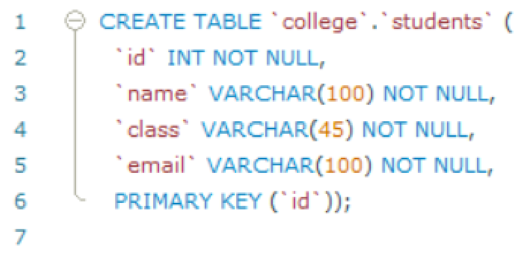
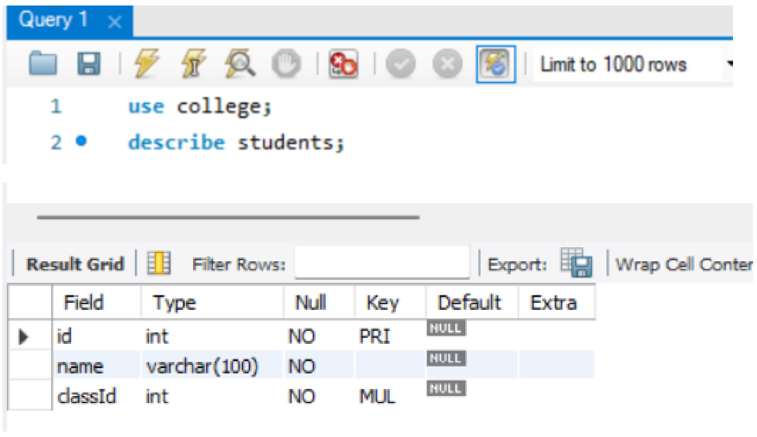
NOTE: I have added the screenshots of the SQL queries with the respective heading.

1. Create table



```
CREATE TABLE `college`.`students` (
  `id` INT NOT NULL,
  `name` VARCHAR(100) NOT NULL,
  `class` VARCHAR(45) NOT NULL,
  `email` VARCHAR(100) NOT NULL,
  PRIMARY KEY (`id`));
```

2. Describe table



Query 1

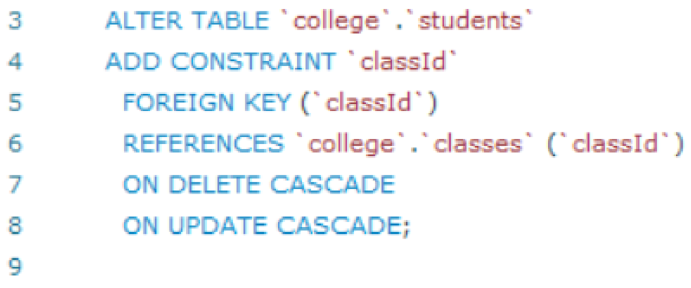
Limit to 1000 rows

```
use college;
describe students;
```

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

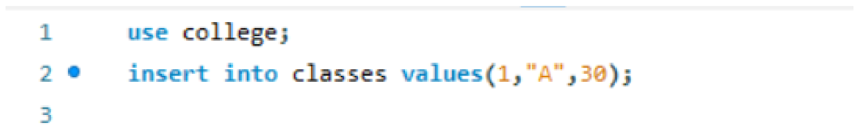
Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	
name	varchar(100)	NO		NULL	
classId	int	NO	MUL	NULL	

3. Alter table



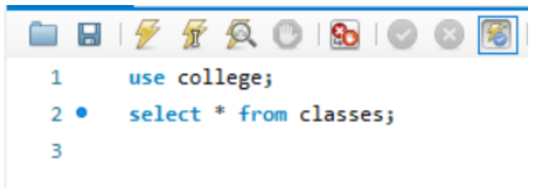
```
ALTER TABLE `college`.`students`
ADD CONSTRAINT `classId`
  FOREIGN KEY (`classId`)
  REFERENCES `college`.`classes` (`classId`)
  ON DELETE CASCADE
  ON UPDATE CASCADE;
```

4. Insert into Table

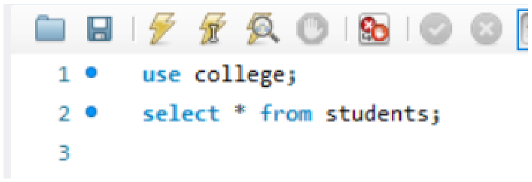


```
use college;
insert into classes values(1,"A",30);
```

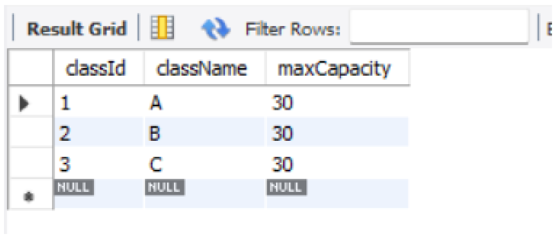
5. Select statement



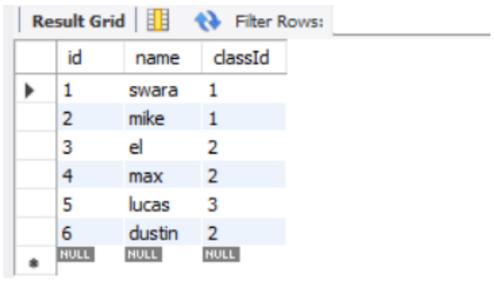
```
use college;
select * from classes;
```



```
use college;
select * from students;
```

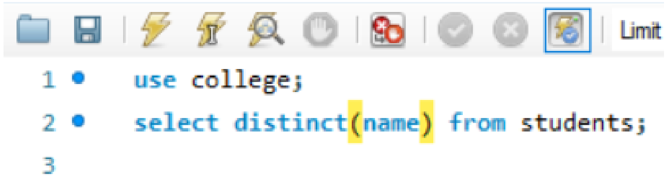


classId	className	maxCapacity
1	A	30
2	B	30
3	C	30
NULL	NULL	NULL

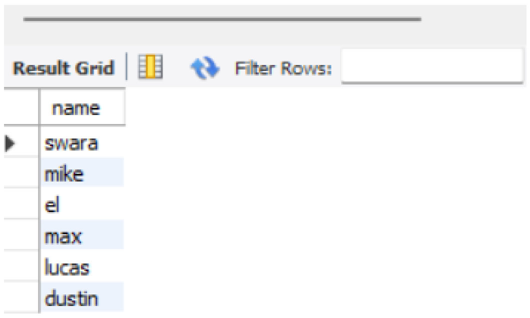


id	name	classId
1	swara	1
2	mike	1
3	el	2
4	max	2
5	lucas	3
6	dustin	2
NULL	NULL	NULL

6. Distinct

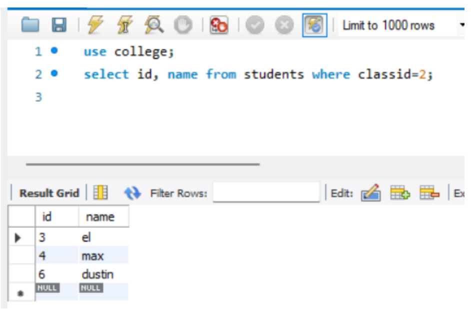


```
use college;
select distinct(name) from students;
```



name
swara
mike
el
max
lucas
dustin

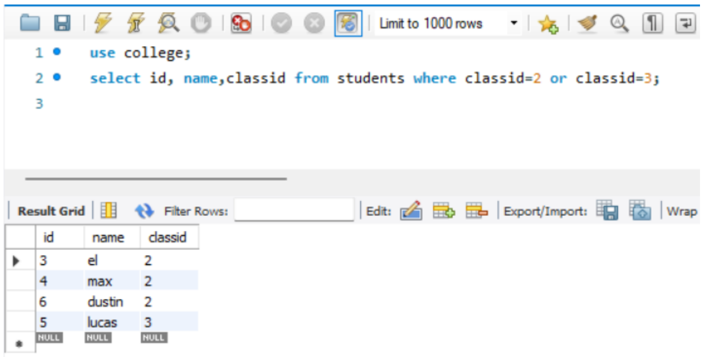
7. Where



```
use college;
select id, name from students where classid=2;
```

id	name
3	el
4	max
6	dustin
NULL	NULL

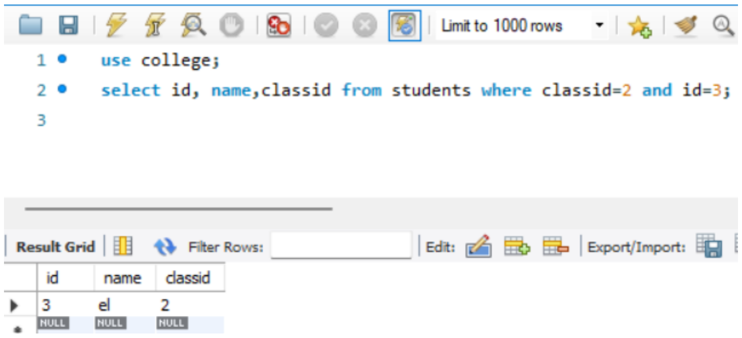
8. Or



```
use college;
select id, name,classid from students where classid=2 or classid=3;
```

```
id   name    classid
3    el      2
4    max     2
6    dustin  2
5    lucas   3
NULL NULL    NULL
```

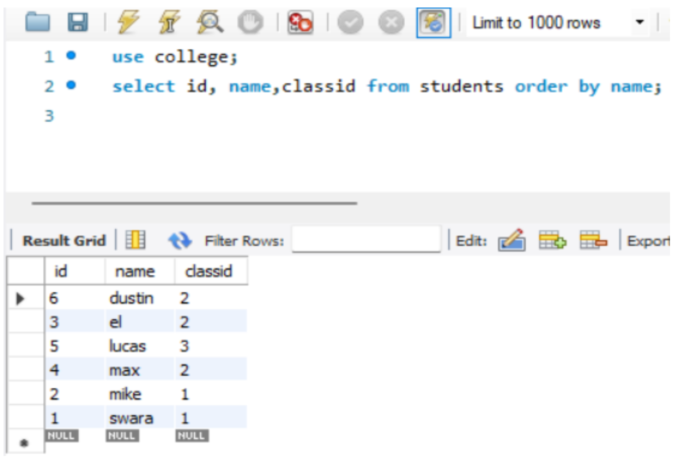
9. And



```
use college;
select id, name,classid from students where classid=2 and id=3;
```

```
id   name    classid
3    el      2
NULL NULL    NULL
```

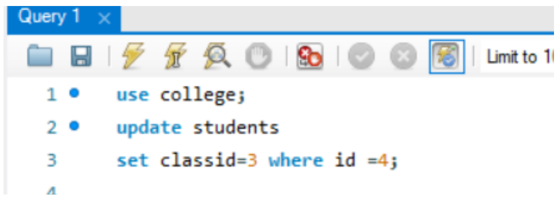
10. Order by



```
use college;
select id, name,classid from students order by name;
```

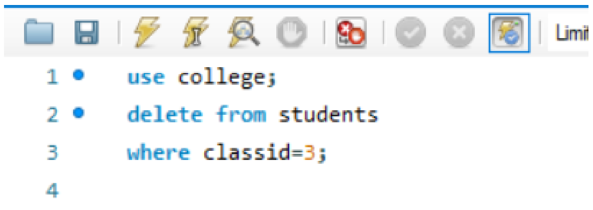
```
id   name    classid
6    dustin  2
3    el      2
5    lucas   3
4    max     2
2    mike    1
1    swara   1
NULL NULL    NULL
```

11. Update statement



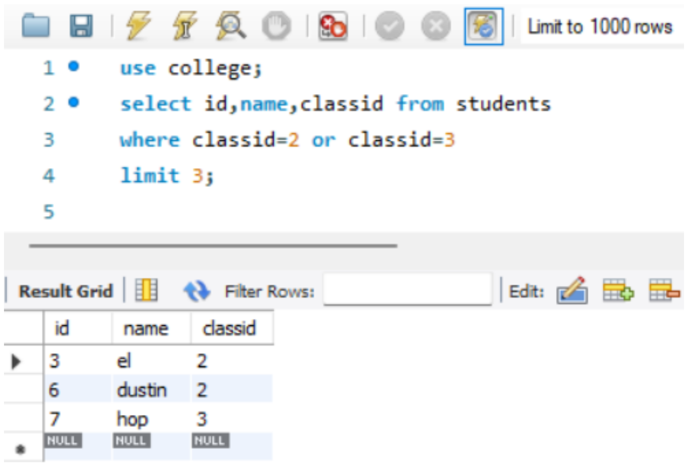
```
use college;
update students
set classid=3 where id =4;
```

12. Delete row



```
use college;
delete from students
where classid=3;
```

13. Select TOP (limit)

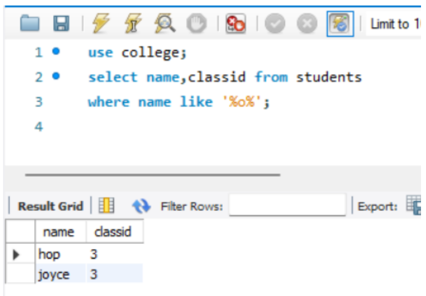


```
use college;
select id,name,classid from students
where classid=2 or classid=3
limit 3;
```

id	name	classid
3	el	2
6	dustin	2
7	hop	3
NULL	NULL	NULL

14. Like (Wildcards are % and _)

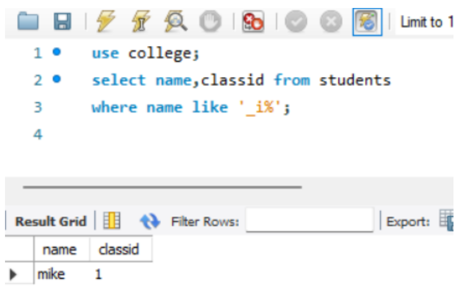
a. %



```
use college;
select name,classid from students
where name like '%o%';
```

name	classid
hop	3
joyce	3

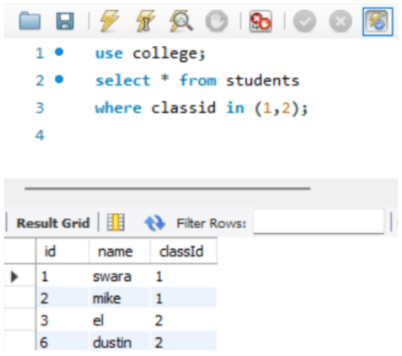
b. _



```
use college;
select name,classid from students
where name like '_i%';
```

name	classid
mike	1

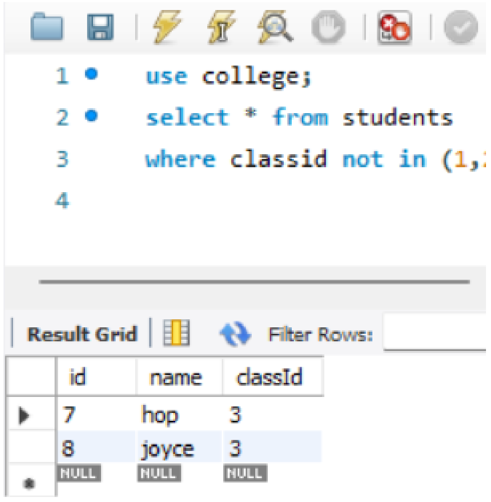
15. In



```
use college;
select * from students
where classid in (1,2);
```

id	name	classId
1	swara	1
2	mike	1
3	el	2
6	dustin	2

16. NOT IN

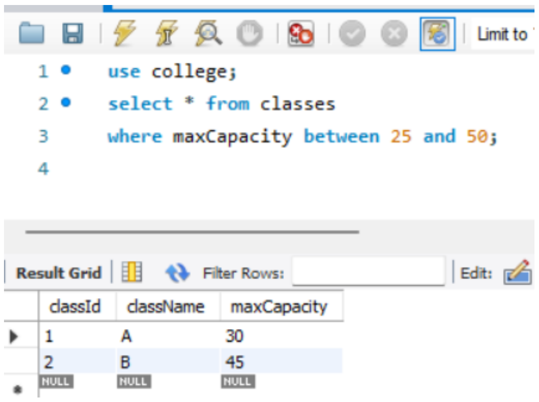


```
use college;
select * from students
where classid not in (1,
```

Result Grid | Filter Rows:

id	name	classId
7	hop	3
8	joyce	3
NULL	NULL	NULL

17. Between



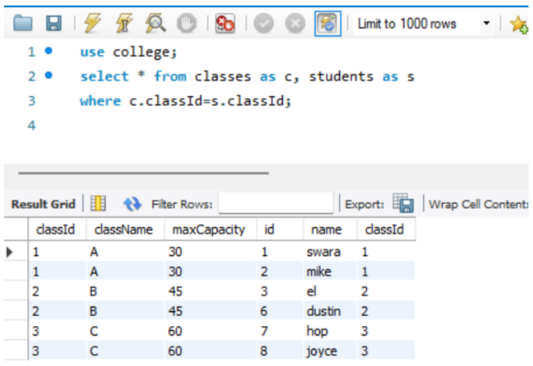
Limit to

```
use college;
select * from classes
where maxCapacity between 25 and 50;
```

Result Grid | Filter Rows: | Edit:

classId	className	maxCapacity
1	A	30
2	B	45
NULL	NULL	NULL

18. Aliases

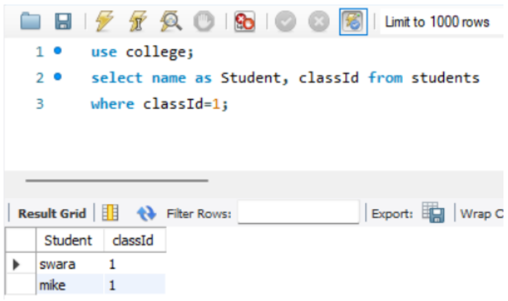


Limit to 1000 rows

```
use college;
select * from classes as c, students as s
where c.classId=s.classId;
```

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

classId	className	maxCapacity	id	name	classId
1	A	30	1	swara	1
1	A	30	2	mike	1
2	B	45	3	el	2
2	B	45	6	dustin	2
3	C	60	7	hop	3
3	C	60	8	joyce	3



Limit to 1000 rows

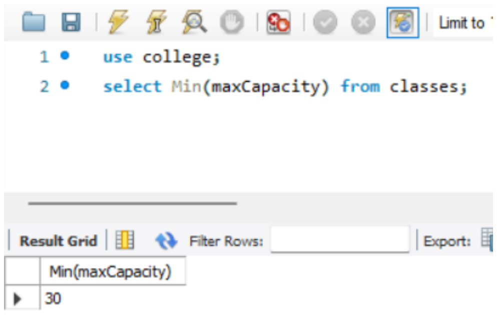
```
use college;
select name as Student, classId from students
where classId=1;
```

Result Grid | Filter Rows: | Export: | Wrap C

Student	classId
swara	1
mike	1

19. Aggregate Functions

a. Min

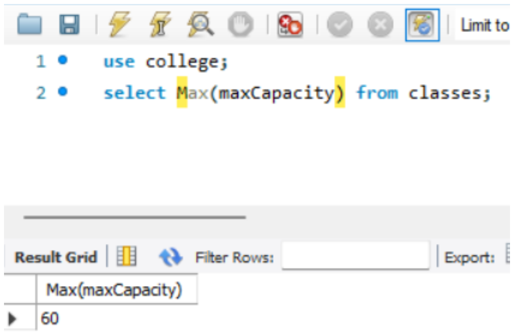


```
use college;
select Min(maxCapacity) from classes;
```

Result Grid | Filter Rows: | Export:

```
Min(maxCapacity)
30
```

b. Max

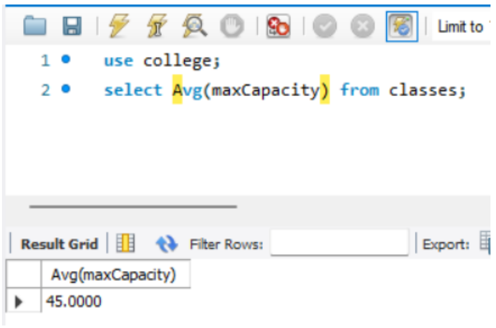


```
use college;
select Max(maxCapacity) from classes;
```

Result Grid | Filter Rows: | Export:

```
Max(maxCapacity)
60
```

c. Avg

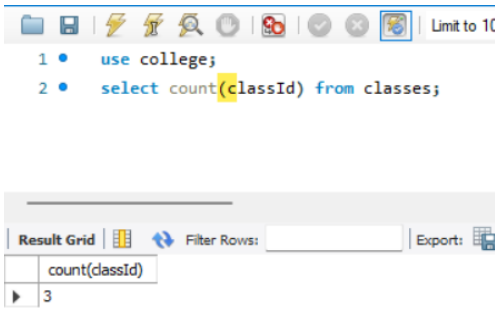


```
use college;
select Avg(maxCapacity) from classes;
```

Result Grid | Filter Rows: | Export:

```
Avg(maxCapacity)
45.0000
```

d. Count

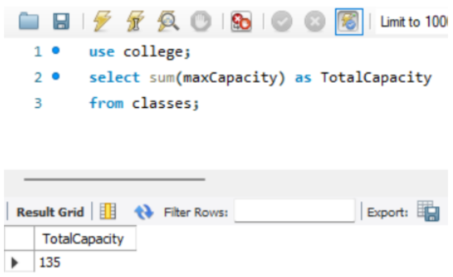


```
use college;
select count(classId) from classes;
```

Result Grid | Filter Rows: | Export:

```
count(classId)
3
```

e. Sum

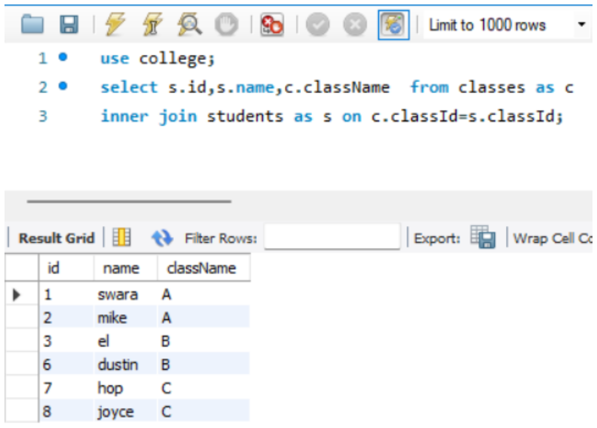


```
use college;
select sum(maxCapacity) as TotalCapacity
from classes;
```

TotalCapacity
135

20. Joins

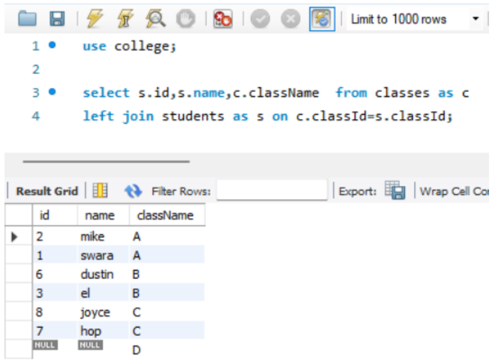
a. Inner Join



```
use college;
select s.id,s.name,c.className  from classes as c
inner join students as s on c.classId=s.classId;
```

id	name	className
1	swara	A
2	mike	A
3	el	B
6	dustin	B
7	hop	C
8	joyce	C

b. Left join

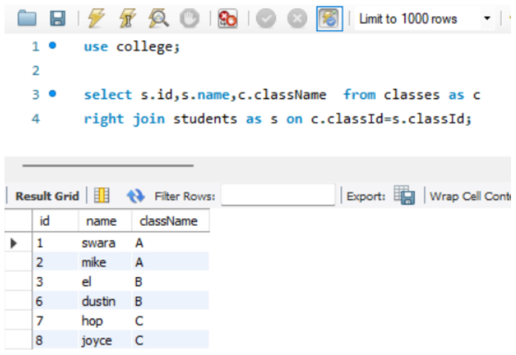


```
use college;

select s.id,s.name,c.className  from classes as c
left join students as s on c.classId=s.classId;
```

id	name	className
2	mike	A
1	swara	A
6	dustin	B
3	el	B
8	joyce	C
7	hop	C
NULL	NULL	D

c. Right join

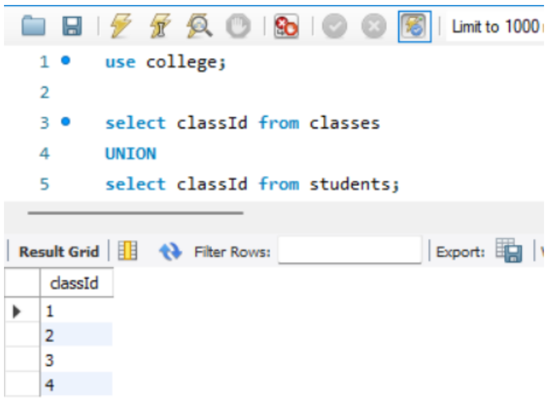


```
use college;

select s.id,s.name,c.className  from classes as c
right join students as s on c.classId=s.classId;
```

id	name	className
1	swara	A
2	mike	A
3	el	B
6	dustin	B
7	hop	C
8	joyce	C

21. UNION

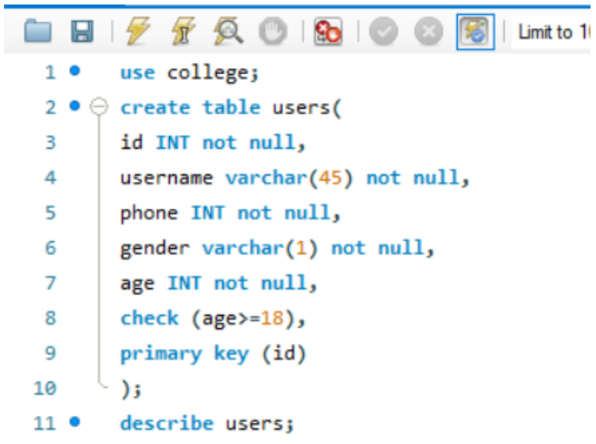


```sql
use college;

select classId from classes
UNION
select classId from students;
```

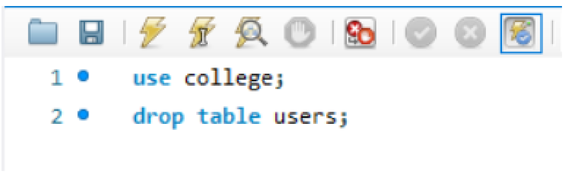
classId
1
2
3
4

22. Create table (check, primary key , not null)



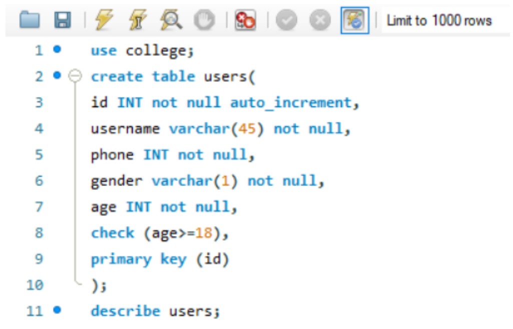
```sql
use college;
create table users(
id INT not null,
username varchar(45) not null,
phone INT not null,
gender varchar(1) not null,
age INT not null,
check (age>=18),
primary key (id)
);
describe users;
```

23. Drop table

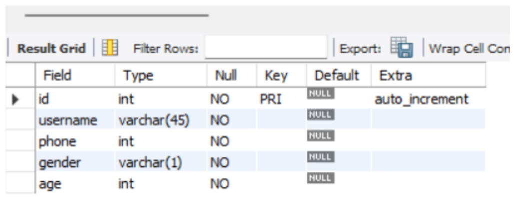


```sql
use college;
drop table users;
```

24. Auto increment

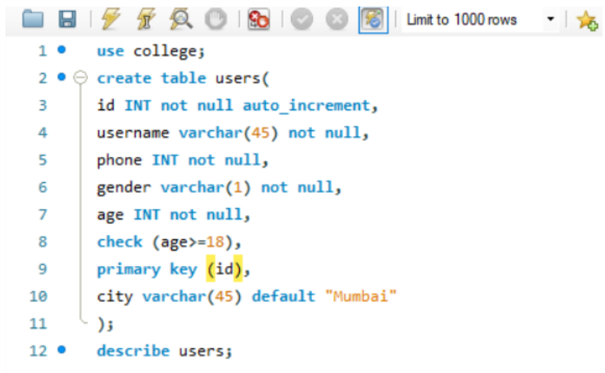


```sql
use college;
create table users(
id INT not null auto_increment,
username varchar(45) not null,
phone INT not null,
gender varchar(1) not null,
age INT not null,
check (age>=18),
primary key (id)
);
describe users;
```

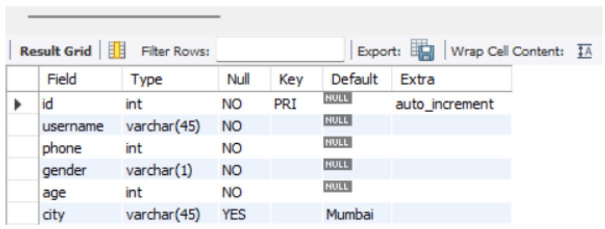


Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	auto_increment
username	varchar(45)	NO		NULL	
phone	int	NO		NULL	
gender	varchar(1)	NO		NULL	
age	int	NO		NULL	

25. Default

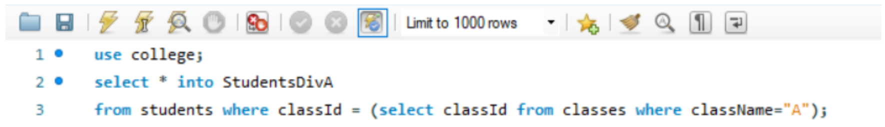


```sql
use college;
create table users(
id INT not null auto_increment,
username varchar(45) not null,
phone INT not null,
gender varchar(1) not null,
age INT not null,
check (age>=18),
primary key (id),
city varchar(45) default "Mumbai"
);
describe users;
```



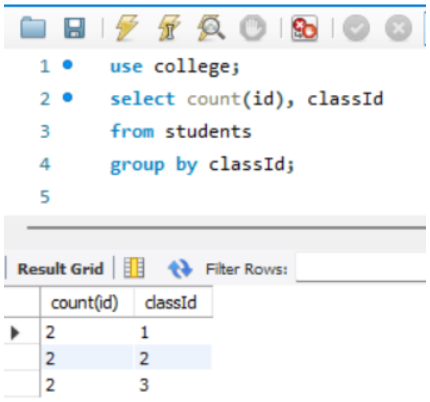
Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	auto_increment
username	varchar(45)	NO		NULL	
phone	int	NO		NULL	
gender	varchar(1)	NO		NULL	
age	int	NO		NULL	
city	varchar(45)	YES		Mumbai	

26. Select Into



```sql
use college;
select * into StudentsDivA
from students where classId = (select classId from classes where className="A");
```

27. Group by

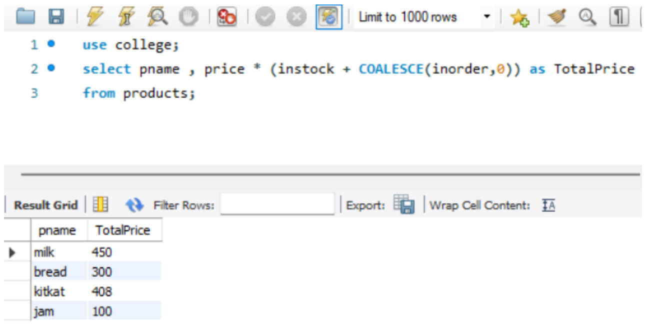


```
use college;
select count(id), classId
from students
group by classId;
```

Result Grid | Filter Rows:

```
count(id)  classId
2          1
2          2
2          3
```

28. Null Functions

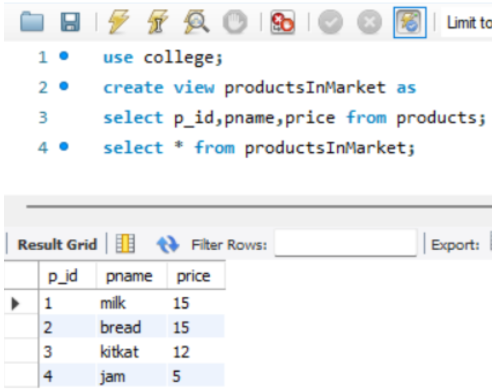


```
use college;
select pname , price * (instock + COALESCE(inorder,0)) as TotalPrice
from products;
```

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

```
pname   TotalPrice
milk    450
bread   300
kitkat  408
jam     100
```

29. Views

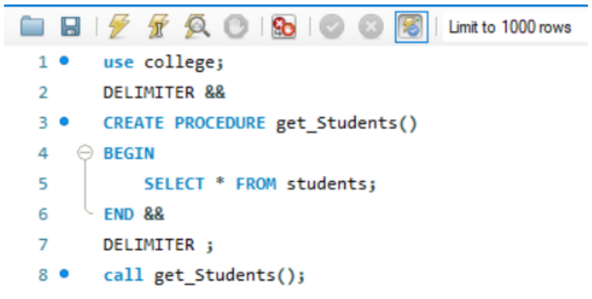


```
use college;
create view productsInMarket as
select p_id,pname,price from products;
select * from productsInMarket;
```

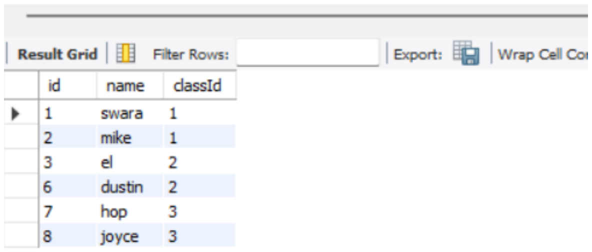
Result Grid | Filter Rows: | Export:

```
p_id  pname   price
1     milk    15
2     bread   15
3     kitkat  12
4     jam     5
```

30. Stored Procedure

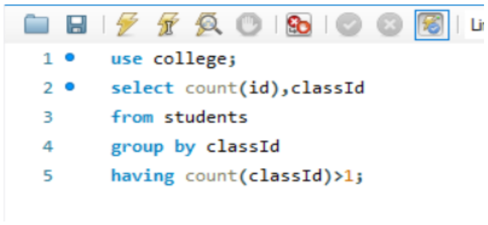


```
use college;
DELIMITER &&
CREATE PROCEDURE get_Students()
BEGIN
    SELECT * FROM students;
END &&
DELIMITER ;
call get_Students();
```

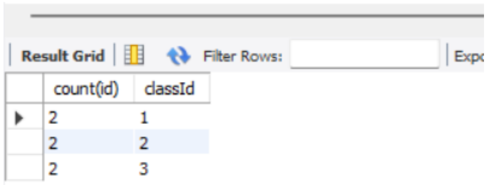


id	name	classId
1	swara	1
2	mike	1
3	el	2
6	dustin	2
7	hop	3
8	joyce	3

31. Having

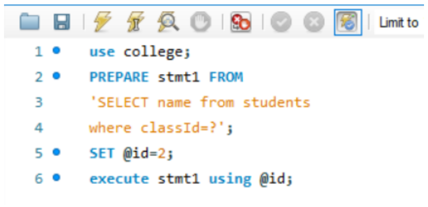


```
use college;
select count(id),classId
from students
group by classId
having count(classId)>1;
```

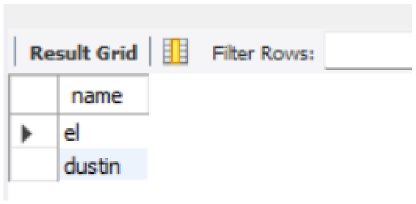


count(id)	classId
2	1
2	2
2	3

32. Prepared Statement



```
use college;
PREPARE stmt1 FROM
'SELECT name from students
where classId=?';
SET @id=2;
execute stmt1 using @id;
```



name
el
dustin