Documentation: School Database

Step 1: Create a Database

sql

```
CREATE DATABASE Schools;
```

This step creates a new database named "Schools" that will be used to store information about subjects, classes, and students.

Step 2: Use the Database

```
USE Schools;
```

This command switches the current database context to "Schools," so all subsequent SQL statements will be executed within this database.

Step 3: Create the Subjects Table

sql

```
CREATE TABLE Subjects (
    SubjectID INT PRIMARY KEY,
    SubjectName VARCHAR(50) NOT NULL
);
```

Here, I create a table called "Subjects" with two columns: "SubjectID" (an integer and the primary key) and "SubjectName" (a non-null string to store the subject names).

Step 4: Insert Data into Subjects Table

sql

```
INSERT INTO Subjects (SubjectID, SubjectName)
VALUES
  (1, 'Hindi'),
    (2, 'Social Science'),
    (3, 'Computer Science'),
    (4, 'Chemistry')
```

I populate the "Subjects" table with sample data, including subject IDs and names.

Step 5: Create the Classes Table

sql

```
CREATE TABLE Classes (
    ClassID INT PRIMARY KEY,
    ClassName VARCHAR(50) NOT NULL
);
```

This code creates a "Classes" table with columns for class IDs and class names.

Step 6: Insert Data into Classes Table

sql

```
INSERT INTO Classes (ClassID, ClassName)
VALUES
  (101, 'Class I'),
    (102, 'Class II'),
    (103, 'Class III'),
    (104, 'Class IV')
```

It can populate the "Classes" table with sample data, including class IDs and names.

Step 7: Create the Students Table

sql

```
CREATE TABLE Students (
    StudentID INT PRIMARY KEY,
    FirstName VARCHAR(50) NOT NULL,
    LastName VARCHAR(50) NOT NULL,
    DateOfBirth DATE,
    ClassID INT REFERENCES Classes(ClassID),
    SubjectID INT REFERENCES Subjects(SubjectID)
);
```

In this step, We create a "Students" table with columns to store student information, including first name, last name, date of birth, class ID (which references the "Classes" table), and subject ID (which references the "Subjects" table).

Step 8: Insert Data into Students Table

sql

```
INSERT INTO Students (StudentID, FirstName, LastName, DateOfBirth, ClassID, SubjectID)
VALUES
  (1, 'Nikhil', 'Sharma', '2000-01-01', 101, 1),
   (2, 'Ashish', 'Bharti', '2001-06-11', 102, 2),
   (3, 'Aman', 'Mittal', '2002-02-10', 103, 3),
   (4, 'Pragya', 'Shankdhar', '2003-12-13', 104, 4),
   (5, 'Apoorva', 'Rastogi', '2004-08-23', 101, 2)
```

We insert sample student data into the "Students" table, including student IDs, names, dates of birth, class IDs, and subject IDs.

Step 9: Create Indexes for Better Performance

```
CREATE INDEX IX_Students_ClassID ON Students(ClassID);
CREATE INDEX IX_Students_SubjectID ON Students(SubjectID);
```

These commands create indexes on the "ClassID" and "SubjectID" columns of the "Students" table to improve query performance.

Step 10: Query Data

Its provided several sample queries to retrieve data from the database, including getting a list of students in a specific class, finding the subject of a specific student and listing all subjects taught in a specific class.

Conclusion:

In this SQL script, we created a database named "Schools" and designed tables to store information about subjects, classes, and students. We inserted sample data into these tables and created indexes for better query performance. Finally, we demonstrated how to query the data to retrieve useful information for a school management system. This script provides the foundation for building a database-driven application for school management.