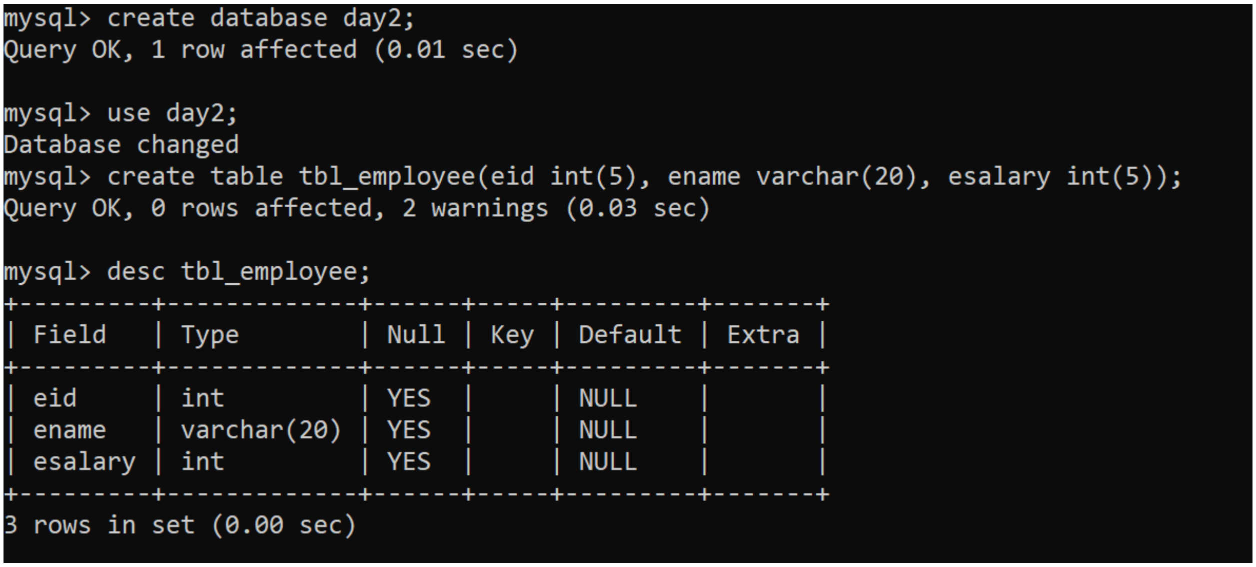
Name: Naveenraj Karthikeyan
Emp id: 11958

Mysql Day 2: 29-09-2023

Mysql Queries:

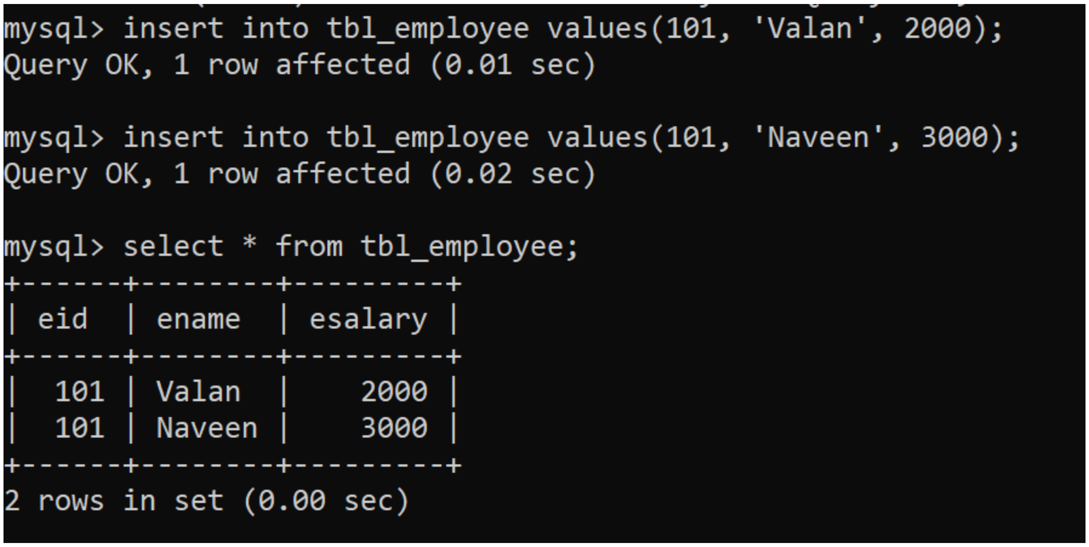
Insert Query:

1) Create table tbl_employee(eid int(5), ename varchar(20), esalary int(5));
2) Desc tbl_employee;



```
mysql> create database day2;
Query OK, 1 row affected (0.01 sec)

mysql> use day2;
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> 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)
```

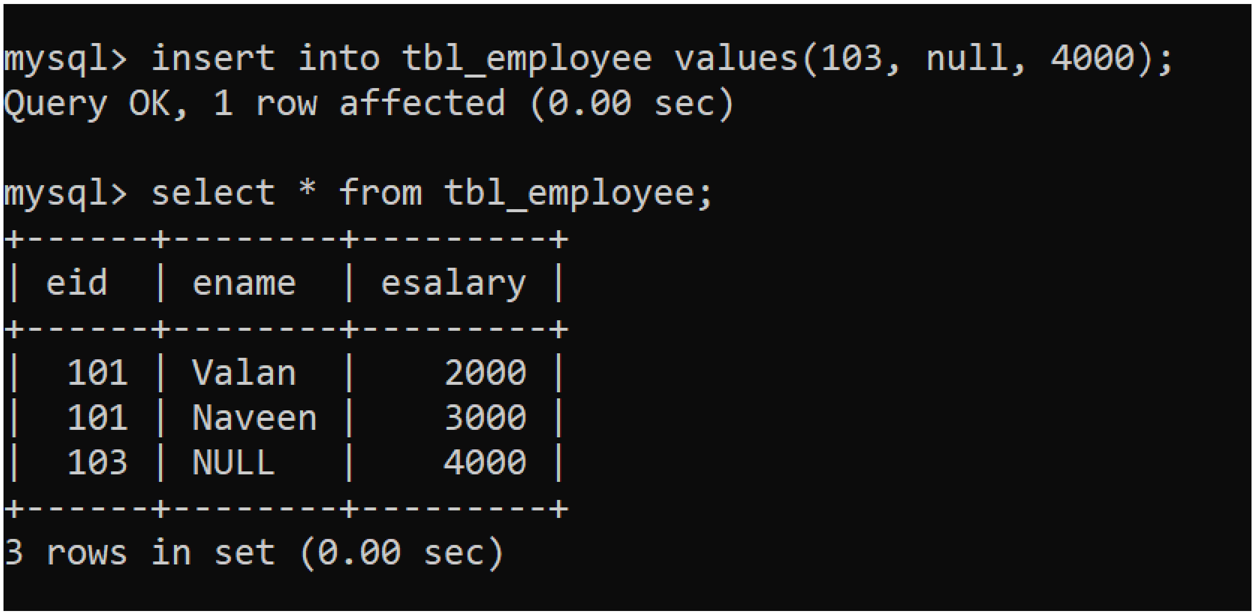
3) Insert into tbl_employee values(101, ’Valan’, 2000);
4) Insert into tbl_employee values(102, ’Naveen’, 3000);
5) Select * from tbl_employee;



```
mysql> insert into tbl_employee values(101, 'Valan', 2000);
Query OK, 1 row affected (0.01 sec)

mysql> insert into tbl_employee values(101, 'Naveen', 3000);
Query OK, 1 row affected (0.02 sec)

mysql> select * from tbl_employee;
+------+--------+---------+
| eid  | ename  | esalary |
+------+--------+---------+
|  101 | Valan  |    2000 |
|  101 | Naveen |    3000 |
+------+--------+---------+
2 rows in set (0.00 sec)
```

6) insert into tbl_employee values(103, null, 4000);

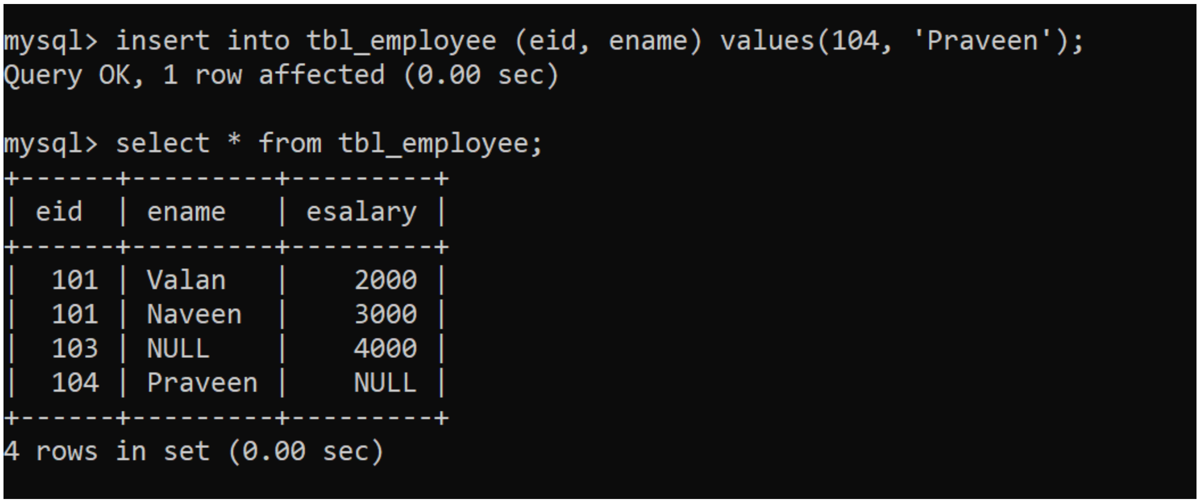
You can pass null values using implicitly or explicitly.



```
mysql> insert into tbl_employee values(103, null, 4000);
Query OK, 1 row affected (0.00 sec)

mysql> select * from tbl_employee;
+------+--------+---------+
| eid  | ename  | esalary |
+------+--------+---------+
|  101 | Valan  |    2000 |
|  101 | Naveen |    3000 |
|  103 | NULL   |    4000 |
+------+--------+---------+
3 rows in set (0.00 sec)
```

7) Insert into tbl_employee (eid, ename) values(104, ‘Praveen’);

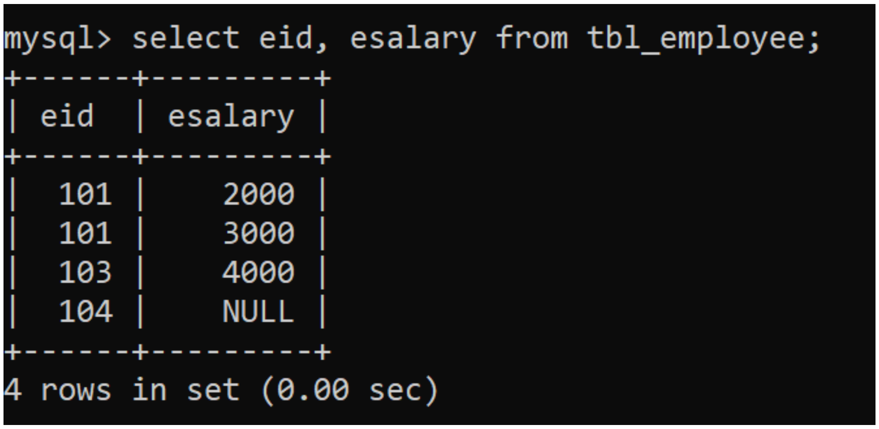
You can pass null values using implicitly or explicitly.



```
mysql> insert into tbl_employee (eid, ename) values(104, 'Praveen');
Query OK, 1 row affected (0.00 sec)

mysql> select * from tbl_employee;
+------+---------+---------+
| eid  | ename   | esalary |
+------+---------+---------+
|  101 | Valan   |    2000 |
|  101 | Naveen  |    3000 |
|  103 | NULL    |    4000 |
|  104 | Praveen |    NULL |
+------+---------+---------+
4 rows in set (0.00 sec)
```

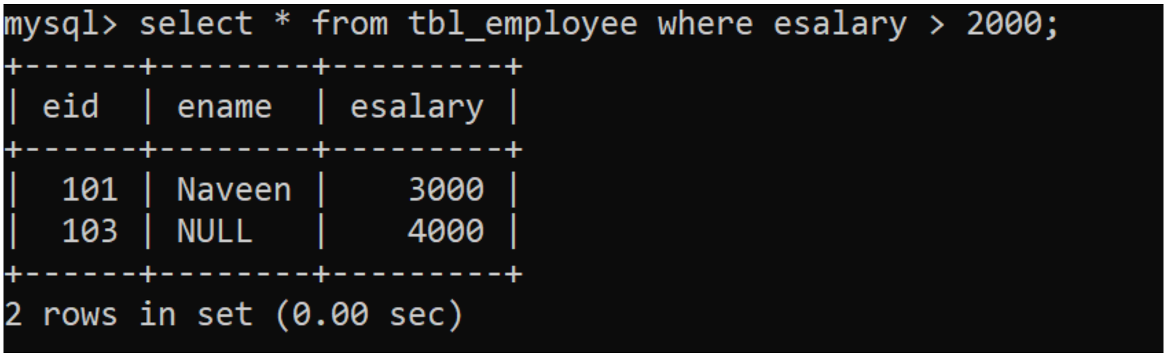
Select Query:

1) select eid, esalary from tbl_employee;



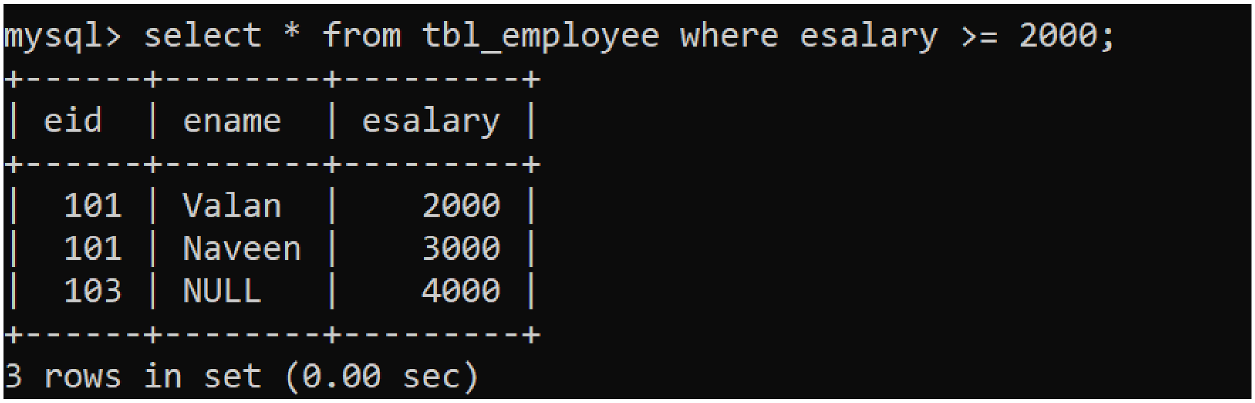
```
mysql> select eid, esalary from tbl_employee;
+------+---------+
| eid  | esalary |
+------+---------+
|  101 |    2000 |
|  101 |    3000 |
|  103 |    4000 |
|  104 |    NULL |
+------+---------+
4 rows in set (0.00 sec)
```

2) select * from tbl_employee where esalary > 2000;



```
mysql> select * from tbl_employee where esalary > 2000;
+------+--------+---------+
| eid  | ename  | esalary |
+------+--------+---------+
|  101 | Naveen |    3000 |
|  103 | NULL   |    4000 |
+------+--------+---------+
2 rows in set (0.00 sec)
```

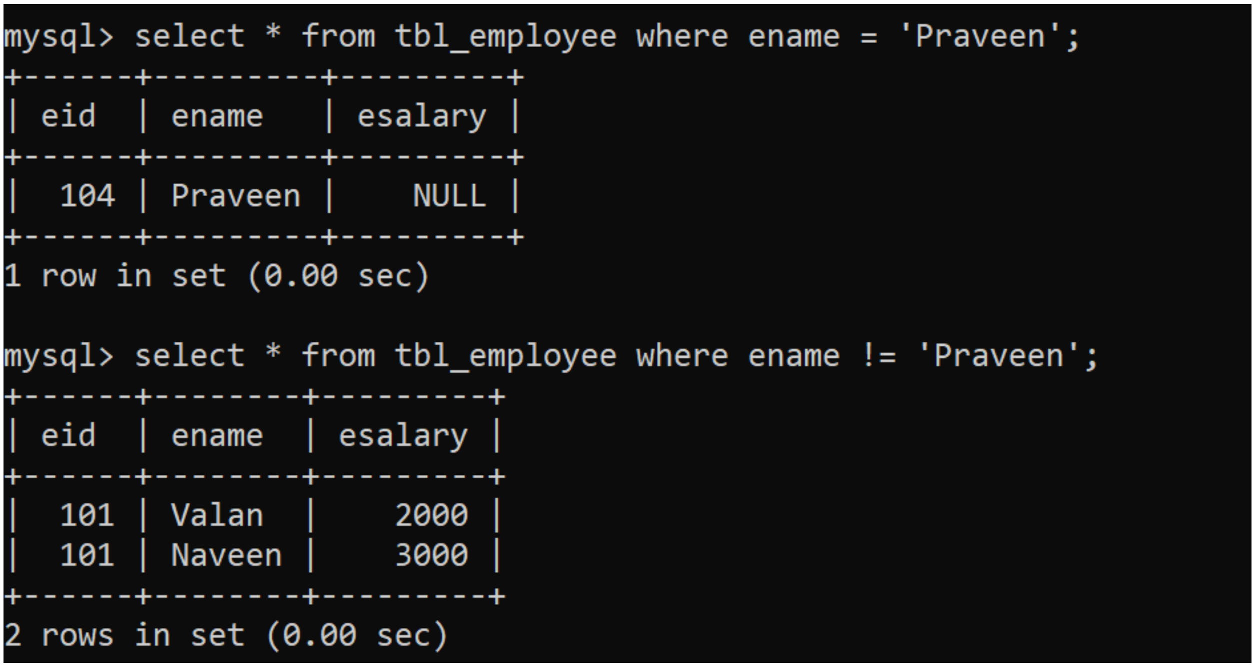
3) select * from tbl_employee where esalary >= 2000;



```
mysql> select * from tbl_employee where esalary >= 2000;
+------+--------+---------+
| eid  | ename  | esalary |
+------+--------+---------+
|  101 | Valan  |    2000 |
|  101 | Naveen |    3000 |
|  103 | NULL   |    4000 |
+------+--------+---------+
3 rows in set (0.00 sec)
```

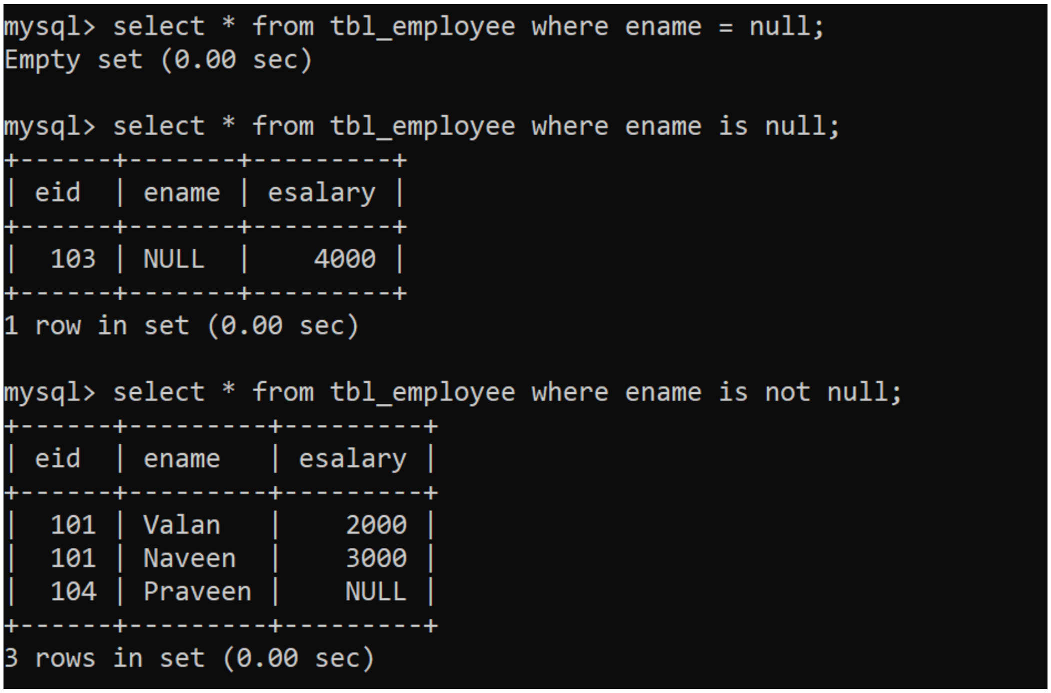
4) select * from tbl_employee where ename = 'Praveen';
5) select * from tbl_employee where ename != 'Praveen';

By using relational operators you cannot compare null values.
There are separate operators for Null values in sql. (IS NULL, IS NOT NULL)



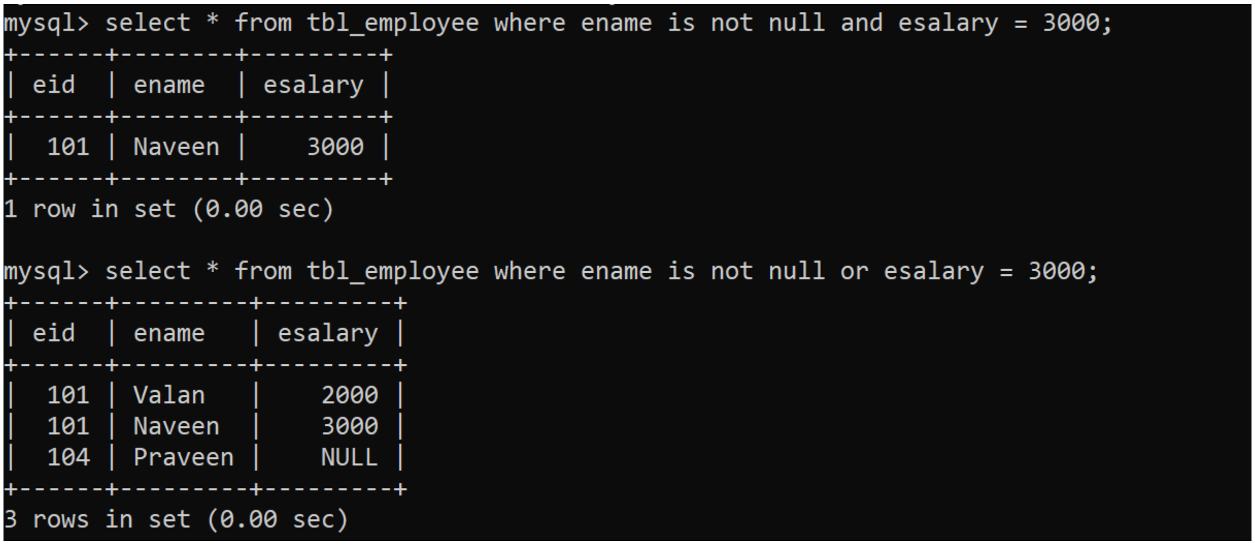
```
mysql> select * from tbl_employee where ename = 'Praveen';
+------+---------+---------+
| eid  | ename   | esalary |
+------+---------+---------+
|  104 | Praveen |    NULL |
+------+---------+---------+
1 row in set (0.00 sec)

mysql> select * from tbl_employee where ename != 'Praveen';
+------+--------+---------+
| eid  | ename  | esalary |
+------+--------+---------+
|  101 | Valan  |    2000 |
|  101 | Naveen |    3000 |
+------+--------+---------+
2 rows in set (0.00 sec)
```

6) select * from tbl_employee where ename is null;
7) select * from tbl_employee where ename is not null;



```
mysql> select * from tbl_employee where ename = null;
Empty set (0.00 sec)

mysql> select * from tbl_employee where ename is null;
+------+-------+---------+
| eid  | ename | esalary |
+------+-------+---------+
|  103 | NULL  |    4000 |
+------+-------+---------+
1 row in set (0.00 sec)

mysql> select * from tbl_employee where ename is not null;
+------+---------+---------+
| eid  | ename   | esalary |
+------+---------+---------+
|  101 | Valan   |    2000 |
|  101 | Naveen  |    3000 |
|  104 | Praveen |    NULL |
+------+---------+---------+
3 rows in set (0.00 sec)
```

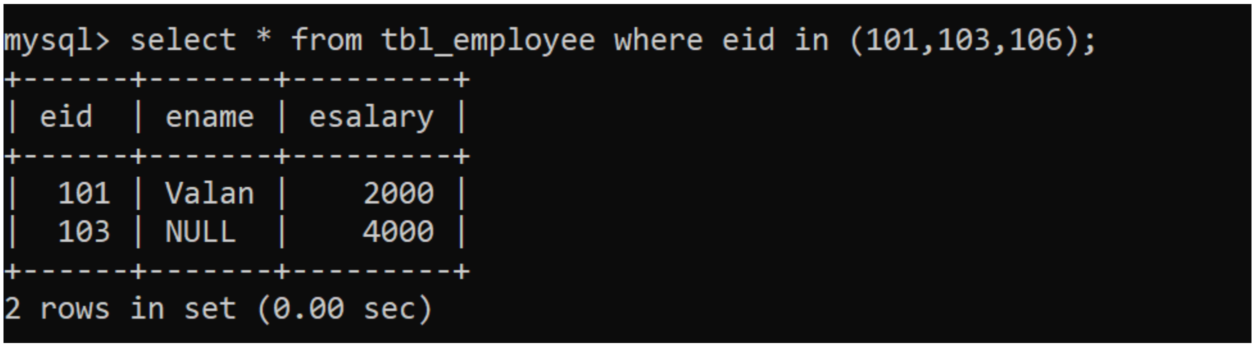
8) select * from tbl_employee where ename **is not null** and esalary = 3000;
9) select * from tbl_employee where ename **is not null** or esalary = 3000;



```
mysql> select * from tbl_employee where ename is not null and esalary = 3000;
+------+--------+---------+
| eid  | ename  | esalary |
+------+--------+---------+
|  101 | Naveen |    3000 |
+------+--------+---------+
1 row in set (0.00 sec)

mysql> select * from tbl_employee where ename is not null or esalary = 3000;
+------+---------+---------+
| eid  | ename   | esalary |
+------+---------+---------+
|  101 | Valan   |    2000 |
|  101 | Naveen  |    3000 |
|  104 | Praveen |    NULL |
+------+---------+---------+
3 rows in set (0.00 sec)
```

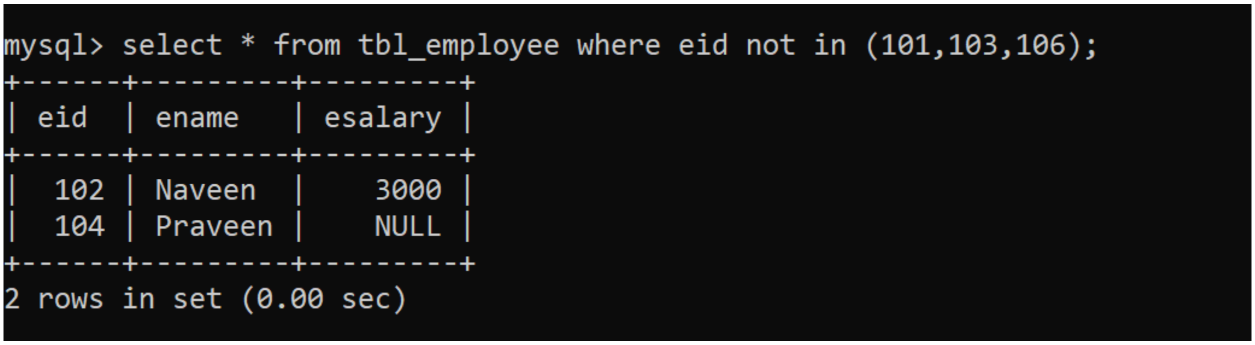
10) select * from tbl_employee where eid **in** (101,103,106);

Using an operator we are able to list the employees which are in the table These are the sql comparison or relational operator.



```
mysql> select * from tbl_employee where eid in (101,103,106);
+------+-------+---------+
| eid  | ename | esalary |
+------+-------+---------+
|  101 | Valan |    2000 |
|  103 | NULL  |    4000 |
+------+-------+---------+
2 rows in set (0.00 sec)
```

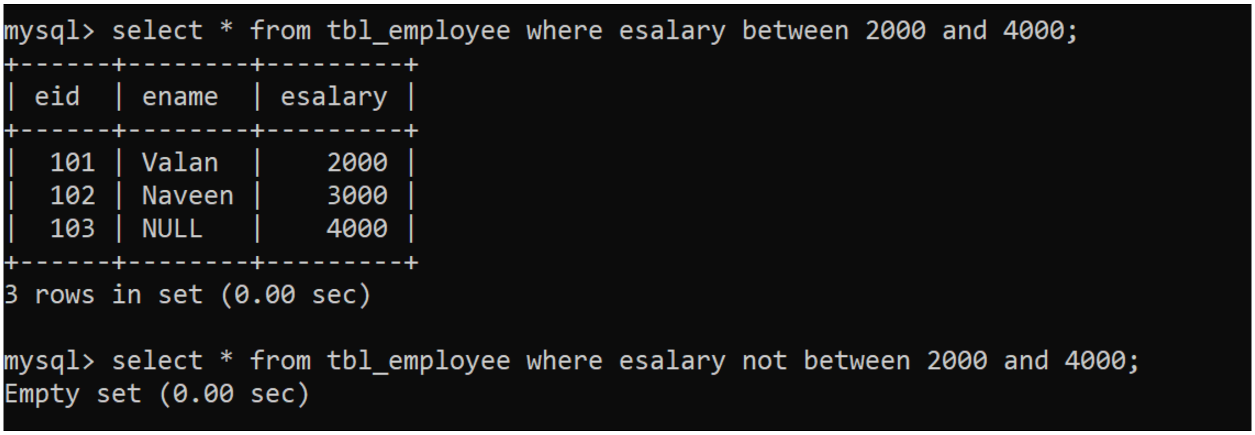
11) select * from tbl_employee where eid **not in** (101,103,106);



```
mysql> select * from tbl_employee where eid not in (101,103,106);
+------+---------+---------+
| eid  | ename   | esalary |
+------+---------+---------+
|  102 | Naveen  |    3000 |
|  104 | Praveen |    NULL |
+------+---------+---------+
2 rows in set (0.00 sec)
```

12) select * from tbl_employee where esalary **between** 2000 and 4000;
13) select * from tbl_employee where esalary **not between** 2000 and 4000;

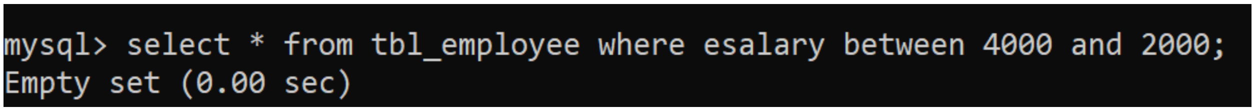
It will include lower limit and upper limit and also not including the null value.



```
mysql> select * from tbl_employee where esalary between 2000 and 4000;
+------+--------+---------+
| eid  | ename  | esalary |
+------+--------+---------+
|  101 | Valan  |    2000 |
|  102 | Naveen |    3000 |
|  103 | NULL   |    4000 |
+------+--------+---------+
3 rows in set (0.00 sec)

mysql> select * from tbl_employee where esalary not between 2000 and 4000;
Empty set (0.00 sec)
```

14) select * from tbl_employee where esalary between 4000 and 2000;

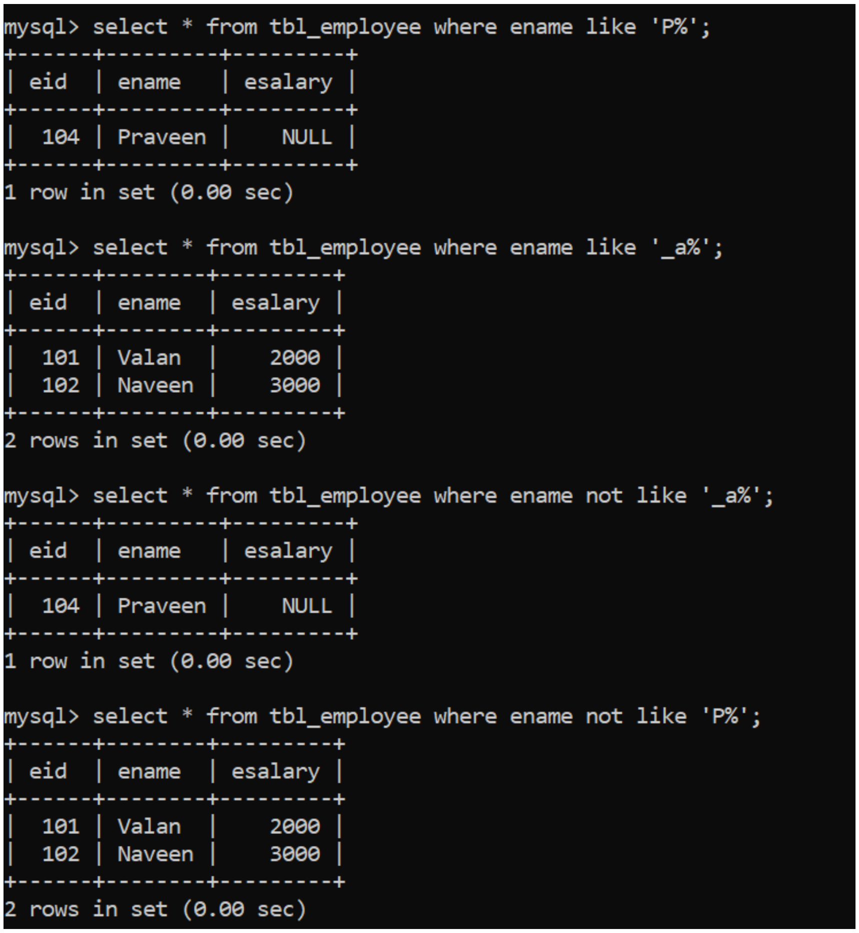
It will not show anything it assumes as a negative value.



```
mysql> select * from tbl_employee where esalary between 4000 and 2000;
Empty set (0.00 sec)
```

15) select * from tbl_employee where ename like 'P%';
16) select * from tbl_employee where ename like '_a%';
17) select * from tbl_employee where ename not like '_a%';
18) select * from tbl_employee where ename not like 'P%';

% indicates zero or more characters. Likewise _ shows anything(_) and after ‘a’ and % gives the output. You can also apply not operator here.



```
mysql> select * from tbl_employee where ename like 'P%';
+------+---------+---------+
| eid  | ename   | esalary |
+------+---------+---------+
|  104 | Praveen |    NULL |
+------+---------+---------+
1 row in set (0.00 sec)

mysql> select * from tbl_employee where ename like '_a%';
+------+--------+---------+
| eid  | ename  | esalary |
+------+--------+---------+
|  101 | Valan  |    2000 |
|  102 | Naveen |    3000 |
+------+--------+---------+
2 rows in set (0.00 sec)

mysql> select * from tbl_employee where ename not like '_a%';
+------+---------+---------+
| eid  | ename   | esalary |
+------+---------+---------+
|  104 | Praveen |    NULL |
+------+---------+---------+
1 row in set (0.00 sec)

mysql> select * from tbl_employee where ename not like 'P%';
+------+--------+---------+
| eid  | ename  | esalary |
+------+--------+---------+
|  101 | Valan  |    2000 |
|  102 | Naveen |    3000 |
+------+--------+---------+
2 rows in set (0.00 sec)
```

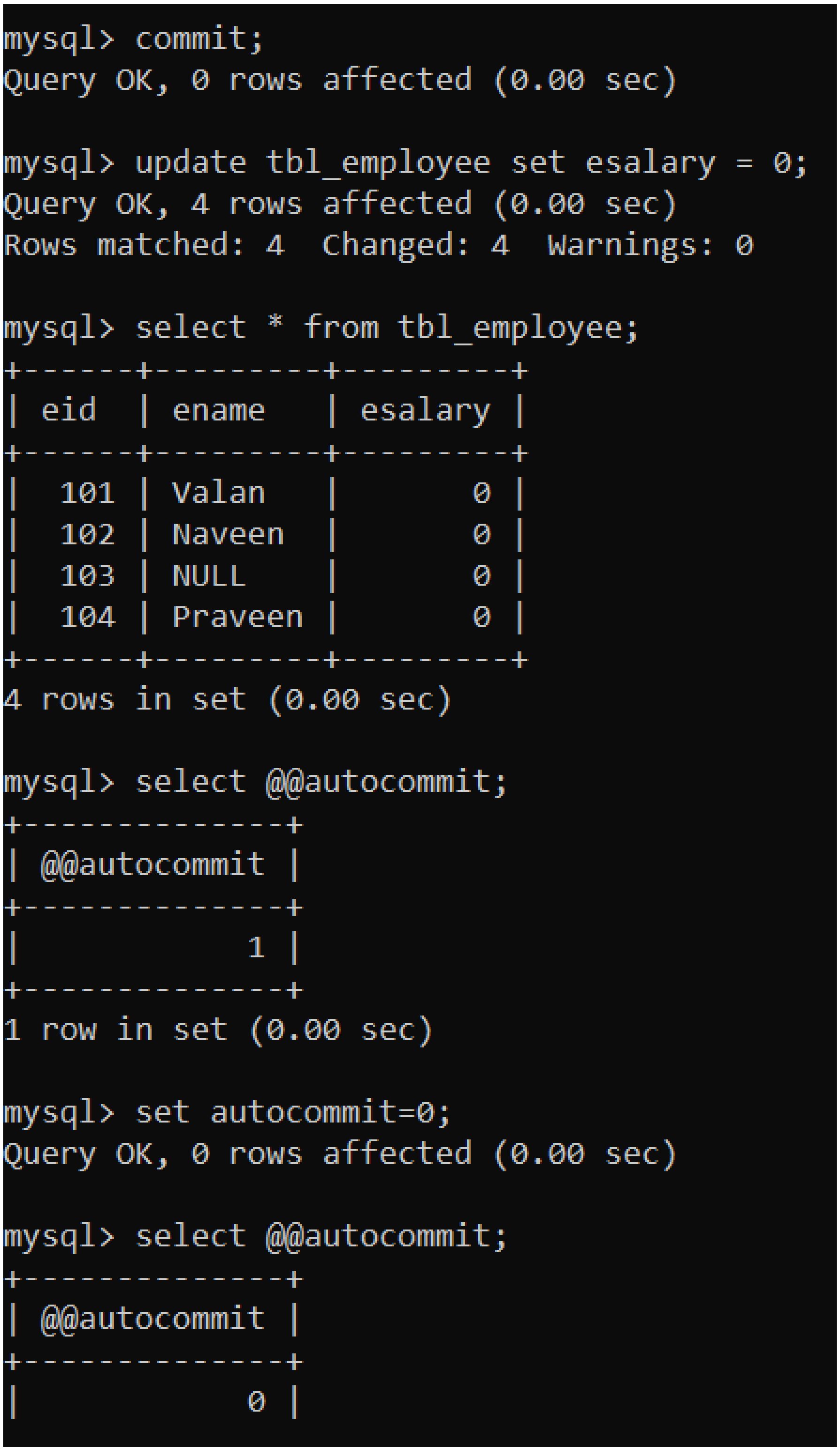
Update Query:

Manual commit command —> commit;

1) Update tbl_employee set esalary = 0;

To check auto commit —--> select @@autocommit;

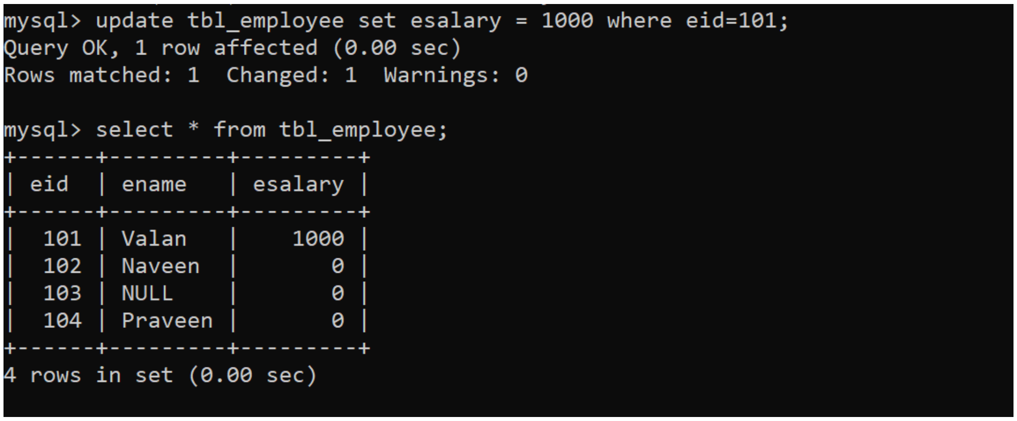
Disable —> set autocommit=0;
@@ means it is a global variable or environmental variable.



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

mysql> update tbl_employee set esalary = 0;
Query OK, 4 rows affected (0.00 sec)
Rows matched: 4  Changed: 4  Warnings: 0

mysql> select * from tbl_employee;
+------+---------+---------+
| eid  | ename   | esalary |
+------+---------+---------+
|  101 | Valan   |       0 |
|  102 | Naveen  |       0 |
|  103 | NULL    |       0 |
|  104 | Praveen |       0 |
+------+---------+---------+
4 rows in set (0.00 sec)

mysql> select @@autocommit;
+--------------+
| @@autocommit |
+--------------+
|            1 |
+--------------+
1 row in set (0.00 sec)

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

mysql> select @@autocommit;
+--------------+
| @@autocommit |
+--------------+
|            0 |
```

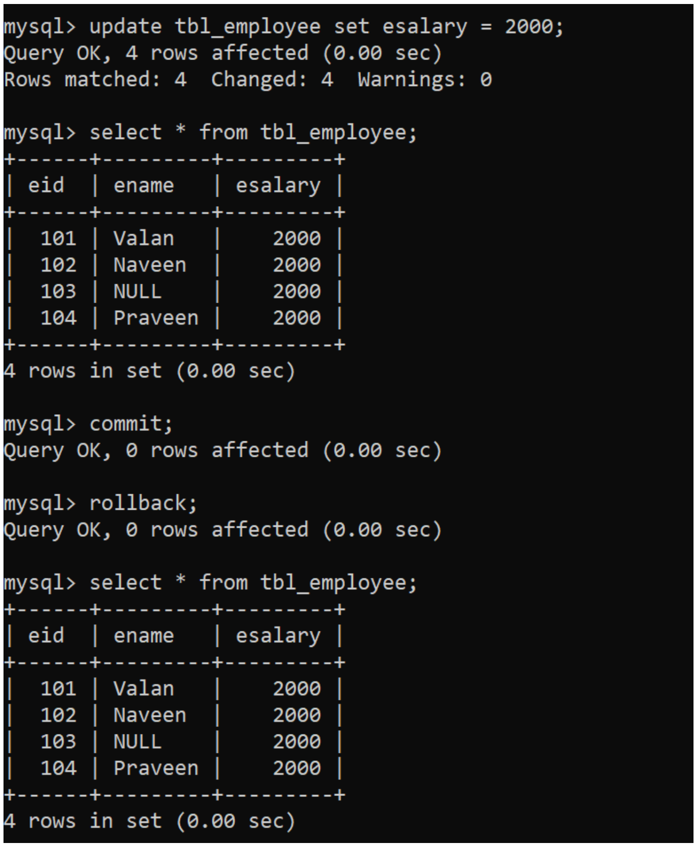
1) Update tbl_employee set esalary = 1000 where eid=101;
2) Rollback;

Whenever you are using ddl commands from that it will auto commit automatically up to before the creation of a new ddl query.



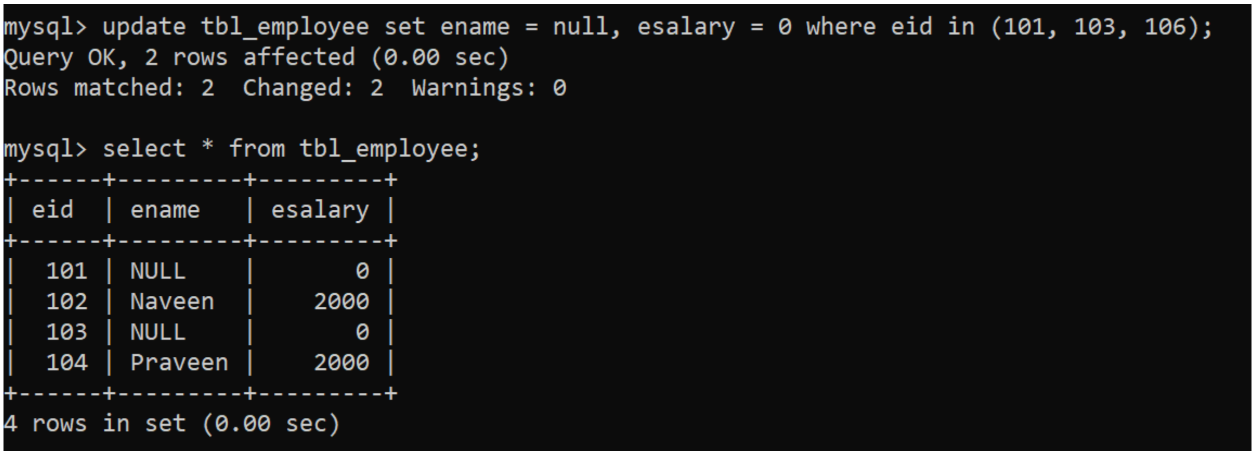
```
mysql> update tbl_employee set esalary = 1000 where eid=101;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from tbl_employee;
+------+---------+---------+
| eid  | ename   | esalary |
+------+---------+---------+
|  101 | Valan   |    1000 |
|  102 | Naveen  |       0 |
|  103 | NULL    |       0 |
|  104 | Praveen |       0 |
+------+---------+---------+
4 rows in set (0.00 sec)
```

3) Update tbl_employee set esalary = 2000;
4) Commit;
5) rollback;



```
mysql> update tbl_employee set esalary = 2000;
Query OK, 4 rows affected (0.00 sec)
Rows matched: 4  Changed: 4  Warnings: 0

mysql> select * from tbl_employee;
+------+---------+---------+
| eid  | ename   | esalary |
+------+---------+---------+
|  101 | Valan   |    2000 |
|  102 | Naveen  |    2000 |
|  103 | NULL    |    2000 |
|  104 | Praveen |    2000 |
+------+---------+---------+
4 rows in set (0.00 sec)

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

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

mysql> select * from tbl_employee;
+------+---------+---------+
| eid  | ename   | esalary |
+------+---------+---------+
|  101 | Valan   |    2000 |
|  102 | Naveen  |    2000 |
|  103 | NULL    |    2000 |
|  104 | Praveen |    2000 |
+------+---------+---------+
4 rows in set (0.00 sec)
```

6) Update tbl_employee set ename = null, esalary = 0 where eid in (101, 103, 106);

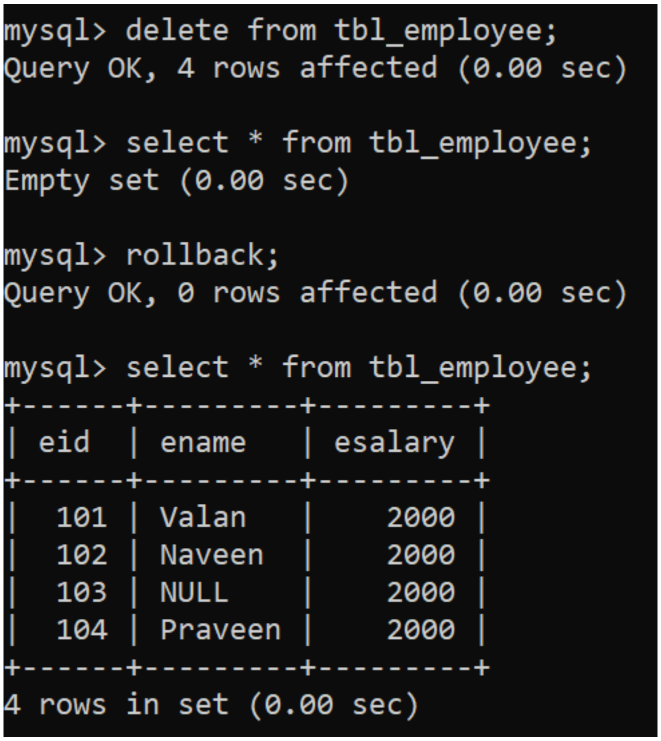


```
mysql> update tbl_employee set ename = null, esalary = 0 where eid in (101, 103, 106);
Query OK, 2 rows affected (0.00 sec)
Rows matched: 2  Changed: 2  Warnings: 0

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

Delete Query:

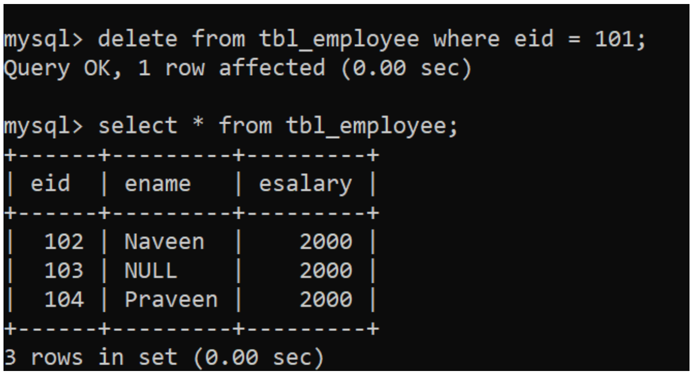
1) **Delete from tbl_employee;**

Truncate and delete are the same; but in delete you have the where keyword. Truncate does not have a where keyword. Truncate values cant get back because it is a ddl command. But delete we can get back the values using rollback.



```
mysql> delete from tbl_employee;
Query OK, 4 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 | Valan   |    2000 |
|  102 | Naveen  |    2000 |
|  103 | NULL    |    2000 |
|  104 | Praveen |    2000 |
+------+---------+---------+
4 rows in set (0.00 sec)
```

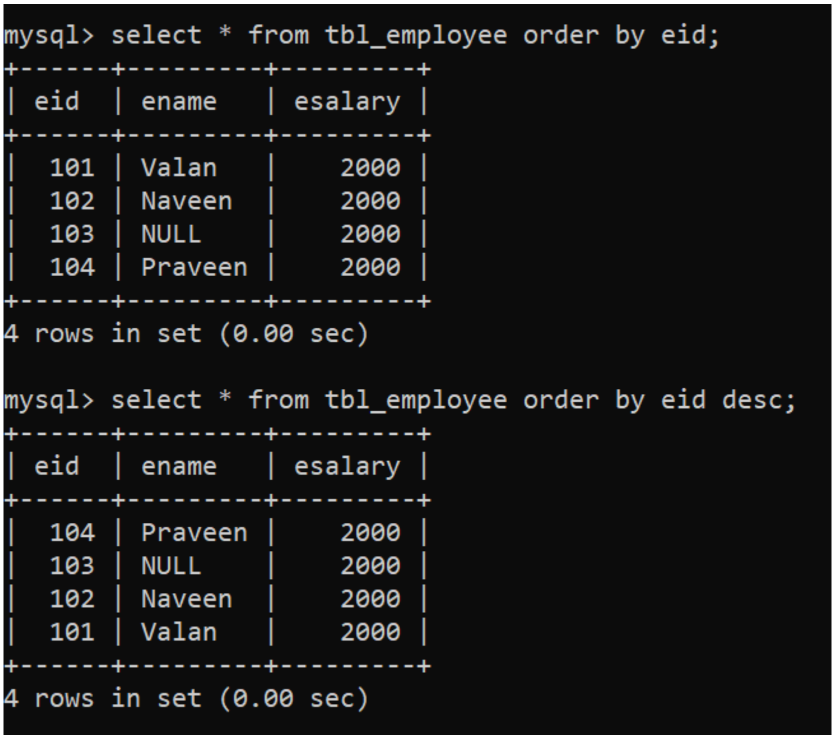
2) Delete from tbl_employee where eid = 101;



```
mysql> delete from tbl_employee where eid = 101;
Query OK, 1 row affected (0.00 sec)

mysql> select * from tbl_employee;
+------+---------+---------+
| eid  | ename   | esalary |
+------+---------+---------+
|  102 | Naveen  |    2000 |
|  103 | NULL    |    2000 |
|  104 | Praveen |    2000 |
+------+---------+---------+
3 rows in set (0.00 sec)
```

Sorting query:

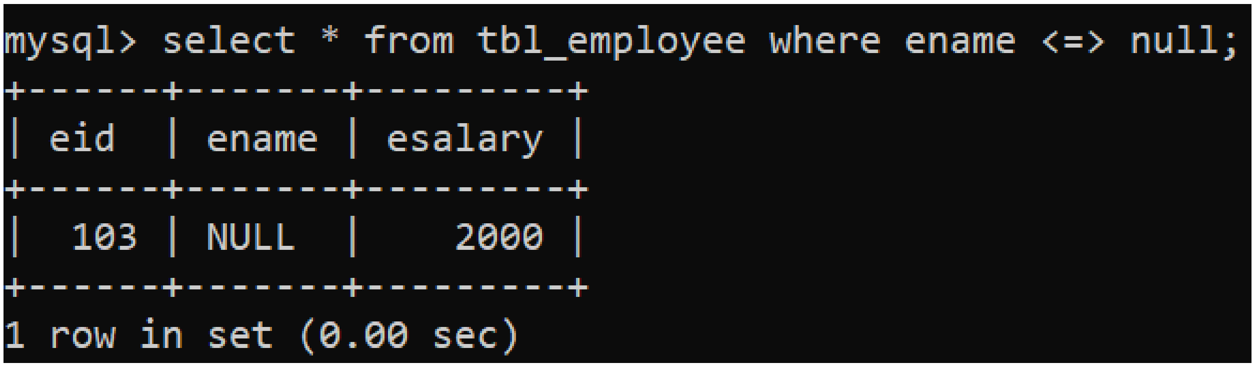
1) select * from tbl_employee order by eid;
2) select * from tbl_employee order by eid desc;



```
mysql> select * from tbl_employee order by eid;
+------+---------+---------+
| eid  | ename   | esalary |
+------+---------+---------+
|  101 | Valan   |    2000 |
|  102 | Naveen  |    2000 |
|  103 | NULL    |    2000 |
|  104 | Praveen |    2000 |
+------+---------+---------+
4 rows in set (0.00 sec)

mysql> select * from tbl_employee order by eid desc;
+------+---------+---------+
| eid  | ename   | esalary |
+------+---------+---------+
|  104 | Praveen |    2000 |
|  103 | NULL    |    2000 |
|  102 | Naveen  |    2000 |
|  101 | Valan   |    2000 |
+------+---------+---------+
4 rows in set (0.00 sec)
```

Null values:

select * from tbl_employee where ename <=> null;



```
mysql> select * from tbl_employee where ename <=> null;
+------+-------+---------+
| eid  | ename | esalary |
+------+-------+---------+
|  103 | NULL  |    2000 |
+------+-------+---------+
1 row in set (0.00 sec)
```