CREATE TABLE employee

(

fname varchar(10) not null,

minit varchar(4),

lname varchar(10),

ssn integer(10) primary key,

bdate date,

address varchar(25),

sex char(1),

salary int(10),

superssn integer(10),

dno int(1), foreign key(superssn) references employee(ssn)

);

create table department

(dname varchar(20),

dnumber int(1) primary key,

mgrssn integer(10),

mgrstartdate date,

foreign key(mgrssn) references employee(ssn)

);

create table dept\_locations

(

dnumber int(1),

dlocation varchar(10),

foreign key(dnumber) references department(dnumber),

primary key(dnumber,dlocation)

);

create table project

(

pname varchar(20) not null,

pnumber int(4) primary key,

plocation varchar(10),

dnum int(1),

foreign key(dnum) references department(dnumber)

);

create table works\_on

(

essn int(10),

pno int(4),hours float,

foreign key(essn) references employee(ssn),

foreign key(pno) references project(pnumber),

primary key(essn,pno)

);

create table dependent

(essn integer (10),

dependent\_name varchar (15),

sex char (1), bdate date,

relationship varchar (20),

foreign key(essn) references employee(ssn),

primary key (essn, dependent\_name)

);

insert into employee(Fname, Minit, Lname, Ssn, Bdate, Address, sex, Salary) values

('John','B','Smith', 123456789,'1965-01-09','731-Fondren-Houston-TX','M', 30000),

('Franklin', 'T', 'Wong', 333445555, '1955-12-08', '638-Voss-Houston-TX','M', 40000),

('Alicia', 'J', 'Zelaya', 999887777, '1968-01-19', '3321-Castle-Spring-TX', 'F', 25000),

('Jennifer', 'S', 'Wallace', 987654321, '1941-06-20', '291-Berry-Bellaire-TX', 'F', 43000),

('Ramesh', 'K', 'Narayan', 666884444, '1962-09-15', '975-Fire-Oak-Humble-TX', 'M', 38000),

('Joyce','A', 'English', 453453453, '1972-07-31', '5631-Rice-Houston-TX','F', 25000),

('Ahmad','V', 'Jabbar', 987987987, '1969-03-29', '980-Dallas-Houston-TX', 'M', 25000),

('James', 'E', 'Borg', 888665555, '1937-11-10', '450-Stone-Houston-TX', 'M', 55000);

insert into department values

('Research', 5, 333445555, '1988-05-22'),

('Administration', 4, 987654321, '1995-01-01'),

('Headquarters', 1, 888665555, '1981-06-19');

insert into dept\_locations values

(1,'Houston'),

(4, 'Stafford'),

(5, 'Bellaire'),

(5, 'Sugarland'),

(5,'Houston');

insert into project values

('ProductX', 1, 'Bellaire', 5),

('ProductY', 2, 'Sugarland', 5),

('ProductZ', 3, 'Houston', 5),

('Computerization', 10, 'Stafford', 4),

('Reorganization', 20, 'Houston', 1),

('Newbenefits', 30, 'Stafford', 4);

insert into works\_on values

(123456789, 1, 32.5),

(123456789, 2, 7.5),

(666884444, 3, 40.0),

(453453453, 1, 20.0),

(453453453, 2, 20.0),

(333445555, 2, 10.0),

(333445555, 3, 10.0),

(333445555, 10, 10.0),

(333445555, 20, 10.0),

(999887777, 30, 30.0),

(999887777, 10, 10.0),

(987987987, 10, 35.0),

(987987987, 30, 5.0),

(987654321, 30, 20.0),

(987654321, 20, 15.0),

(888665555, 20, NULL);

insert into dependent values

(333445555,'Alice','F','1986-04-05','Daughter'),

(333445555,'Theodore','M','1983-10-25','Son'),

(333445555,'Joy','F','1958-05-03','Spouse'),

(987654321,'Abner','M','1942-02-28','Spouse'),

(123456789,'Michael','M','1988-01-04','Son'),

(123456789,'Alice','F','1988-12-30','Daughter'),

(123456789,'Elizabeth','F','1967-05-05','Spouse');

update employee set superssn=333445555,dno=5 where ssn=123456789;

update employee set superssn=888665555,dno=5 where ssn=333445555;

update employee set superssn=987654321,dno=4 where ssn=999887777;

update employee set superssn=888665555,dno=4 where ssn=987654321;

update employee set superssn=333445555,dno=5 where ssn=666884444;

update employee set superssn=333445555,dno=5 where ssn=453453453;

update employee set superssn=987654321,dno=4 where ssn=987987987;

update employee set dno=1 where ssn=888665555;

. Retrieve the name and address of all employees who work for the 'Research' Department

**2)**

**b)select fname,minit,lname from employee,department where dname="research" and**

**department.dnumber=employee.dno;**

**OUTPUT:**

**A black background with white text

AI-generated content may be incorrect.**

**c)For each employee, retrieve the employee's name, and the name of his or her immediate supervisor.**

**SYNTAX:**

**select f.fname as employee,e.fname as supervisor from employee e,employee f where e.ssn=f.superssn ;**

**OUTPUT:**

**A black background with white text

AI-generated content may be incorrect.**

**d)Retrieve the names of all employees who have two or more dependents.**

**SYNTAX:**

**select lname, fname from employee** **where (select count (\*) from dependent where**

**ssn=essn) >= 2;**

**OUTPUT:**

**A black screen with white text

AI-generated content may be incorrect.**

**3)**

a)For every project located in 'Stafford', list the project number, the controlling department number, and the department manager's last name, address, and birthdate

**SYNTAX**

**select pnumber, dnum, lname, address,bdate from project, department, employee where**

**plocation= “Stafford” and dnum=dnumber and mgrssn=ssn;**

**OUTPUT:**

**A screen shot of a computer program

AI-generated content may be incorrect.**

b)Retrieve all the attributes of an employee and attributes of DEPARTMENT he works in for every

employee of ‘Research’ department.

**SYNTAX:**

**select \* from employee, department where dname='research' and dno=dnumber;**

**OUTPUT:**

**A screenshot of a computer screen

AI-generated content may be incorrect.**

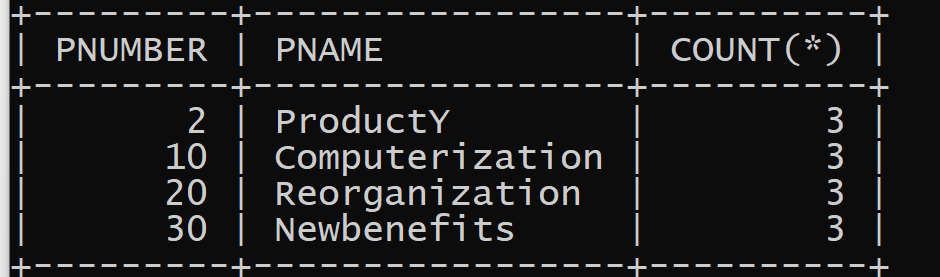
c) select count(salary), min(salary), max(salary), avg(salary), sum(salary) from employee;

4)

a) For each project on which more than two employees work, retrieve the project number, project name, and the number of employees who work on that project.

select pnumber, pname, count (\*) from project, works\_on where pnumber=pno group by pnumber, pname having count (\*) > 2;

**OUTPUT:**



b) For each department, retrieve the department number, the number of employees in the department, and their average salary.

**SYNTAX:**

select dno, count (\*), avg (salary) from employee group by dno;

**OUTPUT:**

A black background with white text

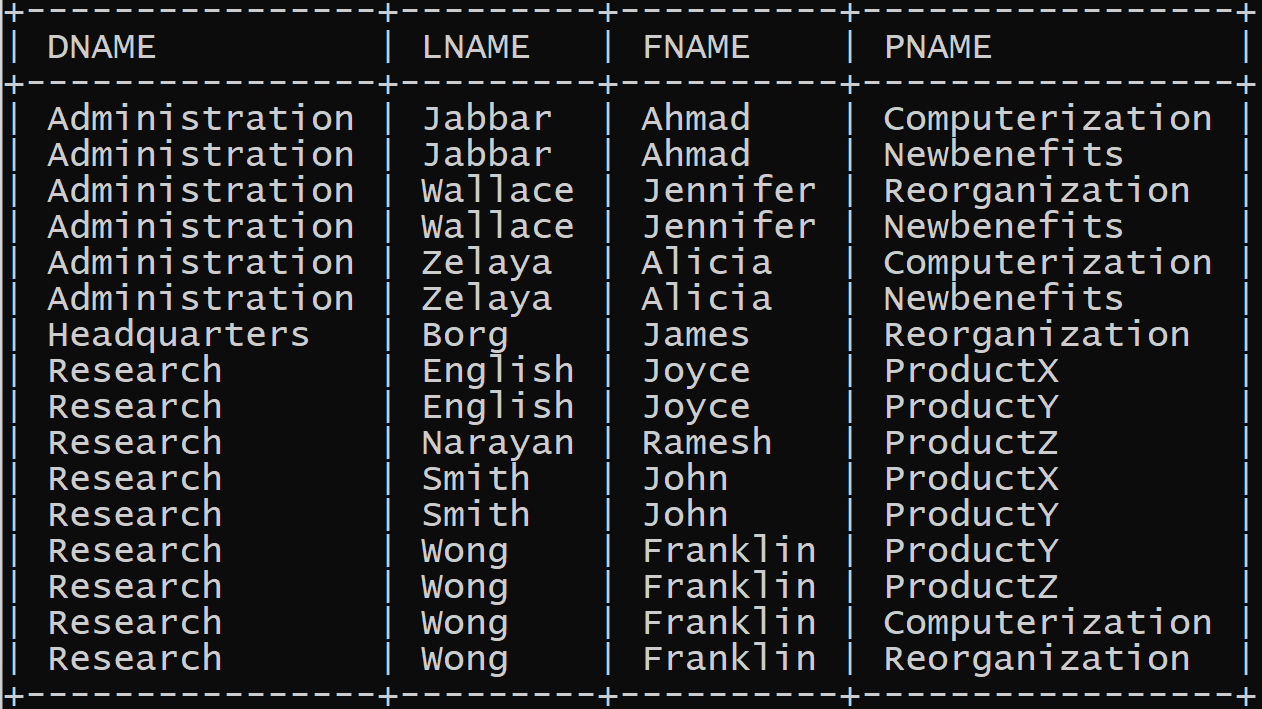
AI-generated content may be incorrect.

c) Retrieve a list of employees and the projects each works in, ordered by the employee's department, and within each department ordered alphabetically by employee last name.

**SYNTAX:**

select dname, lname, fname, pname from department, employee, works\_on, project where dnumber=dno and ssn=essn and pno=pnumber order by dname, lname;

**OUTPUT:**



5)

a) For each department, retrieve the department number, the number of employees in the department, and their average salary.

**SYNTAX:**

select dno, count (\*), avg (salary) from employee group by dno;

**OUTPUT:**

A black background with white text

AI-generated content may be incorrect.

b) For each project on which more than two employees work, retrieve the project number, project name, and the number of employees who work on that project.

select pnumber, pname, count (\*) from project, works\_on where pnumber=pno group by pnumber, pname having count (\*) > 2;

**OUTPUT:**

A black screen with white text

AI-generated content may be incorrect.

c) Retrieve all employees whose address is in Houston, Texas. Here, the value of the ADDRESS attribute must

contain the substring 'Houston,TX‘ in it.

**SYNTAX:**

**select fname, lname from employee where address like '%houston\_tx%';**

**OUTPUT:**

**A black screen with white text

AI-generated content may be incorrect.**

CREATE TABLE flight (

no INT (11),

frm VARCHAR(20),

too VARCHAR(20),

distance INT(11),

departs VARCHAR (20),

arrives VARCHAR(20),

price double,

PRIMARY KEY (no) );

CREATE TABLE aircraft(

aid INT,

aname VARCHAR(20),

cruisingrange INT,

PRIMARY KEY (aid) );

CREATE TABLE employees(

eid INT(20),

ename VARCHAR(20),

salary INT (20),

PRIMARY KEY (eid) );

CREATE TABLE certified(

eid INT(20),

aid INT(10),

PRIMARY KEY (eid,aid),

FOREIGN KEY (eid) REFERENCES employees(eid),

FOREIGN KEY (aid) REFERENCES aircraft (aid) );

INSERT INTO FLIGHT VALUES (1, “BANGALORE”, “MANGALOINSERT VALRE”,360, “10:45:00”, “12:00:00”,10000);

INSERT INTO FLIGHT VALUES (2, “BANGALORE”, “DELHI”,5000, “12:15:00”, “04:30:00”,25000);

INSERT INTO FLIGHT VALUES (3, “BANGALORE”, “MUMBAI”,3500, “02:15:00”, “05:25:00”,30000);

INSERT INTO FLIGHT VALUES (4, “DELHI”, “MUMBAI”,4500, “10:15:00”, “12:05:00”,35000);

INSERT INTO FLIGHT VALUES (5, “DELHI”, “FRANKFURT”,18000, “07:15:00”, “05:30:00”,90000);

INSERT INTO FLIGHT VALUES (6, “BANGALORE”, “FRANFURT”,19500, “10:00:00”, “07:45:00”,95000);

INSERT INTO FLIGHT VALUES (7, “BANGALORE”, “FRANKURT”,17000, “12:00:00”, “06:30:00”,99000);

INSERT INTO AIRCRAFT VALUES (123, “AIRBUS”,1000);

INSERT INTO AIRCRAFT VALUES (302, “BOEING”,5000);

INSERT INTO AIRCRAFT VALUES (306, “JET01”,5000);

INSERT INTO AIRCRAFT VALUES (378, “AIRBUS380”,8000);

INSERT INTO AIRCRAFT VALUES (456, “AIRCRAFT”,500);

INSERT INTO AIRCRAFT VALUES (789, “AIRCRAFT02”,800);

INSERT INTO AIRCRAFT VALUES (951,“AIRCRAFT03”,1000);

INSERT INTO employees VALUES (1, “Ajay”,30000);

INSERT INTO employees VALUES (2, “Ajith”,85000);

INSERT INTO employees VALUES (3, “Arnab”,50000);

INSERT INTO employees VALUES(4, “Harry”,45000);

INSERT INTO employees VALUES(5, “Ron”,90000);

INSERT INTO employees VALUES (6, “Josh”,75000);

INSERT INTO employees VALUES (7, “Ram”,100000);

INSERT INTO certified VALUES (1,123);

INSERT INTO certified VALUES (2,123);

INSERT INTO certified VALUES (1,302);

INSERT INTO certified VALUES (5,302);

INSERT INTO certified VALUES (7,302);

INSERT INTO certified VALUES (1,306);

INSERT INTO certified VALUES (2,306);

INSERT INTO certified VALUES (1,378);

INSERT INTO certified VALUES (2,378);

INSERT INTO certified VALUES (4,378);

INSERT INTO certified VALUES (6,456);

INSERT INTO certified VALUES (3,456);

INSERT INTO certified VALUES (5,789);

INSERT INTO certified VALUES (6,789);

INSERT INTO certified VALUES (3,951);

INSERT INTO certified VALUES (1,951);

INSERT INTO certified VALUES (1,789);

6)

1)Find the names of aircraft such that all pilots certified to operate them have salaries more than Rs 80,000.

SYNTAX:

SELECT DISTINCT a.aname FROM aircraft a,certified c,employees e WHERE a.aid=c.aid AND c.eid=e.eid AND EXISTS (SELECT \* FROM employees e1 WHERE e1.eid=e.eid AND e1.salary>80000);

A computer screen with white text

AI-generated content may be incorrect.

2) For each pilot who is certified for more than three aircrafts, find the eid and the maximum cruising range of the aircraft for which he/she is certified.

SYNTAX:

SELECT c.eid, MAX(cruisingrange) from certified c, aircraft a where c.aid = a.aid group by c.eid having count(\*)>3;

A computer screen with white text

AI-generated content may be incorrect.

3) Find the names of all pilots whose salary is less than the price of the cheapest route from Bangalore to Frankfurt.

SYNTAX:

select ename from employees where salary < ( select min(price) from flight where frm = “bangalore” and too =“franfurt” );

A computer screen with white text

AI-generated content may be incorrect.

4.For all aircrafts with cruisingrange over 1000 kms,find the name of the aircraft and the average salary of all pilots certified for this aircraft.

SYNTAX:

Select a.aid, a.aname, AVG(e.salary) from aircraft a, certified c, employees e where a.aid = c.aid AND c.eid =e.eid AND a.cruisingrange > 1000 GROUP BY a.aid, a.aname;

A computer screen with white text

AI-generated content may be incorrect.

5.Find the names of pilots certified for some Boeing aircraft.

SYNTAX:

SELECT distinct e. ename from employees e, aircraft a, certified c where e.eid = c.eid AND

c.aid = a.aid AND a. aname = “Boeing”;

A computer screen shot of a black screen

AI-generated content may be incorrect.

6.Find the aid”s of all aircraft whose crusing range is grater than the minimum distance from Bangalore to Delhi.

SYNTAX:

Select aid from aircraft where cruisingrange > (Select min(distance) from flight where frm= “Bangalore” and too= “Delhi”);

A computer screen with white text

AI-generated content may be incorrect.

**CREATION OF TABLES IN COLLEGE DATABASE:**

**CREATE TABLE FOR STUDENT:**

**SYNTAX:**

create table student(usn char(10),sname varchar(20),address varchar(25), phone int(10),gender char,constraint a primary key(usn));

**CREATE TABLE FOR SEMSEC:**

**SYNTAX:**

create table semsec(ssid char(2),sem int(1),sec char,constraint b primary key(ssid),constraint c check(sem between 1 and 8));

**CREATE TABLE FOR CLASS:**

**SYNTAX:**

create table class(usn char(10),ssid char(2),constraint d primary key(usn,ssid),constraint e foreign key(usn) references student(usn) on delete cascade,constraint f foreign key(ssid) references semsec(ssid) on delete cascade);

**CREATE TABLE FOR SUBJECT:**

**SYNTAX:**

create table subject(subcode varchar(7),title varchar(20),sem int(1),

credits int(1),constraint g primary key(subcode));

**CREATE TABLE FOR IAMARKS:**

**SYNTAX:**

create table iamarks(usn char(10),subcode varchar(7),ssid char(2), test1 int(2),test2 int(2),test3 int(2),finalia int(2), constraint h primary key(usn,subcode,ssid), constraint i foreign key(usn) references student(usn) on delete cascade,constraint j foreign key(ssid) references semsec(ssid) on delete cascade,constraint k foreign key(subcode) references subject(subcode) on delete cascade);

**INSERTION OF DATA FOR COLLEGE DATABASE:**

**INSERT VALUES INTO STUDENT TABLE:**

**SYNTAX:**

insert into student (usn,sname,address,phone,gender) values

(“1mv17cs001”, “aashish”, “bangalore”,1020304050, “m”),

(“1mv17cs060”, “naela”, “mysore”,1122334455, “f”),

(“1mv17cs130”, “milind”, “jammu”,506070809, “m”),

(“1mv16cs001”, “abhijith”, “pune”,998877655, “m”),

(“1mv16cs060”, “nikitha”, “hyderabad”,900706050, “f”),

(“1mv16cs130”, “sanjana”, “guwahatti”,124567890, “f”),

(“1mv15cs001”, “anshuman”, “panaji”,111223334, “m”),

(“1mv15cs060”, “amrutha”, “bangalore”,102003004, “f”),

(“1mv15cs130”, “bhuvanesh”, “jaipur”,900007006, “m”),

(“1mv14cs001”, “devayani”, “bangalore”,1020030, “f”),

(“1mv14cs060”, “david”, “kochi”,90080070, “m”),

(“1mv14cs130”, “aishwarya”, “mumbai”,100020000, “f”);

**INSERT VALUES INTO SEMSEC TABLE:**

**SYNTAX:**

insert into semsec (ssid,sem,sec) values

(“2a”,2, “a”),(“2b”,2, “b”),(“2c”,2, “c”),(“4a”,4, “a”),(“4b”,4, “b”),(“4c”,4, “c”),

(“6a”,6, “a”),(“6b”,6, “b”),(“6c”,6, “c”),(“8a”,8, “a”),(“8b”,8, “b”),(“8c”,8, “c”);

**INSERT VALUES INTO CLASS TABLE:**

insert into class(usn,ssid) values

(“1mv17cs001”, “2a”),

(“1mv17cs060”, “2b”),

(“1mv17cs130”, “2c”),

(“1mv16cs001”, “4a”),

(“1mv16cs060”, “4b”),

(“1mv16cs130”, “4c”),

(“1mv15cs001”, “6a”),

(“1mv15cs060”, “6b”),

(“1mv15cs130”, “6c”),

(“1mv14cs001”, “8a”),

(“1mv14cs060”, “8b”),

(“1mv14cs130”, “8c”);

**INSERT VALUES INTO SUBJECT TABLE:**

insert into subject(subcode,title,sem,credits) values

(“15cs21”, “m2”,2,4),

(“15pcd23”, “pcd”,2,4),

(“15cs42”, “se”,4,4),

(“15cs44”, “mp”,4,4),

(“15cs64”, “cg”,6,4),

(“15cs62”, “usp”,6,4),

(“10cs81”, “sa”,8,4),

(“10cs842”, “st”,8,3);

**INSERT VALUES INTO IAMARKS TABLE:**

insert into iamarks(usn,subcode,ssid,test1,test2,test3) values

(“1mv17cs001”, “15cs21”, “2a”,15,14,13),

(“1mv17cs060”, “15pcd23”, “2b”,15,15,14),

(“1mv17cs130”, “15cs21”, “2c”,11,12,13),

(“1mv16cs001”, “15cs42”, “4a”,19,19,18),

(“1mv16cs060”, “15cs44”, “4b”,5,8,5),

(“1mv16cs130”, “15cs42”, “4c”,20,20,20),

(“1mv15cs001”, “15cs64”, “6a”,12,12,12),

(“1mv15cs060”, “15cs62”, “6b”,18,19,20),

(“1mv15cs130”, “15cs64”, “6c”,8,12,11),

(“1mv14cs001”, “10cs81”, “8a”,3,11,12),

(“1mv14cs060”, “10cs842”, “8b”,0,0,7),

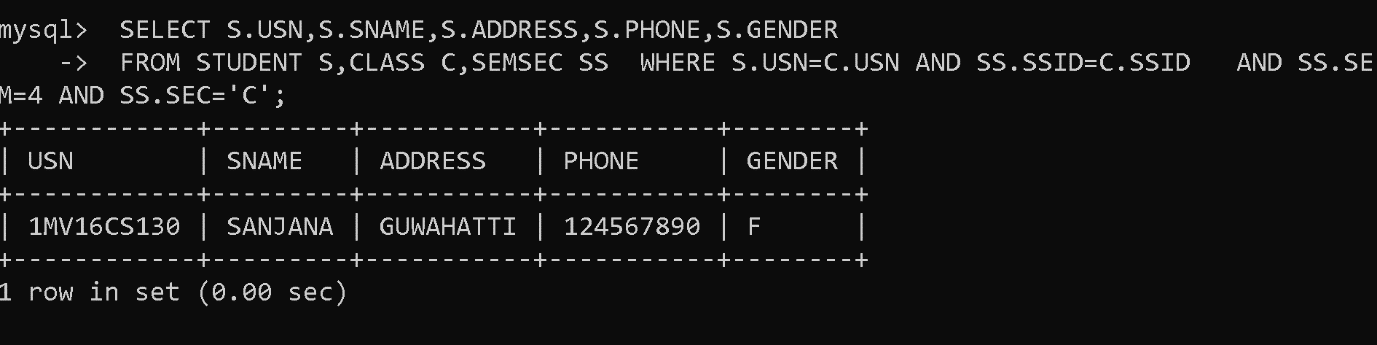
(“1mv14cs130”, “10cs81”, “8c”,0,0,20);

1. **MAKE A LIST OF ALL STUDENTS DETAILS STUDYING IN 4TH SEM C-SEC**

**SYNTAX:**

select s.usn,s.sname,s.address,s.phone,s.gender from student s,class c,semsec ss where s.usn=c.usn and ss.ssid=c.ssid and ss.sem=4 and ss.sec='c';

**OUTPUT:**



2. **COMPUTE THE TOTAL NO.OF MALE AND FEMALE STUDENTS IN EACH SEMESTER AND IN EACH SEC**

**SYNTAX:**

select ss.sem,ss.sec,s.gender,count(s.gender) from student s, semsec ss, class c where

s.usn=c.usn and ss.ssid=c.ssid group by ss.sem, ss.sec, s.gender;

**OUTPUT:**

A black screen with white text

AI-generated content may be incorrect.

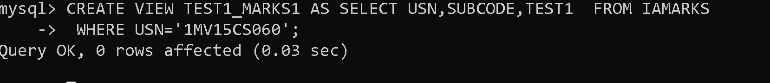
3. **CREATE VIEW OF TEST1 MARKS OF STUDENT 1MV15CS060 IN ALL SUBJECTS**

**SYNTAX:**

create view test1\_marks as select usn,subcode,test1 from iamarks where usn='1mv15cs060';

view created

**OUTPUT:**



**SYNTAX:**

select \* from test1\_marks;

**OUTPUT:**

A black screen with white text

AI-generated content may be incorrect.

**UPDATE VALUES IN IAMARKS TABLE:**

**SYNTAX:**

update iamarks set test1=19,test2=18,test3=17 where usn='1mv14cs001';

**SYNTAX:**

update iamarks set test1=11,test2=0,test3=14 where usn='1mv14cs060';

**SYNTAX:**

update iamarks set test1=10,test2=0,test3=7 where usn='1mv14cs130';

**DISPLAY THE IAMARKS TABLE:**

**SYNTAX:**

select \* from iamarks;

**OUTPUT:**

A screenshot of a computer screen

AI-generated content may be incorrect.

4. **CALCULATE THE FINAL IA MARKS AND UPDATE THE CORRESPONDING TABLE FOR ALL STUDENTS**

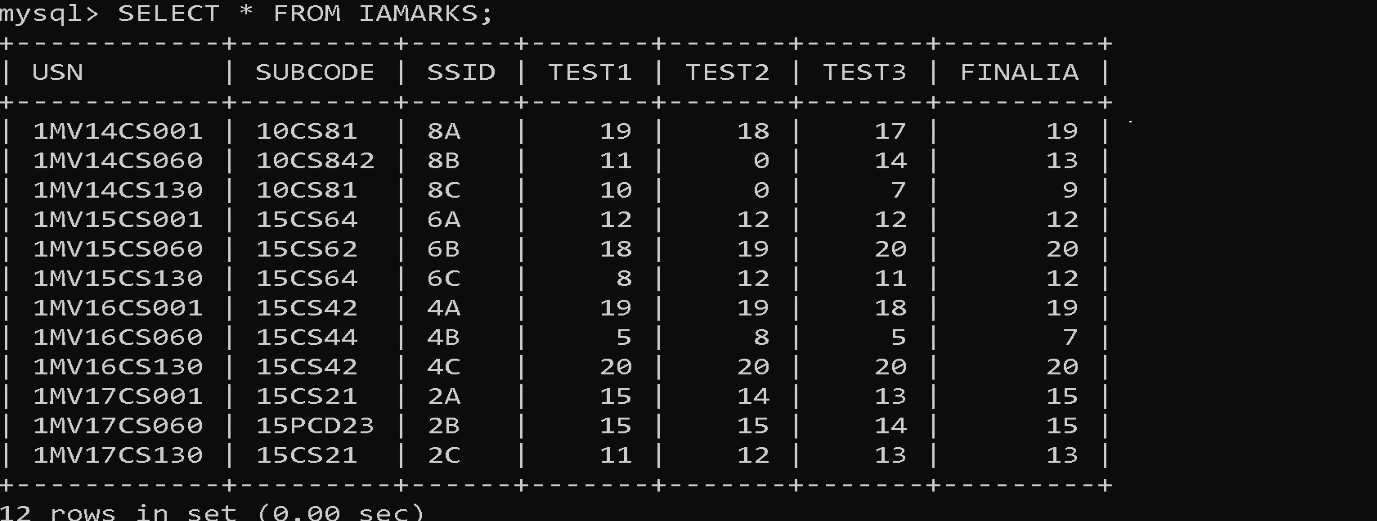
**SYNTAX:**

update iamarks set finalia=(greatest (test1+test2,test2+test3, test3+test1)/2);

**SYNTAX:**

Select \* from iamarks;

**OUTPUT:**



5. a)**CATEGORISE STUDENTS BASED ON THE FOLLOWING CRITERIA:**

if finalia=17 to 20 then cat=’outstanding’

if finalia=12 to 16 then cat=’average’

if finalia=00 to 11 then cat=’weak’

give these details only for 8th semester a, b and c section students.

**SYNTAX:**

select s.usn,s.sname,s.address,s.phone,s.gender,

(

case

when ia.finalia between 17 and 20

then 'outstanding'

when ia.finalia between 12 and 16

then 'average'

else 'weak'

end

)

as cat from student s,semsec ss,iamarks ia where s.usn=ia.usn and ss.ssid=ia.ssid and

ss.sem=8;

**OUTPUT:**



5. b)**UPDATE IAMARKS SET FINALIA=NULL;**

**SYNTAX:**

update iamarks set finalia=null;

**SYNTAX:**

select \* from iamarks;

**OUTPUT:**

