INDEX

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sr. No.** |  | **Definition** | **Page No.** | **Date** | **Signature** |
| 1. | SET – 1 |  |  |  |  |
| 2. | SET – 2 |  |  |  |  |
| 3. | SET – 3 |  |  |  |  |
| 4. | SET – 4 |  |  |  |  |
| 5. | SET – 5 |  |  |  |  |
| 6. | SET – 6 |  |  |  |  |
| 7. | SET – 7 |  |  |  |  |
| 8. | SET – 8 |  |  |  |  |
| 9. | SET – 9 |  |  |  |  |
| 10. | SET – 10 |  |  |  |  |
| 11. | SET – 11 |  |  |  |  |
| 12. | SET – 12 |  |  |  |  |
| 13. | SET – 13 |  |  |  |  |
| 14. | SET – 14 |  |  |  |  |
| 15. | SET – 15 |  |  |  |  |
| 16. | SET – 16 |  |  |  |  |
| 17. | SET – 17 |  |  |  |  |
| 18. | SET – 18 |  |  |  |  |
| 19. | SET – 19 |  |  |  |  |
| 20. | SET – 20 |  |  |  |  |

**SET-1**

**DEPARTMENT (dept\_no, dept\_name, location)**

**1. Create the Simple DEPARTMENT Table. 2. Display structure of department table.**

1. **Insert below records into Department Table.**
2. **Display all records of Department table.**
3. **Display all department belonging to location 'NY'.**
4. **Display details of Department 10 .**
5. **List all department names starting with 'A' .**
6. **List all departments whose number is between 1 and 100 .**
7. **Delete 'TRG' department.**
8. **Change department name 'EDP' to 'IT.**

**DEPARTMENT (dept\_no, dept\_name, location)**

* 1. **Create the Simple DEPARTMENT Table.**

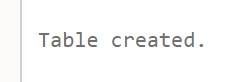
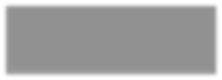
**Query:-** create table Department

(

dept\_no int,

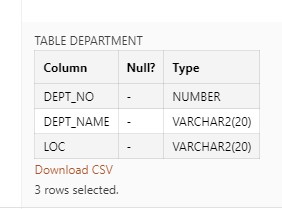
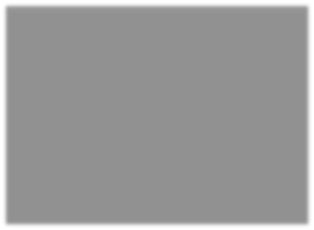
dept\_name varchar(20), loc varchar(20)

);



* 1. **Display structure of department table. Query:-**

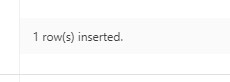
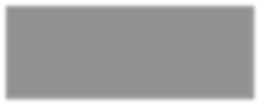
DESC Department



* 1. **Insert below records into Department Table.**

**Query:-**

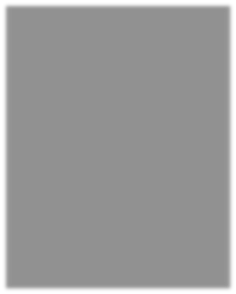
insert into Department values(10,'Account','NY') insert into Department values(20,'HR','NY') insert into Department values(30,'Production','DL') insert into Department values(40,'Sales','NY') insert into Department values(50,'EDP','MU') insert into Department values(60,'TRG','NY') insert into Department values(110,'RND','AH')



* 1. **Display all records of Department table.**

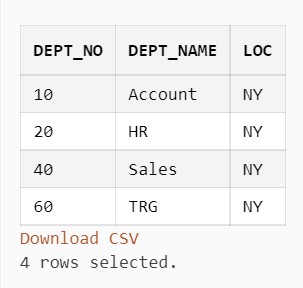
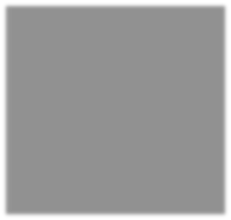
**Query:-**

select \* from Department



* 1. **Display all department belonging to location 'NY'. Query:-**

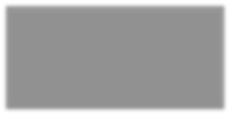
select \* from Department where loc='NY'



* 1. **Display details of Department 10.**

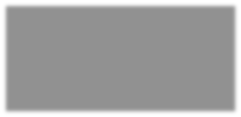
**Query:-**

select \* from Department where dept\_no=10



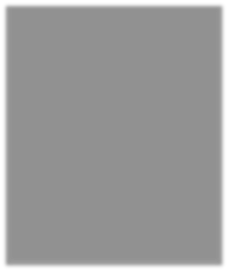
* 1. **List all department names starting with 'A'. Query:-**

select \* from Department where dept\_name like 'A%'



* 1. **List all departments whose number is between 1 and 100 Query:-**

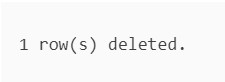
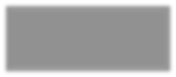
select \* from Department where dept\_no between 1 and 100



* 1. **Delete 'TRG' department.**

**Query:-**

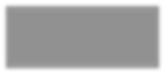
delete from Department where dept\_name='TRG'



* 1. **Change department name 'EDP' to 'IT.**

**Query:-**

update Department set dept\_name='TT'where dept\_name='EDP'



**SET-2**

**EMPLOYEE (emp\_id, emp\_name, birth\_date, gender, dept\_no, address, designation, salary, experience, email) DEPARTMENT (dept\_no, dept\_name, location)**

**Do as directed:**

* + 1. **Create the EMP Table with all necessary constraints such as In EMP TABLE: Employee id should be primary key, Department no should be Foreign key, employee age (birth\_date) should be greater than 18 years, salary should be greater than zero, email should have (@ and dot) sign in address, designation of employee can be “manager”, “clerk”, “leader”, “analyst”, “designer”, “coder”, “tester”.**
    2. **Create DEPT table with neccessary constraint such as**
    3. **Department no should be primary key, department name should be unique.**
    4. **After creation of above tables, modify Employee table by adding the constraints as 5.**

**‘Male’ or ‘Female’ in gender field and display the structure.**

* + 1. **Insert proper data (at least 5 appropriate records) in all the tables.**
    2. **Describe the structure of table created**
    3. **List all records of each table in ascending order.**
    4. **Delete the department whose loction is Ahmedabad.**
    5. **Display female employee list**
    6. **Display Departname wise employee Names**
    7. **Find the names of the employee who has salary less than 5000 and greater than 2000.**
    8. **Display the names and the designation of all female employee in descending order. 14. Display the names of all the employees who names starts with ‘A’ ends with ‘A’. 15. Find the name of employee and salary for those who had obtain minimum salary. 16. Add 10% raise in salary of all employees whose department is ‘IT’.**
    9. **Count total number of employees of ‘IT’ department.**
    10. **List all employees who born in the current month.**
    11. **Print the record of employee and dept table as “Employee works in department ‘MBA’.**
    12. **List names of employees who are fresher’s (less than 1 year of experience).**
    13. **List department wise names of employees who has more than 5 years of experience.**
    14. **Crete Sequence to generate department ID**
    15. **List department having no employees**

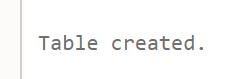
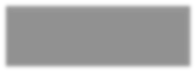
# Query:-

• **Department table Query:-**

## create table Department

( dept\_no int primary key, dept\_name varchar(20) unique, loc varchar(20)

)



• **Employee table Query:-**

create table employee ( emp\_id int primary key, emp\_name varchar(20), birth\_date date, gender varchar(6),

dept\_no int REFERENCES department(dept\_no), address varchar(100),

designation varchar(15) constraint chk\_desig CHECK

(designation in ('manager', 'clerk', 'leader',

'analyst',

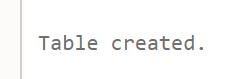
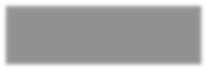
## 'designer', 'coder', 'tester')),

salary int constraint chk\_sal CHECK (salary>0), experience varchar(5),

email varchar2(255) constraint chk\_emp\_email CHECK

(email

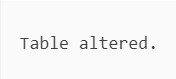
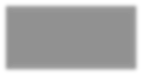
like '%@%' and email like '%.%') )



1. **After creation of above tables, modify Employee table by adding the constraints a ‘Male’ or ‘Female’ in gender field and display the structure. Query:-**

alter table employee add constraint chk\_gender CHECK (gender in

('MALE','FEMALE'))



1. **Insert proper data (at least 5 appropriate records) in all the tables**

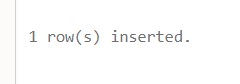
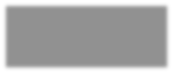
# Query:-

* **Department Table insert:-**

insert into Department values(10, 'IT', 'Bangalore')

insert into Department values(20, 'HR', 'Chennai') insert into Department values(30, 'Accounts', 'Bangalore') insert into Department values(40, 'Sales', 'Chennai') insert into Department values(50, 'Marketing', 'Bangalore')

insert into Department values(60, 'MBA', 'Bangalore')



* **Employee Table insert:-**

insert into employee values(100, 'Pankaj', '01-MAR2002','MALE',10, 'Bangalore', 'manager', 100000, 10,

'pankajsoni@email.com')

insert into employee values(103, 'Fahad', '14-FEB-

1999','MALE', 30, 'Bangalore', 'manager', 25000,

5,'fahad@email.com')

insert into employee values(104, 'Meet', '01-JAN-2003',

'MALE',20, 'Chennai', 'analyst', 25000, 2,

'Meet@email.com')

insert into employee values(105, 'Shruti', '18-MAR-

2002','FEMALE', 40, 'Chennai', 'designer', 30000,

8,'shrutia@email.com')

insert into employee values(102, 'Aniket', '01-MAR-

2000', 'MALE',20, 'Chennai', 'analyst', 25000, 2,

'aniket@email.com')

insert into employee values(106, 'Amna', '01-JAN-2004',

'MALE',20, 'Chennai', 'analyst', 4000, 2,

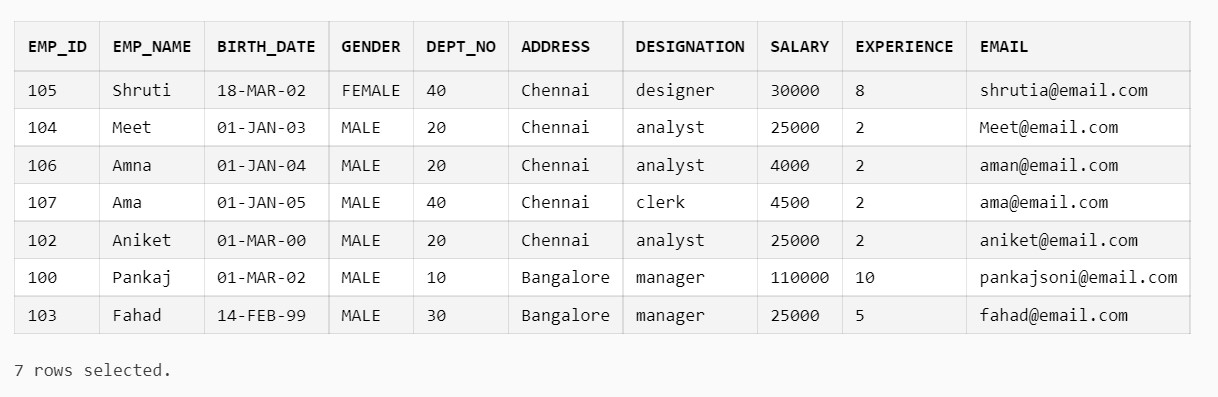
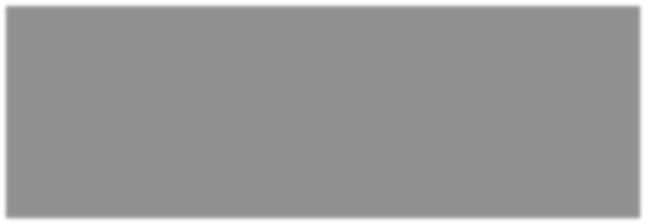
'aman@email.com')

insert into employee values(107, 'Ama', '01-JAN-2005', 'MALE',40, 'Chennai', 'clerk', 4500, 2, 'ama@email.com')

**EMPLOYEE TABLE :**

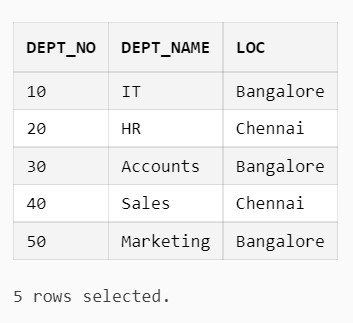
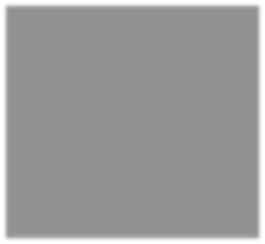
select \* from emplo

yee



# DEPARTMENT TABLE:-

select \* from department

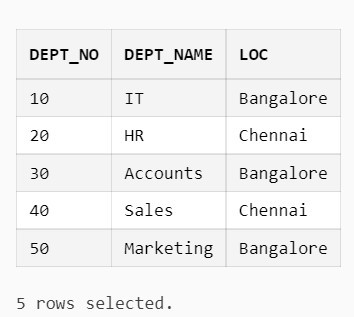
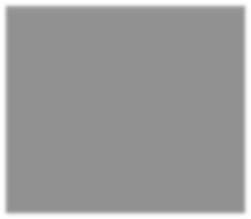


**6. List all records of each table in ascending order.**

# Query:-

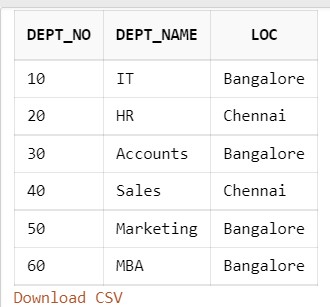
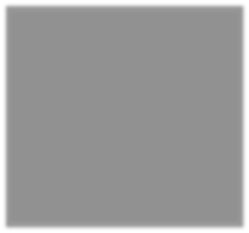
* **DEPARTMENT TABLE ASCENDING ORDER**

select \* from department order by dept\_no asc



* **EMPLOYEE TABLE ASCENDING ORDER**

select \* from department order by dept\_no asc



1. **Delete the department whose location is Ahmedabad. Query:-**

delete from department where loc = 'Ahmedabad'



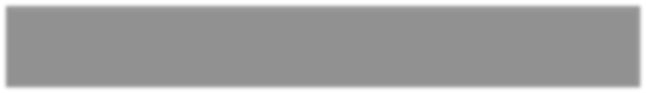
1. **Display female employee list. Query:-**

select \* from employee where gender = 'FEMALE'

**9.**

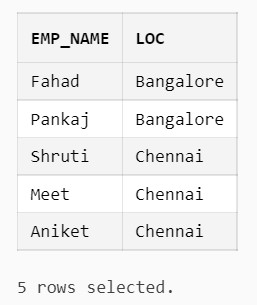
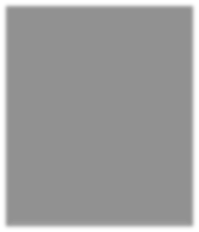
**Dis**

**play Departname wise employee Names**



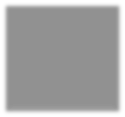
# Query:-

select emp\_name,loc from employee,department where department.dept\_no = employee.dept\_no order by loc



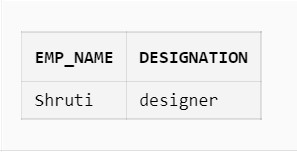
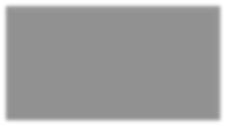
1. **Find the names of the employee who has a salary less than 5000 and greater than 2000. Query:-**

select emp\_name from employee where salary < 5000 and salary >2000



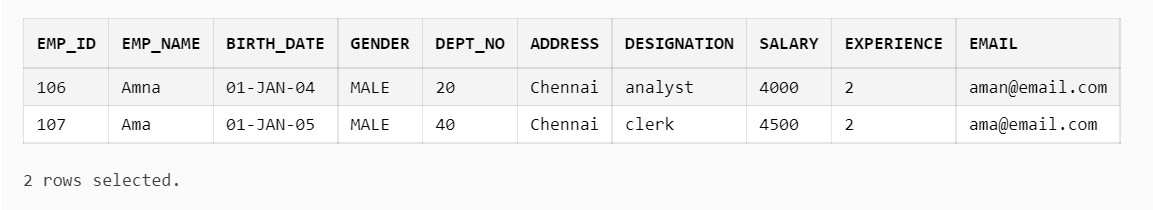
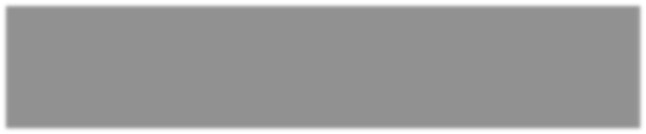
1. **Display the names and the designation of all female employees in descending order. Query:-**

select emp\_name,designation from employee where gender = 'FEMALE' order by emp\_name desc



## 12. Display the names of all the employees whose names start with ‘A’ and end with ‘A’. Query:-

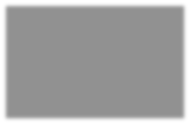
select \* from employee where emp\_name like 'A%a'



**13. Find the name of the employee and salary for those who had obtained minimum salary. Query:-**

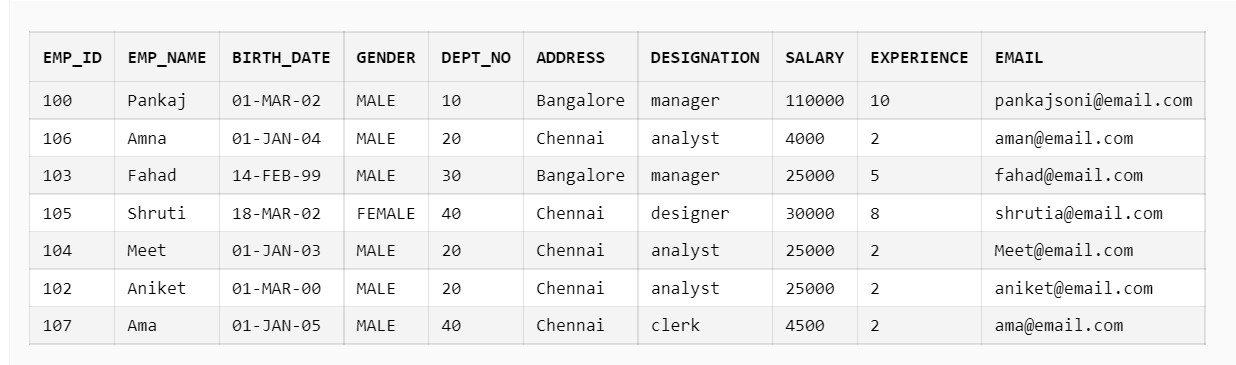
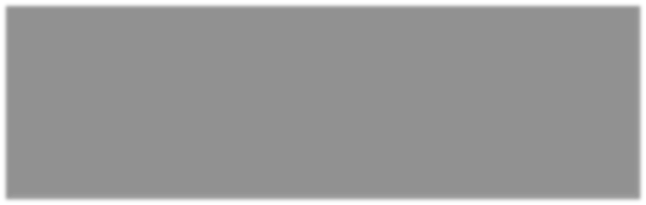
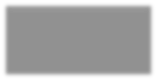
select emp\_name,salary from employee order by salary fetch

first 1 rows only



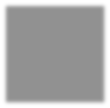
## 14. Add 10% raise in salary of all employees whose department is ‘IT’. Query:-

update employee set salary = salary + (salary\* 0.1) where emp\_id in (select emp\_id from employee,department where employee.dept\_no = department.dept\_no and dept\_name = 'IT')



## 15. Count the total number of employees of the ‘IT’ department. Query:-

select count(\*) "count" from employee,department where employee.dept\_no = department.dept\_no and dept\_name = 'IT'



**16. List all employees who were born in the current month.**

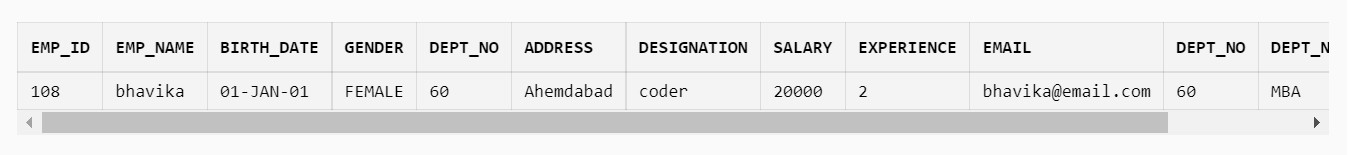
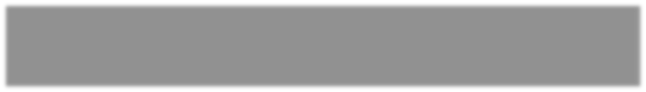
# Query:-

select \* from employee where extract(month from birth\_date) = extract(month from current\_date)

**17. Print the record of employee and dept table as “Employee works in department ‘MBA’.**

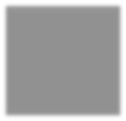
# Query:-

select \* from employee,department where employee.dept\_no = department.dept\_no and DEPT\_NAME = 'MBA'



## 18. List names of employees who are fresher’s (less than 1 year of experience). Query:-

select emp\_name from employee where experience <= 1

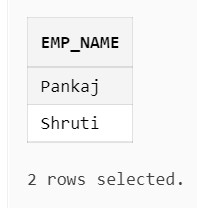
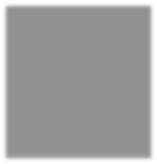


**19. List department wise names of employees who have more than 5 years of experience.**

# Query:-

select emp\_name from employee,department where employee.dept\_no

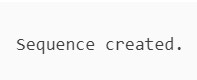
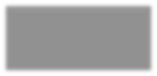
= department.dept\_no and experience > 5 order by dept\_name



**20. Create Sequence to generate department ID**

# Query:-

CREATE SEQUENCE dept\_id\_seq start with 10 increment by 10 minvalue 10 maxvalue 1000 cycle



**21. List department having no employees**

# Query:-

select \* from department where dept\_no not in (select dept\_no from employee group by dept\_no having count(dept\_no)>0)

**SET 3**

**STUDENT (rollno, name, class, birthdate)**

**COURSE (courseno, coursename, max\_marks, pass\_marks)**

**SC (rollno, courseno, marks)**

**1.Create the above three tables along with key constraints.**

1. **Write an Insert script for insertion of rows with substitution variables and insert appropriate data.**
2. **Add a constraint that the marks entered should strictly be between 0 and 100. 4. While creating SC table, composite key constraint was forgotten. Add the composite keynow.**

**5. Display details of student who takes ‘Database Management System’ course. 6. Display the names of students who have scored more than 70% in Computer Networksand have not failed in any subject.**

1. **Display the average marks obtained by each student.**
2. **Select all courses where passing marks are more than 30% of average maximum mark.**
3. **Display details of students who are born in 1980 or 1982.**
4. **Create a view that displays student courseno and its corresponding marks.**

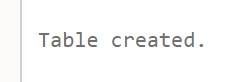
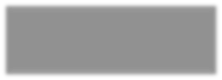
1. **Create the above three tables along with key constraints. Query:-** 
   * **Student table create.**

create table Student

(

rollno int primary key, sname varchar(20), cls varchar(6), birthdate date

)



* + **Course table create.**

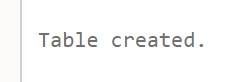
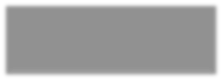
create table Course

(

courseno int primary key, coursename varchar(50), max\_marks int,

pass\_marks int

)



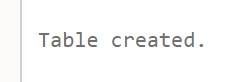
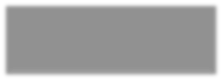
* + **SC table create.**

create table SC

( rollno int, courseno int, marks int,

FOREIGN KEY (rollno) REFERENCES Student(rollno),

FOREIGN KEY (courseno) REFERENCES Course(courseno) )

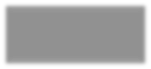


1. **Write an Insert script for insertion of rows with substitution variables and insert appropriate data.**

# Query:-

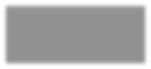
* **Student table insert.**

insert into Student values(1,'PankajSoni','MCA','01-mar-2002') insert into Student values(2,'RajSoni','MCA','01-jan-2000') insert into Student values(3,'Bhavika','MCA','01-feb-2001') insert into Student values(4,'Krishna','MBA','21-mar-2000') insert into Student values(5,'Ashta','MCOM','17-jun-2001') insert into Student values(6,'Fahad','MCA','01-mar-1982') insert into Student values(7,'Sahdev','MCA','01-mar-1980')



* **Course table insert.**

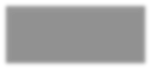
insert into Course values(1,'Database Management System',100,40) insert into Course values(2,'Computer Networks',100,38) insert into Course values(3,'SE',100,30) insert into Course values(4,'JAVA',100,40) insert into Course values(5,'OOP',100,45)



* **SC table insert.**

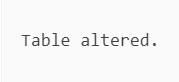
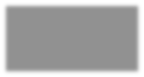
insert into SC values(2,1,75) insert into SC values(1,1,80) insert into SC values(3,1,70) insert into SC values(4,1,85) insert into SC values(1,5,90)

insert into SC values(3,3,95) insert into SC values(2,2,60)



1. **Add a constraint that the marks entered should strictly be between 0 and 100.** **Query:-**

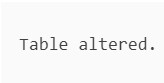
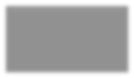
ALTER TABLE sc ADD CHECK (marks>=0 and marks<=100)



1. **While creating the SC table, composite key constraint was forgotten. Add the composite keynow.**

# Query:-

alter table sc ADD PRIMARY KEY (rollno,courseno)



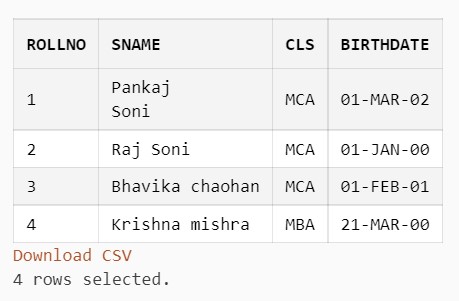
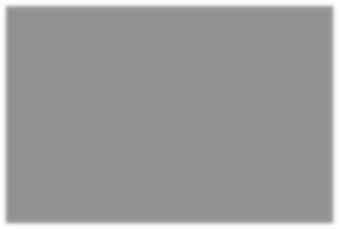
**5. Display details of students who take the ‘Database Management System’ course.**

# Query:-

select Student.\* from Student,Course,SC where Student.rollno =

SC.rollno and Course.courseno = SC.courseno and coursename =

'Database Management System'



**6. Display the names of students who have scored more than 70% in Computer Networksand have not failed in any subject.**

# Query:-

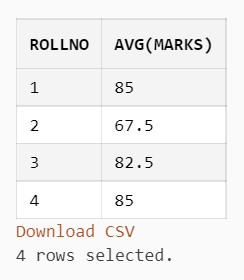
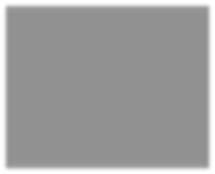
select Student.SNAME from Student,Course,SC

where Student.rollno = SC.rollno and Course.courseno = SC.courseno and (sc.marks/course.max\_marks) > 0.7 and coursename = 'Computer Networks' and Student.rollno not in (select Student.rollno from Student,SC where Student.rollno = SC.rollno and sc.marks <= course.pass\_marks)

1. **Display the average marks obtained by each student. Query:-**

select Student.rollno,avg(marks) from Student,Course,SC

where Student.rollno = SC.rollno and Course.courseno = SC.courseno group by Student.rollno order by Student.rollno

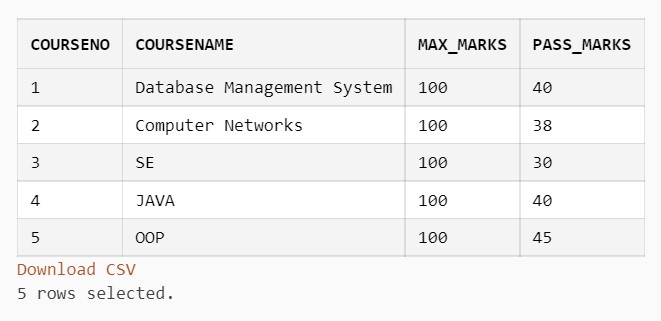
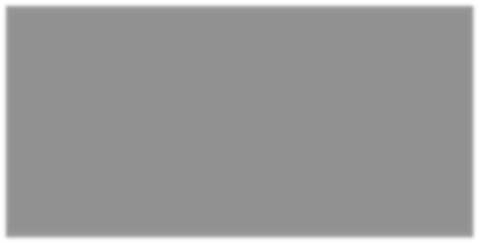


1. **Select all courses where passing marks are more than 30% of average maximum mark.**

# Query:-

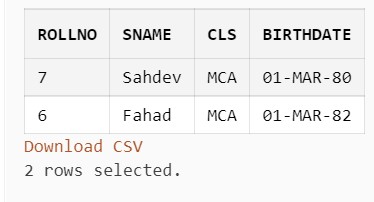
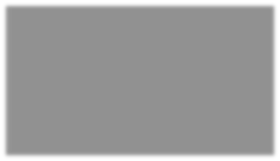
select \* from COURSE where PASS\_MARKS/(select avg(max\_marks)

from course) >= 0.3



**9. Display details of students who were born in 1980 or 1982. Query:-**

select \* from student where EXTRACT(YEAR FROM BIRTHDATE) = 1980 or EXTRACT(YEAR FROM BIRTHDATE) = 1982



**9. Create a view that displays student courseno and its corresponding marks.**

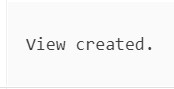
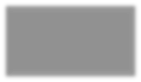
# Query:-

CREATE VIEW displayStud AS

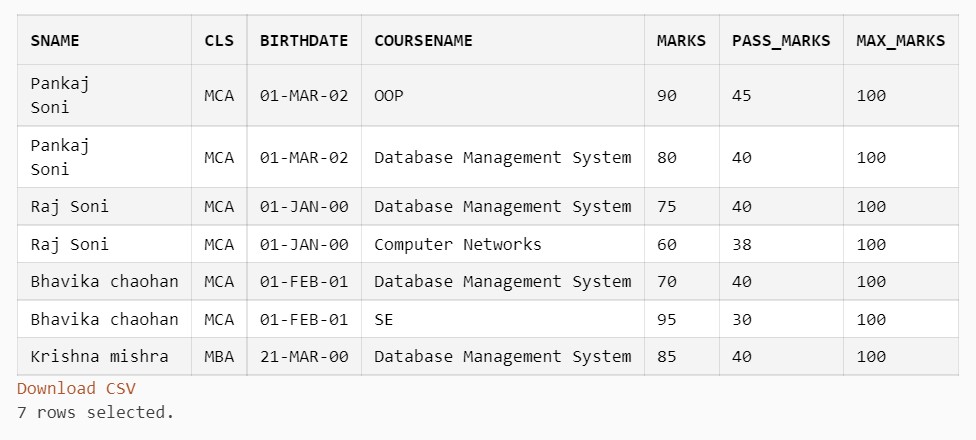
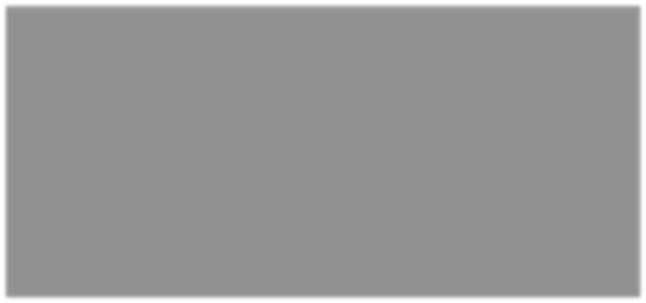
select

SNAME,CLS,BIRTHDATE,COURSENAME,MARKS,PASS\_MARKS,MAX\_MARKS from course,SC,STUDENT where SC.COURSENO = course.COURSENO and

STUDENT.ROLLNO = SC.ROLLNO\



select \* from displayStud



# SET-4

**Create the database COMPANY and create given tables with all necessary constraints such as primary key, foreign key, unique key, not null and check constraints.**

**EMPLOYEE (emp\_id, emp\_name, birth\_date, gender, dept\_no, address, designation, salary, experience, email)**

**DEPART (dept\_no, dept\_name, total\_employees, location) PROJECT (proj\_id, type\_of\_project, status, start\_date, emp\_id)**

**Insert proper data (at least 5 appropriate records) in all the tables.**

1. **Delete the department whose total number of employees less than 1.**
2. **Display the names and the designation of all female employee in descending order.**
3. **Display the names of all the employees who names starts with ‘A’ ends with ‘A’.**
4. **Find the name of employee and salary for those who had obtain minimum salary.**
5. **Add 10% raise in salary of all employees whose department is ‘CIVIL’.**
6. **Count total number of employees of ‘MCA’ department.**
7. **List all employees who born in the current month.**
8. **Print the record of employee and dept table as “Employee works in department ‘CE’.**
9. **List names of employees who are fresher’s(less than 1 year of experience).**
10. **List department wise names of employees who has more than 5 years of experience. 11. Write a function which will display total number of projects based on status (pass status as parameter).**
11. **Write a procedure that will display list of projects which is going to start today.**
12. **Write a trigger which do not allow insertion/updation/deletion into Project table if status type is ‘pending’**

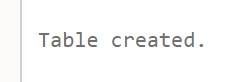
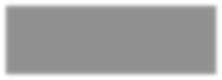
• **DEPARTMENT TABLE CREATE AND INSERT:-**

create table depart

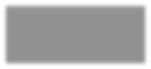
(

dept\_no int primary key, dept\_name varchar(20), total\_employees int, loc varchar(50)

)



insert into depart values(10, 'CIVIL', 10, 'New York') insert into depart values(20, 'MCA', 20, 'New York') insert into depart values(30, 'CE', 30, 'New York') insert into depart values(40, 'Operations', 40, 'Boston') insert into depart values(50, 'Sales',10, 'Boston') insert into depart values(60, 'MBA',10, 'Boston')



• **EMPLOYEE TABLE CREATE AND INSERT:-**

create table employee (

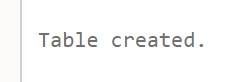
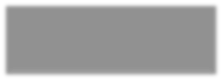
emp\_id number PRIMARY KEY,

emp\_name varchar(30), birth\_date date, gender char(7),

dept\_no number references depart(dept\_no), address varchar(50), designation varchar(20),

salary number constraint chk\_sal CHECK (salary>0), experience varchar(30),

email varchar(30)constraint chk\_emp\_email CHECK (email like '%@%' and email like '%.%'))



insert into employee values(100, 'Panakj', '01-MAR-2002', 'MALE',10, 'Bangalore', 'manager', 10000, 10, 'pankaj@email.com')

insert into employee values(101, 'Fahad', '24-JUL-1999',

'MALE', 10, 'Bangalore', 'coder', 20000, 10, 'fahad@email.com')

insert into employee values(103, 'Aniket', '14-FEB-1999',

'MALE', 30, 'Bangalore', 'manager', 25000, 5,

'Aniketh@email.com')

insert into employee values(104, 'Raj', '01-JAN-1999', 'MALE',

20, 'Chennai', 'analyst', 25000, 2, 'Raj@email.com')

insert into employee values(105, 'Bhavika', '18-MAR-1985',

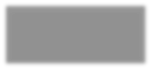
'FEMALE', 40, 'Chennai', 'designer', 3000, 8,

'bhavika@email.com')

insert into employee values(106, 'Ashata', '18-DEC-1995', 'FEMALE', 40, 'Chennai', 'tester', 9000, 1, 'ashata@email.com')

insert into employee values(107, 'Shruti', '18-DEC-1995',

'FEMALE', 50, 'Chennai', 'tester', 9000, 1, 'shruti@email.com')



• **PROJECT TABLE CREATE AND INSERT:-**

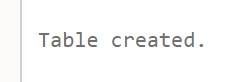
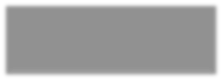
create table project

(

proj\_id int primary key, type\_of\_project varchar(20), status varchar(20), start\_date date,

emp\_id int references employee(emp\_id)

)



insert into project values(10, 'CIVIL', 'completed', '21-MARCH-

2020', 100) insert into project values(20, 'MCA', 'completed', '01-JAN-2020',

101)

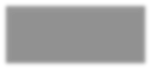
insert into project values(30, 'CE', 'pending', '12-JUL-

2021',103) insert into project values(40, 'Operations', 'pending', '23-AUG-

2021', 104) insert into project values(50, 'Sales', 'pending', '12-OCT-2021',

105) insert into project values(60, 'MBA', 'pending', '11-JUN-2019',

106)



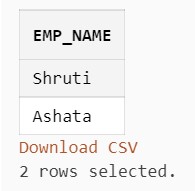
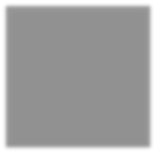
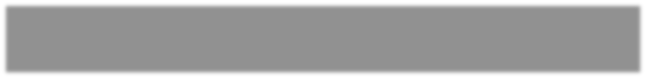
**1. Delete the department whose total number of employees is less than 1. Query:-**

delete from depart where total\_employees <1

|  |  |
| --- | --- |
| **ROLL No :-A-115/D-152 Name:-Himansu Kumar Jena**  **CLASS :-Mca Sem-1 Div:-B**     1. **Display the names and the designation of all female employees in descending order.**   **Query:-**  select emp\_name,designation from employee where gender =  'FEMALE' order by emp\_name desc             1. **Display the names of all the employees whose names start with ‘A’ and end with**   **‘A’.**  **Query:-**  select \* from employee where emp\_name like 'A%a' | |
| **4.**            **26 |** P a g e |  |
| **Find the name of the employee and salary for those who had obtained minimum salary.**  **Query:-**  select emp\_name,salary from employee order by salary fetch first 4 row only |

|  |  |
| --- | --- |
| **ROLL No :-A-115/D-152 Name:-Himansu Kumar Jena**  **CLASS :-Mca Sem-1 Div:-B**        **5. Add 10% raise in salary of all employees whose department is ‘CIVIL’. Query:-**  update employee  set salary = salary + (salary\* 0.1) where emp\_id in (select emp\_id from employee,depart where employee.dept\_no = depart.dept\_no and dept\_name = 'CIVIL') | |
| **6. C**      **27 |** P a g e |  |
| **ount total number of employees of ‘MCA’ department.**  **Query:-**  select count(\*) "count" from employee,depart where employee.dept\_no = depart.dept\_no and dept\_name = 'MCA' |

**7. List all employees who were born in the current month.**



## Query:-

select \* from employee where extract(month from birth\_date) = extract(month from current\_date)

**8. Print the record of employee and dept table as “Employee works in department**

**‘CE’.**

## Query:-

select \* from employee,depart where employee.dept\_no = depart.dept\_no and DEPT\_NAME = 'CE'

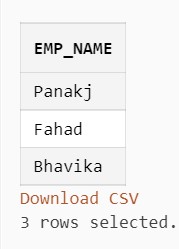
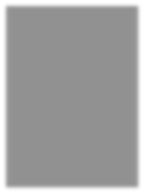
### 9. List names of employees who are fresher’s(less than 1 year of experience). Query:-

select emp\_name from employee where experience <= 1

**10. List department wise names of employees who have more than 5 years of experience.**

## Query:-

select emp\_name from employee,depart where employee.dept\_no = depart. dept\_no and experience > 5 order by dept\_name



# SET 5

**Create the database STUD and create given tables with all necessary constraints such as primary key, foreign key, unique key, not null and check constraints.**

**HOSTEL (HNO, HNAME, HADDR, TOTAL\_CAPACITY, WARDEN)**

**ROOM (HNO, RNO, RTYPE, LOCATION, NO\_OF\_STUDENTS, STATUS)**

**CHARGES (HNO, RTYPE, CHARGES)**

**STUDENT (SID, SNAME, MOBILE-NO, GENDER, FACULTY, DEPT, CLASS, HNO, RNO)**

**FEES (SID, FDATE, FAMOUNT)**

**The STATUS field tells us whether the room is occupied or vacant.**

**The charges represent the term fees to be paid half yearly.**

**A student can pay either the annual fees at one time or the half yearly fees twice a year.**

**Insert proper data (at least 5 appropriate records) in all the tables.**

1. **Display the total number of rooms that are presently vacant.**
2. **Display number of students of each faculty and department wise staying in each hostel.**
3. **Display hostels, which have at least one single-seated room.**
4. **Display the warden name and hostel address of students of Computer Science department.**
5. **Display those hostel details where single seated or double-seated rooms are vacant.**
6. **Display details of hostels occupied by medical students.**
7. **Display hostels, which are totally occupied to its fullest capacity.**
8. **List details about students who are staying in the double-seated rooms of Chanakya Hostel.**
9. **Display the total number of students staying in each room type of each hostel.**
10. **Display details about students who have paid fees in the month of Nov. 2017.**
11. **For those hostels where total capacity is more than 300, display details of students studying in Science faculty.**
12. **Display hostel details where there are at least 10 vacant rooms.**
13. **Display details of students who have still not paid fees.**
14. **Display those hostels where single-seated room is the costliest.**
15. **Write a trigger which do not allow to insert or update student record if mobile\_no length is less than 10 digits.**
16. **Write a PL/SQL block which will count total number of student’s gender wise. Male**

**Students: 999 students Female Students: 999 students**

**Do as directed:**

* + **CREATE TABLE HOSTEL:**

CREATE TABLE HOSTEL(

HNO INT PRIMARY KEY,

HNAME VARCHAR(20),

HADDR VARCHAR(20),

TOTAL\_CAPACITY INT,

WARDEN VARCHAR(20)

)

* + **CREATE TABLE ROOM:**

CREATE TABLE ROOM (

RNO INT,

HNO INT,

RTYPE VARCHAR(20),

RLOCATION VARCHAR(20),

NO\_OF\_STUDENTS INT,

STATUS VARCHAR(20),

CONSTRAINT p\_k PRIMARY KEY(RNO,HNO),

FOREIGN KEY (HNO) REFERENCES HOSTEL(HNO),

CONSTRAINT ck\_status check(status in ('occupied','vacant'))

)

* **CREATE TABLE CHARGES:**

CREATE TABLE CHARGES (

HNO INT PRIMARY KEY,

RTYPE VARCHAR(20),

CHARGES INT,

FOREIGN KEY (HNO) REFERENCES HOSTEL(HNO)

)

* **CREATE TABLE STUDENT:**

CREATE TABLE STUD (

SID INT PRIMARY KEY,

SNAME VARCHAR(20),

MOBILE\_NO VARCHAR(10),

GENDER VARCHAR(1),

FACULTY VARCHAR(20),

DEPT VARCHAR(20),

SCLASS VARCHAR(20),

HNO INT,

RNO INT,

FOREIGN KEY (HNO) REFERENCES HOSTEL(HNO),

FOREIGN KEY (RNO,HNO) REFERENCES ROOM(RNO,HNO))

* **CREATE TABLE FEES:**

CREATE TABLE FEES (

SID INT PRIMARY KEY,

FDATE DATE,

FAMOUNT INT,

FOREIGN KEY (SID) REFERENCES STUD(SID)

)

* **INSERT TABLE HOSTEL:**

insert into hostel values(1,'SCET HOSTEL','SURAT',500,'PANKAJ') insert into hostel values(2,'SVNIT HOSTEL','VAPI',100,'FAHAD') insert into hostel values(3,'VEGOLE HOSTEL','SURAT',500,'RAJ') insert into hostel values(4,'VNSGU HOSTEL','VLSAD',500,'ASHISH') insert into hostel values(5,'DIAMANDHOSTEL','SURAT',150,'SAHDEV') insert into hostel values(6,'GIRLS HOSTEL','SURAT',200,'SHRUTI')

* **INSERT TABLE ROOM:**

insert into room values(1,1,'SINGLE','SURAT',50,'vacant') insert into room values(2,1,'SHARING','SURAT',100,'vacant') insert into room values(2,2,'SINGLE','VAPI',100,'occupied') insert into room values(2,3,'SHARING','SURAT',300,'vacant') insert into room values(4,5,'SINGLE','SURAT',150,'vacant') insert into room values(2,6,'SHARING','SURAT',300,'occupied') insert into room values(5,4,'SINGLE','VALSAD',80,'vacant')

* **INSERT TABLE CHARGES:**

insert into charges values(1,'SINGLE',10000) insert into charges values(2,'SHARING',4400) insert into charges values(3,'SINGLE',11000)insert into charges values(4,'SINGLE',13000) insert into charges values(5,'SHARING',3000) insert into charges values(6,'SINGLE',4000)

* **INSERT TABLE STUDENT:**

insert into stud values(1,'PANKAJ','7043266874','M','COMPUTER

SCIENCE','MCA','SEM-1',1,2)

insert into stud values(2,'FAHAD','7056543210','M','COMPUTER

SCIENCE','MCA','SEM-1',2,2)

insert into stud values(3,'RAJ','9076576210','M','COMPUTER

SCIENCE','MCA','SEM-2',1,2)

insert into stud values(4,'SHRUTI','9898765544','F','MBBS','MEDICAL','SEM-2',3,2)

insert into stud values(5,'BHAVIKA','9034566590','F','COMMERCE','BCOM','SEM-

3',4,5) insert into stud values(6,'AASHATA','7834566890','F','COMMERCE','BCOM','SEM-

2',5,4) insert into stud

values(7,'ANIKET','8034566591','M','ARTS','BBA','SEM-4',1,1)

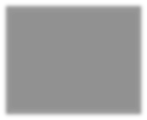
• **INSERT TABLE FEES:**

insert into fees values(1,'02-dec-2021',500) insert into fees values(2,'12-dec-2021',1000) insert into fees values(3,'15-NOV-2017',1000) insert into fees values(4,'01-MAR-2021',110) insert into fees values(5,'08-dec-2021',700) insert into fees values(6,'11-NOV-2019',1000)

1. **Display the total number of rooms that are presently vacant. Query:-**

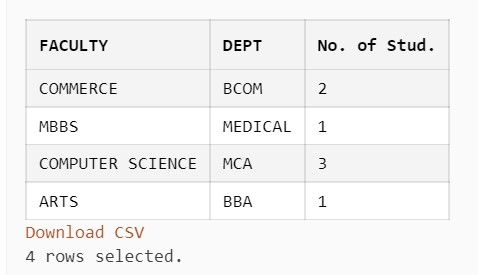
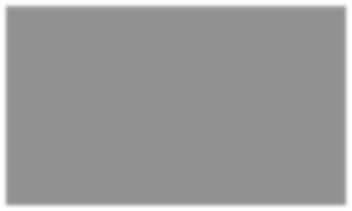
SELECT COUNT(\*) AS "vacant rooms" FROM room WHERE STATUS =

'vacant'



1. **Display number of students of each faculty and department wise staying in each hostel. Query:-**

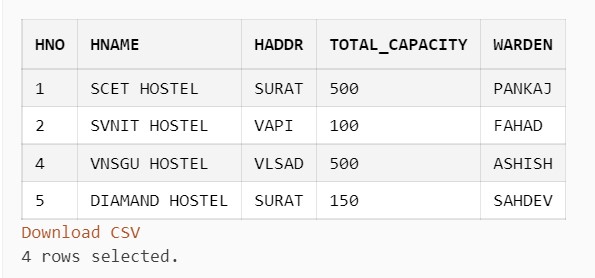
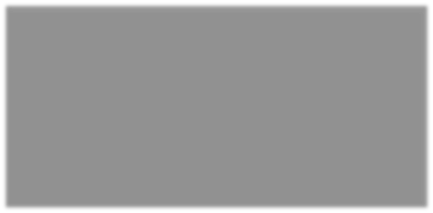
select faculty,dept,count(\*) as "No. of Stud." from stud group by faculty,dept



1. **Display hostels, which have at least one single-seated room. Query:-**

SELECT \* FROM HOSTEL WHERE HNO IN (SELECT HNO FROM ROOM WHERE

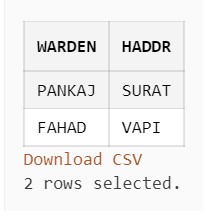
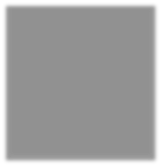
RTYPE = 'SINGLE')



1. **Display the warden name and hostel address of students of Computer Science FACULTY.**

## Query:-

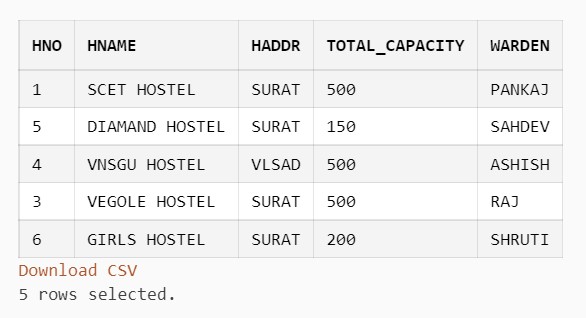
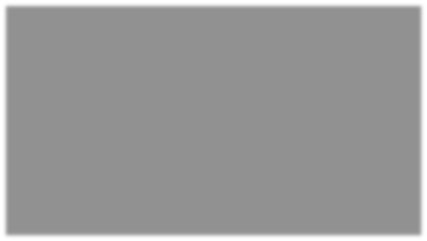
select WARDEN,HADDR from hostel where HNO IN (SELECT HNO FROM stud WHERE FACULTY='COMPUTER SCIENCE')



1. **Display those hostel details where single seated or double-seated rooms are vacant. Query:-**

select \* from hostel where HNO IN (SELECT HNO FROM ROOM WHERE

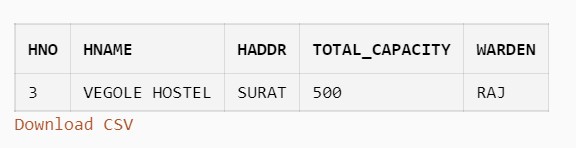
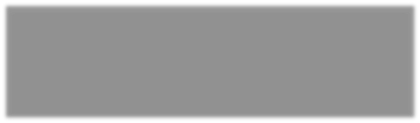
STATUS = 'vacant' AND RTYPE = 'SINGLE' OR RTYPE='SHARING')



1. **Display details of hostels occupied by medical students. Query:-**

select \* from hostel where HNO IN (SELECT HNO FROM STUD WHERE

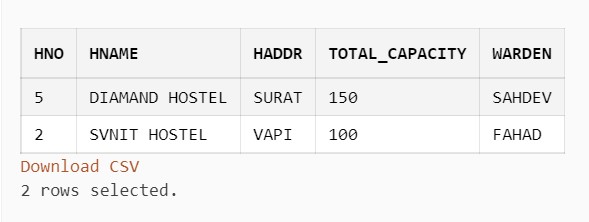
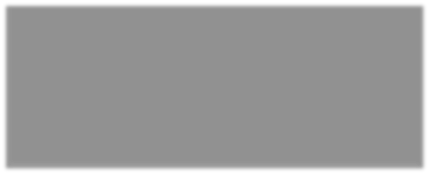
DEPT='MEDICAL')



1. **Display hostels, which are totally occupied to its fullest capacity. Query:-**

SELECT hostel.\* FROM ROOM,HOSTEL WHERE ROOM.HNO = HOSTEL.HNO AND

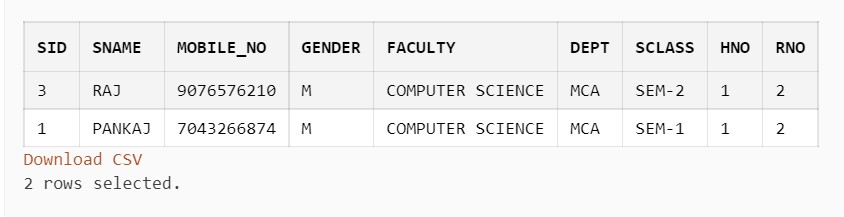
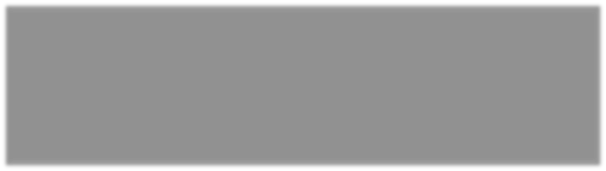
HOSTEL.TOTAL\_CAPACITY = ROOM.NO\_OF\_STUDENTS



1. **List details about students who are staying in the double-seated rooms of SCET HOSTEL .**

## Query:-

SELECT stud.\* FROM HOSTEL,STUD,ROOM WHERE STUD.HNO = HOSTEL.HNO and stud.rno = room.rno and room.hno = hostel.hno and HNAME = 'SCET HOSTEL' and rtype = 'SHARING'



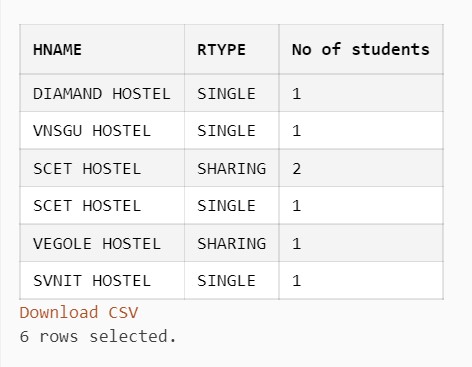
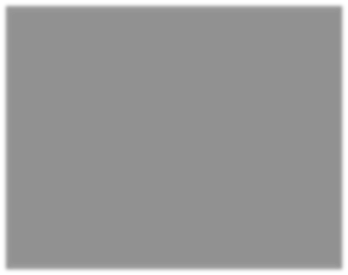
### 9. Display the total number of students staying in each room type of each hostel. Query:-

SELECT HOSTEL.HNAME,ROOM.RTYPE,COUNT(\*) AS "No of students"

FROM ROOM,STUD,HOSTEL WHERE STUD.HNO = ROOM.HNO AND

STUD.RNO = ROOM.RNO AND ROOM.HNO = HOSTEL.HNO GROUP BY

ROOM.RTYPE,HOSTEL.HNAME

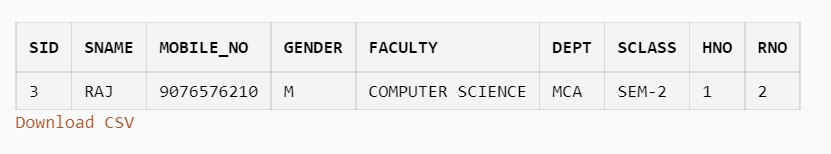
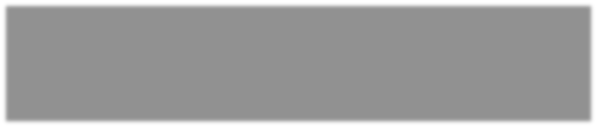


### 10. Display details about students who have paid fees in the month of Nov. 2017. Query:-

SELECT STUD.\* FROM STUD,FEES WHERE STUD.SID = FEES.SID

AND EXTRACT(YEAR FROM FDATE) = 2017 AND EXTRACT(MONTH FROM

FDATE) = 11



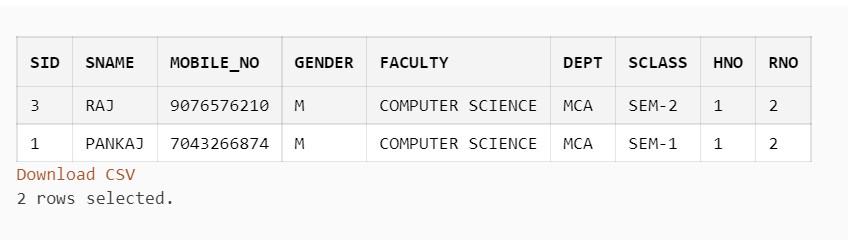
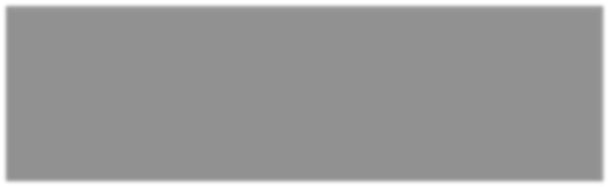
**11. For those hostels where total capacity is more than 300, display details of students studying in the Science faculty.**

## Query:-

SELECT STUD.\* FROM STUD,HOSTEL WHERE STUD.HNO =

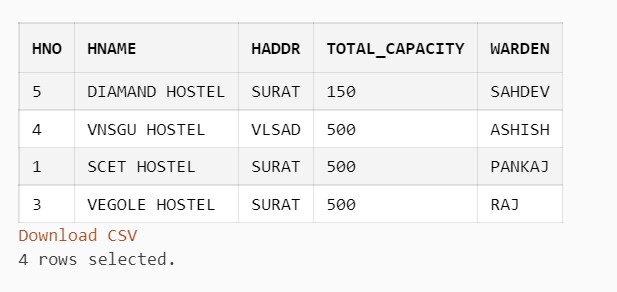
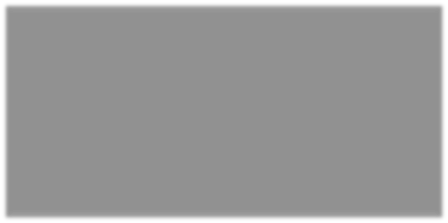
HOSTEL.HNO AND HOSTEL.TOTAL\_CAPACITY > 300 AND FACULTY like

'%SCIENCE%'



1. **Display hostel details where there are at least 10 vacant rooms. Query:-**

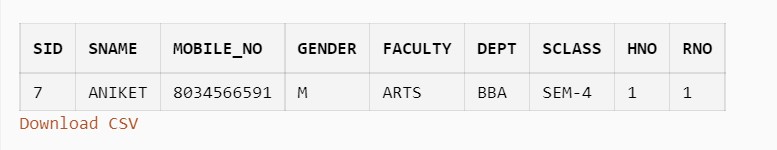
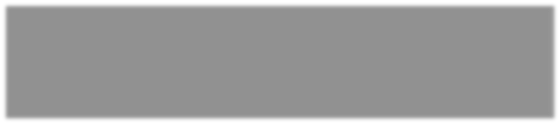
select \* from hostel where hno in (select hno from room where status = 'vacant' group by hno having count(hno)<10)



1. **Display details of students who have still not paid fees**

## Query:-

select \* from stud where sid not in (select sid from fees)



**14. Display those hostels where a single-seated room is the costliest.**

## Query:-

select hostel.\* from HOSTEL,CHARGES where HOSTEL.HNO = CHARGES.HNO and RTYPE = 'SINGLE' order by CHARGES desc fetch

first 1 rows only

