**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**QUES 1 to 4**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

create table StudentBasicInformation

(

StudentName varchar(20),

StudentSurName varchar(20),

StudentRollNo int primary key,

StudentDOB\_YYYY\_MM\_DD date,

StudentAddress varchar(70),

StudentGender char(1),

StudentPhoneNumber char(10)

);

insert into StudentBasicInformation values('Hemant', 'Saini', 1, '1998-01-10', 'Haryana', 'M', '8607080798');

insert into StudentBasicInformation values('Himani', 'Sharma', 2, '1996-04-11', 'Delhi', 'F', '8325848517');

insert into StudentBasicInformation values('Uttam', 'Saini', 3, '1998-12-21', 'Bihar', 'M', '9856741234');

insert into StudentBasicInformation values('Sudhir', 'Arora', 4, '1995-02-13', 'Utrakhand', 'M', '9345768129');

insert into StudentBasicInformation values('Nitu', 'Yadav', 5, '1994-07-20', 'Bihar', 'F', '8532976451');

insert into StudentBasicInformation values('Sourabh', 'Biswas', 6, '1999-02-15', 'Rajasthan', 'M', '8953941576');

insert into StudentBasicInformation values('Bhanu', 'Saini', 7, '2001-01-15', 'Haryana', 'F', '9658375641');

insert into StudentBasicInformation values('Anjali', 'Parashar', 8, '1998-04-30', 'Haryana', 'F', '9845789999');

insert into StudentBasicInformation values('Ridhi', 'Arora', 9, '2000-09-11', 'Jharkhand', 'F', '9955884155');

insert into StudentBasicInformation values('Tanisha', 'Meena', 10, '1998-02-24', 'Maharastra', 'F', '9999941785');

insert into StudentBasicInformation values('Tanu', 'Dutta', 11, '1997-10-23', 'Haryana', 'F', '9842657135');

insert into StudentBasicInformation values('Tanmay', 'Shrivastav', 12, '1997-02-12', 'Maharastra', 'M', '7348596123');

select \* from StudentBasicInformation;



create table StudentAdmissionPaymentDetails

(

StudentRollNo int primary key,

AmountPaid decimal(10,2),

AmountBalance decimal(10,2),

StudentDepartment varchar(50),

PaymentMode varchar(10),

StudentAdmissionDate date,

StudentCategory varchar(10),

foreign key(StudentRollNo) references StudentBasicInformation(StudentRollNo)

);

insert into StudentAdmissionPaymentDetails values(1, 20000.00, 5000.00, 'DUCS', 'Online', '2020-12-13', 'General');

insert into StudentAdmissionPaymentDetails values(2, 22000.00, 0.00, 'OR', 'Offine', '2020-11-28', 'OBC');

insert into StudentAdmissionPaymentDetails values(3, 20000.00, 2000.00, 'DUCS', 'Online', '2020-12-10', 'OBC');

insert into StudentAdmissionPaymentDetails values(4, 25000.00, 0.00, 'MATHEMATICS', 'Offine', '2020-12-13', 'General');

insert into StudentAdmissionPaymentDetails values(5, 20000.00, 0.00, 'POLITICAL SCIENCE', 'Online', '2020-12-13', 'SC');

insert into StudentAdmissionPaymentDetails values(6, 17000.00, 3000.00, 'OR', 'Offine', '2020-12-25', 'ST');

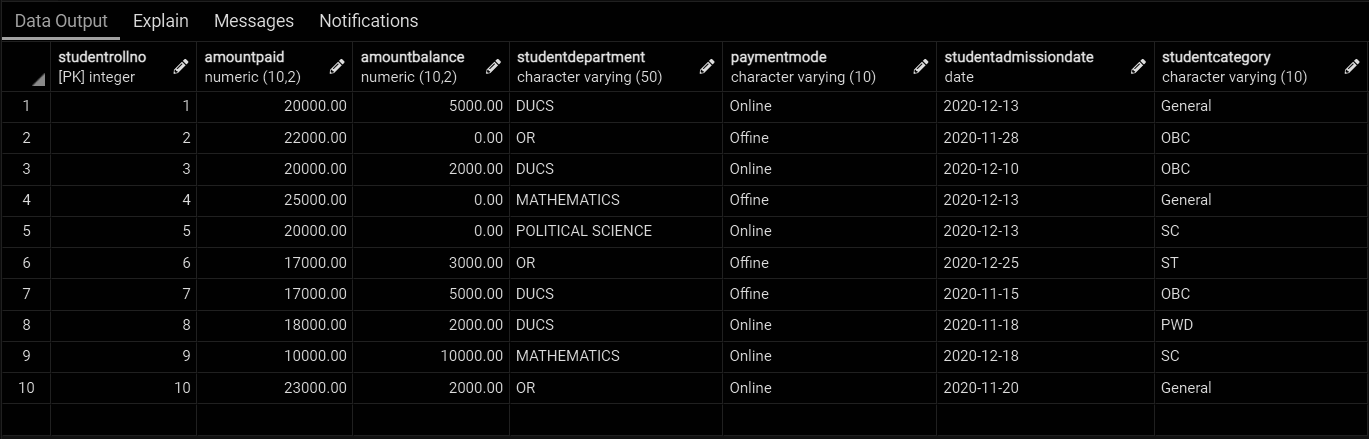
insert into StudentAdmissionPaymentDetails values(7, 17000.00, 5000.00, 'DUCS', 'Offine', '2020-11-15', 'OBC');

insert into StudentAdmissionPaymentDetails values(8, 18000.00, 2000.00, 'DUCS', 'Online', '2020-11-18', 'PWD');

insert into StudentAdmissionPaymentDetails values(9, 10000.00, 10000.00, 'MATHEMATICS', 'Online', '2020-12-18', 'SC');

insert into StudentAdmissionPaymentDetails values(10, 23000.00, 2000.00, 'OR', 'Online', '2020-11-20', 'General');

select \* from StudentAdmissionPaymentDetails;



create table StudentSubjectInformation

(

StudentRollNo int,

SubjectOpted varchar(30),

SubjectTotalMarks int,

SubjectObtainedMarks int,

StudentMarksPercentage float,

SubjectTeacherName varchar(30),

primary key(StudentRollNo, SubjectOpted),

foreign key(StudentRollNo) references StudentBasicInformation(StudentRollNo)

);

insert into StudentSubjectInformation values(1, 'English', 100, 83, 83.0, 'Miss Nisha');

insert into StudentSubjectInformation values(1, 'Computer Science', 50, 45, 90.0, 'Dr. Sapna');

insert into StudentSubjectInformation values(1, 'OR', 50, 43, 86.0, 'Dr. Rochna');

insert into StudentSubjectInformation values(2, 'English', 100, 86, 86.0, 'Miss Nisha');

insert into StudentSubjectInformation values(2, 'Computer Science', 50, 45, 90.0, 'Dr. Sapna');

insert into StudentSubjectInformation values(3, 'English', 100, 90, 90.0, 'Miss Nisha');

insert into StudentSubjectInformation values(3, 'Computer Science', 50, 47, 94.0, 'Dr. Sapna');

insert into StudentSubjectInformation values(4, 'English', 100, 92, 92.0, 'Miss Nisha');

insert into StudentSubjectInformation values(4, 'Computer Science', 50, 45, 90.0, 'Dr. Sapna');

insert into StudentSubjectInformation values(5, 'English', 100, 96, 96.0, 'Miss Nisha');

insert into StudentSubjectInformation values(5, 'Computer Science', 50, 23, 46.0, 'Dr. Sapna');

insert into StudentSubjectInformation values(6, 'English', 100, 99, 99.0, 'Miss Nisha');

insert into StudentSubjectInformation values(6, 'Computer Science', 50, 30, 60.0, 'Dr. Sapna');

insert into StudentSubjectInformation values(7, 'English', 100, 88, 88.0, 'Miss Nisha');

insert into StudentSubjectInformation values(7, 'Computer Science', 50, 7, 14.0, 'Dr. Sapna');

insert into StudentSubjectInformation values(8, 'English', 100, 75, 75.0, 'Miss Nisha');

insert into StudentSubjectInformation values(8, 'Computer Science', 50, 7, 14.0, 'Dr. Sapna');

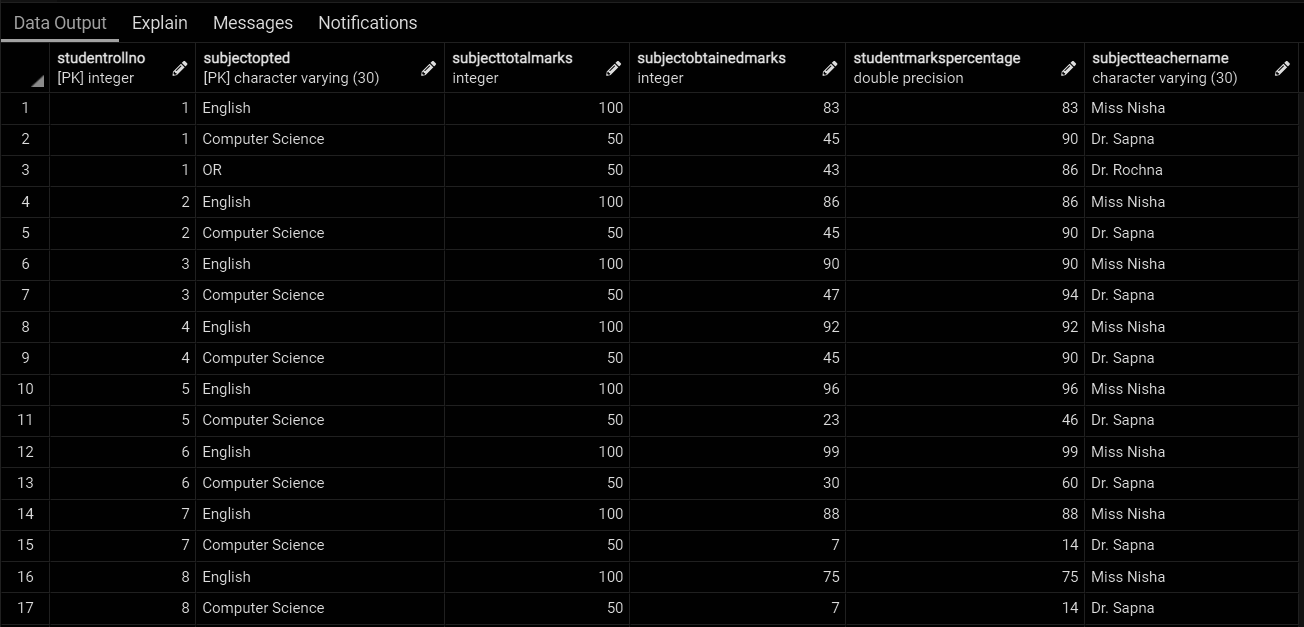
insert into StudentSubjectInformation values(9, 'English', 100, 32, 32.0, 'Miss Nisha');

insert into StudentSubjectInformation values(9, 'Computer Science', 50, 39, 78.0, 'Dr. Sapna');

insert into StudentSubjectInformation values(10, 'English', 100, 18, 18.0, 'Miss Nisha');

insert into StudentSubjectInformation values(10, 'Computer Science', 50, 17, 34.0, 'Dr. Sapna');

select \* from StudentSubjectInformation;



create table SubjectScholarshipInformation

(

StudentRollNo int primary key,

ScholarshipName varchar(20),

ScholarshipDescription varchar(100),

ScholarshipAmount decimal(8,2),

ScholarshipCategory varchar(30),

ScholarshipBeginDate date,

ScholarshipEndDate date,

ScholarshipStatus varchar(10),

foreign key(StudentRollNo) references StudentBasicInformation(StudentRollNo)

);

insert into SubjectScholarshipInformation values(1, 'Govt. Scholarship', 'Provided by Haryana Govt.', 10000.00, 'Medium', '2020-12-13', '2023-12-13', 'pending');

insert into SubjectScholarshipInformation values(2, 'Private Scholarship', 'Provided by Accolite', 10000.00, 'Medium', '2020-12-13', '2023-12-13', 'approved');

insert into SubjectScholarshipInformation values(3, 'Comunity Scholarship', 'Provided by Saini Community', 12000.00, 'High', '2020-12-13', '2023-12-13', 'approved');

insert into SubjectScholarshipInformation values(4, 'Govt. Scholarship', 'Provided by Haryana Govt.', 10000.00, 'Medium', '2020-12-13', '2023-12-13', 'approved');

insert into SubjectScholarshipInformation values(5, 'Comunity Scholarship', 'Provided by Saini Community', 12000.00, 'High', '2020-12-13', '2023-12-13', 'pending');

insert into SubjectScholarshipInformation values(6, 'Govt. Scholarship', 'Provided by Haryana Govt.', 5000.00, 'low', '2020-12-13', '2023-12-13', 'approved');

insert into SubjectScholarshipInformation values(7, 'Comunity Scholarship', 'Provided by Jaat Community', 5000.00, 'low', '2020-12-13', '2023-12-13', 'approved');

insert into SubjectScholarshipInformation values(8, 'Govt. Scholarship', 'Provided by Haryana Govt.', 5000.00, 'low', '2020-12-13', '2023-12-13', 'pending');

insert into SubjectScholarshipInformation values(9, 'Private Scholarship', 'Provided by Accolite', 5000.00, 'low', '2020-12-13', '2023-12-13', 'approved');

insert into SubjectScholarshipInformation values(10, 'Private Scholarship', 'Provided by Paytm', 12000.00, 'High', '2020-12-13', '2023-12-13', 'pending');

select \* from SubjectScholarshipInformation;



**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**QUES5**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

update SubjectScholarshipInformation

set ScholarshipStatus = 'approved'

where StudentRollNo = 1;

update SubjectScholarshipInformation

set ScholarshipEndDate = '2022-11-30'

where StudentRollNo = 3;

update SubjectScholarshipInformation

set ScholarshipDescription = 'Provided by Accolite Digital Pvt. Ltd.'

where ScholarshipDescription = 'Provided by Accolite';

update SubjectScholarshipInformation

set ScholarshipEndDate = '2024-05-30'

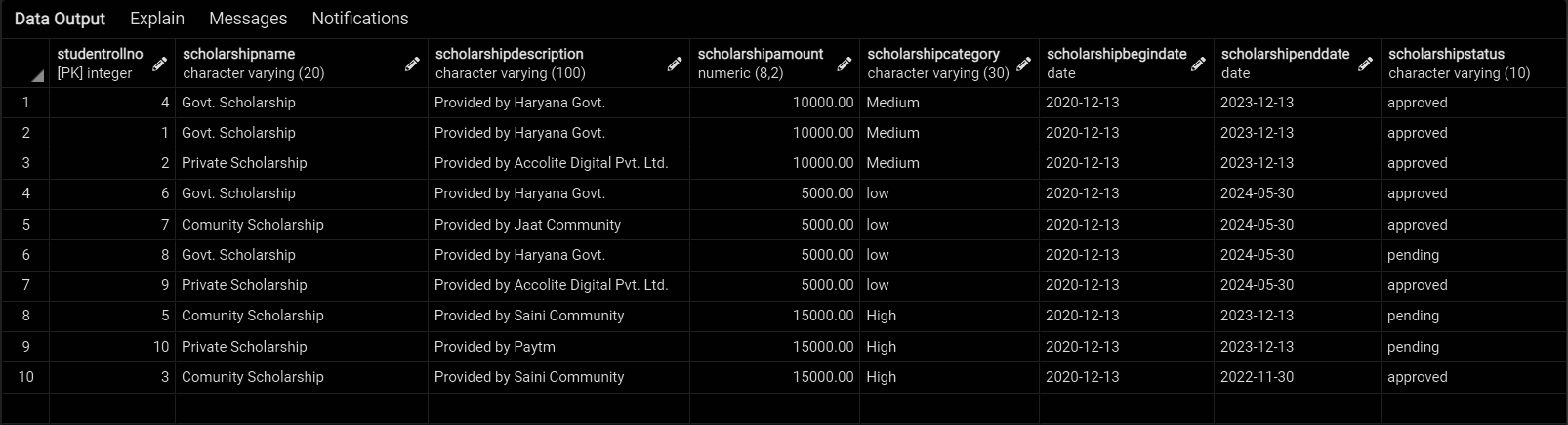
where ScholarshipAmount <= 9000;

update SubjectScholarshipInformation

set ScholarshipAmount = 15000

where ScholarshipCategory = 'High';

select \* from SubjectScholarshipInformation;



**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

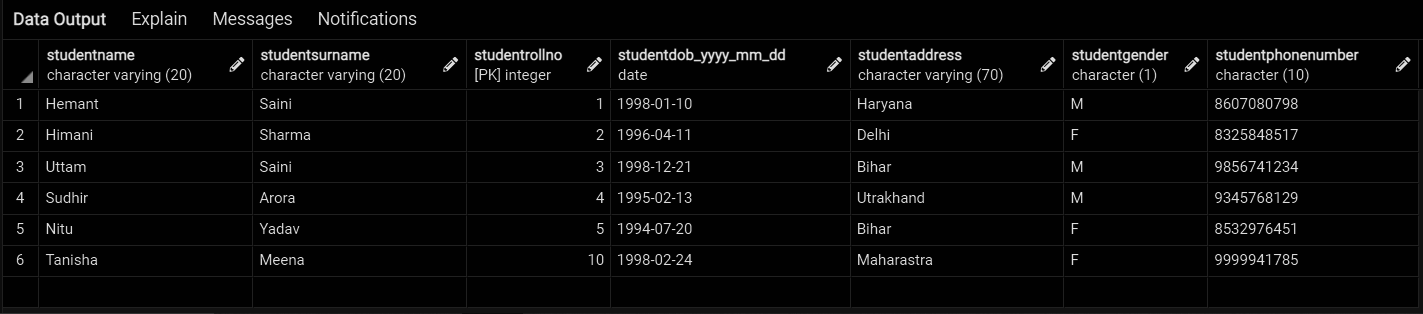
**QUES7**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

select \* from StudentBasicInformation

where StudentRollNo in

(select StudentRollNo from SubjectScholarshipInformation where ScholarshipAmount > 5000);



**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**QUES8**

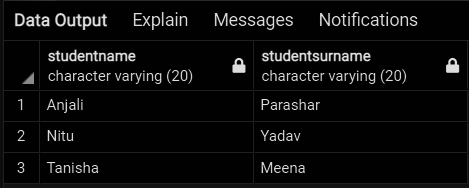
**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

select StudentName,StudentSurname

from StudentBasicInformation A join SubjectScholarshipInformation B

on A.StudentRollNo = B.StudentRollNo

where ScholarshipStatus = 'pending';



**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**QUES9**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

create or replace procedure fillPercentage()

language plpgsql

as $$

begin

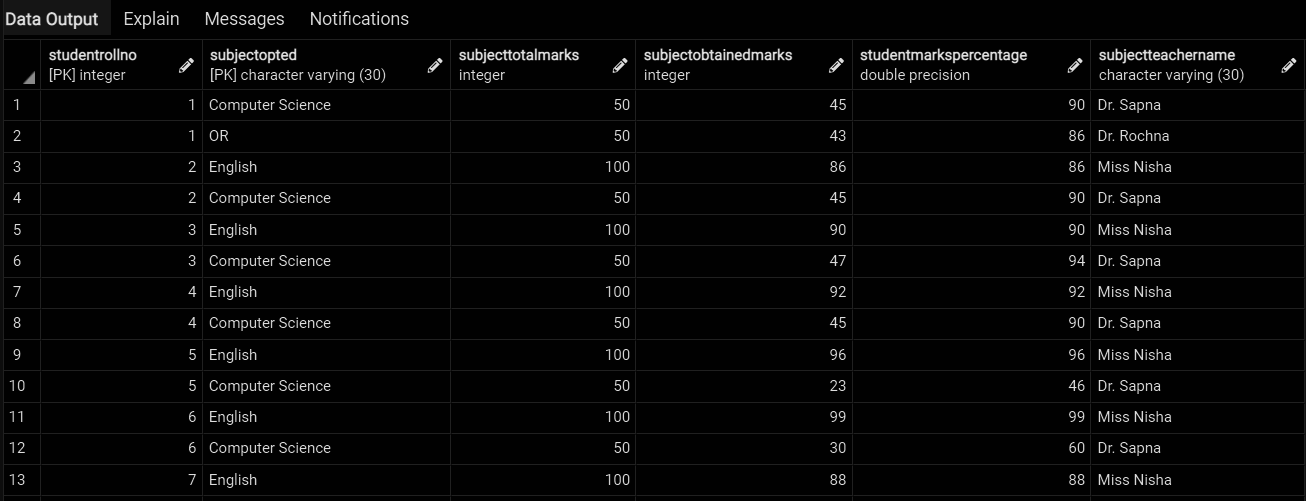
update StudentSubjectInformation

set StudentMarksPercentage= cast(SubjectObtainedMarks as float) /cast(SubjectTotalMarks as float) \* 100.0;

end;$$

call fillPercentage();

select \* from StudentSubjectInformation;



**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**QUES10**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

create or replace procedure setScholarshipCategory(

rollno int

)

language plpgsql

as $$

declare

percentage float;

begin

select StudentMarksPercentage into percentage

from StudentSubjectInformation where StudentRollNo = rollno;

if(percentage > 80 and percentage <= 90) then

update SubjectScholarshipInformation

set ScholarshipCategory = 'Low', ScholarshipAmount = 5000

where StudentRollNo = rollno;

end if;

if (percentage > 90 and percentage <= 95) then

update SubjectScholarshipInformation

set ScholarshipCategory = 'Medium', ScholarshipAmount = 10000

where StudentRollNo = rollno;

end if;

if (percentage > 95) then

update SubjectScholarshipInformation

set ScholarshipCategory = 'High', ScholarshipAmount = 15000

where StudentRollNo = rollno;

end if;

commit;

end;$$

call setScholarshipCategory(6);

select \* from SubjectScholarshipInformation;

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**QUES11**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

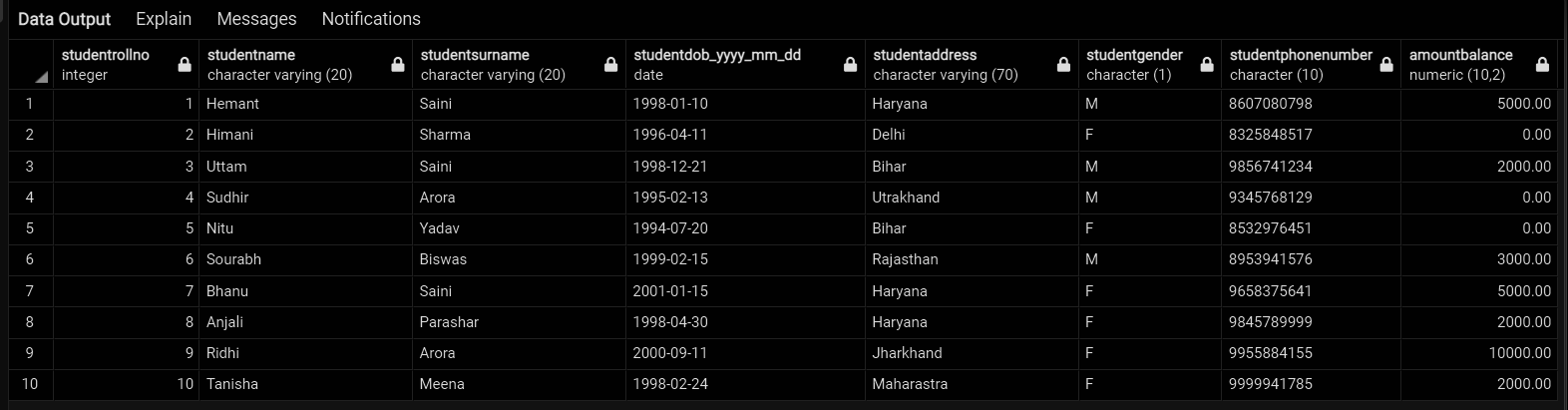
create or replace view BalanceView as

select A.StudentRollNo, StudentName, StudentSurname, StudentDOB\_YYYY\_MM\_DD, StudentAddress , StudentGender, StudentPhoneNumber, AmountBalance

from StudentAdmissionPaymentDetails A join StudentBasicInformation B

on A.StudentRollNo = B.StudentRollNo;

select \* from BalanceView;



**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

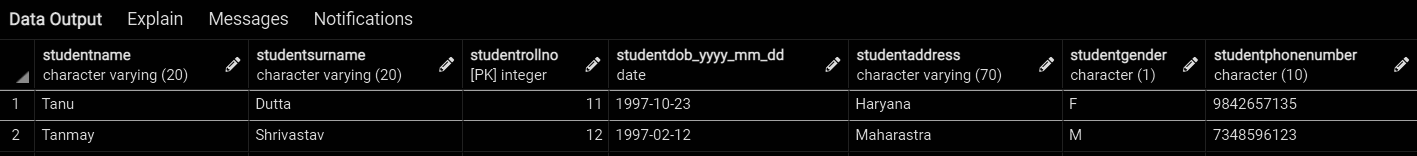
**QUES12**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

select \* from StudentBasicInformation

where StudentRollNo not in

(select StudentRollNo from SubjectScholarshipInformation);



**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**QUES13**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

create or replace function getAmountBalance(rollNo int)

returns float

language plpgsql

as $$

declare

balance integer;

begin

select AmountBalance into balance from StudentAdmissionPaymentDetails

where StudentRollNo = rollNo;

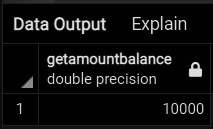
return balance;

end;$$

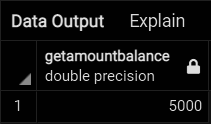
select getAmountBalance(7);

select getAmountBalance(9);

For 9:



For 7 :

****

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**QUES14**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

select \* from StudentBasicInformation

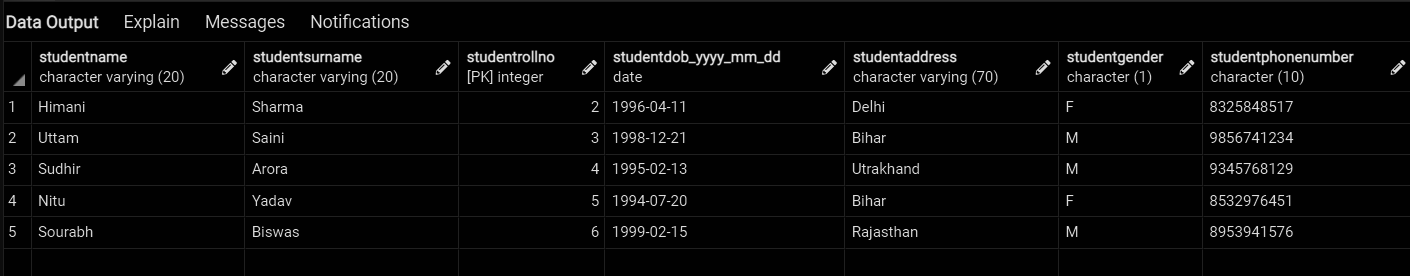
where StudentRollNo in (

select StudentRollNo from StudentSubjectInformation

order by StudentMarksPercentage desc

limit 5

);

****

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**QUES15**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

Left Join : Selecting name of students who did not get any schlorship

select StudentName, StudentSurName from

StudentBasicInformation A left join SubjectScholarshipInformation B

on A.StudentRollNo = B.StudentRollNo

where ScholarshipAmount is null;

inner Join : Same as ques11

create or replace view BalanceView as

select A.StudentRollNo, StudentName, StudentSurname, StudentDOB\_YYYY\_MM\_DD, StudentAddress , StudentGender, StudentPhoneNumber, AmountBalance

from StudentAdmissionPaymentDetails A inner join StudentBasicInformation B

on A.StudentRollNo = B.StudentRollNo;

select \* from BalanceView;

Right Join : Selecting students who have got schlorship

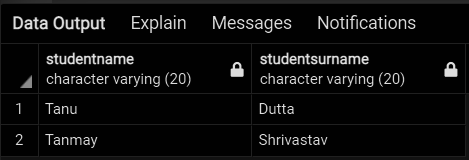
select A.StudentRollNo, StudentSurName from SubjectScholarshipInformation A

right join StudentBasicInformation B

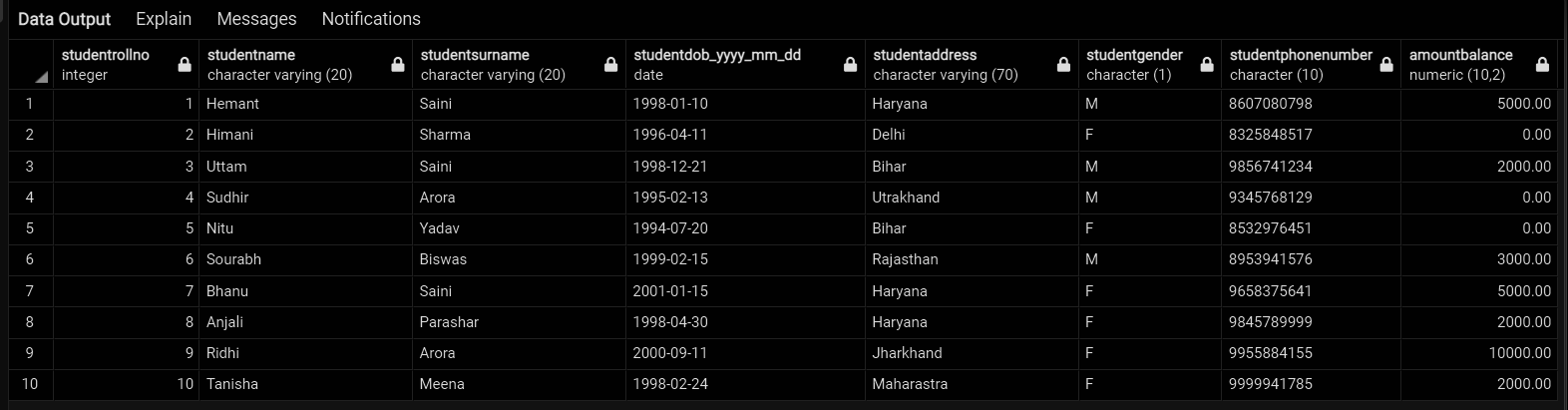
on A.StudentRollNo = B.StudentRollNo

where ScholarshipAmount is not null;

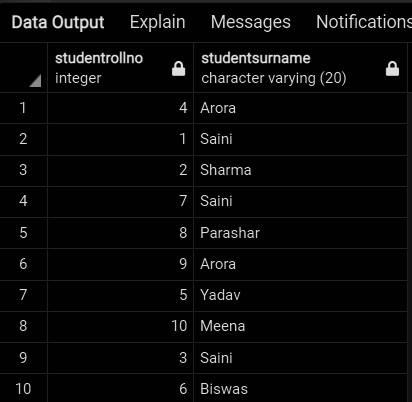
**Left Join :**

****

**Inner Join :**

****

**Right Join :**

****

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**QUES16**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

Delete :

1. The DELETE statement in SQL is a Data Manipulation Language (DML) Command.

2. It is used to delete tuples from a table.

3. The space for the tuples remains after deleting.

4. The structure or schema of the table is preserved.

Drop :

1. DROP statement is a Data Definition Language (DDL) Command.

2. It is used to drop the whole table.

3. The space for the tuples does not remain intact.

4. The structure or schema of the table is not preserved.

Truncate :

1. TRUNCATE command is a Data Definition Language (DDL)operation.

2. It is used to delete all the rows of a relation (table) in one go but not the table itself.

3. All spaces allocated for the tuples are removed.

4. The structure or schema of the table is preserved.

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**QUES17**

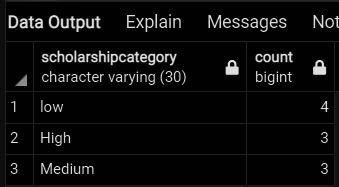
**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

select ScholarshipCategory, count(StudentRollNo)

from SubjectScholarshipInformation

group by ScholarshipCategory

order by count(StudentRollNo) desc;



**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**QUES18**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

create or replace view getScholarshipCategoryCount as

select ScholarshipCategory, count(StudentRollNo) as countOfCategory from SubjectScholarshipInformation

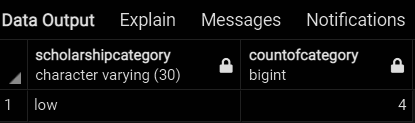
group by ScholarshipCategory

order by count(StudentRollNo) desc;

select ScholarshipCategory, countOfCategory from getScholarshipCategoryCount

where countOfCategory in

(select max(countOfCategory) from getScholarshipCategoryCount);



**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**QUES19**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

select StudentName, StudentSurname, A.StudentRollNo, StudentMarksPercentage, ScholarshipAmount

from StudentBasicInformation A join StudentSubjectInformation B

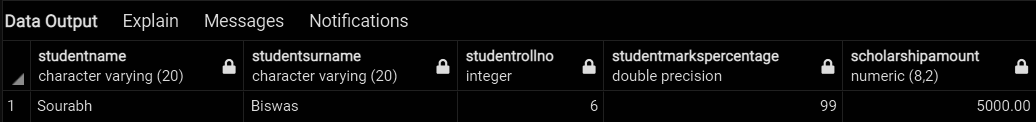
on A.StudentRollNo = B.StudentRollNo

left join SubjectScholarshipInformation C

on A.StudentRollNo = C.StudentRollNo

order by StudentMarksPercentage desc,ScholarshipAmount desc

limit 1;



**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**QUES20**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

Triggers :

=> A trigger is a special type of stored procedure that automatically runs when an event occurs in the database server.

=> DML triggers run when a user tries to modify data through a data manipulation language (DML) event. DML events are INSERT, UPDATE, or DELETE statements on a table or view.

Stored Procedure :

=> Stored procedure is a batch of statements grouped as a logical unit and stored in the database.

=> The stored procedure accepts the parameters and executes the T-SQL statements in the procedure, returns the result set if any.

=> It is a prepared SQL code that you can save, so the code can be reused over and over again.

View :

=> In SQL, a view is a virtual table based on the result-set of an SQL statement.

=> A view contains rows and columns, just like a real table. The fields in a view are fields from one or more real tables in the database.

=> You can add SQL functions, WHERE, and JOIN statements to a view and present the data as if the data were coming from one single table.

Functions :

=> A function is a routine that accepts parameters, performs an action, such as a complex calculation, and returns the result of that action as a value.

=> The return value can either be a scalar (single) value or a table.