

## SQL Assignment

Ashit Mundra

1)

Create database student;  
Use student;

2)

```
create table StudentBasicInformation(  
    RollNumber varchar(20) primary key,  
    StudentName varchar(25),  
    StudentSurname varchar(25),  
    StudentAddress varchar(150),  
    Gender varchar(10),  
    Email varchar(25),  
    Mobile numeric(10,0)  
);
```

Table(1)

```
create table StudentAdmissionPaymentDetails(  
    StudentRollNo varchar(20),  
    AmountPaid int,  
    AmountBalance int,  
    Semester int,  
    StudentBank varchar(20),  
    PaymentDate date,  
    MethodOfPayment varchar(15),  
    primary key (StudentRollNo,Semester)  
);
```

Table(2)

```
create table StudentSubjectInformation(  
    StudentRollNo varchar(20),  
    SubjectOpted varchar(20),  
    SubjectTotalMarks int,  
    SubjectObtainedMarks int,  
    StudentMarksPercentage float,  
    Division varchar(5),  
    primary key (StudentRollNo,SubjectOpted));
```

### Table(3)

```
create table SubjectScholarshipInformation(  
    StudentRollNo varchar(20),  
    ScholarshipName varchar(25),  
    ScholarshipDescription varchar(80),  
    ScholarshipAmount int,  
    ScholarshipCategory varchar(20),  
    StudentBank varchar(20),  
    ReceiptID varchar(25) primary key);
```

### Table(4)

3) and 4)

### For Table(1)

Insert into StudentBasicInformation values

```
("1001","Ashit","Mundra","Rajasthan","Male", "Nathdwara", 8005518456),  
("1002","Axar","Patel","Rajasthan","Male", "Jaipur", 80546),  
("1003","Bhavya","Agrawal","Gujrat","Female", "Ahemdabad", 8054611),  
("1004","Bob","Sharma","Karnataka","Male", "Bangalore", 8011546),  
("1005","Carry","Minati","Maharashtra","Male", "Mumbai", 80541096),  
("1006","Chaya","Sharma","Maharashtra","Female", "Mumbai", 86660546),  
("1007","Daisy","Shah","UP","Female", "banaras", 80500046),  
("1008","Dhairya","Verma","Rajasthan","Male", "Udaipur", 81110546),
```

("1009","Hitesh","Jain","Gujrat","Male", "Ahemdabad", 800999546),  
 ("1010","Hitesh","Patel","UP","Male", "agra", 80556646),  
 ("1011","Jayant","Dixit","Rajasthan","Male", "Jaipur", 809797546);

	StudentRollNo	StudentName	StudentSurname	StudentAddress	Gender	City	Mobile
1	1001	Ashit	Mundra	Rajasthan	Male	Nathdwara	8005518456
2	1002	Axar	Patel	Rajasthan	Male	Jaipur	80546
3	1003	Bhavya	Agrawal	Gujrat	Female	Ahemdabad	8054611
4	1004	Bob	Sharma	Karnataka	Male	Bangalore	8011546
5	1005	Carry	Minati	Maharashtra	Male	Mumbai	80541096
6	1006	Chaya	Sharma	Maharashtra	Female	Mumbai	86660546
7	1007	Daisy	Shah	UP	Female	banaras	80500046
8	1008	Dhairya	Verma	Rajasthan	Male	Udaipur	81110546
9	1009	Hitesh	Jain	Gujrat	Male	Ahemdabad	800999546
10	1010	Hitesh	Patel	UP	Male	agra	80556646
11	1011	Jayant	Dixit	Rajasthan	Male	Jaipur	809797546

### For Table(2)

Insert into StudentAdmissionPaymentDetails values

("1001", 15000, 2000, 3, "Axis", '2012-01-01', "UPI"),  
 ("1005", 18000, 7000, 6, "ICICI", '2012-05-02', "Draft"),  
 ("1001", 25000, 5000, 2, "Axis", '2011-04-06', "UPI"),  
 ("1008", 17000, 0, 3, "BOB", '2012-01-06', "Cheque"),  
 ("1004", 15000, 2000, 3, "ICICI", '2015-10-01', "Draft"),  
 ("1004", 35000, 2000, 5, "BOB", '2016-01-01', "UPI"),  
 ("1010", 25000, 3000, 6, "Axis", '2014-04-03', "Net"),  
 ("1009", 15000, 2000, 3, "Axis", '2012-01-01', "UPI"),  
 ("1007", 25000, 2000, 3, "Axis", '2012-01-01', "UPI"),  
 ("1007", 15000, 2000, 1, "Axis", '2012-01-01', "UPI");

	StudentRollNo	AmountPaid	AmountBalance	Semester	StudentBank	PaymentDate	MethodOfPayment
1	1001	25000	5000	2	Axis	06.04.2011 00:00:00	UPI
2	1001	15000	2000	3	Axis	01.01.2012 00:00:00	UPI
3	1004	15000	2000	3	ICICI	01.10.2015 00:00:00	Draft
4	1004	35000	2000	5	BOB	01.01.2016 00:00:00	UPI
5	1005	18000	7000	6	ICICI	02.05.2012 00:00:00	Draft
6	1007	15000	2000	1	Axis	01.01.2012 00:00:00	UPI
7	1007	25000	2000	3	Axis	01.01.2012 00:00:00	UPI
8	1008	17000	0	3	BOB	06.01.2012 00:00:00	Cheque
9	1009	15000	2000	3	Axis	01.01.2012 00:00:00	UPI
10	1010	25000	3000	6	Axis	03.04.2014 00:00:00	Net

### For Table(3)

Insert into StudentSubjectInformation values

```
("1001", "Physics", 100, 64, 72, "B-"),
("1001", "Maths", 100, 78, 72, "B-"),
("1003", "Physics", 100, 84, 88, "B+"),
("1004", "Physics", 100, 94, 96, "A"),
("1005", "Maths", 100, 64, 72, "B-"),
("1007", "Chemistry", 100, 94, 91, "A-"),
("1007", "Maths", 100, 90, 91, "B-"),
("1002", "Physics", 100, 64, 72, "B-"),
("1009", "Physics", 100, 64, 72, "B-"),
("1010", "Physics", 100, 64, 72, "B-");
```

	StudentRollNo	SubjectOpted	SubjectTotalMarks	SubjectObtainedMarks	StudentMarksPercentage	Division
1	1001	Maths	100	78	72	B-
2	1001	Physics	100	64	72	B-
3	1002	Physics	100	64	72	B-
4	1003	Physics	100	84	88	B+
5	1004	Physics	100	94	96	A
6	1005	Maths	100	64	72	B-
7	1007	Chemistry	100	94	91	A-
8	1007	Maths	100	90	91	B-
9	1009	Physics	100	64	72	B-
10	1010	Physics	100	64	72	B-

### For Table(4)

Insert into SubjectScholarshipInformation values

```
("1004", "Merit", NULL, 1000, "Marks", "BOB", "101"),
("1004", "Merit", NULL, 1000, "Marks", "BOB", "102"),
("1004", "Merit", NULL, 2000, "Marks", "BOB", "103"),
("1001", "Football", NULL, 1000, "Sports", "Axis", "104"),
("1007", "Merit", NULL, 1000, "Marks", "Axis", "105"),
("1007", "Merit", NULL, 1000, "Marks", "Axis", "106"),
("1005", "Volleyball", NULL, 1000, "Sports", "BOB", "107"),
("1005", "Volleyball", NULL, 1000, "Sports", "BOB", "108"),
("1003", "Merit", NULL, 1000, "Marks", "BOB", "109"),
("1003", "Cricket", NULL, 1000, "Sports", "BOB", "110");
```

	StudentRollNo	ScholarshipName	ScholarshipDescription	ScholarshipAmount	ScholarshipCategory	StudentBank	ReceiptID
1	1004	Merit	NULL	1000	Marks	B0B	101
2	1004	Merit	NULL	1000	Marks	B0B	102
3	1004	Merit	NULL	2000	Marks	B0B	103
4	1001	Football	NULL	1000	Sports	Axis	104
5	1007	Merit	NULL	1000	Marks	Axis	105
6	1007	Merit	NULL	1000	Marks	Axis	106
7	1005	Volleyball	NULL	1000	Sports	B0B	107
8	1005	Volleyball	NULL	1000	Sports	B0B	108
9	1003	Merit	NULL	1000	Marks	B0B	109
10	1003	Cricket	NULL	1000	Sports	B0B	110

5)

a)

update StudentBasicInformation set City="Bombay" where  
City="Mumbai"; # 2 updations

update StudentBasicInformation set StudentName="Kishan" where  
StudentSurname="Sharma"; # 2 updations

update StudentBasicInformation set StudentAddress="Uttar Pradesh"  
where StudentAddress="UP"; # 2 updations

b)

update StudentAdmissionPaymentDetails set AmountBalance=0 where  
StudentRollNo="1001"; # 2 updations

update StudentAdmissionPaymentDetails set Semester=7 where  
StudentRollNo="1010"; # 1 updations

update StudentAdmissionPaymentDetails set AmountBalance=5000 where  
StudentBank="ICICI"; # 2 updations

c)

update StudentSubjectInformation set StudentMarksPercentage=NULL  
where Division="B-"; # 7 updations

d)

update SubjectScholarshipInformation set ScholarshipAmount=0 where  
ScholarshipName="Volleyball"; # 2 updations

6)

	StudentRollNo	StudentName	StudentSurname	StudentAddress	Gender	City	Mobile
1	1001	Ashit	Mundra	Rajasthan	Male	Nathdwara	8005518456
2	1002	Axar	Patel	Rajasthan	Male	Jaipur	80546
3	1003	Bhavya	Agrawal	Gujrat	Female	Ahemdabad	8054611
4	1004	Kishan	Sharma	Karnataka	Male	Bangalore	8011546
5	1005	Carry	Minati	Maharashtra	Male	Bombay	80541096
6	1006	Kishan	Sharma	Maharashtra	Female	Bombay	86660546
7	1007	Daisy	Shah	Uttar Pradesh	Female	banaras	80500046
8	1008	Dhairya	Verma	Rajasthan	Male	Udaipur	81110546
9	1009	Hitesh	Jain	Gujrat	Male	Ahemdabad	800999546
10	1010	Hitesh	Patel	Uttar Pradesh	Male	agra	80556646
11	1011	Jayant	Dixit	Rajasthan	Male	Jaipur	809797546

	StudentRollNo	AmountPaid	AmountBalance	Semester	StudentBank	PaymentDate	MethodOfPayment
1	1001	25000	0	2	Axis	06.04.2011 00:00:00	UPI
2	1001	15000	0	3	Axis	01.01.2012 00:00:00	UPI
3	1004	15000	5000	3	ICICI	01.10.2015 00:00:00	Draft
4	1004	35000	2000	5	BOB	01.01.2016 00:00:00	UPI
5	1005	18000	5000	6	ICICI	02.05.2012 00:00:00	Draft
6	1007	15000	2000	1	Axis	01.01.2012 00:00:00	UPI
7	1007	25000	2000	3	Axis	01.01.2012 00:00:00	UPI
8	1008	17000	0	3	BOB	06.01.2012 00:00:00	Cheque
9	1009	15000	2000	3	Axis	01.01.2012 00:00:00	UPI
10	1010	25000	3000	7	Axis	03.04.2014 00:00:00	Net

	StudentRollNo	SubjectOpted	SubjectTotalMarks	SubjectObtainedMarks	StudentMarksPercentage	Division
1	1001	Maths	100	78	NULL	B-
2	1001	Physics	100	64	NULL	B-
3	1002	Physics	100	64	NULL	B-
4	1003	Physics	100	84	88	B+
5	1004	Physics	100	94	96	A
6	1005	Maths	100	64	NULL	B-
7	1007	Chemistry	100	94	91	A-
8	1007	Maths	100	90	NULL	B-
9	1009	Physics	100	64	NULL	B-
10	1010	Physics	100	64	NULL	B-

	StudentRollNo	ScholarshipName	ScholarshipDescription	ScholarshipAmount	ScholarshipCategory	StudentBank	ReceiptID
1	1004	Merit	NULL	1000	Marks	BOB	101
2	1004	Merit	NULL	1000	Marks	BOB	102
3	1004	Merit	NULL	2000	Marks	BOB	103
4	1001	Football	NULL	1000	Sports	Axis	104
5	1007	Merit	NULL	1000	Marks	Axis	105
6	1007	Merit	NULL	1000	Marks	Axis	106
7	1005	Volleyball	NULL	0	Sports	BOB	107
8	1005	Volleyball	NULL	0	Sports	BOB	108
9	1003	Merit	NULL	1000	Marks	BOB	109
10	1003	Cricket	NULL	1000	Sports	BOB	110

7)

```
update SubjectScholarshipInformation set ScholarshipAmount=4000
where ScholarshipAmount=2000; # For high scholarship
```

```
Create VIEW temp_scholarship AS SELECT StudentRollNo,
ScholarshipAmount from SubjectScholarshipInformation;
```

```
Create View total_scholarship as select StudentRollNo,
SUM(ScholarshipAmount) as total from temp_scholarship group by
StudentRollNo;
```

	StudentRollNo	total
1	1001	1000
2	1003	2000
3	1004	6000
4	1005	0
5	1007	2000

```
create view ttemp as select StudentRollNo from total_scholarship where
total>5000;
```

```
select StudentName from ttemp inner join StudentBasicInformation on
ttemp.StudentRollNo=StudentBasicInformation.StudentRollNo;
```

	StudentName
1	Kishan

8)

```
select * from StudentBasicInformation A inner join  
SubjectScholarshipInformation B on A.StudentRollNo = B.StudentRollNo  
where B.ScholarshipAmount=0;
```

	StudentRollNo	StudentName	StudentSurname
1	1005	Carry	Minati
2	1005	Carry	Minati

9)

```
DELIMITER $$  
CREATE PROCEDURE percentF()  
BEGIN  
update StudentSubjectInformation set StudentMarksPercentage =  
(SubjectObtainedMarks/SubjectTotalMarks)*100;  
end $$  
delimiter ;  
call percentF();  
select * from StudentSubjectInformation;
```

StudentRollNo	SubjectOpted	SubjectTotalMarks	SubjectObtainedMarks	StudentMarksPercentage	Division
1001	Maths	100	78	78	B-
1001	Physics	100	64	64	B-
1002	Physics	100	64	64	B-
1003	Physics	100	84	84	B+
1004	Physics	100	94	94	A
1005	Maths	100	64	64	B-
1007	Chemistry	100	94	94	A-
1007	Maths	100	90	90	B-
1009	Physics	100	64	64	B-
1010	Physics	100	64	64	B-



10)

```
Create view scholarTemp as Select A.StudentRollNo,  
StudentMarksPercentage, ScholarshipAmount, ScholarshipCategory from  
StudentSubjectInformation A inner join SubjectScholarshipInformation B  
on A.StudentRollNo = B.StudentRollNo;  
DELIMITER $$  
CREATE PROCEDURE scholarshipTT()  
BEGIN  
update scholarTemp set ScholarshipAmount=2000 where  
StudentMarksPercentage>80;  
update scholarTemp set ScholarshipAmount = 4000 where  
StudentMarksPercentage>=91;  
end $$  
delimiter ;  
call scholarshipTT();
```

StudentRollNo	StudentMarksPercentage	ScholarshipAmount	ScholarshipCategory
1004	94	4000	Marks
1004	94	4000	Marks
1004	94	4000	Marks
1001	78	1000	Sports
1001	64	1000	Sports
1007	94	4000	Marks
1007	90	4000	Marks
1007	94	4000	Marks
1007	90	4000	Marks
1005	64	0	Sports
1005	64	0	Sports
1003	84	2000	Marks
1003	84	2000	Sports

```
update SubjectScholarshipInformation A, scholarTemp B set  
A.ScholarshipAmount=B.ScholarshipAmount where  
A.StudentRollNo=B.StudentRollNo;
```

```
mysql> select * from SubjectScholarshipInformation;
```

StudentRollNo	ScholarshipName	ScholarshipDescription	ScholarshipAmount	ScholarshipCategory	StudentBank	ReceiptID
1004	Merit	NULL	4000	Marks	BOB	101
1004	Merit	NULL	4000	Marks	BOB	102
1004	Merit	NULL	4000	Marks	BOB	103
1001	Football	NULL	1000	Sports	Axis	104
1007	Merit	NULL	4000	Marks	Axis	105
1007	Merit	NULL	4000	Marks	Axis	106
1005	Volleyball	NULL	0	Sports	BOB	107
1005	Volleyball	NULL	0	Sports	BOB	108
1003	Merit	NULL	2000	Marks	BOB	109
1003	Cricket	NULL	2000	Sports	BOB	110

11)

create view balance as Select A.\*, B.AmountBalance from  
StudentBasicInformation A inner join StudentAdmissionPaymentDetails B  
where A.StudentRollNo=B.StudentRollNo;

StudentRollNo	StudentName	StudentSurname	StudentAddress	Gender	City	Mobile	AmountBalance
1001	Ashit	Mundra	Rajasthan	Male	Nathdwara	8005518456	0
1001	Ashit	Mundra	Rajasthan	Male	Nathdwara	8005518456	0
1004	Kishan	Sharma	Karnataka	Male	Bangalore	8011546	5000
1004	Kishan	Sharma	Karnataka	Male	Bangalore	8011546	2000
1005	Carry	Minati	Maharashtra	Male	Bombay	80541096	5000
1007	Daisy	Shah	Uttar Pradesh	Female	banaras	80500046	2000
1007	Daisy	Shah	Uttar Pradesh	Female	banaras	80500046	2000
1008	Dhairya	Verma	Rajasthan	Male	Udaipur	81110546	0
1009	Hitesh	Jain	Gujrat	Male	Ahemdabad	800999546	2000
1010	Hitesh	Patel	Uttar Pradesh	Male	agra	80556646	3000

12)

Select A.\* from StudentBasicInformation A left join  
SubjectScholarshipInformation B on A.StudentRollNo= B.StudentRollNo  
where(ScholarshipAmount=0 or ScholarshipAmount is NULL);

StudentRollNo	StudentName	StudentSurname	StudentAddress	Gender	City	Mobile
1005	Carry	Minati	Maharashtra	Male	Bombay	80541096
1005	Carry	Minati	Maharashtra	Male	Bombay	80541096
1002	Axar	Patel	Rajasthan	Male	Jaipur	80546
1006	Kishan	Sharma	Maharashtra	Female	Bombay	86660546
1008	Dhairya	Verma	Rajasthan	Male	Udaipur	81110546
1009	Hitesh	Jain	Gujrat	Male	Ahemdabad	800999546
1010	Hitesh	Patel	Uttar Pradesh	Male	agra	80556646
1011	Jayant	Dixit	Rajasthan	Male	Jaipur	809797546

13)

delimiter \$\$

create procedure checkBalanceS(rollno varchar(20))

BEGIN

select AmountBalance from StudentAdmissionPaymentDetails where  
StudentRollNo = rollno;

END\$\$

delimiter ;

call checkBalanceS("1005");

```
mysql> call checkBalanceS("1005");
+-----+
| AmountBalance |
+-----+
|          5000 |
+-----+
```

14)

Select \* from StudentSubjectInformation order by  
StudentMarksPercentage DESC

LIMIT 5;

StudentRollNo	SubjectOpted	SubjectTotalMarks	SubjectObtainedMarks	StudentMarksPercentage	Division
1004	Physics	100	94	94	A
1007	Chemistry	100	94	94	A-
1007	Maths	100	90	90	B-
1003	Physics	100	84	84	B+
1001	Maths	100	78	78	B-

15)

a) Inner Join

Select A.StudentRollNo, A. StudentName, B.AmountBalance from  
StudentBasicInformation A inner join StudentAdmissionPaymentDetails B  
where A. StudentRollNo=B. StudentRollNo;

This Will tell us name of students too with the amount balance left along with roll number.

#### b) Left join

```
Select A.* from StudentBasicInformation A left join  
SubjectScholarshipInformation B on A.StudentRollNo= B.StudentRollNo  
where(ScholarshipAmount=0 or ScholarshipAmount is NULL);
```

like in question 12 we can use this to merge the students who applied for scholarship but didnt got it and students who didnt applied for scholarship

#### c)Right Join

It is same as left join in above query we can swap A and B with right outer join to get same result.

16)

Delete- Will Delete rows from the table but memory will still not be freed.

Truncate- Delete all rows from table with memory also getting freed

Drop- Will delete complete table unlike truncate it will delete schema as well.

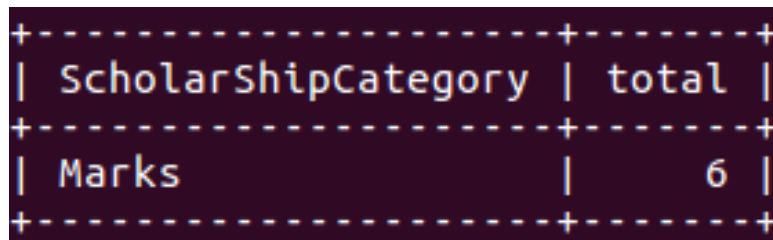
17)

```
select ScholarShipCategory,count(*) as total from  
SubjectScholarshipInformation group by ScholarshipCategory;
```

ScholarShipCategory	total
Marks	6
Sports	4

18)

```
select * from category order by total DESC  
LIMIT 1;
```



```
+-----+-----+  
| ScholarshipCategory | total |  
+-----+-----+  
| Marks              | 6    |  
+-----+-----+
```

19)

```
create view gtemp as select A.*, B.StudentMarksPercentage,  
B.SubjectOpted from StudentBasicInformation A inner join  
StudentSubjectInformation B where A.StudentRollNo=B.StudentRollNo  
order by StudentMarksPercentage DESC  
LIMIT 1;
```

```
select A.*, B.ScholarshipAmount from gtemp A inner join  
SubjectScholarshipInformation B where  
A.StudentRollNo=B.StudentRollNo;
```

StudentRollNo	StudentName	StudentSurname	StudentAddress	Gender	City	Mobile	StudentMarksPercentage	SubjectOpted	ScholarshipAmount
1004	Kishan	Sharma	Karnataka	Male	Bangalore	8011546	94	Physics	4000
1004	Kishan	Sharma	Karnataka	Male	Bangalore	8011546	94	Physics	4000
1004	Kishan	Sharma	Karnataka	Male	Bangalore	8011546	94	Physics	4000

20)

Triggers- Execute on occurrence of some event

Stored Procedure- Execute when being called also returning value is not compulsory.

View- Virtual Table with selected attributes can be used to copy data so don't need to alter main table for any data loss.

Functions- Same as Procedure but it must return some value.