OPERATORS

Relational / Comparison Operator:

Let us now understand relational/comparison operator by considering the table below:

student

s_id	name	age	grade
1	John	21	72
2	Mike	20	68
3	Kate	22	86
4	Andy	21	94

Scenario 1: Get all the students with grade greater than 80

```
SELECT * FROM student WHERE grade > 80;
```

Output:

student

s_id	name	age	grade
3	Kate	22	86
4	Andy	21	94

If you observe the result above, you got the table with student greater than 80

Scenario 2: Get all the students with grade less than 80

```
SELECT * FROM student WHERE grade < 80;
```

Output:

student

s_id	name	age	grade
1	John	21	72
2	Mike	20	68

Scenario 3: Get all the students with grade equal to 72

```
SELECT * FROM student WHERE grade = 72;
```

Output:

student

s_id	name	age	grade
1	John	21	72

Scenario 4: Get all the students name and age with grade equal to 86

```
SELECT name, age FROM student WHERE grade = 86;
```

Output:

student

name	age
Kate	22

Scenario 5: Get all the students with age not equal to 21

```
SELECT * FROM sql_notes.student WHERE age != 21;
```

Output:

student

s_id	name	age	grade
2	Mike	20	68
3	Kate	22	86

Whenever you are working with Not equal you can use **<> or !=** both will perform same operations

Scenario 6: Get all the students with name greater than John

```
SELECT * FROM sql_notes.student WHERE name > 'John';
```

Output:

student

s_id	name	age	grade

2	Mike	20	68
3	Kate	22	86

Here as you are passing string and trying to fetch the data where student name is greater than John, now it will fetch the data of the student where alphabetically(ASCII Value) the student name starts with greater than J

Let's add one more column DOJ to the student table

```
ALTER TABLE student ADD doj DATE;
```

To cross verify weather doj column is added in the student you can execute the below query

```
DESCRIBE student;
```

Student

Field	Type	Null	Key
s_id	tinyint	NO	PRI
name	varchar(10)	YES	
age	tinyint	YES	
grade	decimal(10,0)	YES	
doj	date	YES	

If you observe from the above output the doj field is successfully added.

Let's try to insert few more entry into the student table by executing the following query

```
INSERT INTO student(s_id, name, age, grade, doj) VALUES (5,
'Alex', 23, 98, '2021-01-26');
INSERT INTO student (s_id, name, age, grade, doj) VALUES (6,
'Bob', 23, 98, '2020-12-26');
INSERT INTO student (s_id, name, age, grade, doj) VALUES (7,
'Tom', 23, 98, '2021-12-26');
```

Output:

student

s_id	name	age	grade	doj
1	John	21	72	NULL
2	Mike	20	68	NULL
3	Kate	22	86	NULL
4	Andy	21	94	NULL
5	Alex	23	98	2021-01-26
6	Bob	23	98	2020-12-26
7	Tom	23	98	2021-12-26

Scenario 6: Write a query to fetch the students of all the data from the above table where date of joining(doj) is greater than '2020-12-26'

```
SELECT * FROM sql_notes.student WHERE doj > '2020-12-26';
```

Output:

s_id	name	age	grade	doj
5	Alex	23	98	2021-01-26
7	Tom	23	98	2021-12-26

The output gets the data to match doj greater than **`'2020-12-26'`**

Logical Operator:

Syntax Of AND

```
SELECT column1, column2 FROM table_name WHERE condition1
AND condition2 ....;
```

Syntax of OR

```
SELECT column1, column2 FROM table_name WHERE condition1
OR condition2 ....;
```

Syntax of NOT

```
SELECT column1, column2 FROM table_name WHERE NOT
condition;
```

Scenario 7: Write a query to fetch the data from the student table where age is greater than 20 and grade is greater 70

```
SELECT * FROM sql_notes.student WHERE age > 20 AND grade
> 70;
```

Output:

s_id	name	age	grade	doj
1	John	21	72	NULL
3	Kate	22	86	NULL
4	Andy	21	94	NULL
5	Alex	23	98	2021-01-26

6	Bob	23	98	2020-12-26
7	Tom	23	98	2021-12-26

Scenario 8: Write a query to fetch the data from the student table where age is greater than 20 or grade is greater 70

```
SELECT * FROM sql_notes.student WHERE age > 20 OR grade > 70;
```

Output:

s_id	name	age	grade	doj
1	John	21	72	NULL
3	Kate	22	86	NULL
4	Andy	21	94	NULL
5	Alex	23	98	2021-01-26
6	Bob	23	98	2020-12-26
7	Tom	23	98	2021-12-26

Scenario 9: Write a query to fetch the data from the student table where age is Not equal to 22

```
SELECT * FROM student WHERE NOT age > 22;
```

Output:

s_id	name	age	grade	doj
1	John	21	72	NULL
2	Mike	20	68	NULL
3	Kate	22	86	NULL
4	Andy	21	94	NULL

Scenario 10: write a query to fetch the data of the students when student id is not less than 4 and either age should be greater than 22 or grade should be greater than 80

```
SELECT * FROM sql_notes.student WHERE NOT s_id < 4 AND
(age < 22 OR grade > 80);
```

Output:

student

s_id	name	age	grade	doj
4	Andy	21	94	NULL
5	Alex	23	98	2021-01-26
6	Bob	23	98	2020-12-26
7	Tom	23	98	2021-12-26