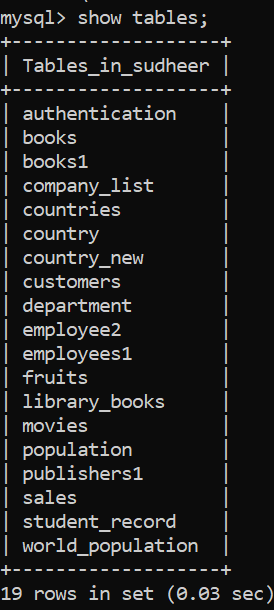
**MySQL practice**

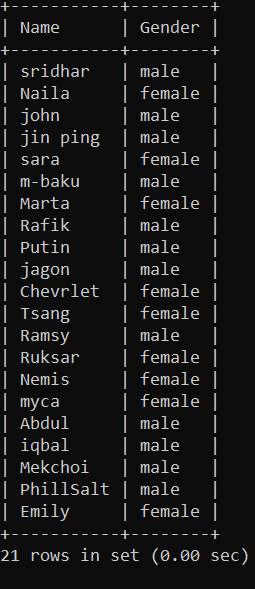


Select statement in MySQL is used to select data from Database(tables).

Syntax: select column1,column2 ------column n from table name.

Eg : 1) select Name, Gender from population …. ……… ……..[that we have created in CLI ] it shows only columns that we asked for in whole table.

mysql> select Name,Gender from Population;



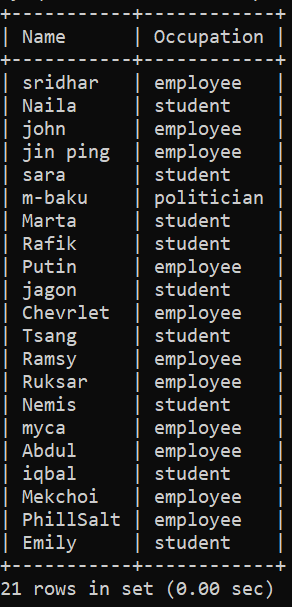
2) mysql> select Age,Qualification from Population;



3) mysql> select Country\_Name,Population from Population;



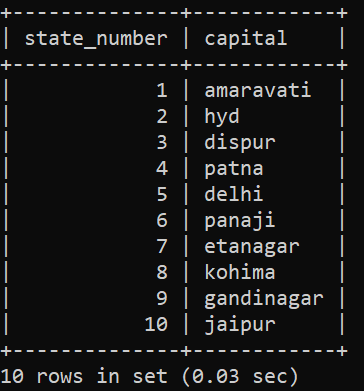
4) mysql> select Name,Occupation from Population;



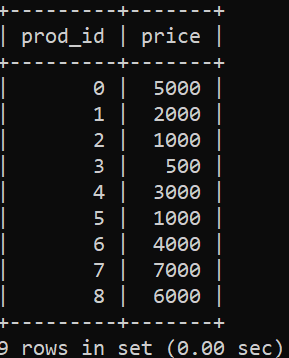
5) mysql> select Gender,Country\_Name from Population;



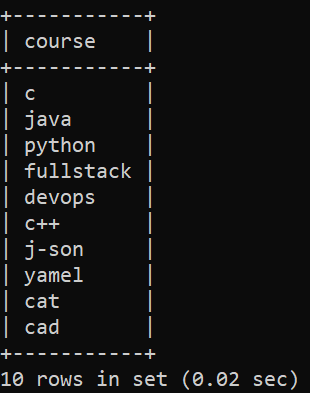
6) mysql> select state\_number,capital from states;



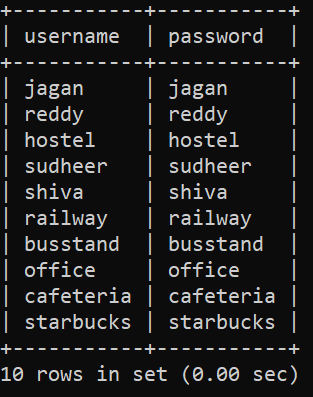
7) mysql> select prod\_id,price from inventory;



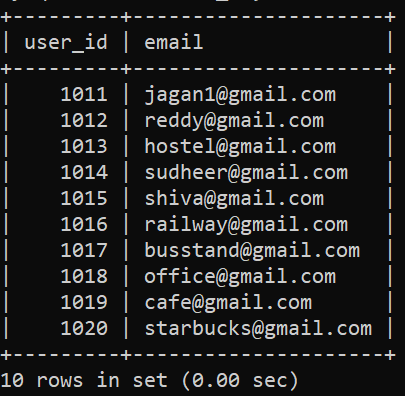
8) mysql> select course from student\_record;



9) mysql> select username,password from Authentication;



10) mysql> select user\_id,email from Authentication;

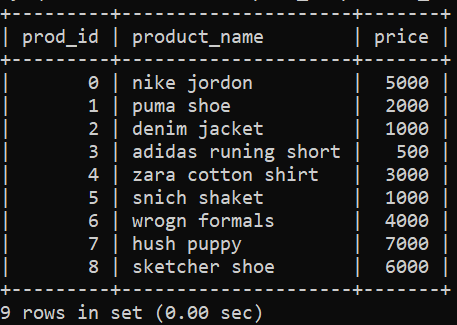


Distinct select:

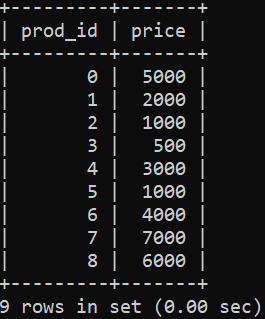
1) mysql> select distinct Name,Country\_Name from Population;

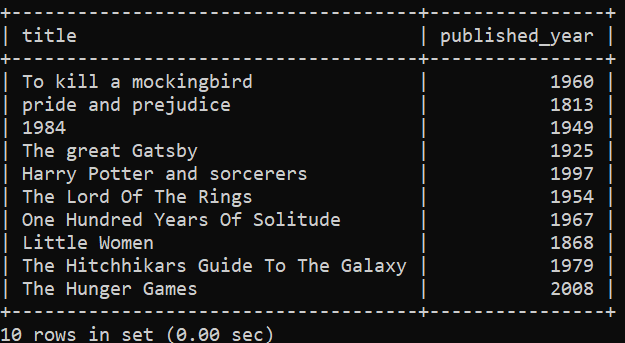


2) mysql> select distinct prod\_id,product\_name,price from inventory;



3) mysql> select distinct prod\_id,price from inventory;



4) mysql> select distinct title,published\_year from Library\_books ;

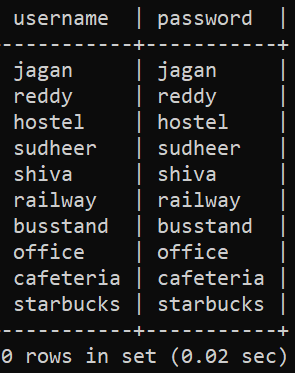
5) mysql> select distinct name,course from student\_record;



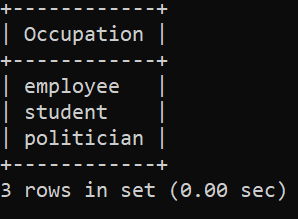
6) mysql> select distinct population from Population;

 here out of 21 we got 19,bcs distinct selection won’t allows duplicate.

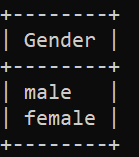
7) mysql> select distinct username,password from Authentication;



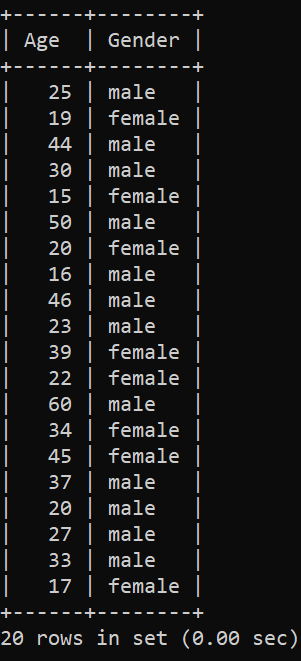
8) mysql> select distinct Occupation from Population;



9) mysql> select distinct Gender from Population;

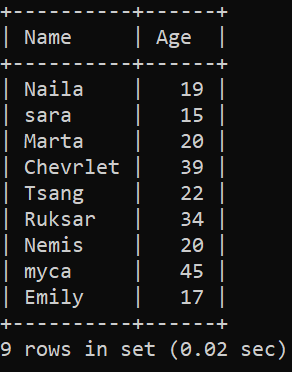


10) mysql> select distinct Age,Gender from Population;

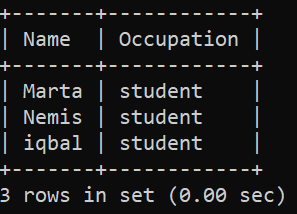


Where clause :

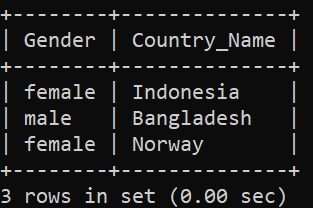
1. mysql> select Name,Age from Population where Gender='female';



1. mysql> select Name,Occupation from Population where Age=20;



1. mysql> select Gender,Country\_Name from Population where Qualification='school';



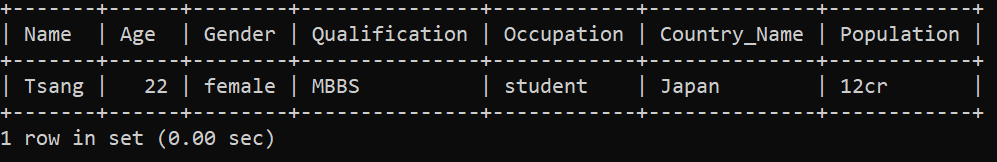
1. mysql> select Name,Population from Population where Occupation='employee';



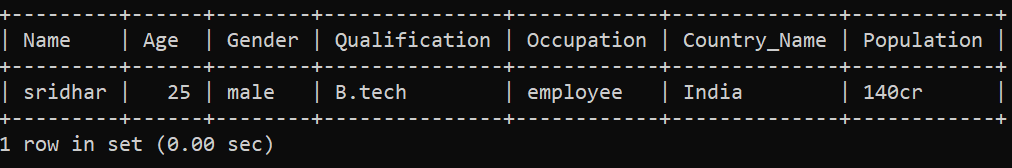
1. mysql> select Country\_Name from Population where Gender='male';



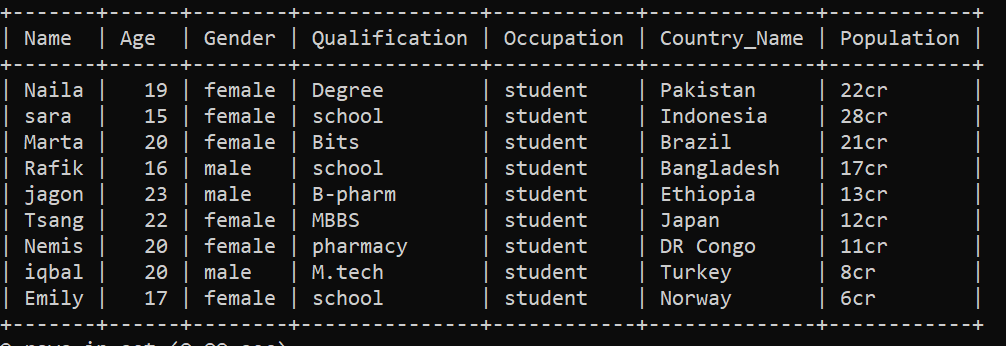
1. mysql> select \* from Population where Age=22;



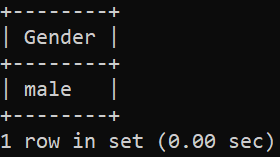
1. mysql> select \* from Population where Population=140;



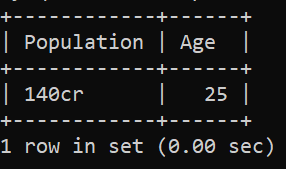
1. mysql> select \* from Population where Occupation='student';



1. mysql> select Gender from Population where Age=50;

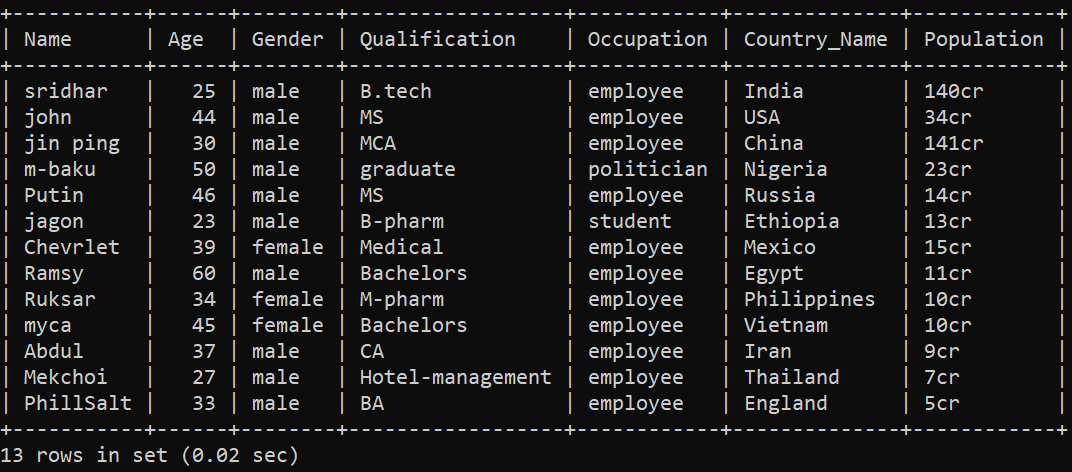


1. mysql> select Population,Age from Population where Age=25;



* we can use this clause for Arithmetic, between, like, In and Logical operators also.
* For grater than :

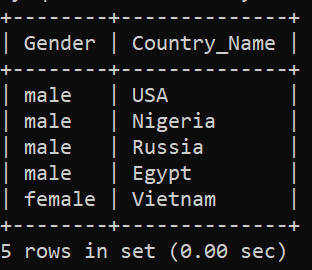
1. mysql> select \* from Population where Age>22;



1. mysql> select Name,Population from Population where Age>22;



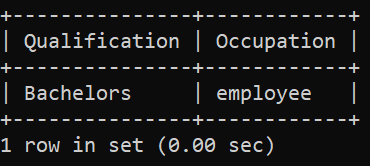
1. mysql> select Gender,Country\_Name from Population where Age>40;



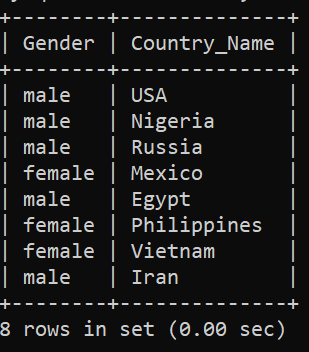
1. mysql> select Country\_Name from Population where Qualification>'school';

Empty set (0.00 sec)

1. mysql> select Qualification,Occupation from Population where Age>50;



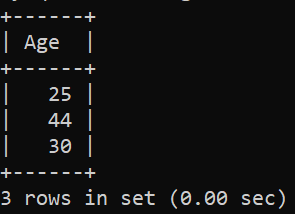
1. mysql> select Gender,Country\_Name from Population where Age>33;



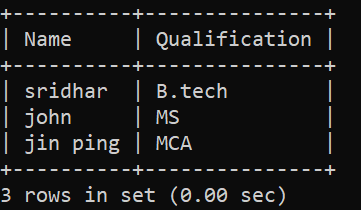
1. mysql> select Gender,Country\_Name from Population where Age>15;



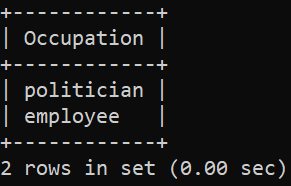
1. mysql> select Age from Population where Population>28;



1. mysql> select Name,Qualification from Population where Population>28;

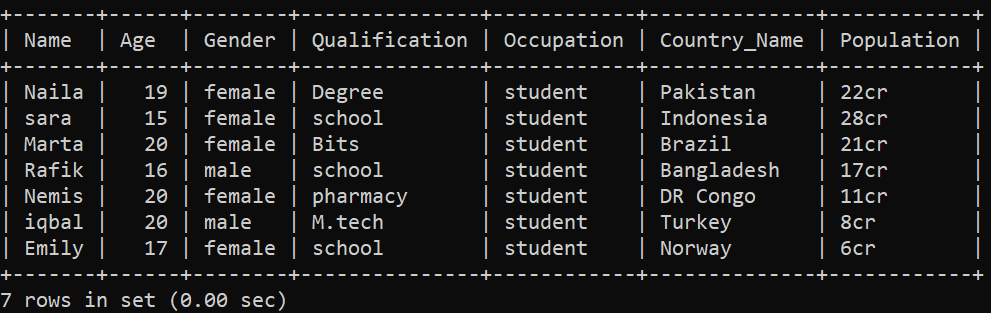


1. mysql> select Occupation from Population where Age>46;

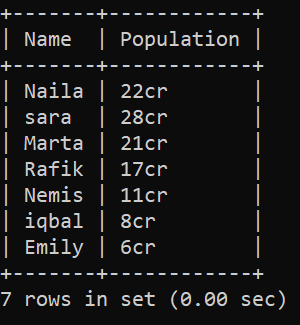


* for less than:

1. mysql> select \* from Population where Age<22;



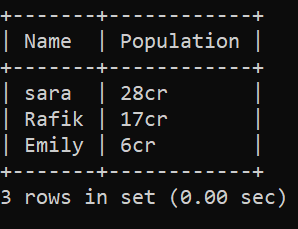
1. mysql> select Name,Population from Population where Age<22;



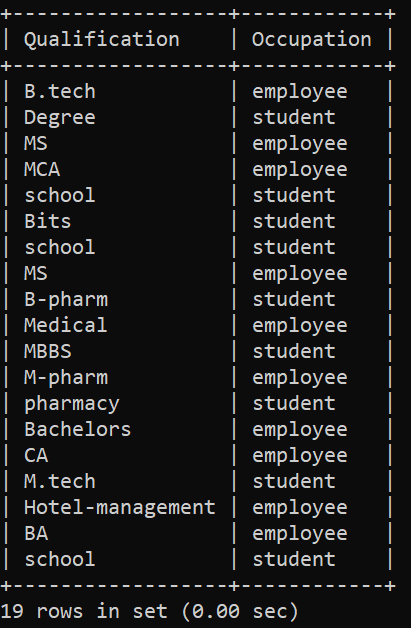
1. mysql> select Gender,Country\_Name from Population where Age<40;



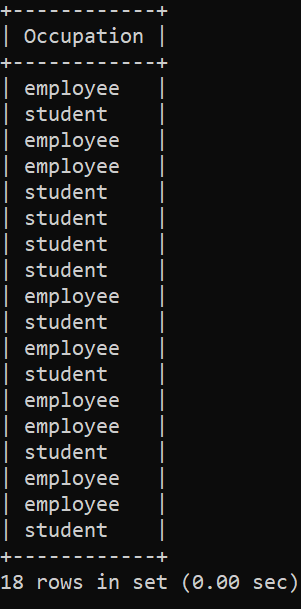
1. mysql> select Name,Population from Population where Age<18;



1. mysql> select Qualification,Occupation from Population where Age<50;



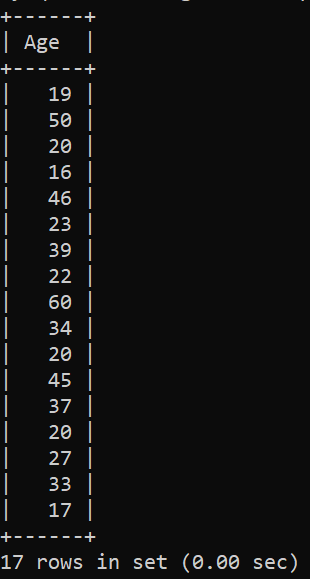
1. mysql> select Occupation from Population where Age<46;



1. mysql> select Name,Qualification from Population where Population<28;



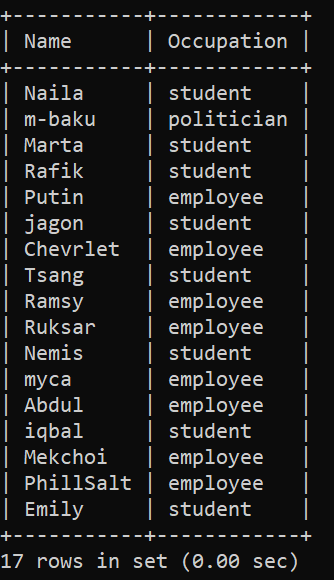
1. mysql> select Age from Population where Population<28;



1. mysql> select Gender,Country\_Name from Population where Age<33;

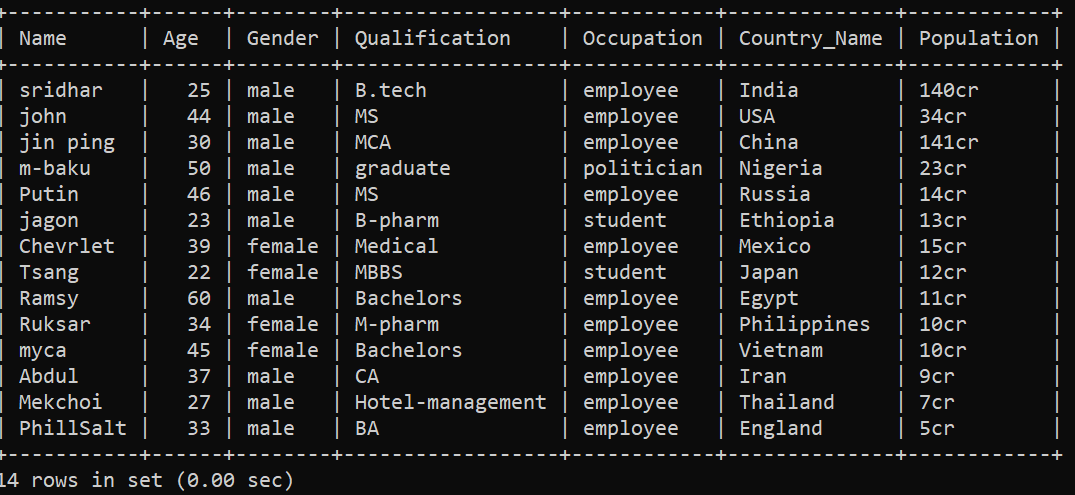


1. mysql> select Name,Occupation from Population where Population<28;



* for grater than equal to:

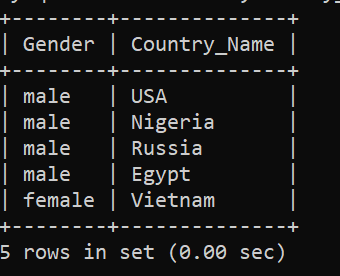
1. mysql> select \* from Population where Age>=22;



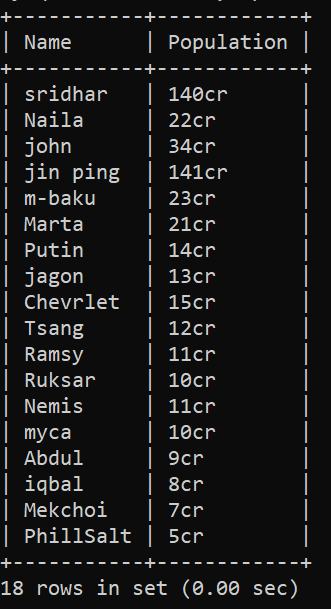
1. mysql> select Name,Population from Population where Age>=22;



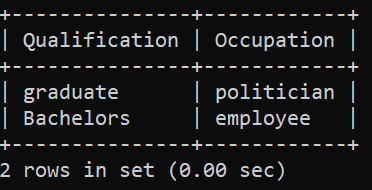
1. mysql> select Gender,Country\_Name from Population where Age>=40;



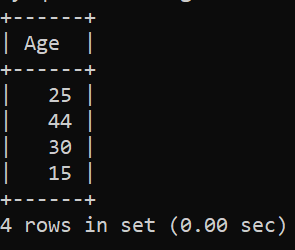
1. mysql> select Name,Population from Population where Age>=18;



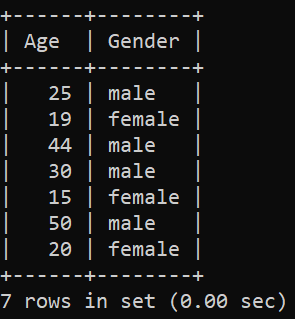
1. mysql> select Qualification,Occupation from Population where Age>=50;



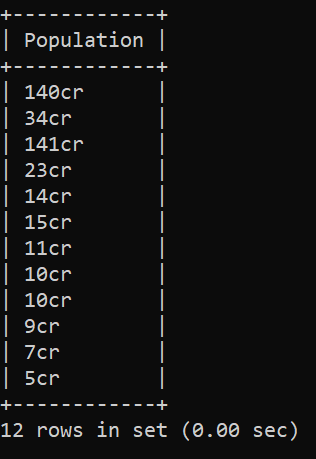
1. mysql> select Age from Population where Population>=25;



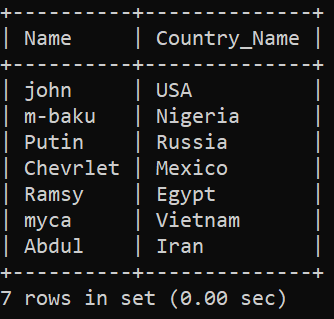
1. mysql> select Age,Gender from Population where Population>=18;



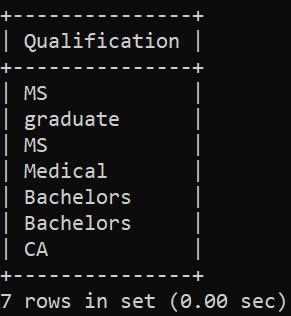
1. mysql> select Population from Population where Age>=25;



1. mysql> select Name,Country\_Name from Population where Age>=35;

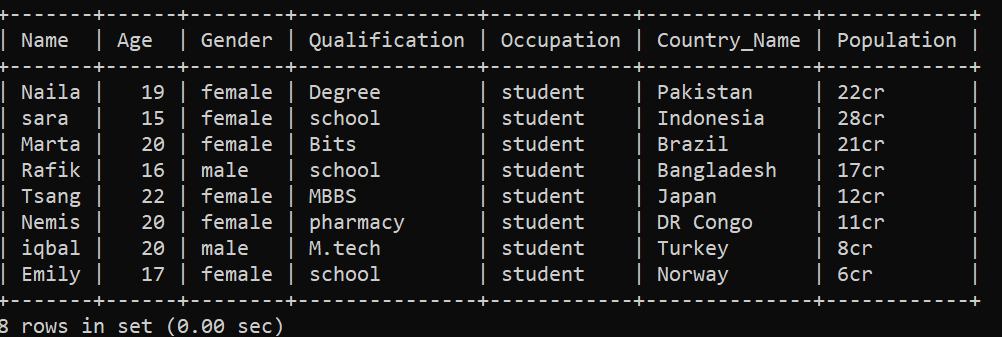


1. mysql> select Qualification from Population where Age>=35;



* less than equal to:

1. mysql> select \* from Population where Age<=22;



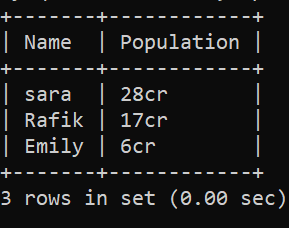
1. mysql> select Name,Population from Population where Age<=22;



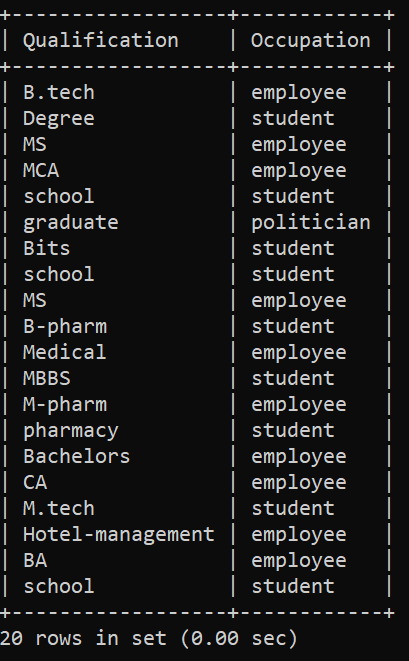
1. mysql> select Gender,Country\_Name from Population where Age<=40;



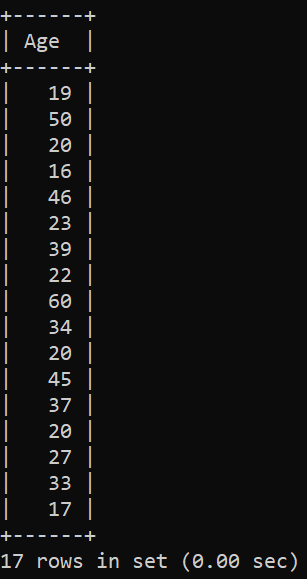
1. mysql> select Name,Population from Population where Age<=18;



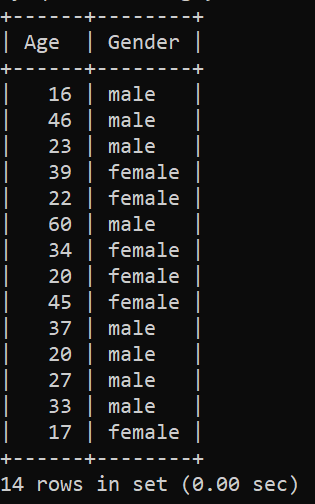
1. mysql> select Qualification,Occupation from Population where Age<=50;



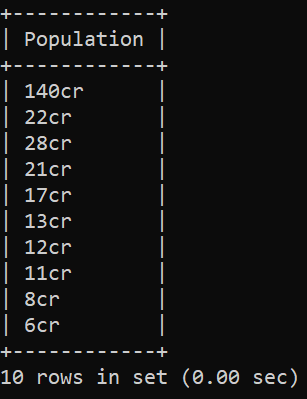
1. mysql> select Age from Population where Population<=25;



1. mysql> select Age,Gender from Population where Population<=18;



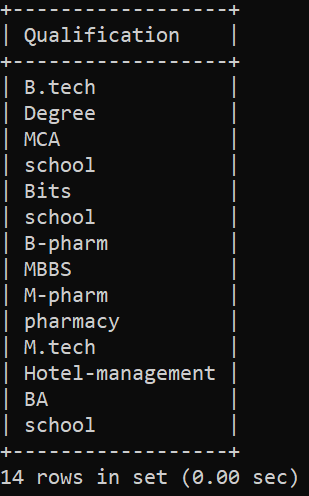
1. mysql> select Population from Population where Age<=25;



1. mysql> select Name,Country\_Name from Population where Age<=35;

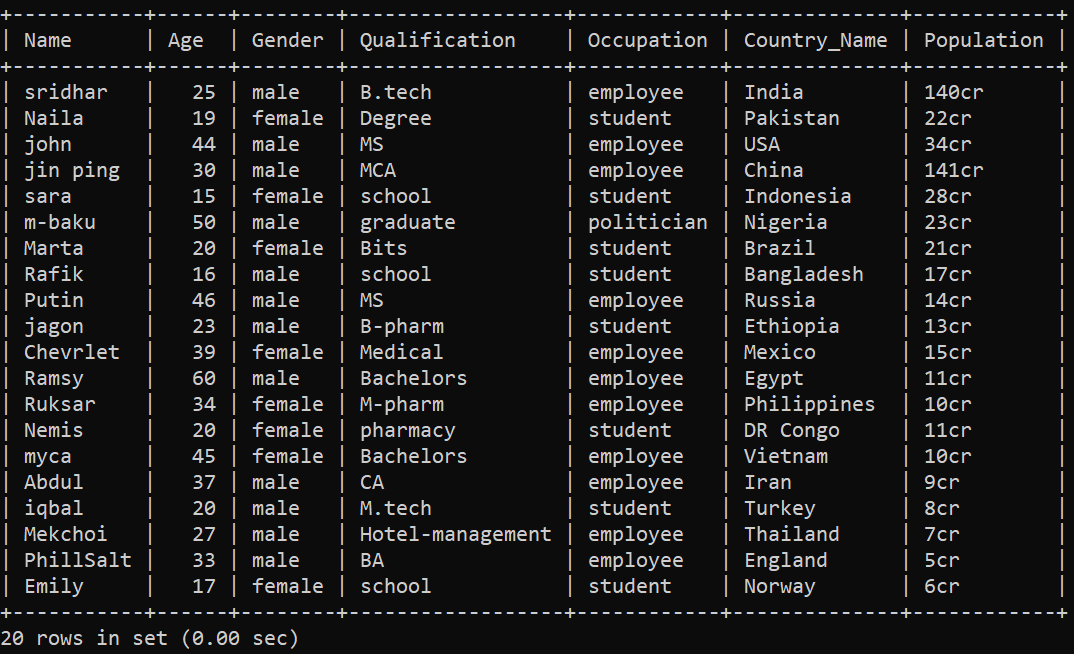


1. mysql> select Qualification from Population where Age<=35;



* not equal to:

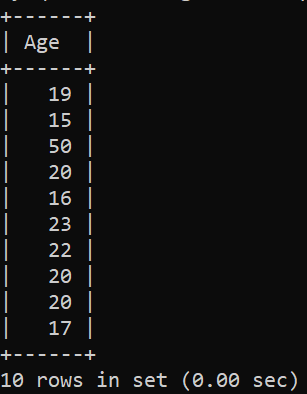
1. mysql> select \* from Population where Age!=22;



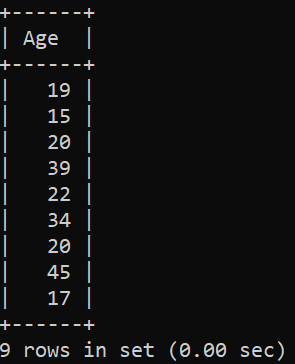
1. mysql> select Name,Population from Population where Age!=22;



1. mysql> select Age from Population where Occupation!='employee';



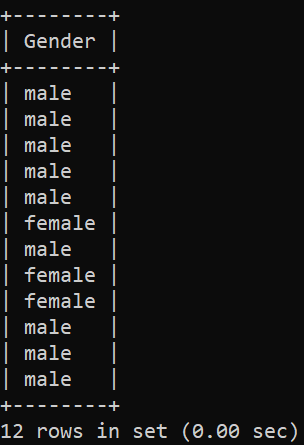
1. mysql> select Age from Population where Gender!='male';



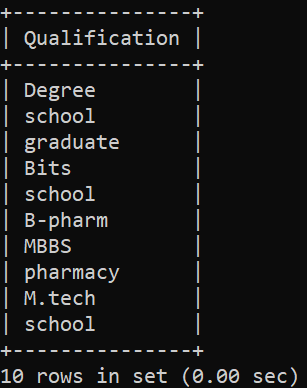
1. mysql> select Country\_Name,Population from Population where Gender!='female';



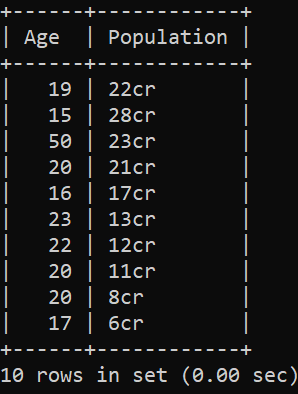
1. mysql> select Gender from Population where Occupation !='student';



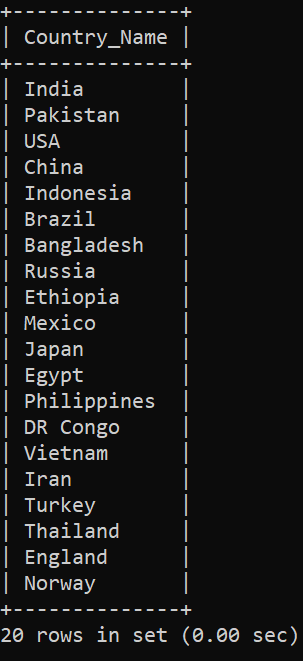
1. mysql> select Qualification from Population where Occupation !='employee';



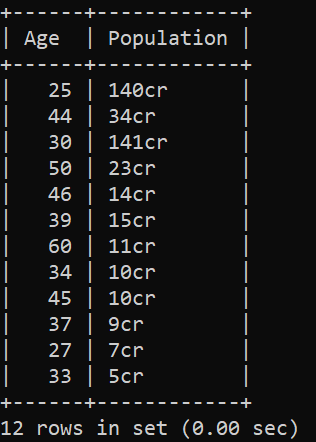
1. mysql> select Age,Population from Population where Occupation !='employee';



1. mysql> select Country\_Name from Population where Age!=50;

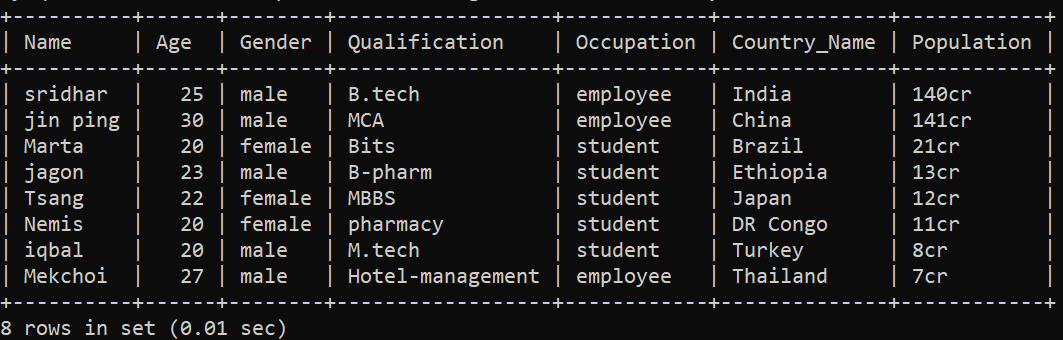


1. mysql> select Age,Population from Population where Occupation !='student';

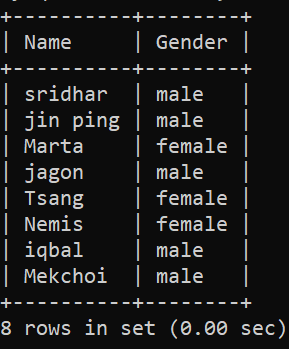


* for Between:

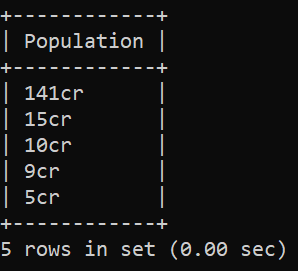
1. mysql> select \* from Population where Age between 20 AND 30;



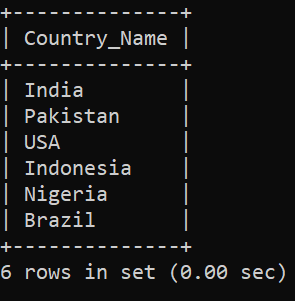
1. mysql> select Name,Gender from Population where Age between 20 AND 30;



1. mysql> select Population from Population where Age between 30 AND 40;



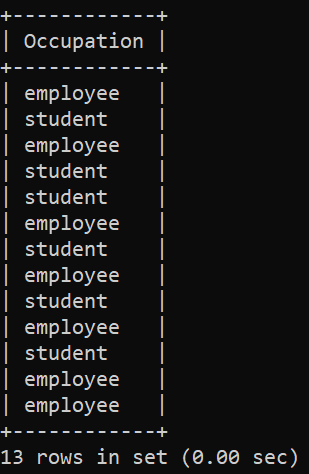
1. mysql> select Country\_Name from Population where Population between 20 AND 140;



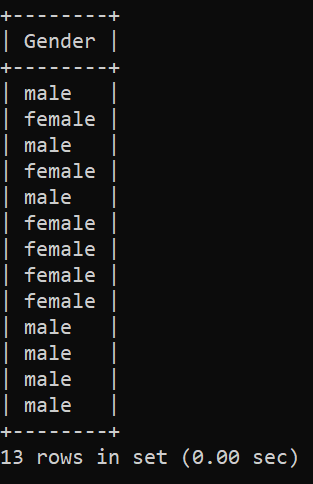
1. mysql> select Country\_Name from Population where Age between 15 AND 40;



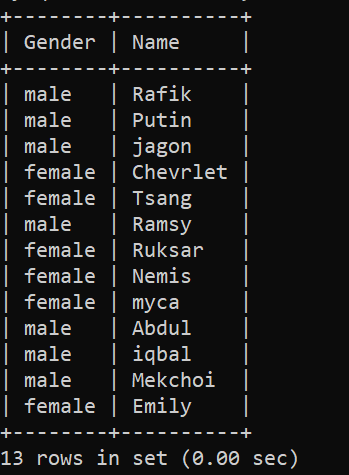
1. mysql> select Occupation from Population where Age between 18 and 40;



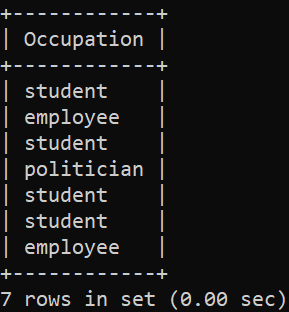
1. mysql> select Gender from Population where Age between 18 and 40;



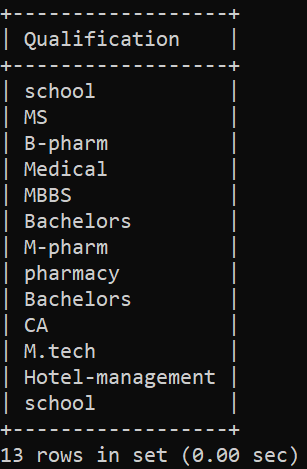
1. mysql> select Gender,Name from Population where Population between 6 and 20;



1. mysql> select Occupation from Population where Population between 15 and 35;

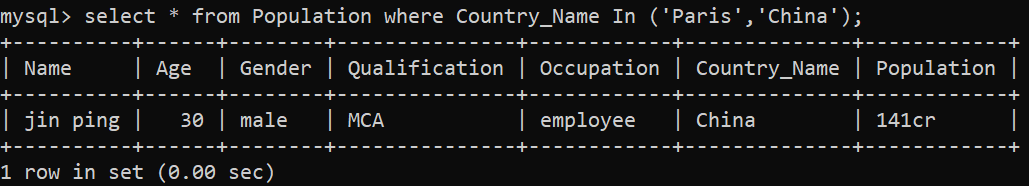


1. mysql> select Qualification from Population where Population between 6 and 20;



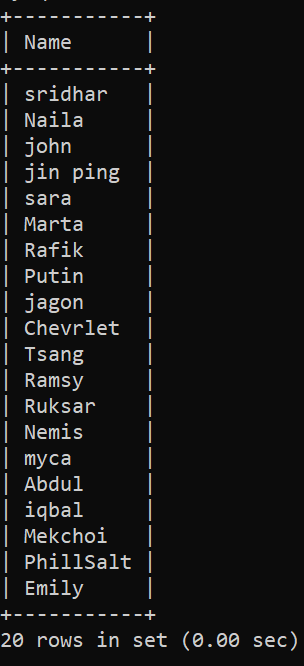
* for in: used fetch either of given conditions.

1. mysql> select \* from Population where Country\_Name in ('Paris','China');

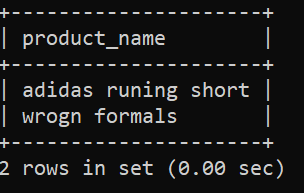


1. mysql> select Name from Population where Occupation in ('employee','student');

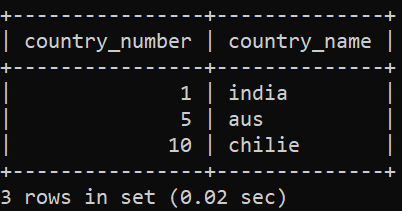
Retrieves all names where occupation is **either 'employee' or 'student'**.



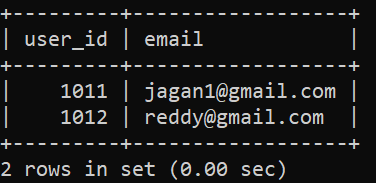
1. mysql> select product\_name from inventory where price in(500,4000);



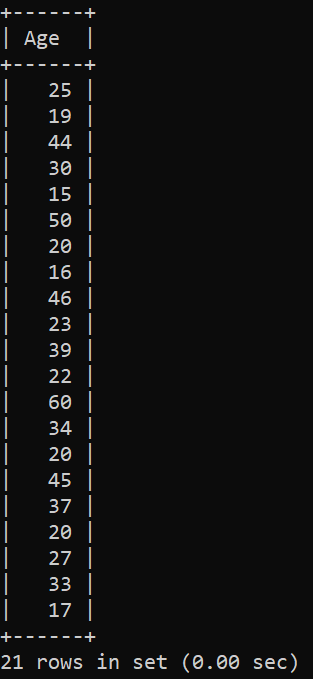
1. mysql> select country\_number,country\_name from country where capital in ('delhi','kangaroo','peru');



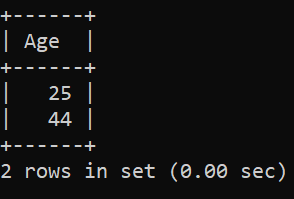
1. mysql> select user\_id,email from Authentication where password in('jagan','reddy');



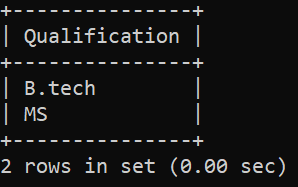
1. mysql> select Age from Population where Gender in('male','female');



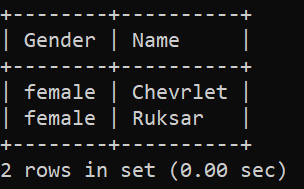
1. mysql> select Age from Population where Country\_Name in('India','USA');



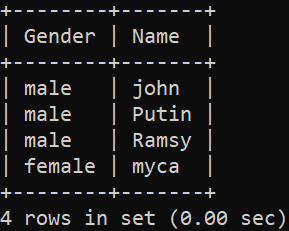
1. mysql> select Qualification from Population where Country\_Name in('India','USA');



1. mysql> select Gender,Name from Population where Country\_Name in('Philippines','Mexico');



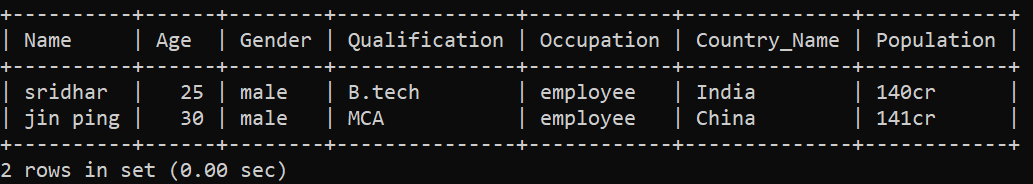
1. mysql> select Gender,Name from Population where Qualification in('MS','Bachelors');



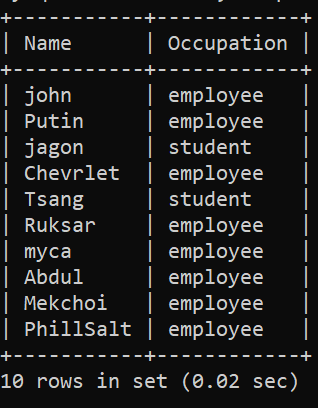
Where clause using Logical operators:

*AND*:

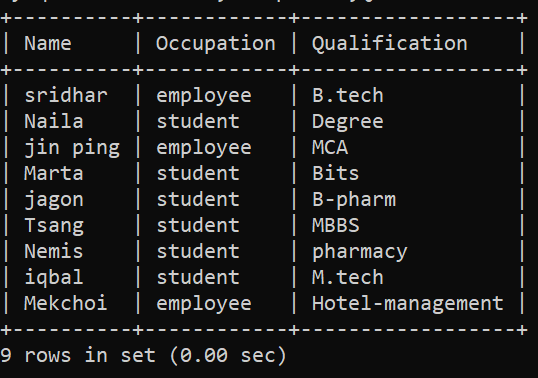
Eg:1) mysql> select \* from Population where Age>=18 and Population>50;



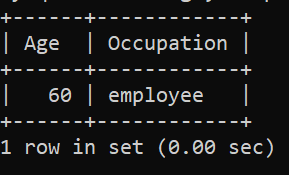
2) mysql> select Name,Occupation from Population where Age<50 and Population<140 and Age>20 ;



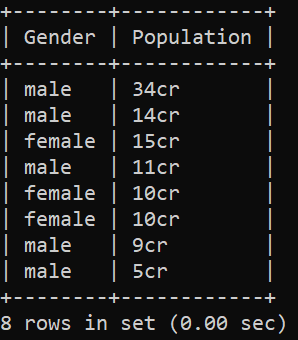
3) mysql> select Name,Occupation,Qualification from Population where Age<=30 and Age>18;



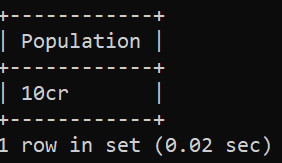
4) mysql> select Age,Occupation from Population where Qualification='Bachelors' and Gender='male';



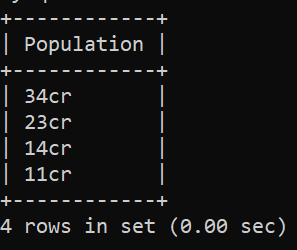
1. mysql> select Gender,Population from Population where Occupation='employee' and Age>30;



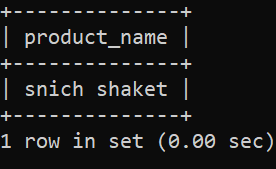
1. mysql> select distinct Population from Population where Age>40 and Gender='female';



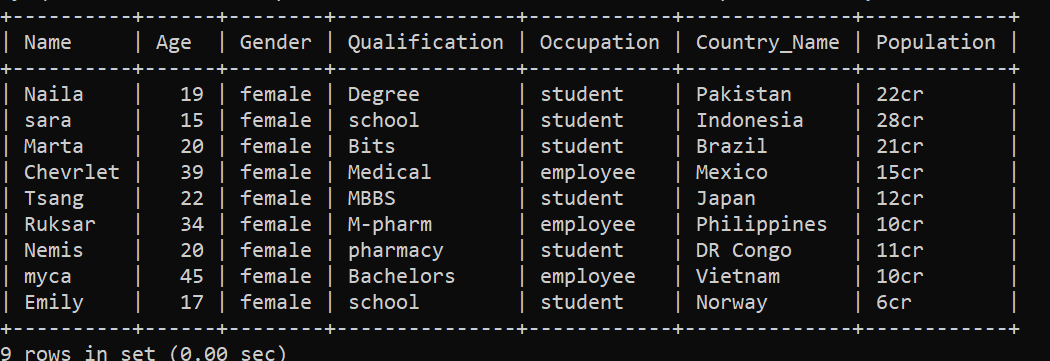
1. mysql> select distinct Population from Population where Age>40 and Gender='male';



1. mysql> select product\_name from inventory where price<3000 and prod\_id=5;



1. mysql> select \* from Population where Gender='female' and Population<140;

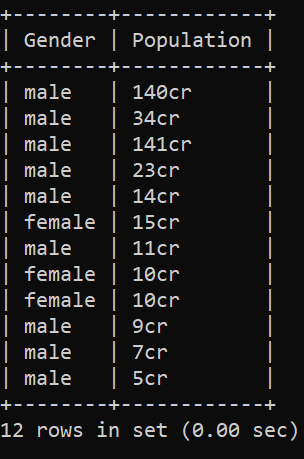


*OR:*

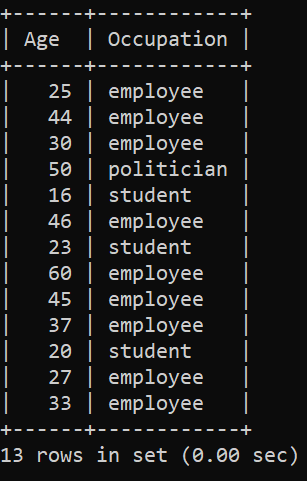
Eg :1) mysql> select \* from Population where Age>=18 or Population>50;



2) mysql> select Gender,Population from Population where Occupation='employee' or Age>30;



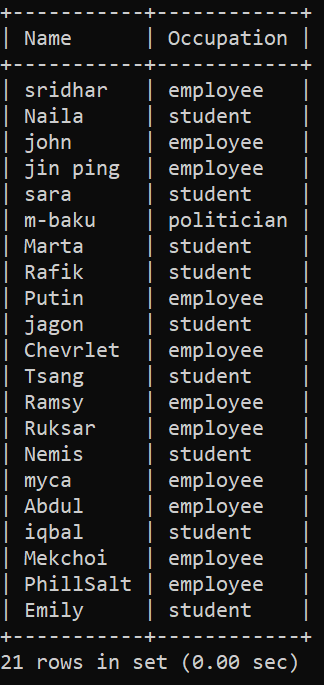
3) mysql> select Age,Occupation from Population where Qualification='Bachelors' or Gender='male';



4) mysql> select Name,Occupation,Qualification from Population where Age<=30 or Age>18;



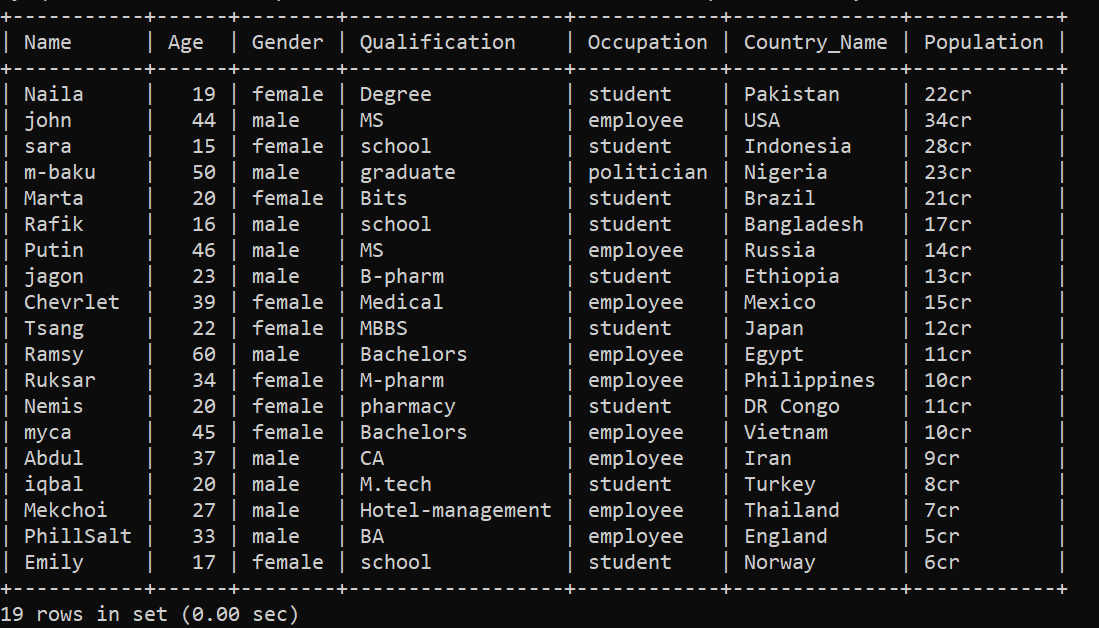
5)mysql> select Name,Occupation from Population where Age<50 and Population<140 or Age>20 ;



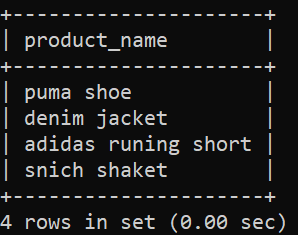
6) mysql> select Name from Population where Age=20 or Gender='male';



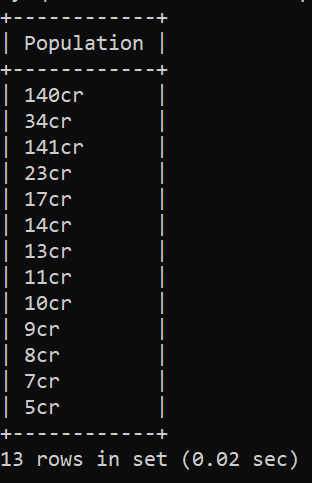
7) mysql> select \* from Population where Gender='female' or Population<140;



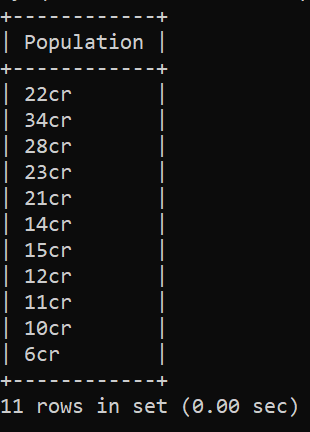
8) mysql> select product\_name from inventory where price<3000 or prod\_id=5;



9) mysql> select distinct Population from Population where Age>40 or Gender='male';

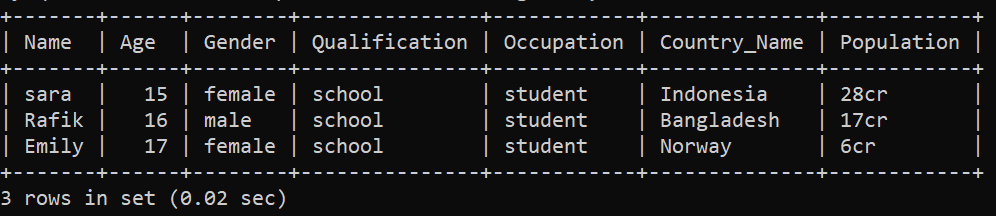


10) mysql> select distinct Population from Population where Age>40 or Gender='female';

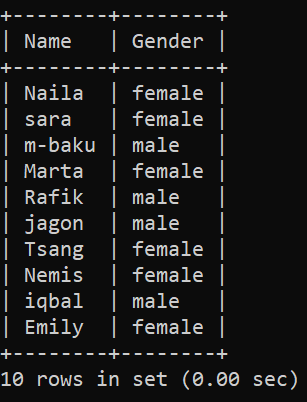


*NOT:*

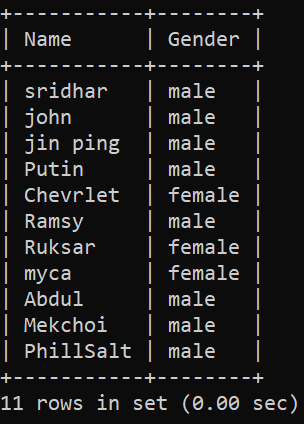
Eg: 1) mysql> select \* from Population where not Age>=18;



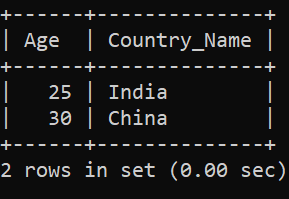
2) mysql> select Name,Gender from Population where not Occupation='employee';



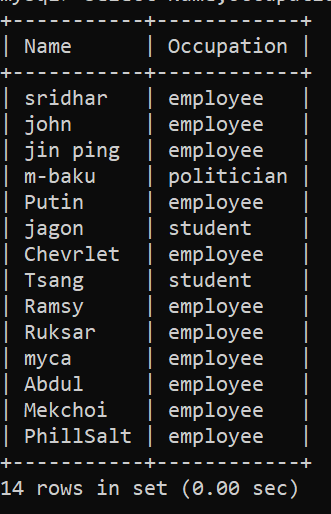
3) mysql> select Name,Gender from Population where not Occupation!='employee';



4) mysql> select Age,Country\_Name from Population where not Population<140;



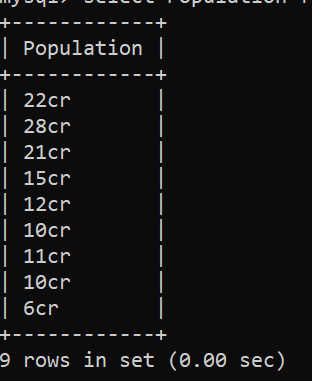
5) mysql> select Name,Occupation from Population where not Age<50 and Population<140 or Age>20 ;



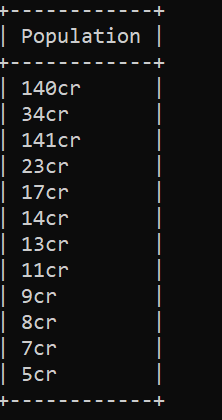
6) mysql> select Country\_Name from Population where Population not between 20 AND 140;



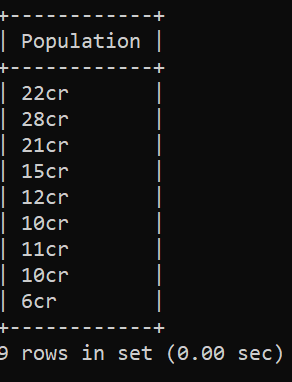
7) mysql> select Population from Population where not Gender='male';



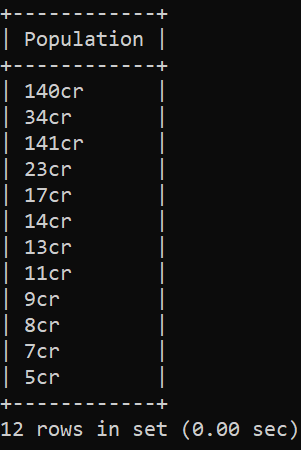
8) mysql> select Population from Population where not Gender!='male';



9) mysql> select Population from Population where not Gender!='female';



10) mysql> select Population from Population where not Gender='female';



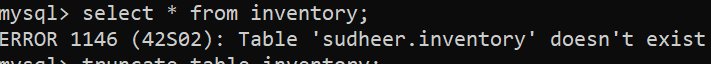
DDL & DML commands :-



1. *drop* : deletes data from tables.

Syntax ; drop table table-name

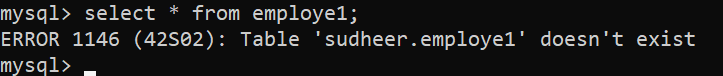
1. drop table inventory;



1. drop table states;



1. drop table employe1;



1. drop table customer\_order;

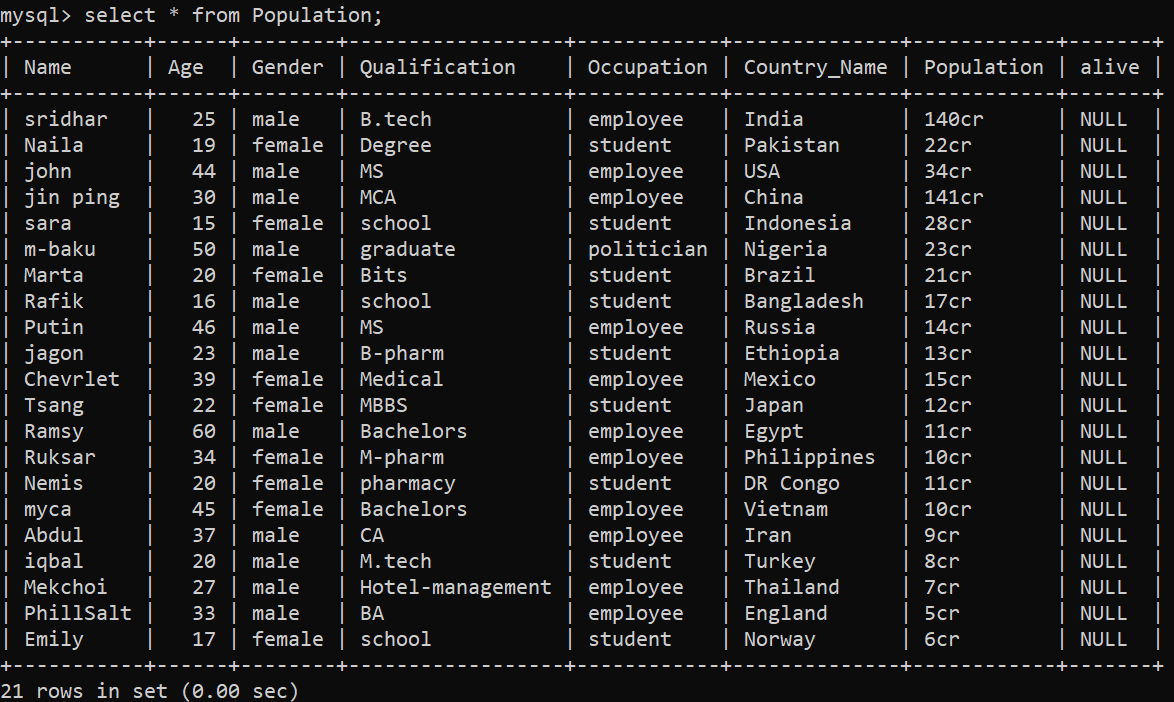


1. *alter* : alters the structure of database.

Syntax ; alter table table-name add column column-name datatype.

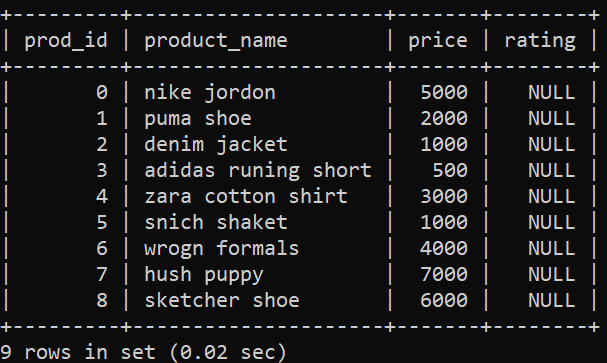
1. mysql> alter table Population add column alive varchar(10);

Query OK, 0 rows affected (0.14 sec)

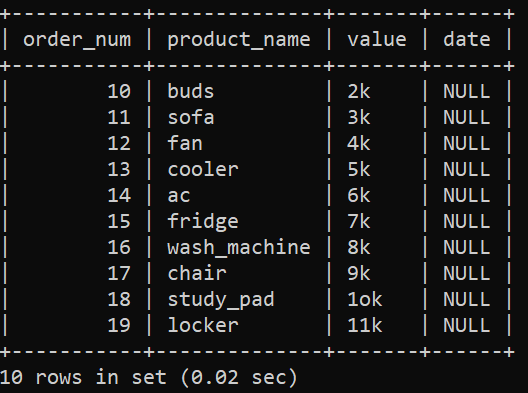


Here showing null bcs we didn’t insert any values

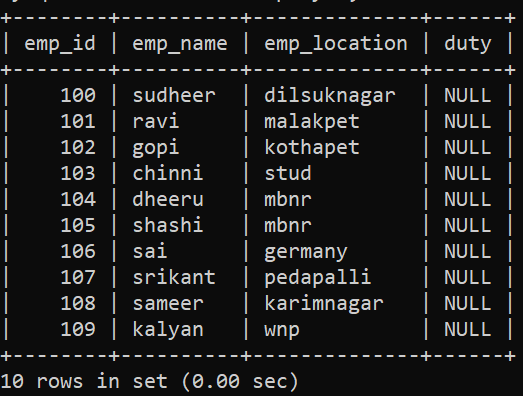
1. alter table inventory add column rating int;



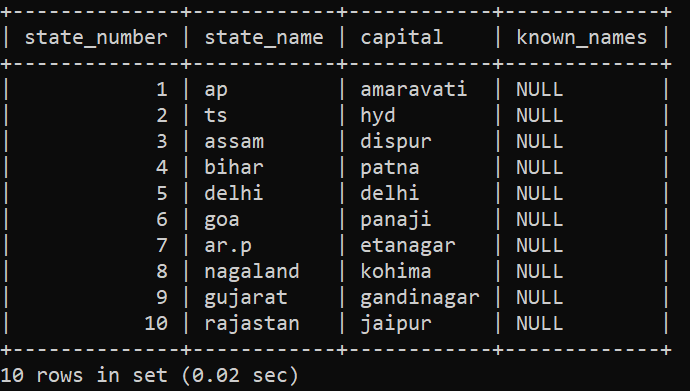
1. alter table customer\_order add column date int(19);



1. alter table employe1 add column duty varchar(29);



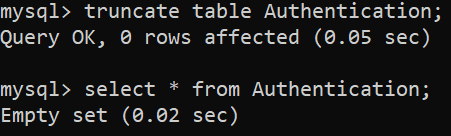
1. alter table states add column known\_names varchar(20);



1. *Truncate* : it removes all the records from table.

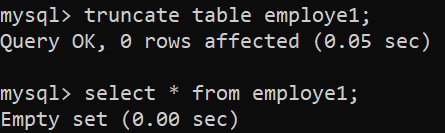
Syntax ; truncate table table-name;

1. truncate table Authentication;

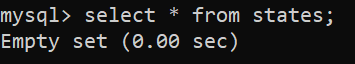


1. truncate table inventory;
2. truncate table customer\_order;





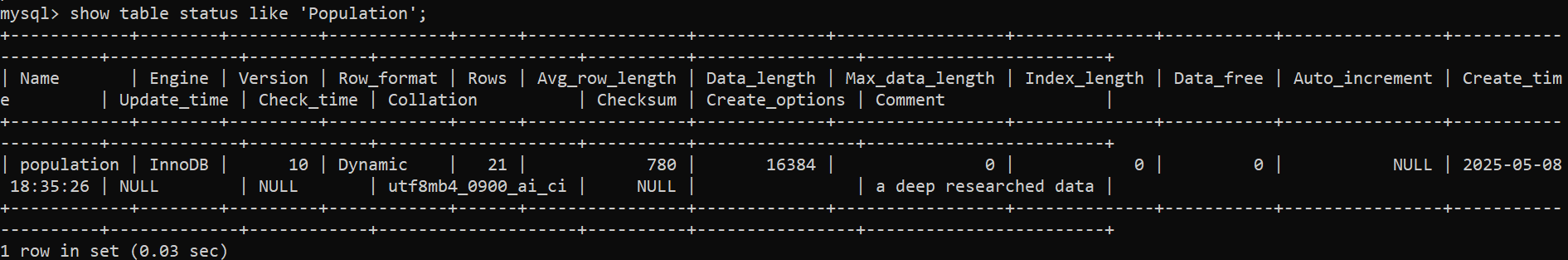
1. truncate table states;



1. *Comment* : adds comment to the data dictionary.

Syntax ; comment ‘comment-text’ on table table-name;

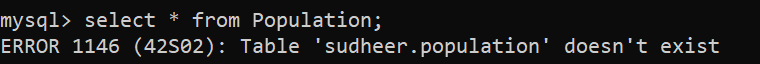
1. alter table Population comment='a deep researched data';



1. *Rename* : renames the existing table.

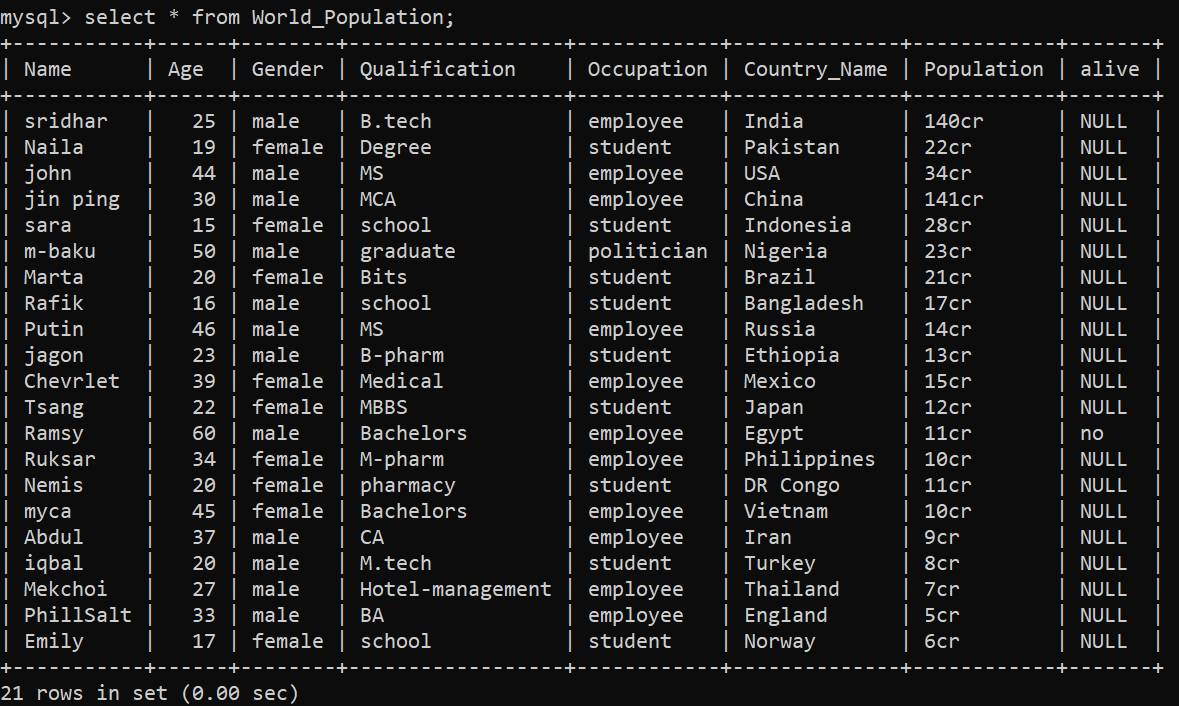
Syntax ; rename table old-table-name to new-table-name;

1. rename table Population to World\_Population;

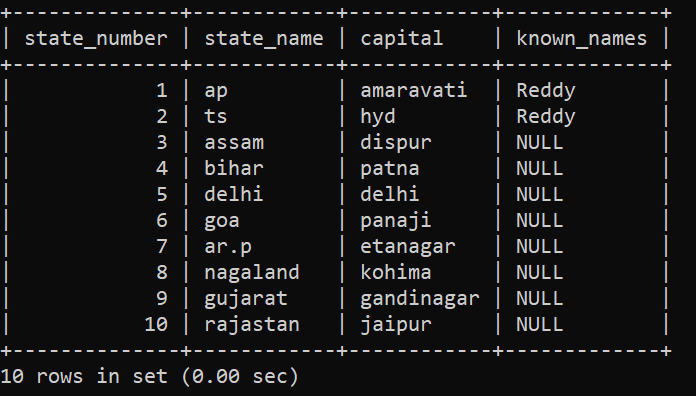


1. *Insert* : inserts data into table.

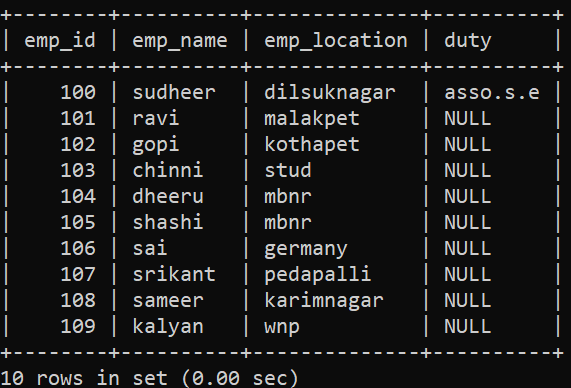
*Update* :



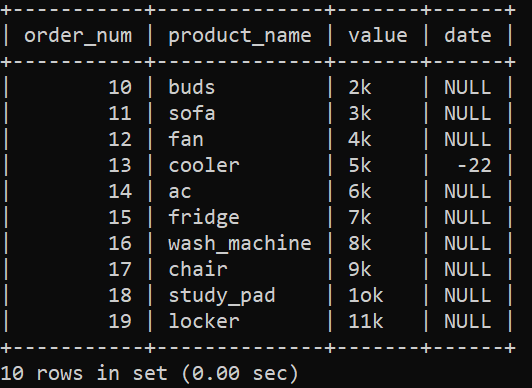
1. update states set known\_names='Reddy' where state\_name in('ap','ts');



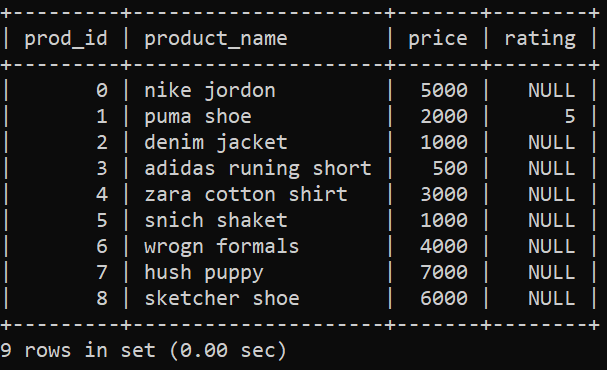
1. update employe1 set duty='asso.s.e' where emp\_id=100;



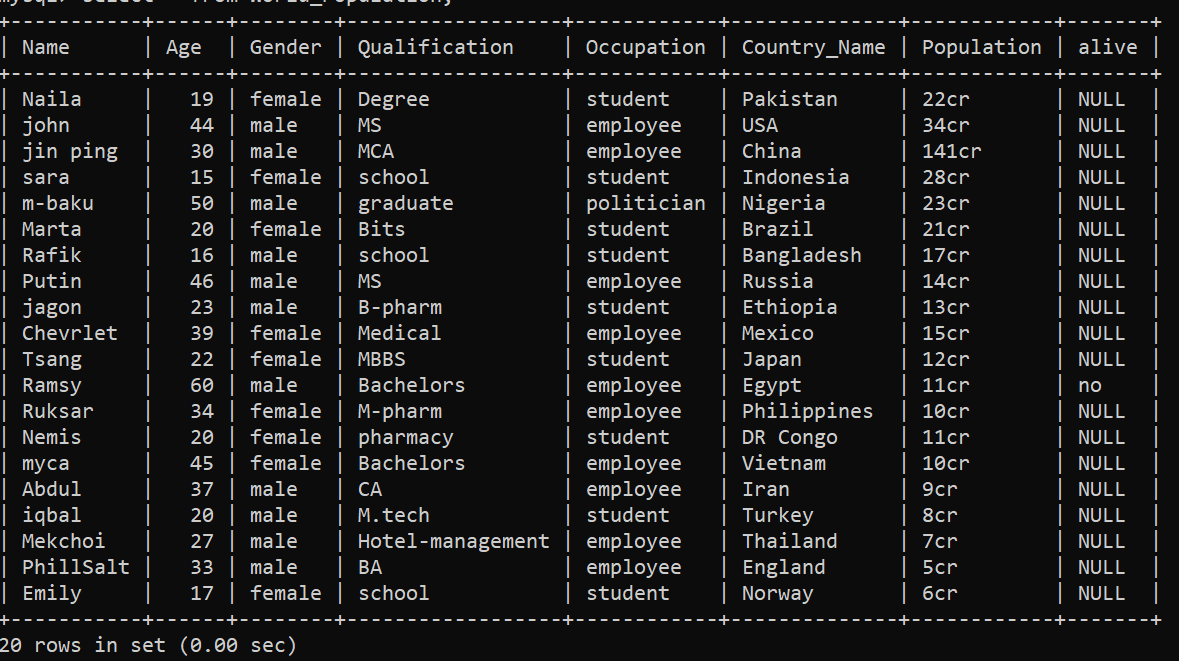
1. update customer\_order set date=13-10-25 where order\_num=13;

 I mean to enter \_ but took ’ –‘

1. update inventory set rating=5 where product\_name='puma shoe';



1. delete from World\_Population where Name='sridhar';



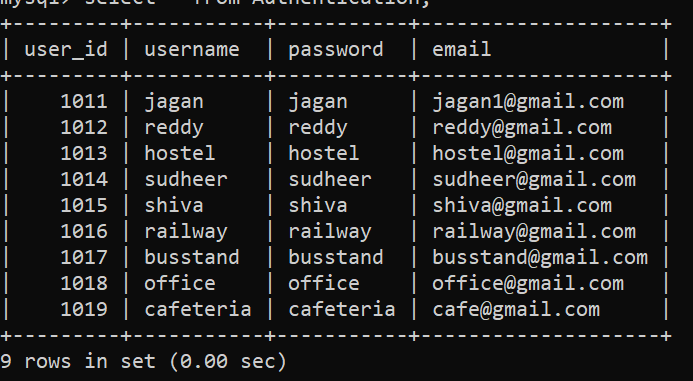
1. delete from World\_Population where Gender='female';



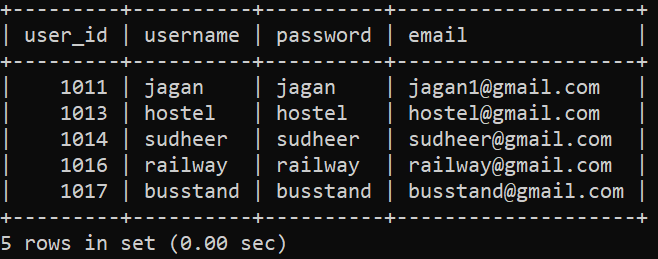
1. delete from World\_Population where alive='no';



1. delete from Authentication where user\_id=1020;

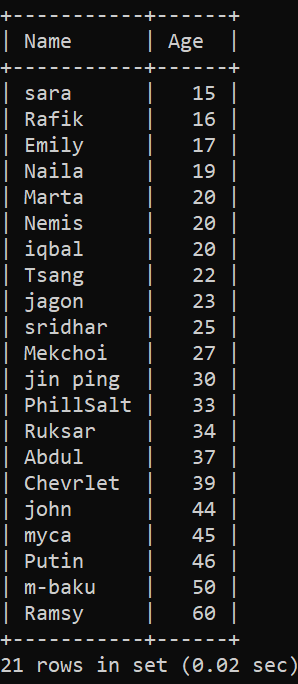


1. delete from Authentication where password='reddy' or user\_id=1019;

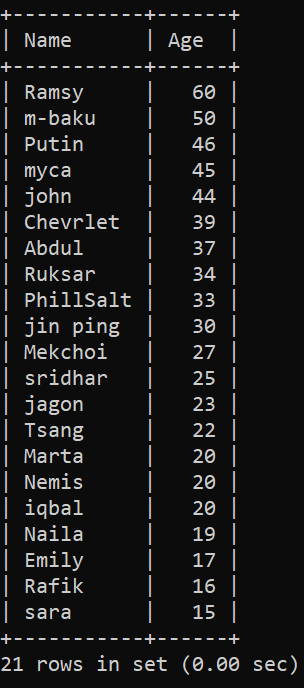


Order by :-

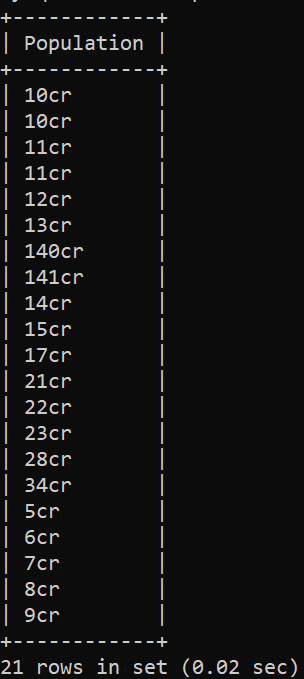
1. Name select,Age from Population order by Age ASC;



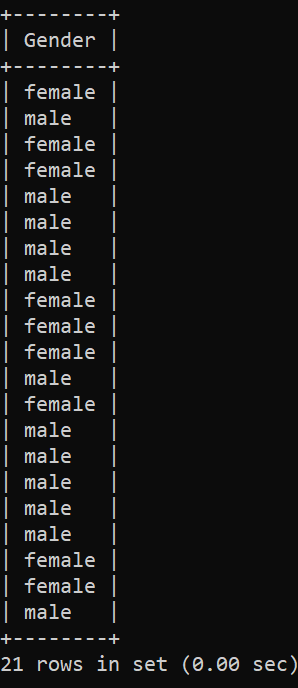
1. select Name,Age from Population order by Age DESC;



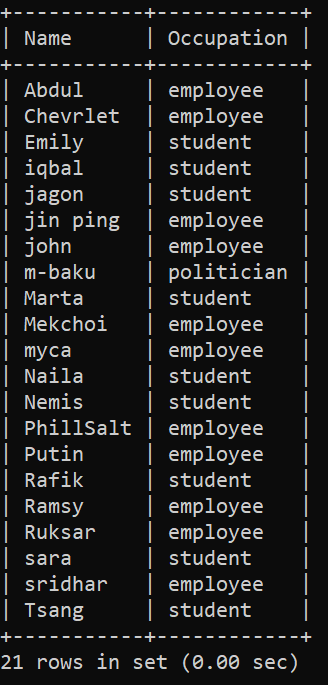
1. select Population from Population order by Population ASC;

 it won’t shows bcs here ‘cr’ added to values.

1. select Gender from Population order by Name DESC;

 it in compatible with select distinct.

1. select Name,Occupation from Population order by Name ASC;

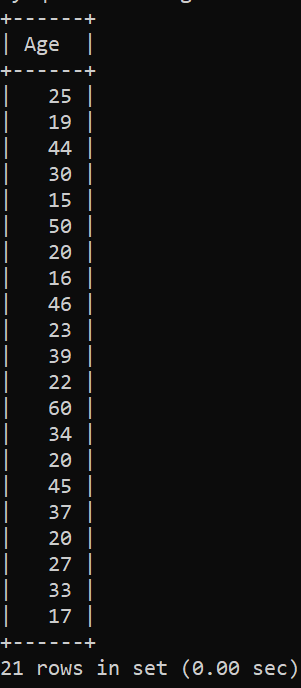


Null values :-

1. select \* from Population where health is null;



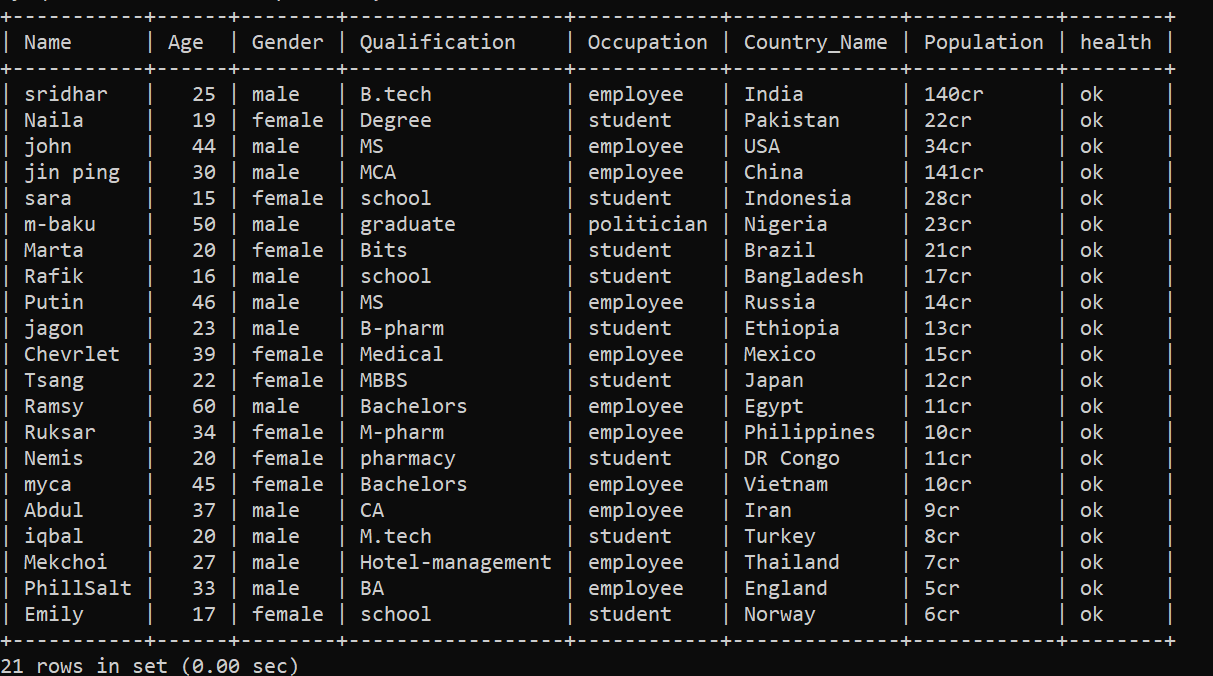
1. select Age from Population where health is null;



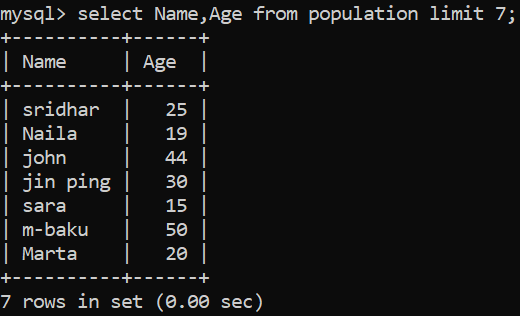
1. select Age from Population where health is not null;



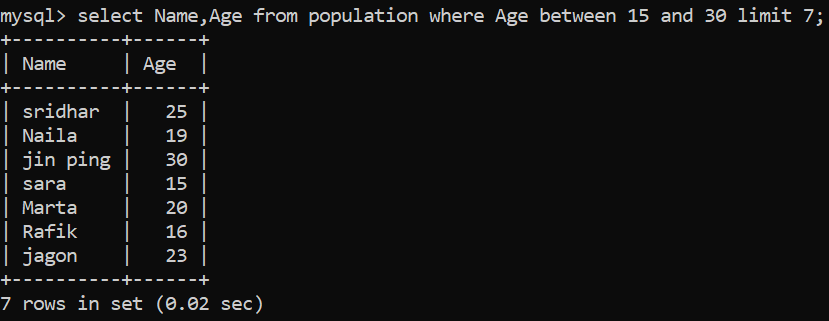
1. update Population set health='ok' where health is null;



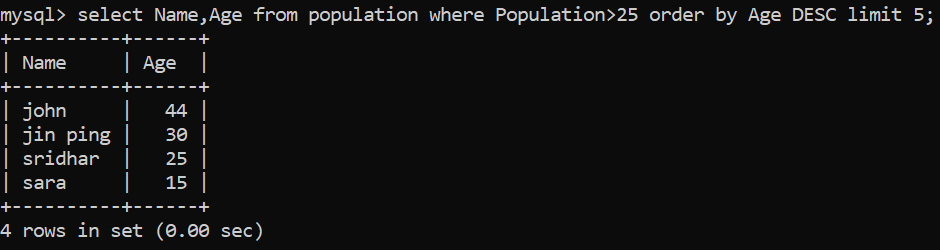
***Limit clause*** :-



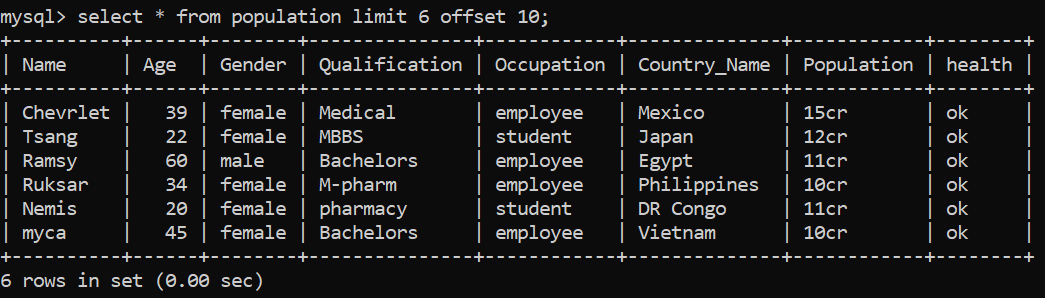
2.



3.

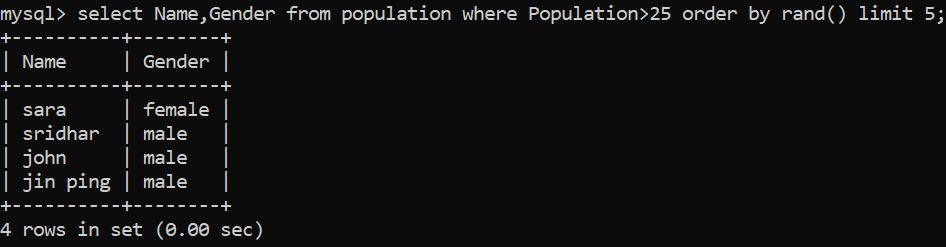


4.



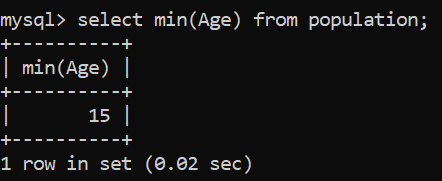
Pagination🡪 here by using offset 10 we can retrieve limit 6 by skipping first 10 rows.

5.



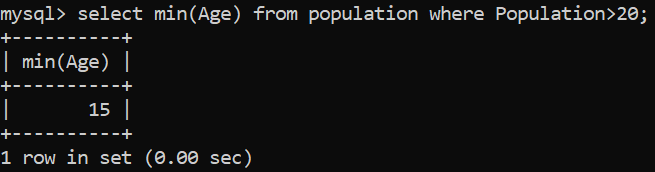
*Min() function*:

1.

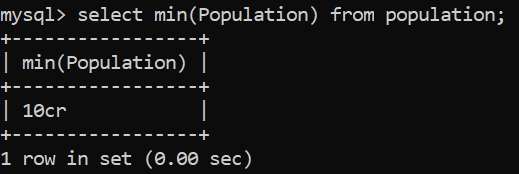


Also : select min(Age) as lowestAge from population;

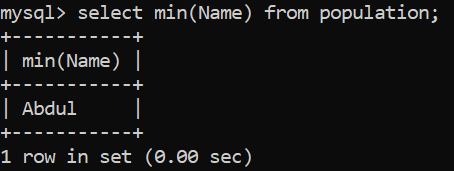
1. select min(Age) from population where Population; (or)



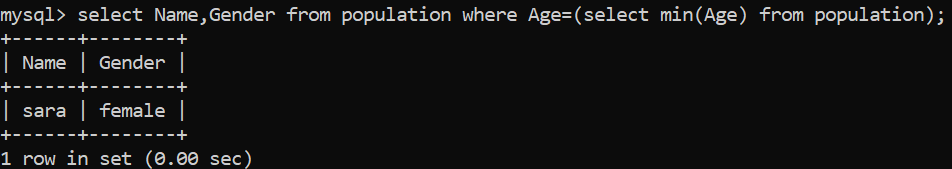
3.



4.For strings like names if we entered min() we get by alphabetical order.



5.

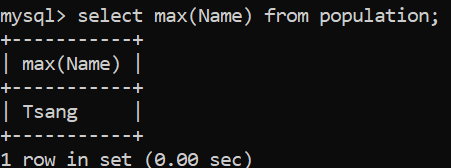


*Max() function* :

1.

 also use: select max(Age) as highestAge from population;

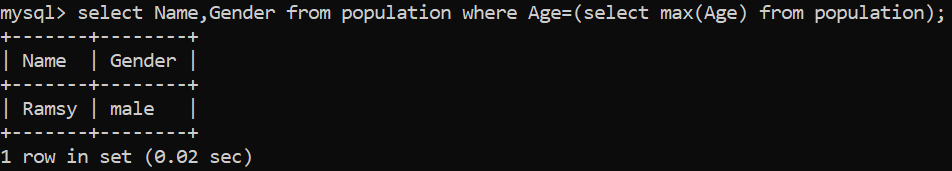
2.

 by using string values like names we get alphabetical order by descending order.

3.

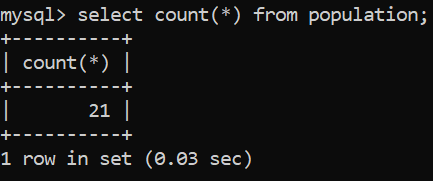


4.

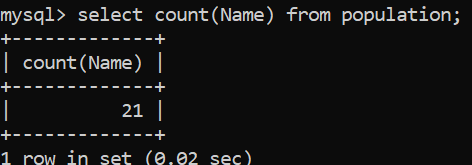


*Count() function* :

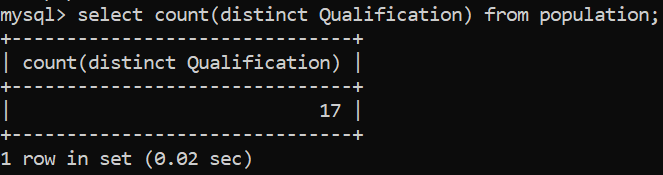
1.



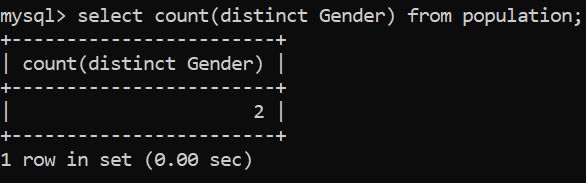
2.



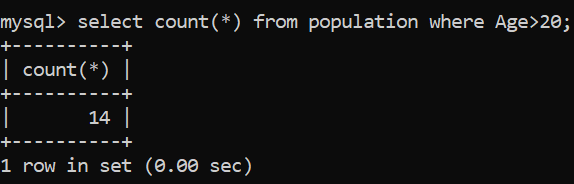
3.

 by using distinct here we can select unique values, no duplicates.

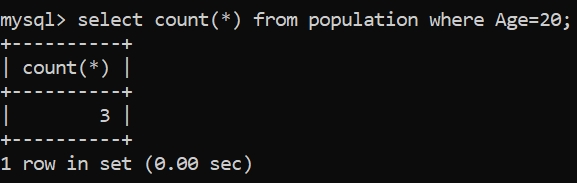
4.



5.

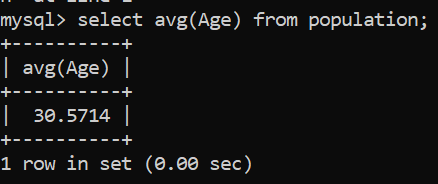


6.

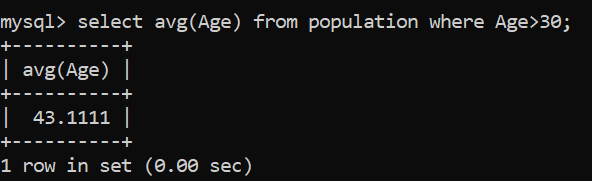


*Avg() function* :

1.



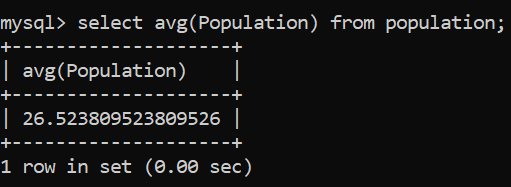
2.



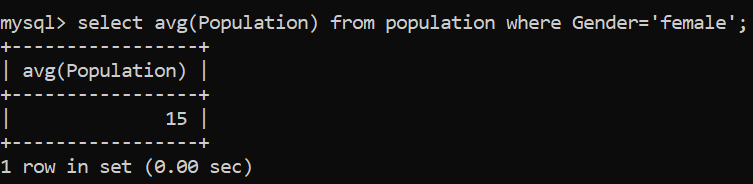
3.



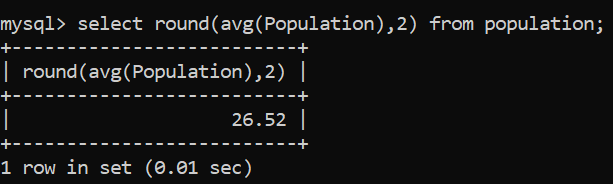
4.



5.

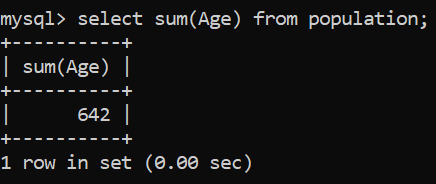


6.



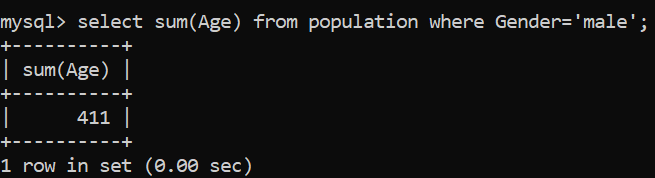
*Sum() function* :

1.

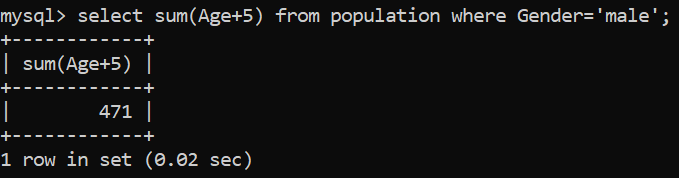


SELECT SUM(Age) AS TotalAge FROM population; can use this syntax also.

2.

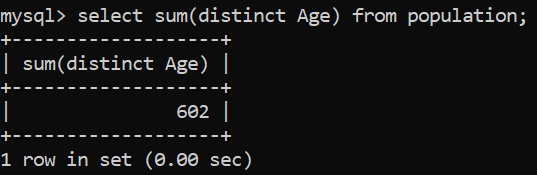


3.

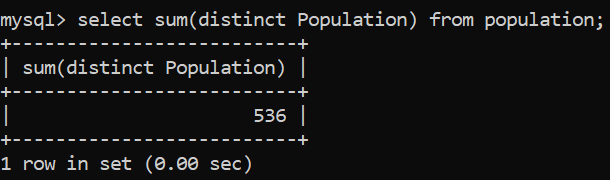


Sum(Age+5) => means it adds num.5 to each & every male ages before going to add all.

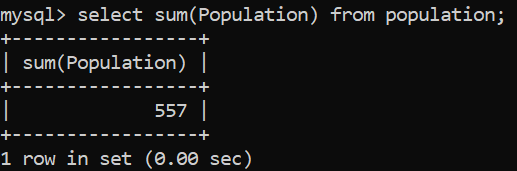
4.



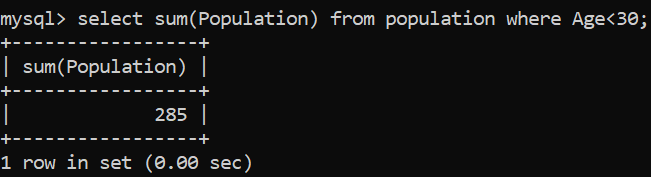
5.



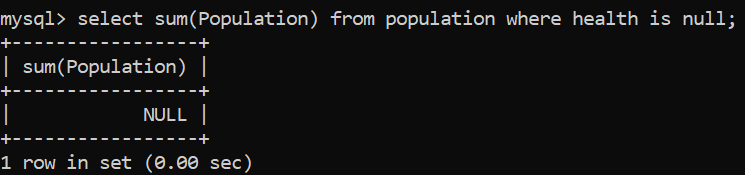
6.



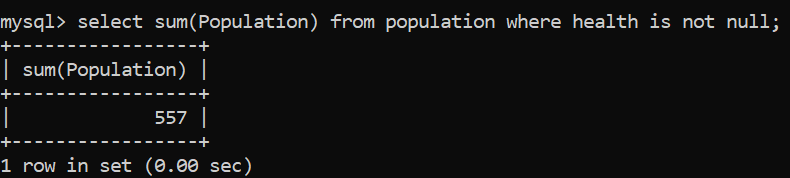
7.



8.

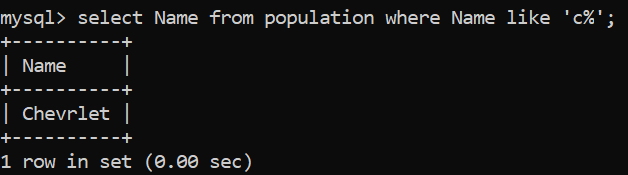


9.



*Like() operator* :

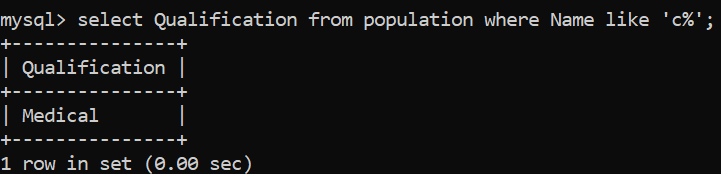
1.



2.



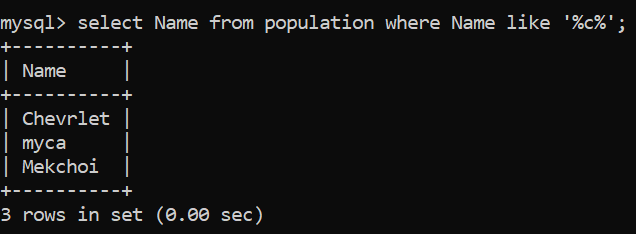
3.



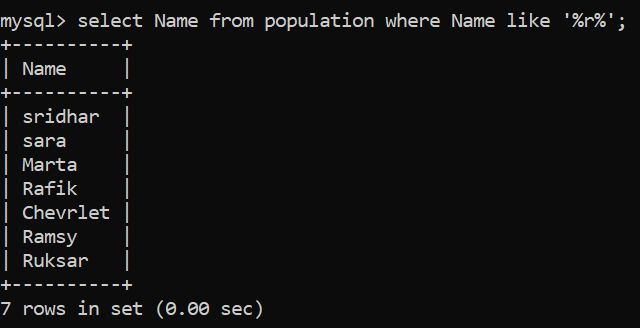
4.



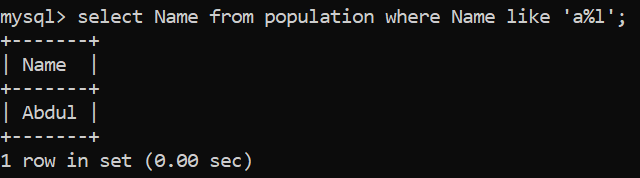
5.



6.

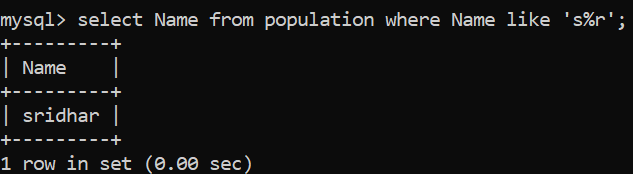


7.

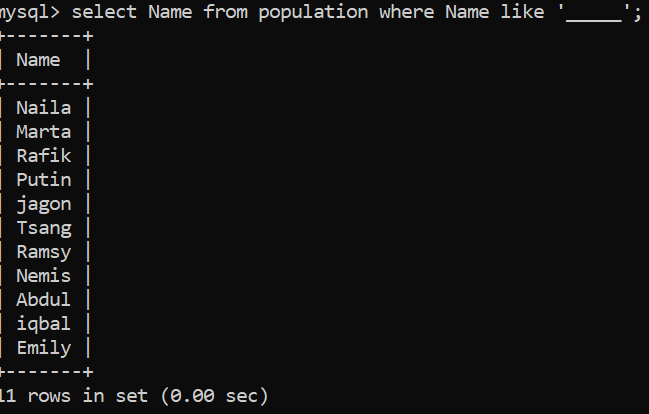


a%l rep. name starts with ‘a’ and ends with ‘l’.

8.

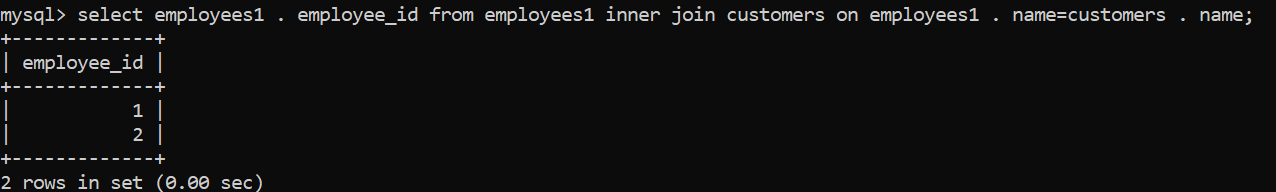


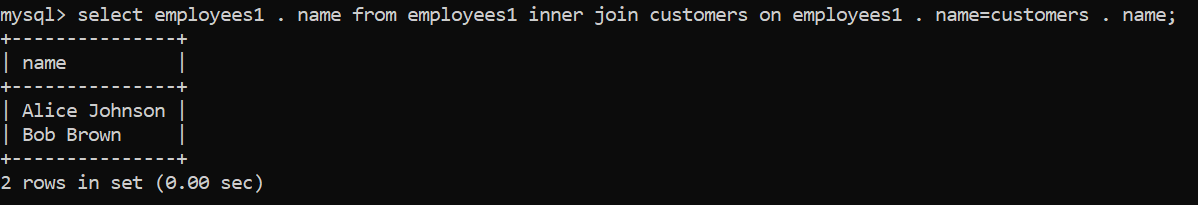
9.



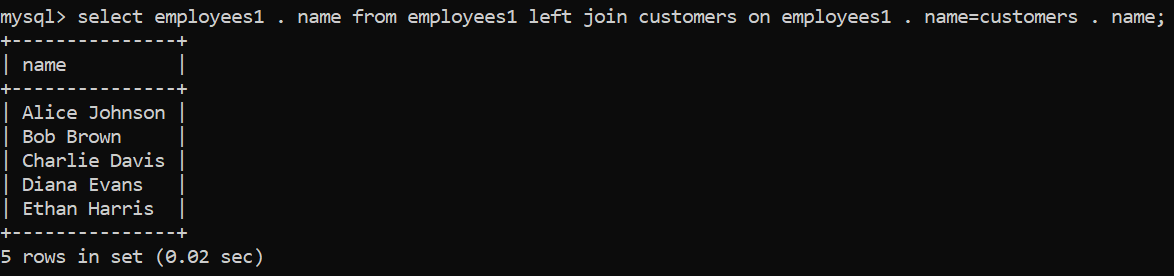
**Join clause** :-

*inner join*

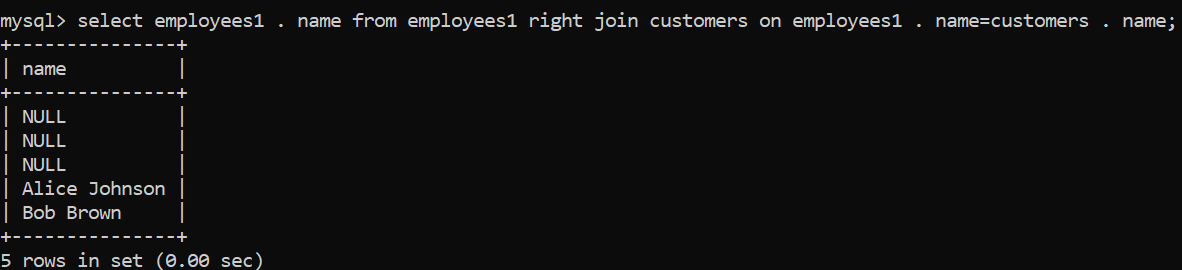




*Left join*

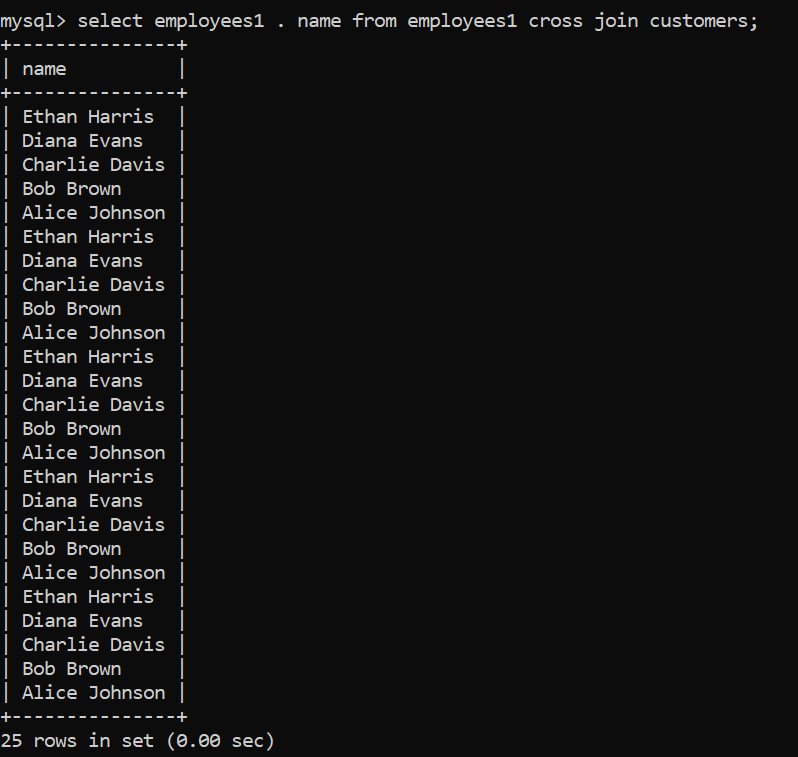


1. *Right join* :



1. *Cross join* : returns all the records from both tables.

Syntax; select column\_name from table\_name cross join table2;

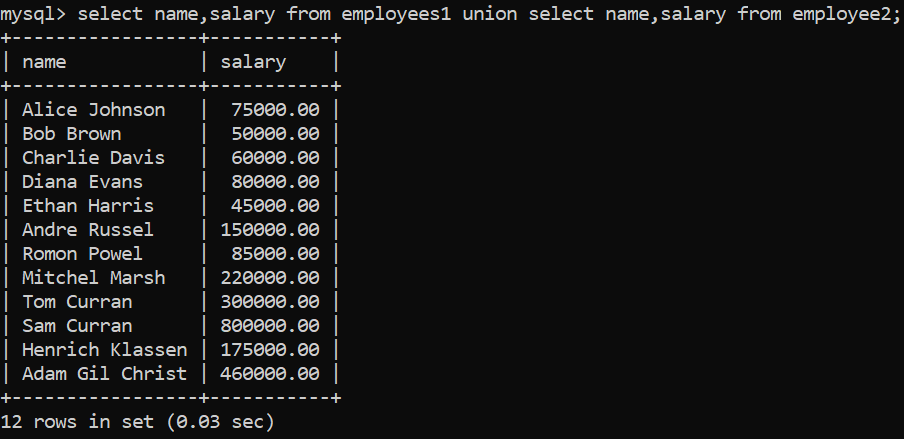


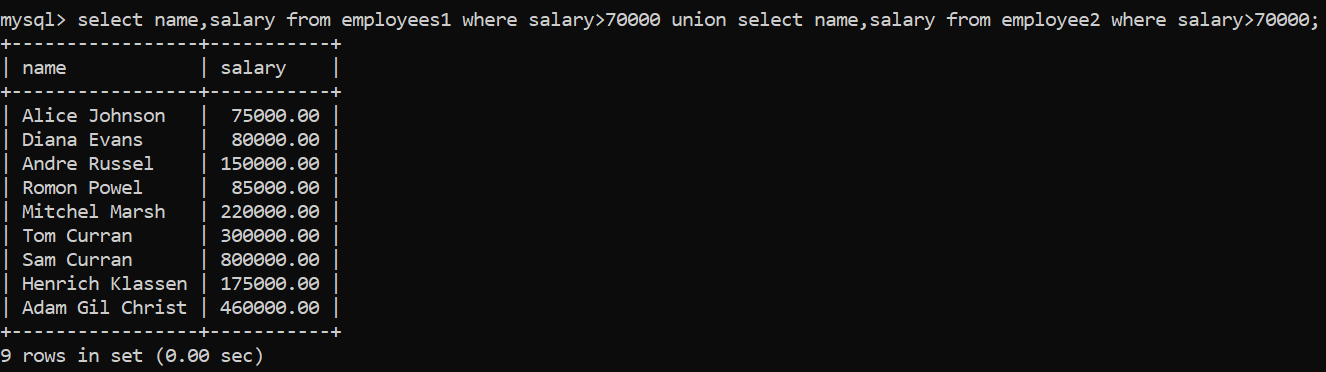
1. *Self join* : a table is joined with it self.



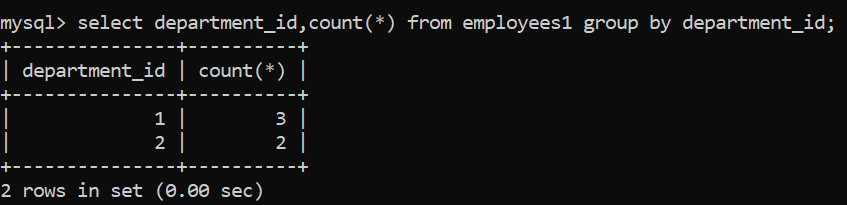


*Union operator* :

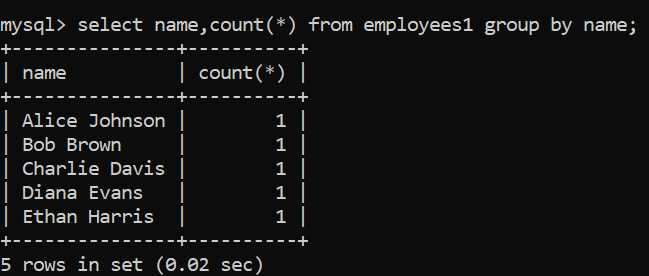




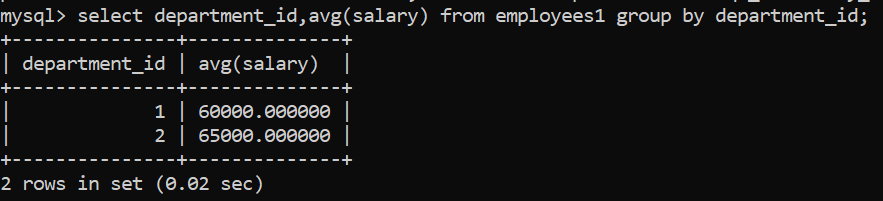
**Group by statement** :-



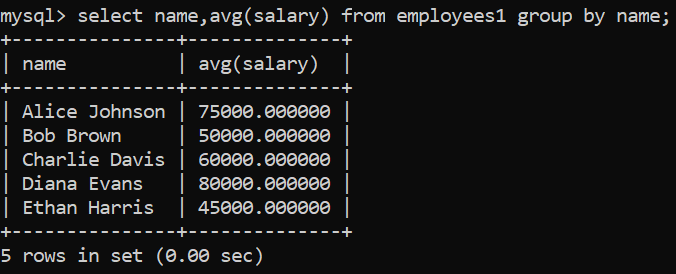
2.

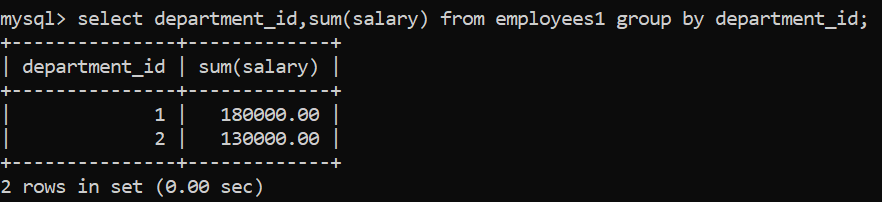


3.calulates average salary for employees under each department\_id.

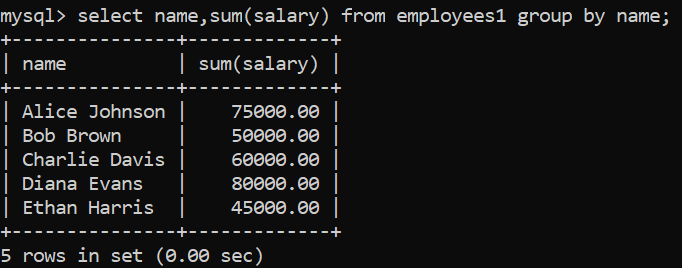


4.



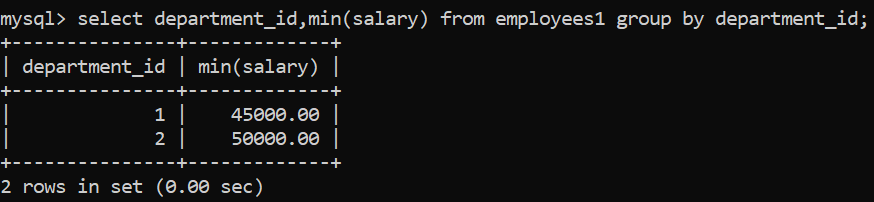


6.



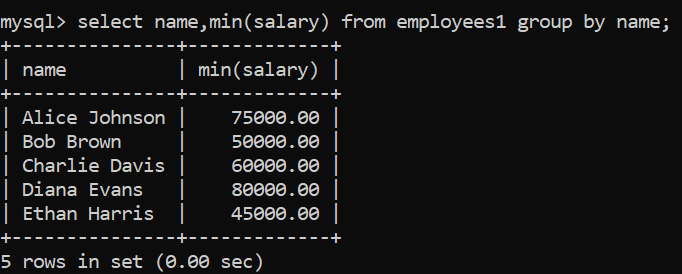
Calculates sum all salaries in each name. by making each name as one group.

7.



Retrieves lowest salary from each department.

8.

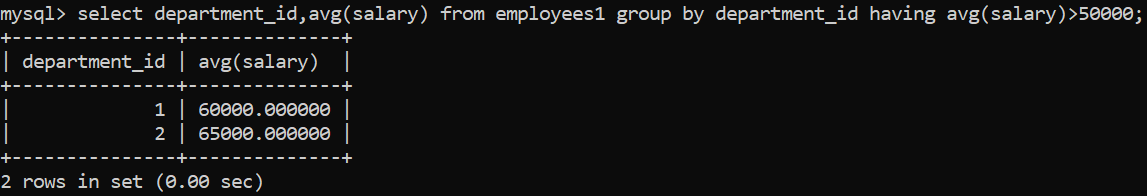


**Having clause** :-

1.

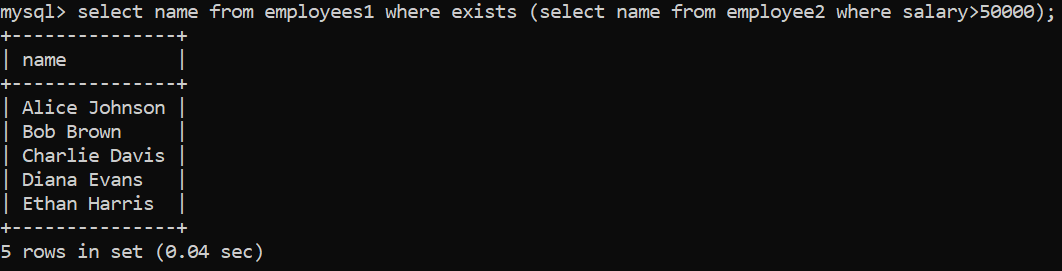


2.

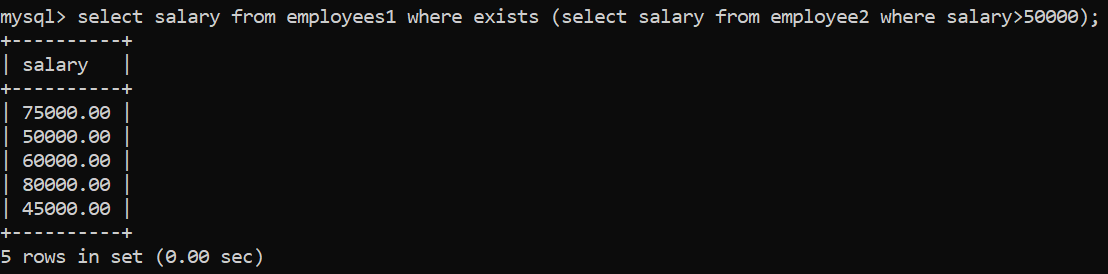


**Exists operator** :-

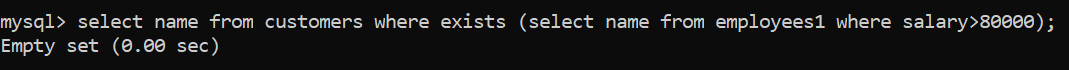
1.



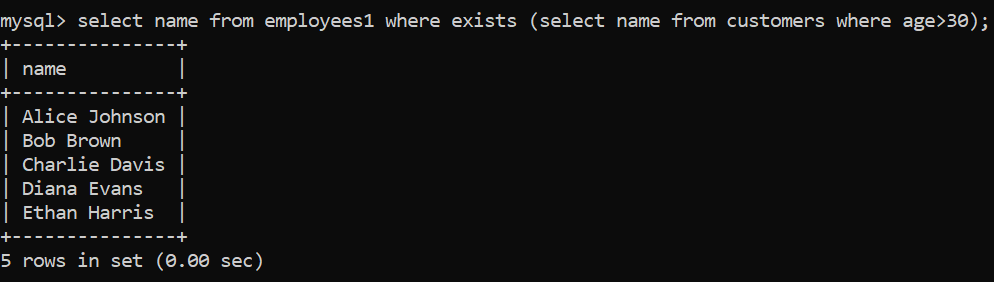
2.



3.

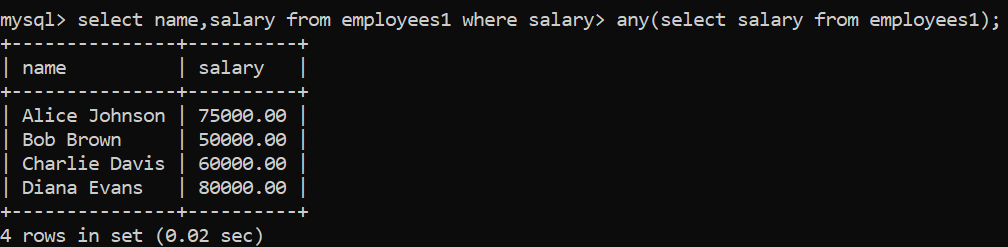


4.

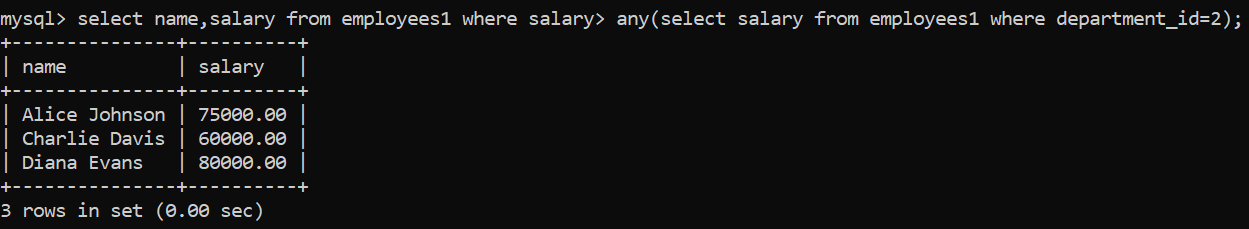


**Any operator** :-

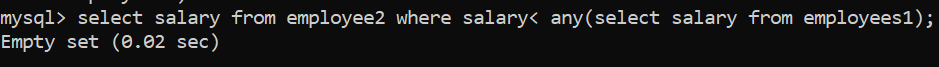
1.



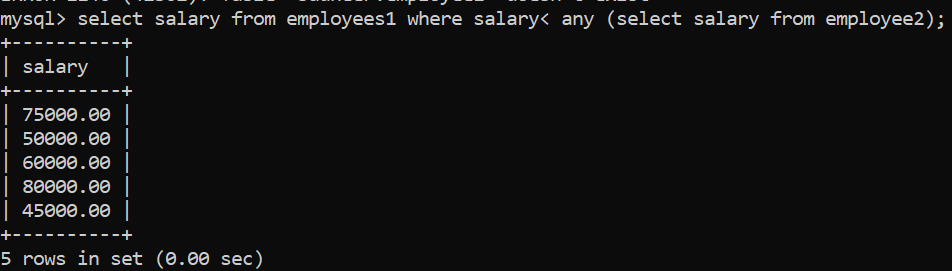
2.



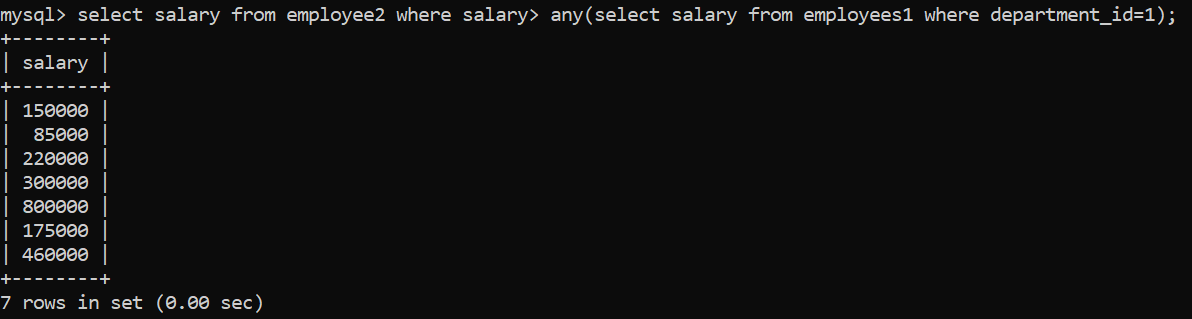
3.



4.



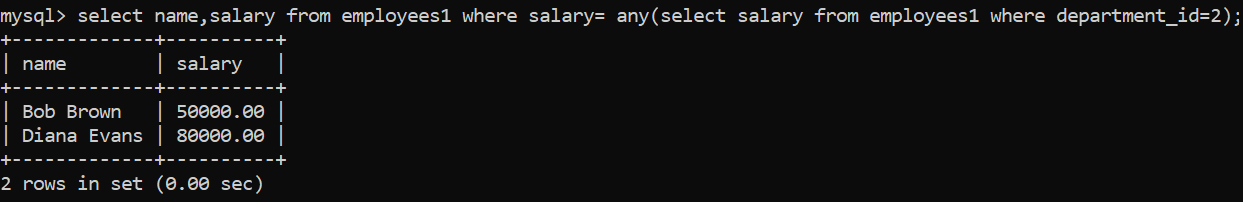
5.



6.

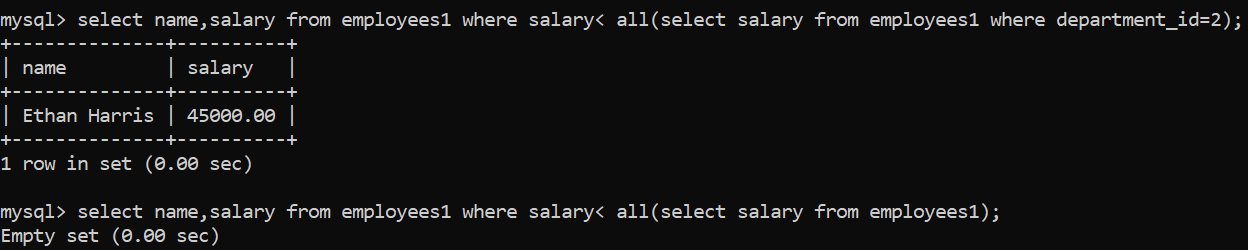


7.

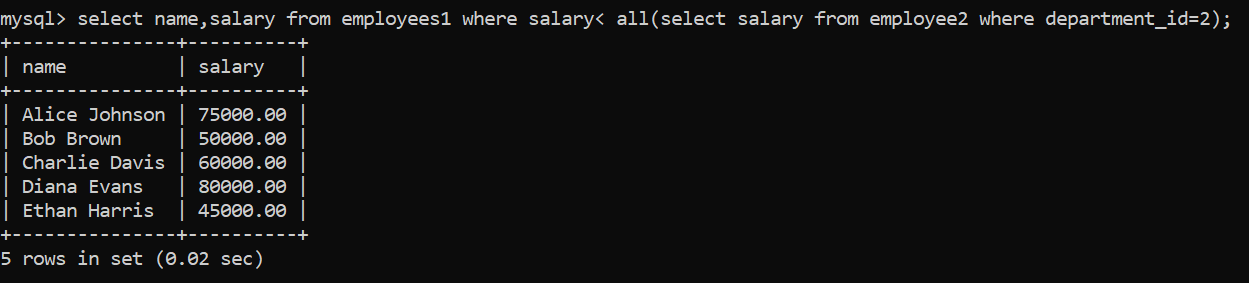


**All operator** :-

2.



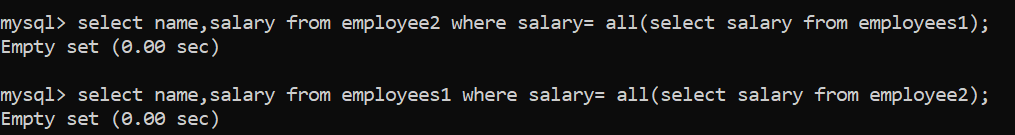
3.



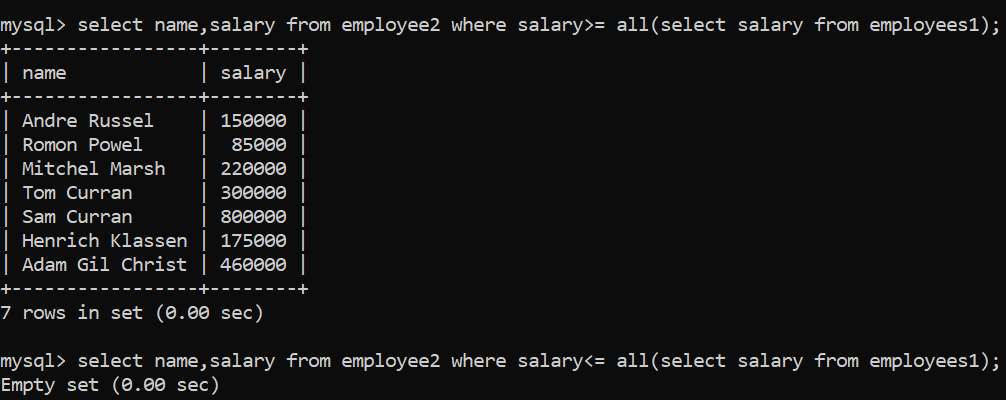
4.



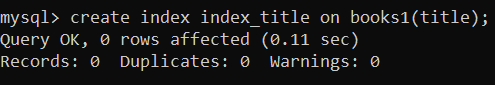
5.

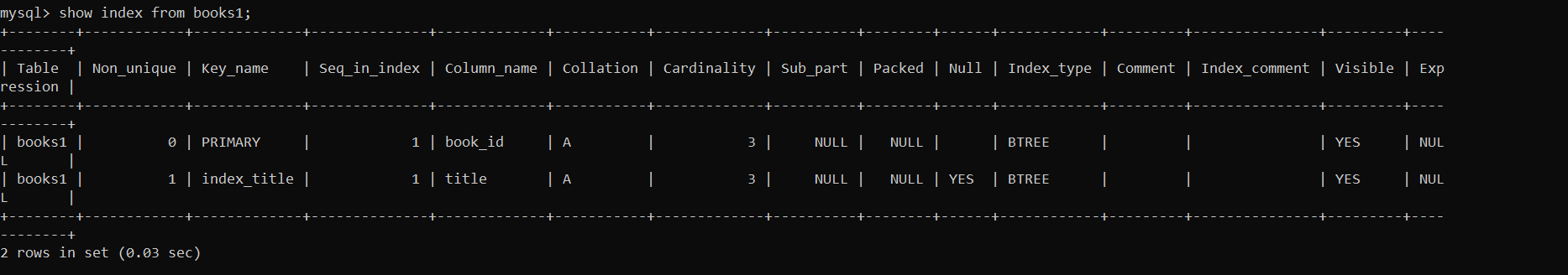


6.



**Index** :





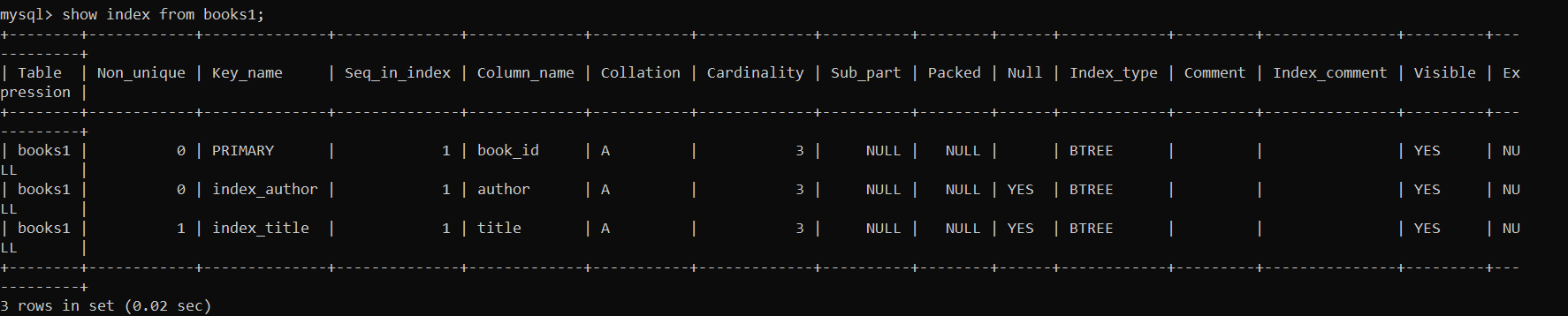
*To create unique index*:

Syntax; create unique index index\_name on table\_name(col1, col2….);



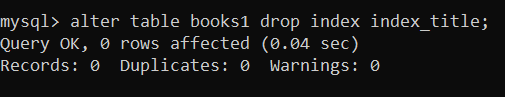
Unique index throws an error, if you tries to insert duplicate index. Because we already created an index for column “title”.





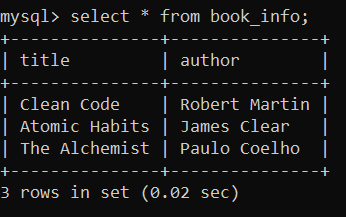
*Drop index* : used to delete the index that was already created on a table.

Syntax; alter table table\_name drop index index\_name;



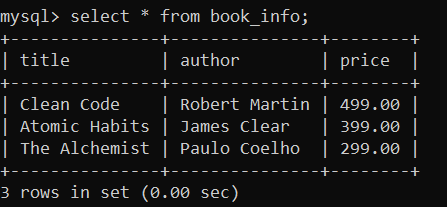
**Views** :





*Updating a view*





*Dropping view* : to delete a view

Drop view view\_name;

