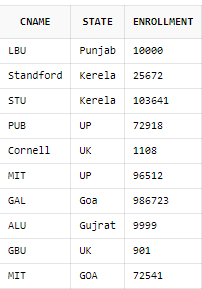
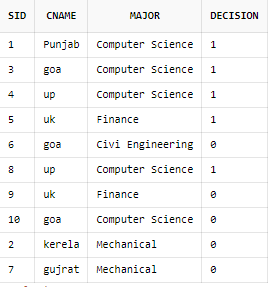
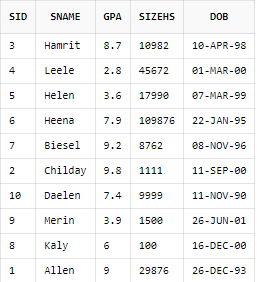
**Q2. Create the following tables and perform the queries that follow.**

**College Apply**



**Student**



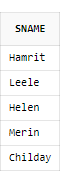
1. Select sname,dob from student;



1. Select sname from student where gpa>3.7;



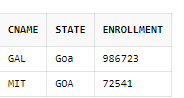
1. Select sname from student where sizeHS>=1000 AND DOB>'31-Dec-1996';



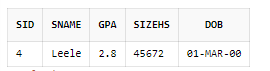
1. Select sname from student where GPA>2.9 AND GPA<3.9;



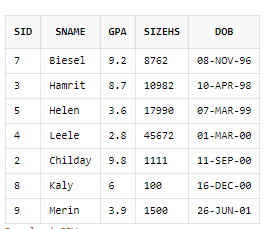
1. Select \* from college where state='goa';



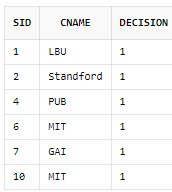
1. select \* from student where gpa>2.0 AND gpa<3.5;



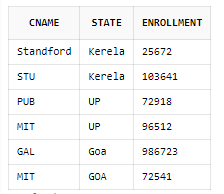
1. select \* from student where dob>'01-JUL-1996' order by DOB ;



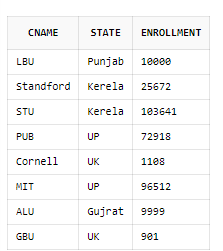
1. Select sid,cname,decision from apply where decision =1;



1. Select \* from college where enrollment>10001;



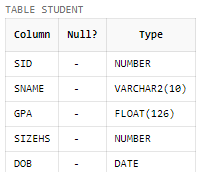
1. select \* from college where NOT state='goa';



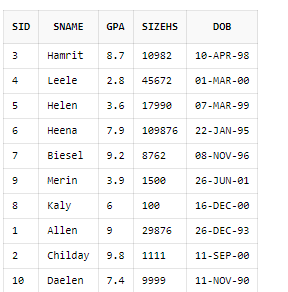
1. select sname from student where sizeHS>17000 AND GPA<3.8;



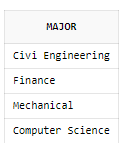
1. desc student;



1. select \* from student;



1. SELECT DISTINCT major from apply;



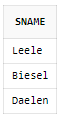
1. select sname from student where sname like '\_\_\_\_\_';



1. Select sname from student where sname like 'H\_\_\_\_';



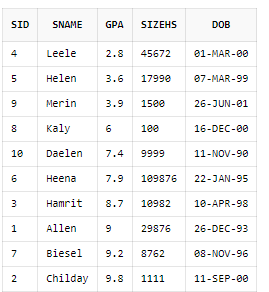
1. Select sname from student where sname like '\_\_e\_e%';



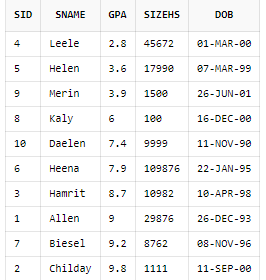
1. Select sname from student where sname like '%y';



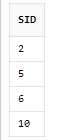
1. Select \* from student order by GPA;



1. Select \* from student order by GPA ,dob desc ;

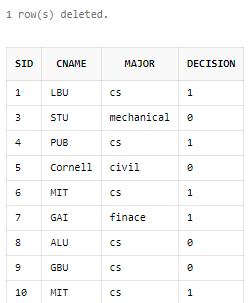


1. Select Sid from apply where cname='standford' OR cname='cornell' OR cname='mit';



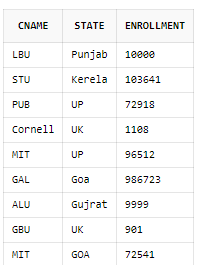
1. delete from apply where lower(cname)='standford';

Select \* from apply;



1. delete from college where lower(cname)='standford';

Select \* from college;



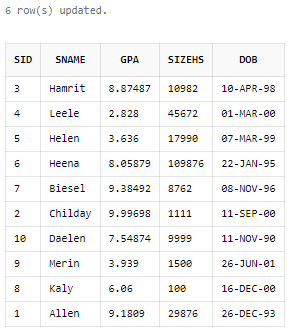
1. Update student set gpa=gpa\*0.01+gpa;

Select \* from student;



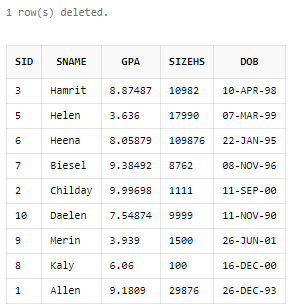
1. Update student set gpa=gpa\*0.01+gpa where gpa>7 and sizehs>500;

Select \* from student;



1. delete from student where GPA<3.2;

Select \* from student;



**Q3.Execute the following queries:**

**1.** **candidate (candidate\_id, candidate\_name, class, city, fee, DoB)**

a) create table candidate(

candidate\_idint NOT NULL PRIMARY KEY,

candidate\_namevarchar2(20),

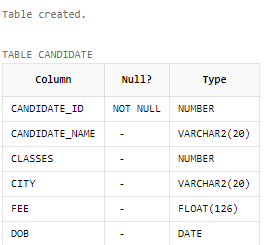
classesint,

city varchar(20),

fee float,

DoB DATE);

desc candidate;



b) insert into candidate values(1001,'Ritik',8,'Dehradun',1888,'3-MAR-2000');

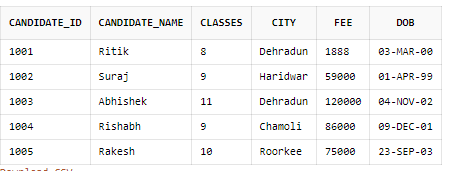
insert into candidate values(1002,'Suraj',9,'Haridwar',59000,'1-APR-1999');

insert into candidate values(1003,'Abhishek',11,'Dehradun',120000,'4-NOV-2002');

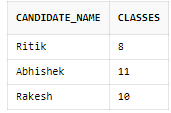
insert into candidate values(1004,'Rishabh',9,'Chamoli',86000,'9-DEC-2001');

insert into candidate values(1005,'Rakesh',10,'Roorkee',75000,'23-SEP-2003');

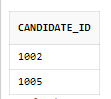
Select \* from candidate;



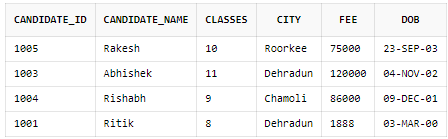
c) Select candidate\_name,classes from candidate where city='dehradun' OR city='roorkee';



d) Select candidate\_id from candidate where fee>50000 AND fee<85000;



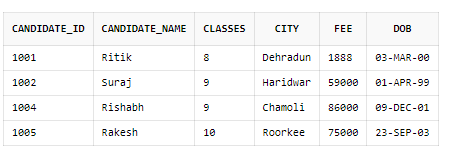
e) Select \* from candidate where dob>'10-JAN-2000' order by dob desc;



f) delete from candidate where candidate\_name like '%b%k' ;



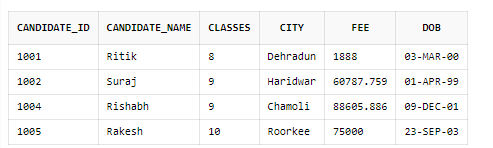
Select \* from candidate;



g) update candidate set fee=fee\*0.01+fee where classes=9;



Select \* from candidate;



h) Alter table candidate add GPA float;



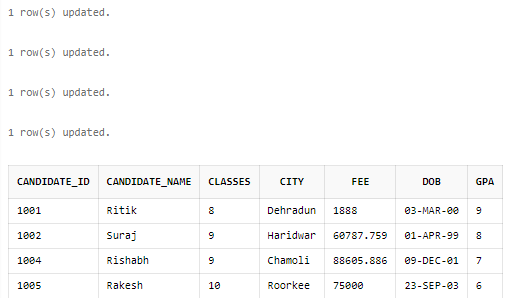
update candidate set gpa=9 where candidate\_id=1001;

update candidate set gpa=8 where candidate\_id=1002;

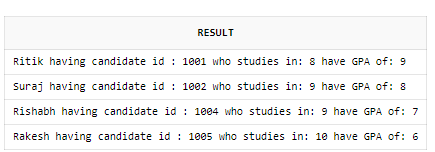
update candidate set gpa=7 where candidate\_id=1004;

update candidate set gpa=6 where candidate\_id=1005;

Select \* from candidate;



i)Select (candidate\_name || ' having candidate id : ' || candidate\_id || ' who studies in: '|| classes|| ' have GPA of: '|| GPA) as result from candidate ;



**2.** **Database Schema for a Employee-Detail**

**employee(emp\_id, emp\_name, dateofbirth, designation, department, salary, gender).**

a) create table employee(

emp\_idint NOT NULL PRIMARY KEY,

emp\_name varchar(20),

dateofbirth DATE,

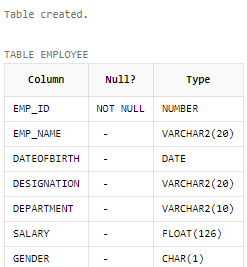
designation varchar2(20),

department varchar(10),

salary float,

gender char);

desc employee;



b) insert into employee values(1,'Akshay','20-JAN-1965','Tester','CS',80000,'M');

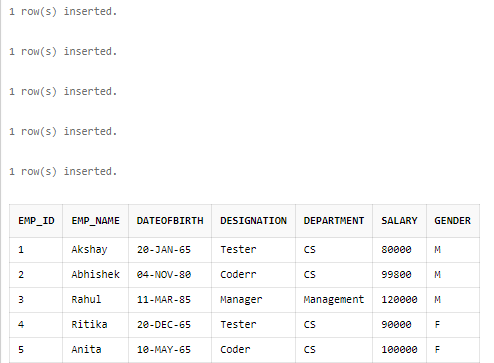
insert into employee values(2,'Abhishek','4-NOV-1980','Coderr','CS',99800,'M');

insert into employee values(3,'Rahul','11-MAR-1985','Manager','Management',120000,'M');

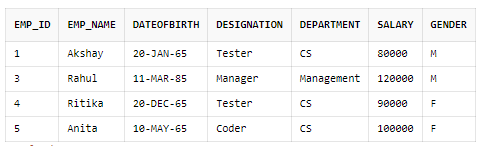
insert into employee values(4,'Ritika','20-DEC-1965','Tester','CS',90000,'F');

insert into employee values(5,'Anita','10-MAY-1965','Coder','CS',100000,'F');

select \* from employee;



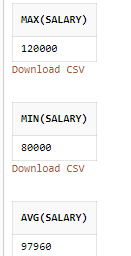
c) Select \* from employee where NOT emp\_name like '%b%';



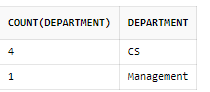
d)select max(salary) from employee ;

select min(salary) from employee ;

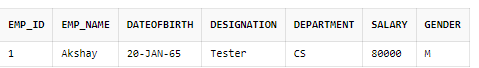
selectavg(salary) from employee ;



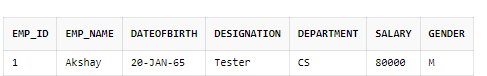
e) select count(department),department from employee group by department;



f) Select \* from employee where gender='M' AND salary<(Select avg(salary)from employee where gender='F') ;



g) Select \* from employee where dateofbirth=(select min(dateofbirth) from employee where dateofbirth<'1-JAN-1966' AND dateofbirth>'31-DEC-1964');



**3.** **Database Schema for a customer-sale scenario**

**Customer(Cust id : integer, cust\_name: string)**

**Item(item\_id: integer, item\_name: string, price: integer)**

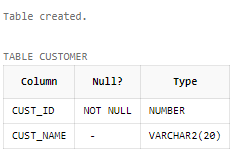
**Sale(bill\_no: integer, bill\_date: date, cust\_id: integer, item\_id: integer, qty\_sold: integer)**

a) create table customer(

cust\_idint NOT NULL PRIMARY KEY,

cust\_namevarchar2(20));

desc customer;



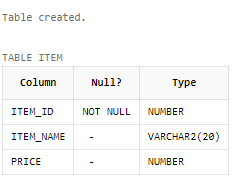
create table item(

item\_idint NOT NULL PRIMARY KEY,

item\_namevarchar2(20),

priceint);

desc item;



create table sale(

bill\_noint NOT NULL PRIMARY KEY,

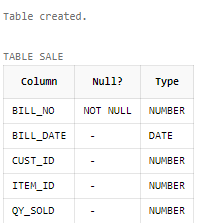
bill\_date DATE,

cust\_idint,

item\_idint,

qy\_soldint);

desc sale;



b) insert into customer values(1,'Allen');

insert into customer values(2,'Manan');

insert into customer values(3,'Rishabh');

insert into customer values(4,'Akshay');

insert into customer values(5,'Ayush');

insert into customer values(6,'Chinmay');

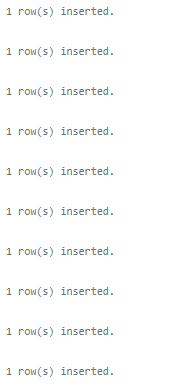
insert into customer values(7,'Harshit');

insert into customer values(8,'Sagar');

insert into customer values(9,'Aryan');

insert into customer values(10,'Yashasvi');

Select \* from customer;





Insert into item values(1001,’TV’,52000);

insert into item values(1002,'Fridge',20000);

insert into item values(1003,'Washing Machine',18000);

insert into item values(1004,'Toaster',3000);

insert into item values(1005,'Charger',100);

insert into item values(1006,'Mobile',15000);

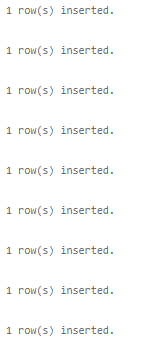
insert into item values(1007,'Oven',10000);

insert into item values(1008,'Cooler',12000);

insert into item values(1009,'Heater',800);

insert into item values(1010,'Geyser',12000);

Select \* from item;





insert into sale values(2001,'01-NOV-2020',9,1003,1);

insert into sale values(2002,'01-NOV-2020',7,1004,4);

insert into sale values(2003,'01-NOV-2020',6,1006,2);

insert into sale values(2004,'09-SEP-2020',2,1009,3);

insert into sale values(2005,'01-NOV-2020',8,1001,1);

insert into sale values(2006,'07-SEP-2020',8,1008,2);

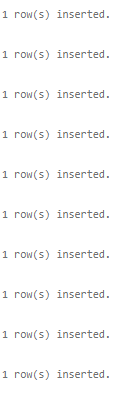
insert into sale values(2007,'01-NOV-2020',5,1001,3);

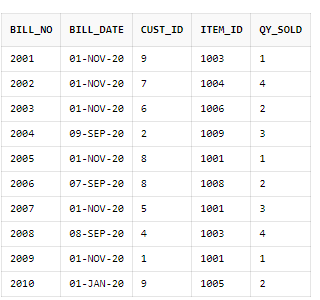
insert into sale values(2008,'08-SEP-2020',4,1003,4);

insert into sale values(2009,'01-NOV-2020',1,1001,1);

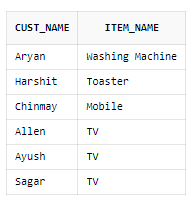
insert into sale values(2010,'01-JAN-2020',9,1005,2);

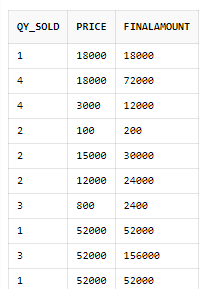
Select \* from sale;





c) Select cust\_name,item\_name from customer Natural JOIN sale Natural Join item where bill\_date='1-NOV-2020';

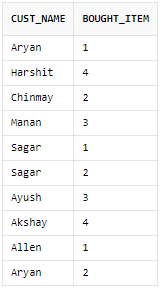


d) Select qy\_sold,price,(qy\_sold\*price)as finalamount from sale Natural Join item ;

e) Select cust\_id,cust\_name from customer Natural join sale natural join item where price>200;



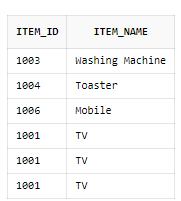
f) Select cust\_name,qy\_sold as bought\_item from customer Natural JOIN sale;



g) Select item\_name from item Natural JOIN sale where cust\_id=5 ;



h) Select item\_id,item\_name from item Natural JOIN sale where bill\_date='1-NOV-2020' ;



**4. Database Schema for a Student Library scenario**

**Student(Stud\_no : integer, Stud\_name: string)**

**Membership(Mem\_no: integer, Stud\_no: integer)**

**Book(book\_no: integer, book\_name:string, author: string)**

**Iss\_rec(iss\_no:integer, iss\_date: date, Mem\_no: integer, book\_no: integer)**

1. Create the tables with the appropriate integrity constraints

create table Student(Stud\_no int not null, Stud\_name varchar2(20),primary key(Stud\_no));

create table Membership(Mem\_no: int not null, Stud\_no int,primary key(Mem\_no));

create table Book(book\_no int not null, book\_name varchar2(20), author varchar(15),primary key(book\_no));

create table Iss\_rec(iss\_no int not null, iss\_date date, Mem\_no int, book\_no int,primary key(iss\_no));

2. Insert around 10 records in each of the tables

insert into student values(1,'Kartik');

insert into student values(2,'Vansh');

insert into student values(3,'Mehta');

insert into student values(4,'Tyagi');

insert into student values(5,'Aditya');

insert into student values(6,'Gautam');

insert into student values(7,'Siddharth');

insert into student values(8,'Megha');

insert into student values(9,'Nandini');

insert into student values(10,'Diamond');

insert into Membership values(01,1);

insert into Membership values(02,2);

insert into Membership values(03,3);

insert into Membership values(04,4);

insert into Membership values(05,5);

insert into Membership values(06,6);

insert into Membership values(07,7);

insert into Membership values(08,8);

insert into Membership values(09,9);

insert into Membership values(010,10);

insert into book values(001,'Harry Potter','JK Rowling');

insert into book values(002,'Goosebums','RL stine');

insert into book values(003,'The BFG','Roald Dahl');

insert into book values(004,'Death','Sadhguru');

insert into book values(005,'Its All In Your Head','Russ');

insert into book values(006,'Harry Potter','JK Rowling');

insert into book values(007,'Harry Potter','JK Rowling');

insert into book values(008,'Harry Potter','JK Rowling');

insert into book values(009,'Harry Potter','JK Rowling');

insert into book values(0010,'Harry Potter','JK Rowling');

insert into Iss\_rec values(1,'31-jul-2016',1,1);  
insert into Iss\_rec values(2,'31-aug-2018',2,2);  
insert into Iss\_rec values(3,'15-jan-2016',3,3);

insert into Iss\_rec values(4,'3-dec-2019',4,4);

insert into Iss\_rec values(5,'6-mar-2020',5,5);

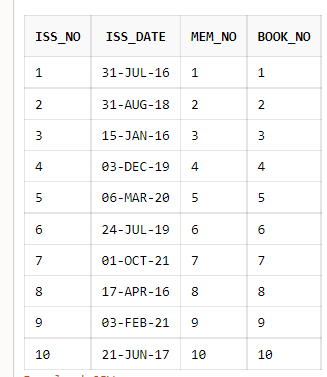
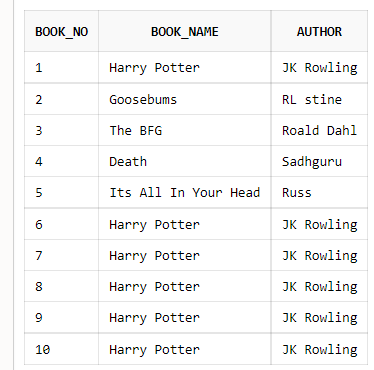
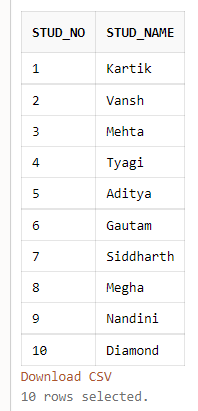
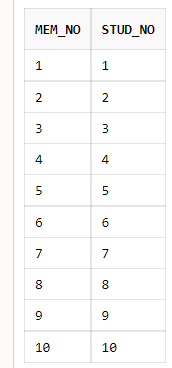
insert into Iss\_rec values(6,'24-jul-2019',6,6);

insert into Iss\_rec values(7,'1-oct-2021',7,7);

insert into Iss\_rec values(8,'17-apr-2016',8,8);

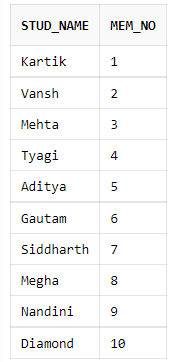
insert into Iss\_rec values(9,'3-feb-2021',9,9);

insert into Iss\_rec values(10,'21-jun-2017',10,10);



3. List all the student names with their membership numbers.

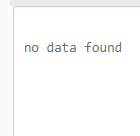
select stud\_name,mem\_no from student natural join membership;



4. List all the issues for the current date with student and Book names

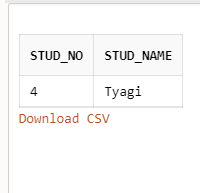
select stud\_name,book\_name from ((iss\_rec natural join book)natural join student)natural join membership

where iss\_date=to\_char(sysdate)



5. List the details of students who borrowed book whose author is ‘Sadhguru’.

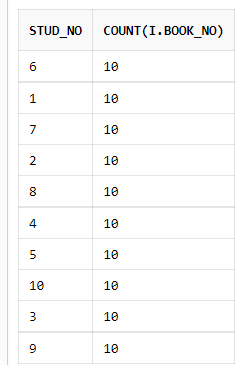
select \* from student where stud\_no in(select stud\_no from membership where mem\_no in (select mem\_no from iss\_rec where book\_no in(select book\_no from book where author='Sadhguru')));



6. Give a count of how many books have been bought by each student

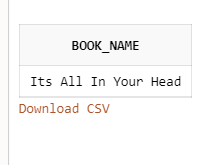
select s.stud\_no, count(i.book\_no) from student s,membership m, book b, iss\_rec i

where s.stud\_no=m.stud\_no and b.book\_no=i.book\_no group by s.stud\_no ;



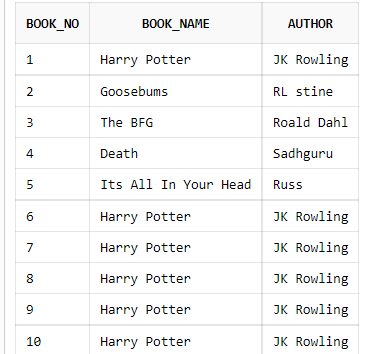
7. Give a list of books taken by student with stud\_no as 5-:

select book\_name from book where book\_no in (select book\_no from iss\_rec where mem\_no in(select mem\_no from membership where stud\_no in(select stud\_no from student where stud\_no=5)));



8. List the book details which are issued as of today.

delete from book where book\_no in (select book\_no from iss\_rec where iss\_date=to\_char(sysdate));



**5) Database Schema for a Employee-pay scenario**

**employee(emp\_id : integer, emp\_name: string)**

**department(dept\_id: integer, dept\_name:string)**

**paydetails(emp\_id : integer, dept\_id: integer, basic: integer, deductions: integer, additions: integer, DOJ: date)**

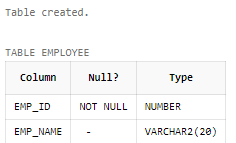
**payroll(emp\_id : integer, pay\_date: date)**

a) create table employee(

emp\_idint NOT NULL PRIMARY KEY,

emp\_namevarchar2(20));

desc employee;

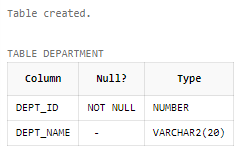


create table department(

dept\_idint NOT NULL PRIMARY KEY,

dept\_namevarchar2(20));

desc department;



create table paydetails(

emp\_idint,

dept\_idint,

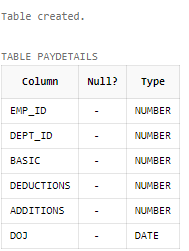
basicint,

deductionsint,

additionsint,

doj DATE);

descpaydetails;

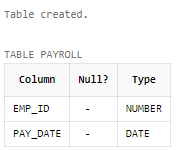


create table payroll(

emp\_idint,

pay\_date DATE);

desc payroll;



b) insert into employee values(1,'Abhishek');

insert into employee values(2,'Akshat');

insert into employee values(3,'Harshit');

insert into employee values(4,'Ayush');

insert into employee values(5,'Sagar');

insert into employee values(6,'Chinmay');

insert into employee values(7,'Manan');

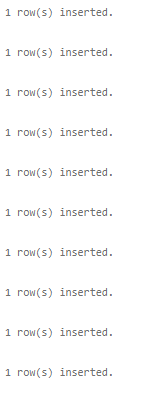
insert into employee values(8,'Aryan');

insert into employee values(9,'Yashasvi');

insert into employee values(10,'Aman');

Select \* from employee;

Graphical user interface

Description automatically generated with medium confidence

insert into department values(101,'Mechanical');

insert into department values(102,'Telecom');

insert into department values(103,'Civil');

insert into department values(104,'CS');

insert into department values(105,'Management');

insert into department values(106,'Helping');

insert into department values(107,'Electrical');

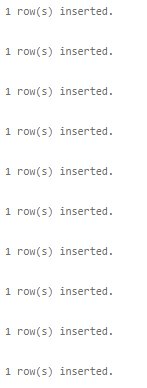
insert into department values(108,'Plumbing');

insert into department values(109,'Accounts');

insert into department values(110,'HR');

Select \* from department;

Table

Description automatically generated with medium confidence

insert into paydetails values(5,110,110000,10000,5000,'3-MAR-2000');

insert into paydetails values(6,109,76000,5000,500,'1-DEC-1999');

insert into paydetails values(1,110,110000,9000,1000,'31-MAR-1998');

insert into paydetails values(2,106,12000,1100,200,'13-OCT-2000');

insert into paydetails values(9,106,12000,5000,100,'11-MAR-1965');

insert into paydetails values(10,108,11000,999,900,'4-JAN-1970');

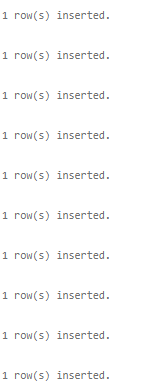
insert into paydetails values(3,109,76000,8000,1000,'9-JUN-1999');

insert into paydetails values(8,104,90000,7700,400,'18-NOV-2003');

insert into paydetails values(4,104,90000,1112,900,'1-FEB-2008');

insert into paydetails values(7,104,90000,9998,1000,'9-MAY-2005');

Select \* from paydetails;

Table

Description automatically generated

insert into payroll values(1,'1-JAN-2022');

insert into payroll values(2,'5-JAN-2022');

insert into payroll values(3,'9-JAN-2022');

insert into payroll values(4,'7-JAN-2022');

insert into payroll values(5,'1-JAN-2022');

insert into payroll values(6,'1-JAN-2022');

insert into payroll values(7,'1-JAN-2022');

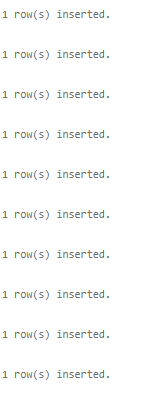
insert into payroll values(8,'31-DEC-2021');

insert into payroll values(9,'31-DEC-2021');

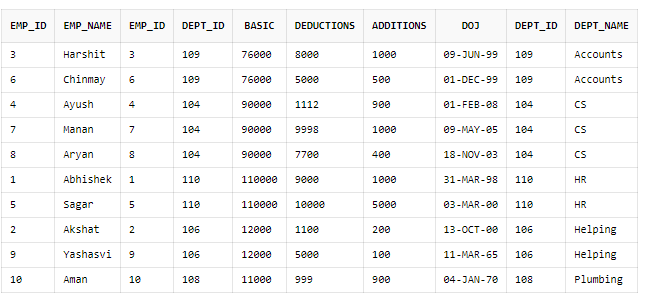
insert into payroll values(10,'1-JAN-2022');

Select \* from payroll;

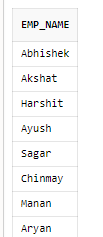
Table

Description automatically generated with medium confidence

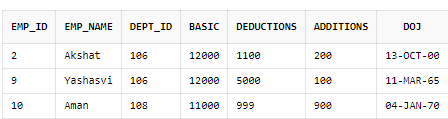
c) Select \* from employee, paydetails,department where employee.emp\_id=paydetails.emp\_id and department.dept\_id=paydetails.dept\_id order by dept\_name;



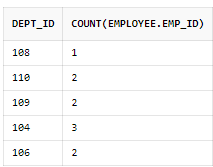
d) Select emp\_name from employee NAtural JOIN paydetails where doj>'1-JAN-1998' ;



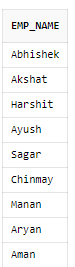
e) Select \* from employee Natural Join paydetails where basic>10000 and basic<20000;



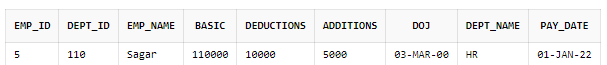
f) Select department.dept\_id,count(employee.emp\_id) from paydetails,employee,department where department.dept\_id=paydetails.dept\_id and employee.emp\_id=paydetails.emp\_id group by department.dept\_id;



g) Select emp\_name from employee Natural join paydetails where (basic-deductions+additions)>10000;



h) Select \* from employee Natural Join paydetails Natural JOIN department Natural JOIN payroll where emp\_id=5;



6.Database Schema for a Video Library scenario

**Customer(cust\_no: integer,cust\_name: string)**

**Membership(Mem\_no: integer, cust\_no: integer)**

**Cassette(cass\_no:integer, cass\_name:string, Language: String)**

**Iss\_rec(iss\_no: integer, iss\_date: date, mem\_no: integer, cass\_no: integer)**

1. Create the tables with the appropriate integrity constraints-

Customer(cust\_no: integer,cust\_name: string)

Membership(Mem\_no: integer, cust\_no: integer)

Cassette(cass\_no:integer, cass\_name:string, Language: String)

Iss\_rec(iss\_no: integer, iss\_date: date, mem\_no: integer, cass\_no: integer)

2. Insert around 10 records in each of the tables-:

insert into customer values(1,'Kartik');

insert into customer values(2,'Vansh');

insert into customer values(3,'Mehta');

insert into customer values(4,'Tyagi');

insert into customer values(5,'Aditya');

insert into customer values(6,'Gautam');

insert into customer values(7,'Siddharth');

insert into customer values(8,'Megha');

insert into customer values(9,'Nandini');

insert into customer values(10,'Diamond');

insert into Membership values(01,1);

insert into Membership values(02,2);

insert into Membership values(03,3);

insert into Membership values(04,4);

insert into Membership values(05,5);

insert into Membership values(06,6);

insert into Membership values(07,7);

insert into Membership values(08,8);

insert into Membership values(09,9);

insert into Membership values(010,10);

insert into cassette values(1,'dhoom','hindi');

insert into cassette values(2,'dhoom2','hindi');

insert into cassette values(3,'dhoom3','hindi');

insert into cassette values(4,'lata songs','english');

insert into cassette values(5,'lata songs','english');

insert into cassette values(6,'asha song','english');

insert into cassette values(7,'despacito','spanish');

insert into cassette values(8,'scars to be beautiful','english');

insert into cassette values(9,'gully boy','hindi');

insert into cassette values(10,'tulsi kumar songs','hindi');

insert into iss\_reco values(11,"1998-08-02",1,1);

insert into iss\_reco values(12,"1999-05-03",2,2);

insert into iss\_reco values(13,"2000-07-09",3,3);

insert into iss\_reco values(14,"2004-07-09",4,4);

insert into iss\_reco values(15,"2008-12-12",5,5);

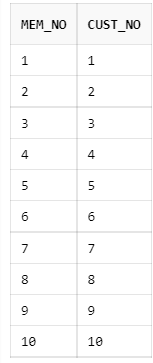
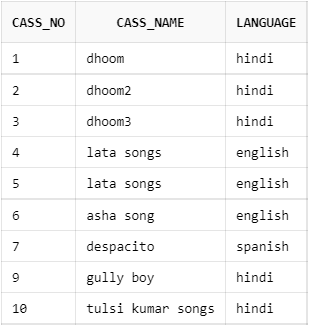
insert into iss\_reco values(16,"2007-06-06",6,6);

insert into iss\_reco values(17,"2012-02-06",7,7);

insert into iss\_reco values(18,"2016-02-06",8,8);

insert into iss\_reco values(19,"2016-08-12",9,9);

insert into iss\_reco values(20,"2019-07-30",10,10);

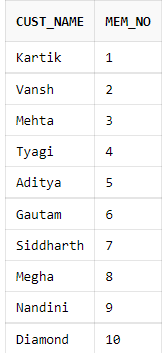


Calendar

Description automatically generated with low confidence

3.List all the customer names with their membership numbers-:

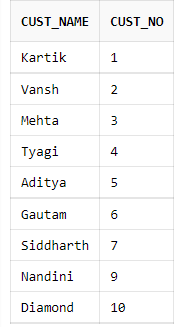
select cust\_name,mem\_no from customer natural join membership



4. List all the issues for the current date with the customer names and cassette names.

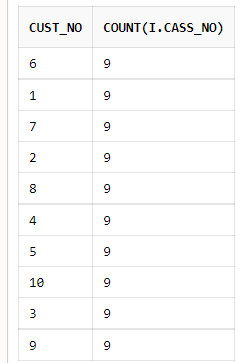
select i.iss\_no , c.cust\_name,cs.cass\_name from customer c, membership m , cassette cs , iss\_rec i where c.cust\_no=m.cust\_no and m.mem\_no=i.mem\_no and cs.cass\_no=i.cass\_no and i.iss\_date=to\_char(sysdate);



5. List the details of the customer who has borrowed the cassette whose title is “lata songs”.

6. Give a count of how many cassettes have been borrowed by each customer.

select s.cust\_no, count(i.cass\_no) from customer s,membership m, cassette b, iss\_reco i where s.cust\_no=m.cust\_no and b.cass\_no=i.cass\_no group by s.cust\_no ;



7. Give a list of book which has been taken by the student with mem\_no as 5;-

select cass\_name from cassette where cass\_no in (select cass\_no from iss\_reco where mem\_no in(select mem\_no from membership where cust\_no in(select cust\_no from customer where cust\_no=5)));



8. List the cassettes issues for today.

select cass\_name from cassette where cass\_no in (select cass\_no from iss\_reco

where iss\_date=to\_char(sysdate));

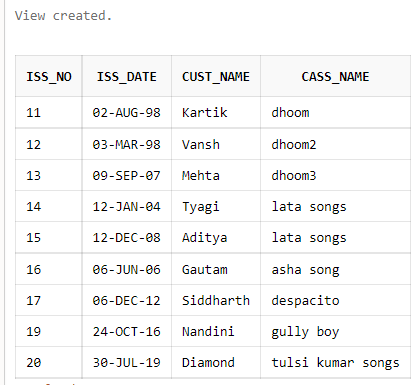
Graphical user interface, application

Description automatically generated

9. Create a view which lists outs the iss\_no, iss\_date, cust\_name, cass\_name.

create view cus\_view as select i.iss\_no,i.iss\_date,c.cust\_name,cass\_name from customer c,membership m,cassette c,iss\_reco i where c.cust\_no=m.cust\_no and c.cass\_no=i.cass\_no and m.mem\_no=i.mem\_no;

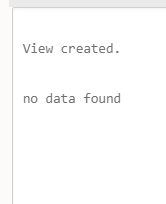
select \* from cus\_view;



10. Create a view which lists issues-date wise for the last one week.

create view cust\_view as select c.cust\_no,i.iss\_no,c.cust\_name,c.cass\_name from customer c,membership m,cassette c,iss\_reco i where c.cust\_no=m.cust\_no and c.cass\_no=i.cass\_no and m.mem\_no=i.mem\_no and iss\_date>=ADD\_MONTHS( SYSDATE, -7 );

select \* from cust\_view;



**7.) Database Schema for a student-Lab scenario**

**Student(stud\_no: integer, stud\_name: string, class: string)**

**Class(class: string, descrip: string)**

**Lab(mach\_no: integer, Lab\_no: integer, description: String)**

**Allotment(Stud\_no: Integer, mach\_no: integer, dayof week: string)**

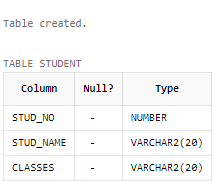
a)create table student(

stud\_noint ,

stud\_namevarchar2(20),

classes varchar2(20));

desc student;

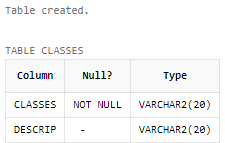


create table classes(

classes varchar2(20) NOT NULL PRIMARY KEY ,

descrip varchar2(20));

desc classes;



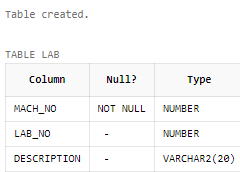
create table lab(

mach\_noint NOT NULL PRIMARY KEY ,

lab\_noint,

description varchar2(20));

desc lab;



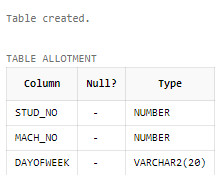
create table allotment(

stud\_noint,

mach\_noint,

dayofweek varchar2(20));

desc allotment;



b)insert into student values(1,'Abhijeet',8);

insert into student values(2,'Riyan',8);

insert into student values(3,'Ronak',12);

insert into student values(4,'Manshi',9);

insert into student values(5,'Manisha',7);

insert into student values(6,'Akansha',10);

insert into student values(7,'Aditya',11);

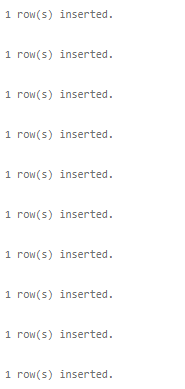
insert into student values(8,'Anshul',11);

insert into student values(9,'Shreya',10);

insert into student values(10,'Ravindra',10);

Select\* from student;

Calendar

Description automatically generated with medium confidence

insert into classes values(1,'first floor');

insert into classes values(2,'second floor');

insert into classes values(3,'third floor');

insert into classes values(4,'first floor');

insert into classes values(5,'second floor');

insert into classes values(6,'first floor');

insert into classes values(7,'third floor');

insert into classes values(8,'first floor');

insert into classes values(9,'second floor');

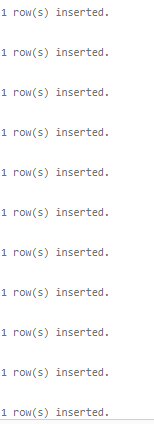
insert into classes values(10,'first floor');

insert into classes values(11,'first floor');

insert into classes values(12,'first floor');

Select \* from classes;

Table

Description automatically generated with low confidence

insert into lab values( 11,1,'basement');

insert into lab values( 12,2,'basement');

insert into lab values( 13,3,'basement');

insert into lab values( 14,4,'first floor');

insert into lab values( 15,5,'basement');

insert into lab values( 16,6,'third floor');

insert into lab values( 17,7,'basement');

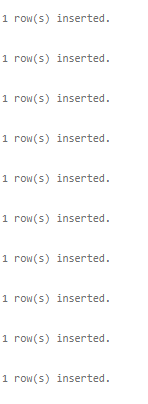
insert into lab values( 18,8,'basement');

insert into lab values( 19,9,'second floor');

insert into lab values( 20,10,'basement');

Select \* from lab;

Calendar

Description automatically generated with medium confidence

insert into allotment values(9,12,'monday');

insert into allotment values(7,18,'friday');

insert into allotment values(9,17,'tuesday');

insert into allotment values(8,11,'wednesday');

insert into allotment values(1,12,'monday');

insert into allotment values(5,17,'thursay');

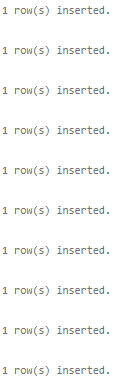
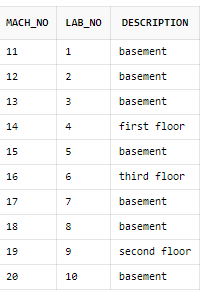
insert into allotment values(6,15,'monday');

insert into allotment values(10,12,'wednesday');

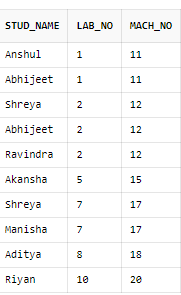
insert into allotment values(1,11,'tuesday');

insert into allotment values(2,20,'monday');

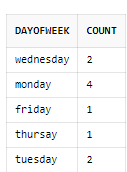
Select \* from lab;



c) Select stud\_name,lab\_no,mach\_no from student NAtural Join lab NAtural Join allotment;



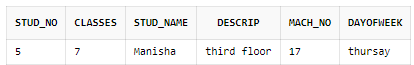
d) Select allotment.dayofweek,count(lab.lab\_no)as count from allotment,lab where lab.mach\_no=allotment.mach\_no group by allotment.dayofweek;



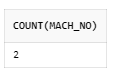
e)Select count(lab\_no)as count from student NAtural JOIN classes NAtural JOIN lab NAtural JOIN allotment where classes=10;



f) select \* from student Natural JOIN classes Natural join allotment where stud\_no=5;



g) Select count(mach\_no) from lab Natural join allotment where lab\_no=2 and dayofweek='monday';



h) Select student.classes,count(allotment.mach\_no)as count from allotment,lab,student where lab.mach\_no=allotment.mach\_no and student.stud\_no=allotment.stud\_no group by student.classes;

