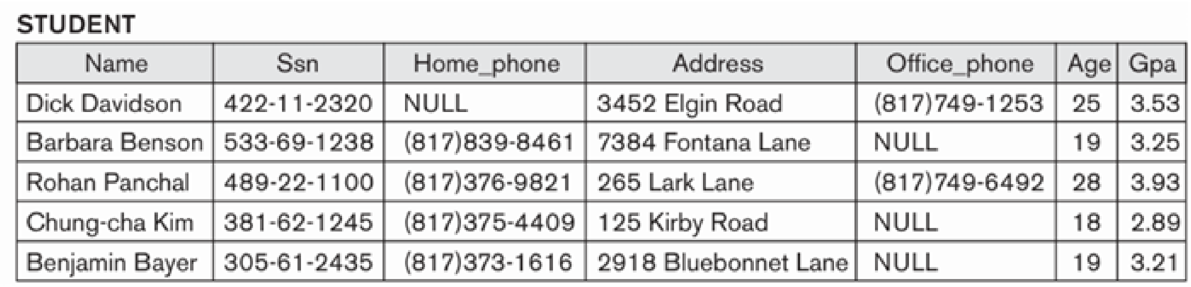
ADBMS Assignment No. 3

Q.1.) Write SQL statements for the following

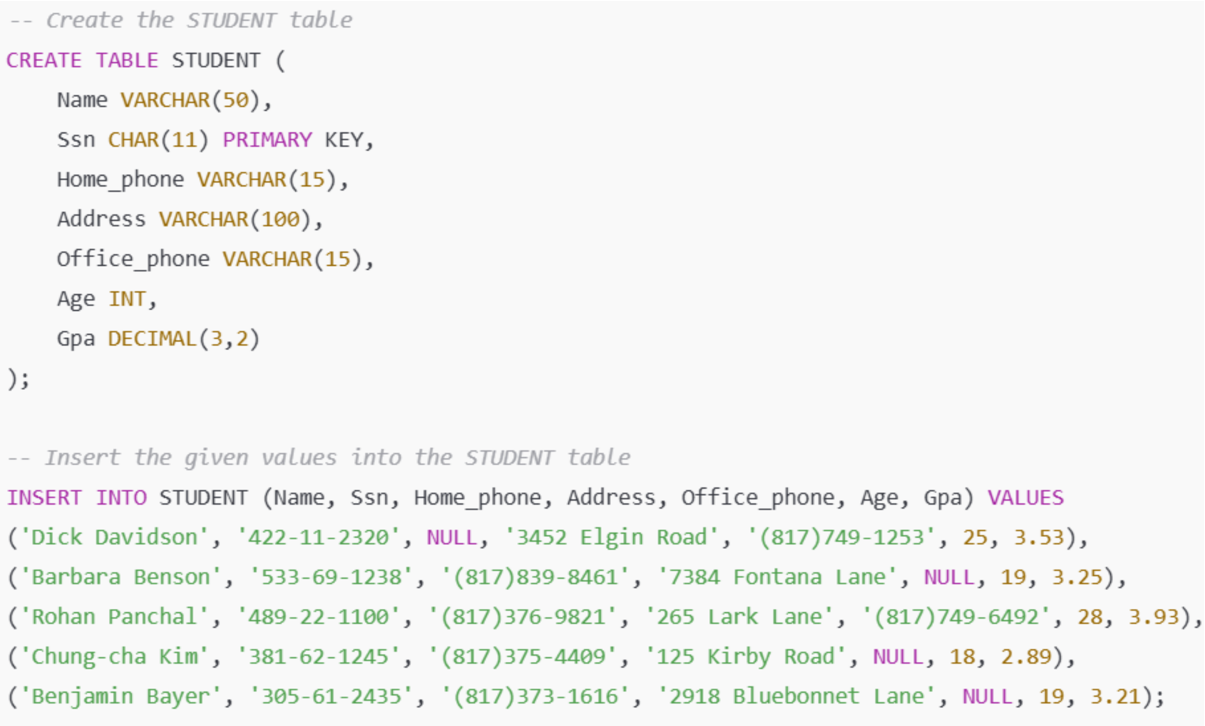


STUDENT

Name	Ssn	Home_phone	Address	Office_phone	Age	Gpa
Dick Davidson	422-11-2320	NULL	3452 Elgin Road	(817)749-1253	25	3.53
Barbara Benson	533-69-1238	(817)839-8461	7384 Fontana Lane	NULL	19	3.25
Rohan Panchal	489-22-1100	(817)376-9821	265 Lark Lane	(817)749-6492	28	3.93
Chung-cha Kim	381-62-1245	(817)375-4409	125 Kirby Road	NULL	18	2.89
Benjamin Bayer	305-61-2435	(817)373-1616	2918 Bluebonnet Lane	NULL	19	3.21

Write an DDL statement to create the above table and then insert into it the above given values.

Ans :-

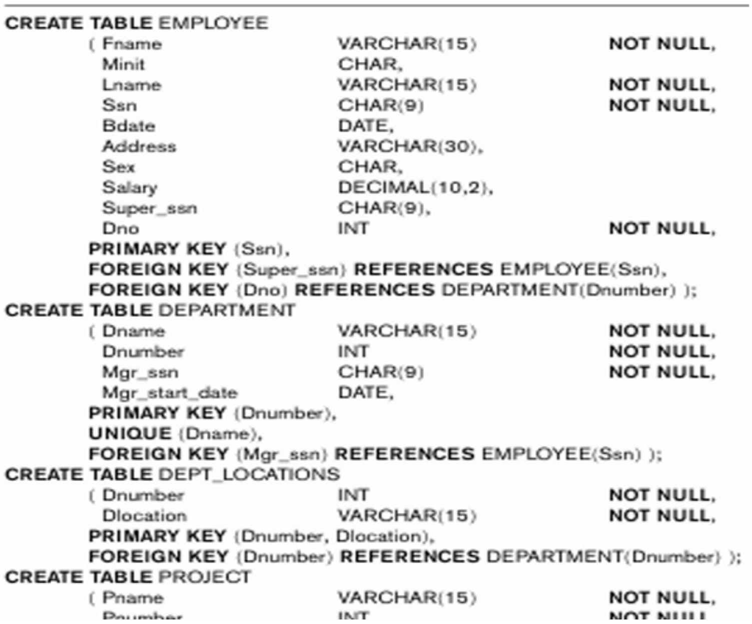


```
-- Create the STUDENT table
CREATE TABLE STUDENT (
    Name VARCHAR(50),
    Ssn CHAR(11) PRIMARY KEY,
    Home_phone VARCHAR(15),
    Address VARCHAR(100),
    Office_phone VARCHAR(15),
    Age INT,
    Gpa DECIMAL(3,2)
);

-- Insert the given values into the STUDENT table
INSERT INTO STUDENT (Name, Ssn, Home_phone, Address, Office_phone, Age, Gpa) VALUES
('Dick Davidson', '422-11-2320', NULL, '3452 Elgin Road', '(817)749-1253', 25, 3.53),
('Barbara Benson', '533-69-1238', '(817)839-8461', '7384 Fontana Lane', NULL, 19, 3.25),
('Rohan Panchal', '489-22-1100', '(817)376-9821', '265 Lark Lane', '(817)749-6492', 28, 3.93),
('Chung-cha Kim', '381-62-1245', '(817)375-4409', '125 Kirby Road', NULL, 18, 2.89),
('Benjamin Bayer', '305-61-2435', '(817)373-1616', '2918 Bluebonnet Lane', NULL, 19, 3.21);
```

Q.2.) Consider the following relations for a database that keeps track of student enrollment in courses and the books adopted for each course: STUDENT(Ssn, Name, Major, Bdate) COURSE(Course#, Cname, Dept) ENROLL(Ssn, Course#, Quarter, Grade) BOOK_ADOPTION(Course#, Quarter, Book_isbn) TEXT(Book_isbn, Book_title, Publisher, Author)

i) Use DDL statements to create the schema and

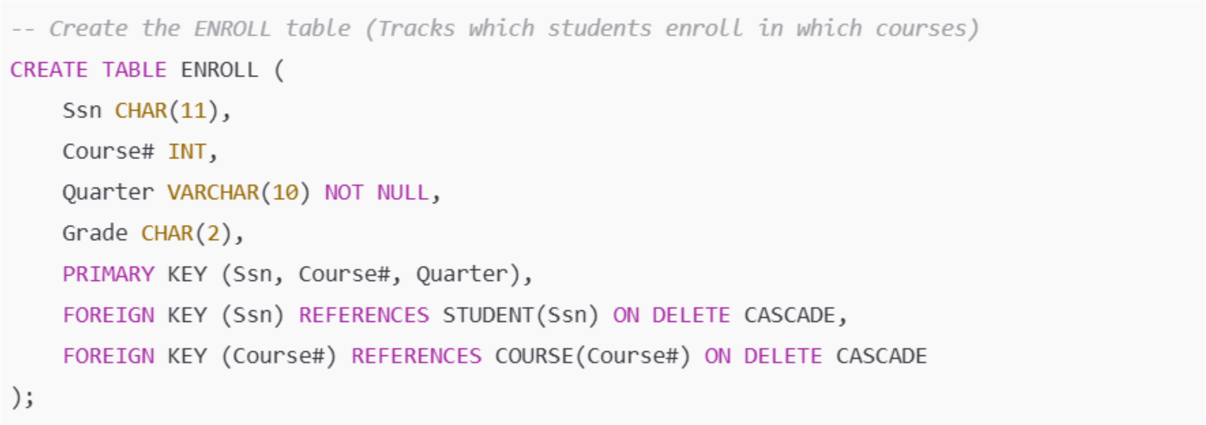
ii) Specify the foreign keys for this schema, stating any assumptions you make.



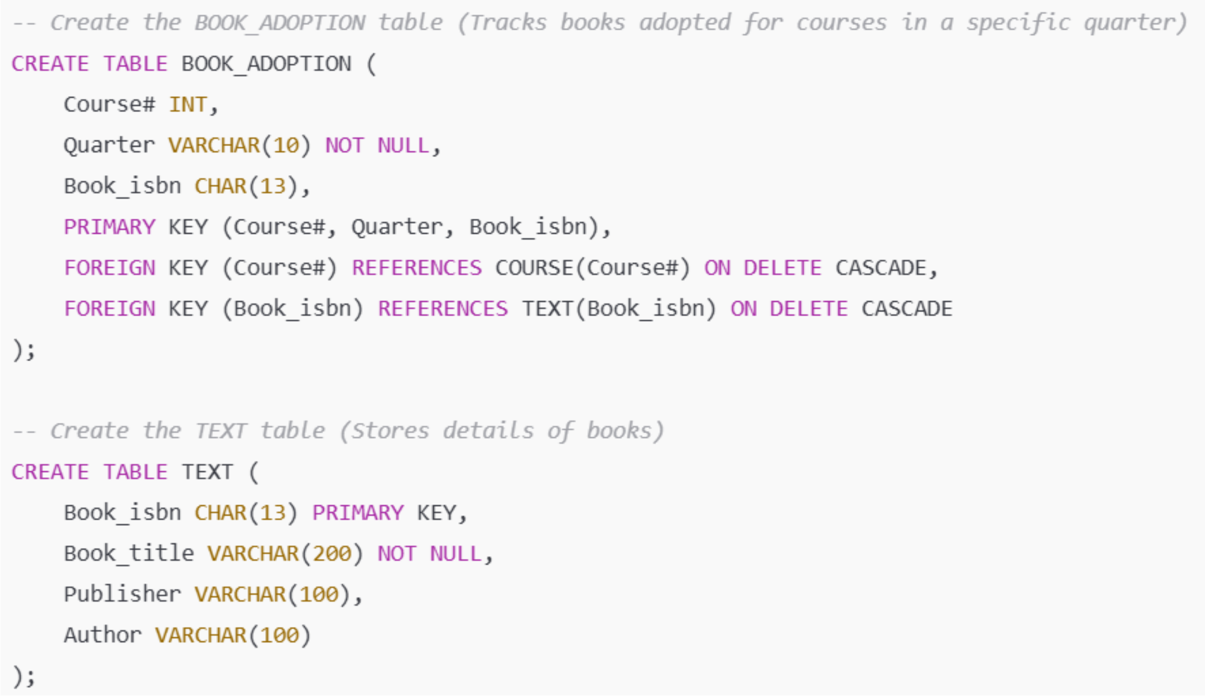
```
CREATE TABLE EMPLOYEE
        ( Fname               VARCHAR(15)          NOT NULL,
          Minit               CHAR,
          Lname               VARCHAR(15)          NOT NULL,
          Ssn                 CHAR(9)              NOT NULL,
          Bdate               DATE,
          Address             VARCHAR(30),
          Sex                 CHAR,
          Salary              DECIMAL(10,2),
          Super_ssn           CHAR(9),
          Dno                 INT                  NOT NULL,
        PRIMARY KEY (Ssn),
        FOREIGN KEY (Super_ssn) REFERENCES EMPLOYEE(Ssn),
        FOREIGN KEY (Dno) REFERENCES DEPARTMENT(Dnumber) );
CREATE TABLE DEPARTMENT
        ( Dname               VARCHAR(15)          NOT NULL,
          Dnumber             INT                  NOT NULL,
          Mgr_ssn             CHAR(9)              NOT NULL,
          Mgr_start_date      DATE,
        PRIMARY KEY (Dnumber),
        UNIQUE (Dname),
        FOREIGN KEY (Mgr_ssn) REFERENCES EMPLOYEE(Ssn) );
CREATE TABLE DEPT_LOCATIONS
        ( Dnumber             INT                  NOT NULL,
          Dlocation           VARCHAR(15)          NOT NULL,
        PRIMARY KEY (Dnumber, Dlocation),
        FOREIGN KEY (Dnumber) REFERENCES DEPARTMENT(Dnumber) );
CREATE TABLE PROJECT
        ( Pname               VARCHAR(15)          NOT NULL,
          Pnumber             INT                  NOT NULL
```

Ans :-

```
-- Create the STUDENT table
CREATE TABLE STUDENT (
    Ssn CHAR(11) PRIMARY KEY,
    Name VARCHAR(50) NOT NULL,
    Major VARCHAR(50),
    Bdate DATE
);

-- Create the COURSE table
CREATE TABLE COURSE (
    Course# INT PRIMARY KEY,
    Cname VARCHAR(100) NOT NULL,
    Dept VARCHAR(50) NOT NULL
);
```



```sql
-- Create the ENROLL table (Tracks which students enroll in which courses)
CREATE TABLE ENROLL (
    Ssn CHAR(11),
    Course# INT,
    Quarter VARCHAR(10) NOT NULL,
    Grade CHAR(2),
    PRIMARY KEY (Ssn, Course#, Quarter),
    FOREIGN KEY (Ssn) REFERENCES STUDENT(Ssn) ON DELETE CASCADE,
    FOREIGN KEY (Course#) REFERENCES COURSE(Course#) ON DELETE CASCADE
);
```



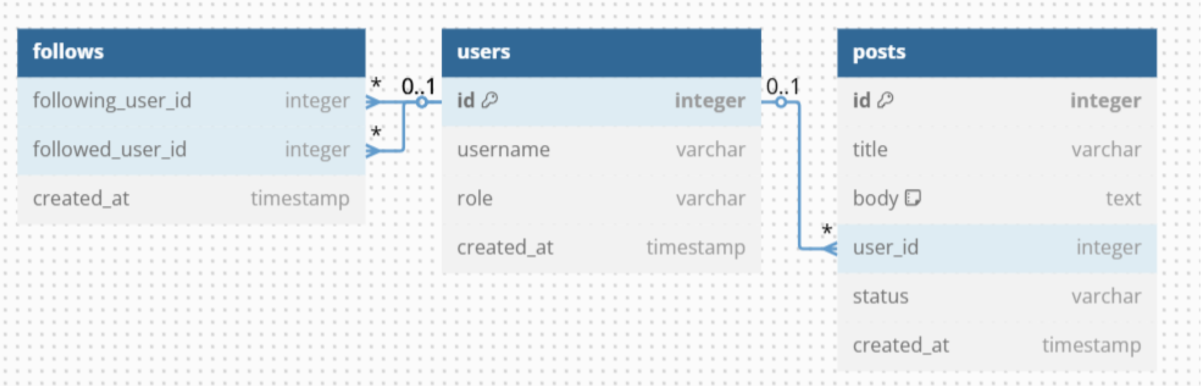
```sql
-- Create the BOOK_ADOPTION table (Tracks books adopted for courses in a specific quarter)
CREATE TABLE BOOK_ADOPTION (
    Course# INT,
    Quarter VARCHAR(10) NOT NULL,
    Book_isbn CHAR(13),
    PRIMARY KEY (Course#, Quarter, Book_isbn),
    FOREIGN KEY (Course#) REFERENCES COURSE(Course#) ON DELETE CASCADE,
    FOREIGN KEY (Book_isbn) REFERENCES TEXT(Book_isbn) ON DELETE CASCADE
);

-- Create the TEXT table (Stores details of books)
CREATE TABLE TEXT (
    Book_isbn CHAR(13) PRIMARY KEY,
    Book_title VARCHAR(200) NOT NULL,
    Publisher VARCHAR(100),
    Author VARCHAR(100)
);
```

Q.3.) Draw the database schema created by your RDBMS for the above DDL statements.

Ans :- // Use DBML to define your database structure

// Docs: https://dbml.dbdiagram.io/docs

```
Table follows {
  following_user_id integer
  followed_user_id integer
  created_at timestamp
}

Table users {
  id integer [primary key]
  username varchar
  role varchar
  created_at timestamp
}

Table posts {
  id integer [primary key]
  title varchar
  body text [note: 'Content of the post']
  user_id integer
  status varchar
  created_at timestamp
}

Ref: posts.user_id > users.id // many-to-one
Ref: users.id < follows.following_user_id
Ref: users.id < follows.followed_user_id
```



Q.4.) Write appropriate SQL DDL statements for declaring the LIBRARY relational database schema of Figure 4.6. Specify the keys and referential triggered actions.

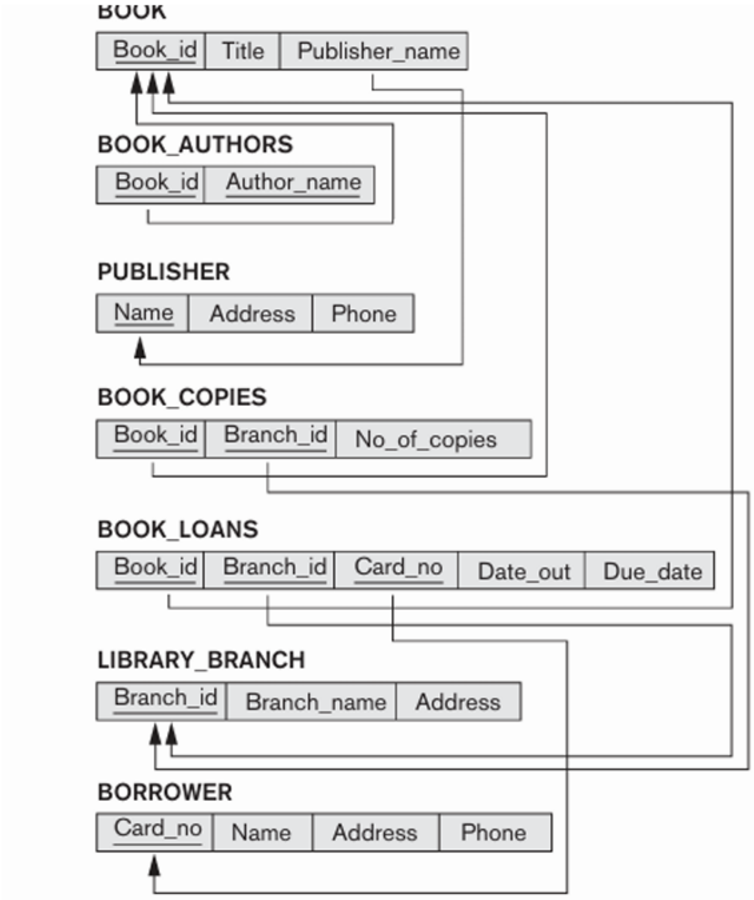


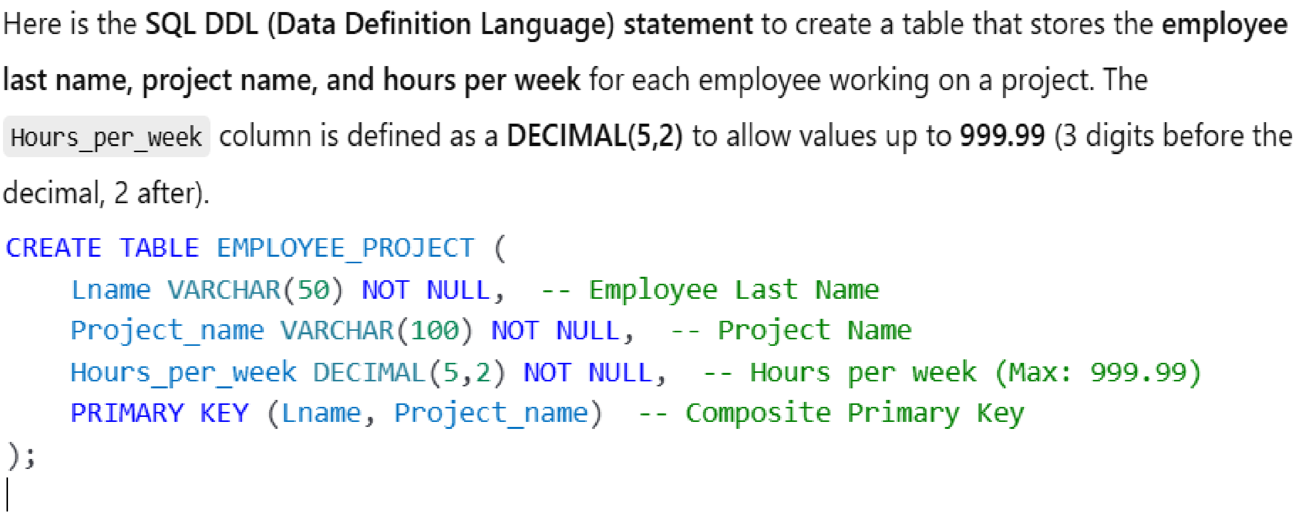
Figure 4.6
A relational database schema for a LIBRARY database.

Ans :-

```sql
-- Create the BOOK table
CREATE TABLE BOOK (
    Book_id INT PRIMARY KEY,
    Title VARCHAR(255) NOT NULL,
    Publisher_name VARCHAR(100) NOT NULL
);

-- Create the BOOK_AUTHORS table (Many-to-Many relationship)
CREATE TABLE BOOK_AUTHORS (
    Book_id INT,
    Author_name VARCHAR(100) NOT NULL,
    PRIMARY KEY (Book_id, Author_name),
    FOREIGN KEY (Book_id) REFERENCES BOOK(Book_id) ON DELETE CASCADE
);

-- Create the PUBLISHER table
CREATE TABLE PUBLISHER (
    Name VARCHAR(100) PRIMARY KEY,
    Address VARCHAR(255),
    Phone VARCHAR(20)
);

-- Create the BOOK_COPIES table (Tracks copies of books at branches)
CREATE TABLE BOOK_COPIES (
    Book_id INT,
    Branch_id INT,
    No_of_copies INT DEFAULT 0,
    PRIMARY KEY (Book_id, Branch_id),
    FOREIGN KEY (Book_id) REFERENCES BOOK(Book_id) ON DELETE CASCADE,
    FOREIGN KEY (Branch_id) REFERENCES LIBRARY_BRANCH(Branch_id) ON DELETE CASCADE
);

-- Create the BOOK_LOANS table (Tracks borrowed books)
CREATE TABLE BOOK_LOANS (
    Book_id INT,
    Branch_id INT,
    Card_no INT,
    Date_out DATE NOT NULL,
    Due_date DATE NOT NULL,
    PRIMARY KEY (Book_id, Branch_id, Card_no),
    FOREIGN KEY (Book_id, Branch_id) REFERENCES BOOK_COPIES(Book_id, Branch_id) ON DELETE CASCADE,
    FOREIGN KEY (Card_no) REFERENCES BORROWER(Card_no) ON DELETE CASCADE
);

-- Create the LIBRARY_BRANCH table
CREATE TABLE LIBRARY_BRANCH (
    Branch_id INT PRIMARY KEY,
    Branch_name VARCHAR(100) NOT NULL,
    Address VARCHAR(255) NOT NULL
);

-- Create the BORROWER table
CREATE TABLE BORROWER (
    Card_no INT PRIMARY KEY,
    Name VARCHAR(100) NOT NULL,
    Address VARCHAR(255),
    Phone VARCHAR(20)
);
```

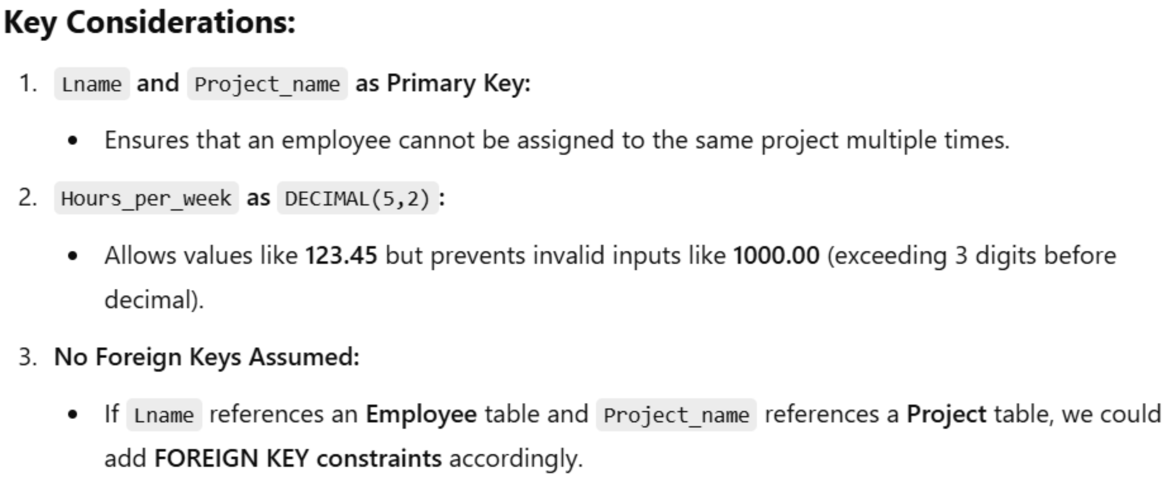
Q.5.) Write DDL to create a table that has the employee last name, project name, and hours per week(let it a decimal number accepting no's up to 3digit with 2 decimal places) for each employee working on a project.

Ans :-



Here is the **SQL DDL (Data Definition Language) statement** to create a table that stores the **employee last name, project name, and hours per week** for each employee working on a project. The `Hours_per_week` column is defined as a **DECIMAL(5,2)** to allow values up to **999.99** (3 digits before the decimal, 2 after).

```
CREATE TABLE EMPLOYEE_PROJECT (
    Lname VARCHAR(50) NOT NULL,  -- Employee Last Name
    Project_name VARCHAR(100) NOT NULL,  -- Project Name
    Hours_per_week DECIMAL(5,2) NOT NULL,  -- Hours per week (Max: 999.99)
    PRIMARY KEY (Lname, Project_name)  -- Composite Primary Key
);
```

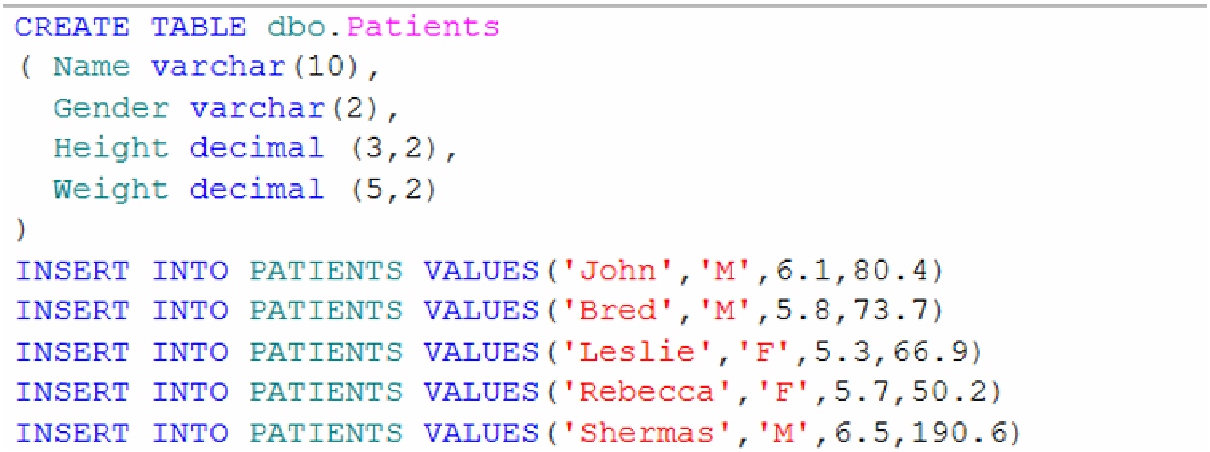


Key Considerations:

1. `Lname` **and** `Project_name` **as Primary Key:**
   - Ensures that an employee cannot be assigned to the same project multiple times.
2. `Hours_per_week` **as** `DECIMAL(5,2)`**:**
   - Allows values like **123.45** but prevents invalid inputs like **1000.00** (exceeding 3 digits before decimal).
3. **No Foreign Keys Assumed:**
   - If `Lname` references an **Employee** table and `Project_name` references a **Project** table, we could add **FOREIGN KEY constraints** accordingly.

Q.6.) Write a DDL statement to create a table containing following columns of following datatypes

Numeric, Character-string, Bit-string, Boolean data type, DATE data type, timestamp data type (TIMESTAMP) includes the DATE and TIME fields, INTERVAL data type, Also introduce constraints default, NOT NULL, CHECK Clause (eg. CHECK (Dnumber > 0 AND Dnumber < 21))



```
CREATE TABLE dbo.Patients
( Name varchar(10),
  Gender varchar(2),
  Height decimal (3,2),
  Weight decimal (5,2)
)
INSERT INTO PATIENTS VALUES('John','M',6.1,80.4)
INSERT INTO PATIENTS VALUES('Bred','M',5.8,73.7)
INSERT INTO PATIENTS VALUES('Leslie','F',5.3,66.9)
INSERT INTO PATIENTS VALUES('Rebecca','F',5.7,50.2)
INSERT INTO PATIENTS VALUES('Shermas','M',6.5,190.6)
```

Show how the above table looks like

Ans :-

DDL Statement to Create a Table with Various Data Types and Constraints

The following SQL **DDL (Data Definition Language)** statement defines a table that includes:

- **Numeric (INT, DECIMAL)**
- **Character-string (VARCHAR, CHAR)**
- **Bit-string (BIT)**
- **Boolean (BOOLEAN)**
- **Date and Timestamp (DATE, TIMESTAMP)**
- **Interval (INTERVAL)**
- **Constraints (NOT NULL, DEFAULT, CHECK)**

```
CREATE TABLE Employee_Details (
    Emp_ID INT PRIMARY KEY,  -- Numeric (Integer) as Primary Key
    Name VARCHAR(50) NOT NULL,  -- Character String with NOT NULL
    Gender CHAR(1) CHECK (Gender IN ('M', 'F')),  -- Gender can only be 'M' or 'F'
    Salary DECIMAL(10,2) DEFAULT 50000.00,  -- Numeric with Default Value
    Is_Active BIT DEFAULT 1,  -- Bit-String with Default Value 1 (Active)
    Is_FullTime BOOLEAN DEFAULT TRUE,  -- Boolean Data Type with Default TRUE
    Birth_Date DATE NOT NULL,  -- Date Type (Cannot be NULL)
    Joining_Timestamp TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  -- Timestamp with Default Current Time
    Work_Experience INTERVAL YEAR TO MONTH,  -- Interval Data Type (Years & Months)
    Department_Number INT CHECK (Department_Number > 0 AND Department_Number < 21)  -- CHECK Constraint
);
```

Table Representation Example

Emp_ID	Name	Gender	Salary	Is_Active	Is_FullTime	Birth_Date	Joining_Timestamp	Work_Experience	Department_Number
101	John	M	60000.00	1	TRUE	1990-05-15	2024-03-01 12:34:56	5-3	5
102	Alice	F	55000.00	1	TRUE	1995-07-20	2024-02-28 10:15:30	3-6	8
103	Mark	M	70000.00	1	FALSE	1988-12-10	2024-03-01 09:00:00	10-2	2