Operators :

=	equal
<>	not equal (!=)
>	
<	
>=	
<=	

Students

st_id	name	marks	total_marks	result
1	Nitesh	72	100	
2	Aman	28	100	
3	Amrendra	98	100	
4	Payyabula	85	100	
5	Ajay	78	100	
6	vikash	67	100	
7	sashil	82	100	

```
select  *
from    students
where   marks > 70;
```

LOGICAL OPERATORS

AND

OR

NOT

marks >= 70 and marks <= 90

```
select *
from students
where marks >= 70 and marks <= 90;
```

```
select *
from students
where marks between 70 and 90 ;
```

70 → inclusive, 90 → inclusive

8) Retrive data from table where age is (22, 24, 28, 32]

```
select *
from students
where age = 22 OR age = 24 OR age = 28 OR age = 32;
```

```
select *
from students
```

where age IN (22, 24, 28, 32);

UPdate

this keyword is used for modify any existing record.

```
UPdate {table_name}
SET { column = value}
where { conditions}
```

if you update a single column for all row.

```
UPdate students
set total_marks = 100;
```

to update a specific row.

```
UPdate students
set name = 'virat'
where name = 'Ajay';
```

- Update result column if marks ≥ 70 then put Pass otherwise fail.

```
UPdate students
Set result = IF(marks ≥ 70, 'Pass', 'fail');
```

Alter

this keyword is used to modify structure

```
alter table {table_name}
add column { column_name data_type}
```

* add result coulumn in student table.

```
alter table students
add column result varchar(10);
```

* modify data type of a column.

change datatype of column name to varchar(25)

```
alter table student
Modify column name varchar(25);
```

* to remove any column we use drop.

Order by

- it give the value based on specific column in increasing or decreasing order.
- default it set as increasing order.
- to get in decreasing order we use desc.
- retrieve all records from student table in increasing order based on age the f_name.

```
select *
from students
order by age, f_name.
```


Students

st_id	name	marks	total_marks	result.
1	Nitesh	72	100	
2	Aman	28	100	
3	Amrendra	98	100	
4	Payyabula	85	100	
5	Ajay	78	100	
6	vikash	67	100	
7	sachil	82	100	
8	NILESH	75	100	
9	nilesh	83	100	
10	NILesh	74	100	

* retrive all records from students table where f_name is nilesh.
(1, 8, 9, 10)

SQL is case incensitive

DISTINCT → to get different records.

```
select distinct name
from students
```

```
select distinct c1, c2, c3
from {table name}
```

LIMIT

```
select *
from students
Limit 3;
```

AS

it is used for aliasing.

- for temp change.

```
select name as 'st_name', marks as st_marks
from students.
```

LIKE

- get all records from students table where name is Aman.

```
select *
from students
where Like Aman;
```

am%.

(Perc) % → any number of any charcters

(underscore) _ → any single charater

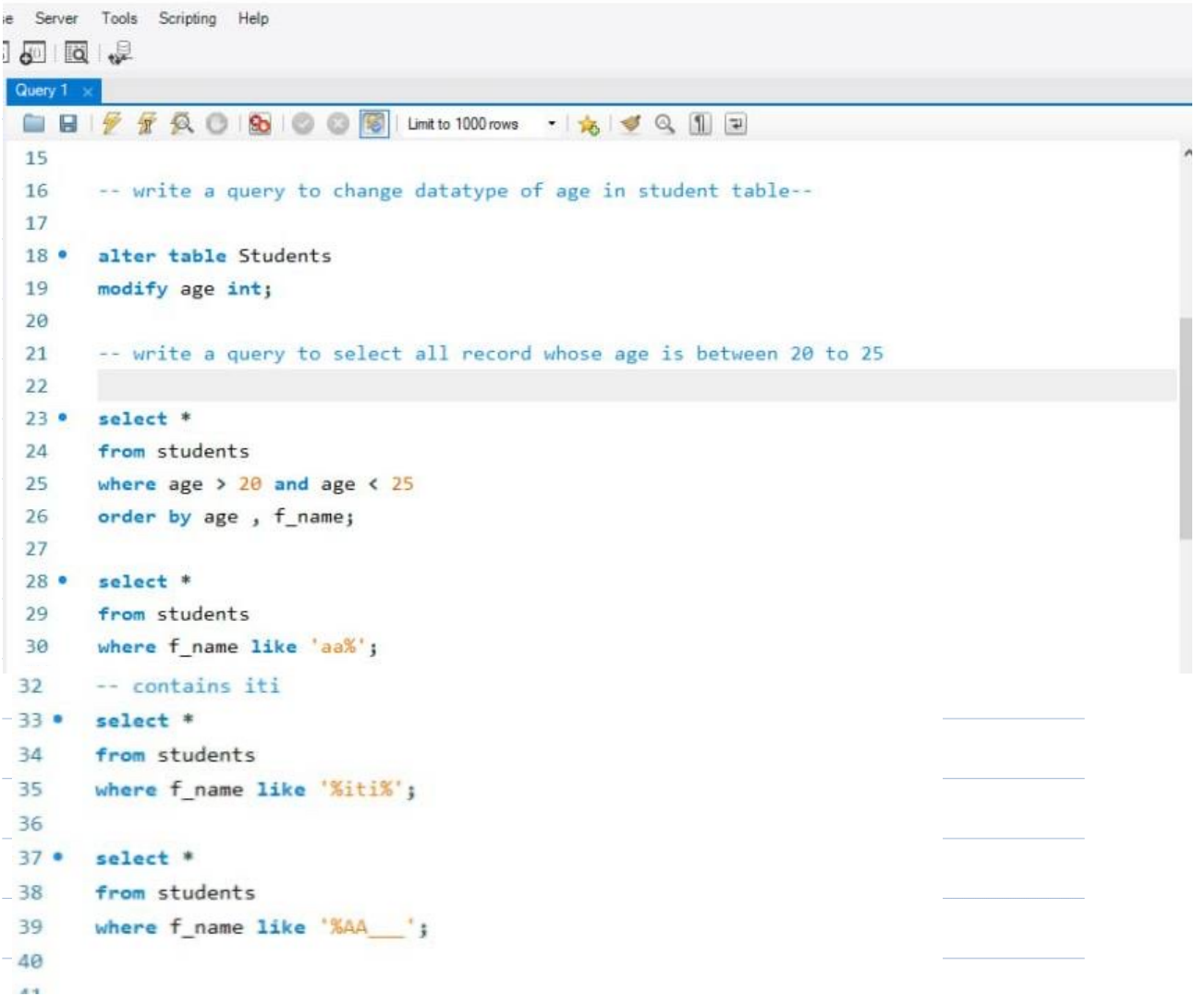
```
select  *                    Aman
from    students             Amrendra
where   Like  Am%;

select  *
from    students             aman
where   Like  Am__;
```

- retrive the record from student table where name is contains deep.

```
select  *
from    students
where   Like  %deep%;
```

retrive the record from student table where length of name is 8 and start with "aa" ?.

aa ------


Server
Tools
Scripting
Help
Query 1
Limit to 1000 rows
-- write a query to change datatype of age in student table--
alter table Students
modify age int;
-- write a query to select all record whose age is between 20 to 25
select *
from students
where age > 20 and age < 25
order by age , f_name;
select *
from students
where f_name like 'aa%';
-- contains iti
select *
from students
where f_name like '%iti%';
select *
from students
where f_name like '%AA___';