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 subjectscholarshipinfo (
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.

```
select * from subjectscholarshipinfo;
```

Grid view | Form view

Total rows loaded: 11

	ScholarshipID	StudentRollNo	ScholarshipName	ScholarshipDescription	ScholarshipAmount	ScholarshipCategory	ScholarshipPeriod
1	1	1	S	quert	10000	NULL	3
2	2	2	A	afaf	8000	NULL	2
3	3	3	B	qeer	5000	NULL	4
4	4	4	S	vbfd	10000	NULL	2
5	5	5	S	weq	2200	NULL	2
6	6	6	B	xwqdf	5000	NULL	3
7	7	7	C	bttte	2000	NULL	2
8	8	8	B	wdSD	5000	NULL	2
9	9	9	A	egg	8000	NULL	1
10	10	10	A	rrr	8000	NULL	4
11	11	11	S	linbt	10000	NULL	3

```
select * from studentbasicinformation ;
```

Grid view | Form view

Total rows loaded: 11

	StudentRollNo	StudentName	StudentSurname	StudentJoiningYear	StudentAddress	StudentDOB	StudentMobileNumber
1	1	Anubhav	Singh	2020	Delhi	27-04-1999	8660856205
2	2	Ketan	Kumar	2021	Mumbai	03-08-2000	9480924739
3	3	Nirali	Sharma	2019	Chennai	19-07-1999	9483112290
4	4	Bhavya	Singh	2020	Kolkata	31-03-1999	8754287575
5	5	Kavya	Tripathi	2019	Bengaluru	30-08-1999	9449713707
6	6	Shaurya	Sinha	2020	Surat	18-10-1999	9741833223
7	7	Karan	Sharma	2021	Gurgaon	06-05-2001	9034886484
8	8	Abhinav	Tiwari	2018	Pune	23-11-1999	8447715086
9	9	Vishal	Kashyap	2020	Rajasthan	19-02-1999	9945691920
10	10	Shubham	Loya	2021	Hyderabad	01-10-1999	8747853989
11	11	Nikhil	Patil	2019	Kochi	31-07-1999	9481824680

```
select * from studentadmissionpaymentdetails ;
```

Grid view | Form view

Total rows loaded: 11

	AdmissionNo	StudentRollNo	AmountPaid	AmountBalance	ReceiptNo	PaymentMode	BankName
1	1	1	11000	14000	1	cheque	SBI
2	2	2	21000	4000	2	card	SBI
3	3	3	20000	5000	3	cash	SBI
4	4	4	21000	4000	4	demand draft	HDFC
5	5	5	25000	0	5	cheque	PNB
6	6	6	25000	0	6	demand draft	ICICI
7	7	7	25000	0	7	card	AXIS BANK
8	8	8	21000	4000	8	cash	HDFC
9	9	9	23000	2000	9	cash	SBI
10	10	10	8000	17000	10	cash	YES BANK
11	11	11	21000	4000	11	cash	CANARA BANK

```
select * from studentsubjectinformation ;
```

Grid view | Form view

Total rows loaded: 11

	StudentID	StudentRollNo	SubjectOpted	SubjectTotalMarks	SubjectObtainedMarks	StudentMarksPercentage
1	1	1	CHemistry	500	380	NULL
2	2	2	Biology	450	420	NULL
3	3	3	Physics	520	510	NULL
4	4	4	Maths	480	400	NULL
5	5	5	Chemistry	500	412	NULL
6	6	6	Physics	520	500	NULL
7	7	7	Maths	480	420	NULL
8	8	8	Biology	450	400	NULL
9	9	9	Computers	400	388	NULL
10	10	10	Computers	400	350	NULL
11	11	11	Computers	400	340	NULL

5 & 6.

```
update studentbasicinformation set StudentAddress="rajkot" where
StudentRollNo=7;
select * from studentbasicinformation;
```

Grid view | Form view

Total rows loaded: 11

	StudentRollNo	StudentName	StudentSurname	StudentJoiningYear	StudentAddress	StudentDOB	StudentMobileNumber
1	1	Anubhav	Singh	2020	Delhi	27-04-1999	8660856205
2	2	Ketan	Kumar	2021	Mumbai	03-08-2000	9480924739
3	3	Nirali	Sharma	2019	Chennai	19-07-1999	9483112290
4	4	Bhavya	Singh	2020	Kolkata	31-03-1999	8754287575
5	5	Kavya	Tripathi	2019	Bengaluru	30-08-1999	9449713707
6	6	Shaurya	Sinha	2020	Surat	18-10-1999	9741833223
7	7	Karan	Sharma	2021	rajkot	06-05-2001	9034886484
8	8	Abhinav	Tiwari	2018	Pune	23-11-1999	8447715086
9	9	Vishal	Kashyap	2020	Rajasthan	19-02-1999	9945691920
10	10	Shubham	Loya	2021	Hyderabad	01-10-1999	8747853989
11	11	Nikhil	Patil	2019	Kochi	31-07-1999	9481824680

```
update studentsubjectinformation set SubjectOpted="Maths" where StudentRollNo=2;
update studentsubjectinformation set SubjectTotalMarks=480 where StudentRollNo=2;

select * from studentsubjectinformation
```

Grid view | Form view

Total rows loaded: 11

	StudentID	StudentRollNo	SubjectOpted	SubjectTotalMarks	SubjectObtainedMarks	StudentMarksPercentage
1	1	1	CHemistry	500	380	NULL
2	2	2	Maths	480	420	NULL
3	3	3	Physics	520	510	NULL
4	4	4	Maths	480	400	NULL
5	5	5	Chemistry	500	412	NULL
6	6	6	Physics	520	500	NULL
7	7	7	Maths	480	420	NULL
8	8	8	Biology	450	400	NULL
9	9	9	Computers	400	388	NULL
10	10	10	Computers	400	350	NULL
11	11	11	Computers	400	340	NULL

```
update studentadmissionpaymentdetails set PaymentMode="card" where
StudentRollNo=3;
select * from studentadmissionpaymentdetails ;
```

Grid view | Form view

Total rows loaded: 11

	AdmissionNo	StudentRollNo	AmountPaid	AmountBalance	ReceiptNo	PaymentMode	BankName
1	1	1	11000	14000	1	cheque	SBI
2	2	2	21000	4000	2	card	SBI
3	3	3	20000	5000	3	card	SBI
4	4	4	21000	4000	4	demand draft	HDFC
5	5	5	25000	0	5	cheque	PNB
6	6	6	25000	0	6	demand draft	ICICI
7	7	7	25000	0	7	cash	AXIS BANK
8	8	8	21000	4000	8	cash	HDFC
9	9	9	23000	2000	9	cash	SBI
10	10	10	8000	17000	10	cash	YES BANK
11	11	11	21000	4000	11	cash	CANARA BANK

```
update studentadmissionpaymentdetails set PaymentMode="cash" where
StudentRollNo=7;
select * from studentadmissionpaymentdetails ;
```

Grid view | Form view

Total rows loaded: 11

	AdmissionNo	StudentRollNo	AmountPaid	AmountBalance	ReceiptNo	PaymentMode	BankName
1	1	1	11000	14000	1	cheque	SBI
2	2	2	21000	4000	2	card	SBI
3	3	3	20000	5000	3	cash	SBI
4	4	4	21000	4000	4	demand draft	HDFC
5	5	5	25000	0	5	cheque	PNB
6	6	6	25000	0	6	demand draft	ICICI
7	7	7	25000	0	7	cash	AXIS BANK
8	8	8	21000	4000	8	cash	HDFC
9	9	9	23000	2000	9	cash	SBI
10	10	10	8000	17000	10	cash	YES BANK
11	11	11	21000	4000	11	cash	CANARA BANK

7.

```
select si.StudentRollNo,
si.StudentName,si.StudentSurname,ss.ScholarshipName,ss.ScholarshipAmount
from studentbasicinformation as si,subjectscholarshipinfo as ss
where si.StudentRollNo=ss.StudentRollNo AND ss.ScholarshipAmount>5000;
```

Grid view | Form view

Total rows loaded: 6

	StudentRollNo	StudentName	StudentSurname	ScholarshipName	ScholarshipAmount
1	1	Anubhav	Singh	S	10000
2	2	Ketan	Kumar	A	8000
3	4	Bhavya	Singh	S	10000
4	9	Vishal	Kashyap	A	8000
5	10	Shubham	Loya	A	8000
6	11	Nikhil	Patil	S	10000

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

```
select si.StudentRollNo, si.StudentName,si.StudentMobileNumber,ss.ScholarshipAmount
from studentbasicinformation as si, subjectscholarshipinfo as ss
where si.StudentRollNo=ss.StudentRollNo AND ss.ScholarshipCategory='U'
order by si.StudentRollNo;
```

Grid view | Form view

Total rows loaded: 1

	StudentRollNo	StudentName	StudentMobileNumber	ScholarshipAmount
1	1	Anubhav	8660856205	10000

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();
```

```
1 select * from studentsubjectinformation;
```

Grid view | Form view

Total rows loaded: 11

	StudentID	StudentRollNo	SubjectOpted	SubjectTotalMarks	SubjectObtainedMarks	StudentMarksPercentage
1	1	1	CHemistry	500	380	76
2	2	2	Maths	480	420	93
3	3	3	Physics	520	510	98
4	4	4	Maths	480	400	83
5	5	5	Chemistry	500	412	82
6	6	6	Physics	520	500	96
7	7	7	Maths	480	420	87
8	8	8	Biology	450	400	88
9	9	9	Computers	400	388	97
10	10	10	Computers	400	350	87
11	11	11	Computers	400	340	85

10.

```
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();
```

```
1 select StudentRollNo,StudentMArksPercentage,ScholarshipCategory from
2 studentsubjectinformation natural join subjectscholarshipinfo;
```

Grid view | Form view

Total rows loaded: 11

	StudentRollNo	StudentMArksPercentage	ScholarshipCategory
1	1	76	U
2	2	93	L
3	3	98	L
4	4	83	M
5	5	82	M
6	6	96	L
7	7	87	M
8	8	88	M
9	9	97	L
10	10	87	M
11	11	85	M

11.

```
create view Student_balance as
select
si.StudentRollNo,si.StudentName,si.StudentSurname,si.StudentDOB,
si.StudentAddress,si.StudentMobileNumber,sa.AmountBalance
from studentbasicinformation as si natural join studentadmissionpaymentdetails as sa;
SELECT * FROM student.student_balance;
```

Grid view | Form view

Total rows loaded: 11

	StudentRollNo	StudentName	StudentSurname	StudentDOB	StudentAddress	StudentMobileNumber	AmountBalance
1	1	Anubhav	Singh	27-04-1999	Delhi	8660856205	14000
2	2	Ketan	Kumar	03-08-2000	Mumbai	9480924739	4000
3	3	Nirali	Sharma	19-07-1999	Chennai	9483112290	5000
4	4	Bhavya	Singh	31-03-1999	Kolkata	8754287575	4000
5	5	Kavya	Tripathi	30-08-1999	Bengaluru	9449713707	0
6	6	Shaurya	Sinha	18-10-1999	Surat	9741833223	0
7	7	Karan	Sharma	06-05-2001	rajkot	9034886484	0
8	8	Abhinav	Tiwari	23-11-1999	Pune	8447715086	4000
9	9	Vishal	Kashyap	19-02-1999	Rajasthan	9945691920	2000
10	10	Shubham	Loya	01-10-1999	Hyderabad	8747853989	17000
11	11	Nikhil	Patil	31-07-1999	Kochi	9481824680	4000

12.

```
select
StudentRollNo,StudentName,StudentSurname,StudentDOB,StudentAddress,
StudentMobileNumber from studentbasicinformation where StudentRollNo not in(select distinct StudentRollNo from subjectscholarshipinfo)
```

Grid view | Form view

Total rows loaded: 1

	StudentRollNo	StudentName	StudentSurname	StudentDOB	StudentAddress	StudentMobileNumber
1	12	Luban	Tureshi	01-08-1999	Hyderabad	9983921349

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);
```

	StudentRollNo	StudentName	StudentSurname	StudentMobileNumber	StudentAddress	AmountBalance
1	5	Kavya	Tripathi	9449713707	Bengaluru	0

14.

```
select * from (
select si.*,ss.StudentMArksPercentage
from studentbasicinformation as si, studentsubjectinformation as ss
where si.StudentRollNo=ss.StudentRollNo order by ss.StudentMArksPercentage desc)
as top_five_students limit 5;
```

Grid view | Form view — Total rows loaded: 5

	StudentRollNo	StudentName	StudentSurname	StudentJoiningYear	StudentAddress	StudentDOB	StudentMobileNumber	StudentMArksPercentage
1	3	Nirali	Sharma	2019	Chennai	19-07-1999	9483112290	98
2	9	Vishal	Kashyap	2020	Rajasthan	19-02-1999	9945691920	97
3	6	Shaurya	Sinha	2020	Surat	18-10-1999	9741833223	96
4	2	Ketan	Kumar	2021	Mumbai	03-08-2000	9480924739	93
5	8	Abhinav	Tiwari	2018	Pune	23-11-1999	8447715086	88

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.

```
select * from studentbasicinformation as si inner join studentadmissionpaymentdetails as sa on si.StudentRollNo = sa.AdmissionNo;
```

Grid view | Form view — Total rows loaded: 11

	StudentRollNo	StudentName	StudentSurname	StudentJoiningYear	StudentAddress	StudentDOB	StudentMobileNumber	AdmissionNo	StudentRollNo	AmountPaid	AmountBalance	ReceiptNo	PaymentMode	BankName
1	1	Anubhav	Singh	2020	Delhi	27-04-1999	8660856205	1	1	11000	14000	1	cheque	SBI
2	2	Ketan	Kumar	2021	Mumbai	03-08-2000	9480924739	2	2	21000	4000	2	card	SBI
3	3	Nirali	Sharma	2019	Chennai	19-07-1999	9483112290	3	3	20000	5000	3	card	SBI
4	4	Bhavya	Singh	2020	Kolkata	31-03-1999	8754287575	4	4	21000	4000	4	demand draft	HDFC
5	5	Kavya	Tripathi	2019	Bengaluru	30-08-1999	9449713707	5	5	25000	0	5	cheque	PNB
6	6	Shaurya	Sinha	2020	Surat	18-10-1999	9741833223	6	6	25000	0	6	demand draft	ICICI
7	7	Karan	Sharma	2021	rajkot	06-05-2001	9034886484	7	7	25000	0	7	cash	AXIS BANK
8	8	Abhinav	Tiwari	2018	Pune	23-11-1999	8447715086	8	8	21000	4000	8	cash	HDFC
9	9	Vishal	Kashyap	2020	Rajasthan	19-02-1999	9945691920	9	9	23000	2000	9	cash	SBI
10	10	Shubham	Loya	2021	Hyderabad	01-10-1999	8747853989	10	10	8000	17000	10	cash	YES BANK
11	11	Nikhil	Patil	2019	Kochi	31-07-1999	9481824680	11	11	21000	4000	11	cash	CANARA BANK

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.

```
select * from studentbasicinformation as si left join subjectscholarshipinfo as sa on si.StudentRollNo = sa.ScholarshipID;
```

Grid view | Form view — Total rows loaded: 12

	StudentRollNo	StudentName	StudentSurname	StudentJoiningYear	StudentAddress	StudentDOB	StudentMobileN	Scholar	Student	Scholar	Scholar	Scholar	Scholar	Scholar
1	1	Anubhav	Singh	2020	Delhi	27-04-1999	8660856205	1	1	S	quert	10000	U	3
2	2	Ketan	Kumar	2021	Mumbai	03-08-2000	9480924739	2	2	A	afaf	8000	L	2
3	3	Nirali	Sharma	2019	Chennai	19-07-1999	9483112290	3	3	B	qeer	5000	L	4
4	4	Bhavya	Singh	2020	Kolkata	31-03-1999	8754287575	4	4	S	vbfd	10000	M	2
5	5	Kavya	Tripathi	2019	Bengaluru	30-08-1999	9449713707	5	5	S	weq	2200	M	2
6	6	Shaurya	Sinha	2020	Surat	18-10-1999	9741833223	6	6	B	xwqdf	5000	L	3
7	7	Karan	Sharma	2021	rajkot	06-05-2001	9034886484	7	7	C	bttte	2000	M	2
8	8	Abhinav	Tiwari	2018	Pune	23-11-1999	8447715086	8	8	B	wdSD	5000	M	2
9	9	Vishal	Kashyap	2020	Rajasthan	19-02-1999	9945691920	9	9	A	egg	8000	L	1
10	10	Shubham	Loya	2021	Hyderabad	01-10-1999	8747853989	10	10	A	rrr	8000	M	4
11	11	Nikhil	Patil	2019	Kochi	31-07-1999	9481824680	11	11	S	linbt	10000	M	3
12	12	Luban	Tureshi	2021	Hyderabad	01-08-1999	9983921349	NULL	NULL	NULL	NULL	NULL	NULL	NULL

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	Subiect	Subiect	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.

```
select ScholarshipCategory,count(*) as number_of_Scholarships
from subjectscholarshipinfo
group by ScholarshipCategory
order by ScholarshipCategory;
```

Grid view | Form view

Total rows loaded: 3

	ScholarshipCategory	number of Scholarships
1	L	4
2	M	6
3	U	1

18.

```
select all ScholarshipCategory, max(number_of_Scholarships) as
max_number_scholarship from(
select ScholarshipCategory,count(*) as number_of_Scholarships
from subjectscholarshipinfo
group by ScholarshipCategory
order by ScholarshipCategory)
as nn;
```

Grid view | Form view

Total rows loaded: 1

	ScholarshipCategory	max number scholarship
1	M	6

19.

```
select si.*,s.StudentMArksPercentage from studentbasicinformation as si natural join
studentsubjectinformation as s where StudentRollNo in
(select sch.StudentRollNo from subjectscholarshipinfo as sch where
sch.ScholarshipAmount=
(
select max(ScholarshipAmount) from subjectscholarshipinfo
)) ;
```

Grid view | Form view

Total rows loaded: 3

	StudentRollNo	StudentName	StudentSurname	StudentJoiningYear	StudentAddress	StudentDOB	StudentMobileNumber	StudentMArksPercentage
1	1	Anubhav	Singh	2020	Delhi	27-04-1999	8660856205	76
2	4	Bhavya	Singh	2020	Kolkata	31-03-1999	8754287575	83
3	11	Nikhil	Patil	2019	Kochi	31-07-1999	9481824680	85

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().