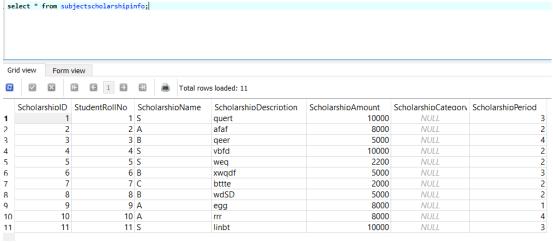
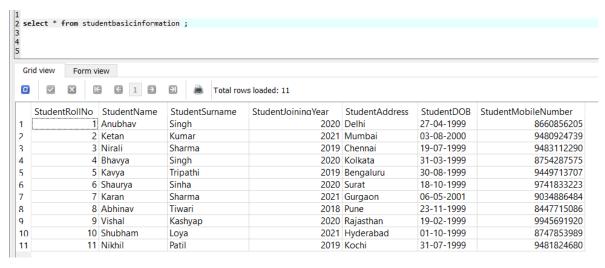
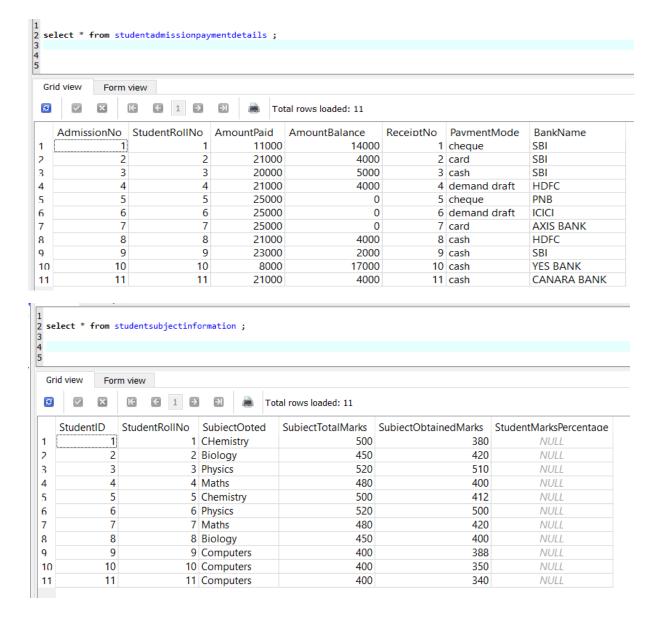
SQL ASSIGNMENT

```
1. CREATE DATABASE Student:
2. CREATE TABLE studentbasicinformation (
  StudentRollNo INT NOT NULL,
  StudentName VARCHAR (45),
  StudentSurname VARCHAR (45),
  StudentJoiningYear INT (4),
  StudentAddress VARCHAR (45),
  StudentDOB DATE,
  StudentMobileNumber INT,
  PRIMARY KEY (
  StudentRollNo
  )
  );
3. CREATE TABLE studentadmissionpaymentdetails (
  AdmissionNo INT,
  StudentRollNo INT,
  AmountPaid INT,
  AmountBalance INT,
  PaymentDate DATE,
  PaymentMode VARCHAR (45),
  BankBranch VARCHAR (100),
  PRIMARY KEY (
  AdmissionNo
  FOREIGN KEY (
  StudentRollNo
  REFERENCES studentbasicinformation (StudentRollNo)
  );
4. CREATE TABLE studentsubjectinformation (
  StudentID INT NOT NULL,
  StudentRollNo INT,
  StudentOpted VARCHAR (45),
  SubjectTotalMarks INT,
  SubjectObtainedMarks INT,
  StudentMarksPercentage INT,
  PRIMARY KEY (
  StudentID
  ),
  FOREIGN KEY (
  StudentRollNo
  REFERENCES studentbasicinformation (StudentRollNO)
  );
```

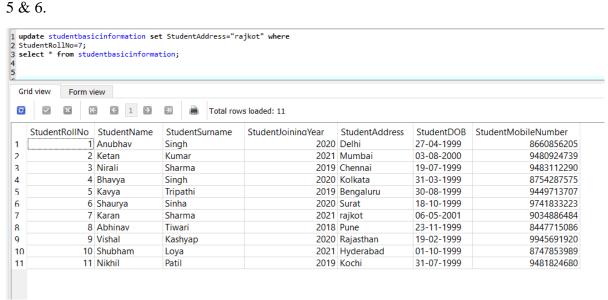
```
5. CREATE TABLE subjects cholar ship info (
  ScholarshipID INT,
  StudentRollNo INT,
  ScholarshipName VARCHAR (45),
  ScholarshipDescription VARCHAR (45),
  ScholarshipAmount INT,
  ScholarshipCategory INT,
  ScholarshipPeriod INT,
  PRIMARY KEY (
  ScholarshipID
  ),
  FOREIGN KEY (
  StudentRollNo
  REFERENCES studentbasicinformation (StudentRollNo)
);
3 & 4.
```

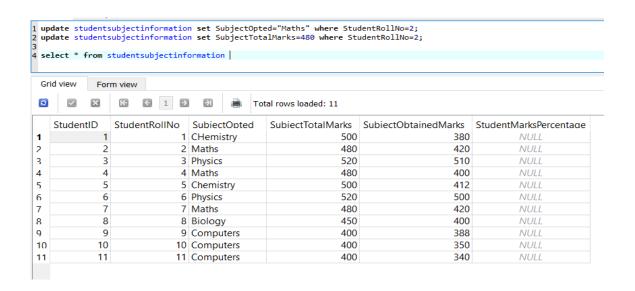


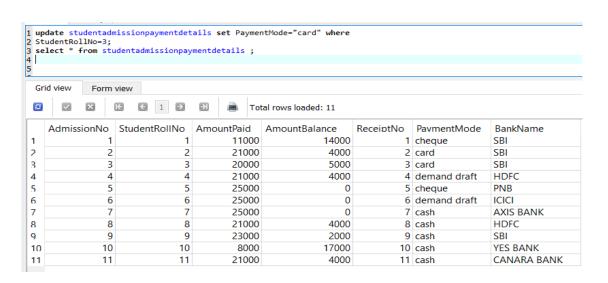


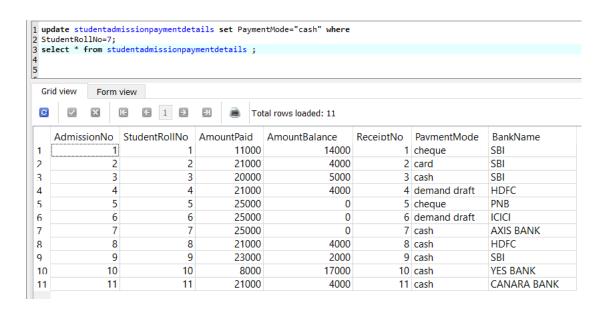


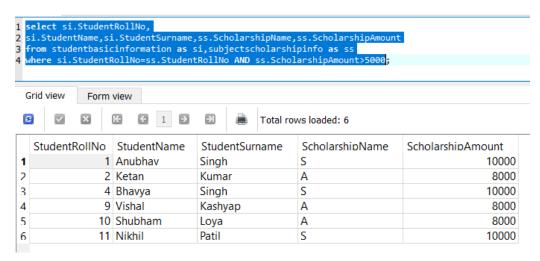
5 & 6.



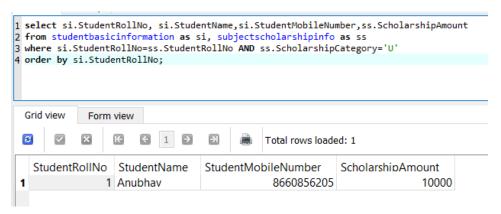








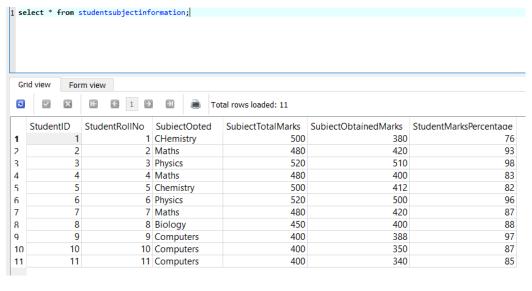
8. Students with U scholarship catergory haven't received their scholarships yet.



9.

The percentage is calculated as obtained/total marks.

```
drop procedure if exists calc_percentage;
DELIMITER $$
create procedure calc_percentage()
begin
update studentsubjectinformation set StudentMArksPercentage = (SubjectTotalMarks)/5;
end$$
call calc_percentage();
```



drop procedure if exists calc_category;

DELIMITER \$\$

create procedure calc_category()

begin

update subjectscholarshipinfo ss inner join studentsubjectinformation su on ss.StudentRollNo=su.StudentRollNo

set ss.ScholarshipCategory = case

when su.StudentMArksPercentage>=90 then 'L'

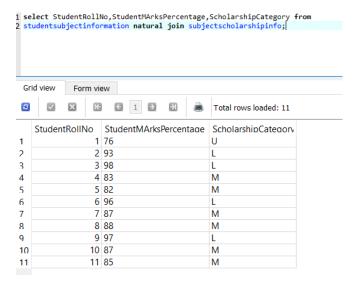
when su.StudentMArksPercentage>=80 then 'M'

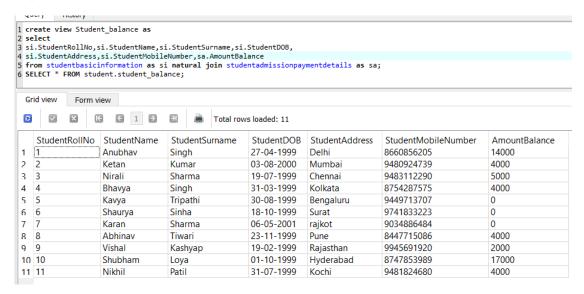
when su.StudentMArksPercentage>=70 then 'U'

end;

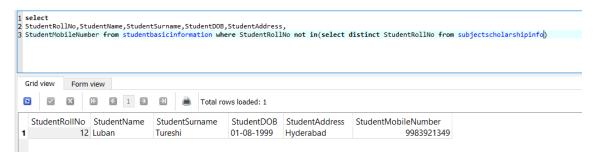
END\$\$

call calc_category();





12.



13.

DELIMITER \$\$

create procedure student balance(IN rollno int)

begin

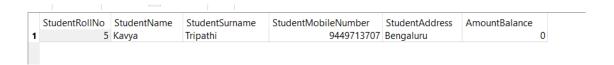
select

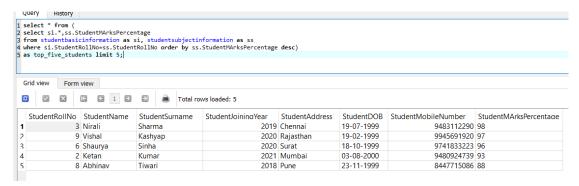
si.StudentRollNo,si.StudentName,si.StudentSurname,si.StudentMobileNumber, si.StudentAddress, sa.AmountBalance from studentbasicinformation as si, studentadmissionpaymentdetails as sa where si.StudentRollNo=rollno and

si.StudentRollNo=sa.StudentRollNo;

end \$\$

call student balance(5);

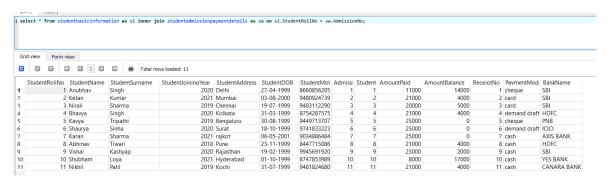




15. JOINS

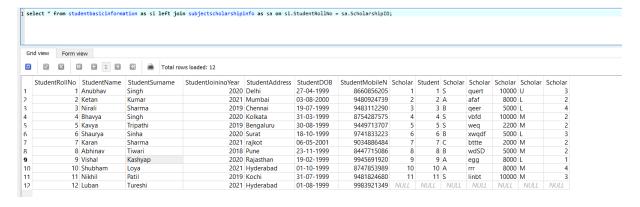
INNER JOIN

This join can be used for the basic info and the admission details so that we can see the student and the corresponding admission information about the student.



LEFT OUTER JOIN

This join can be used on the basic info and scholarship info and this can be prepared so as to check which student has not applied for or got scholarships.



RIGHT OUTER JOIN

The join can be used with the basic info and the subject selected table to find out the students who have selected a particular subject.

select * from studentbasicinformation as si right join studentsubjectinformation as sa on si.StudentRollNo = sa.StudentID;

	Student	Student	SubiectOp	Subject	Subject	Student	StudentRollNo:	StudentName	StudentSurname	StudentJoiningYear	StudentAddress	StudentDOB	StudentMobileN
1	1	1	CHemistry	500	380	76	1	Anubhav	Singh	2020	Delhi	27-04-1999	8660856205
2	2	2	Maths	480	420	93	2	Ketan	Kumar	2021	Mumbai	03-08-2000	9480924739
3	3	3	Physics	520	510	98	3	Nirali	Sharma	2019	Chennai	19-07-1999	9483112290
4	4	4	Maths	480	400	83	4	Bhavya	Singh	2020	Kolkata	31-03-1999	8754287575
5	5	5	Chemistry	500	412	82	5	Kavya	Tripathi	2019	Bengaluru	30-08-1999	9449713707
6	6	6	Physics	520	500	96	6	Shaurya	Sinha	2020	Surat	18-10-1999	9741833223
7	7	7	Maths	480	420	87	7	Karan	Sharma	2021	rajkot	06-05-2001	9034886484
8	8	8	Biology	450	400	88	8	Abhinav	Tiwari	2018	Pune	23-11-1999	8447715086
9	9	9	Computers	400	388	97	9	Vishal	Kashyap	2020	Rajasthan	19-02-1999	9945691920
10	10	10	Computers	400	350	87	10	Shubham	Loya	2021	Hyderabad	01-10-1999	8747853989
11	11	11	Computers	400	340	85	11	Nikhil	Patil	2019	Kochi	31-07-1999	9481824680

16.

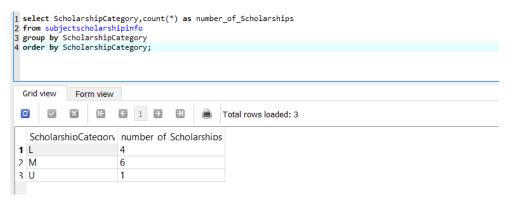
Difference between drop, delete and truncate:

DROP: It is a DDL command where all the contents of the table are lost, i.e. the rows as well as the attributes(columns) are deleted and the space for the table is also freed.

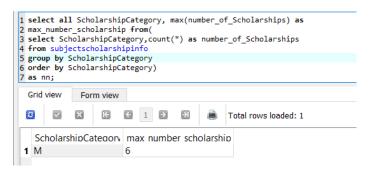
DELETE: It is a DML command where a particular row is deleted based on a particular condition. If the condition is met that particular row is deleted.

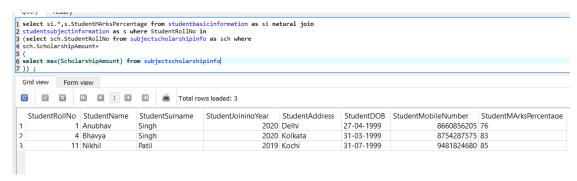
TRUNCATE: It is DDL command where all the rows are deleted but the attributes(columns) in the table are not deleted

17.



18.





20.

Triggers: It is a stored program which is invoked automatically when an event occurs such as an insertion, deletion or updation of a table.

Stored Procedures: A stored procedure is a query in the form of a function which has to be invoked again and again. Thus, in this the stored procedure which is defined only once can be called again instead of writing the query again.

Views: A view is a virtual table which is formed by joining 2 or more tables and has columns and rows. This is helpful when a particular joined table needs to be accessed again and again.

Functions: The functions can be used for summarizing data in a table. Some of the most commonly functions provided by SQL are AVG(),COUNT(),MAX(),MIN().