Experiment No 6

Objective:

Learn how to create a table with primary key and foreign key constraints.

Instructions:

- Open your preferred SQL environment (e.g., MySQL Workbench, SQL Server Management Studio).
- Create a new database (if not already existing) and use it.

Theory:

Keys are one of the most important elements in a relational database to maintain the relationship between the tables and it also helps in uniquely identifying the data from a table. The primary Key is a key that helps in uniquely identifying the tuple of the database whereas the Foreign Key is a key that is used to identify the relationship between the tables through the primary key of one table that is the primary key one table acts as a foreign key to another table. Now, let’s discuss both of them in some detail.

What is Primary Key?

A primary key is used to ensure that data in the specific column is unique. A column cannot have NULL values. It is either an existing table column or a column that is specifically generated by the database according to a defined sequence.

Example: STUD_NO, as well as STUD_PHONE both, are candidate keys for relation STUDENT but STUD_NO can be chosen as the primary key (only one out of many candidate keys).

Table STUDENT

STUD_NO	STUD_NAME	STUD_PHONE	STUD_STATE	STUD_COUNT	STUD_AGE
1	RAM	9865278251	Haryana	India	20

STUD_NO	STUD_NAME	STUD_PHONE	STUD_STATE	STUD_COUNT	STUD_AGE
2	RAM	9655470231	Punjab	India	19
3	SUJIT	7514290359	Rajasthan	India	18
4	SURESH	8564103258	Punjab	India	21

Table STUDENT_COURSE

STUD_NO	COURSE_NO	COURSE_NAME
1	C1	DBMS
2	C2	Computer Networks
1	C2	Computer Networks

What is Foreign Key?

A foreign key is a column or group of columns in a relational database table that provides a link between data in two tables. It is a column (or columns) that references a column (most often the primary key) of another table.

Example: STUD_NO in STUDENT_COURSE is a foreign key to STUD_NO in STUDENT relation.

1. Create Table with Primary Key and Foreign Key Constraints:

-- Create the main table

```
CREATE TABLE Employees (
    EmployeeID INT PRIMARY KEY,
    FirstName VARCHAR(50),
    LastName VARCHAR(50),
    DepartmentID INT,
    FOREIGN KEY (DepartmentID) REFERENCES Departments(DepartmentID)
);
```

-- Create the referenced table

```
CREATE TABLE Departments (
    DepartmentID INT PRIMARY KEY,
    DepartmentName VARCHAR(50)
);
```

In this example, **Employees** table has a primary key constraint on **EmployeeID** and a foreign key constraint on **DepartmentID** referencing the **DepartmentID** column in the **Departments** table.

2. Alter Table with Add and Modify:

a. Add a new column:

```
ALTER TABLE Employees
ADD Email VARCHAR(100);
```

This adds a new column named **Email** to the **Employees** table.

b. Modify an existing column:

```
ALTER TABLE Employees
```

MODIFY COLUMN FirstName NVARCHAR(50);

This modifies the data type of the `FirstName` column to NVARCHAR(50).

3. Drop Table:

-- Drop foreign key constraint before dropping the table

ALTER TABLE Employees

DROP FOREIGN KEY fk_Department;

-- Drop the Employees table

DROP TABLE Employees;

-- Drop the Departments table

DROP TABLE Departments;

In this example, we first drop the foreign key constraint (**fk_Department**) associated with the **DepartmentID** column in the **Employees** table. Then, we drop the **Employees** and **Departments** tables.

Conclusion:

we have completed the experiment on creating tables with primary and foreign key constraints, altering tables by adding and modifying columns, and dropping tables. Practice these SQL commands to enhance your database management skills.