##### EXPERIMENT NO:-03

**Aim: Create a database using Data Definition Language (DDL) and apply integrity constraints for the specified system**

**Theory:**

**DDL – Data Definition Language:**

**Data definition language (DDL)** is a language that allows the user to define the data and their relationship to other types of data.

Data Definition language statements work with the structure of the database table.

* Various data types used in defining columns in a database table
* Integrity and value constraints
* Viewing, modifying and removing a table structure

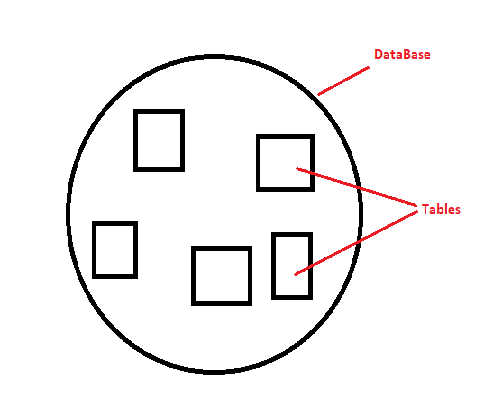
## DDL Commands

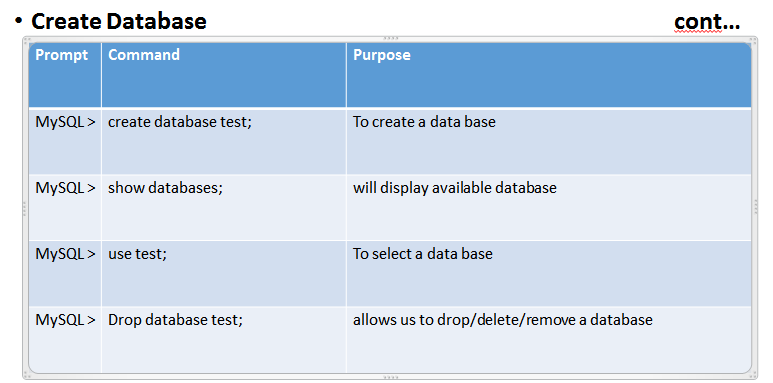
The Data Definition Languages (DDL) Commands are as follows −

* **Create** − It is used to create a new table or a new database.
* **Alter** − It is used to alter or change the structure of the database table.
* **Drop** − It is used to delete a table, index, or views from the database.
* **Truncate** − It is used to delete the records or data from the table, but its structure remains as it is.
* **Rename** − It is used to rename an object from the database.

**Create Database**

* The CREATE DATABASE statement is used to create a new SQL database.





## Data types

When a table is created, each column in the table is assigned a data type.

Some of the important data types are as follows −

* Varchar2
* Char
* Number
* **Entity Integrity constraint:**
  + The term Data integrity refers to the correctness and completeness of the data in a database.
  + Entity Integrity constraint are the part of the table definition that are used to limit the values entered into the columns.
  + There are Six types of Entity Integrity constraint.

|  |  |  |
| --- | --- | --- |
| **Sr.no.** | **Entity Integrity Constraint** | **Purpose** |
| 1 | NOT NULL | This constraint specifies that the column cannot have NULL or empty values. The below statement creates a table with NOT NULL constraint. |
| 2 | UNIQUE | This constraint ensures that all values inserted into the column will be unique. It means a column cannot stores duplicate values. MySQL allows us to use more than one column with UNIQUE constraint in a table. |
| 3 | CHECK | It controls the value in a particular column.  It ensures that the inserted value in a column must be satisfied with the given condition. |
| 4 | DEFAULT | This constraint is used to set the default value for the particular column where we have not specified any value. |
| 5 | PRIMARY KEY | This constraint is used to identify each record in a table uniquely. If the column contains primary key constraints, then it cannot be null or empty. |
| 6 | Foreign Key | This constraint is used to link two tables together. It is also known as the referencing key. A foreign key column matches the primary key field of another table. It means a foreign key field in one table refers to the primary key field of another table. |

## DDL Commands with example:

Let’s see each DDL command with an example.

**Create**

It is used to create a new table or a new database.

An example of the create command is as follows −

create table student(stdname varchar(20) , branch varchar(20),college varchar(20), age number, telephone number, address varchar(20));

A student table is created with the fields given below −

| **Stdname** | **Branch** | **College** | **Age** | **Telephone** | **Address** |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**Alter**

It is used to alter or change the structure of the database table

An example of the alter command is as follows −

ALTER TABLE student ADD birthdate DATETIME

**Drop**

It is used to delete a table, index, or views from the database.

An example of the drop command is as follows −

 DROP TABLE student

**Application of DDL commands on Specified Case study with Input & Output:**

* **Creating Table using constraint: PATIENT**

create table patient

(

pid int PRIMARY KEY,

pname varchar(20) UNIQUE,

p\_blood\_grp varchar(20) NOT NULL,

page int(3) NOT NULL,

pweight int(3) NOT NULL,

p\_dob date,

p\_disease varchar(20) UNIQUE,

did int,

FOREIGN KEY (did) REFERENCES donor (did)

);

* **Creating Table using constraint: P\_MOBILE\_NO**

create table p\_mobile\_no

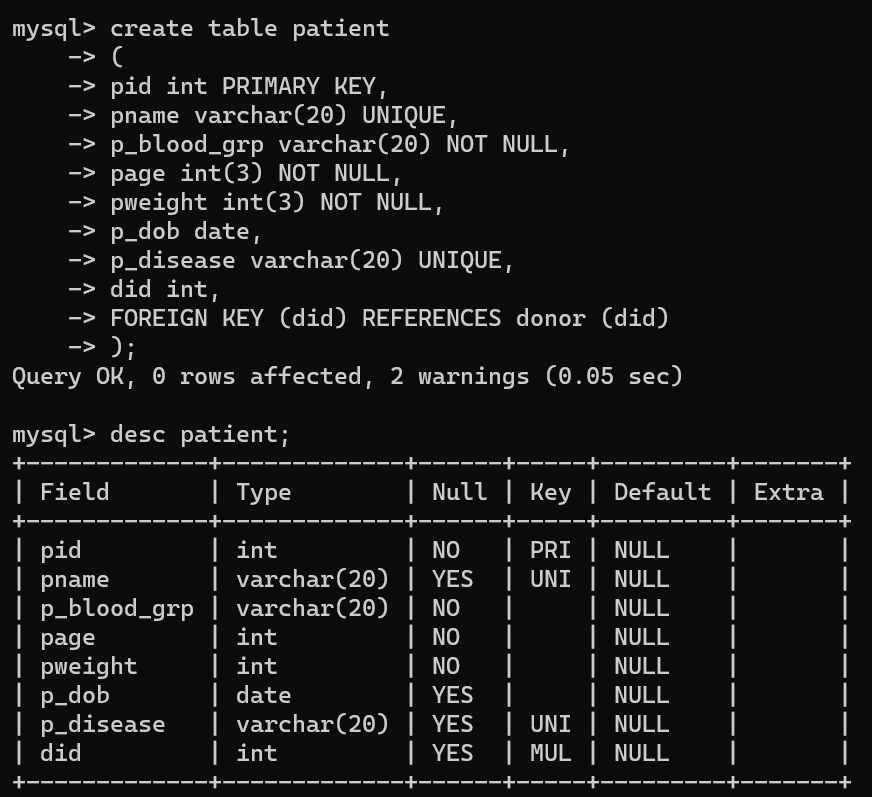
(

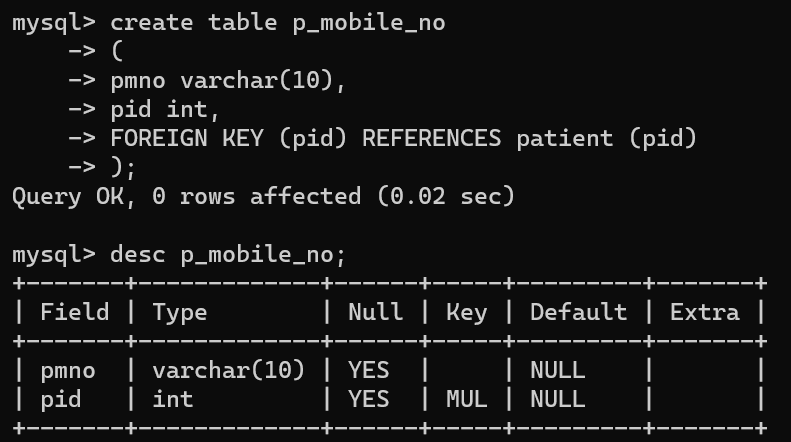
pmno varchar(10),

pid int,

FOREIGN KEY (pid) REFERENCES patient (pid)

);





* **Creating Table using constraint: DONOR**

create table donor

(

did int PRIMARY KEY,

dname varchar(20) UNIQUE,

dage int(3) NOT NULL,

dweight int(3) NOT NULL,

d\_blood\_grp varchar(5) NOT NULL,

d\_disease varchar(20) NOT NULL,

d\_dob date

);

* **Creating Table using constraint: D\_MOBILE\_NO**

create table d\_mobile\_no

(

did int PRIMARY KEY,

dmno VARCHAR(20)

);

* **Creating Table using constraint: BLOOD\_BANK**

create table blood\_bank

(

bb\_regno int PRIMARY KEY,

bb\_name varchar(20) UNIQUE,

bb\_add varchar(40) UNIQUE,

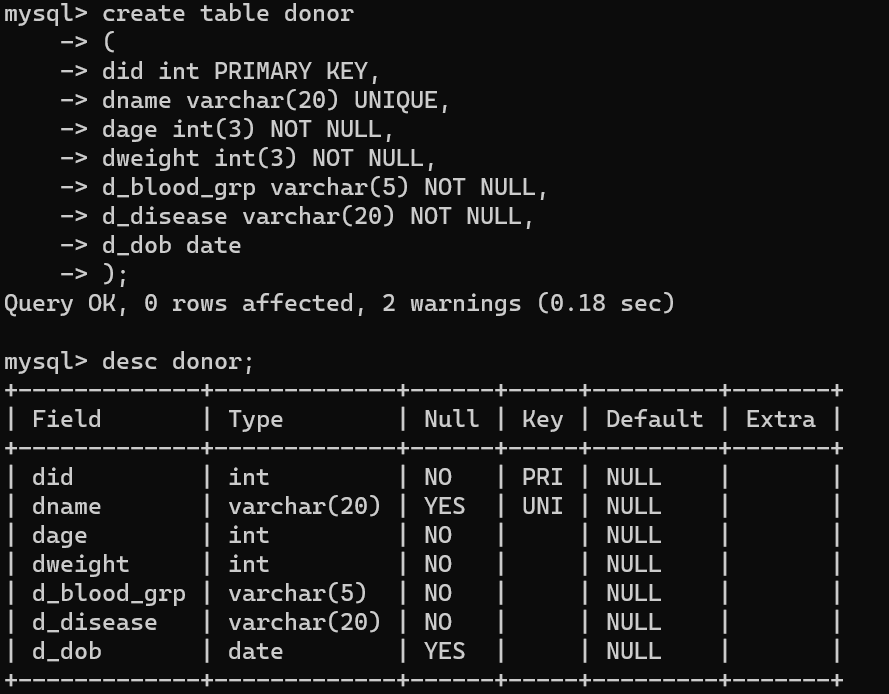
pid int,

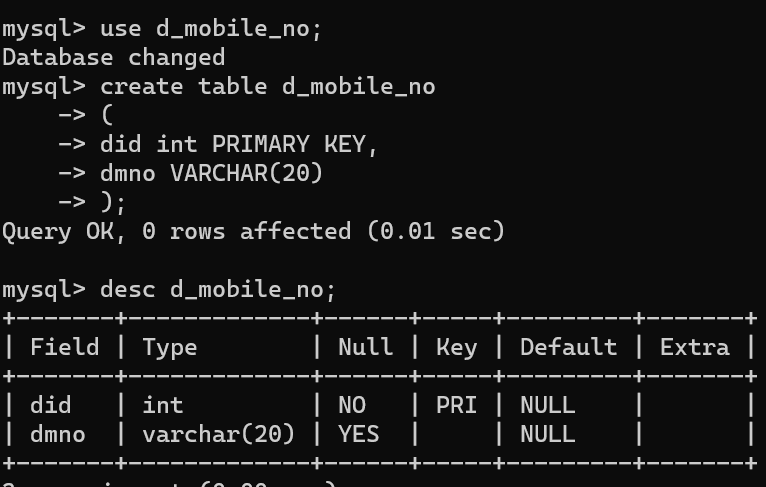
did int,

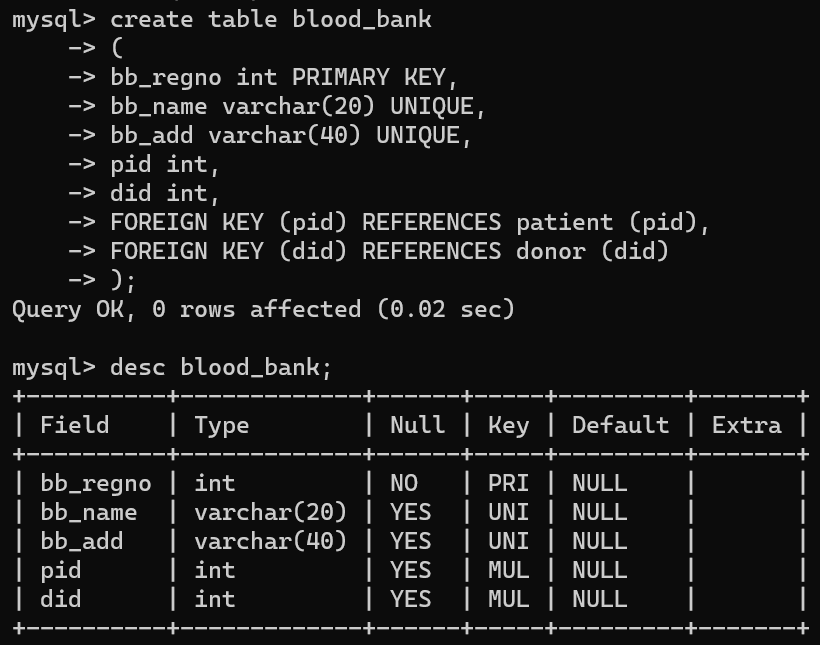
FOREIGN KEY (pid) REFERENCES patient (pid),

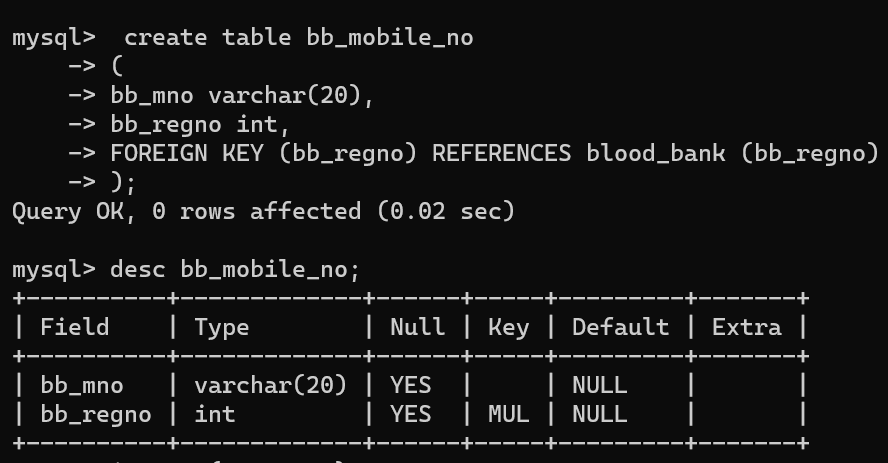
FOREIGN KEY (did) REFERENCES donor (did)

);









* **Creating Table using constraint: BB­\_MOBILE\_NO**

create table bb\_mobile\_no

(

bb\_mno varchar(20),

bb\_regno int,

FOREIGN KEY (bb\_regno) REFERENCES blood\_bank (bb\_regno)

);

* **Alter Table**
* Is used to change definition of a table
* Example
  + mysql> alter table emp

add grade char(2);

mysql>alter table emp

modify grade char(4);

* + mysql> alter table emp

drop grade;

* **Drop Table:**
* Drop Table statement to delete the existing table. This statement removes the complete data of a table along with the whole structure or definition permanently from the database.
  + - * Example : mysql> drop table emp;
* **Truncate table :** TRUNCATE statement removes the complete data without removing its structure.
  + - * Example : mysql>truncate table emp;
* **Rename table :** to rename existing table name with new table name.
  + - * Example: mysql>rename table emp to dmce;

**Conclusion:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **R1** | **R2** | **R3** | **R4** | **Total** | **Sign with Date** |
| **(3)** | **(5)** | **(4)** | **(3)** | **(15)** |  |
|  |  |  |  |  |