

Data Definition Language

- Create a database.
- Show database.(show all your databases).
- Use a database.

```
MySQL 8.0 Command Line Client
mysql> create database mydb;
Query OK, 1 row affected (0.01 sec)

mysql> show databases;
+-----+
| Database |
+-----+
| dharshu  |
| information_schema |
| mydb     |
| mysql    |
| performance_schema |
| sakila   |
| sys      |
| world    |
+-----+
8 rows in set (0.00 sec)

mysql> use mydb;
Database changed
mysql>
```

- Create a table.
- Show tables;(show the created tables).

```
mysql> use mydb;
Database changed
mysql> create table tbl_employee(eid int(5),ename varchar(20),esalary int(5));
Query OK, 0 rows affected, 2 warnings (0.03 sec)

mysql> show tables;
+-----+
| Tables_in_mydb |
+-----+
| tbl_employee    |
+-----+
1 row in set (0.00 sec)

mysql>
```

- Desc cmd to describe the table structure.

desc table_name. (or) describe table_name;

```
mysql> desc tbl_employee;
```

Field	Type	Null	Key	Default	Extra
eid	int	YES		NULL	
ename	varchar(20)	YES		NULL	
esalary	int	YES		NULL	

```
3 rows in set (0.01 sec)
```

- Rename the table

rename table current table_name to new table_name;
Show tables; --> used to verify the changes.

```
mysql> rename table tbl_employee to tbl_employee1;
Query OK, 0 rows affected (0.02 sec)

mysql> show tables;
```

Tables_in_mydb
tbl_employee1

```
1 row in set (0.00 sec)

mysql> desc tbl_employee1;
```

Field	Type	Null	Key	Default	Extra
eid	int	YES		NULL	
ename	varchar(20)	YES		NULL	
esalary	int	YES		NULL	

```
3 rows in set (0.00 sec)
```

- Drop table table_name (along with the data) → to delete a table.
- Drop database database_name; → to delete a database.
- Alter cmd is used to add or modify the table contents.

alter table table_name add field data type(range); (adding new column)

```
mysql> alter table tbl_employee1 add gender char(1);
Query OK, 0 rows affected (0.02 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> desc tbl_employee1;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| eid   | int           | YES  |     | NULL    |       |
| ename | varchar(20)   | YES  |     | NULL    |       |
| esalary | int         | YES  |     | NULL    |       |
| gender | char(1)       | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

- ★ Varchar —> Dynamic memory allocation.(check the length of the data).
- ★ Char —> Static memory allocation, fixed size , faster (no need to check the length of the data).
- Alter command is also used to update or modify or make any change in an existing column.

alter table table_name modify field data type(range);

```
mysql> alter table tbl_employee1 modify gender varchar(10);
Query OK, 0 rows affected (0.04 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> desc tbl_employee1;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| eid   | int           | YES  |     | NULL    |       |
| ename | varchar(20)   | YES  |     | NULL    |       |
| esalary | int         | YES  |     | NULL    |       |
| gender | varchar(10)   | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

- Alter command is used to rename the particular column in an existing table.

alter table table_name rename column oldfield to newfield;

```
mysql> alter table tbl_employee1 rename column gender to egender;
Query OK, 0 rows affected (0.01 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> desc tbl_employee1;
```

Field	Type	Null	Key	Default	Extra
eid	int	YES		NULL	
ename	varchar(20)	YES		NULL	
esalary	int	YES		NULL	
egender	varchar(10)	YES		NULL	

```
4 rows in set (0.00 sec)
```

- Alter command is to drop a particular column in an existing table.

alter table table_name drop column column_name;

```
mysql> alter table tbl_employee1 drop column egender;
Query OK, 0 rows affected (0.01 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> desc tbl_employee1;
```

Field	Type	Null	Key	Default	Extra
eid	int	YES		NULL	
ename	varchar(20)	YES		NULL	
esalary	int	YES		NULL	

```
3 rows in set (0.00 sec)
```

- Drop Table:

Drop table table_name;

Show tables; —> to verify whether the table is deleted or not.

```
mysql> drop table tbl_employee1;  
Query OK, 0 rows affected (0.01 sec)  
  
mysql> show tables;  
Empty set (0.00 sec)  
  
mysql>
```

Data Manipulation Language:

Insert Query:

Syntax:

INSERT INTO table-name (column1,column2,...columnN) VALUES
(value1,value2,value3,...valueN);

- To insert string data types, it is required to keep all the values into **double or single quotes**.

```
mysql> show databases;
+-----+
| Database |
+-----+
| dharshu  |
| information_schema |
| lab_1    |
| mydb     |
| mysql    |
| performance_schema |
| sakila   |
| sys      |
| world    |
+-----+
9 rows in set (0.00 sec)

mysql> use mydb;
Database changed
mysql> show tables;
Empty set (0.00 sec)

mysql> create table tbl_employee(eid int(5), ename varchar(20),esalary int(5));
Query OK, 0 rows affected, 2 warnings (0.02 sec)

mysql> desc tbl_employee;
+-----+-----+-----+-----+-----+-----+
| Field | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| eid   | int       | YES  |     | NULL    |       |
| ename | varchar(20) | YES  |     | NULL    |       |
| esalary | int       | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

- Insert Command : we can insert 'n' number of records in a table.

```
mysql> insert into tbl_employee values ( 101,"Dharshu",12000);
Query OK, 1 row affected (0.00 sec)

mysql> insert into tbl_employee values ( 102,"Minion",16000);
Query OK, 1 row affected (0.00 sec)

mysql> insert into tbl_employee values ( 103,"Dharshana",15000);
Query OK, 1 row affected (0.00 sec)

mysql>
```

- Display or view all the records in a table.

```
mysql> select * from tbl_employee;
+-----+-----+-----+
| eid   | ename   | esalary |
+-----+-----+-----+
| 101   | Dharshu | 12000   |
| 102   | Minion  | 16000   |
| 103   | Dharshana | 15000   |
+-----+-----+-----+
3 rows in set (0.00 sec)
```

- Insert NULL value in the record.(NULL—> Empty,Blank)

```
mysql> insert into tbl_employee values ( 104,null,17000);
Query OK, 1 row affected (0.00 sec)
```

Insert the record into a table using column name:

- The salary column become null because we can't specify the salary record.

```
mysql> insert into tbl_employee ( eid, ename) values (105,"Jenish");
Query OK, 1 row affected (0.01 sec)
```

- View The Table:

```
mysql> select * from tbl_employee;
+-----+-----+-----+
| eid   | ename   | esalary |
+-----+-----+-----+
| 101   | Dharshu | 12000   |
| 102   | Minion  | 16000   |
| 103   | Dharshana | 15000   |
| 104   | NULL    | 17000   |
| 105   | Jenish  | NULL    |
+-----+-----+-----+
5 rows in set (0.00 sec)
```

Select Query:

- Select command is used to fetch all the data from the MySQL table.

Syntax:

```
SELECT field1,field2,...fieldN  
FROM table_name1,table_name2....  
[WHERE Clause]  
[OFFSET M] [LIMIT N]
```

- View Particular column using select Query:

```
mysql> select eid,esalary from tbl_employee;  
+-----+-----+  
| eid | esalary |  
+-----+-----+  
| 101 | 12000 |  
| 102 | 16000 |  
| 103 | 15000 |  
| 104 | 17000 |  
| 105 | NULL |  
+-----+-----+  
5 rows in set (0.00 sec)
```

- Select Query using Where condition in salary.

```
mysql> select * from tbl_employee where esalary > 15000;  
+-----+-----+-----+  
| eid | ename | esalary |  
+-----+-----+-----+  
| 102 | Minion | 16000 |  
| 104 | NULL | 17000 |  
+-----+-----+-----+  
2 rows in set (0.00 sec)
```



```
mysql> select * from tbl_employee where esalary >= 15000;
+-----+-----+-----+
| eid   | ename   | esalary |
+-----+-----+-----+
| 102   | Minion  | 16000   |
| 103   | Dharshana | 15000   |
| 104   | NULL    | 17000   |
+-----+-----+-----+
3 rows in set (0.00 sec)
```

```
mysql> select * from tbl_employee where esalary = 15000;
+-----+-----+-----+
| eid   | ename   | esalary |
+-----+-----+-----+
| 103   | Dharshana | 15000   |
+-----+-----+-----+
1 row in set (0.00 sec)
```

```
mysql> select * from tbl_employee where esalary != 15000;
+-----+-----+-----+
| eid   | ename   | esalary |
+-----+-----+-----+
| 101   | Dharshu  | 12000   |
| 102   | Minion  | 16000   |
| 104   | NULL    | 17000   |
+-----+-----+-----+
3 rows in set (0.00 sec)
```

By using relational operators or comparison operators we cannot compare the NULL value.

- Select Query using Where condition in Name.

```
mysql> select * from tbl_employee where ename = 'Jenish';
+-----+-----+-----+
| eid | ename | esalary |
+-----+-----+-----+
| 105 | Jenish | NULL |
+-----+-----+-----+
1 row in set (0.00 sec)
```

```
mysql> select * from tbl_employee where ename != 'Jenish';
+-----+-----+-----+
| eid | ename | esalary |
+-----+-----+-----+
| 101 | Dharshu | 12000 |
| 102 | Minion | 16000 |
| 103 | Dharshana | 15000 |
+-----+-----+-----+
3 rows in set (0.00 sec)
```

NULL Comparison:

- Is Null:

```
mysql> select * from tbl_employee where ename is null;
+-----+-----+-----+
| eid | ename | esalary |
+-----+-----+-----+
| 104 | NULL | 17000 |
+-----+-----+-----+
1 row in set (0.00 sec)
```

- Is Not Null:

```
mysql> select * from tbl_employee where ename is not null;
+-----+-----+-----+
| eid | ename | esalary |
+-----+-----+-----+
| 101 | Dharshu | 12000 |
| 102 | Minion | 16000 |
| 103 | Dharshana | 15000 |
| 105 | Jenish | NULL |
+-----+-----+-----+
4 rows in set (0.00 sec)
```

Applying 2 conditions:

- **And** condition

```
mysql> select * from tbl_employee where ename is not null and esalary = 16000;
+-----+-----+-----+
| eid | ename | esalary |
+-----+-----+-----+
| 102 | Minion | 16000 |
+-----+-----+-----+
1 row in set (0.00 sec)
```

- **Or** condition

```
mysql> select * from tbl_employee where ename is not null or esalary = 16000;
+-----+-----+-----+
| eid | ename | esalary |
+-----+-----+-----+
| 101 | Dharshu | 12000 |
| 102 | Minion | 16000 |
| 103 | Dharshana | 15000 |
| 105 | Jenish | NULL |
+-----+-----+-----+
4 rows in set (0.00 sec)
```

- By using the “**in**” operator we can display the list of records.

If the given eid is present in the table it will display the record otherwise it will not display.

```
mysql> select * from tbl_employee where eid in(101, 103, 106);
+-----+-----+-----+
| eid | ename | esalary |
+-----+-----+-----+
| 101 | Dharshu | 12000 |
| 103 | Dharshana | 15000 |
+-----+-----+-----+
2 rows in set (0.00 sec)
```

```
mysql> select * from tbl_employee where eid not in (101, 103, 106);
+-----+-----+-----+
| eid | ename | esalary |
+-----+-----+-----+
| 102 | Minion | 16000 |
| 104 | NULL | 17000 |
| 105 | Jenish | NULL |
+-----+-----+-----+
3 rows in set (0.00 sec)
```

- Select command using the between operator:

It does not include the NULL value.

```
mysql> select * from tbl_employee where esalary between 12000 and 16000;
+-----+-----+-----+
| eid | ename | esalary |
+-----+-----+-----+
| 101 | Dharshu | 12000 |
| 102 | Minion | 16000 |
| 103 | Dharshana | 15000 |
+-----+-----+-----+
3 rows in set (0.00 sec)
```

We can't specify the between operator in higher to lower.

```
mysql> select * from tbl_employee where esalary between 16000 and 12000;
Empty set (0.00 sec)
```

```
mysql> select * from tbl_employee where esalary not between 12000 and 16000;
+-----+-----+-----+
| eid | ename | esalary |
+-----+-----+-----+
| 104 | NULL | 17000 |
+-----+-----+-----+
1 row in set (0.00 sec)

mysql> select * from tbl_employee where esalary not between 12000 and 15000;
+-----+-----+-----+
| eid | ename | esalary |
+-----+-----+-----+
| 102 | Minion | 16000 |
| 104 | NULL | 17000 |
+-----+-----+-----+
2 rows in set (0.00 sec)
```

- Select command using to display the pattern (%-->0 or n characters, _ → Single character)

```
mysql> select * from tbl_employee where ename like 'D%';
+-----+-----+-----+
| eid | ename | esalary |
+-----+-----+-----+
| 101 | Dharshu | 12000 |
| 103 | Dharshana | 15000 |
+-----+-----+-----+
2 rows in set (0.00 sec)
```

- Like '_e%' → based on the second character it will display the record.

```
mysql> select * from tbl_employee where ename like '_e%';
+-----+-----+-----+
| eid | ename | esalary |
+-----+-----+-----+
| 105 | Jenish | NULL |
+-----+-----+-----+
1 row in set (0.00 sec)
```

Update Query:

There may be a requirement where the existing data in a MySQL table needs to be modified.

Syntax:

UPDATE table_name SET field1= new-value1,field2= new-value2
[WHERE Clause].

- You can update one or more fields altogether.
- You can specify any condition using the WHERE clause.
- You can update the values in a single table at a time.

The WHERE clause is very useful when you want to update the selected rows in a table.

- **Commit** —> will commit the transaction.
- **Rollback** —> will recommit the transaction.

```
mysql> select * from tbl_employee;
+-----+-----+-----+
| eid | ename   | esalary |
+-----+-----+-----+
| 101 | Dharshu | 12000   |
| 102 | Minion  | 16000   |
| 103 | Dharshana | 15000   |
| 104 | NULL    | 17000   |
| 105 | Jenish  | NULL    |
+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> commit;
Query OK, 0 rows affected (0.00 sec)

mysql> update tbl_employee set esalary = 0;
Query OK, 5 rows affected (0.00 sec)
Rows matched: 5  Changed: 5  Warnings: 0

mysql> select * from tbl_employee;
+-----+-----+-----+
| eid | ename   | esalary |
+-----+-----+-----+
| 101 | Dharshu | 0        |
| 102 | Minion  | 0        |
| 103 | Dharshana | 0        |
| 104 | NULL    | 0        |
| 105 | Jenish  | 0        |
+-----+-----+-----+
5 rows in set (0.00 sec)

mysql>
```

```
mysql> rollback;  
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> select * from tbl_employee;
```

eid	ename	esalary
101	Dharshu	0
102	Minion	0
103	Dharshana	0
104	NULL	0
105	Jenish	0

```
5 rows in set (0.00 sec)
```

```
mysql> ROLLBACK;  
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> select * from tbl_employee;
```

eid	ename	esalary
101	Dharshu	0
102	Minion	0
103	Dharshana	0
104	NULL	0
105	Jenish	0

```
5 rows in set (0.00 sec)
```

```
mysql> SET autocommit=0;  
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> select * from tbl_employee;
```

eid	ename	esalary
101	Dharshu	0
102	Minion	0
103	Dharshana	0
104	NULL	0
105	Jenish	0

```
5 rows in set (0.00 sec)
```

```
mysql> SELECT @@autocommit from dual;
```

```
+-----+
| @@autocommit |
+-----+
|           0 |
+-----+
```

```
1 row in set (0.00 sec)
```

```
mysql> select * from tbl_employee;
```

```
+-----+-----+-----+
| eid |  ename   | esalary |
+-----+-----+-----+
| 101 | Dharshu  |        0 |
| 102 | Minion   |        0 |
| 103 | Dharshana |        0 |
| 104 | NULL     |        0 |
| 105 | Jenish   |        0 |
+-----+-----+-----+
```

```
5 rows in set (0.00 sec)
```

```
mysql> update tbl_employee set esalary = 17000 where eid=101;
```

```
Query OK, 1 row affected (0.00 sec)
```

```
Rows matched: 1  Changed: 1  Warnings: 0
```

```
mysql> select * from tbl_employee;
```

```
+-----+-----+-----+
| eid |  ename   | esalary |
+-----+-----+-----+
| 101 | Dharshu  |    17000 |
| 102 | Minion   |        0 |
| 103 | Dharshana |        0 |
| 104 | NULL     |        0 |
| 105 | Jenish   |        0 |
+-----+-----+-----+
```

```
5 rows in set (0.00 sec)
```

```
mysql> rollback;
```

```
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> select * from tbl_employee;
```

```
+-----+-----+-----+
| eid |  ename   | esalary |
+-----+-----+-----+
| 101 | Dharshu  |        0 |
| 102 | Minion   |        0 |
| 103 | Dharshana |        0 |
| 104 | NULL     |        0 |
| 105 | Jenish   |        0 |
+-----+-----+-----+
```


- Update all the salary as 2000

```
mysql> update tbl_employee set esalary = 2000;
Query OK, 5 rows affected (0.00 sec)
Rows matched: 5  Changed: 5  Warnings: 0

mysql> select * from tbl_employee;
+-----+-----+-----+
| eid  | ename    | esalary |
+-----+-----+-----+
| 101  | Dharshu  | 2000    |
| 102  | Minion   | 2000    |
| 103  | Dharshana | 2000    |
| 104  | NULL     | 2000    |
| 105  | Jenish   | 2000    |
+-----+-----+-----+
5 rows in set (0.00 sec)

mysql>
```

Committing the records:

```
mysql> commit;
Query OK, 0 rows affected (0.00 sec)
```

Rollback the records:

```
mysql> rollback;
Query OK, 0 rows affected (0.00 sec)

mysql> select * from tbl_employee;
+-----+-----+-----+
| eid  | ename    | esalary |
+-----+-----+-----+
| 101  | Dharshu  | 2000    |
| 102  | Minion   | 2000    |
| 103  | Dharshana | 2000    |
| 104  | NULL     | 2000    |
| 105  | Jenish   | 2000    |
+-----+-----+-----+
5 rows in set (0.00 sec)
```

- Does not change anything because the records are already committed after committing the record if we modify anything and then we want to give rollback .

```
mysql> update tbl_employee set ename= null, esalary =0 where eid in(101,103,106);
Query OK, 2 rows affected (0.00 sec)
Rows matched: 2  Changed: 2  Warnings: 0

mysql> select * from tbl_employee;
+-----+-----+-----+
| eid | ename | esalary |
+-----+-----+-----+
| 101 | NULL | 0 |
| 102 | Minion | 2000 |
| 103 | NULL | 0 |
| 104 | NULL | 2000 |
| 105 | Jenish | 2000 |
+-----+-----+-----+
5 rows in set (0.00 sec)
```

Delete Query:

If you want to delete a record from an mysql table , then you can use the Sql command DELETE FROM .

Syntax:

DELETE FROM table_name [WHERE Clause]

- If the WHERE clause is not specified, then all the records will be deleted from the given MySql table.
- You can specify any condition using the WHERE clause.
- You can delete records in a single table at a time.

```
mysql> delete from tbl_employee;
Query OK, 5 rows affected (0.00 sec)

mysql> select * from tbl_employee;
Empty set (0.00 sec)

mysql> rollback;
Query OK, 0 rows affected (0.00 sec)

mysql> select * from tbl_employee;
+-----+-----+-----+
| eid   | ename   | esalary |
+-----+-----+-----+
| 101   | Dharshu | 2000    |
| 102   | Minion  | 2000    |
| 103   | Dharshana | 2000    |
| 104   | NULL    | 2000    |
| 105   | Jenish  | 2000    |
+-----+-----+-----+
5 rows in set (0.00 sec)
```

- To delete a particular record .

```
mysql> delete from tbl_employee where eid=104;
Query OK, 1 row affected (0.00 sec)

mysql> select * from tbl_employee;
+-----+-----+-----+
| eid   | ename   | esalary |
+-----+-----+-----+
| 101   | Dharshu | 2000    |
| 102   | Minion  | 2000    |
| 103   | Dharshana | 2000    |
| 105   | Jenish  | 2000    |
+-----+-----+-----+
4 rows in set (0.00 sec)
```

Difference Between Delete and Truncate:

Delete is a DML command so the commit and rollback is worked.

Truncate is a DDL command so the commit and rollback does not work.

Like Clause:

- We have seen the SQL SELECT command to fetch data from the MySQL table. We can also use a conditional clause called the WHERE clause to select the required records.
- A WHERE clause with the 'equal to' sign(=) works fine where we want to do an exact match. Like if "employee_name = 'Minion' ". But there may be a requirement where we want to filter out all the results where the employee_name should contain "ion". This can be handled by using SQL LIKE Clause along with the WHERE Clause.
- If SQL LIKE clause is used along with the % character, then it will work like a meta character(*) as in UNIX, while listing out of all the files or directories at the command prompt. Without a % character, the LIKE clause is very same as the equal to sign along with the WHERE Clause.

Syntax:

```
SELECT field1,field2,...fieldN table_name1,table_name2,....  
WHERE field1 LIKE condition1 [AND [OR]] field2= 'Somevalue'
```

- You can specify any condition using the WHERE Clause.
- You can use the LIKE clause along with the WHERE Clause.
- You can use the LIKE clause in place of the **Equals to sign**.
- When LIKE is used along with % sign then it will work like a meta character search.
- You can specify more than one condition using **AND** or **OR** Operators.
- A WHERELIKE clauses can be used along with DELETE or UPDATE SQL commands also to specify a condition.

Sorting Results:

- We have seen the SQL SELECT command to fetch data from MySQL table. when you select rows, the MySQL server is free to return them in any order, unless you instruct it otherwise by saying how to sort the result.
- But, you sort a result set by adding an **ORDER BY** clause that names the column or columns which you want to sort.

Syntax:

```
SELECT field1,field2,...fieldN table_name1,table_name2,....
```

ORDER BY field1, [field2...] [ASC [DESC]]

- You can sort the returned result on any field, if that field is being listed out.
 - You can sort the result on more than one field.
 - You can use the keyword ASC or DESC to get the result in ascending or descending order. **By default, it's the ascending order.**
 - You can use the WHERELIKE clause in the usual way to put a condition.
-
- Ascending (By Default):

```
mysql> select * from tbl_employee order by eid;
+-----+-----+-----+
| eid |  ename  | esalary |
+-----+-----+-----+
| 101 | Dharshu |    2000 |
| 102 | Minion  |    2000 |
| 103 | Dharshana |    2000 |
| 105 | Jenish  |    2000 |
+-----+-----+-----+
4 rows in set (0.00 sec)
```

- Descending:

```
mysql> select * from tbl_employee order by eid desc;
+-----+-----+-----+
| eid |  ename  | esalary |
+-----+-----+-----+
| 105 | Jenish  |    2000 |
| 103 | Dharshana |    2000 |
| 102 | Minion  |    2000 |
| 101 | Dharshu |    2000 |
+-----+-----+-----+
4 rows in set (0.00 sec)
```

- Name in Descending order:

```
mysql> select * from tbl_employee order by ename desc;
+-----+-----+-----+
| eid | ename   | esalary |
+-----+-----+-----+
| 102 | Minion  | 2000    |
| 105 | Jenish  | 2000    |
| 101 | Dharshu | 2000    |
| 103 | Dharshana | 2000    |
+-----+-----+-----+
4 rows in set (0.00 sec)
```

MySQL NULL Values:

We have seen the **SQL SELECT** Command along with the **WHERE** clause to fetch data from a MySQL table, but when we try to give a condition , which compares the field or the column value to **NULL**, it does not work properly.

To handle such a situation , MySQL provides three operators:

- **IS NULL** — This operator returns true, if the column value is NULL.
- **IS NOT NULL** — This operator returns true, if the column value is not NULL.
- **< = >** — This operator compares values, which (unlike the = operator) is true even for two NULL values.

The conditions involving NULL are special. You cannot use **=NULL** or **!=NULL** to look for NULL values in columns. Such comparisons always fail because it is impossible to tell whether they are true or not. Sometimes, even **NULL = NULL** fails.

To look for columns that are or are not NULL , use **IS NULL** or **IS NOT NULL**.