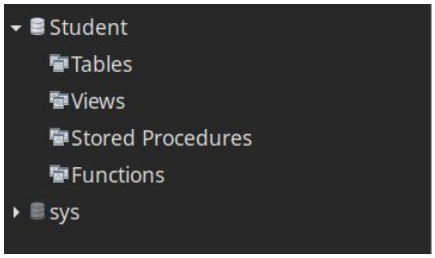
RDBMS Assignment

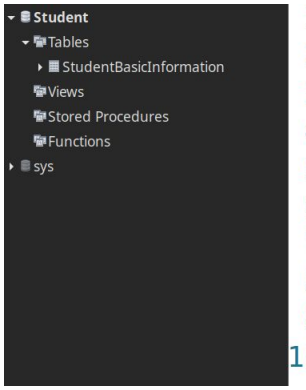
1 - Creating Student Database




```
create database Student;
```

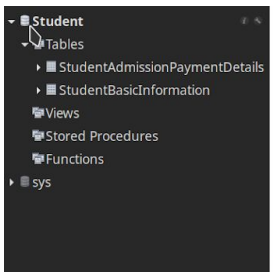
2 - Creating tables

a - creating student info table



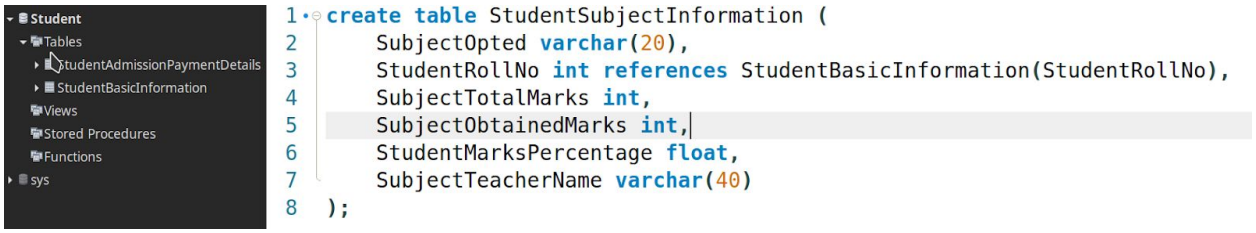

```
create table StudentBasicInformation (
    StudentRollNo int not null primary key,
    StudentName varchar(20),
    StudentSurname varchar(20),
    StudentAddress varchar(100),
    StudentGender varchar(1),
    GaurdianName varchar(40),
    GaurdianContact varchar(10),
    OptedForScholarship boolean
);
```

b - creating student payment details

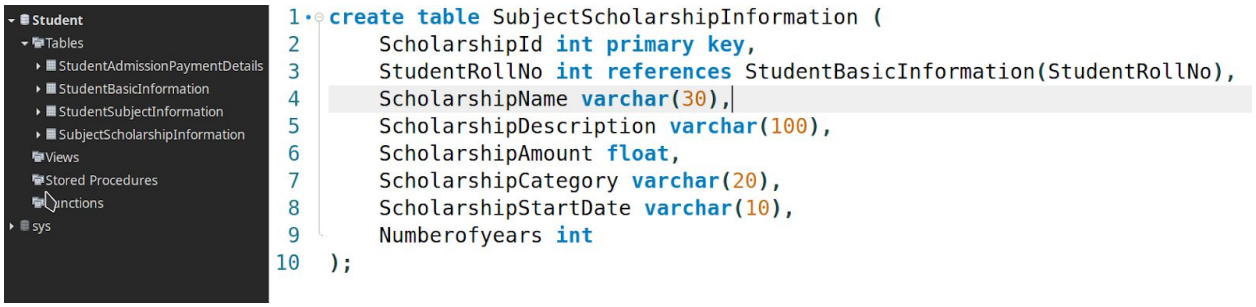



```
create table StudentAdmissionPaymentDetails (
    PaymentId int not null primary key,
    StudentRollNo int references StudentBasicInformation(StudentRollNo),
    AmountPaid float,
    AmountBalance float,
    PaymentMethod varchar(20),
    BankTransectionId varchar(10),
    PaymentDate varchar(10) -- DD/MM/YYYY
);
```

c - creating subject information



d - creating scholarship information



3, 4 - After inserting data in table

Student information table

StudentRollNo	StudentName	StudentSurname	StudentAddress	StudentGender	GaurdianName	GaurdianContact	OptedForScholarship
1	neeraj	kumar	12 main street, delhi	m	himanshu kumar	9492789285	1
2	anil	chaudhary	north delhi, delhi	m	pawan chaudh…	8420185792	1
3	prashant	sharma	gtb nagar, delhi	m	devender shar…	7947203849	0
4	sandeep	singh	malviya nagar, delhi	m	mayank singh	9134928344	1
5	sheetal	tiwari	mumbai, maharastra	f	naveen tiwari	6832982033	1
6	harsh	malik	east delhi, delhi	m	sandeep malik	9455548900	1
7	annu	kumari	kanpur, utter predesh	f	siddhant kumar	8746589989	1
8	ankur	yadav	noida, up	m	tushar yadav	9575292442	1
9	paras	singh	agra, up	m	shubham singh	9898812121	1
10	sheetal	tiwari	mumbai, maharastra	f	naveen tiwari	6832982033	1
11	pratosh	kumar	nagpur, maharashtra	m	abilash kumar	6832877728	1
12	shikha	aggrawal	dwarka, delhi	f	sushil aggrawal	7494279990	0
13	aastha	mishra	north delhi, delhi	f	shivam mishra	9994223880	0
14	uditi	gupta	ramnagar, bihar	f	anmol gupta	7771981888	1
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Student admission payment details

PaymentId	StudentRollNo	AmountPaid	AmountBalance	PaymentMethod	BankTransectionId	PaymentDate
100	4	10000	8000	debit card	tydjb08328	08/01/2021
101	2	15000	7000	cash	aakjh89248	03/01/2021
102	8	20000	12000	debit card	bbakd82498	03/01/2021
103	10	10000	2000	credit card	aahlf82502	08/01/2021
104	1	13000	9000	net banking	lojds79202	10/01/2021
105	14	14000	10000	net banking	kajdh23232	06/01/2021
106	7	25000	21000	debit card	ssdfc77839	04/01/2021
107	3	60000	50000	debit card	naadk32853	03/01/2021
108	6	30000	20000	net banking	opejd32800	02/01/2021
109	5	15000	11000	debit card	nvhdb85729	01/01/2021
110	9	17000	15000	credit card	modhc92844	09/01/2021
111	13	19000	9000	cash	ejdif85793	08/01/2021
112	11	12000	10000	cash	hfaks83958	07/01/2021
113	12	40000	30000	cash	fhakj17409	06/01/2021
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Student Subject information table

SubjectOpted	StudentRollNo	SubjectTotalMarks	SubjectObtainedMarks	StudentMarksPercentage	SubjectTeacherName
science	1	150	110	NULL	rakesh kumar
science	2	150	99	NULL	rakesh kumar
computer	3	150	88	NULL	anmol sharma
science	4	150	123	NULL	rakesh kumar
maths	5	100	73	NULL	tushar pandey
commerce	6	100	55	NULL	rakesh kumar
commerce	7	100	77	NULL	rakesh kumar
biology	8	100	67	NULL	ankita verma
science	9	150	133	NULL	rakesh kumar
science	10	150	70	NULL	rakesh kumar
commerce	11	100	83	NULL	rakesh kumar
computer	12	150	99	NULL	anmol sharma
biology	13	150	102	NULL	ankita verma
computer	14	150	113	NULL	anmol sharma

Scholarship information

ScholarshipId	StudentRollNo	ScholarshipName	ScholarshipDescription	ScholarshipAmount	ScholarshipCategory	ScholarshipStartDate	Numberofyears
401	1	B	Description b	6000	cat 2	02/01/2021	2
402	2	C	Description c	2000	cat 3	08/01/2021	1
403	4	A	Description a	8000	cat 1	03/01/2021	3
404	5	B	Description b	6000	cat 2	06/01/2021	2
405	7	B	Description b	6000	cat 2	02/01/2021	2
406	8	C	Description c	2000	cat 3	06/01/2021	1
407	9	A	Description a	8000	cat 1	04/01/2021	3
408	11	A	Description a	8000	cat 1	01/01/2021	3
409	14	B	Description b	6000	cat 2	01/01/2021	2
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

5 - Updating 5 records in table

1

```
update StudentBasicInformation
set StudentAddress='island no 2, andaman and nicobar'
where StudentRollNo=1;
```

2

```
update StudentBasicInformation
set GaurdianContact='7773289990'
where StudentRollNo=4;
```

3

```
update StudentBasicInformation
set StudentAddress='jaipur, rajsthan'
where StudentRollNo=14;
```

4

```
update StudentBasicInformation
set GaurdianName='Ramesh roy'
where StudentRollNo=10;
```

5

```
update StudentBasicInformation
set StudentSurname='kumar'
where StudentRollNo=6;
```

6 - Table after update

StudentRollNo	StudentName	StudentSurname	StudentAddress	StudentGender	GaurdianName	GaurdianContact	OptedForScholarship
1	neeraj	kumar	12 main street, delhi	m	himanshu kumar	9492789285	1
2	anil	chaudhary	north delhi, delhi	m	pawan chaudh…	8420185792	1
3	prashant	sharma	gtb nagar, delhi	m	devender shar…	7947203849	0
4	sandeep	singh	malviya nagar, delhi	m	mayank singh	9134928344	1
5	sheetal	tiwari	mumbai, maharastra	f	naveen tiwari	6832982033	1
6	harsh	malik	east delhi, delhi	m	sandeep malik	9455548900	1
7	annu	kumari	kanpur, utter predesh	f	siddhant kumar	8746589989	1
8	ankur	yadav	noida, up	m	tushar yadav	9575292442	1
9	paras	singh	agra, up	m	shubham singh	9898812121	1
10	sheetal	tiwari	mumbai, maharastra	f	naveen tiwari	6832982033	1
11	pratosh	kumar	nagpur, maharashtra	m	abilash kumar	6832877728	1
12	shikha	aggrawal	dwarka, delhi	f	sushil aggrawal	7494279990	0
13	aastha	mishra	north delhi, delhi	f	shivam mishra	9994223880	0
14	uditi	gupta	ramnagar, bihar	f	anmol gupta	7771981888	1
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

7 - Students with scholarship with more than 5000

```
1  select sbi.StudentRollNo, StudentName, StudentSurname,
2  StudentGender, ScholarshipCategory, ScholarshipAmount
3  from StudentBasicInformation sbi
4  inner join SubjectScholarshipInformation ssi
5      on sbi.StudentRollNo = ssi.StudentRollNo
6  where ScholarshipAmount > 5000;
7
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

#	StudentRollNo	StudentName	StudentSurname	StudentGender	ScholarshipCategory	ScholarshipAmount
1	1	neeraj	kumar	m	cat 2	6000
2	4	sandeep	singh	m	cat 1	8000
3	5	sheetal	tiwari	f	cat 2	6000
4	7	annu	kumari	f	cat 2	6000
5	9	paras	singh	m	cat 1	8000
6	11	pratosh	kumar	m	cat 1	8000
7	14	uditi	gupta	f	cat 2	6000

8 - who opted for scholarship but didn't get it

```sql
select *
from StudentBasicInformation
where OptedForScholarship=true and
    StudentRollNo not in (
    select StudentRollNo from SubjectScholarshipInformation
    );
```

Filter Rows: Edit: Export/Import: Wrap Cell Content:

StudentRollNo	StudentName	StudentSurname	StudentAddress	StudentGender	GaurdianName	GaurdianContact	OptedForScholarship
6	harsh	malik	east delhi, delhi	m	sandeep malik	9455548900	1
10	sheetal	tiwari	mumbai, maharastra	f	naveen tiwari	6832982033	1
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

9 - procedure to get percentage

```sql
DELIMITER //
create procedure calculate_percentage
()
begin
    update StudentSubjectInformation
    set StudentMarksPercentage=SubjectObtainedMarks*100/SubjectTotalMarks;
end //
DELIMITER ;

call calculate_percentage();
select * from StudentSubjectInformation;
```

Table after update

SubjectOpted	StudentRollNo	SubjectTotalMarks	SubjectObtainedMarks	StudentMarksPercentage	SubjectTeacherName
science	1	150	110	73.3333	rakesh kumar
science	2	150	99	66	rakesh kumar
computer	3	150	88	58.6667	anmol sharma
science	4	150	123	82	rakesh kumar
maths	5	100	73	73	tushar pandey
commerce	6	100	55	55	rakesh kumar
commerce	7	100	77	77	rakesh kumar
biology	8	100	67	67	ankita verma
science	9	150	133	88.6667	rakesh kumar
science	10	150	70	46.6667	rakesh kumar
commerce	11	100	83	83	rakesh kumar
computer	12	150	99	66	anmol sharma
biology	13	150	102	68	ankita verma
computer	14	150	113	75.3333	anmol sharma

10 - Stored procedure to find category based upon percentage

```
DELIMITER //
create procedure fill_catagory()
begin
    update SubjectScholarshipInformation
    set ScholarshipCategory='cat 1'
    where StudentRollNo in (
        select StudentRollNo
        from StudentSubjectInformation
        where StudentMarksPercentage >= 80
    );

    update SubjectScholarshipInformation
    set ScholarshipCategory='cat 2'
    where StudentRollNo in (
        select StudentRollNo
        from StudentSubjectInformation
        where StudentMarksPercentage >= 70 and StudentMarksPercentage < 80
    );

    update SubjectScholarshipInformation
    set ScholarshipCategory='cat 1'
    where StudentRollNo in (
        select StudentRollNo
        from StudentSubjectInformation
        where StudentMarksPercentage >= 60 and StudentMarksPercentage <70
    );
end //
DELIMITER ;
```

11 - View to show balance along with student details

```
1  create view studentBalance as
2      select sbi.StudentRollNo, StudentName, StudentSurname,
3      StudentGender, AmountBalance
4      from StudentBasicInformation sbi
5      inner join StudentAdmissionPaymentDetails sapd
6          on sbi.StudentRollNo = sapd.StudentRollNo;
7
8  select * from studentBalance;
```

Table

StudentRollNo	StudentName	StudentSurname	StudentGender	AmountBalance
4	sandeep	singh	m	8000
2	anil	chaudhary	m	7000
8	ankur	yadav	m	12000
10	sheetal	tiwari	f	2000
1	neeraj	kumar	m	9000
14	uditi	gupta	f	10000
7	annu	kumari	f	21000
3	prashant	sharma	m	50000
6	harsh	malik	m	20000
5	sheetal	tiwari	f	11000
9	paras	singh	m	15000
13	aastha	mishra	f	9000
11	pratosh	kumar	m	10000
12	shikha	aggrawal	f	30000

12 - students who haven't got any scholarship

```
1  select *
2  from StudentBasicInformation
3  where StudentRollNo not in (
4      select StudentRollNo from SubjectScholarshipInformation
5  );
```

Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

#	StudentRollNo	StudentName	StudentSurname	StudentAddress	StudentGender	GaurdianName	GaurdianContact	OptedForScholarship
	3	prashant	sharma	gtb nagar, delhi	m	devender shar…	7947203849	0
	6	harsh	malik	east delhi, delhi	m	sandeep malik	9455548900	1
	10	sheetal	tiwari	mumbai, maharastra	f	naveen tiwari	6832982033	1
	12	shikha	aggrawal	dwarka, delhi	f	sushil aggrawal	7494279990	0
	13	aastha	mishra	north delhi, delhi	f	shivam mishra	9994223880	0
	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

13 - stored procedure to get amount balance by roll number

```
1   DELIMITER //
2   create procedure get_balance
3   (in rollNo int)
4   begin
5       select AmountBalance
6       from StudentAdmissionPaymentDetails
7       where StudentRollNo = rollNo;
8   end //
9   DELIMITER ;
10  call get_balance(1);
```

Filter Rows: | Export: | Wrap Cell Content:

#	AmountBalance
1	9000

14 - Top five student details as per the StudentMarksPercentage (by subquery)

```
select *
from StudentBasicInformation
where StudentRollNo in (
    select StudentRollNo
    from StudentSubjectInformation
    order by StudentMarksPercentage desc
    limit 5
);
```

Table

StudentRollNo	StudentName	StudentSurname	StudentAddress	StudentGender	GaurdianName	GaurdianContact	OptedForScholarship
9	paras	singh	agra, up	m	shubham singh	9898812121	1
11	pratosh	kumar	nagpur, maharashtra	m	abilash kumar	6832877728	1
4	sandeep	singh	malviya nagar, delhi	m	mayank singh	9134928344	1
7	annu	kumari	kanpur, utter predesh	f	siddhant kumar	8746589989	1
14	uditi	gupta	ramnagar, bihar	f	anmol gupta	7771981888	1

15 - Uses of all three joins

```
select sbi.StudentRollNo, StudentName, StudentSurname, StudentMarksPercentage
from StudentBasicInformation sbi
inner join StudentSubjectInformation ssi
    on sbi.StudentRollNo = ssi.StudentRollNo;
```

StudentRollNo	StudentName	StudentSurname	StudentMarksPercentage
1	neeraj	kumar	73.3333
2	anil	chaudhary	66
3	prashant	sharma	58.6667
4	sandeep	singh	82
5	sheetal	tiwari	73
6	harsh	malik	55
7	annu	kumari	77
8	ankur	yadav	67
9	paras	singh	88.6667
10	sheetal	tiwari	46.6667

Inner joins - In the above data, we can see inner join in action. We can find the details of all the students with their respective percentage. Inner Joins are useful when we need to get the data from both the tables using some condition.

```
1  select sbi.StudentRollNo, StudentName, StudentSurname, ScholarshipCategory
2  from StudentBasicInformation sbi
3  left join SubjectScholarshipInformation ssi
4      on sbi.StudentRollNo = ssi.StudentRollNo;
```

StudentRollNo	StudentName	StudentSurname	ScholarshipCategory
1	neeraj	kumar	cat 2
2	anil	chaudhary	cat 3
3	prashant	sharma	NULL
4	sandeep	singh	cat 1
5	sheetal	tiwari	cat 2
6	harsh	malik	NULL
7	annu	kumari	cat 2
8	ankur	yadav	cat 3
9	paras	singh	cat 1
10	sheetal	tiwari	NULL

Left Join - Left joins can be used when we need to have all the rows of the left table, regardless of the matching condition. It is really useful when we need to see all the data as well as their category. If there is no category for data don't remove it like inner join rather show it with null.

```
1  select sbi.StudentRollNo, StudentName, StudentSurname, ScholarshipCategory
2  from SubjectScholarshipInformation ssi
3  right join StudentBasicInformation sbi
4      on sbi.StudentRollNo = ssi.StudentRollNo;
```

Right join - Right and left joins are nearly the same, just that left join shows all the table rows of the left table, whereas right join shows all the values of the right table. The above query result will be exactly the same as the left join because we swapped the left and right position in the table.

16 - Difference between delete drop and truncate

DELETE - Delete command is used for deleting one or more rows/records in the database. When deleted using the delete command it does not deallocate the

space table is using. Database will later make use of that space. After drop we can normally insert records in it.

DROP - Drop command is used for removing table/database (and other database objects). After removing it we can't use that object in the database.

TRUNCATE - Truncate command can be used for deleting all the records from the table. It also removes the allocated space for the table and makes the table of minimal size the same as when we created the table. With truncate the table structure and its columns, constraints, indexes will remain in the database.

17 - Count of the total number of student corresponding to each scholarship category

```
1  select ScholarshipCategory, count(*) as count
2  from SubjectScholarshipInformation
3  group by ScholarshipCategory;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

#	ScholarshipCategory	count
1	cat 2	4
2	cat 3	2
3	cat 1	3

18 - Maximum used scholarship category

```
1  select ScholarshipCategory
2  from SubjectScholarshipInformation
3  group by ScholarshipCategory
4  limit 1;
```

Filter Rows: | Export: | Wrap Cell Content:

#	ScholarshipCategory
1	cat 2

19 - Student with highest percentage with scholarship amount

```
1  select sbi.StudentRollNo, StudentName, ssi.StudentMarksPercentage,
2  sssi.ScholarshipAmount
3  from StudentBasicInformation sbi
4  inner join StudentSubjectInformation ssi
5      on sbi.StudentRollNo = ssi.StudentRollNo
6  inner join SubjectScholarshipInformation sssi
7      on sbi.StudentRollNo = sssi.StudentRollNo
8  order by ssi.StudentMarksPercentage desc
9  limit 1;
```

Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

#	StudentRollNo	StudentName	StudentMarksPercentage	ScholarshipAmount
1	9	paras	88.6667	8000

20 - Difference between Triggers, Stored procedures, views and functions

Triggers - Triggers are the piece of code that execute automatically whenever there is a related event, like insert, update, delete. They are stored at tabular level. For example we can invoke a trigger when there is deletion in the table to store history.

Stored Procedures - Stored procedure is the sequence of sql statements that takes a list of input and gives back output. It can be used on code that we need to run again and again. Stored procedures are stored at database level.

Views - Views are the virtual table that is derived from one or more physical tables. They are the resultset of SELECT statements. Views can help in improving the performance of the database by avoiding joins again and again which could be really costly operation.

Functions - Function in database are the programs that database provides. They take input and return the value depending on the function itself. There are a lot of built-in functions in databases like mysql, example - MAX(), AVG(), COUNT(), SUM() etc.