

SQL QUERIES

1. CREATE DATABASE Student;
2. CREATE TABLE `student`.`studentbasicinformation` (
`StudentRollNo` INT NOT NULL ,
`StudentName` VARCHAR(45) ,
`StudentSurname` VARCHAR(45) ,
`StudentAddress` VARCHAR(45) ,
`StudentDOB` DATE ,
`StudentGender` VARCHAR(1) ,
`StudentMobileNumber` INT,
PRIMARY KEY (`StudentRollNo`));

```
CREATE TABLE studentadmissionpaymentdetails (  
AdmissionID INT,  
StudentRollNo INT,  
AmountPaid INT,  
AmountBalance INT,  
DateOfPayment DATE,  
ModeOfPayment VARCHAR(45),  
PRIMARY KEY(AdmissionID),  
FOREIGN KEY (StudentRollNo) REFERENCES studentbasicinformation  
(StudentRollNo));
```

```
CREATE TABLE studentsubjectinformation (  
StudentInfoID INT NOT NULL,  
StudentRollNo INT,  
StudentOpted VARCHAR(45),  
SubjectTotalMarks INT,  
SubjectObtainedMarks INT, StudentMarksPercentage INT,  
PRIMARY KEY(StudentInfoID),  
FOREIGN KEY(StudentRollNo) REFERENCES  
studentbasicinformation(StudentRollNo));
```

```
CREATE TABLE subjectscholarshipinfo (  
ScholarshipID INT,  
StudentRollNo INT,  
ScholarshipName VARCHAR(45),  
ScholarshipDescription VARCHAR(45),  
ScholarshipAmount INT,  
ScholarshipCategory INT,  
ScholarshipObtainedDate DATE,
```

PRIMARY KEY (ScholarshipID),
FOREIGN KEY (StudentRollNo) REFERENCES studentbasicinformation
(StudentRollNo));

3 & 4

```
1 select * from studentbasicinformation;
```

StudentRollNo	StudentName	StudentSurname	StudentAddress	StudentDOB	StudentGender	StudentMobileNumber
1	niyati	mehta	bangalore	1999-05-31	f	87091234
2	abc	xyz	mumbai	2000-03-21	m	128956342
3	john	doe	indore	1994-06-12	m	22890123
4	alan	walker	new york	1992-01-09	m	905612456
5	harry	potter	london	1997-07-31	m	90876234
6	chloe	NULL	california	1989-08-29	f	22456123
7	jasmine	mavani	vadodara	1998-11-22	f	233467856
8	adam	smith	goa	1997-09-21	m	21345265
9	nadia	tarannum	bhopal	1998-11-01	f	237512345
10	manisha	nahta	udaipur	1998-08-04	f	84253612
NULL	NULL	NULL	NULL	NULL	NULL	NULL

```
1 select * from studentadmissionpaymentdetails;
```

AdmissionID	StudentRollNo	AmountPaid	AmountBalance	DateOfPayment	ModeOfPayment
1	1	20000	5000	2020-03-02	cash
2	2	21000	4000	2020-12-12	credit
3	3	19000	6000	2019-08-21	cash
4	4	20000	5000	2020-07-31	cheque
5	5	18000	7000	2018-05-09	credit
6	6	22000	3000	2019-11-28	cash
7	7	21000	4000	2018-09-11	credit
8	8	19000	6000	2020-12-11	cheque
9	9	24000	1000	2019-11-09	cash
10	10	18000	7000	2020-01-12	cheque
NULL	NULL	NULL	NULL	NULL	NULL

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

Result Grid						
Filter Rows:						
Edit:						
Export/Import:						
Wrap Cell Content:						
StudentInfoID	StudentRollNo	SubjectOpted	SubjectTotalMarks	SubjectObtainedMarks	StudentMarksPercentage	
1	1	Maths	450	89	NULL	
2	2	Physics	390	75	NULL	
3	3	Computers	470	80	NULL	
4	4	Maths	470	90	NULL	
5	5	Biology	400	69	NULL	
6	6	Computers	480	95	NULL	
7	7	Physics	390	84	NULL	
8	8	Biology	460	80	NULL	
9	9	Maths	445	79	NULL	
10	10	Computers	465	88	NULL	
NULL	NULL	NULL	NULL	NULL	NULL	

```
1 select * from subjectscholarshipinfo;
```

Result Grid							
Filter Rows:							
Edit:							
Export/Import:							
Wrap Cell Content:							
ScholarshipID	StudentRollNo	ScholarshipName	ScholarshipDescription	ScholarshipAmount	ScholarshipCategory	ScholarshipObtainedDate	
1	1	Sc1	abc	5000	NULL	2021-07-12	
2	3	Sc2	qwe	6700	NULL	2021-03-01	
3	4	Sc3	rty	3200	NULL	2020-11-09	
4	1	Sc2	abc	5000	NULL	2020-09-21	
5	5	Sc2	qwe	6100	NULL	2021-01-02	
6	7	Sc4	ghj	1000	NULL	2021-02-09	
7	3	Sc6	qwe	3200	NULL	2020-05-12	
8	2	Sc3	rty	7000	NULL	2021-05-31	
9	5	Sc5	zxc	5500	NULL	2021-11-11	
10	2	Sc1	abc	5100	NULL	2021-07-22	
NULL	NULL	NULL	NULL	NULL	NULL	NULL	

Updation List

5.& 6.

(i) update studentbasicinformation set StudentAddress="rajkot" where StudentRollNo=7;

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

StudentRollNo	StudentName	StudentSurname	StudentAddress	StudentDOB	StudentGender	StudentMobileNumber
1	niyati	mehta	bangalore	1999-05-31	f	87091234
2	abc	xyz	mumbai	2000-03-21	m	128956342
3	john	doe	indore	1994-06-12	m	22890123
4	alan	walker	new york	1992-01-09	m	905612456
5	harry	potter	london	1997-07-31	m	90876234
6	chloe	NULL	california	1989-08-29	f	22456123
7	jasmine	mavani	rajkot	1998-11-22	f	233467856
8	adam	smith	goa	1997-09-21	m	21345265
9	nadia	tarannum	bhopal	1998-11-01	f	237512345
10	manisha	nahta	udaipur	1998-08-04	f	84253612
NULL	NULL	NULL	NULL	NULL	NULL	NULL

(ii) update studentadmissionpaymentdetails set ModeOfPayment="cash" where StudentRollNo=2;

```
1 • update studentadmissionpaymentdetails set ModeOfpayment="cash" where StudentRollNo=2;
2 • select * from studentadmissionpaymentdetails;
```

AdmissionID	StudentRollNo	AmountPaid	AmountBalance	DateOfPayment	ModeOfPayment
1	1	20000	5000	2020-03-02	cash
2	2	21000	4000	2020-12-12	cash
3	3	19000	6000	2019-08-21	cash
4	4	20000	5000	2020-07-31	cheque
5	5	18000	7000	2018-05-09	credit
6	6	22000	3000	2019-11-28	cash
7	7	21000	4000	2018-09-11	credit
8	8	19000	6000	2020-12-11	cheque
9	9	24000	1000	2019-11-09	cash
10	10	18000	7000	2020-01-12	cheque
NULL	NULL	NULL	NULL	NULL	NULL

(iii) studentbasicinformation set StudentMobileNumber=22901256 where StudentRollNo=1;

```
1 • update studentbasicinformation set StudentMobileNumber=22901256 where StudentRollNo=1;
2 • select * from studentbasicinformation;
```

StudentRollNo	StudentName	StudentSurname	StudentAddress	StudentDOB	StudentGender	StudentMobileNumber
1	niyati	mehta	bangalore	1999-05-31	f	22901256
2	abc	xyz	mumbai	2000-03-21	m	128956342
3	john	doe	indore	1994-06-12	m	22890123
4	alan	walker	new york	1992-01-09	m	905612456
5	harry	potter	london	1997-07-31	m	90876234
6	chloe	NULL	california	1989-08-29	f	22456123
7	jasmine	mavani	rajkot	1998-11-22	f	233467856
8	adam	smith	goa	1997-09-21	m	21345265
9	nadia	tarannum	bhopal	1998-11-01	f	237512345
10	manisha	nahta	udaipur	1998-08-04	f	84253612
NULL	NULL	NULL	NULL	NULL	NULL	NULL

(iv) update subjectscholarshipinfo set ScholarshipAmount=6100 where ScholarshipID=3;

```
1 • update subjectscholarshipinfo set ScholarshipAmount=6100 where ScholarshipID=3;
2 • select * from subjectscholarshipinfo;
```

ScholarshipID	StudentRollNo	ScholarshipName	ScholarshipDescription	ScholarshipAmount	ScholarshipCategory	ScholarshipObtainedDate
1	1	Sc1	abc	5000	NULL	2021-07-12
2	3	Sc2	qwe	6700	NULL	2021-03-01
3	4	Sc3	rty	6100	NULL	2020-11-09
4	1	Sc2	abc	5000	NULL	2020-09-21
5	5	Sc2	qwe	6100	NULL	2021-01-02
6	7	Sc4	ghj	1000	NULL	2021-02-09
7	3	Sc6	qwe	3200	NULL	2020-05-12
8	2	Sc3	rty	7000	NULL	2021-05-31
9	5	Sc5	zxc	5500	NULL	2021-11-11
10	2	Sc1	abc	5100	NULL	2021-07-22
NULL	NULL	NULL	NULL	NULL	NULL	NULL

(v) update subjectscholarshipinfo set StudentRollNo=5 where ScholarshipID=3;

```
1 • update subjectscholarshipinfo set StudentRollNo=5 where ScholarshipID=3;
2 • select * from subjectscholarshipinfo;
```

ScholarshipID	StudentRollNo	ScholarshipName	ScholarshipDescription	ScholarshipAmount	ScholarshipCategory	ScholarshipObtainedDate
1	1	Sc1	abc	5000	NULL	2021-07-12
2	3	Sc2	qwe	6700	NULL	2021-03-01
3	5	Sc3	rty	6100	NULL	2020-11-09
4	1	Sc2	abc	5000	NULL	2020-09-21
5	5	Sc2	qwe	6100	NULL	2021-01-02
6	7	Sc4	ghj	1000	NULL	2021-02-09
7	3	Sc6	qwe	3200	NULL	2020-05-12
8	2	Sc3	rty	7000	NULL	2021-05-31
9	5	Sc5	zxc	5500	NULL	2021-11-11
10	2	Sc1	abc	5100	NULL	2021-07-22
NULL	NULL	NULL	NULL	NULL	NULL	NULL

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

```
1 • select sb.StudentRollNo, StudentName,StudentSurname,ss.ScholarshipName,ScholarshipAmount
2   from studentbasicinformation as sb,subjectscholarshipinfo as ss
3  where sb.StudentRollNo=ss.StudentRollNo AND ss.ScholarshipAmount>5000;
```

StudentRollNo	StudentName	StudentSurname	ScholarshipName	ScholarshipAmount
3	john	doe	Sc2	6700
5	harry	potter	Sc3	6100
5	harry	potter	Sc2	6100
2	abc	xyz	Sc3	7000
5	harry	potter	Sc5	5500
2	abc	xyz	Sc1	5100

8. Students who have Applied for scholarships but have not received it till “DATE”

```
select sb.StudentRollNo, sb.StudentName, sb.StudentMobileNumber, ss.ScholarshipAmount,  
ss.ScholarshipObtainedDate  
from studentbasicinformation as sb, subjectscholarshipinfo as ss  
where sb.StudentRollNo=ss.StudentRollNo AND ss.ScholarshipObtainedDate>2021-02-14  
order by sb.StudentRollNo;
```

```
1 • select StudentRollNo, StudentName, StudentSurname, StudentGender, StudentDOB, StudentAddress, StudentMobileNumber  
2 from studentbasicinformation where StudentRollNo not in  
3 (select distinct StudentRollNo from subjectscholarshipinfo)  
4
```

Result Grid							
Filter Rows: <input type="text"/>							
Edit: Export/Import: Wrap Cell Content:							
	StudentRollNo	StudentName	StudentSurname	StudentGender	StudentDOB	StudentAddress	StudentMobileNumber
▶	4	alan	walker	m	1992-01-09	new york	905612456
	6	chloe	NULL	f	1989-08-29	california	22456123
	8	adam	smith	m	1997-09-21	goa	21345265
	9	nadia	tarannum	f	1998-11-01	bhopal	237512345
	10	manisha	nahta	f	1998-08-04	udaipur	84253612
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

9. The percentage is taken as a calculation on the total marks which is assumed to be a summation of 5 subjects

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

```
call calc_percentage();  
select * from studentsubjectinformation;
```

```

1 • drop procedure if exists calc_percentage;
2   DELIMITER $$
3 • create procedure calc_percentage()
4   begin
5     update studentsubjectinformation set StudentMarksPercentage = (SubjectTotalMarks)/5;
6   end$$
7
8 • call calc_percentage();
9   select * from studentsubjectinformation;

```

Result Grid Filter Rows: Export: Wrap Cell Content:						
	StudentInfoID	StudentRollNo	SubjectOpted	SubjectTotalMarks	SubjectObtainedMarks	StudentMarksPercentage
▶	1	1	Maths	450	89	90
	2	2	Physics	390	75	78
	3	3	Computers	470	80	94
	4	4	Maths	470	90	94
	5	5	Biology	400	69	80
	6	6	Computers	480	95	96
	7	7	Physics	390	84	78
	8	8	Biology	460	80	92
	9	9	Maths	445	79	89
	10	10	Computers	465	88	93

10. The category is chosen based on the percentage obtained by the student.

```

drop procedure if exists calc_category;
DELIMITER $$
create procedure calc_category()
begin
update subjectscholarshipinfo ss inner join studentsubjectinformation si on
ss.StudentRollNo=si.StudentRollNo
set ss.ScholarshipCategory = case
when si.StudentMarksPercentage>=94 then 1
when si.StudentMarksPercentage>=90 then 2
when si.StudentMarksPercentage>=80 then 3
when si.StudentMarksPercentage>70 then 4
end;
END$$
call calc_category();

```

```

select StudentRollNo,StudentMarksPercentage,ScholarshipCategory from
studentsubjectinformation natural join subjectscholarshipinfo;

```



```
1 • select StudentRollNo,StudentMarksPercentage,ScholarshipCategory from studentsubjectinformation natural join subjectscholarshipinfo;
```

StudentRollNo	StudentMarksPercentage	ScholarshipCategory
1	90	2
1	90	2
2	78	4
2	78	4
3	94	1
3	94	1
5	80	3
5	80	3
5	80	3
7	78	4

11. create view Student_balance as

select

sb.StudentRollNo,sb.StudentName,sb.StudentSurname,sb.StudentGender,sb.StudentDOB,

sb.StudentAddress,sb.StudentMobileNumber,sa.AmountBalance

from studentbasicinformation as sb natural join studentadmissionpaymentdetails as sa;

SELECT * FROM student.student_balance;

```
1 • SELECT * FROM student.student_balance;
```

StudentRollNo	StudentName	StudentSurname	StudentGender	StudentDOB	StudentAddress	StudentMobileNumber	AmountBalance
1	niyati	mehta	f	1999-05-31	bangalore	22901256	5000
2	abc	xyz	m	2000-03-21	mumbai	128956342	4000
3	john	doe	m	1994-06-12	indore	22890123	6000
4	alan	walker	m	1992-01-09	new york	905612456	5000
5	harry	potter	m	1997-07-31	london	90876234	7000
6	chloe	NULL	f	1989-08-29	california	22456123	3000
7	jasmine	mavani	f	1998-11-22	rajkot	233467856	4000
8	adam	smith	m	1997-09-21	goa	21345265	6000
9	nadia	tarannum	f	1998-11-01	bhopal	237512345	1000
10	manisha	nahta	f	1998-08-04	udaipur	84253612	7000

12. SUB-QUERY

select

StudentRollNo,StudentName,StudentSurname,StudentGender,StudentDOB,StudentAddress,

StudentMobileNumber from studentbasicinformation where StudentRollNo not in

(select distinct StudentRollNo from subjectscholarshipinfo)

OR

JOIN

```
select distinct
sb.StudentRollNo,StudentName,StudentSurname,StudentGender,StudentDOB,StudentAddress,StudentMobileNumber
from studentbasicinformation as sb
left join subjectscholarshipinfo as ss
on sb.StudentRollNo=ss.StudentRollNo
where ss.StudentRollNo is NULL;
```

```
1 • select StudentRollNo,StudentName,StudentSurname,StudentGender,StudentDOB,StudentAddress,StudentMobileNumber
2   from studentbasicinformation where StudentRollNo not in
3   (select distinct StudentRollNo from subjectscholarshipinfo)
4
```



Result Grid							
Filter Rows:							
Edit: Export/Import: Wrap Cell Content: IA							
	StudentRollNo	StudentName	StudentSurname	StudentGender	StudentDOB	StudentAddress	StudentMobileNumber
▶	4	alan	walker	m	1992-01-09	new york	905612456
	6	chloe	NULL	f	1989-08-29	california	22456123
	8	adam	smith	m	1997-09-21	goa	21345265
	9	nadia	tarannum	f	1998-11-01	bhopal	237512345
	10	manisha	nahta	f	1998-08-04	udaipur	84253612
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

13. DELIMITER \$\$

```
create procedure student_balance( IN rollno int)
begin
select
sb.StudentRollNo,sb.StudentName,sb.StudentSurname,sb.StudentMobileNumber,sb.StudentAddress,sa.AmountBalance from studentbasicinformation as sb,
studentadmissionpaymentdetails as sa where sb.StudentRollNo=rollno and
sb.StudentRollNo=sa.StudentRollNo;
end $$
```




```
call student_balance(3);
```

```
1 call student_balance(3);
```

<						
Result Grid						
Filter Rows: <input type="text"/>						
Export: 						
Wrap Cell Content: 						
	StudentRollNo	StudentName	StudentSurname	StudentMobileNumber	StudentAddress	AmountBalance
▶	3	john	doe	22890123	indore	6000

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

```
1 select * from (  
2   select sb.*,ss.StudentMArksPercentage  
3   from studentbasicinformation as sb, studentsubjectinformation as ss  
4   where sb.StudentRollNo=ss.StudentRollNo order by ss.StudentMArksPercentage desc)  
5   as top_five_students limit 5;
```

<								
Result Grid								
Filter Rows: <input type="text"/>								
Export: 								
Wrap Cell Content: 								
Fetch rows: 								
	StudentRollNo	StudentName	StudentSurname	StudentAddress	StudentDOB	StudentGender	StudentMobileNumber	StudentMArksPercentage
▶	6	chloe	NULL	california	1989-08-29	f	22456123	96
	3	john	doe	indore	1994-06-12	m	22890123	94
	4	alan	walker	new york	1992-01-09	m	905612456	94
	10	manisha	nahta	udaipur	1998-08-04	f	84253612	93
	8	adam	smith	goa	1997-09-21	m	21345265	92

15. Three Types of 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.

```
1 select * from studentbasicinformation inner join studentadmissionpaymentdetails;
```

Result Grid											
Filter Rows:											
Exports: Wrap Cell Content:											
	StudentRollNo	StudentName	StudentSurname	StudentAddress	StudentDOB	StudentGender	StudentMobileNumber	AdmissionID	StudentRollNo	AmountPaid	AmountBalance
1	niyati	mehta	bangalore	1999-05-31	f	22901256	1	1	1	20000	5000
2	abc	xyz	mumbai	2000-03-21	m	128956342	1	1	1	20000	5000
3	john	doe	indore	1994-06-12	m	22890123	1	1	1	20000	5000
4	alan	walker	new york	1992-01-09	m	905612456	1	1	1	20000	5000
5	harry	potter	london	1997-07-31	m	90876234	1	1	1	20000	5000
6	chloe	NULL	california	1989-08-29	f	22456123	1	1	1	20000	5000
7	jasmine	mavani	rajkot	1998-11-22	f	233467856	1	1	1	20000	5000
8	adam	smith	goa	1997-09-21	m	21345265	1	1	1	20000	5000
9	nadia	tarannum	bhopal	1998-11-01	f	237512345	1	1	1	20000	5000
10	manisha	nahta	udaipur	1998-08-04	f	84253612	1	1	1	20000	5000
1	niyati	mehta	bangalore	1999-05-31	f	22901256	2	2	2	21000	4000
2	abc	xyz	mumbai	2000-03-21	m	128956342	2	2	2	21000	4000
3	john	doe	indore	1994-06-12	m	22890123	2	2	2	21000	4000
4	alan	walker	new york	1992-01-09	m	905612456	2	2	2	21000	4000
5	harry	potter	london	1997-07-31	m	90876234	2	2	2	21000	4000
6	chloe	NULL	california	1989-08-29	f	22456123	2	2	2	21000	4000
7	jasmine	mavani	rajkot	1998-11-22	f	233467856	2	2	2	21000	4000
8	adam	smith	goa	1997-09-21	m	21345265	2	2	2	21000	4000

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.

```
1 select * from studentbasicinformation as sb left join subjectscholarshipinfo as ss on sb.StudentRollNo=ss.StudentRollNo ;
```

Result Grid									
Filter Rows:									
Exports: Wrap Cell Content:									
	StudentRollNo	StudentName	StudentSurname	StudentAddress	StudentDOB	StudentGender	StudentMobileNumber	ScholarshipID	ScholarshipName
1	niyati	mehta	bangalore	1999-05-31	f	22901256	1	1	Sc1
2	niyati	mehta	bangalore	1999-05-31	f	22901256	4	1	Sc2
3	abc	xyz	mumbai	2000-03-21	m	128956342	8	2	Sc3
4	abc	xyz	mumbai	2000-03-21	m	128956342	10	2	Sc1
5	john	doe	indore	1994-06-12	m	22890123	2	3	Sc2
6	john	doe	indore	1994-06-12	m	22890123	7	3	Sc6
7	alan	walker	new york	1992-01-09	m	905612456	NULL	NULL	NULL
8	harry	potter	london	1997-07-31	m	90876234	3	5	Sc3
9	harry	potter	london	1997-07-31	m	90876234	5	5	Sc2
10	harry	potter	london	1997-07-31	m	90876234	9	5	Sc5
11	chloe	NULL	california	1989-08-29	f	22456123	NULL	NULL	NULL
12	jasmine	mavani	rajkot	1998-11-22	f	233467856	6	7	Sc4
13	adam	smith	goa	1997-09-21	m	21345265	NULL	NULL	NULL
14	nadia	tarannum	bhopal	1998-11-01	f	237512345	NULL	NULL	NULL
15	manisha	nahta	udaipur	1998-08-04	f	84253612	NULL	NULL	NULL

RIGHT OUTER JOIN

This join can be used with scholarship info and admission details to compare the scholarships given and the Fees paid so that scholarship amount can be redeemed in the fees.

```
1 select * from subjectscholarshipinfo as sb right join studentadmissionpaymentdetails as ss on sb.StudentRollNo=ss.StudentRollNo ;
```

Result Grid Filter Rows: Export: Wrap Cell Content:									
ScholarshipID	StudentRollNo	ScholarshipName	ScholarshipDescription	ScholarshipAmount	ScholarshipCategory	ScholarshipObtainedDate	AdmissionID	StudentRollNo	AmountPaid
1	1	Sc1	abc	5000	2	2021-07-12	1	1	20000
4	1	Sc2	abc	5000	2	2020-09-21	1	1	20000
8	2	Sc3	rty	7000	4	2021-05-31	2	2	21000
10	2	Sc1	abc	5100	4	2021-07-22	2	2	21000
2	3	Sc2	qwe	6700	1	2021-03-01	3	3	19000
7	3	Sc6	qwe	3200	1	2020-05-12	3	3	19000
NULL	NULL	NULL	NULL	NULL	NULL	NULL	4	4	20000
3	5	Sc3	rty	6100	3	2020-11-09	5	5	18000
5	5	Sc2	qwe	6100	3	2021-01-02	5	5	18000
9	5	Sc5	zxc	5500	3	2021-11-11	5	5	18000
NULL	NULL	NULL	NULL	NULL	NULL	NULL	6	6	22000
6	7	Sc4	ghj	1000	4	2021-02-09	7	7	21000
NULL	NULL	NULL	NULL	NULL	NULL	NULL	8	8	19000
NULL	NULL	NULL	NULL	NULL	NULL	NULL	9	9	24000
NULL	NULL	NULL	NULL	NULL	NULL	NULL	10	10	18000

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;

```
1 • select ScholarshipCategory,count(*) as number_of_Scholarships
2   from subjectscholarshipinfo
3  group by ScholarshipCategory
4  order by ScholarshipCategory;
5
```

<		
Result Grid		
Filter Rows: <input type="text"/>		
Export:  Wrap Cell Content: 		
	ScholarshipCategory	number_of_Scholarships
▶	1	2
	2	2
	3	3
	4	3

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 bb;

```
1 • select all ScholarshipCategory, max(number_of_Scholarships) as max_number_scholarship from(
2   select ScholarshipCategory,count(*) as number_of_Scholarships
3   from subjectscholarshipinfo
4  group by ScholarshipCategory
5  order by ScholarshipCategory)
6  as bb;
```

<		
Result Grid		
Filter Rows: <input type="text"/>		
Export:  Wrap Cell Content: 		
	ScholarshipCategory	max_number_scholarship
▶	1	3

19. select sb.*,s.StudentMarksPercentage from studentbasicinformation as sb natural join studentsubjectinformation as s where StudentRollNo in (select sch.StudentRollNo from subjectscholarshipinfo as sch where sch.ScholarshipAmount= (select max(ScholarshipAmount) from subjectscholarshipinfo));

```

1 • select sb.*,s.StudentMarksPercentage from studentbasicinformation as sb
2   natural join studentsubjectinformation as s where StudentRollNo in
3   (select sch.StudentRollNo from subjectscholarshipinfo as sch where sch.ScholarshipAmount=
4   (
5     select max(ScholarshipAmount) from subjectscholarshipinfo
6   )) ;

```

StudentRollNo	StudentName	StudentSurname	StudentAddress	StudentDOB	StudentGender	StudentMobileNumber	StudentMarksPercentage
2	abc	xyz	mumbai	2000-03-21	m	128956342	78

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().