**SQL Documentation**

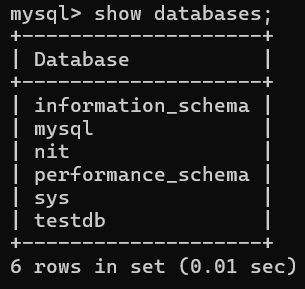
1. **Creating a Database**

* Command:
  + CREATE DATABASE ‘DatabaseName’;



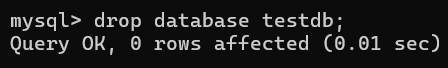
1. **Checking the list of Databases**

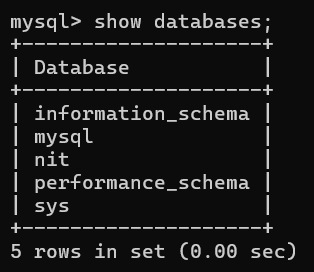
* Command:
  + SHOW DATABASES;



1. **Drop or Delete Database**

* Command:
  + DROP DATABASE ‘DataBaseName’;





1. **Select Database you want to use**

* Command:
  + USE ‘DataBaseName’;



1. **Creating a Table**

* Command:
  + CREATE TABLE ‘TableName’(

column1 datatype,

column2 datatype,

column3 datatype,

.....

columnN datatype,

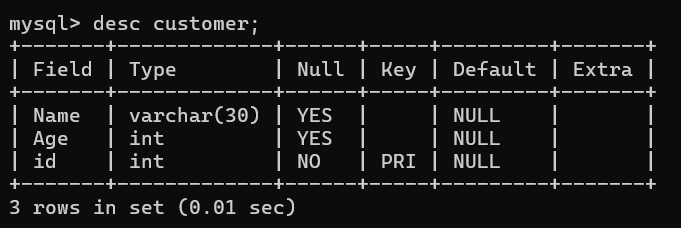
PRIMARY KEY (one or more columns)

);



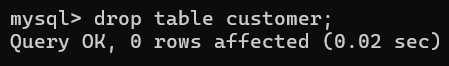
1. **Table Information**

* Command:
  + DESC ‘TableName’;



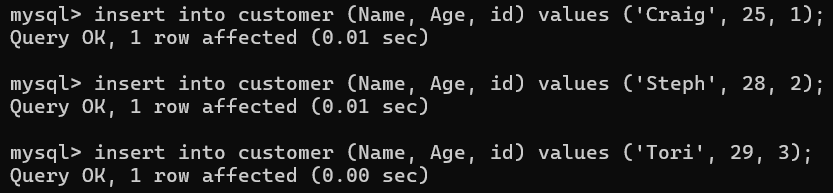
1. **Drop or Delete Table**

* Command:
  + DROP TABLE ‘TableName’;

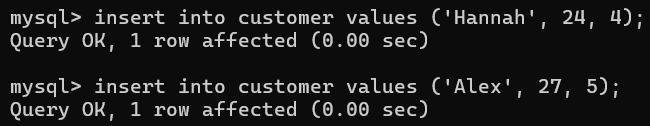


1. **Inserting Values into the Table**

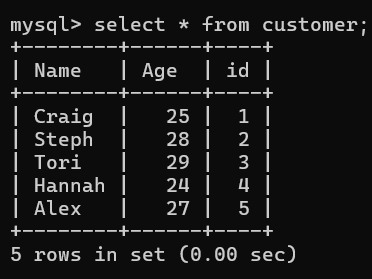
* Command:
  + **Method # 1:**
    - INSERT INTO TABLE\_NAME (column1, column2, column3, columnN)] VALUES (value1, value2, value 3, valueN);



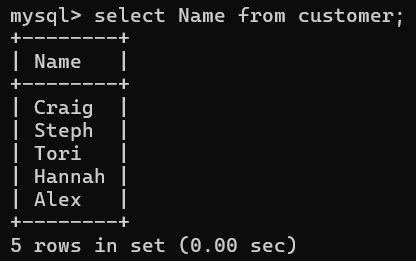
* + **Method # 2:**
    - INSERT INTO TABLE\_NAME VALUES (value1, value2, value3, valueN);



1. **SQL Select Query**
   1. **To Display the Entire Table Values**
   * Command:
     + SELECT \* FROM ‘TableName’;



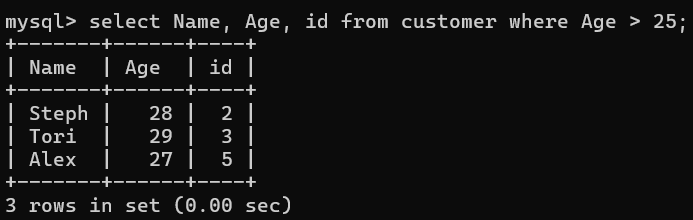
* 1. **To Display Specific Columns of the Table**
  + Command:
    - SELECT ‘ColumnName’ FROM ‘TableName’;



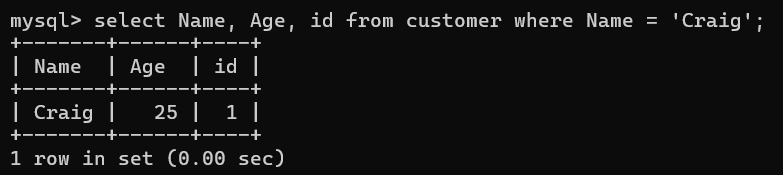
1. **SQL WHERE Clause**

* WHERE clause is used to specify a condition while fetching data from the table.
* Command:
  + SELECT column1, column2, columnN FROM table\_name

WERE [condition]

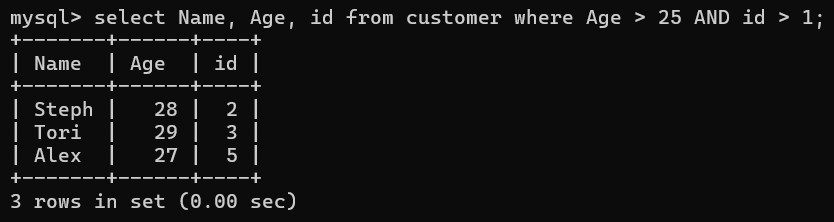


* **If we need a specific customer details**
* Command:
  + SELECT ‘ColumnName’ FROM ‘TableName’ WHERE ‘ColumnName’ = ‘Value’;



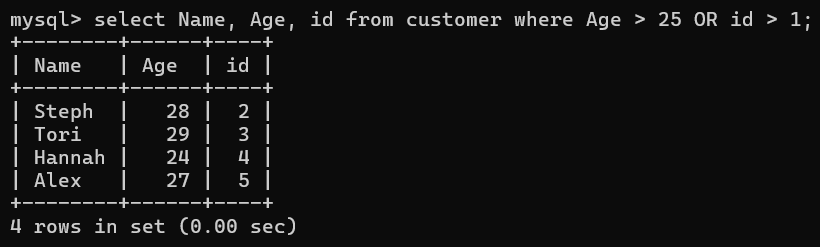
1. **SQL AND, OR Operator**
   1. **AND Operator**
   * We can use this for multiple conditions
   * Command:
     + SELECT column1, column2, columnN FROM table\_name

WHERE [condition1] AND [condition2]...AND [conditionN];



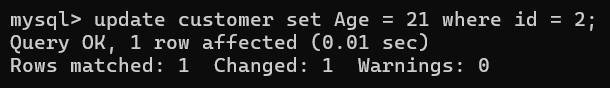
* 1. **OR Operator**
  + Here any one of the condition is satisfied
  + Command:
    - SELECT column1, column2, columnN FROM table\_name

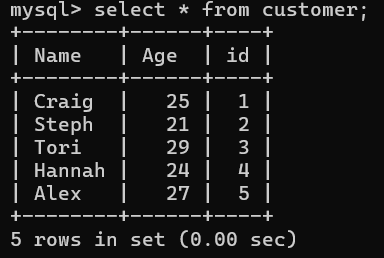
WHERE [condition1] OR [condition2] ...OR [conditionN]



1. **SQL UPDATE Query**

* UPDATE Query is used to modify the existing records in the table.
* Command:
  + UPDATE table\_name SET column1 = value1, column2 = value2...., columnN = valueN WHERE [condition];

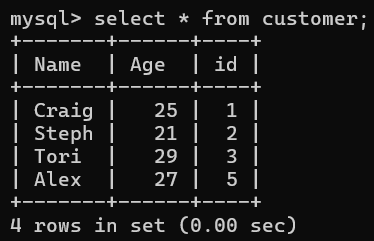




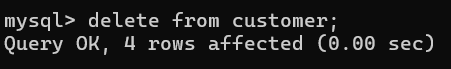
1. **SQL DELETE Query**

* DELETE Query is used to delete the existing record from the table.
  1. **Deleting a specific record**
  + Command:
    - DELETE FROM ‘TableName’ WHERE [condition];





* 1. **Deleting entire table records**
  + Command:
    - DELETE FROM ‘TableName’;





1. **SQL LIKE Clause**

* LIKE Clause is used to compare a value to similar values using wildcard operators.
  1. **The Percent Sign (%)**
  + It represents zero, one or multiple characters
  + Command:
    - SELECT FROM table\_name WHERE column LIKE 'XXXX%'

or

* + - SELECT FROM table\_name WHERE column LIKE '%XXXX%'

or

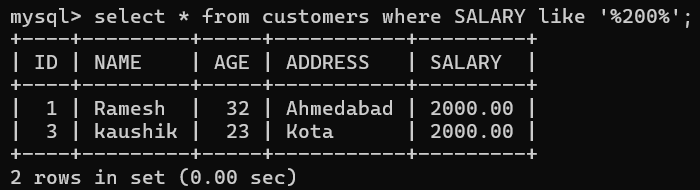
* + - SELECT FROM table\_name WHERE column LIKE 'XXXX\_'

or

* + - SELECT FROM table\_name WHERE column LIKE '\_XXXX'

or

* + - SELECT FROM table\_name WHERE column LIKE '\_XXXX\_'



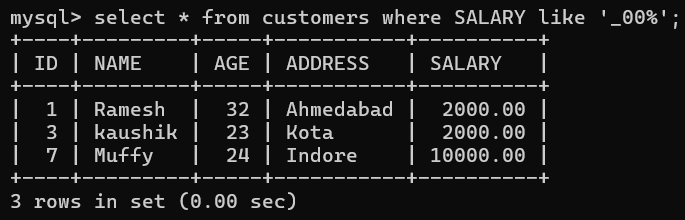
* 1. **The Underscore Sign ( \_ )**
  + It represents a single number or character
  + Command:
    - SELECT FROM table\_name WHERE column LIKE 'XXXX\_'

or

* + - SELECT FROM table\_name WHERE column LIKE '\_XXXX'

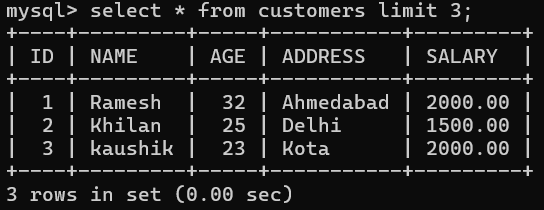
or

* + - SELECT FROM table\_name WHERE column LIKE '\_XXXX\_'



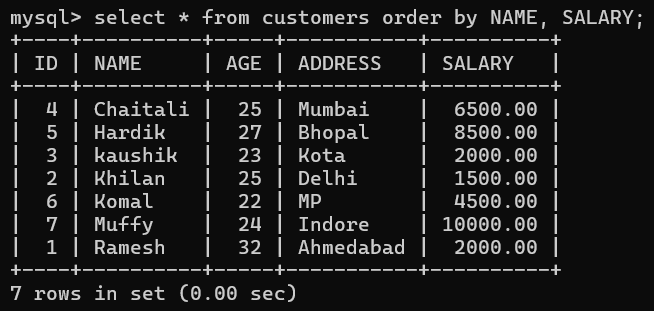
1. **SQL TOP Clause**

* TOP Clause is used to fetch N number of records from the table.
* We can also use the LIMIT Clause for this
  1. **Using LIMIT Clause**
  + This will give N number of records from the table.
  + Command:
    - SELECT \* FROM ‘TableName’ LIMIT ‘N’;

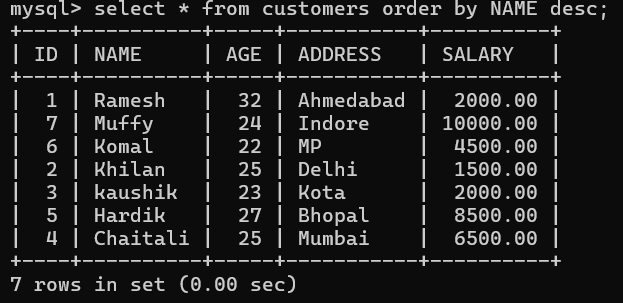


1. **SQL ORDER BY Clause**

* ORDER BY Clause is used to sort the data in ascending or descending order, based on one or more columns.
* Command:
  + SELECT column-list FROM ‘TableName’ [WHERE condition] [ORDER BY column1, column2, .. columnN] [ASC | DESC];
  + Re-arranges the ‘NAME’ column in alphabetical order

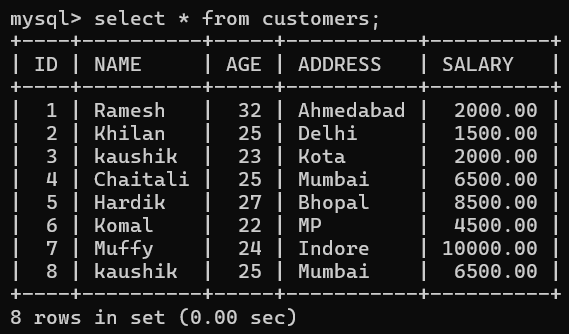


* + Arranging in reverse alphabetical order

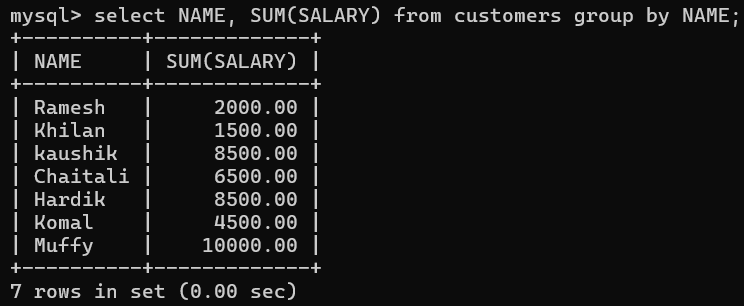


1. **SQL GROUP BY Clause**

* GROUP BY Clause is used in collaboration with the SELECT statement to arrange identical data into groups.
* Command:
  + SELECT column1, column2 FROM ‘TableName’ WHERE [ conditions ] GROUP BY column1, column2 ORDER BY column1, column2
  + We added another value with the same name ‘kaushik;

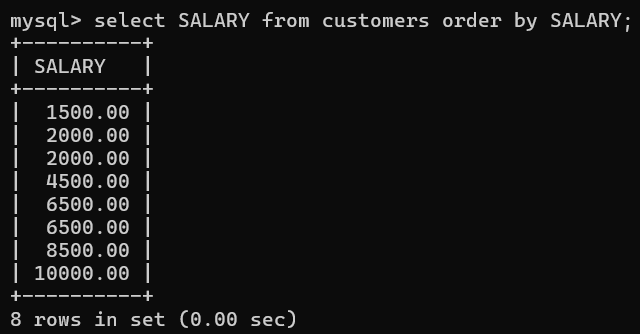


* + Now, when we use GROUP BY, this will add both the SALARY of ‘Kaushik’ and return as a single value

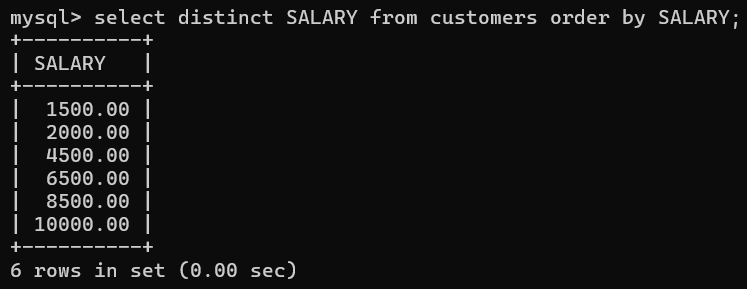


1. **SQL DISTINCT Keyword**

* DISTINCT Keyword is used in conjunction with SELECT statement to eliminate all the duplicate records and fetching only unique records.
* Command:
  + SELECT DISTINCT column1, column2,.....columnN FROM ‘TableName’ WHERE [condition]
  + When there are same values, DISTINCT will return only one

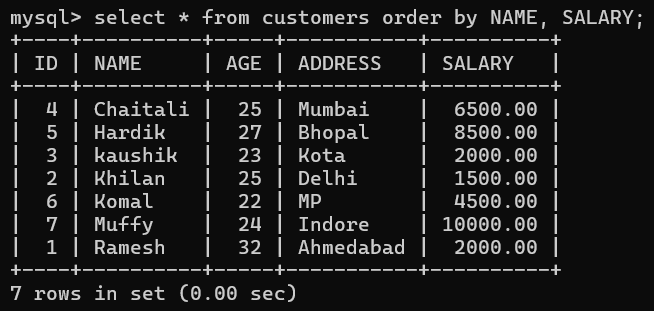


* + Using DISTINCT
  + In this the duplicate values are not printed

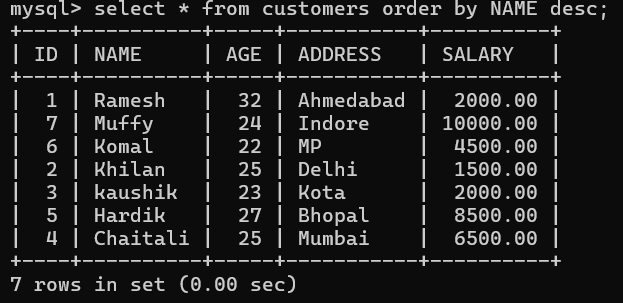


1. **SQL SORTING RESULTS**

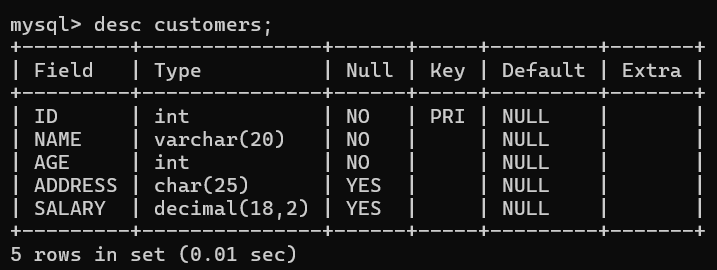
* ORDER BY Clause is used to sort the data in ascending or descending order, based on one or more columns.
* Command:
  + SELECT column-list FROM ‘TableName’ [WHERE condition] [ORDER BY column1, column2, .. columnN] [ASC | DESC];
  + Re-arranges the ‘NAME’ column in alphabetical order



* + Arranging in reverse alphabetical order

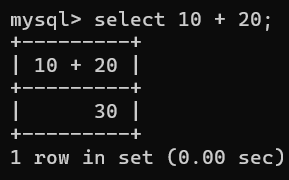


1. **SQL DATA TYPES**

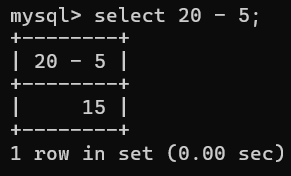


1. **SQL OPERATORS**

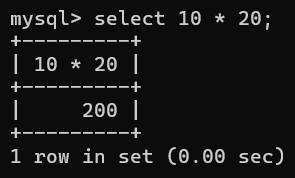
* **Arithmetic Operators**
  + Addition (+):



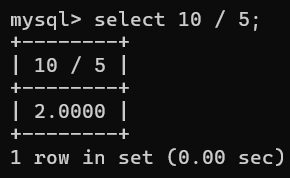
* + Subtraction (-):



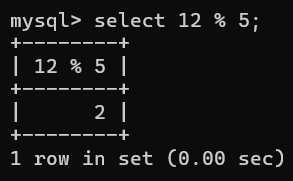
* + Multiplication (\*):



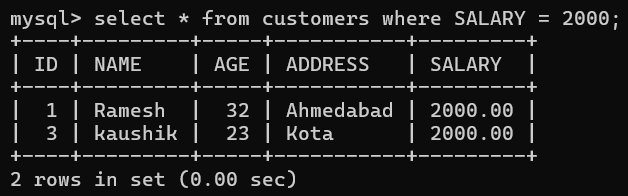
* + Division (/):



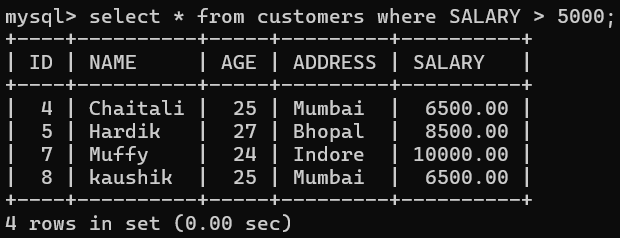
* + Modulus (%):



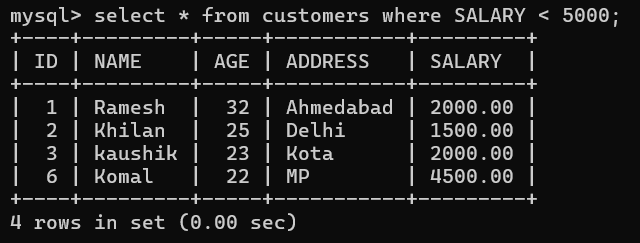
* **Comparison Operator:**
  + Equal To (=):



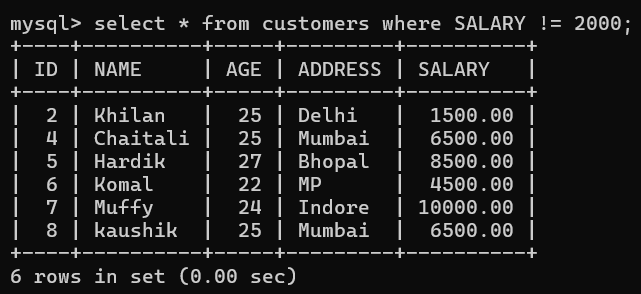
* + Greater Than (>):



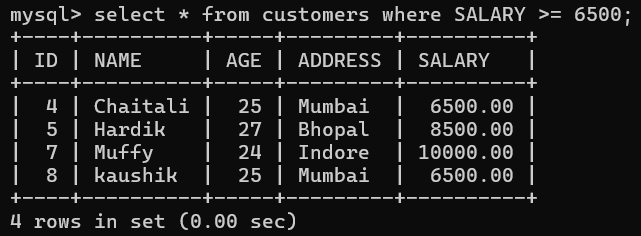
* + Less Than (<):



* + Not Equal To (!=):

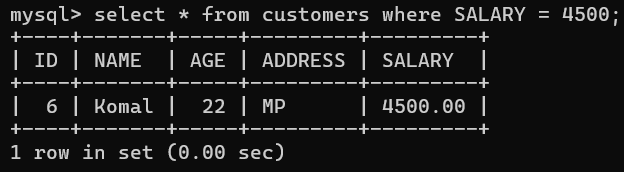


* + Greater Than or Equal To (>=):

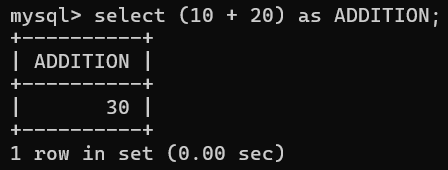


1. **SQL EXPRESSIONS**

* **Boolean Expression**
  + Command:
    - SELECT column1, column2, columnN FROM ‘TableName’ WHERE SINGLE VALUE MATCHTING EXPRESSION;



* **Numeric Expression**
  + Command:
    - SELECT numerical\_expression as OPERATION\_NAME [FROM ‘TableName’ WHERE CONDITION];



* **Date Expression**
  + Command:
    - SELECT CURRENT\_TIMESTAMP;

