**DDL (DATA DEFINITION LANGUAGE):**

• CREATE  
• ALTER  
• DROP  
• TRUNCATE  
• COMMENT  
• RENAME

**SQL>** CREATE TABLE EMP (EMPNO NUMBER (4), ENAME VARCHAR2 (10),  
DESIGNATIN VARCHAR2 (10), SALARY NUMBER (8,2));

***Table created.***

**SQL>** DESC <TABLE NAME>;

**SQL>** DESC EMP;

**Name Null? Type**

EMPNO NUMBER(4)  
ENAME VARCHAR2(10)  
DESIGNATIN VARCHAR2(10)  
SALARY NUMBER(8,2)

**SQL>**ALTER TABLE EMP MODIFY EMPNO NUMBER (6);

***Table altered.***

**SQL>** DESC EMP;

Name Null? Type

EMPNO NUMBER(6)  
 ENAME VARCHAR2(10)  
 DESIGNATIN VARCHAR2(10)  
 SALARY NUMBER(8,2)

**SQL>**ALTER TABLE EMP ADD (DOB DATE, DOJ DATE);

***Table altered.***

**SQL>** DESC EMP;

Name Null? Type

EMPNO NUMBER(7)ENAME VARCHAR2(12)DESIGNATIN VARCHAR2(10)SALARY NUMBER(8,2)QUALIFICATION VARCHAR2(6)DOB DATEDOJ DATE

**REMOVE / DROP :**

**SQL>** ALTER TABLE EMP DROP COLUMN DOJ;

**SQL>** DESC EMP;

**Name Null? Type**

EMPNO NUMBER(7)  
ENAME VARCHAR2(12)  
DESIGNATIN VARCHAR2(10)  
SALARY NUMBER(8,2)  
QUALIFICATION VARCHAR2(6)  
DOB DATE

SQL>ALTER TABLE EMP DROP (DOB, QUALIFICATION);

*Table altered.*

SQL> DESC EMP;

**Name Null? Type**

|  |  |
| --- | --- |
| EMPNO | NUMBER(7) |
| ENAME | VARCHAR2(12) |
| DESIGNATIN | VARCHAR2(10) |
| SALARY | NUMBER(8,2) |

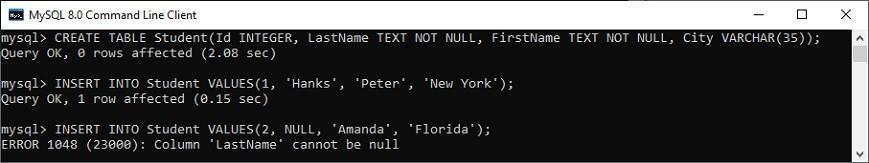
NOT NULL Constraint:

**MySQL>** CREATE TABLE Student (Id INTEGER, Last Name TEXT NOT NULL, FirstName TEXT NOT NULL, City VARCHAR (35));

**MySQL>** INSERT INTO Student VALUES(1, 'Hanks', 'Peter', 'New York');

**MySQL>** INSERT INTO Student VALUES(2, NULL, 'Amanda', 'Florida');

OUTPUT:



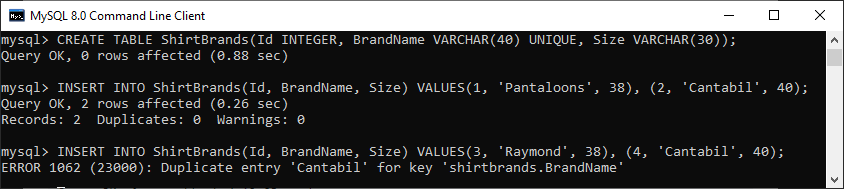
UNIQUE Constraint :

**MySQL>** CREATE TABLE ShirtBrands(Id INTEGER, BrandName VARCHAR(40) UNIQUE, Size VARCHAR(30));

**MySQL>**INSERT INTO ShirtBrands(Id,BrandName,Size)VALUES(1,'Pantaloons',38),(2, 'Cantabil',40);

**MySQL>**INSERT INTO ShirtBrands(Id,BrandName,Size)VALUES(1,'Raymond',38),(2, 'Cantabil',40);

**OUTPUT:**



CHECK CONSTRAINT :

**CHECK (expr)**

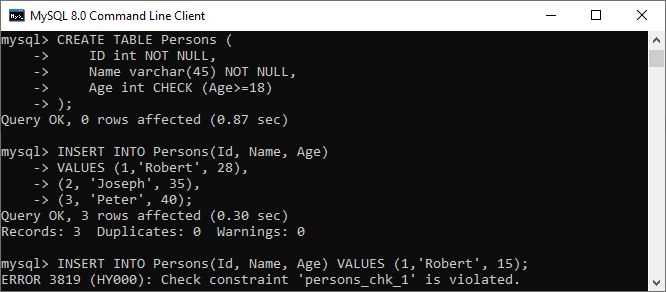
**MySQL>** CREATE TABLE Persons ( IDint NOT NULL,Namevarchar(45) NOT NULL, Age int  
CHECK (Age>=18) );

**MySQL>** INSERT INTO Persons(Id, Name, Age) VALUES (1,'Robert', 28), (2, 'Joseph', 35), (3, 'Peter',40);

**MySQL>** INSERT INTO Persons(Id, Name, Age) VALUES (1,'Robert', 15);

**OUTPUT:**

*In the below output, we can see that the first INSERT query executes successfully, but the**second statement fails and gives an error that says: CHECK constraint is violated for key Age.*

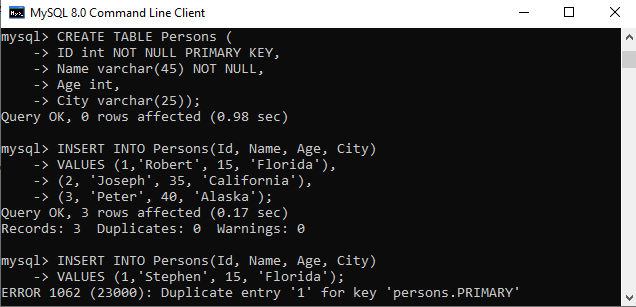


PRIMARY KEY CONSTRAINT:

**MYSQL>**CREATE TABLE Persons(ID int NOT NULLPRIMARY KEY,Name varchar(45)NO T NULL, Age int,City varchar(25));

**MYSQL>**INSERT INTO Persons(Id, Name, Age, City) VALUES (1,'Robert', 15, 'Florida') (2,'Joseph',35, 'California'), (3,'Peter', 40, 'Alaska');

**MYSQL>**INSERT INTO Persons(Id, Name, Age, City) VALUES (1,'Stephen', 15, 'Florida');



**FOREIGN KEYAND REFERENTIAL INTEGRITY CONSTRAINT**

**DEPARTMENT :**

**MYSQL>**CREATE TABLE Department (Id INT PRIMARY KEY, Name VARCHAR (50));

***--Insert some test data in Department Table***

**MYSQL>**Insert into Department values (10, 'IT');

**MYSQL>**Insert into Department values (20, 'HR'); Insert into Department values (30, 'INFRA');

**EMPLOYEES :**

**MYSQL>**CREATE TABLE Employees (Id INT PRIMARY KEY,Name VARCHAR(100) NOT NULL, Department ID INT);

***-- Adding the Foreign Key Constraint***

**MYSQL>**ALTER TABLE Employees ADD FOREIGN KEY (DepartmentId) REFERENCES  
Department(Id);

***-- Insert some test data in Employees Table***

**MYSQL>**INSERT into Employees VALUES (101, 'Anurag', 10);  
**MYSQL>**INSERT into Employees VALUES (102, 'Pranaya', 20);  
**MYSQL>**INSERT into Employees VALUES (103, 'Hina', 30);

***Delete from Parent Table***

**MYSQL>**DELETE FROM Department WHERE Id = 10;

***Output :***

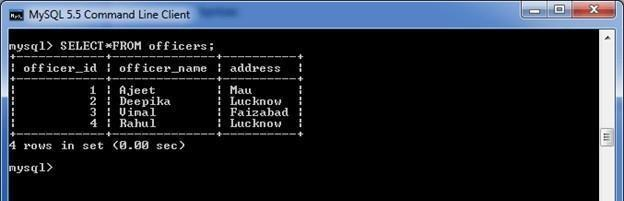
**MYSQL>**DELETE from Department where Id=10;

*ERROR 1451 (23000): Cannot delete or update a parent row: a foreign key constraint fails  
(`db2`.`employees`, CONSTRAINT `employees\_ibfk\_1` FOREIGN KEY (`DepartmentID`) REFERENCES  
`department` (`Id`))*

**QUERIES WITH WHERE CLAUSE AND AGGREATE FUNCTIONS.**

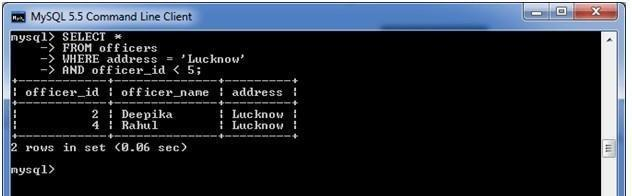
**MySQL WHERE Clause Syntax:**

**MYSQL>**Select \* from Tablename WHERE conditions;



**MySQL WHERE Clause with AND condition**

**MYSQL>**SELECT \* FROM officers WHERE address = 'Lucknow' AND officer\_id< 5;



**WHERE Clause with OR condition :**

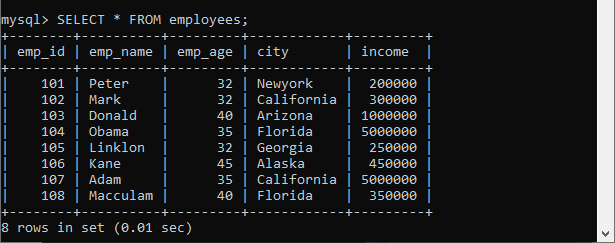
**MYSQL>**SELECT \* FROM officers WHERE address = 'Lucknow' OR address = 'Mau'  
MySQL WHERE Clause with combination of AND & OR conditions

**MYSQL>**SELECT \* FROM officers WHERE (address = 'Mau' AND officer\_name = 'Ajeet')  
OR (officer\_id< 5)



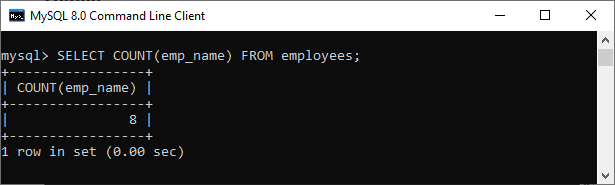
**AGGREGATE FUNCTIONS**

Consider a table named "employees" that contains the following data.



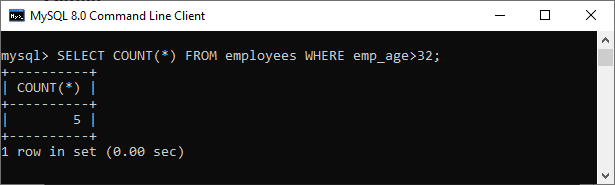
MYSQL>SELECT COUNT(emp\_name) FROM employees;

Output:



MySQL> SELECT COUNT(\*) FROM employees WHERE emp\_age>32;

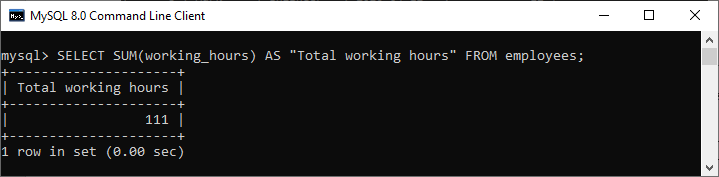
Output:



*Consider our database has a table name employees, having the following data. Now, we are going  
to understand this function with various example*

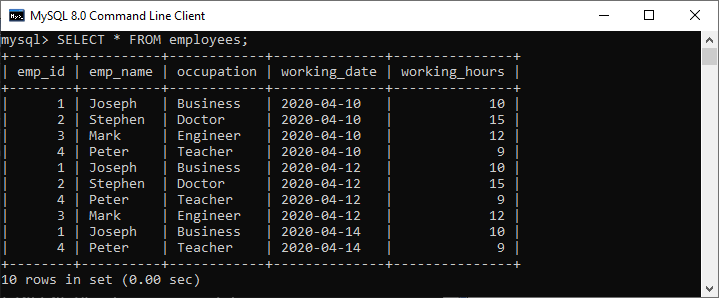
MySQL> SELECT SUM(working\_hours) AS "Total working hours" FROM employees;

Output:



MySQL avg() function example :

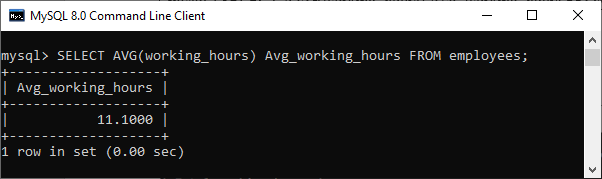
*Consider our database has a table name employees, having the following data. Now, we are going to**understand this function with variousexamples:*



MySQL> SELECT AVG(working hours) Avg\_working\_hours FROM employees;

Output:

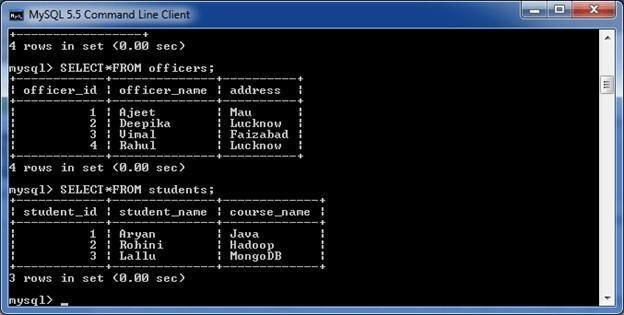
*We will get the result as below:*



**SIMPLE JOIN AND SUBQUERIES**

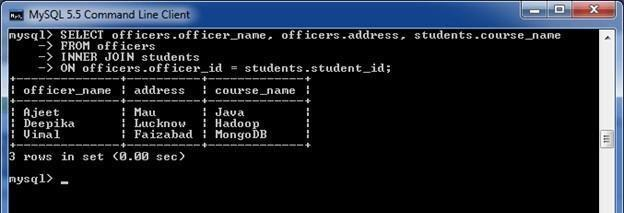
**MYSQL INNER JOIN (SIMPLE JOIN):**

Consider two tables "officers" and "students", having the following data.

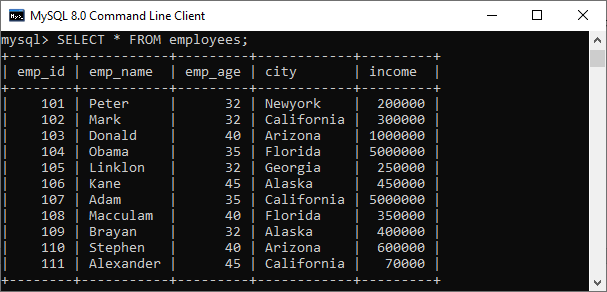


**MYSQL>** SELECT officers.officer\_name, officers.address, students.course\_name FROM officers INNER JOIN students ON officers.officer\_id =students.student\_id;

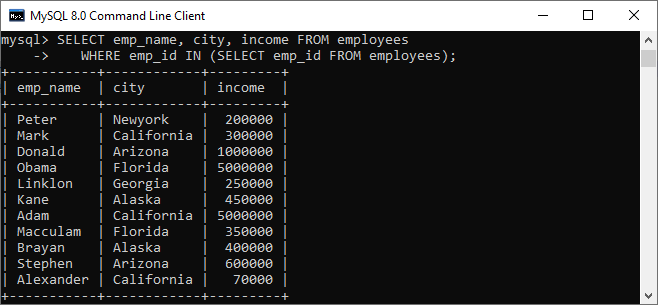
**Output:**



**MYSQL SUBQUERY**



MYSQL>SELECT emp\_name, city, income FROM employees WHERE emp\_id IN (SELECT e mp\_id FROMemployees);



**NATURAL JOIN,EQUI JOIN AND OUTERJOIN**

**Syntax:**

**MYSQL>**SELECT [column\_names | \*] FROM table\_name1 NATURAL JOIN table\_name2;

***/\* -- Table name: customer -\*/***

**MYSQL>**CREATE TABLEcustomer(idINTAUTO\_INCREMENTPRIMARYKEY,  
customer\_name VARCHAR(55), account int, email VARCHAR(55) );

***/\* -- Table name: balance -\*/***

**MYSQL>**CREATE TABLE balance ( id INT AUTO\_INCREMENT PRIMARY KEY,  
accountint, balance FLOAT(10, 2) );

***/\* -- Data for customer table -\*/***

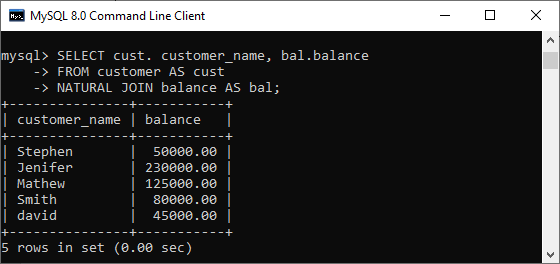
**MYSQL>**INSERT INTO customer(customer\_name, account, email) VALUES('Stephen', 1030,'stephen  
@javatpoint.com'), ('Jenifer', 2035, 'jenifer@javatpoint.com'), ('Mathew', 5564, 'mathew@java  
tpoint.com'), ('Smith', 4534, 'smith@javatpoint.com'), ('David', 7648,'david@javatpoint.com');

***/\* -- Data for balance table -\*/***

**MYSQL>**INSERT INTO balance(account, balance)  
VALUES(1030, 50000.00), (2035, 230000.00), (5564, 125000.00), (4534, 80000.00),  
(7648, 45000.00);

**NATURAL JOIN:**

**MySQL>** SELECT cust.customer\_name, bal.balance FROM customer AS cust NATURAL JOIN balance AS bal;

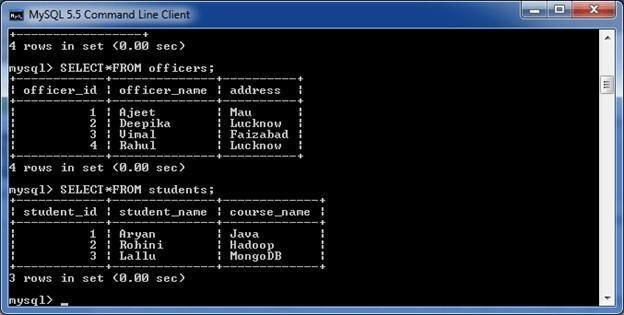


**MYSQL RIGHT OUTER JOIN**

**Syntax:**

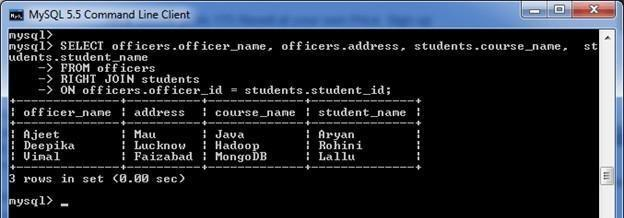
**MYSQL>**SELECT columns FROM table1 RIGHT [OUTER] JOIN table2 ON table1.column = table2.column;

*Consider two tables "officers" and "students", having the following data.*



MySQL>SELECT officers.officer\_name, officers.address, students.course\_name, students.student\_nameFROM officers RIGHT JOIN students ON officers.officer\_id = students.student\_id;

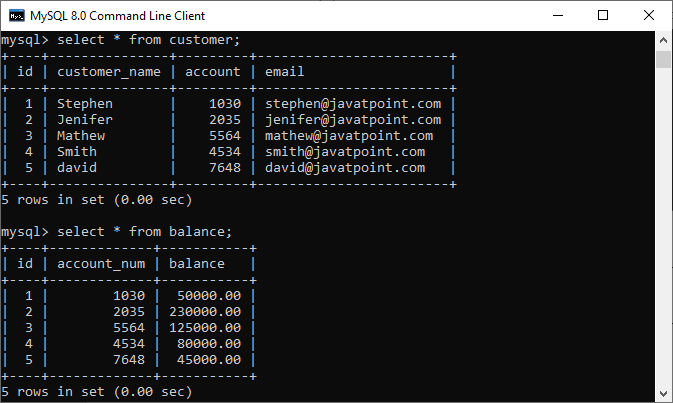
Output



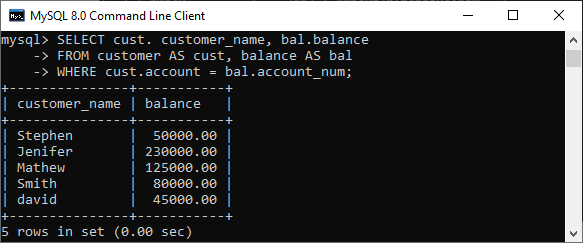
EQUI JOIN :

**MYSQL>**SELECT column\_name (s) FROMtable\_name1,table\_name2, ................,table\_nameN  
WHERE table\_name1.column\_name = table\_name2.column\_name;

*Consider two tables named customer and balance*



MySQL>SELECT cust.customer\_name, bal.balanceFROM customer AS cust, balance AS balWHERE cust.account = bal.account\_num;



**PROCEDURE AND FUNCTION**

**SETTING SERVEROUTPUT ON:**

**SQL>** SET SERVEROUTPUT ON

**PROGRAM:**

*PROCEDURE USING POSITIONAL PARAMETERS:*

**SQL>** SET SERVEROUTPUT ON  
**SQL>** CREATE OR REPLACE PROCEDURE PROC

1 AS

2 BEGIN  
3 DBMS\_OUTPUT.PUT\_LINE('Hello fromprocedure...');  
4 END;  
5/

**Output:**

***Procedure created.***

**MYSQL>** EXECUTE PROC1

Hello from procedure...  
PL/SQL procedure successfully completed.

**MYSQL>** create table student(regno number(4),name varchar2)20),mark1 number(3), mark2 number(3), mark3 number(3), mark4 number(3), mark5 number(3));

**Table created**

**MYSQL>** insert into student values (101,'priya', 78, 88,77,60,89);

**1 row created.**

**MYSQL>** insert into student values (102,'surya', 99,77,69,81,99);

**1 row created.**

**MYSQL>** insert into student values (103,'suryapriya', 100,90,97,89,91);

**1 row created.**

**MYSQL>** select \* from student;

regno name mark1 mark2 mark3 mark4 mark5  
101 priya 78 88 77 60 89  
102 surya 99 77 69 81 99

103 suryapriya 100 90 97 89 91

|  |
| --- |
| **SQL>** declare  2 avenumber(5,2); 3 totnumber(3); 4 cursorc\_mark isselect\*fromstudentwheremark1>=40andmark2>=40 and 5 mark3>=40 and mark4>=40 andmark5>=40; 6 begin 7 dbms\_output.put\_line('regnonamemark1mark2mark3mark4mark4mark5total 8 average'); 9 dbms\_output.put\_line(' '); 10 for student inc\_mark 11 loop 12 tot:=student.mark1+student.mark2+student.mark3+student.mark4+student.mark5; 13 ave:=tot/5; 14 dbms\_output.put\_line(student.regno||rpad(student.name,15) 15 ||rpad(student.mark1,6)||rpad(student.mark2,6)||rpad(student.mark3,6) 16 ||rpad(student.mark4,6)||rpad(student.mark5,6)||rpad(tot,8)||rpad(ave,5)); 17 endloop; 18 end; 19 / |

**Output**

regno name mark1 mark2 mark3 mark4 mark5 total average  
101 priya 78 88 77 60 89 393 79  
102 surya 99 77 69 81 99 425 85  
103 suryapriya 100 90 97 89 91 467 93

FUNCTIONS

**MYSQL>**create table phonebook (phone\_no number (6) primary key,username varchar2(30),doorno

varchar2(10),street varchar2(30),place varchar2(30),pincode char(6));

***Table created.***  
**MYSQL>** insert into phonebook values(20312,'vijay','120/5D','bharathi street','NGO colony','629002');

***1 row created.***

**MYSQL>**insert into phonebook values(29467,'vasanth','39D4','RK bhavan','sarakkal vilai','629002');

***1 row created.***

**MYSQL>**select \* from phonebook;

PHONE\_NOUSERNAME DOORNO STREET PLACE PINCODE:

20312 vijay 120/5D bharathistreet NGOcolony 629002  
29467 vasanth 39D4 RKbhavan sarakkalvilai 629002

**MYSQL>** create or replace function findAddress(phone in number) return varchar2 as address varchar2(100);

beginselect username||','||doorno ||','||street ||','||place||','||pincode into address from phonebook  
where phone\_no=phone;  
return address;  
exception  
whenno\_data\_found then return 'address not found';  
end;

/Function created  
**SQL>**declare  
2 addressvarchar2(100);  
3 begin  
4 address:=findaddress(20312);  
5 dbms\_output.put\_line(address);  
6 end;  
7 /

**Output**:

Vijay,120/5D,bharathi street,NGO colony,629002.

**DCL AND TCL COMMANDS**

**DCL COMMANDS GRANT:**GRANT privilege\_name ON object\_name TO {user\_name |PUBLIC |role\_name}[WITH GRANT OPTION];

**MYSQL>**GRANT SELECT ON employee TO user1;

Command Successfully Completed

REVOKE

REVOKE privilege\_name ON object\_name FROM {user\_name |PUBLIC |role\_name}

**MYSQL>**REVOKE SELECT ON employee FROMuser1;

***Command Successfully Completed***

**TCL(TRNSACTION CONTROL LANGUAGE):**

**MYSQL>** SAVEPOINT S1;

***Savepoint created.***

**MYSQL>** SELECT \* FROM EMP;

EMPNO ENAME DESIGNATIN SALARY  
101 NAGARAJAN LECTURER 16000  
102 SARAVANAN ASST. PROF 16000  
104 CHINNI HOD, PROF 45000

**MYSQL>** INSERT INTO EMP VALUES(105,'PARTHASAR','STUDENT',100);

***1 row created.***

**MYSQL>** SELECT \* FROM EMP;

|  |
| --- |
| EMPNO ENAME DESIGNATIN SALARY105 PARTHASAR STUDENT 100101 NAGARAJAN LECTURER 16000102 SARAVANAN ASST. PROF 16000104 CHINNI HOD, PROF45000 |

**ROLL BACK:**

|  |
| --- |
| **MYSQL>** ROLL BACK S1;  ***Rollback complete.***  **MYSQL>** SELECT \* FROM EMP; |
| EMPNO ENAME DESIGNATIN SALARY 101 NAGARAJAN LECTURER 16000 102 SARAVANAN ASST. PROF 16000 104 CHINNI HOD, PROF 45000 |
|  |
| COMMIT  **MYSQL>** COMMIT;  ***Commit complete.*** |
|  |
| CREATION OF DATABASETRIGGERS |
|  |

**SYNTAX**

create or replace trigger trigger name [before/after] {DML statements} on [table name] [for each row/statement] begin exception end;

|  |
| --- |
| **PROGRAM**  **MYSQL>**createtablepoo(rnonumber(5),namevarchar2(10));  ***Tablecreated.***  **MYSQL>**insert into poo values(01.‟kala‟);  ***1 row created.***  **MYSQL>**select \* from poo;  RNO NAME  ------ ----------  1 kala  2 priya  **MYSQL>**create or replace trigger pool before insert on poo for each row 2 declare 3 rnopoo.rno%type 4 cursor c is select rno frompoo; 5begin 6 openc; 7 loop; 8 fetch c intorno; 9 if:new.rno=rnothen 10 raise\_application\_error(-20005,‟rno already exist‟);  11 endif; 12 exit when c%NOTFOUND 13 end loop; 14 closec; 15 end; 16 /  ***Trigger created***  **OUTPUT :**  **MYSQL>**insert into poo values(01,‟kala‟) Insert into poo values (01,‟kala‟)  \*ERROR at line1: ORA-20005:rno already exist ORA-06512:”SECONDCSEA.POOL”,line 9 ORA-04088:error during execution at trigger “SECONDCSEA.POOL;  VIEWS AND INDEX |

**CREATION OF TABLE**

**MYSQL>**CREATE TABLE EMPLOYEE ( EMPLOYEE\_NAMEVARCHAR2(10), EMPLOYEE\_NONUMBER(8), DEPT\_NAMEVARCHAR2(10),DEPT\_NO NUMBER (5),DATE\_OF\_JOIN DATE);

***Table created.***

**TABLE DESCRIPTION**

**MYSQL>** DESC EMPLOYEE;

NAME NULL? TYPE

EMPLOYEE\_NAME VARCHAR2(10)  
EMPLOYEE\_NO NUMBER(8)  
DEPT\_NAME VARCHAR2(10)  
DEPT\_NO NUMBER(5)  
DATE\_OF\_JOIN DATE

**CREATION OF VIEW**

**MYSQL>** CREATE VIEW EMPVIEW AS SELECT  
EMPLOYEE\_NAME,EMPLOYEE\_NO,DEPT\_NAME,DEPT\_NO,DATE\_OF\_JOIN FROM EMPLOYEE;

***view created.***

**DESCRIPTION OF VIEW**

**MYSQL>** DESC EMPVIEW;

NAME NULL? TYPE  
EMPLOYEE\_NAME VARCHAR2(10)EMPLOYEE\_NO NUMBER(8)DEPT\_NAME VARCHAR2(10)DEPT\_NO NUMBER(5)DATE\_OF\_JOIN DATE

**DISPLAY VIEW:**

**MYSQL>** SELECT \* FROM EMPVIEW;

EMPLOYEE\_N EMPLOYEE\_NO DEPT\_NAME DEPT\_NO

RAVI 124 ECE 89VIJAY 345 CSE 21RAJ 98 IT 22GIRI 100 CSE 67

**INSERTION INTO VIEW:**

**MYSQL>** INSERT INTO EMPVIEW VALUES ('SRI', 120,'CSE', 67,'16-NOV-1981');

***1 ROW CREATED.***

**MYSQL>** SELECT \* FROM EMPVIEW;

EMPLOYEE\_N EMPLOYEE\_NO DEPT\_NAME DEPT\_NO  
RAVI 124 ECE 89  
VIJAY 345 CSE 21  
RAJ 98 IT 22  
GIRI 100 CSE 67  
SRI 120 CSE 67