EXPERIMENT – 5

AIM : Implementation of Operators and Views on a given database.

THEORY : ( OPERATORS)

SQL Arithmetic Operators

Assume **'variable a'** holds 10 and **'variable b'** holds 20,

|  |  |  |
| --- | --- | --- |
| **Operator** | **Description** | **Example** |
| + (Addition) | Adds values on either side of the operator. | a + b will give 30 |
| - (Subtraction) | Subtracts right hand operand from left hand operand. | a - b will give -10 |
| \* (Multiplication) | Multiplies values on either side of the operator. | a \* b will give 200 |
| / (Division) | Divides left hand operand by right hand operand. | b / a will give 2 |
| % (Modulus) | Divides left hand operand by right hand operand and returns remainder. | b % a will give 0 |

## SQL Logical Operators

|  |  |
| --- | --- |
| **Sr.No.** | **Operator & Description** |
| 1 | **ALL**  The ALL operator is used to compare a value to all values in another value set. |
| 2 | **AND**  The AND operator allows the existence of multiple conditions in an SQL statement's WHERE clause. |
| 3 | **ANY**  The ANY operator is used to compare a value to any applicable value in the list as per the condition. |
| 4 | **BETWEEN**  The BETWEEN operator is used to search for values that are within a set of values, given the minimum value and the maximum value. |
| 5 | **EXISTS**  The EXISTS operator is used to search for the presence of a row in a specified table that meets a certain criterion. |
| 6 | **IN**  The IN operator is used to compare a value to a list of literal values that have been specified. |
| 7 | **LIKE**  The LIKE operator is used to compare a value to similar values using wildcard operators. |
| 8 | **NOT**  The NOT operator reverses the meaning of the logical operator with which it is used. Eg: NOT EXISTS, NOT BETWEEN, NOT IN, etc. **This is a negate operator.** |
| 9 | **OR**  The OR operator is used to combine multiple conditions in an SQL statement's WHERE clause. |

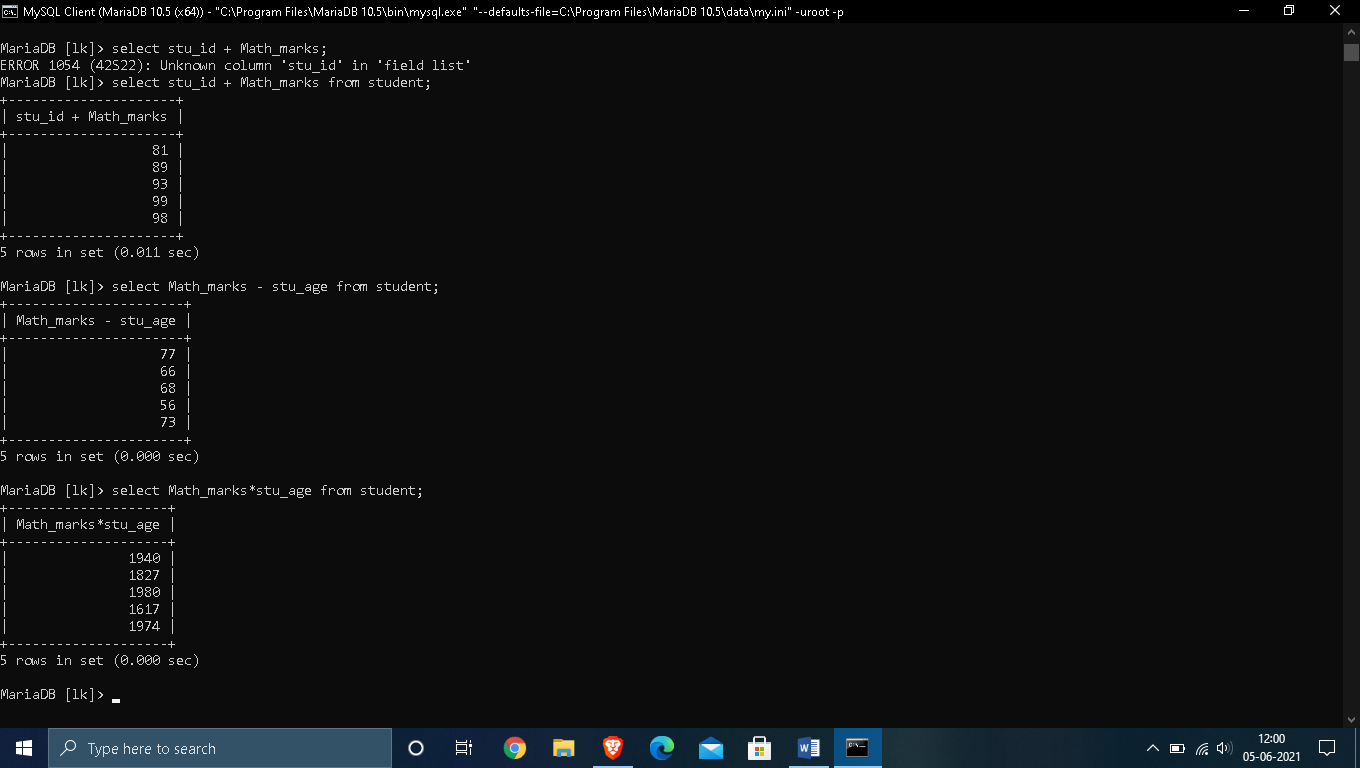
## SQL Comparison Operators

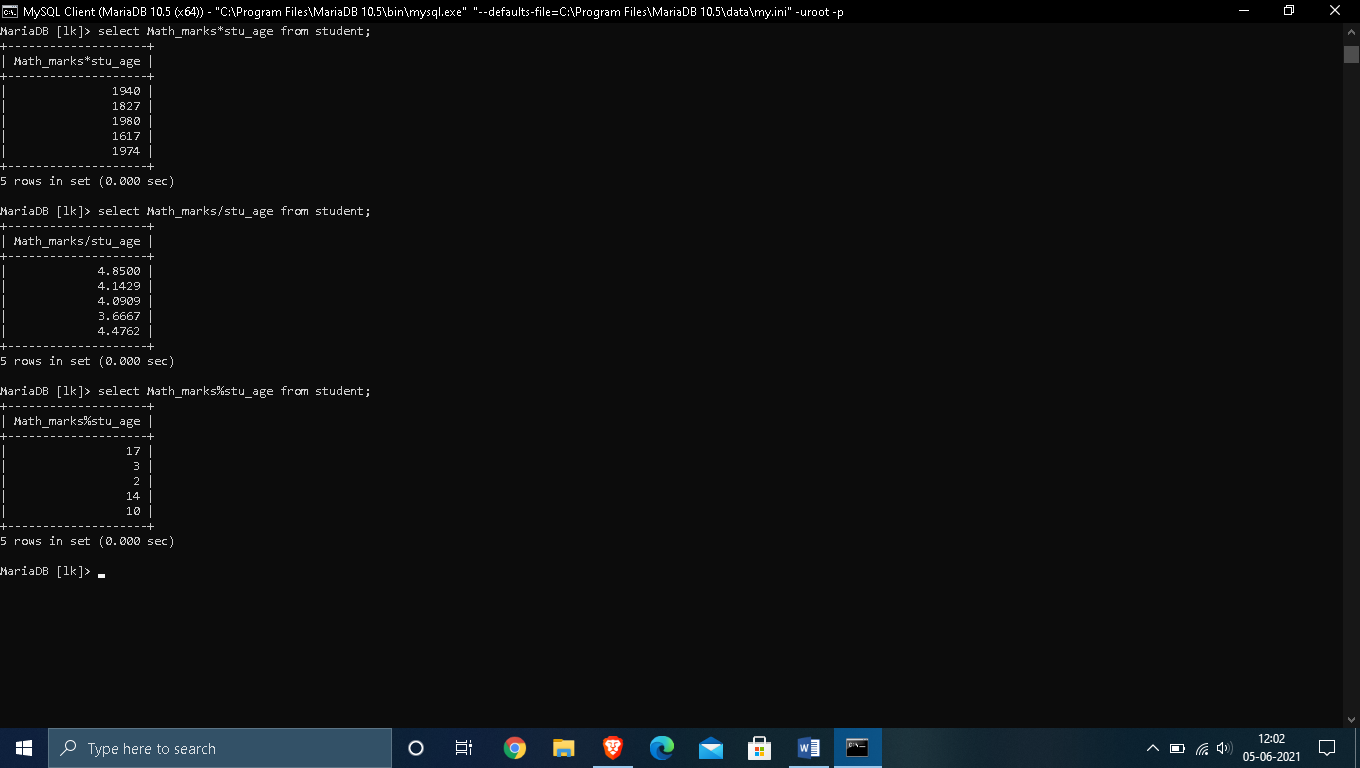
Assume **'variable a'** holds 10 and **'variable b'** holds 20, then −

|  |  |  |
| --- | --- | --- |
| **Operator** | **Description** | **Example** |
| = | Checks if the values of two operands are equal or not, if yes then condition becomes true. | (a = b) is not true. |
| != | Checks if the values of two operands are equal or not, if values are not equal then condition becomes true. | (a != b) is true. |
| <> | Checks if the values of two operands are equal or not, if values are not equal then condition becomes true. | (a <> b) is true. |
| > | Checks if the value of left operand is greater than the value of right operand, if yes then condition becomes true. | (a > b) is not true. |
| < | Checks if the value of left operand is less than the value of right operand, if yes then condition becomes true. | (a < b) is true. |
| >= | Checks if the value of left operand is greater than or equal to the value of right operand, if yes then condition becomes true. | (a >= b) is not true. |
| <= | Checks if the value of left operand is less than or equal to the value of right operand, if yes then condition becomes true. | (a <= b) is true. |

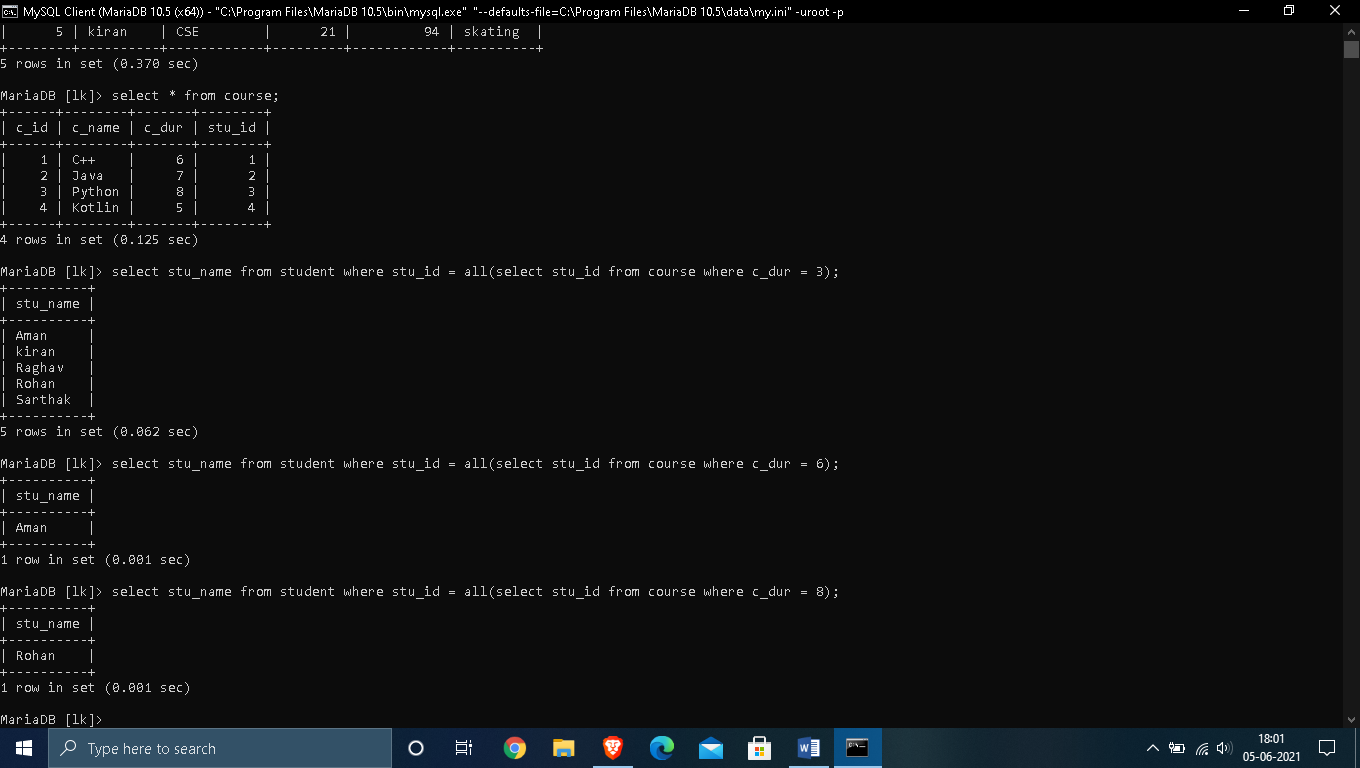
OUTPUT :

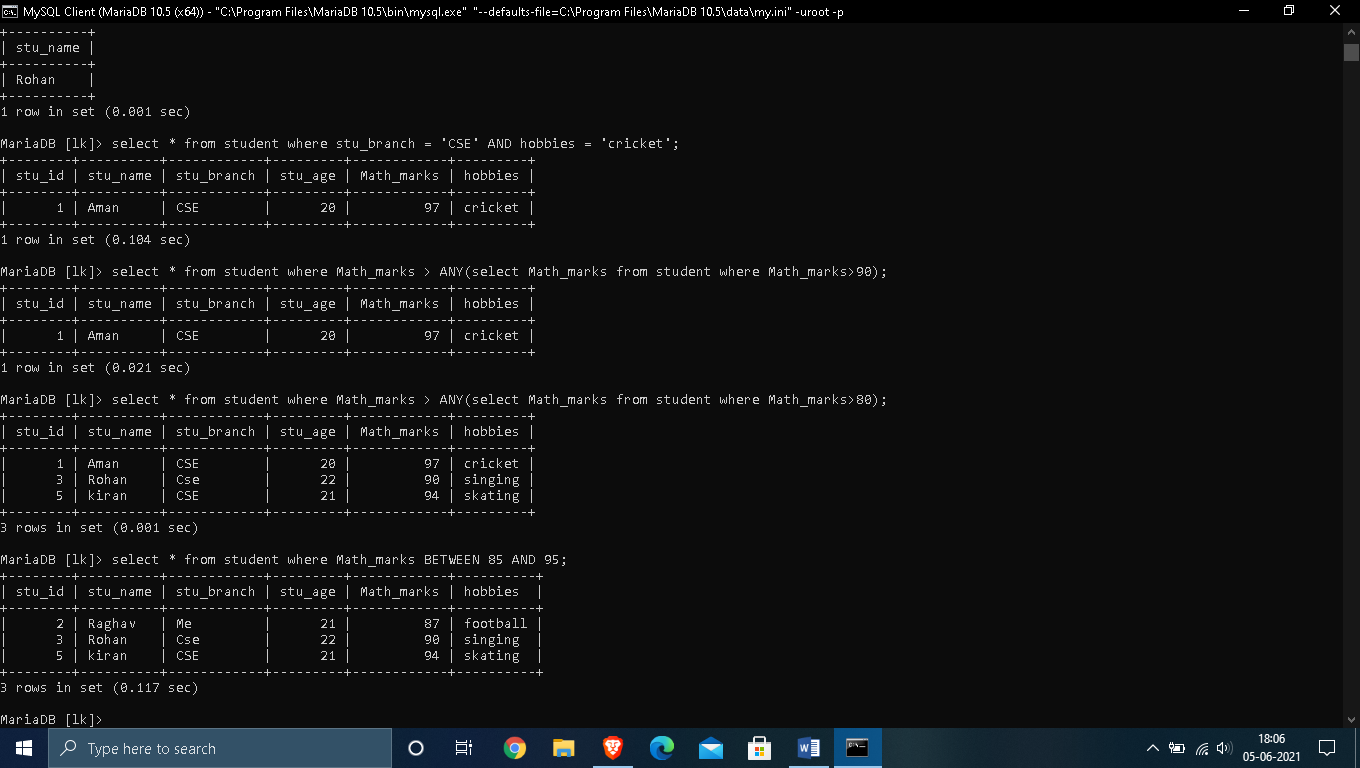
1. Arithmetic Operators:

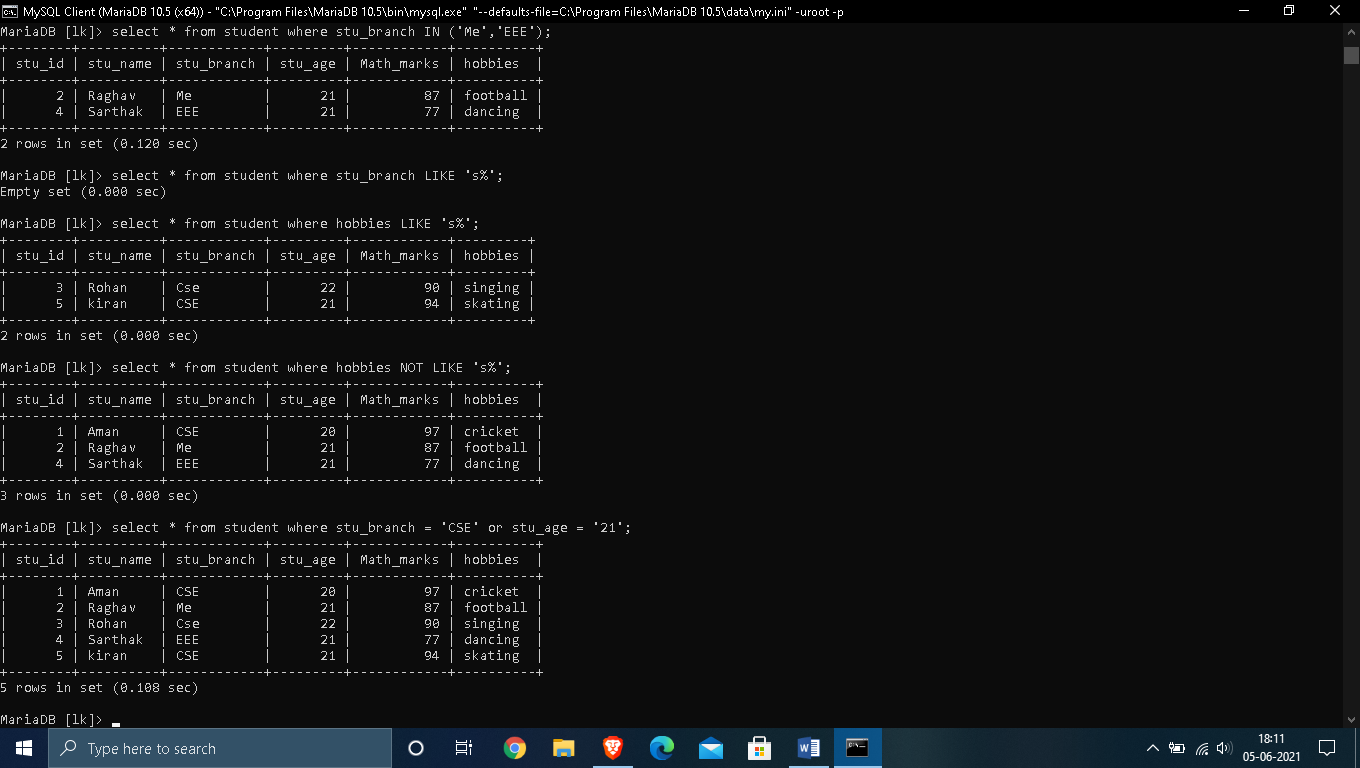


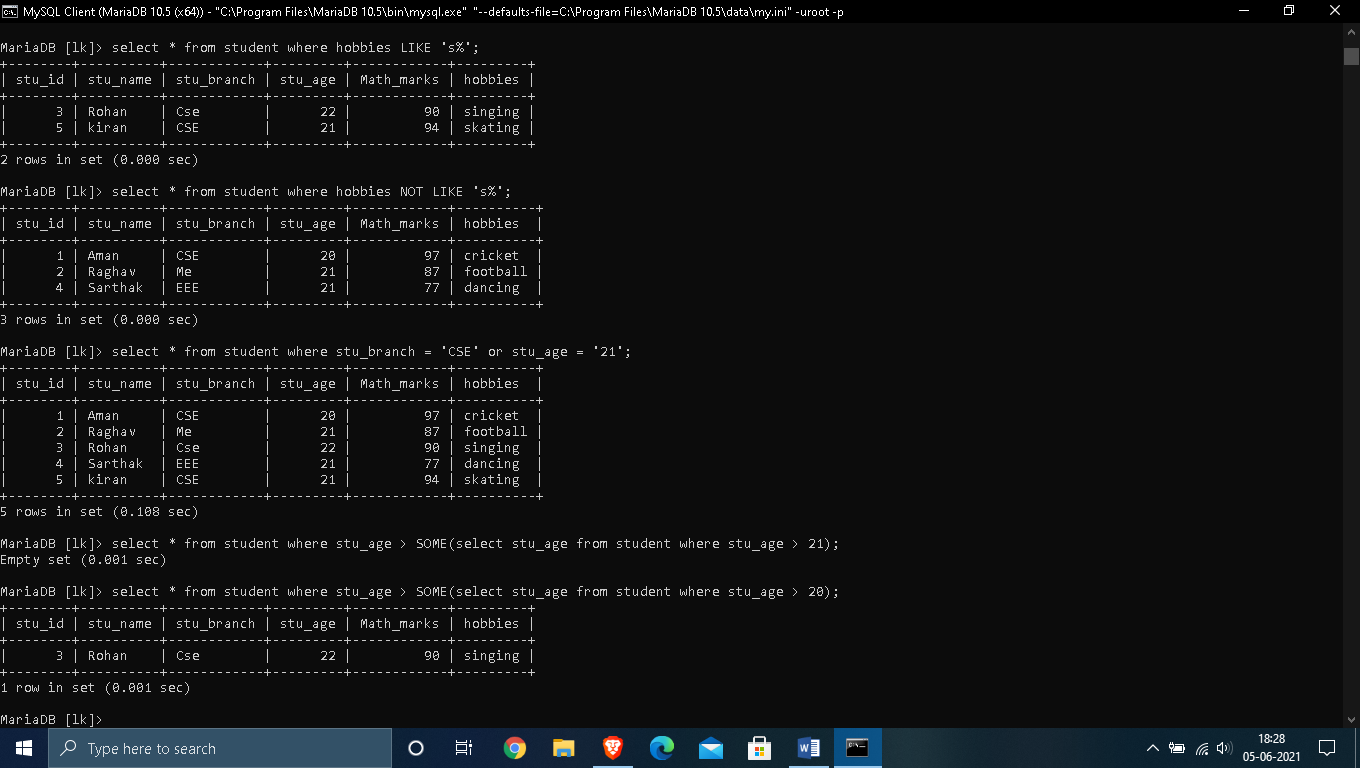


2. Logical Operators :

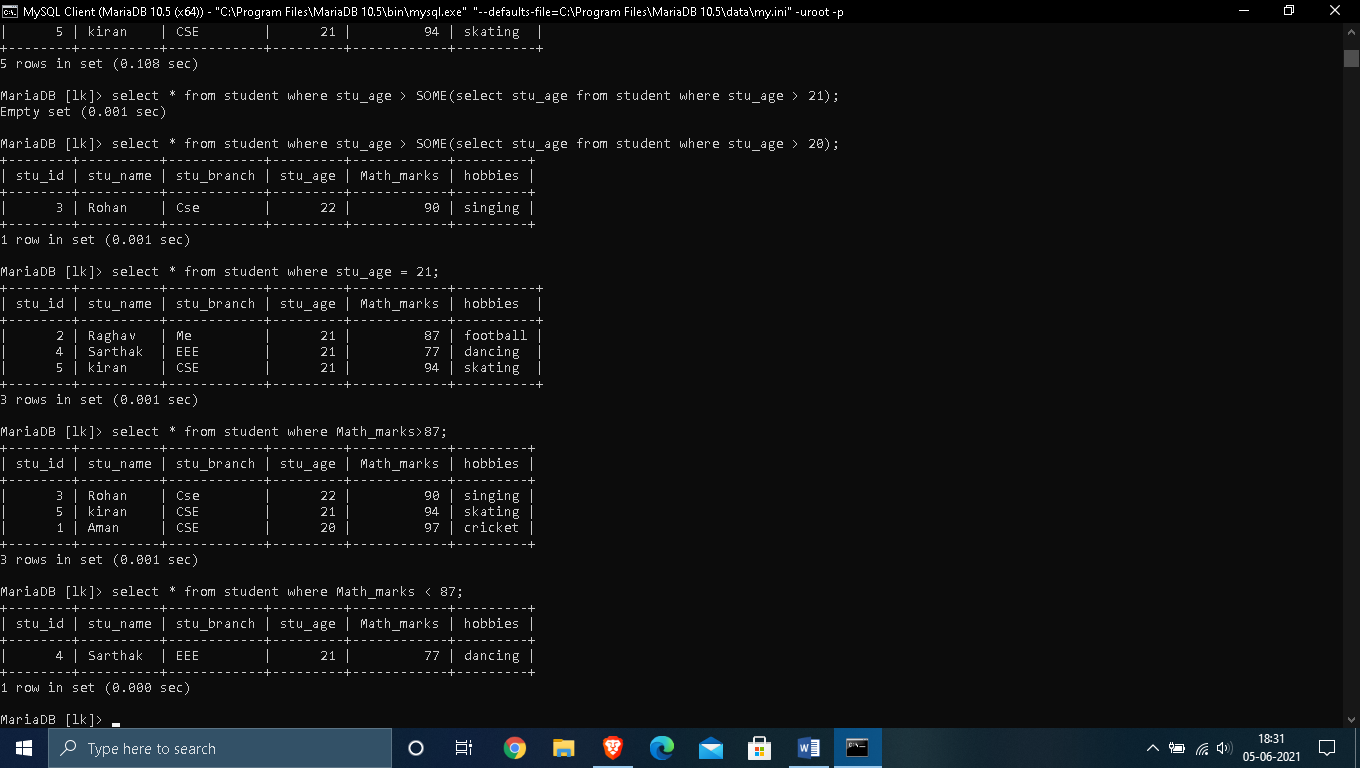


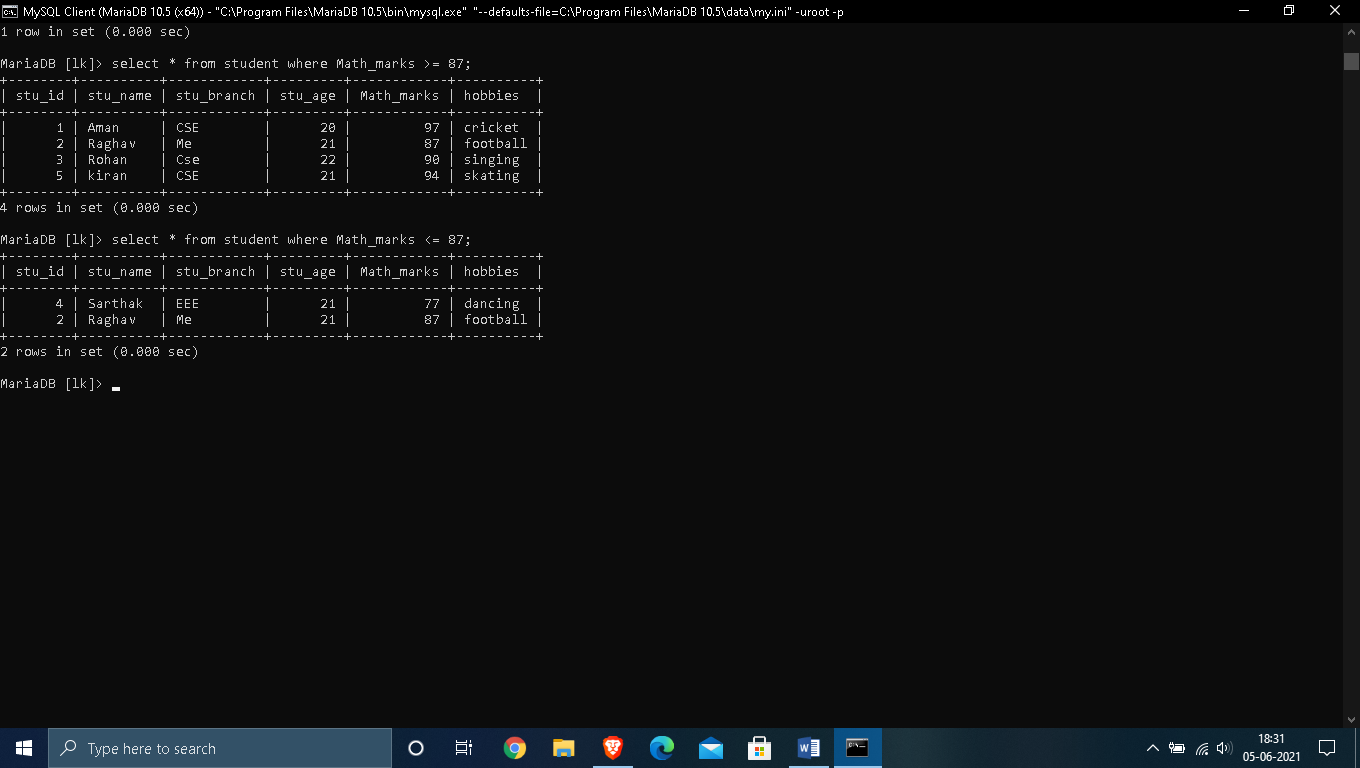






3. Comparison Operators :





VIEWS

* Views in SQL are considered as a virtual table. A view also contains rows and columns.
* To create the view, we can select the fields from one or more tables present in the database.
* A view can either have specific rows based on certain condition or all the rows of a table.

1. Creating view

A view can be created using the **CREATE VIEW** statement. We can create a view from a single table or multiple tables.

**Syntax:**

CREATE VIEW view\_name AS

SELECT column1, column2.....

FROM table\_name

WHERE condition;

2. Creating View from a single table

In this example, we create a View named DetailsView from the table Student\_Detail.

**Query:**

CREATE VIEW DetailsView AS

SELECT NAME, ADDRESS

FROM Student\_Details

WHERE STU\_ID < 4;

Just like table query, we can query the view to view the data.

SELECT \* FROM DetailsView;

3. Creating View from multiple tables

View from multiple tables can be created by simply include multiple tables in the SELECT statement.

In the given example, a view is created named MarksView from two tables Student\_Detail and Student\_Marks.

**Query:**

CREATE VIEW MarksView AS

SELECT Student\_Detail.NAME, Student\_Detail.ADDRESS, Student\_Marks.MARKS

FROM Student\_Detail, Student\_Mark

WHERE Student\_Detail.NAME = Student\_Marks.NAME;

To display data of View MarksView:

SELECT \* FROM MarksView;

4. Deleting View

A view can be deleted using the Drop View statement.

**Syntax**

DROP VIEW view\_name;

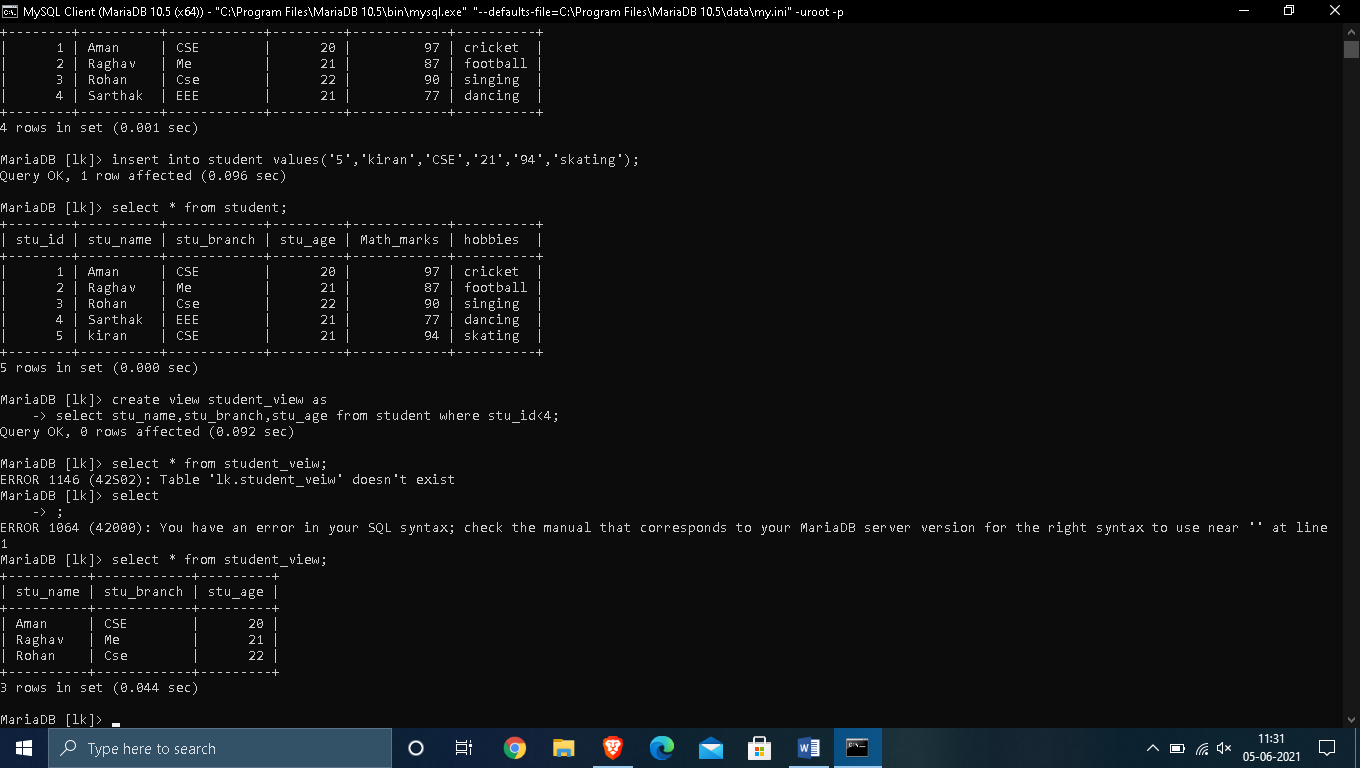
**Example:**

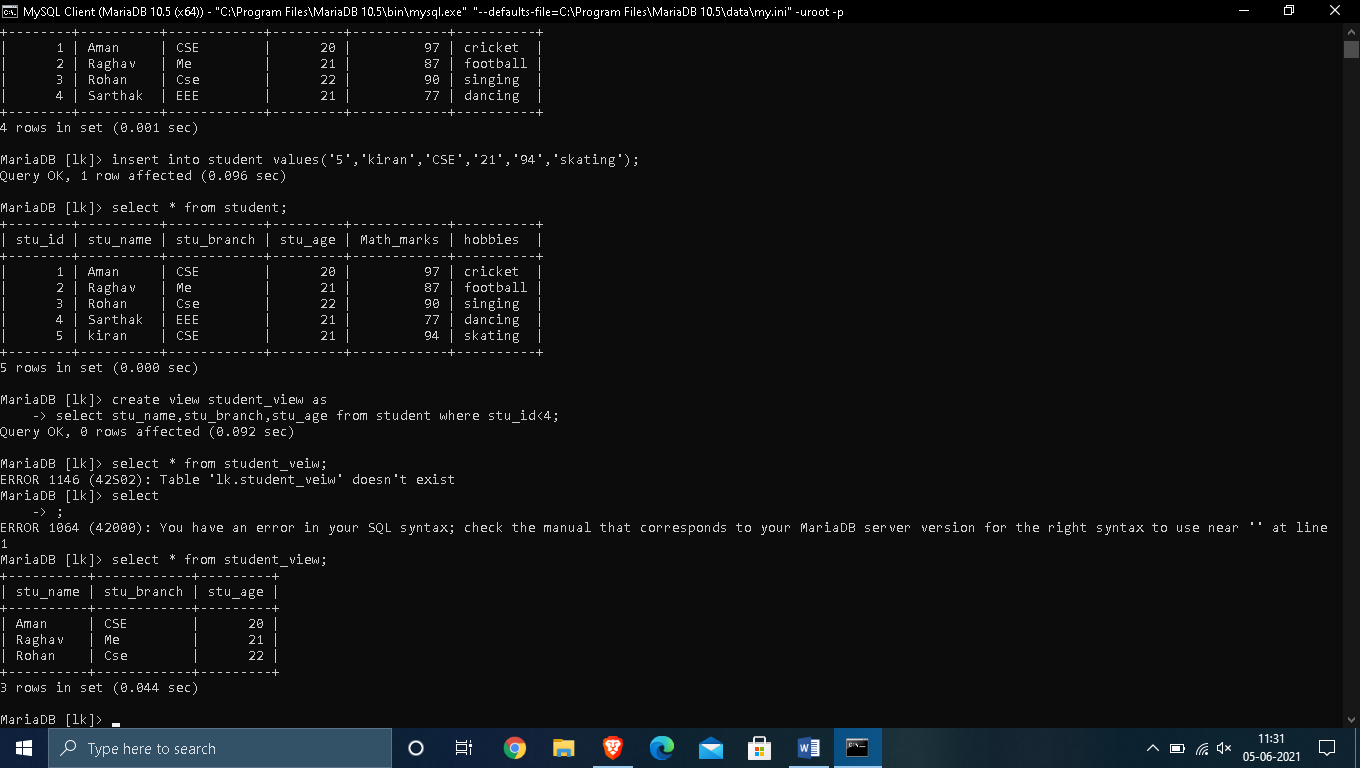
If we want to delete the View **MarksView**, we can do this as:

DROP VIEW MarksView;

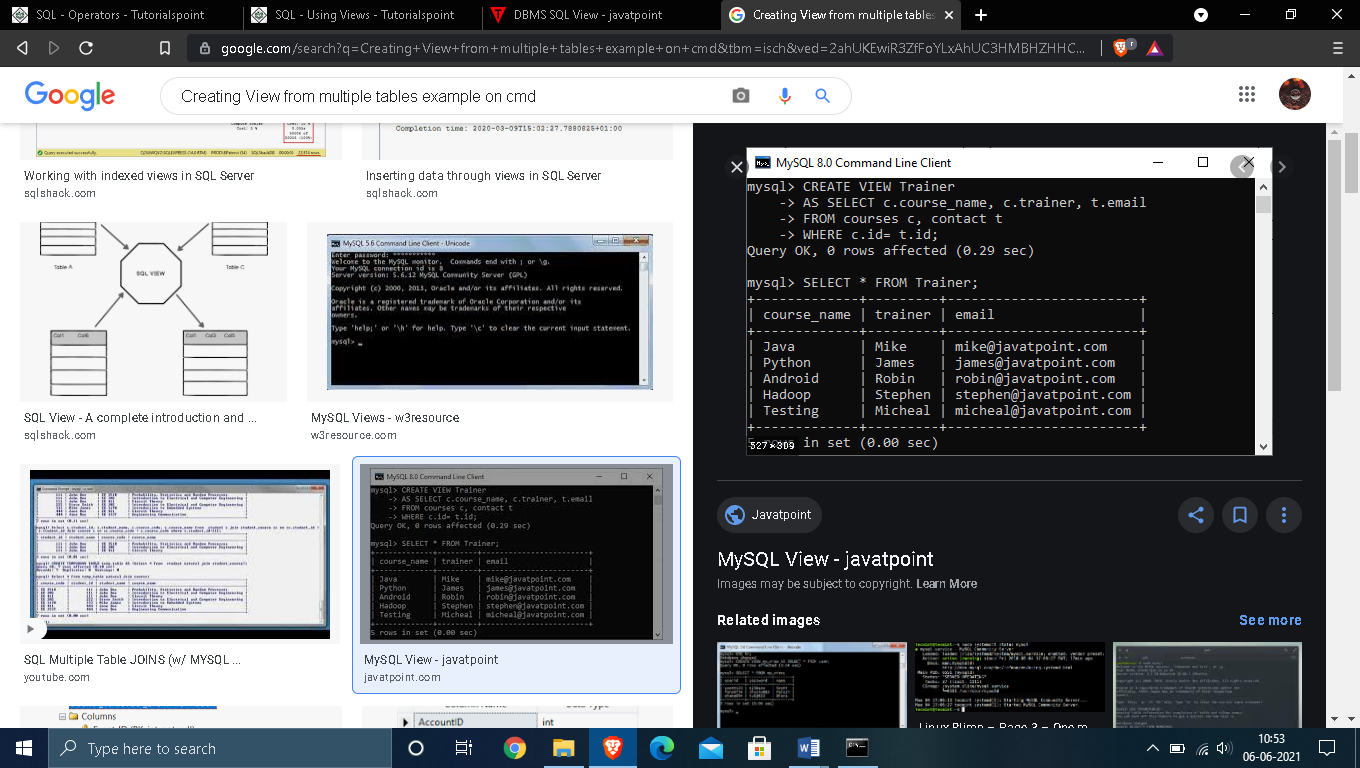
* OUTPUT :

1. Creating a view :





## 2. . Creating View from multiple tables



## 3.  Deleting View

## 