

Task:

- **Students Table :**

- `Student ID` (integer, primary key, not null)
- `StudentName` (nvarchar(50), not null)
- `Email` (nvarchar(50), not null)
- `Age` (integer, not null)

- **Courses Table :**

- `CourseID` (integer, primary key, not null)
- `CourseName` (nvarchar(50), not null)
- `Credits` (integer, not null)
- `StudentID` (integer, foreign key referencing `Students(StudentID)`, not null)

Solution -

Create a table Students -

```
CREATE TABLE Students (  
    StudentID INT PRIMARY KEY,  
    StudentName NVARCHAR(50) NOT NULL,  
    Email NVARCHAR(50) NOT NULL,  
    Age INT NOT NULL  
);
```

Create a table Courses with foreign key -

```
CREATE TABLE Courses (  
    CourseID INT PRIMARY KEY,  
    CourseName NVARCHAR(50) NOT NULL,  
    Credits INT NOT NULL,
```

```
StudentID INT NOT NULL,  
FOREIGN KEY (StudentID) REFERENCES Students(StudentID)  
);
```

Inserting data into Students table -

```
INSERT INTO Students (StudentID, StudentName, Email, Age) VALUES  
(1, 'Alice Johnson', 'alice.johnson@example.com', 20),  
(2, 'Bob Smith', 'bob.smith@example.com', 22),  
(3, 'Carol White', 'carol.white@example.com', 21),  
(4, 'David Brown', 'david.brown@example.com', 23);
```

Inserting data into Employees table -

```
INSERT INTO Courses (CourseID, CourseName, Credits, StudentID) VALUES  
(1, 'Mathematics', 3, 1),  
(2, 'Physics', 4, 2),  
(3, 'Chemistry', 3, 3),  
(4, 'Biology', 4, 4),  
(5, 'Computer Science', 3, 2);
```

INNER JOIN -

```
SELECT std.StudentID, std.StudentName, c.CourseName, c.Credits  
FROM Students std  
INNER JOIN Courses c ON std.StudentID = c.StudentID;
```

LEFT JOIN -

```
SELECT std.StudentID, std.StudentName, c.CourseName, c.Credits  
FROM Students std  
LEFT JOIN Courses c ON std.StudentID = c.StudentID;
```