17-DEC-80	800		20	30	SALES	CHICAGO
7369	SMITH	CLERK	7902	17-DEC-80	800		20	10	ACCOUNTING	NEW YORK
7369	SMITH	CLERK	7902	17-DEC-80	800		20	40	OPERATIONS	BOSTON
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30	20	RESEARCH	DALLAS
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30	30	SALES	CHICAGO
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30	10	ACCOUNTING	NEW YORK
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30	40	OPERATIONS	BOSTON
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30	20	RESEARCH	DALLAS
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30	30	SALES	CHICAGO
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30	10	ACCOUNTING	NEW YORK
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30	40	OPERATIONS	BOSTON
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30	20	RESEARCH	DALLAS
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30	30	SALES	CHICAGO
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30	10	ACCOUNTING	NEW YORK
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30	40	OPERATIONS	BOSTON
7844	TURNER	SALESMAN	7698	08-SEP-81	1500	0	30	20	RESEARCH	DALLAS
7844	TURNER	SALESMAN	7698	08-SEP-81	1500	0	30	30	SALES	CHICAGO
7844	TURNER	SALESMAN	7698	08-SEP-81	1500	0	30	10	ACCOUNTING	NEW YORK
7844	TURNER	SALESMAN	7698	08-SEP-81	1500	0	30	40	OPERATIONS	BOSTON
7876	ADAMS	CLERK	7788	23-MAY-87	1100		20	20	RESEARCH	DALLAS
7876	ADAMS	CLERK	7788	23-MAY-87	1100		20	30	SALES	CHICAGO
7876	ADAMS	CLERK	7788	23-MAY-87	1100		20	10	ACCOUNTING	NEW YORK
7876	ADAMS	CLERK	7788	23-MAY-87	1100		20	40	OPERATIONS	BOSTON
7900	JAMES	CLERK	7698	03-DEC-81	950		30	20	RESEARCH	DALLAS
7900	JAMES	CLERK	7698	03-DEC-81	950		30	30	SALES	CHICAGO
7900	JAMES	CLERK	7698	03-DEC-81	950		30	10	ACCOUNTING	NEW YORK
7900	JAMES	CLERK	7698	03-DEC-81	950		30	40	OPERATIONS	BOSTON
7934	MILLER	CLERK	7782	23-JAN-82	1300		10	20	RESEARCH	DALLAS
7934	MILLER	CLERK	7782	23-JAN-82	1300		10	30	SALES	CHICAGO
7934	MILLER	CLERK	7782	23-JAN-82	1300		10	10	ACCOUNTING	NEW YORK
7934	MILLER	CLERK	7782	23-JAN-82	1300		10	40	OPERATIONS	BOSTON

56 rows selected.

SELF:

- As the name signifies, in SELF JOIN a table is joined to itself. That is, each row of the table is joined with itself and all other rows depending on some conditions

- SYNTAX SELECT a.column 1 , b.column2 FROM table_name a, table_name b WHERE some_condition;

```
SQL> select e2.ename employee,e1.ename manager
2   from akshat_EMP e1,akshat_EMP e2
3   where e1.EMP_no=e2.mgr;
```

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

13 rows selected.

NAME : AKSHAT CHUDASAMA

ROLL NO. : 13

FYCS

PRACTICAL NO.6

Numeric Functions

Absolute(Abs):

```
SQL> select abs (-4) from dual;

  ABS(-4)
-----
        4
```

Ceil:

```
SQL> select ceil (12.30) from dual;

CEIL(12.30)
-----
        13
```

Floor:

```
SQL> select floor (12.30) from dual;

FLOOR(12.30)
-----
        12
```

SQRT:

```
SQL> select sqrt (3397) from dual;

SQRT(3397)
-----
58.2837885
```

MOD:

```
SQL> select mod (-34,3) from dual;

MOD(-34,3)
-----
       -1
```

ROUND:

```
SQL> select round (12.30) from dual;

ROUND(12.30)
-----
          12
```

```
SQL> select round (10.379,2) from dual;

ROUND(10.379,2)
-----
         10.38
```

```
SQL> select round (10.379,-2) from dual;

ROUND(10.379,-2)
-----
           0
```

REMAINDER:

```
SQL> select remainder (23,2) from dual;

REMAINDER(23,2)
-----
          -1
```

POWER:

```
SQL> select power (27,3) from dual;

POWER(27,3)
-----
        19683
```

TRUNC:

```
SQL> select trunc (123.231,2) from dual;

TRUNC(123.231,2)
-----
         123.23
```

EXP:

```
SQL> select exp (22) from dual;

EXP(22)
-----
3584912846
```

Character Function

lower:

```
SQL> select lower ('AKSHAT') from dual;

LOWER(
-----
akshat
```

Upper:

```
SQL> select upper ('AKSHAT') from dual;

UPPER(
-----
AKSHAT
```

Initcap:

```
SQL> select initcap ('I AM AKSHAT') from dual;

INITCAP('IA
-----
I Am Akshat
```

Length:

```
SQL> select length ('hello i am Akshat') from dual;

LENGTH('HELLOIAMAKSHAT')
-----
17
```

Substr:

```
SQL> select substr (' hello i am student',6,15) from dual ;

SUBSTR('HELLOI
-----
o i am student
```

Concat:

```
SQL> select concat('hello everybody',' all good') from dual;

CONCAT('HELLOEVERYBODY',
-----
hello everybody all good
```

Instr:

```
SQL> select instr ('welcome my friend','m') from dual;

INSTR('WELCOMEMYFRIEND','M')
-----
                             6
```

Trim:

```
SQL> select trim ( both 'o' from 'good 939') from dual;

TRIM(BOTH
-----
good 939
```

Rtrim:

```
SQL> select rtrim ('hfd gdhg 939','3') from dual;

RTRIM('HFDGD
-----
hfd gdhg 939
```

Ltrim:

```
SQL> select ltrim('hfd gdhg 9390000000','3') from dual;

LTRIM('HFDGDHG939000
-----
hfd gdhg 9390000000
```

TRANSLATE:

```
SQL> select translate('123hublot','23','#$') from dual;

TRANSLATE
-----
1#$hublot
```

RPAD:

```
SQL> select rpad('well',10,'$') from dual;

RPAD('WELL
-----
well$$$$$$
```

LPAD:

```
SQL> select lpad('hello',4,'hii') from dual;

LPAD
----
hell
```

DATA FUNCTION:

SYSDATE:

```
SQL> select sysdate from dual
2   ;

SYSDATE
-----
27-FEB-21
```

NEXT_DAY:

```
SQL> select next_day('24-feb-2021','sunday') "next_day" from dual;

next_day
-----
28-FEB-21
```

LAST_DAY:

```
SQL> select sysdate,last_day(sysdate)"last",last_day(sysdate)-sysdate "days left" from dual;

SYSDATE    last        days left
-----
27-FEB-21  28-FEB-21         1
```

ADD_MONTHS:

```
SQL> select add_months (sysdate,3) from dual;

ADD_MONTH
-----
27-MAY-21
```

MONTHS_BETWEEN:

```
SQL> select months_between(date'2017-06-01',date'2017-01-01')month_diff from dual;

MONTH_DIFF
-----
5
```

SYSTIMESTAMP:

```
SQL> select systimestamp from dual;

SYSTIMESTAMP
-----
27-FEB-21 10.54.49.458000 PM +05:30
```

CURRENT_DATE:

```
SQL> select current_date from dual;

CURRENT_D
-----
27-FEB-21
```

Name: AKSHAT CHUDASAMA

Roll no: 13

Class: FYCS

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.)

```
SQL> create table Vendor(VEND_CODE int primary key);
Table created.
SQL> create table Product(PROD_CODE varchar(10),VEND_CODE references Vendor(VEND_CODE));
Table created.
```

```
SQL> insert into Vendor values(125);
1 row created.
SQL> insert into Vendor values(126);
1 row created.
SQL> insert into Vendor values(124);
1 row created.
SQL> insert into Vendor values(123);
1 row created.
SQL> select * from Vendor;

VEND_CODE
-----
      125
      126
      124
      123
```



```

SQL> insert into Product values('ABC',125);
1 row created.

SQL> insert into Product values('DEF',124);
1 row created.

SQL> insert into Product value('GHI',124);
insert into Product value('GHI',124)
                                *
ERROR at line 1:
ORA-00928: missing SELECT keyword

SQL> insert into Product values('GHI',124);
1 row created.

SQL> insert into Product values('JKL',123);
1 row created.

SQL> select * from Product;

PROD_CODE  VEND_CODE
-----
ABC          125
DEF          124
GHI          124
JKL          123

```

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

- a) A UNION query based on these two tables

```

SQL> select VEND_CODE from Vendor
2 union
3 select VEND_CODE from Product;

VEND_CODE
-----
123
124
125
126

```

- b) A UNION ALL query based on these two tables

```
SQL> select VEND_CODE from Vendor
2 union all
3 select VEND_CODE from Product;

VEND_CODE
-----
125
126
124
123
125
124
124
123

8 rows selected.
```

- c) An INTERSECT query based on these two tables

```
SQL> select VEND_CODE from Vendor
2 intersect
3 select VEND_CODE from Product;

VEND_CODE
-----
123
124
125
```

- d) A MINUS query based on these two tables

```
SQL> select VEND_CODE from Vendor
2 minus
3 select VEND_CODE from Product;

VEND_CODE
-----
126
```

Name: AKSHAT CHUDASAMA

Roll No: 13

FYCS

Practical No: 8

Study of various types of views Considering Emp and Dept table, perform the following:

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

```
SQL> connect system/Akshat104
Connected.
SQL> create view emp_hor
  2 as select * from akshat_EMP
  3 where JOB='ANALYST';

View created.
```

```
SQL> select * from emp_hor;
```

EMP_NO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPT_NO
7788	SCOTT	ANALYST	7566	19-APR-87	3000		20
7902	FORD	ANALYST	7566	03-DEC-81	3000		20

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

```
SQL> create view vwemp(ENAME,JOB,SAL)
  2 as
  3 select ENAME,JOB,SAL from akshat_EMP;

View created.
```

```
SQL> 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
ALLEN	SALESMAN	1600
WARD	SALESMAN	1250
MARTIN	SALESMAN	1250
TURNER	SALESMAN	1500
ADAMS	CLERK	1100
JAMES	CLERK	950
MILLER	CLERK	1300

14 rows selected.

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

```
SQL> create view TSAL(TOT_SAL,JOB)
  2  as
  3  select sum(SAL),JOB from akshat_EMP
  4  group by JOB;
```

View created.

```
SQL> select * from TSAL;
```

TOT_SAL	JOB
4150	CLERK
5600	SALESMAN
5000	PRESIDENT
8275	MANAGER
6000	ANALYST

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

```
SQL> create view num4
  2  as
  3  select Ename,dept_no,JOB from akshat_EMP;
```

View created.

```
SQL> select * from num4;
```

ENAME	DEPT_NO	JOB
KING	10	PRESIDENT
BLAKE	30	MANAGER
CLARK	10	MANAGER
JONES	20	MANAGER
SCOTT	20	ANALYST
FORD	20	ANALYST
SMITH	20	CLERK
ALLEN	30	SALESMAN
WARD	30	SALESMAN
MARTIN	30	SALESMAN
TURNER	30	SALESMAN
ADAMS	20	CLERK
JAMES	30	CLERK
MILLER	10	CLERK

14 rows selected.

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

```
SQL> create view new3
  2  as
  3  select SAL,JOB from akshat_EMP
  4  where dept_no=20;

View created.

SQL> select * from new3;

      SAL JOB
-----
    2975 MANAGER
    3000 ANALYST
    3000 ANALYST
     800 CLERK
    1100 CLERK
```

6. Delete all the views created above

```
SQL> drop view emp_hor;

View dropped.

SQL> drop view vwemp;

View dropped.

SQL> drop view TSAL;

View dropped.

SQL> drop view new3;

View dropped.
```

Name: AKSHAT CHUDASAMA

Roll No:13

FYCS

Practical No: 9

Study of subqueries with all its clauses

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

```
SQL> select Ename from akshat_EMP
  2  where SAL>(select SAL from akshat_EMP where Emp_no=7566);

ENAME
-----
KING
SCOTT
FORD
```

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

```
SQL> select Ename,SAL,JOB
  2  from akshat_EMP
  3  where JOB=(select JOB from akshat_EMP where Emp_no=7369);

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

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

```
SQL> select Ename,SAL,JOB
  2  from akshat_EMP
  3  where SAL<any(select SAL from akshat_EMP where JOB='CLERK');

ENAME          SAL  JOB
-----
SMITH           800  CLERK
WARD            1250 SALESMAN
MARTIN          1250 SALESMAN
ADAMS           1100 CLERK
JAMES           950  CLERK
```

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

```
SQL> select Ename,SAL,Dept_no,JOB from akshat_EMP
  2  where JOB=(select JOB from akshat_EMP where Emp_no=7900);
```

ENAME	SAL	DEPT_NO	JOB
SMITH	800	20	CLERK
ADAMS	1100	20	CLERK
JAMES	950	30	CLERK
MILLER	1300	10	CLERK

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

```
SQL> select Ename,SAL,Dept_no,JOB from akshat_EMP
  2  where JOB=(select JOB from akshat_EMP where Emp_no=7698);
```

ENAME	SAL	DEPT_NO	JOB
BLAKE	2850	30	MANAGER
CLARK	2450	10	MANAGER
JONES	2975	20	MANAGER

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

```
SQL> select Ename,SAL
  2  from akshat_EMP
  3  where SAL>any(select SAL from akshat_EMP where JOB='MANAGER');
```

ENAME	SAL
KING	5000
BLAKE	2850
JONES	2975
SCOTT	3000
FORD	3000

AKSHAT CHUDASAMA

FYCS – 13

FYCS

Practical 10

Study of Transaction (Commit/ Rollback), Locks

1.Perform Commit and Rollback on a table.

```
SQL> create table x(rno int);
Table created.
SQL> insert into x values(1);
1 row created.
SQL> insert into x values(2);
1 row created.
SQL> commit;
Commit complete.
SQL> select * from x;

      RNO
-----
        1
        2

SQL> delete from x where rno = 2;
1 row deleted.
SQL> select * from x;

      RNO
-----
        1

SQL> rollback;
Rollback complete.
SQL> select * from x;

      RNO
-----
        1
        2
```


2. Implementation of Share and Exclusive Lock Mode in employee table.

Share mode:

```
SQL> lock table Akshat_EMP in share mode;
Table(s) Locked.
```

```
SQL*Plus: Release 10.2.0.1.0 - Production on Tue Apr 13 14:01:33 2021
Copyright (c) 1982, 2005, Oracle. All rights reserved.

SQL> connect system/Akshat104
Connected.
SQL> lock table Akshat_EMP in share mode;

Table(s) Locked.

SQL>
```

Exclusive mode:

```
SQL> lock table Akshat_EMP in exclusive mode;
Table(s) Locked.
```

```
Run SQL Command Line

SQL*Plus: Release 10.2.0.1.0 - Production on Tue Apr 13 14:05:07 2021
Copyright (c) 1982, 2005, Oracle. All rights reserved.

SQL> connect system/Akshat104
Connected.
SQL> lock table Akshat_EMP in exclusive mode;

Table(s) Locked.

SQL>
```

AKSHAT CHUDASAMA

FYCS 13

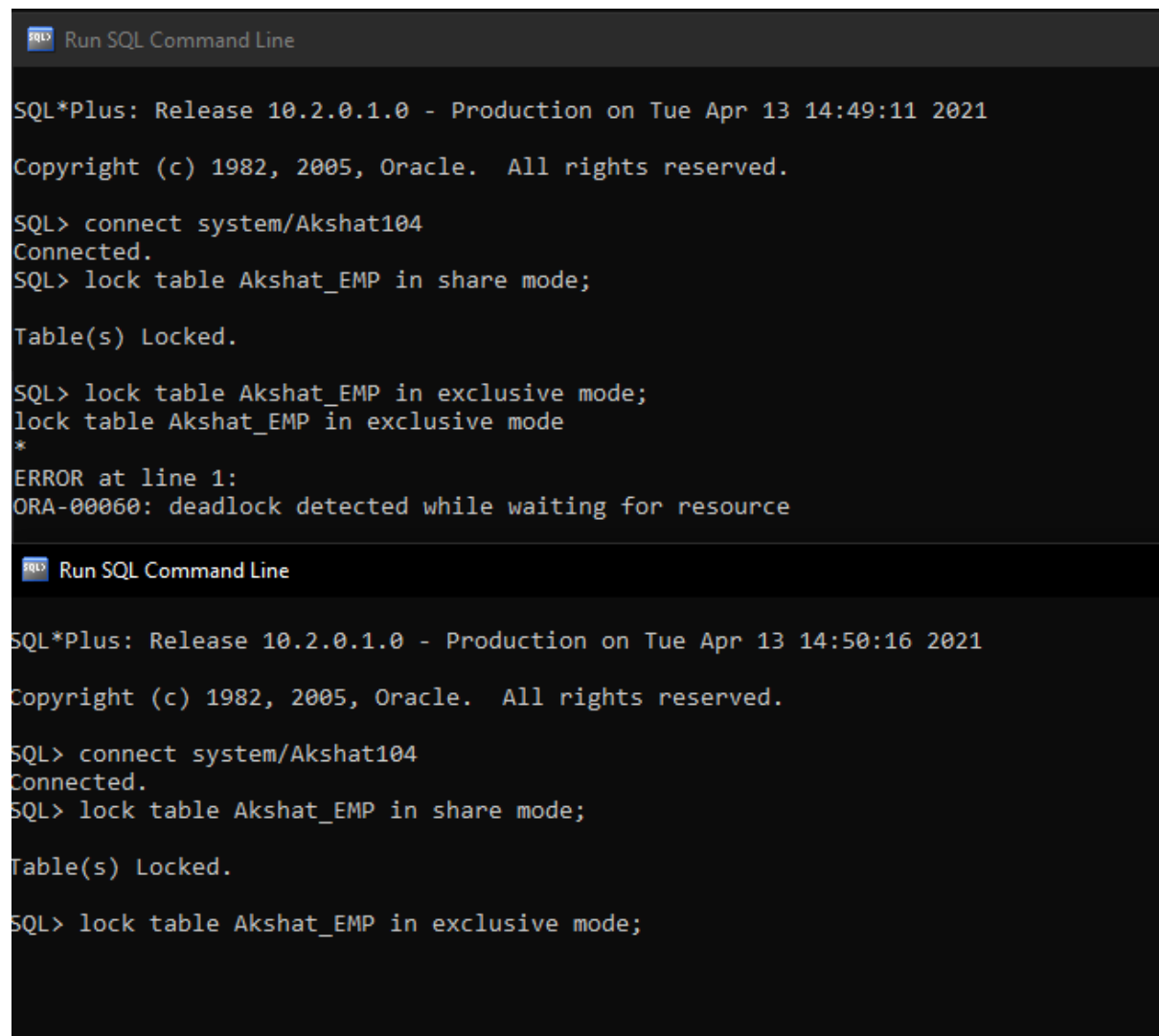
FYCS

PRACTICAL 11

Implementing Deadlocks

Deadlock :

A deadlock is a situation on which two or more transactions are waiting for one another to give up locks



```
SQL> Run SQL Command Line

SQL*Plus: Release 10.2.0.1.0 - Production on Tue Apr 13 14:49:11 2021

Copyright (c) 1982, 2005, Oracle. All rights reserved.

SQL> connect system/Akshat104
Connected.
SQL> lock table Akshat_EMP in share mode;

Table(s) Locked.

SQL> lock table Akshat_EMP in exclusive mode;
lock table Akshat_EMP in exclusive mode
*
ERROR at line 1:
ORA-00060: deadlock detected while waiting for resource
```



```
SQL> Run SQL Command Line

SQL*Plus: Release 10.2.0.1.0 - Production on Tue Apr 13 14:50:16 2021

Copyright (c) 1982, 2005, Oracle. All rights reserved.

SQL> connect system/Akshat104
Connected.
SQL> lock table Akshat_EMP in share mode;

Table(s) Locked.

SQL> lock table Akshat_EMP in exclusive mode;
```