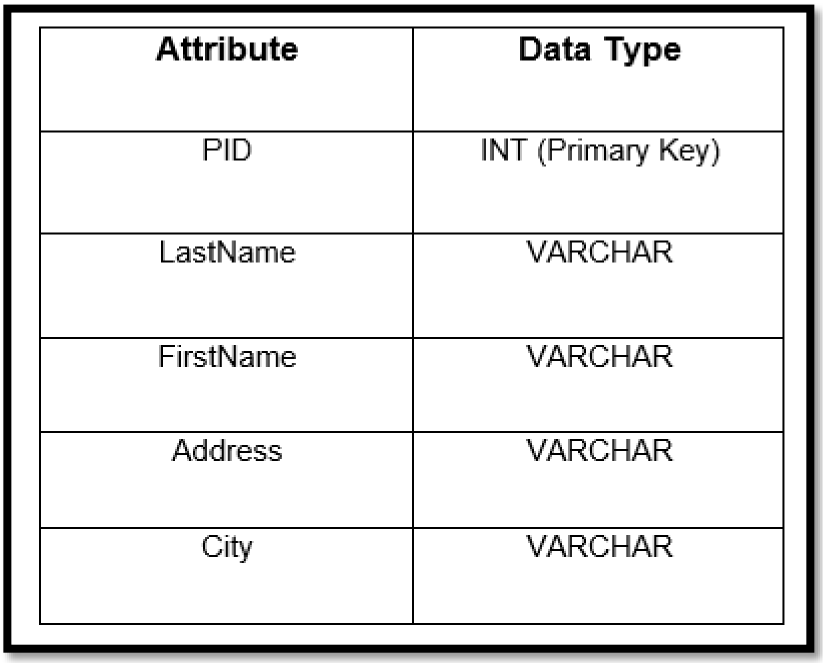
Q51: Write a query for creating a table named People, which contains information given in the table below:



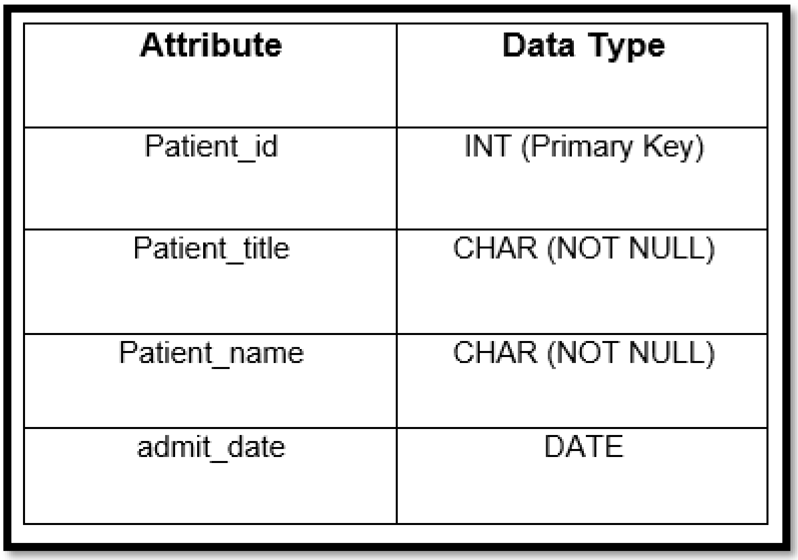
Attribute	Data Type
PID	INT (Primary Key)
LastName	VARCHAR
FirstName	VARCHAR
Address	VARCHAR
City	VARCHAR

```
CREATE TABLE PEOPLE(
    PID INT,
    LASTNAME VARCHAR(255),
    FIRSTNAME VARCHAR(255),
    ADDRESS VARCHAR(255),
    CITY VARCHAR(255),
    PRIMARY KEY (PID)
);
```

OR

```
CREATE TABLE PEOPLE(
    PID INT PRIMARY KEY,
    LASTNAME VARCHAR(255),
    FIRSTNAME VARCHAR(255),
    ADDRESS VARCHAR(255),
    CITY VARCHAR(255)
);
```

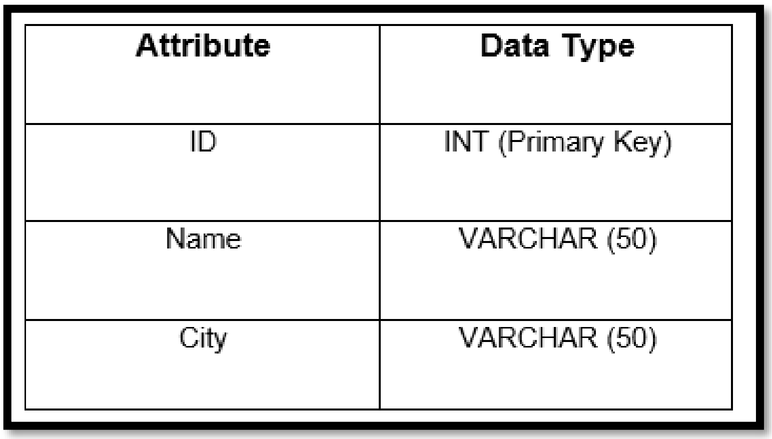
Q52: Write a query for creating a table named Patients, which contains the attribute given in the table below:



Attribute	Data Type
Patient_id	INT (Primary Key)
Patient_title	CHAR (NOT NULL)
Patient_name	CHAR (NOT NULL)
admit_date	DATE

```
CREATE TABLE PATIENTS(
    PATIENT_ID INT PRIMARY KEY,
    PATIENT_TITLE CHAR (255) NOT NULL,
    PATIENT_NAME CHAR (255) NOT NULL,
    ADMIT_DATE DATE
);
```

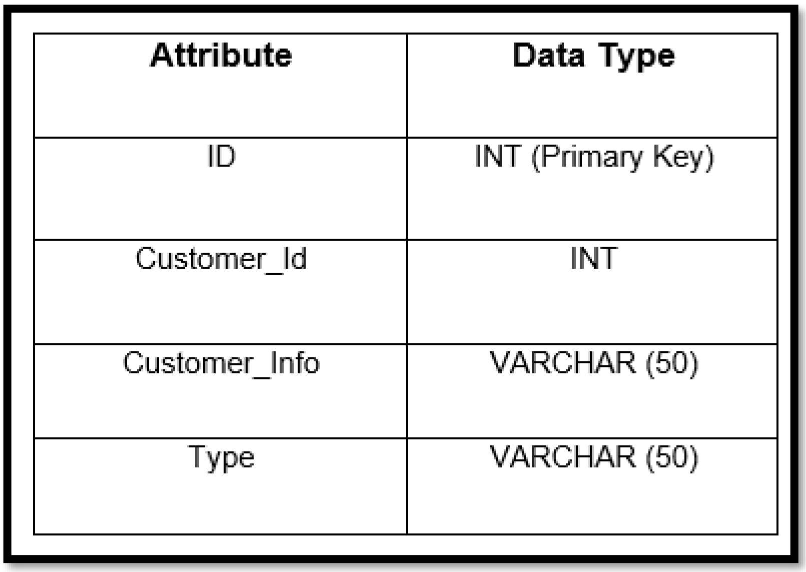
Q53: Write multiple queries to create two tables named customer and contacts:
The table customer contains features like ID, Name, City where all columns are never NULL. The ID here will be the primary key.



Attribute	Data Type
ID	INT (Primary Key)
Name	VARCHAR (50)
City	VARCHAR (50)

```
CREATE TABLE CUSTOMER(
    ID INT PRIMARY KEY,
    NAME VARCHAR(50) NOT NULL,
    CITY VARCHAR(50) NOT NULL
);
```

The Table contacts contain features like ID, Customer_Id, Customer_info, Type. Customer_ id being a foreign key that refers to ID(customer table). Also Customer_info and Type are never NULL.

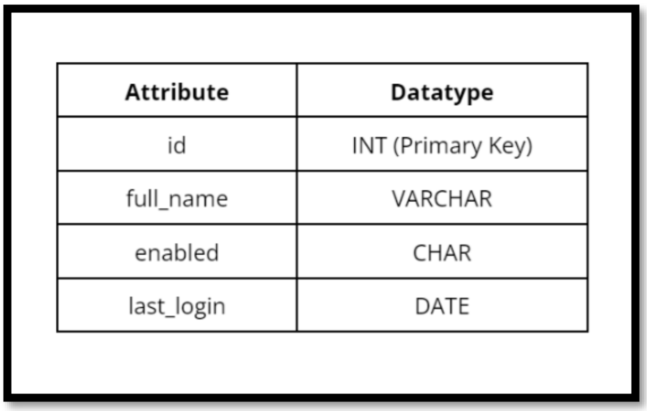


Attribute	Data Type
ID	INT (Primary Key)
Customer_Id	INT
Customer_Info	VARCHAR (50)
Type	VARCHAR (50)

```
CREATE TABLE CONTACTS(
    ID INT PRIMARY KEY,
    CUSTOMER_ID INT,
    CUSTOMER_INFO VARCHAR(50) NOT NULL,
    TYPE VARCHAR(50) NOT NULL,
    FOREIGN KEY (CUSTOMER_ID) REFERENCES CUSTOMER (ID)
);
```

Q54: Consider the tables given below:

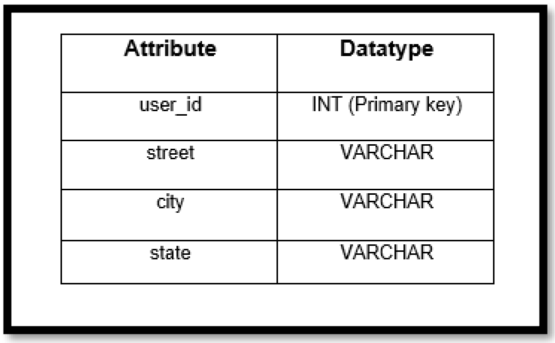
The table users contains features like id, full_name, enabled, last_login. The attribute id here will be the primary key.
Attribute List(s):



Attribute	Datatype
id	INT (Primary Key)
full_name	VARCHAR
enabled	CHAR
last_login	DATE

```
CREATE TABLE users(
    ID INT PRIMARY KEY,
    FULL_NAME VARCHAR(255),
    ENABLED CHAR(255),
    LAST_LOGIN DATE
);
```

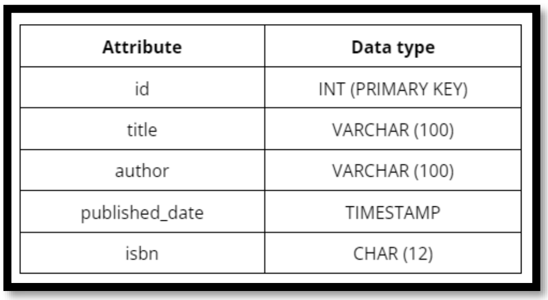
The table addresses contains features like user_id, street, city, state. The user_id here will be the primary key as well as foreign key that refers to id(users table). The attributes street, city and state should be declared NOT NULL.
Attribute List(s):



Attribute	Datatype
user_id	INT (Primary key)
street	VARCHAR
city	VARCHAR
state	VARCHAR

```
CREATE TABLE addresses(
    USER_ID INT PRIMARY KEY,
    STREET VARCHAR(255) NOT NULL,
    CITY VARCHAR(255) NOT NULL,
    STATE VARCHAR(255) NOT NULL,
    FOREIGN KEY (USER_ID) REFERENCES USERS (ID)
);
```

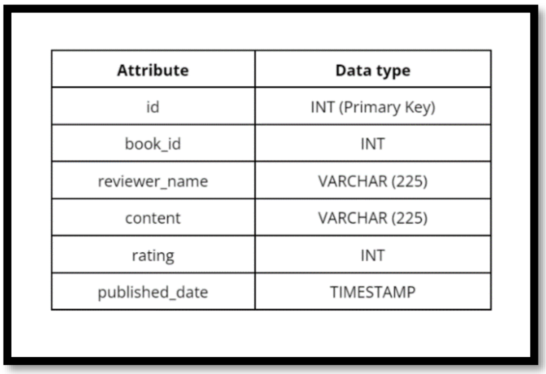
```
Q55: Consider the tables given below and formulate a SQL query to create
these tables with all the constraints.

The table books contain features like id, title, author, published_date,
isbn. The id should be declared as PRIMARY KEY. The isbn should be
declared UNIQUE. The attributes title, author and publish_date should be
declared NOT NULL.
```



Attribute	Data type
id	INT (PRIMARY KEY)
title	VARCHAR (100)
author	VARCHAR (100)
published_date	TIMESTAMP
isbn	CHAR (12)

```
CREATE TABLE BOOKS(
    ID INT PRIMARY KEY,
    TITLE VARCHAR (100) NOT NULL,
    AUTHOR VARCHAR (100) NOT NULL,
    PUBLISHED_DATE TIMESTAMP NOT NULL,
    ISBN CHAR (12) UNIQUE
);
```

```
The Table reviews contains features like id, book_id, reviewer_name,
content, rating, published_date . The id should be declared PRIMARY KEY.
The bookid should be declared FOREIGN KEY referred to id (table books).
The attribute bookid should be declared NOT NULL.
```



Attribute	Data type
id	INT (Primary Key)
book_id	INT
reviewer_name	VARCHAR (225)
content	VARCHAR (225)
rating	INT
published_date	TIMESTAMP

```
CREATE TABLE REVIEWS(
    ID INT PRIMARY KEY,
    BOOK_ID INT NOT NULL,
    REVIEWER_NAME VARCHAR (255),
    CONTENT VARCHAR (255),
    RATING INT,
    PUBLISHED_DATE TIMESTAMP,
    FOREIGN KEY (BOOK_ID) REFERENCES BOOKS (ID)
);
```