**Title of the Experiment**

**EXCERSICE-2-COMPANY DATABASE:**

Study & Implementation of different types of constraints with suitable examples.

1. **PRIMARY KEY,**
2. **FOREIGN KEY,**
3. **NOT NULL,**
4. **UNIQUE,**
5. **CHECK,**
6. **DEFAULT,**

**are the different types of constraints in SQL.**

1. **CREATE DATABASE**

**SYNTAX:**

CREATE DATABASE DBMS2025;

**OUTPUT:**

A black background with white text

AI-generated content may be incorrect.

1. **DISPLAY DATABASES**

**SYNTAX:**

SHOW DATABASES;

**OUTPUT:**



1. **USE DATABASE**

SYNTAX:

USE DBMS2025;

**OUTPUT:**

A black and white sign with white text

AI-generated content may be incorrect.

1. **CREATE TABLE EMPLOYEE:**

**SYNTAX:**

CREATE TABLE employee

(

fname varchar(10) not null,

minit varchar(4),

lname varchar(10),

ssn integer(10) primary key,

bdate date,

address varchar(25),

sex char(1),

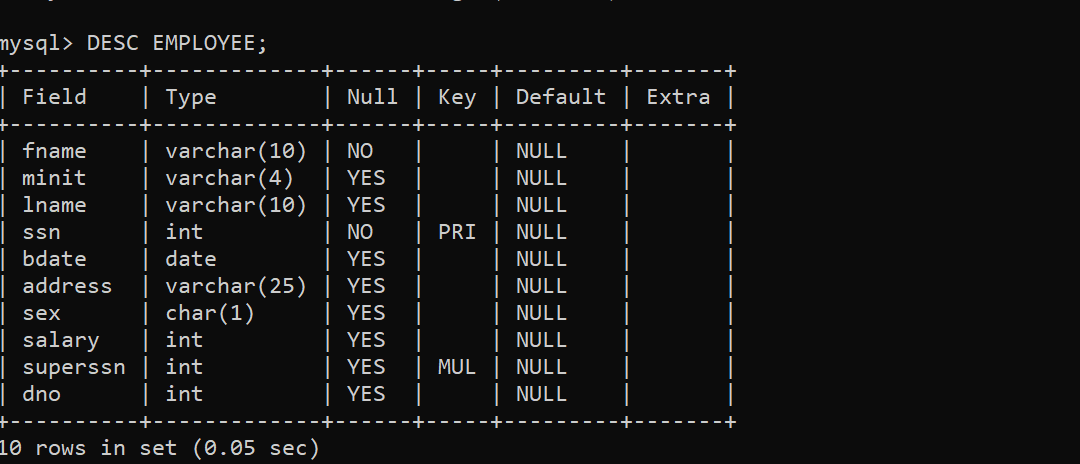
salary int(10),

superssn integer(10),

dno int(1), foreign key(superssn) references employee(ssn)

);

**OUTPUT:**



**5. CREATE TABLE DEPARTMENT:**

**SYNTAX:**

create table department

(dname varchar(20),

dnumber int(1) primary key,

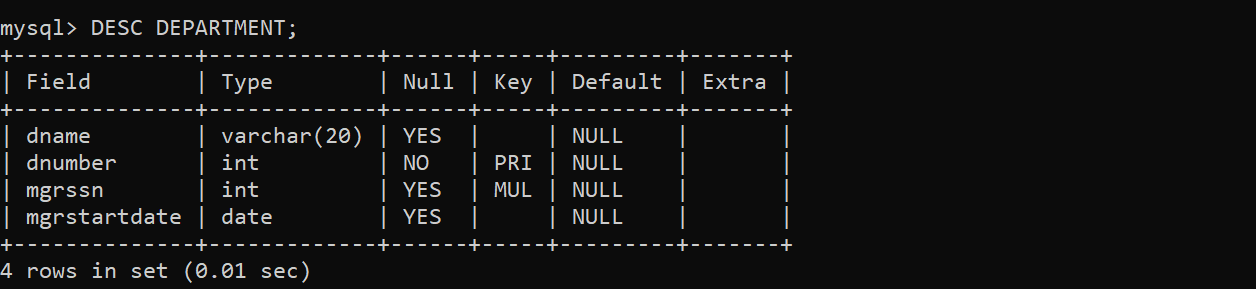
mgrssn integer(10),

mgrstartdate date,

foreign key(mgrssn) references employee(ssn)

);

**OUTPUT:**



**6. CREATE TABLE DEPARTMENT LOCATIONS:**

**SYNTAX:**

create table dept\_locations

(

dnumber int(1),

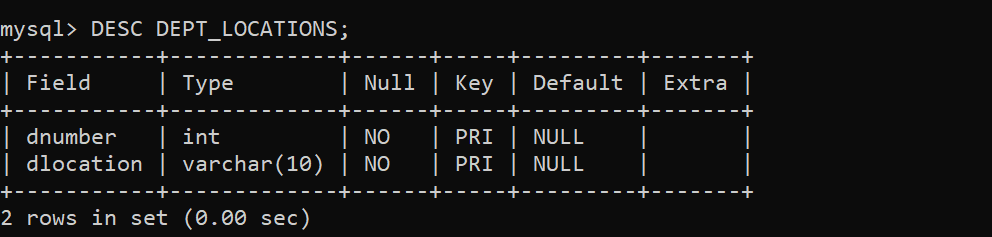
dlocation varchar(10),

foreign key(dnumber) references department(dnumber),

primary key(dnumber,dlocation)

);

**OUTPUT:**

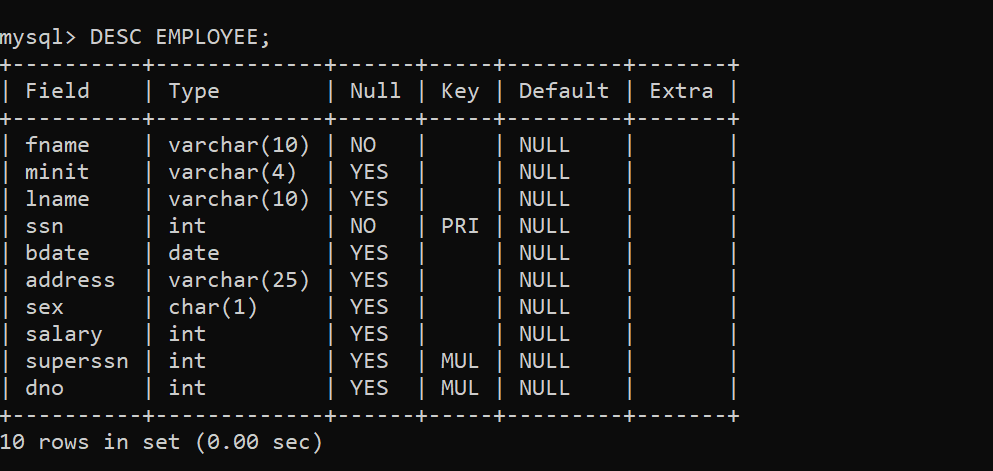


**7. ALTER THE TABLE EMPLOYEE TO ADD FOREIGN KEY:**

**SYNTAX:**

alter table employee add foreign key(dno) references department(dnumber);

**OUTPUT:**



**8. CREATE TABLE PROJECT:**

**SYNTAX:**

create table project

(

pname varchar(20) not null,

pnumber int(4) primary key,

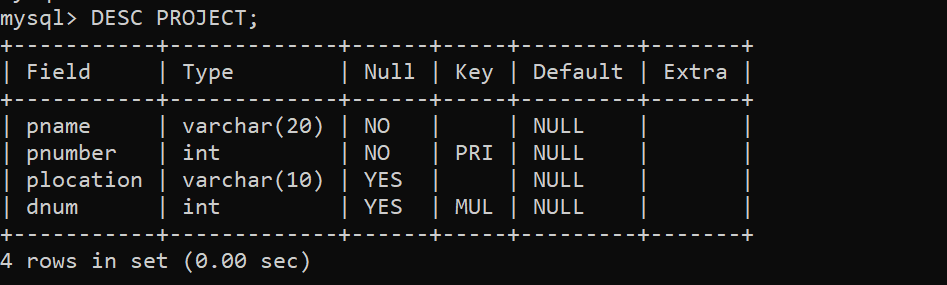
plocation varchar(10),

dnum int(1),

foreign key(dnum) references department(dnumber)

);

**OUTPUT:**



**9. CREATE TABLE WORKS\_ON:**

**SYNTAX:**

create table works\_on

(

essn int(10),

pno int(4),hours float,

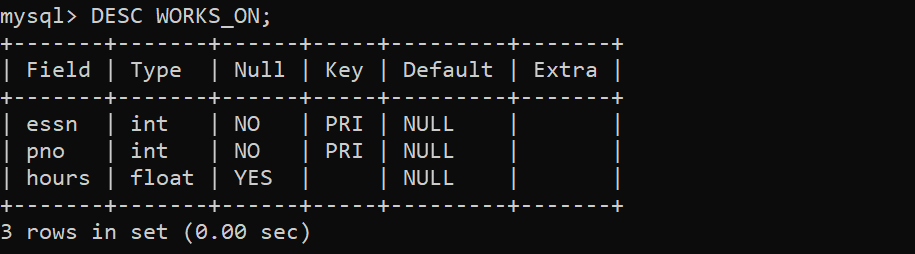
foreign key(essn) references employee(ssn),

foreign key(pno) references project(pnumber),

primary key(essn,pno)

);

**OUTPUT:**



**10. CREATE TABLE DEPENDENT:**

**SYNTAX:**

create table dependent

(essn integer (10),

dependent\_name varchar (15),

sex char (1), bdate date,

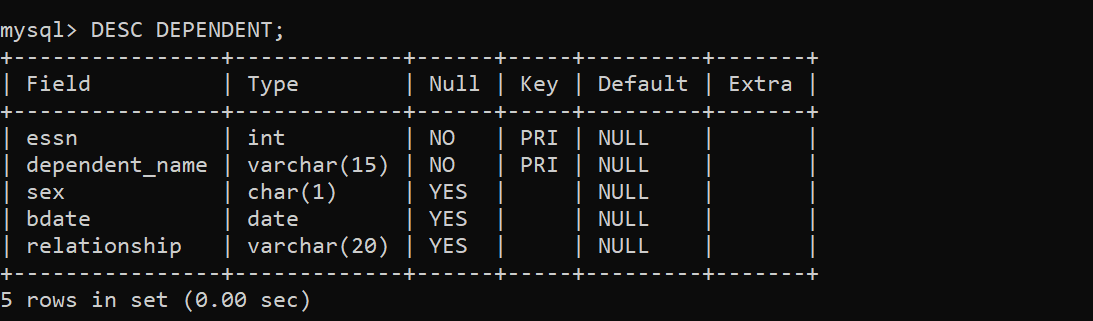
relationship varchar (20),

foreign key(essn) references employee(ssn),

primary key (essn, dependent\_name)

);

**OUTPUT:**



**11. INSERT VALUES INTO EMPLOYEE TABLE:**

**SYNTAX:**

insert into employee(Fname, Minit, Lname, Ssn, Bdate, Address, sex, Salary) values

('John','B','Smith', 123456789,'1965-01-09','731-Fondren-Houston-TX','M', 30000),

('Franklin', 'T', 'Wong', 333445555, '1955-12-08', '638-Voss-Houston-TX','M', 40000),

('Alicia', 'J', 'Zelaya', 999887777, '1968-01-19', '3321-Castle-Spring-TX', 'F', 25000),

('Jennifer', 'S', 'Wallace', 987654321, '1941-06-20', '291-Berry-Bellaire-TX', 'F', 43000),

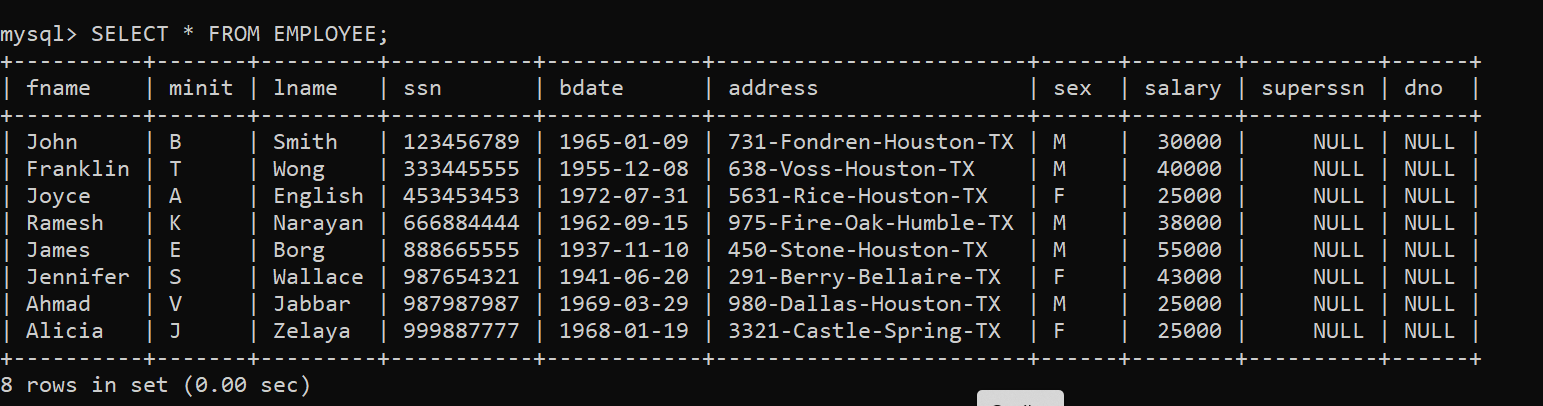
('Ramesh', 'K', 'Narayan', 666884444, '1962-09-15', '975-Fire-Oak-Humble-TX', 'M', 38000),

('Joyce','A', 'English', 453453453, '1972-07-31', '5631-Rice-Houston-TX','F', 25000),

('Ahmad','V', 'Jabbar', 987987987, '1969-03-29', '980-Dallas-Houston-TX', 'M', 25000),

('James', 'E', 'Borg', 888665555, '1937-11-10', '450-Stone-Houston-TX', 'M', 55000);

**OUTPUT:**



**12. INSERT VALUES INTO DEPARTMENT TABLE:**

**SYNTAX:**

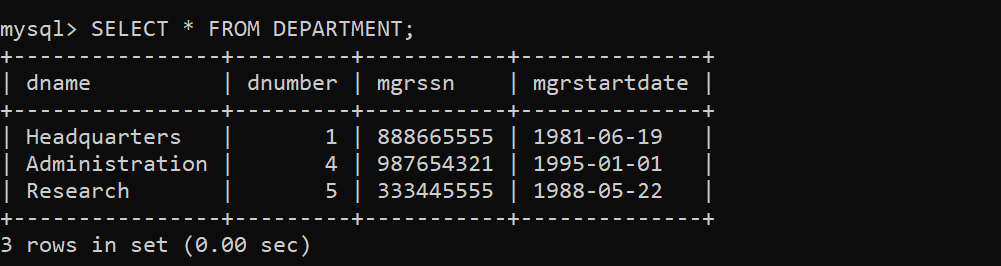
insert into department values

('Research', 5, 333445555, '1988-05-22'),

('Administration', 4, 987654321, '1995-01-01'),

('Headquarters', 1, 888665555, '1981-06-19');

**OUTPUT:**



**13. INSERT VALUES INTO DEPARTMENT LOCATIONS TABLE:**

**SYNTAX:**

insert into dept\_locations values

(1,'Houston'),

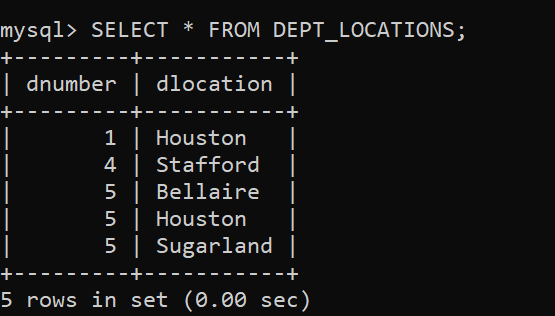
(4, 'Stafford'),

(5, 'Bellaire'),

(5, 'Sugarland'),

(5,'Houston');

**OUTPUT:**



**14. INSERT VALUES INTO PROJECT TABLE:**

**SYNTAX:**

insert into project values

('ProductX', 1, 'Bellaire', 5),

('ProductY', 2, 'Sugarland', 5),

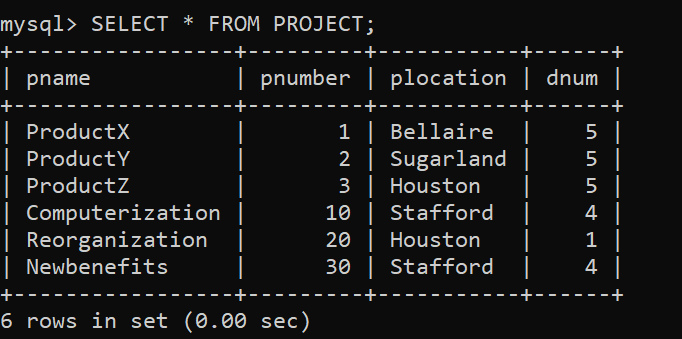
('ProductZ', 3, 'Houston', 5),

('Computerization', 10, 'Stafford', 4),

('Reorganization', 20, 'Houston', 1),

('Newbenefits', 30, 'Stafford', 4);

**OUTPUT:**



**15. INSERT VALUES INTO WORKS\_ON TABLE:**

**SYNTAX:**

insert into works\_on values

(123456789, 1, 32.5),

(123456789, 2, 7.5),

(666884444, 3, 40.0),

(453453453, 1, 20.0),

(453453453, 2, 20.0),

(333445555, 2, 10.0),

(333445555, 3, 10.0),

(333445555, 10, 10.0),

(333445555, 20, 10.0),

(999887777, 30, 30.0),

(999887777, 10, 10.0),

(987987987, 10, 35.0),

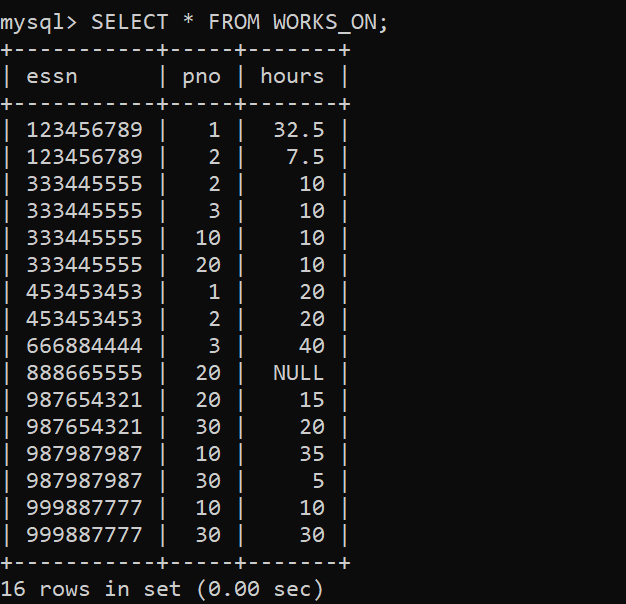
(987987987, 30, 5.0),

(987654321, 30, 20.0),

(987654321, 20, 15.0),

(888665555, 20, NULL);

**OUTPUT:**



**16. INSERT VALUES INTO DEPENDENT TABLE:**

**SYNTAX:**

insert into dependent values

(333445555,'Alice','F','1986-04-05','Daughter'),

(333445555,'Theodore','M','1983-10-25','Son'),

(333445555,'Joy','F','1958-05-03','Spouse'),

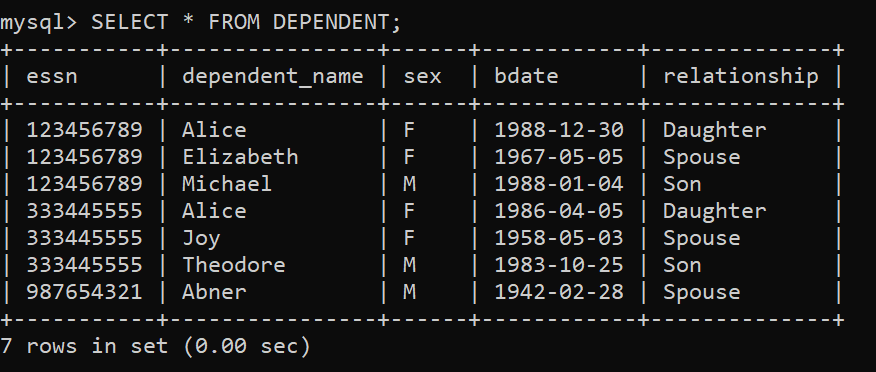
(987654321,'Abner','M','1942-02-28','Spouse'),

(123456789,'Michael','M','1988-01-04','Son'),

(123456789,'Alice','F','1988-12-30','Daughter'),

(123456789,'Elizabeth','F','1967-05-05','Spouse');

**OUTPUT:**



**17.**  **Update Employee table with Superssn and dno:**

**SYNTAX:**

update employee set superssn=333445555,dno=5 where ssn=123456789;

update employee set superssn=888665555,dno=5 where ssn=333445555;

update employee set superssn=987654321,dno=4 where ssn=999887777;

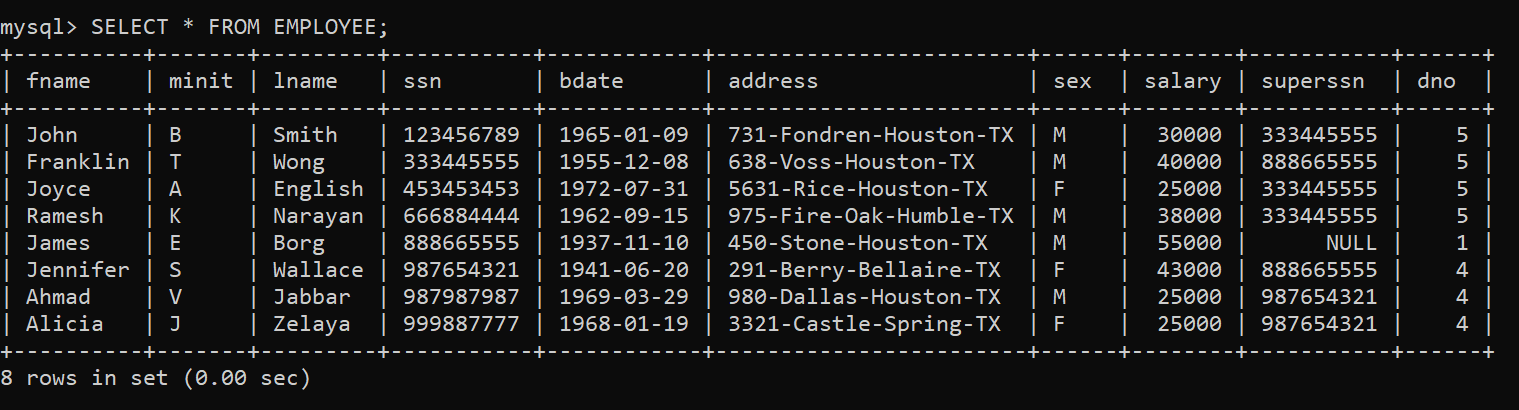
update employee set superssn=888665555,dno=4 where ssn=987654321;

update employee set superssn=333445555,dno=5 where ssn=666884444;

update employee set superssn=333445555,dno=5 where ssn=453453453;

update employee set superssn=987654321,dno=4 where ssn=987987987;

update employee set dno=1 where ssn=888665555;

**OUTPUT:** 

**18. QUERIES:**

1. **Write a query to drop a table.**

**SYNTAX:**

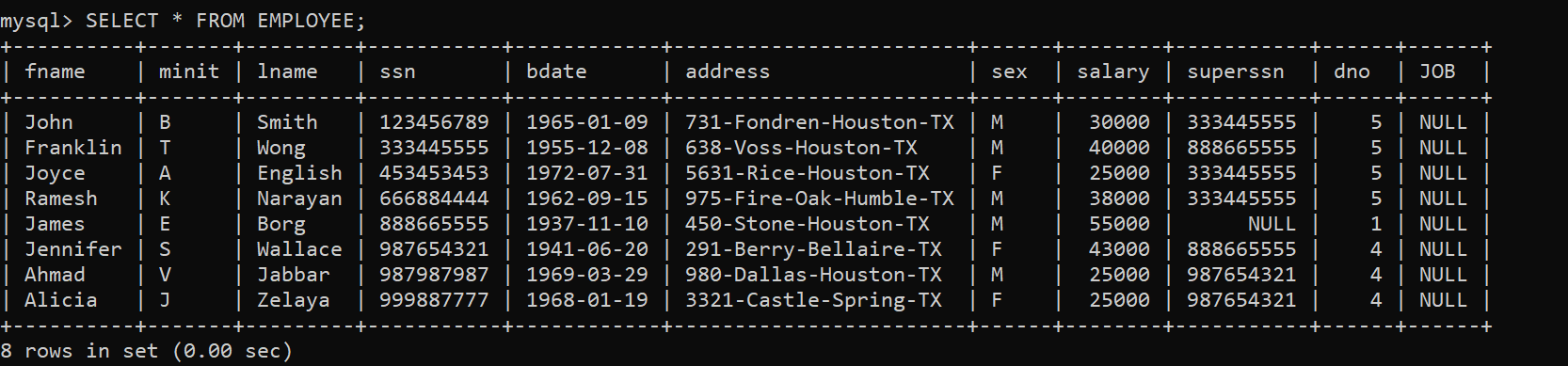
DROP TABLE DEPENDENT;

DO NOT EXECUTE

1. **Write a query to ADD new attribute JOB for EMPLOYEE table.**

**SYNTAX:**

ALTER TABLE EMPLOYEE ADD JOB VARCHAR (12);

**OUTPUT:** 

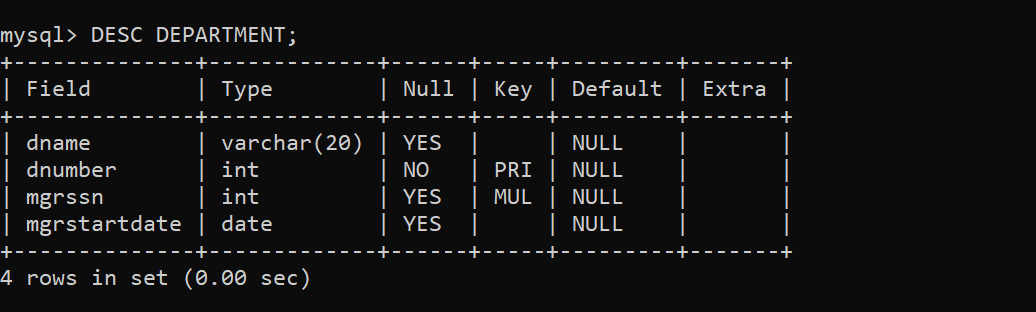
1. **. Write a query using alter command the drop the column**

**SYNTAX:**

ALTER TABLE DEPARTMENT DROP COLUMN MGRSSN;

DO NOT EXECUTE

**OUTPUT:**



1. **. Write a query to delete the employee whose Lname is Brown**

**SYNTAX:**

DELETE FROM EMPLOYEE WHERE LNAME='Brown‘;

mysql> DELETE FROM EMPLOYEE WHERE LNAME='Borg';

ERROR 1451 (23000): Cannot delete or update a parent row: a foreign key constraint

fails(`dbms1`.`department`, CONSTRAINT `department\_ibfk\_1` FOREIGN KEY

(`mgrssn`) REFERENCES `employee` (`ssn`))

1. **Write a query to delete the employee whose SSN is 123456789;**

**SYNTAX:**

DELETE FROM EMPLOYEE WHERE SSN='123456789‘;

mysql> DELETE FROM EMPLOYEE WHERE SSN='123456789';

ERROR 1451 (23000): Cannot delete or update a parent row: a foreign key constraint fails (`dbms1`.`dependent`, CONSTRAINT `dependent\_ibfk\_1` FOREIGN KEY (`essn`) REFERENCES `employee` (`ssn`))

1. **Write a query to delete the employee who is working for Research department.**

**SYNTAX:**

DELETE FROM EMPLOYEE WHERE DNO IN (SELECT DNUMBER FROM DEPARTMENT WHERE DNAME='Research');

mysql> DELETE FROM EMPLOYEE WHERE DNO IN (SELECT DNUMBER FROM DEPARTMENT WHERE DNAME='Research');

ERROR 1451 (23000): Cannot delete or update a parent row: a foreign key constraint fails (`dbms1`.`dependent`, CONSTRAINT `dependent\_ibfk\_1` FOREIGN KEY (`essn`) REFERENCES `employee` (`ssn`))

1. **Write a query to delete the employee whose SSN is 123456789;**

**SYNTAX:**

DELETE FROM EMPLOYEE WHERE SSN='123456789‘;

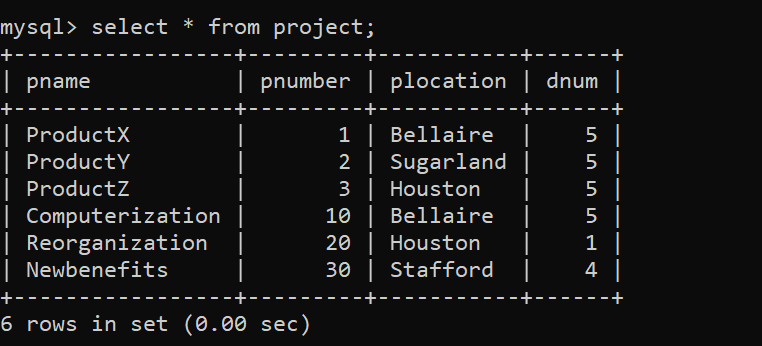
mysql> DELETE FROM EMPLOYEE WHERE SSN='123456789';

ERROR 1451 (23000): Cannot delete or update a parent row: a foreign key constraint fails (`dbms1`.`dependent`, CONSTRAINT `dependent\_ibfk\_1` FOREIGN KEY (`essn`) REFERENCES `employee` (`ssn`))

1. **. Change the location and controlling department number of project number 10 to 'Bellaire' and 5, respectively.**

**SYNTAX:**

UPDATE PROJECT SET PLOCATION = 'Bellaire', DNUM = 5 WHERE PNUMBER=10;

**OUTPUT:** 

**9. Give all employees in the 'Research' department a 10% raise in**

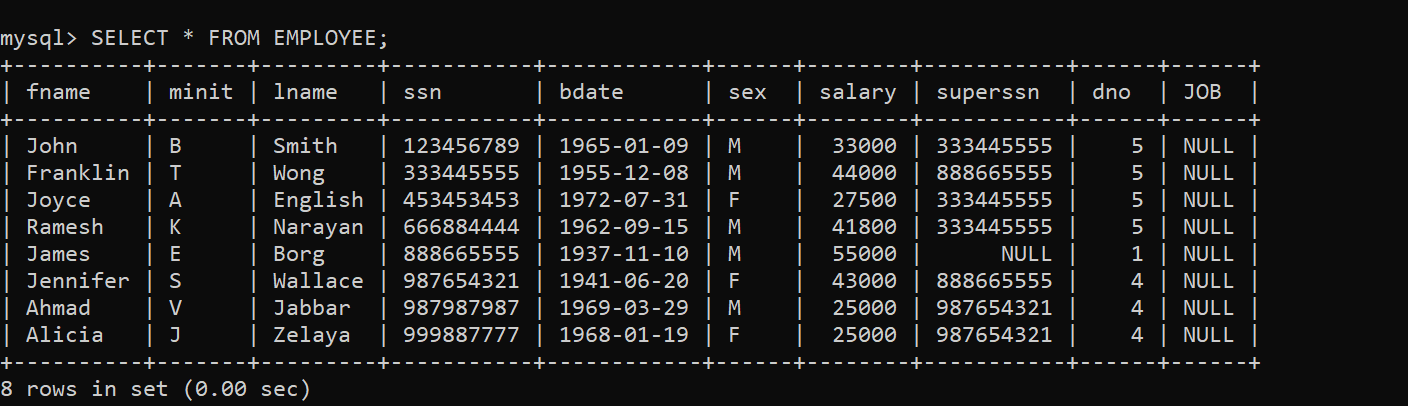
**salary**.

**SYNTAX:**

UPDATE EMPLOYEE SET SALARY = SALARY \*1.1 WHERE DNO IN (SELECT

DNUMBER FROM DEPARTMENT WHERE DNAME='Research');

**OUTPUT:**



**10. Display all the details of all employees working in the company**.

**SYNTAX:**

select \* from employee;

**OUTPUT:**A black and white screen with white text

Description automatically generated

**11. Display ssn, lname, fname, address of employees who work in department no 5.**

**SYNTAX:**

select ssn,lname,fname,address from employee where dno=5;

**OUTPUT:**

A screen shot of a computer screen

Description automatically generated

**12. Retrieve the birthdate and address of the employee whose name is 'Franklin T. Wong'**

**SYNTAX:**

select bdate, address from employee where fname="franklin" and minit="t" and

lname="wong";

**OUTPUT:**

A black screen with white text

Description automatically generated

**13. Retrieve the name and salary of every employee**

**SYNTAX:**

select fname,salary from employee;

**OUTPUT:**

A screenshot of a computer

Description automatically generated

**14. Retrieve all distinct salary values**

**SYNTAX:**

select distinct salary from employee;

**OUTPUT:**

A black screen with white text

Description automatically generated

**15. Write a Query to add primary key if not added.**

**SYNTAX:**

ALTER TABLE Employee ADD PRIMARY KEY (SSN);

**ALREADY ADDED**

**16. Write a Query to allow naming of a PRIMARY KEY constraint, and for**

**defining a PRIMARY KEY constraint on multiple columns**

**SYNTAX:**

ALTER TABLE Employee ADD CONSTRAINT PK\_Employee

PRIMARY KEY (ID,Fathername);

**ALREADY ADDED**

**17. Write a query to drop the primary key constraint.**

**SYNTAX:**

ALTER TABLE Employee DROP PRIMARY KEY;

ERROR 1553 (HY000): Cannot drop index 'PRIMARY': needed in a foreign key

constraint

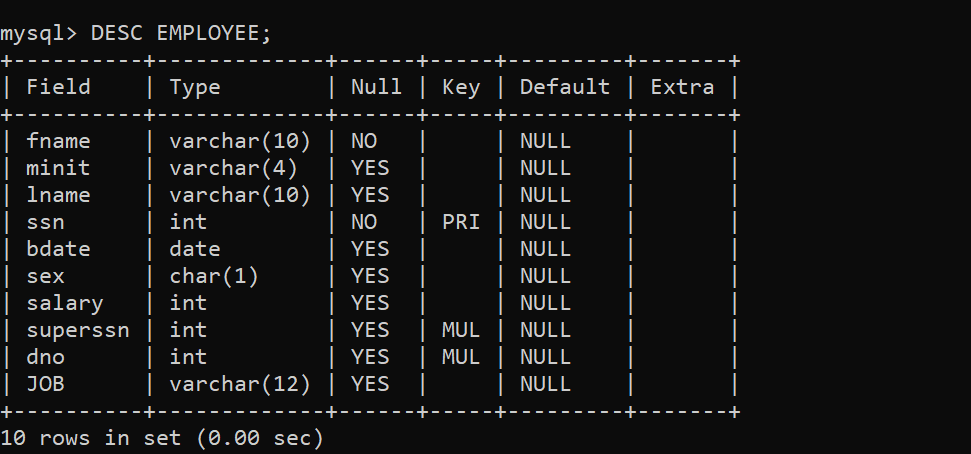
**18. Write a Query to add foreign key if not added.**

**SYNTAX:**

ALTER TABLE Employee ADD FOREIGN KEY (Dno)

  REFERENCES Department(Dnumber);

**OUTPUT:**



**19. Write a query to allow naming of a FOREIGN KEY constraint, and for defining**

**a FOREIGN KEY constraint on multiple columns:**-

**SYNTAX:**

ALTER TABLE Employee ADD CONSTRAINT FK\_Employee   
 FOREIGN KEY (Dno) REFERENCES Department(Dnumber);

DO NOT EXECUTE

**20. Write a query to drop the FOREIGN key constraint.**

**SYNTAX:**

ALTER TABLE Employee DROP FOREIGN KEY FK\_Employee;

DO NOT EXECUTE

**21. Write a query to create a CHECK constraint on the "Age" column when the table**

**is already created**.

**SYNTAX:**

ALTER TABLE Employee ADD CHECK (Salary>=80000);

ALTER TABLE Employee ADD CHECK (Salary>=80000);

ERROR 3819 (HY000): Check constraint 'employee\_chk\_1' is violated.

**22. Write a query to allow naming of a CHECK constraint, and for defining**

**a CHECK constraint on multiple columns**.

**SYNTAX:**

ALTER TABLE Employee ADD CONSTRAINT CHK\_Salary

CHECK (Salary>=80000 AND Lname='Smith');

ERROR 3819 (HY000): Check constraint 'CHK\_Salary' is violated.

**23. Write a query to drop a check constraint**

**SYNTAX:**

ALTER TABLE Employee DROP CHECK CHK\_Salary;

ERROR 3821 (HY000): Check constraint 'CHK\_Salary' is not found in the table.

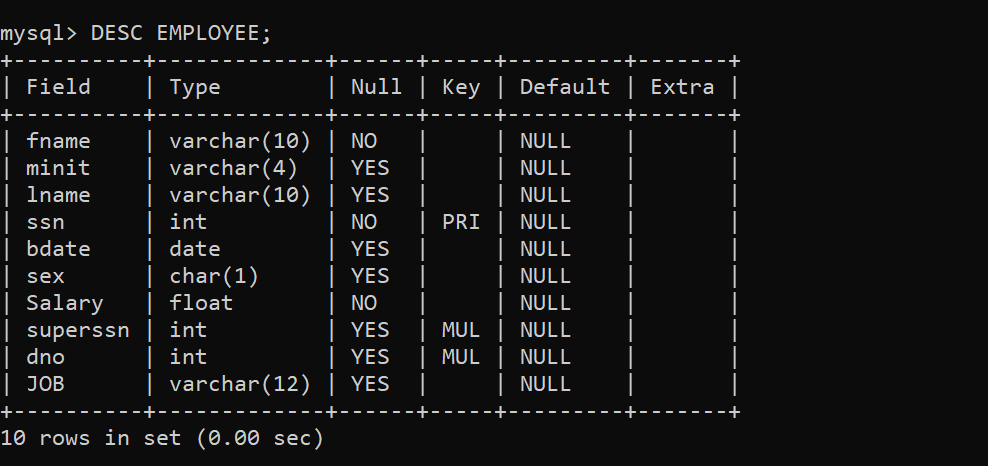
**24. Write a query to create a NOT NULL constraint on the "Salary" column when the**

**"Employee" table is already created.**

**SYNTAX:**

ALTER TABLE Employee MODIFY Salary float NOT NULL;

**OUTPUT:**



**25. Write a query to create a UNIQUE constraint on the ID column when the table is already created.**

**SYNTAX:**

ALTER TABLE Employee ADD UNIQUE (bdate);

**OUTPUT:**

