

**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**

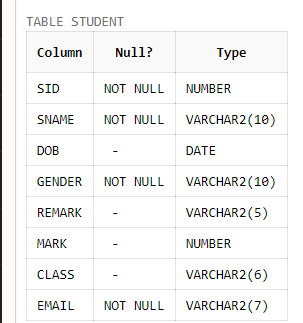
A) Write the query for the following

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);

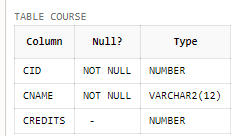
desc student;



b) Course (cid, cname, credits)

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

desc course;



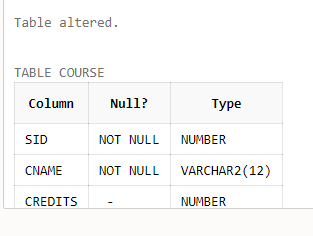
**2) Alter the structure of the Course table**

a) Modify datatype of cname.

alter table course

modify cname varchar(12);

desc course;

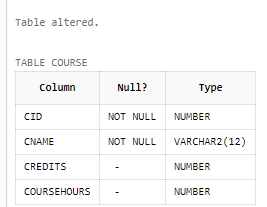


b) Add a column coursehours with minimum course hours greater than 45.

alter table course

add coursehours int check(coursehour>45);

desc course;

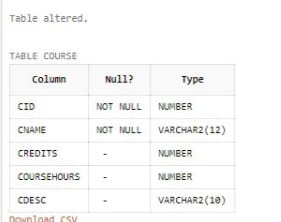


c) Add a column cdesc

alter table course

add cdesc varchar(10);

desc course;



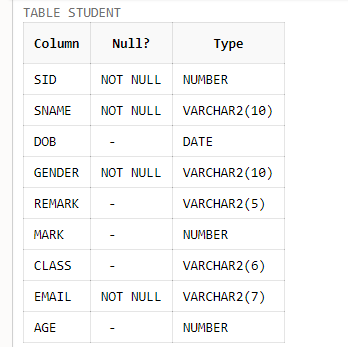
**3) Alter the structure of Student Table**

a) Add column age with minimum age as 17.

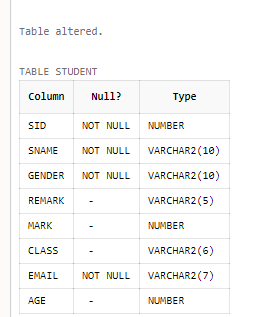
alter table student

add age int check(age>16);

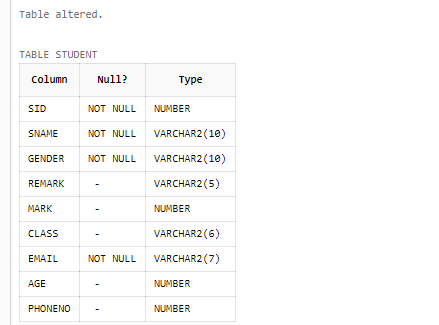
desc student;



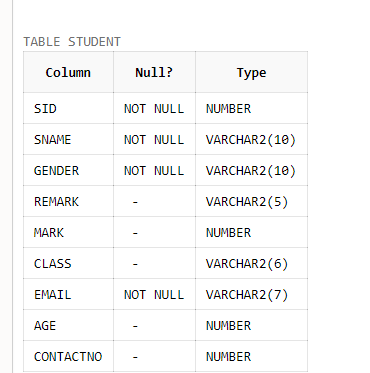
b) Delete the column dob



c) Add a column phoneno.



d) Rename phoneno to contactno

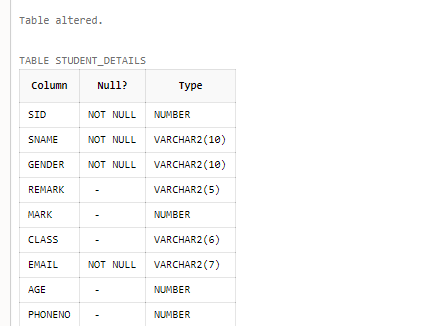


4) Rename Student table as Student\_details.

alter table student

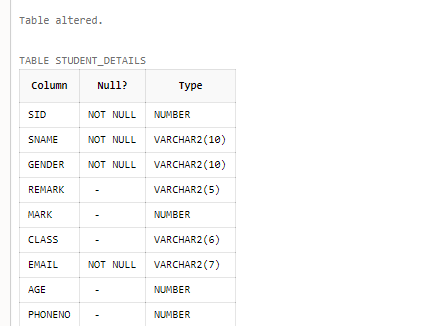
rename to student\_details;

desc student\_details;



5) Describe the structure of both the tables.

desc student\_details;



6) Drop the table Student\_details and Course.

alter table student\_details;

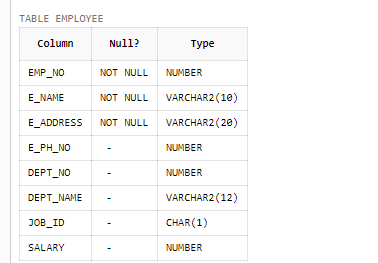
drop table student\_details;



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;

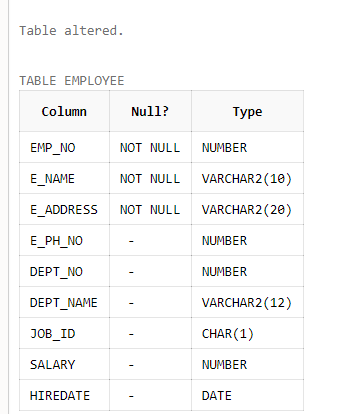


2. Add a new column HIREDATE to the existing relation.

alter table EMPLOYEE

add HIREDATE date;

desc EMPLOYEE;

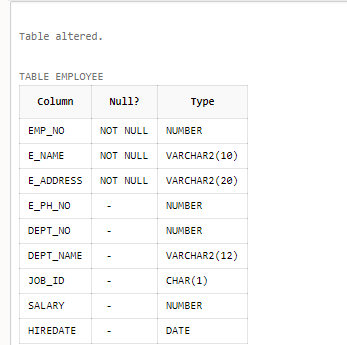


3. Change the datatype of JOB\_ID from char to varchar2.

alter table EMPLOYEE

modify job\_id varchar2;

desc EMPLOYEE;

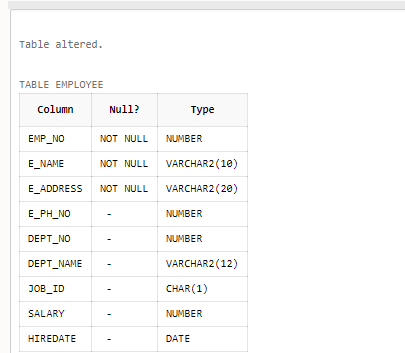


4. Change the name of column/field Emp\_no to E\_no.

alter table EMPLOYEE

rename column Emp\_no to E\_no;

desc EMPLOYEE;

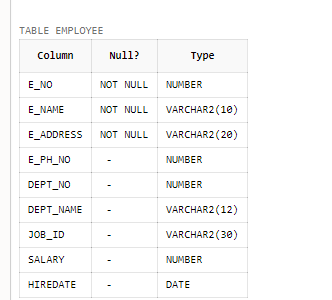


5. Modify the column width of the job field of emp table.

alter table EMPLOYEE

modify job\_id varchar(30);

desc EMPLOYEE;

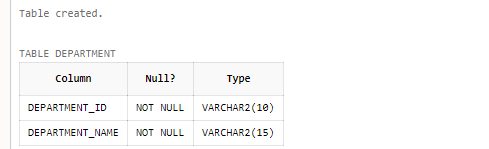


C**) Create the following tables with specified attributes and constraints Department Table**:

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);

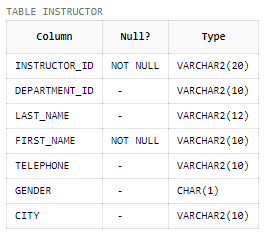
desc Department;



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;



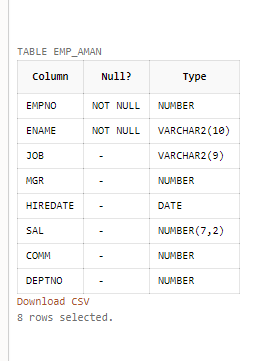
1. **Create the following described below:**

# 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;

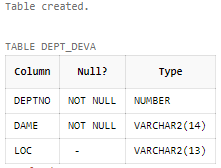


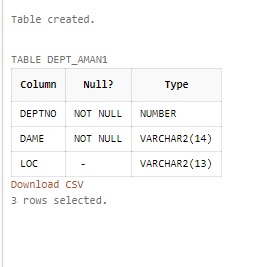
# Table Name: DEPT

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **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;



­­­­­­­­­­­­­­­

# Practical 2: Study of Data Manipulation Language Statement

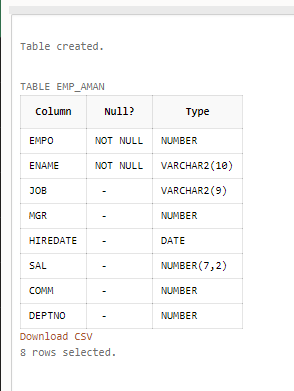
**A) Insert the following records in above created table**

**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;



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



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



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



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



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



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



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



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



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



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



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



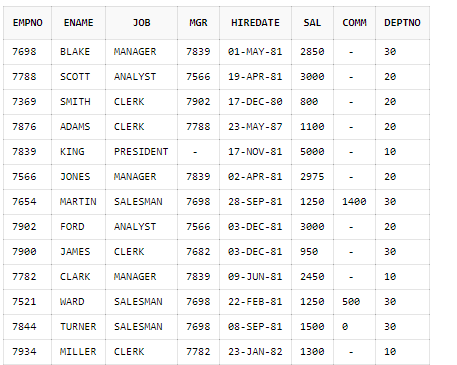
insert into Emp\_Aman values(7900,'JAMES','CLERK',7682,'03-Dec-81',950,'',30);



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



select \* from Emp\_Aman;



# 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');



insert into Dept\_Aman1 values(20,'RESEARCH','DALLAS');

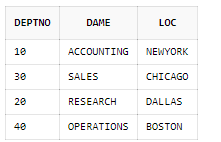


insert into Dept\_Aman1 values(30,'SALES','CHICAGO');



insert into Dept\_Aman1 values(40,'OPERATIONS','BOSTON');





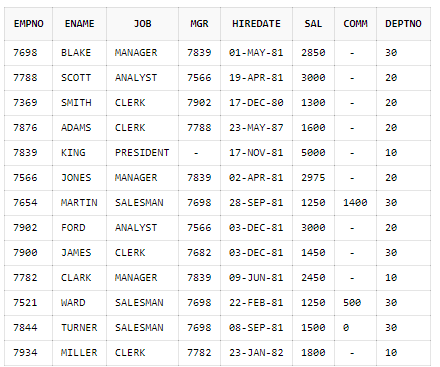
**B) Update and Delete Queries**

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;



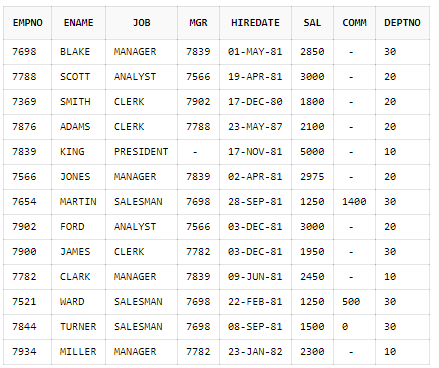


2)Update the manager of James as CLARK.

UPDATE Emp\_Aman SET MGR=7782 WHERE ENAME='JAMES';



select \* from Emp\_Aman;

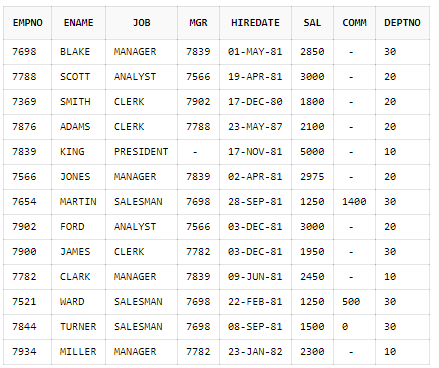


3)Change the role of Miller as MANAGER.

UPDATE Emp\_Aman SET JOB='MANAGER'WHERE ENAME='MILLER';



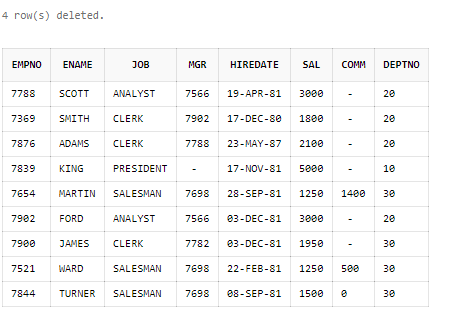
select \* from Emp\_Aman;



4)Delete the records of Manager

Delete from Emp\_Aman WHERE JOB='MANAGER';

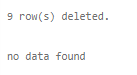
select \* from Emp\_Aman;



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

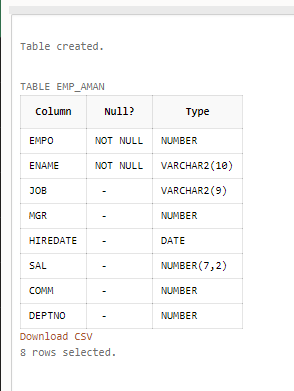
Delete from Emp\_Aman WHERE SAL>1000;

select \* from Emp\_Aman;



• 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;

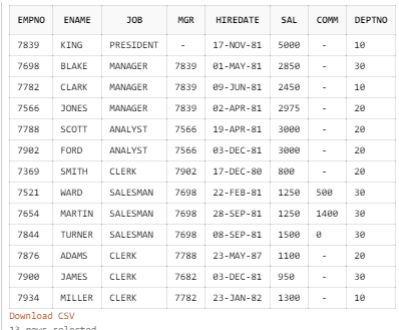


PRACTICAL 3

A) Using emp table, perform the following queries:

1) Display the details of all employees.

select \* from Emp\_Aman;



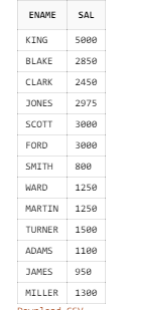
2) Display the name and job for all employees.

select ename,job from Emp\_Aman;



3) Display name and salary for all employees.

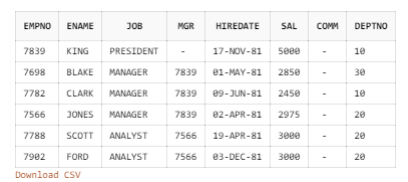
select ename,sal from Emp\_Aman;



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

select \* from Emp\_Aman

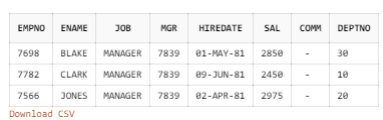
where sal>2000;



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

select \* from Emp\_Aman

where job='MANAGER';



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

select ename from Emp\_Aman

where deptno=10;



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;



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

select empno,ename from Emp\_Aman

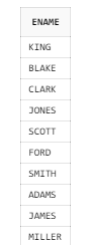
where comm is not null;



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

select ename from Emp\_Aman

where comm is null;



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;



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');



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);



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;



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

select ename from Emp\_Aman

order by ename asc;



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

select Empno,ename,sal

from Emp\_Aman

where mgr=7369;

Screenshot (143)

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;



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

select ename from Emp\_Aman

where deptno in (10,20);



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

select ename from Emp\_Aman

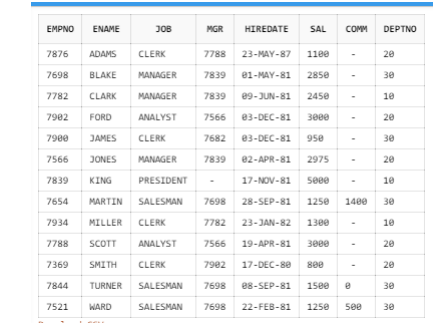
where ename like '%LL%';



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

select \* from Emp\_Aman

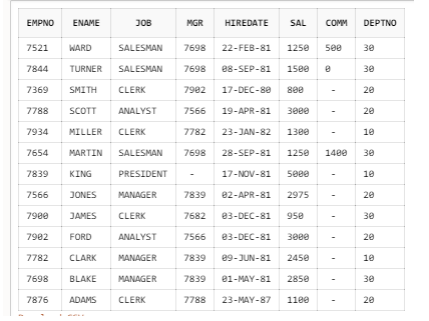
order by ename asc;



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

select \* from Emp\_Aman

order by ename desc;



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

select ename from Emp\_Aman

where deptno in (10,30);



22) List all the employees except PRESIDENT and MANAGER.

select ename from Emp\_ Aman

where job not in ('PRESIDENT','MANAGER');



23) List the employees whose name starts with A.

select ename from Emp\_Aman

where ename like 'A%';

Screenshot (158)

24) List all the Clerks of Deptno 20.

select ename from Emp\_Aman

where deptno=20 and job='CLERK';



25) List the employees whose names ends with S.

select ename from Emp\_Aman

where ename like '%S';



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

select \* from Emp\_Aman

where ename like '\_\_\_\_';



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;



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

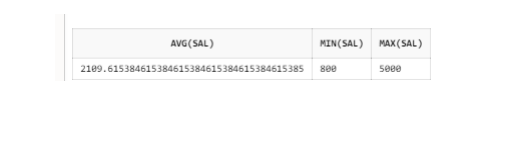
select sum(sal) from Emp\_Aman

where job='ANALYST';



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

select avg(sal),min(sal),max(sal) from Emp\_Aman;



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

select count(\*) from Emp\_Aman

where deptno=10;

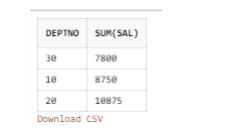


B) Answer the following queries:

1) Display the total salary of employees department wise.

select deptno,sum(sal) from Emp\_Aman

group by deptno;



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;



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

select job,count(\*) from Emp\_Aman

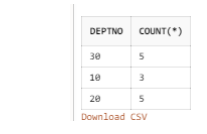
group by job;



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

select deptno,count(\*) from Emp\_Aman

group by deptno;

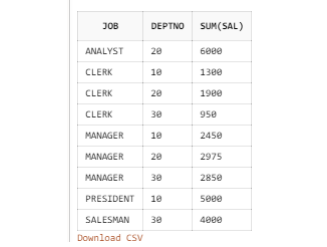


5) Display the total salary of employees specific to job and department in ascending order of job.

select job,deptno,sum(sal) from Emp\_Aman

group by job,deptno

order by job asc;

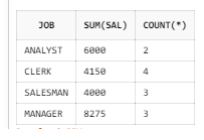


6) Display the total salary of the employees specific to job when employee count is greater than 1.

select job,sum(sal),count(\*) from Emp\_Aman

group by job

having count(job)>1;

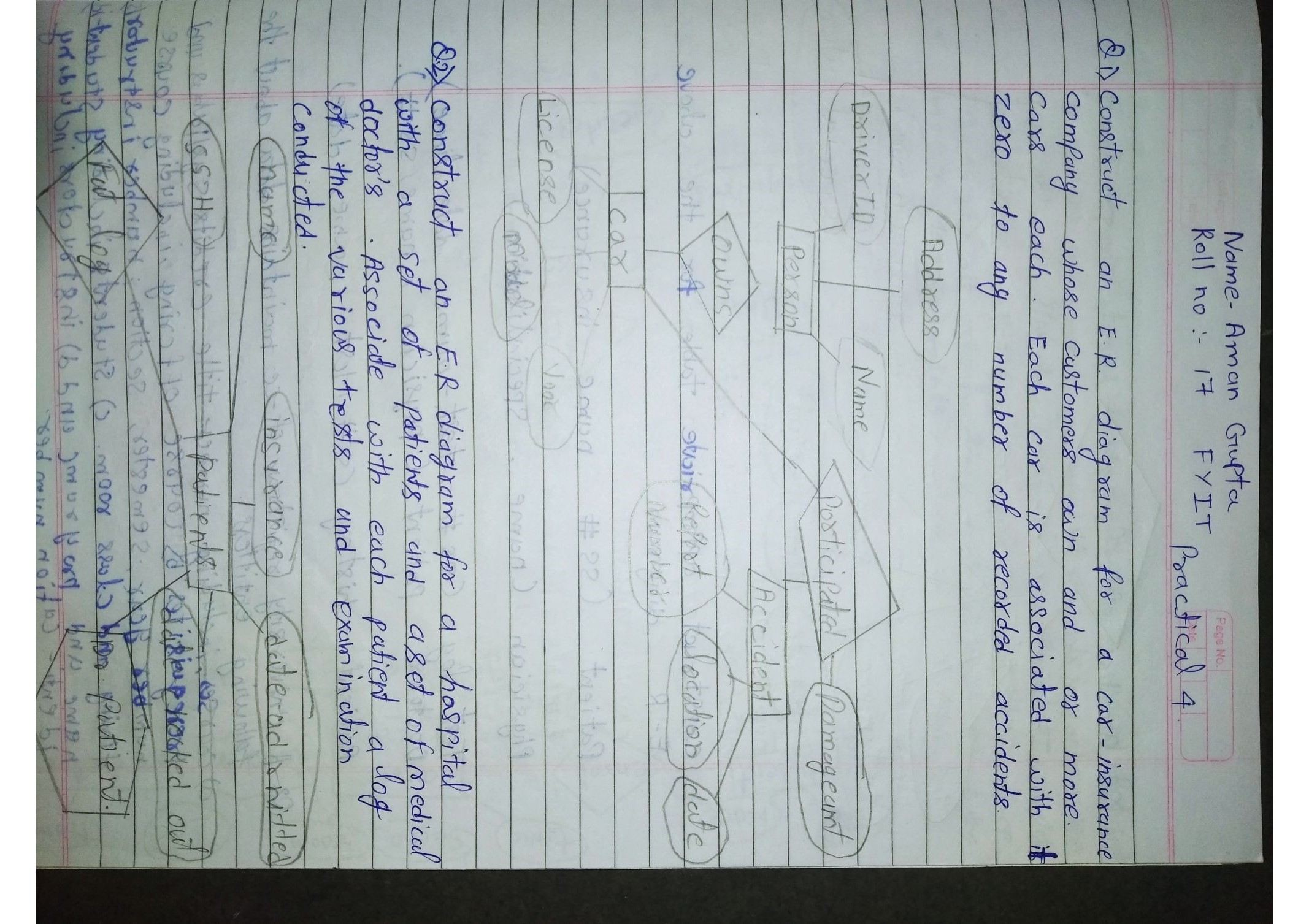


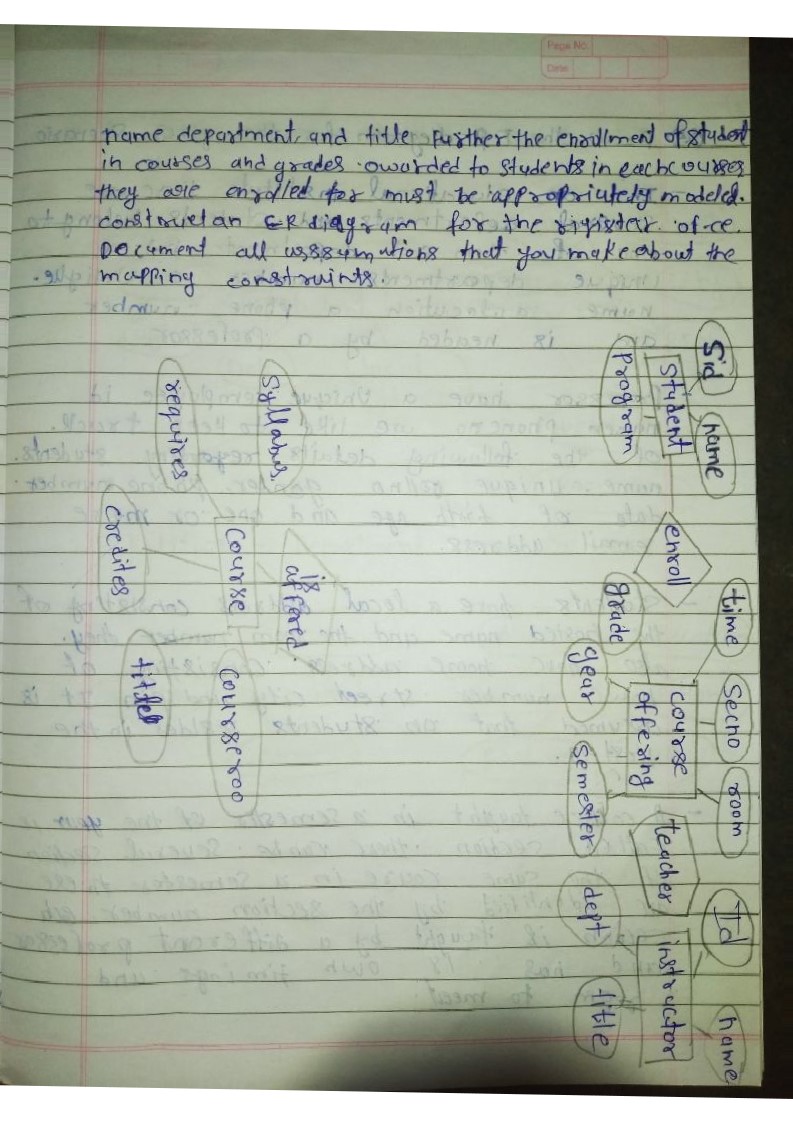
7) Display unique jobs of employees.

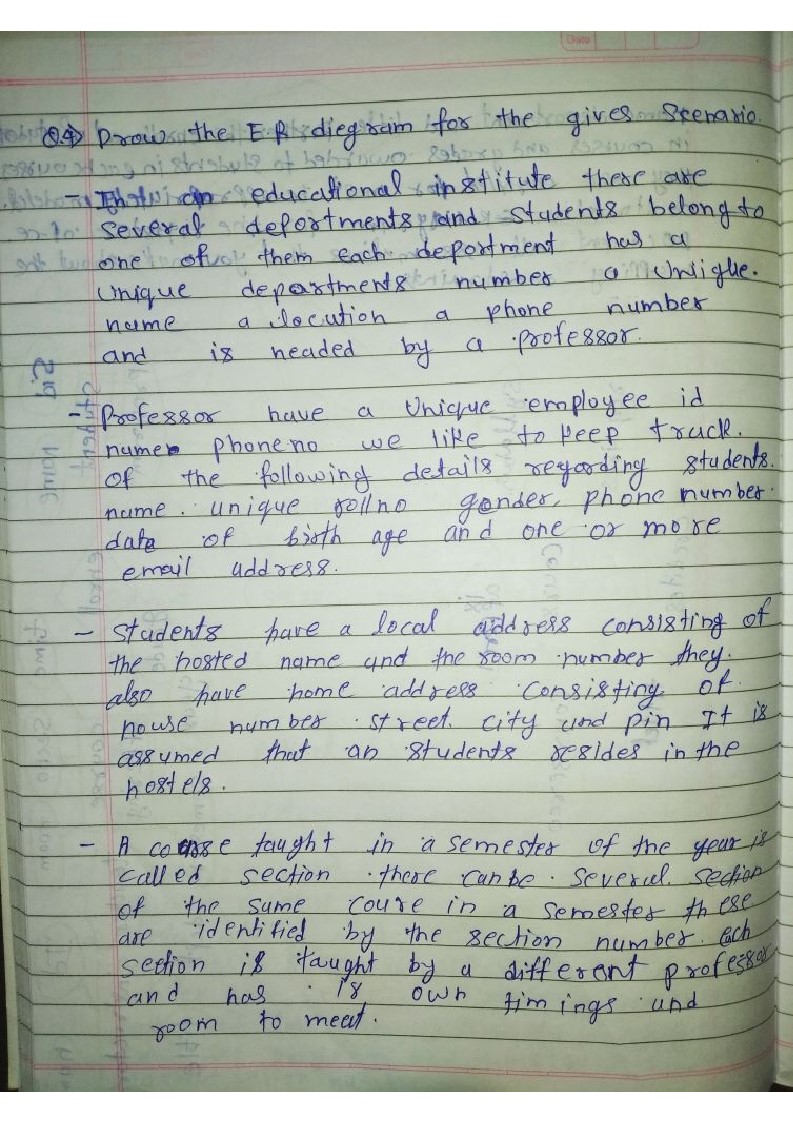
select distinct job from Emp\_Aman;

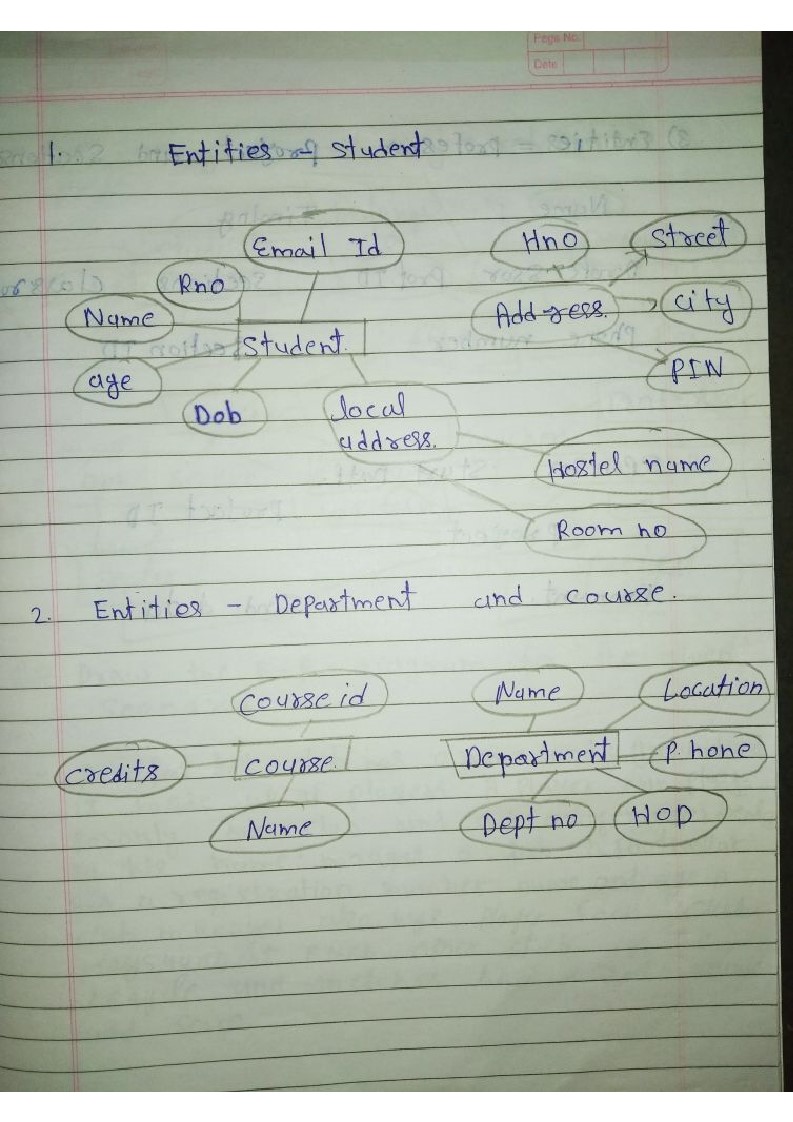


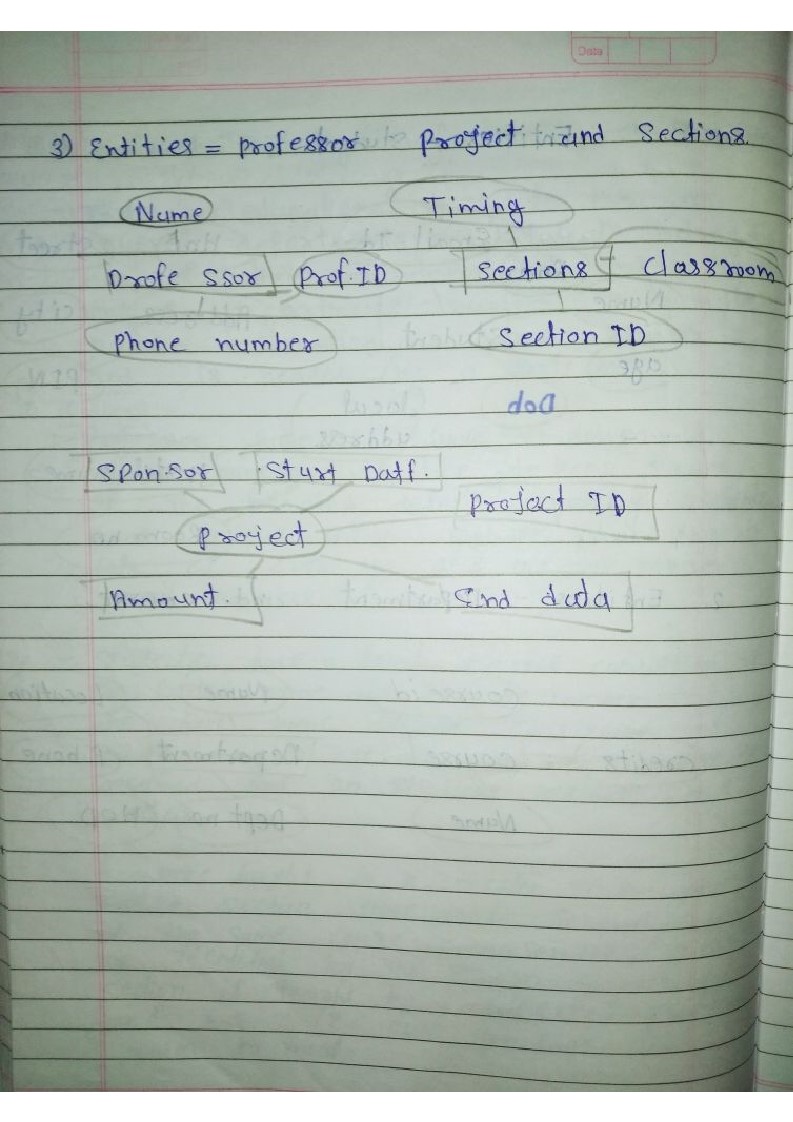
PRACTICAL 4

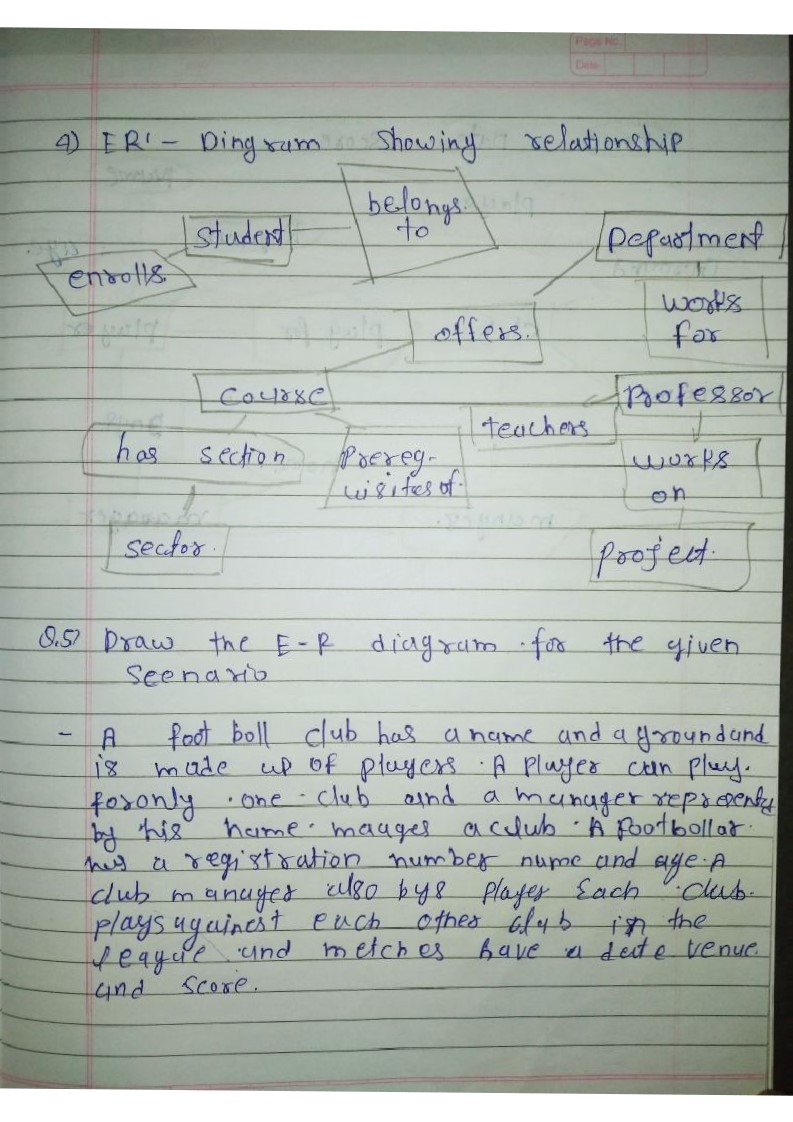


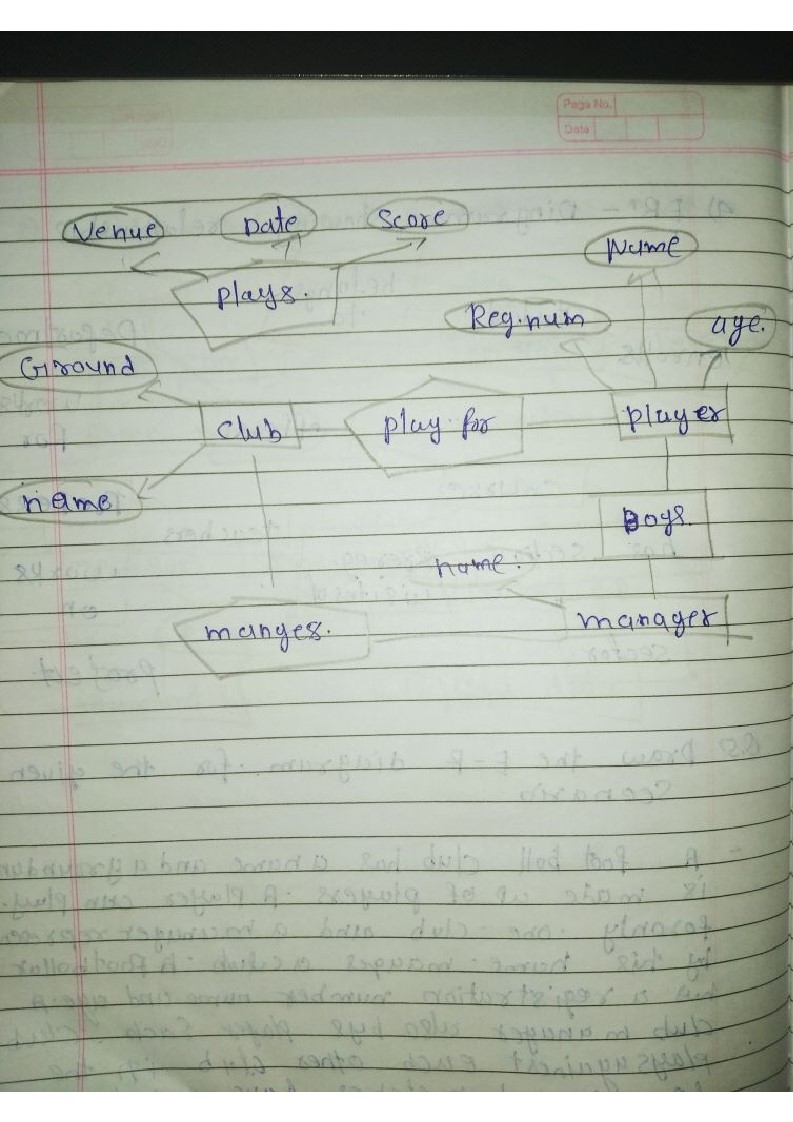












## **Practial No .5 (JOIN)**

**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.

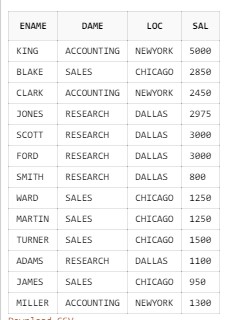
**Inner Join:** the INNER JOIN to fetch rows that have matching values in both the tables that we are joining.

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;



**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;



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

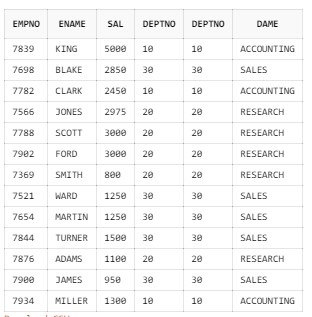
Syntax:

1. SELECT column list 4. FROM table1, table2....
2. WHERE table1.column\_name =
3. 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;



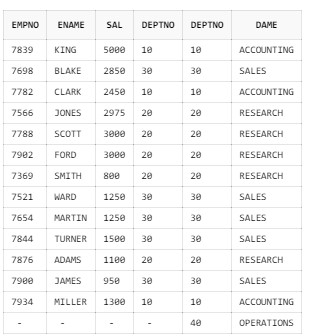
Outer Join: In an outer join, along with tuples that satisfy the matching criteria, we also include some or all tuples that do not match the criteria.

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;

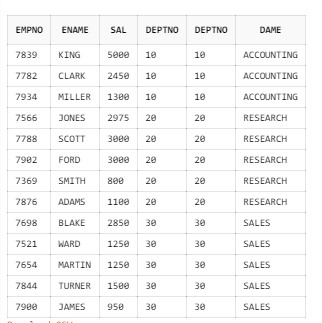


**Left join:** left join returns all the values from left table and the matching values from the right table. If there is no matching join value, it will return NULL.

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;



**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;

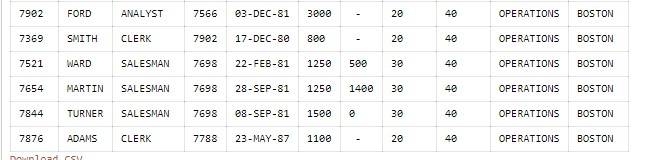
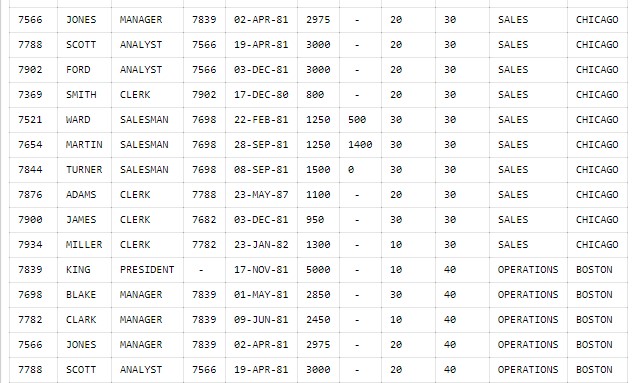
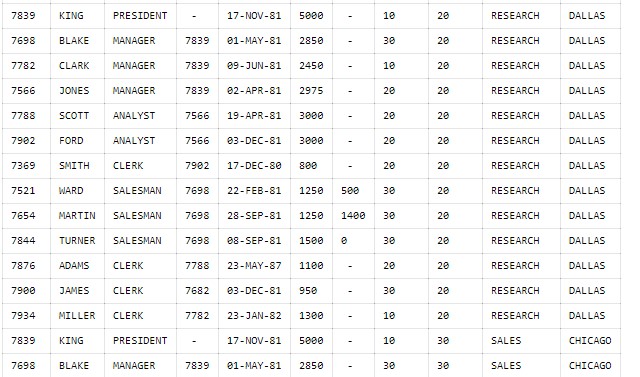
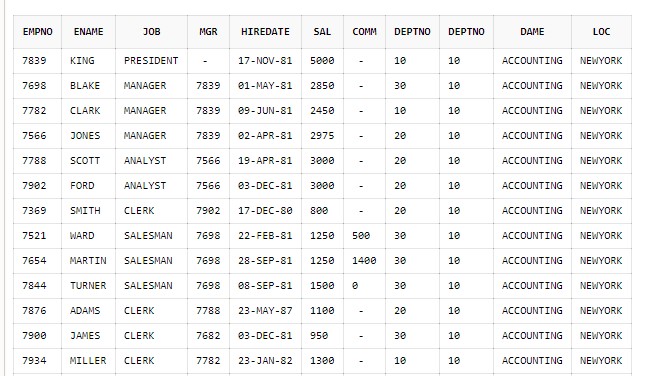


**Cross join:** CROSS JOIN specifies that all rows from first table join with all of the rows of second table.

Syntax:

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

select \* from Emp\_Aman cross join Dept\_Aman;



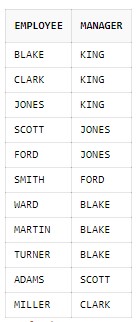
**Self-join:** A self-join is a join in which a table is joined with itself. To join a table itself means that each row of the table is combined with itself and with every other row of the table.

Syntax:

1. SELECT a. column\_name, b.column\_name... 2. FROM table1 a, table1 b

3. WHERE a.common\_filed = b.common\_field;

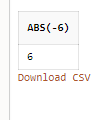
select e2.ename employee,e1.ename manager from Emp\_ Aman e1,Emp\_ Aman e2 where e1.empno=e2.mgr;



**PRACTICAL NO:- 6**

1. **Numeric function :-**
2. **Absolute :-**

**select abs(-9) from dual;**

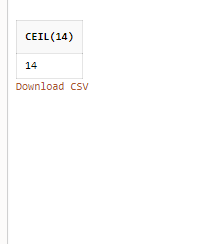
****

**select abs(6) from dual;**

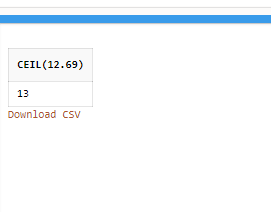
****

1. **Ceil :-**

**select ceil(14) from dual;**

****

**select ceil(12.69) from dual;**

****

1. **Floor:-**

**select floor(12.17) from dual;**

****

**select floor(12) from dual;**

****

1. **Round**

**select round(12.18) from dual;**

****

**select round(12.56) from dual;**

****

1. **Remainder:-**

**select remainder(13,3) from dual;**

****

1. **Square root:-**

**select sqrt(25)from dual;**

****

1. **Modules :-**

**select mod(25,4) from dual;**

****

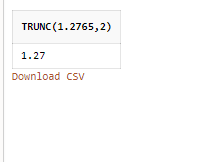
1. **Power :-**

**select power(6,6) from dual;**

****

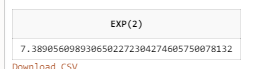
1. **Trunc :-**

**select trunc(1.2765,2) from dual;**

****

1. **Exponential :-**

**select exp(2) from dual;**

****

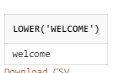
1. **Log :-**

**select log(10,100) from dual;**

****

1. **Character function :-**
2. **Lower :-**

**select lower('WELCOME') from dual;**

****

1. **Upper :-**

**select upper('welcome') from dual;**

****

1. **Initcap:-**

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

****

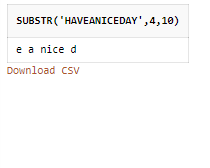
1. **Length** :-

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

****

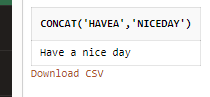
1. **Substr :-**

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

****

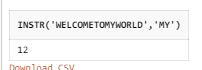
1. **Concat :-**

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

****

1. **Instr :-**

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

****

1. **Trim :-**

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

****

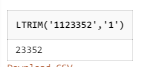
**select trim(trailing '4' from '1123424') from dual;**

****

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

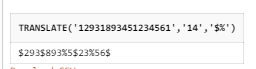
****

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

****

1. **Translate :-**

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

****

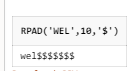
1. **Replace:-**

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

****

1. **Rpad:-**

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

****

1. **Lpad:-**

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

****

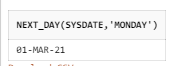
1. **Date :-**
2. **Sysdate:-**

**select sysdate from dual;**

****

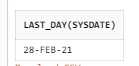
1. **Next\_Day:-**

**select next\_day(sysdate,'monday') from dual;**

****

1. **Last\_Day:-**

**select last\_day(sysdate) from dual;**

****

1. **Add\_months:-**

**select add\_months(sysdate,-1) from dual;**

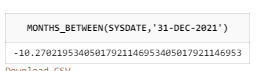
****

1. **Months\_Between:-**

**select months\_between('22-dec-21',sysdate) from dual;**

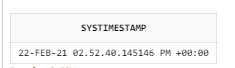
****

**select months\_between(sysdate,'31-dec-2021') from dual;**

****

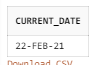
1. **Systimestamp:-**

**select systimestamp from dual;**

****

1. **Current\_date:-**

**select current\_date from dual;**

****

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);

Screenshot (182)

create table product(PROD\_CODE varchar(3),VEND\_CODE int,foreign key(VEND\_CODE)references vendor(VEND\_CODE));

Screenshot (182)

insert into vendor values('123');

Screenshot (183)

insert into vendor values('124');

Screenshot (183)

insert into vendor values('125');

Screenshot (183)

insert into vendor values('126');

Screenshot (183)

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

Screenshot (183)

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

Screenshot (183)

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

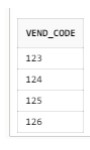
Screenshot (183)

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

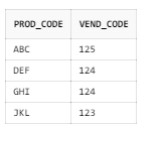
Screenshot (183)

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

select \* from vendor;



select \* from product;



1. A UNION query based on these two tables

select VEND\_CODE from vendor union

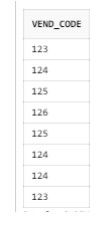
select VEND\_CODE from product;



1. A UNION ALL query based on these two tables

select VEND\_CODE from vendor

union all select VEND\_CODE from product;



1. An INTERSECT query based on these two tables

select VEND\_CODE from vendor intersect

select VEND\_CODE from product;



1. A MINUS query based on these two tables

select VEND\_CODE from vendor

minus

select VEND\_CODE from product;

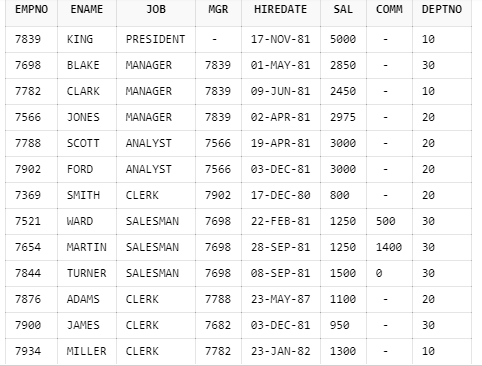


**PRACTICAL 8: Study of various types of views**

Considering Emp and Dept table, perform the following:

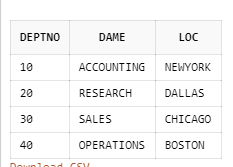
EMP Table:

select \* from Emp\_Aman;



DEPT Table:

Select \* from Dept\_Aman1;



1. Create a view named emp\_hor with the job titled as ‘ANALYST’.

create view Emp\_A

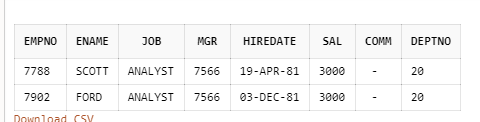
as

select \* from Emp\_Aman

where job='ANALYST';



select \* from Emp\_A;



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

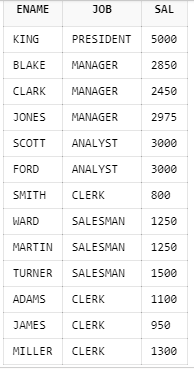
create view vwemp

as

select ENAME,JOB,SAL from Emp\_Aman;



select \* from vwemp;



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

create view hemp(job,sal)

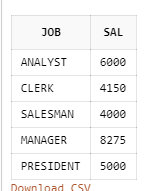
as

select job,sum(SAL) from Emp\_Aman

group by job;



select \* from hemp;



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

create view AG

as

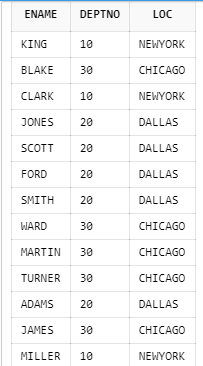
select ENAME,Emp\_Aman.deptno,loc

from Emp\_Aman inner join Dept\_Aman1

on Emp\_Aman.deptno=Dept\_Aman1.deptno;



select \* from AG;



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

create view Aman12

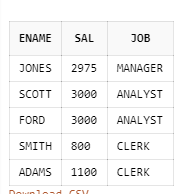
as

select ENAME,SAL,JOB from Emp\_Aman

where deptno=20;



select \* from Aman12;



1. Delete all the views created above

drop view Emp\_A;



drop view vwemp;



drop view hemp;



drop view AG;



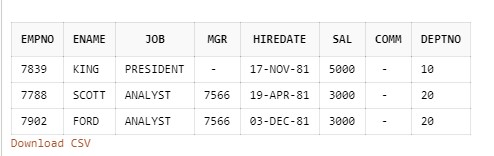
drop view Aman12;



**Practical 9: Study of subqueries with all its clauses**

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

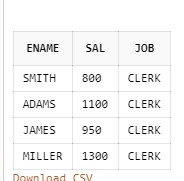
select \* from Emp\_Aman where sal>(select sal from Emp\_Aman where empno = 7566);



1. 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

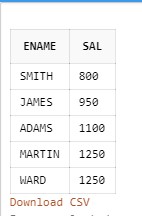
from Emp\_Aman where empno=7369);



1. 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');



1. 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 =

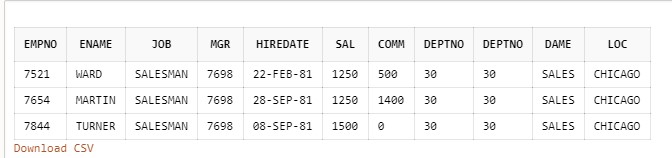
(select job from Emp\_Aman where empno=7900);



1. 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);



1. 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');

