Name: Ruksana **Date:**

Roll No: 24481A12G3 **Expt. No:1**

Aim: Creation, altering and dropping of tables and inserting rows into a table(usecreat constraints while creating tables) examples using select command.

DATA DEFINITION LANGUAGE(DDL) COMMANDS

DDL is a computer language that is used for creating,modidfying and dropping the structure of database objects such as schema, tables,views,indices etc is managed by DDL.

DDL has four important commands. They are:

1. Create
2. Alter
3. Drop
4. Truncate

<u>1.create:</u>

Create cmd defines each column of table uniquely.

Each column has three attributes:

1. Name
2. Datatype
3. Size

Main use of create cmd is to build a new table and it comes from pre defined syntax. There is a component in RDBMS.

Syntax:

Create tablename(columnname1 datatype(size),columnname2 datatype(size),columnnamen datatype(size));

Query1: <u>creating table without constraints:</u>

1. Create table book(bname varchar(20),pages int,author varchar(50));

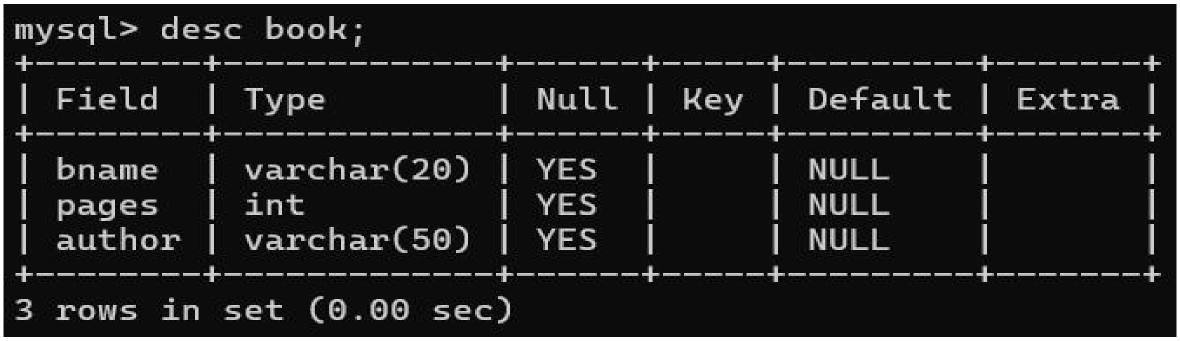
<u>Output:</u>

Table is created.

<u>Checking:</u>

->Desc book;

<u>Output:</u>



```
mysql> desc book;
+--------+-------------+------+-----+---------+-------+
| Field  | Type        | Null | Key | Default | Extra |
+--------+-------------+------+-----+---------+-------+
| bname  | varchar(20) | YES  |     | NULL    |       |
| pages  | int         | YES  |     | NULL    |       |
| author | varchar(50) | YES  |     | NULL    |       |
+--------+-------------+------+-----+---------+-------+
3 rows in set (0.00 sec)
```

Query2: Creating table with constraints:

Syntax:

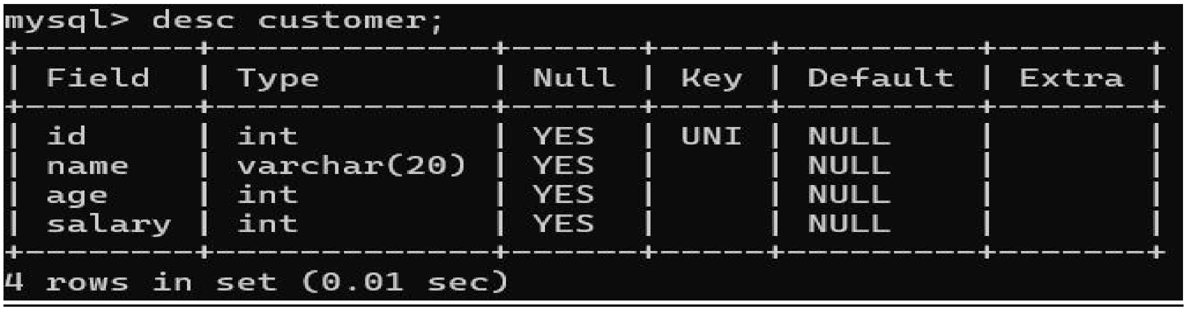
Create table tablename(columnname1 datatype(size) constraint1…columnnamen datatype(size) constraint);

Create table customer(id int unique,name varchar(20),age int,salary int);

Checking:

->Desc customer;

Output:



```
mysql> desc customer;
+--------+-------------+------+-----+---------+-------+
| Field  | Type        | Null | Key | Default | Extra |
+--------+-------------+------+-----+---------+-------+
| id     | int         | YES  | UNI | NULL    |       |
| name   | varchar(20) | YES  |     | NULL    |       |
| age    | int         | YES  |     | NULL    |       |
| salary | int         | YES  |     | NULL    |       |
+--------+-------------+------+-----+---------+-------+
4 rows in set (0.01 sec)
```

Query3: displaying constraint violations

->Insert into customer values(101,’spoorthi’,20,50000);

->Insert into customer values(101,’sana’,21,45000);

Checking:

->Select * from customer;

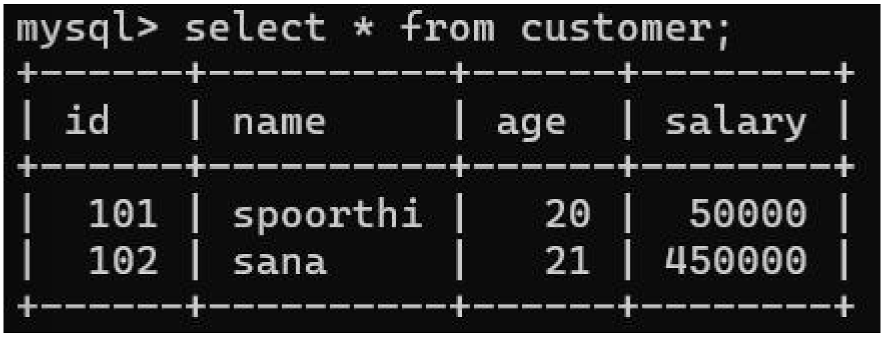
Output:

Duplicate entry '101' for key 'customer.id'

Correction:

->Insert into customer values(102,’sana’,21,45000);

Output:



```
mysql> select * from customer;
+------+----------+------+--------+
| id   | name     | age  | salary |
+------+----------+------+--------+
|  101 | spoorthi |   20 |  50000 |
|  102 | sana     |   21 | 450000 |
+------+----------+------+--------+
```

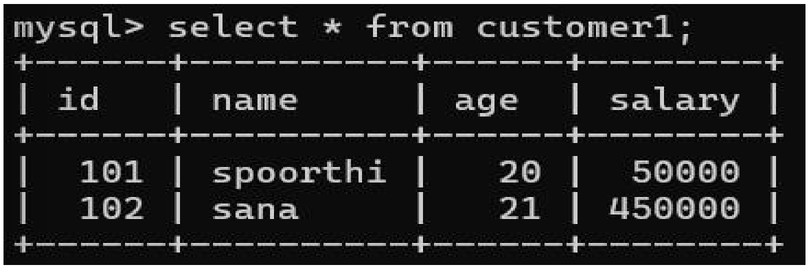
Query4: <u>creating table with existing table with data</u>

->Create table customer1 as select * from customer where 1=1;

<u>Output:</u>

Query OK, 2 rows affected (0.06 sec)

<u>Checking:</u>



```
mysql> select * from customer1;
+------+----------+------+--------+
| id   | name     | age  | salary |
+------+----------+------+--------+
|  101 | spoorthi |   20 |  50000 |
|  102 | sana     |   21 | 450000 |
+------+----------+------+--------+
```

Query5: <u>creating table with existing table without data</u>

->Create table customer2 as select * from customer where 1=2;

<u>Output:</u>

Query OK, 0 rows affected (0.07 sec)

<u>Checking:</u>

->Select * from customer2;

Empty set (0.00 sec)

<u>2.Alter:</u>

The alter command in SQL is used to modify the structure of an existing table without deleting the table or its data.

It allows us to add, delete, or modify columns and constraints in a table.

Types of alter Command

add– Add a new column

modify – Change datatype or size of a column

drop– Remove a column

rename – Rename a column or table

Syntax:

Alter table table_name
Add column_name datatype (or)
modify column_name datatype (or)
drop column_name (or)
rename column name old_name to new_name;

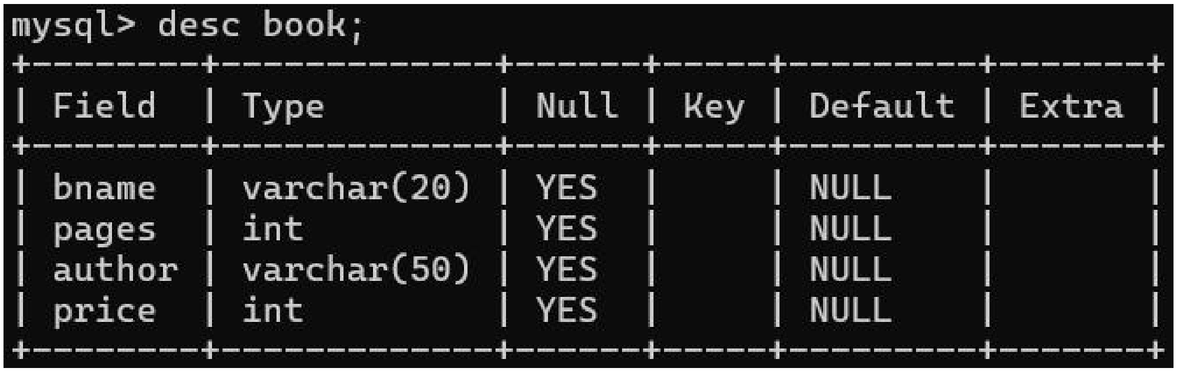
Query1: Adding new row

->alter table book add price int;

Output: Query OK, 0 rows affected (0.11 sec)

Checking:

->Desc book;



```
mysql> desc book;
+--------+-------------+------+-----+---------+-------+
| Field  | Type        | Null | Key | Default | Extra |
+--------+-------------+------+-----+---------+-------+
| bname  | varchar(20) | YES  |     | NULL    |       |
| pages  | int         | YES  |     | NULL    |       |
| author | varchar(50) | YES  |     | NULL    |       |
| price  | int         | YES  |     | NULL    |       |
+--------+-------------+------+-----+---------+-------+
```

(Added new row price)

Query2: Modify size of column name

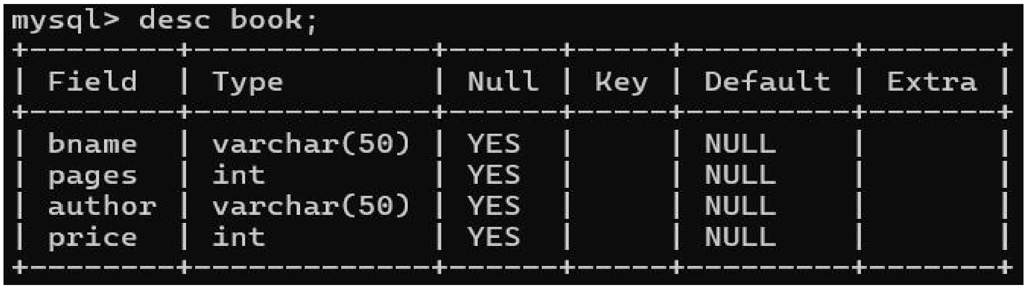
-> alter table book modify bname varchar(50);

Output:

Query OK, 0 rows affected (0.03 sec)

Checking:

->desc book;



```
mysql> desc book;
+--------+-------------+------+-----+---------+-------+
| Field  | Type        | Null | Key | Default | Extra |
+--------+-------------+------+-----+---------+-------+
| bname  | varchar(50) | YES  |     | NULL    |       |
| pages  | int         | YES  |     | NULL    |       |
| author | varchar(50) | YES  |     | NULL    |       |
| price  | int         | YES  |     | NULL    |       |
+--------+-------------+------+-----+---------+-------+
```

This changes the column size from_20 to 50.

3.Drop:

All the data in the table and structure of the table is deleted.

By the use of this cmd,users can delete an index,table or view.
You cannot rollback the drop table statement.

Syntax:

Drop table tablename;

Query1:

```
->drop table book;
```

Output:

```
Query OK, 0 rows affected (0.05 sec)
```

Checking:

```
->desc book;
Table 'mydb1.book' doesn't exist.
```

DATA MANIPULATION LANGUAGE(DML) COMMANDS

By using DML , we can insert data, update data,delete data and retrieve data from database.
It contains 4 important commands.

1.insert
2.update
3.delete
4.select

1.Insert:

It is used to add new rows of data to a table in database.
Insert a new row containing values for each column.

Syntax:

Insert into taablename(columnname1,columnname2…columnnamen)
Values(value1,value2…valuen);

Query: inserting values into the table.

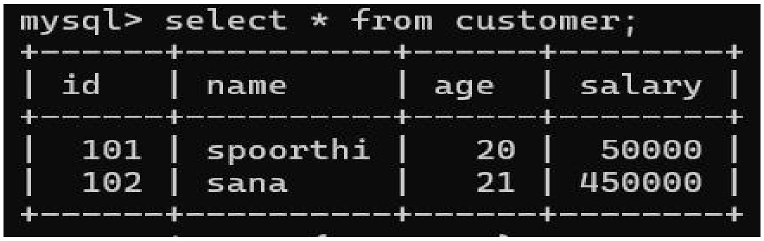
```
->insert into customer(id,name,age,salary)values(103,’priya’,22,60000);
```

Output:

```
Query OK, 1 row affected (0.03 sec)
```

Checking:

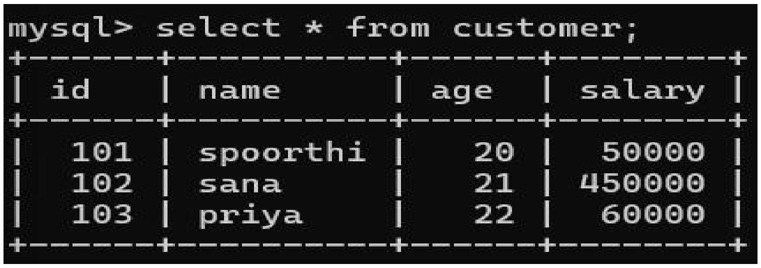
Before inserting:



```
mysql> select * from customer;
+------+----------+------+--------+
| id   | name     | age  | salary |
+------+----------+------+--------+
|  101 | spoorthi |   20 |  50000 |
|  102 | sana     |   21 | 450000 |
+------+----------+------+--------+
```

After inserting:

->select * from customer;



```
mysql> select * from customer;
+------+----------+------+--------+
| id   | name     | age  | salary |
+------+----------+------+--------+
|  101 | spoorthi |   20 |  50000 |
|  102 | sana     |   21 | 450000 |
|  103 | priya    |   22 |  60000 |
+------+----------+------+--------+
```

2.select:

The select command is used in sql to retrieve data from one or more tables in a database.

It does not change the data,it only displays it.

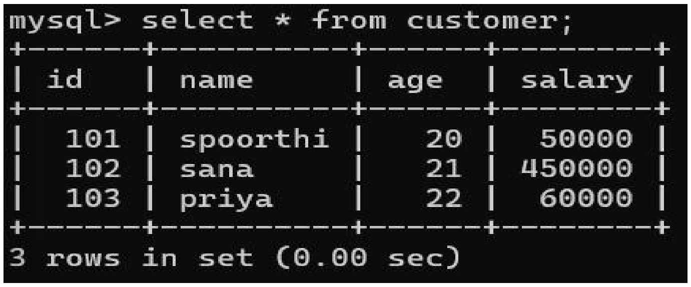
Syntax:

Select * from tablename;

Query1: select all records from a table.

->select * from customer;

Output:



```
mysql> select * from customer;
+------+----------+------+--------+
| id   | name     | age  | salary |
+------+----------+------+--------+
|  101 | spoorthi |   20 |  50000 |
|  102 | sana     |   21 | 450000 |
|  103 | priya    |   22 |  60000 |
+------+----------+------+--------+
3 rows in set (0.00 sec)
```

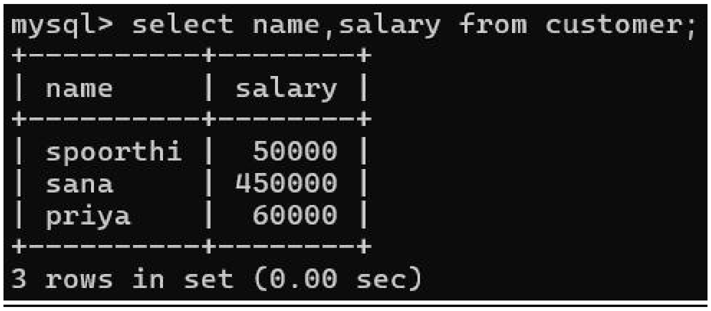
Query2: select specific coumns.

Syntax:

Select column1,column2,column3….columnn from tablename;

->select name,salary from customer;

Output:



```
mysql> select name,salary from customer;
+----------+--------+
| name     | salary |
+----------+--------+
| spoorthi |  50000 |
| sana     | 450000 |
| priya    |  60000 |
+----------+--------+
3 rows in set (0.00 sec)
```

Constraints

1. A constraint is a rule applied to table column to restrict the type of data that can be stored in it.
2. It is used to maintain data accuracy,consistency and integrity in a database.

<u>Types:</u>

1. Not null
2. Unique
3. Primary key
4. Foreign key
5. Check

1. **<u>Not Null constraint</u>**<u>:</u> ensures that a column cannot have empty values.it forces the user to always enter a value for that column.

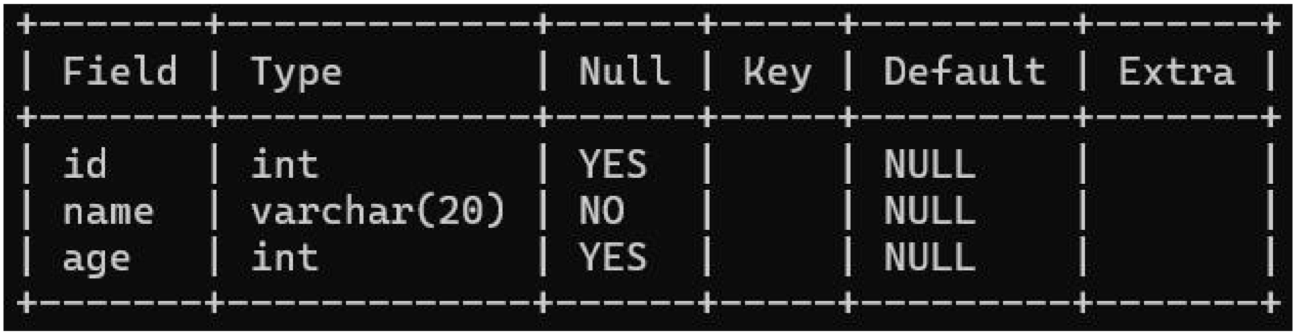
->Create table student1(id int,name varchar(20) not null,age int);

<u>Output:</u>

Query OK, 0 rows affected (0.03 sec)

<u>Checking:</u>

->Desc student1;



```
+-------+-------------+------+-----+---------+-------+
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| id    | int         | YES  |     | NULL    |       |
| name  | varchar(20) | NO   |     | NULL    |       |
| age   | int         | YES  |     | NULL    |       |
+-------+-------------+------+-----+---------+-------+
```

->insert into student1(id,age)values(1,20);

Output:

Field 'name' doesn't have a default value

Here , constraint is violated.

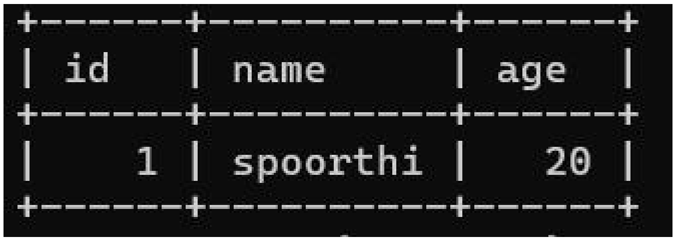
->insert into student1 values(1,"spoorthi",20);

Output:

Query OK, 1 row affected (0.01 sec)

Checking:

->select * from student1;



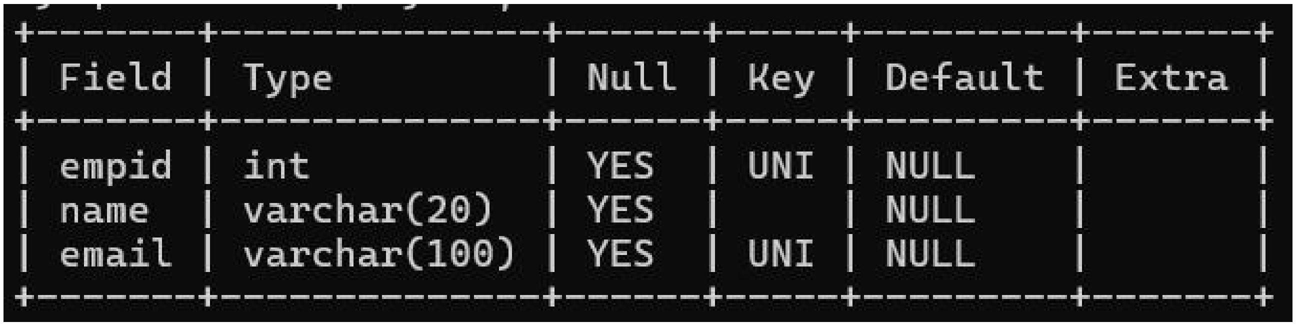
```
+------+----------+------+
| id   | name     | age  |
+------+----------+------+
|    1 | spoorthi |   20 |
+------+----------+------+
```

2. **Unique constraint**: The unique constraint ensures that all values in a column are different. It does not allow duplicate values.

->create table employees(empid int unique,name varchar(20),email varchar(100) unique);

Checking:

->desc employees;



```
+-------+--------------+------+-----+---------+-------+
| Field | Type         | Null | Key | Default | Extra |
+-------+--------------+------+-----+---------+-------+
| empid | int          | YES  | UNI | NULL    |       |
| name  | varchar(20)  | YES  |     | NULL    |       |
| email | varchar(100) | YES  | UNI | NULL    |       |
+-------+--------------+------+-----+---------+-------+
```

-> insert into employees values(11, 'Amit', 'amit@gmail.com'),(11, 'Rahul', 'rahul@gmail.com');

Output:

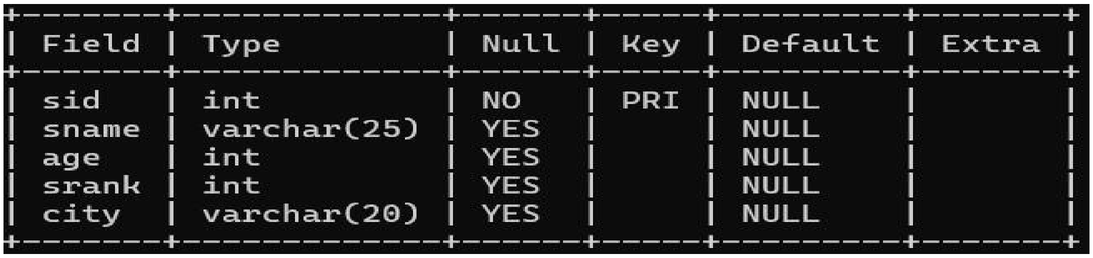
Duplicate entry 'amit@gmail.com' for key 'employees.email'

3. **Primary key:** a primary key constraint is used to uniqyely identify each record in a table. It does not allow NULL values.

->create table sailors(sid int primary key,sname varchar(25),age int,srank int,city varchar(20));

Checking:

->desc sailors;



```
+-------+-------------+------+-----+---------+-------+
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| sid   | int         | NO   | PRI | NULL    |       |
| sname | varchar(25) | YES  |     | NULL    |       |
| age   | int         | YES  |     | NULL    |       |
| srank | int         | YES  |     | NULL    |       |
| city  | varchar(20) | YES  |     | NULL    |       |
+-------+-------------+------+-----+---------+-------+
```

4. **<u>Foreign key</u>:** foreign key represents relationships between tables. It ensures that the value in one table matches a primary key value in another table,maintaining referential integrity.

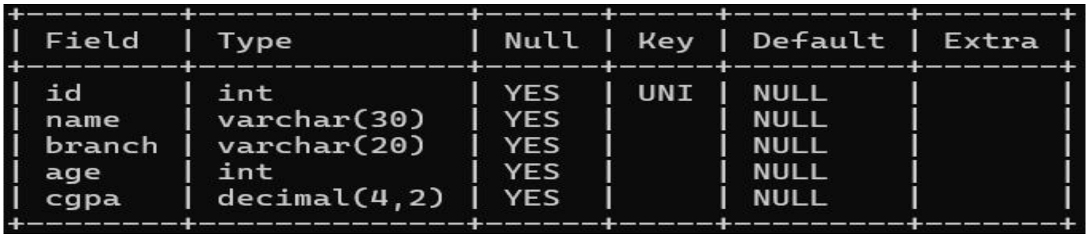
<u>Syntax:</u>

Create table child_table(column_name datatype(size),foreign key(column_name)references parent_table);

1.) ->create table students1(id int unique,name varchar(30),branch varchar(20),age int,cgpa decimal(4,2));

<u>Checking:</u>

->desc students1;

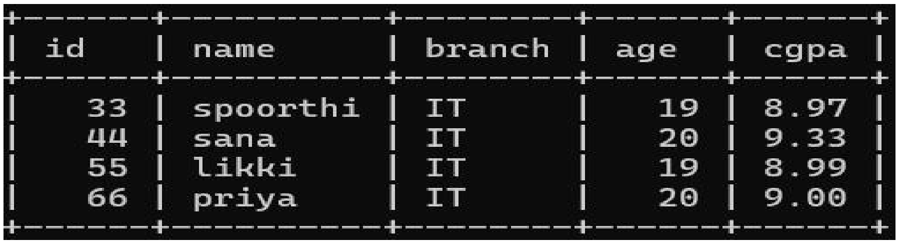


```
+--------+--------------+------+-----+---------+-------+
| Field  | Type         | Null | Key | Default | Extra |
+--------+--------------+------+-----+---------+-------+
| id     | int          | YES  | UNI | NULL    |       |
| name   | varchar(30)  | YES  |     | NULL    |       |
| branch | varchar(20)  | YES  |     | NULL    |       |
| age    | int          | YES  |     | NULL    |       |
| cgpa   | decimal(4,2) | YES  |     | NULL    |       |
+--------+--------------+------+-----+---------+-------+
```

->insert into students1 values(33,'spoorthi','IT',19,8.97),(44,'sana','IT',20,9.33),(55,'likki','IT',19,8.99),(66,'priya','IT',20,9.00);

<u>Checking:</u>

->select * from students1;

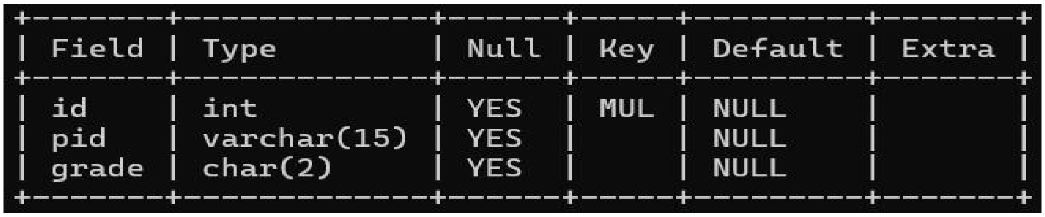


```
+------+----------+--------+------+------+
| id   | name     | branch | age  | cgpa |
+------+----------+--------+------+------+
|   33 | spoorthi | IT     |   19 | 8.97 |
|   44 | sana     | IT     |   20 | 9.33 |
|   55 | likki    | IT     |   19 | 8.99 |
|   66 | priya    | IT     |   20 | 9.00 |
+------+----------+--------+------+------+
```

2.) -> create table projects(id int,pid varchar(15),grade char(2),foreign key(id) references students1(id));

<u>Checking:</u>

->desc projects;



```
+-------+-------------+------+-----+---------+-------+
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| id    | int         | YES  | MUL | NULL    |       |
| pid   | varchar(15) | YES  |     | NULL    |       |
| grade | char(2)     | YES  |     | NULL    |       |
+-------+-------------+------+-----+---------+-------+
```

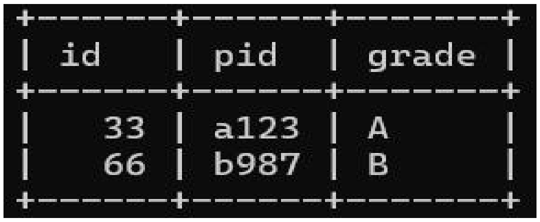
-> insert into projects values(33,'a123','A'),(66,'b987','B');

Output:

Query OK, 2 rows affected

Checking:

->select * from projects;



```
+------+------+-------+
| id   | pid  | grade |
+------+------+-------+
|   33 | a123 | A     |
|   66 | b987 | B     |
+------+------+-------+
```

->insert into projects values(77 ,'q123','A');

Output:

Cannot add or update a child row: a foreign key constraint fails

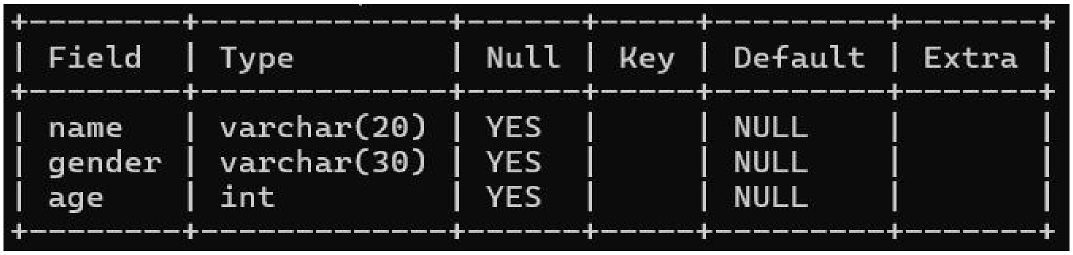
5. Check constraint:

1. Business rules validations can be applied to a table column by using check constraint.
2. It is used to limit the value range that can be placed in a column.

-> create table voters(name varchar(20),gender varchar(30),age int check(age>18));

Checking:

->desc voters;

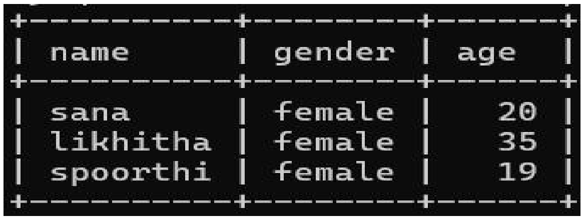


```
+--------+-------------+------+-----+---------+-------+
| Field  | Type        | Null | Key | Default | Extra |
+--------+-------------+------+-----+---------+-------+
| name   | varchar(20) | YES  |     | NULL    |       |
| gender | varchar(30) | YES  |     | NULL    |       |
| age    | int         | YES  |     | NULL    |       |
+--------+-------------+------+-----+---------+-------+
```

-> insert into voters values('sana','female',20),('likhitha','female',35),('spoorthi','female',19);

Checking:

->select * from voters;



```
+----------+--------+------+
| name     | gender | age  |
+----------+--------+------+
| sana     | female |   20 |
| likhitha | female |   35 |
| spoorthi | female |   19 |
+----------+--------+------+
```

->insert into voters values('yashu','male',16);

Output:

Check constraint 'voters_chk_1' is violated.