The Dream Program- SHSAT Prep Database Project

CIS 3400 FMWA [31511]

Susan Lin

Business Scenario:

The DREAM is a free extracurricular academic program that prepares eligible New York City public school students for SHSAT. The Specialized High Schools that require the SHSAT for admission can be considered elite. After the tests are scored, the students are put on a list from highest to lowest and would place the students into schools based on their rankings and available seats. The DREAM program is open to students from all NYC districts that meet all of the criteria. It is a summer academic program that prepares seventh-grade students to take the SHSAT in eighth grade. The program continued to run during the fall until the students took their tests.

The main issue faced by this program is most of the practice tests they give to students are in paper format which makes it difficult to keep track. This problem could be optimized through the use of a database that will optimize day-to-day operations for both staff and students. We can also keep track of the number of students that actually got into the Specialized High School after the program.

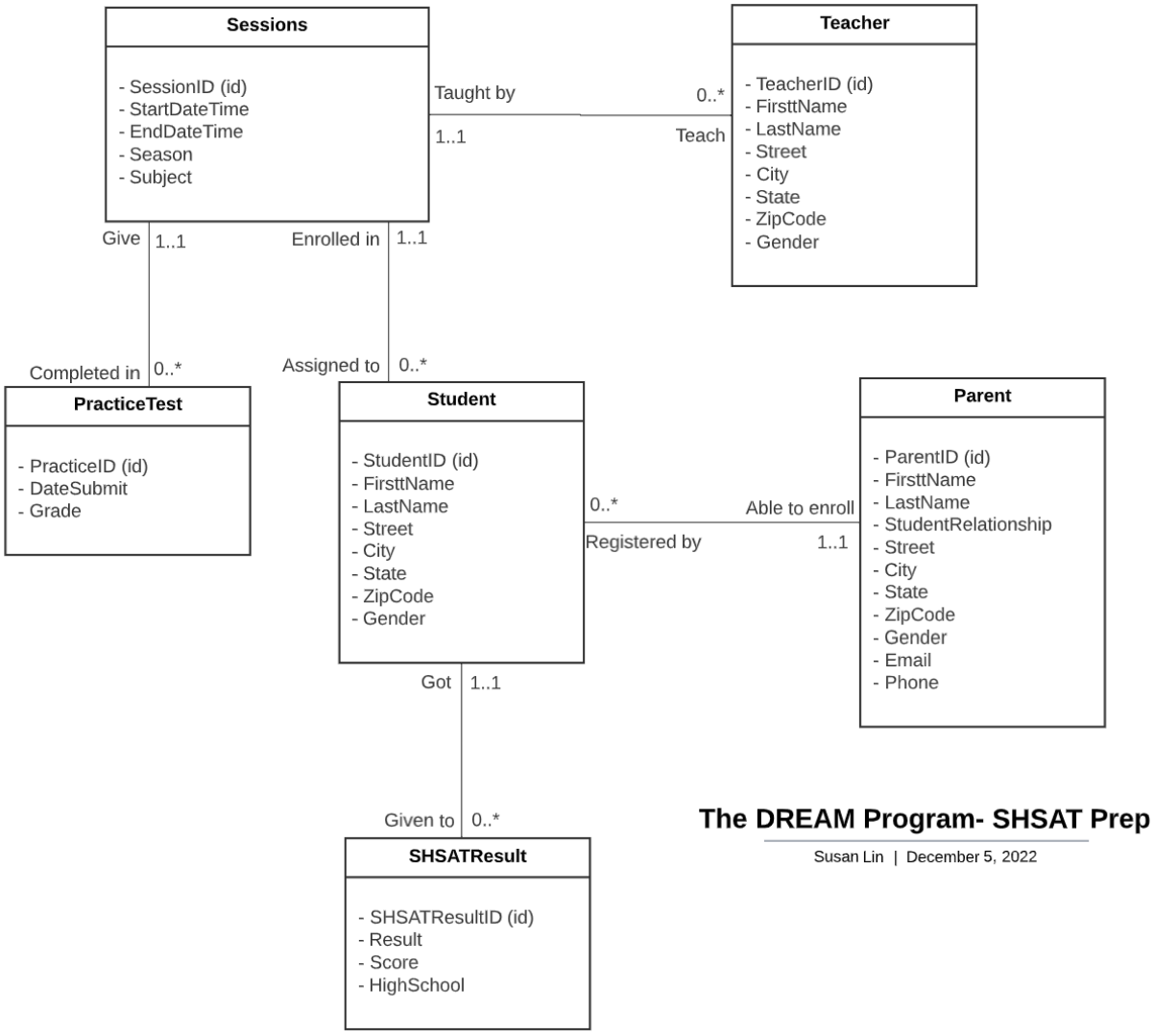
Information Needs: A database includes student names, levels, and progress in the program. The database should keep track of:

- Student
    - General information like name and address
    - Their progress and grades on tests
- Student's parent or guardian
    - General information like name and address
    - Student's relationship
- Teacher
    - General information like name and address
- Sessions
    - The date and time
- Practice Test and SHSAT
    - Every class time there will be a practice test for the students to complete
    - Keeping track of the grade the students got on each practice test

- How many students got into the Specialized High Scholl

Entities Needed: Student, Parent/ Guardian, Teacher, Sessions, Practice Test

ER Model using UML Notation:



Relationship Sentence:

One **SHSATResult** *may be* given to one and only **Student**

One **Student** *must be* got one or more **SHSATResult**

One **Parent** *may be* able to enroll one or more **Student**

One **Student** *must be* registered by one and only one **Parent**

One **Student** *may be* assigned to one and only one **Session**

One **Session** *must be* enrolled in one or more **Student**

One **Practice Test** *may be* completed in one and only **Session**

One **Session** *must be* given one or more **Practice Test**

One **Teacher** *may be* teaching one and only one **Session**

One **Session** *must be* taught by one or more **Teacher**

Conversion to Relational Model:

Parent (ParentID (key), FirstName, LastName, Street, City, State, ZipCode, Gender)

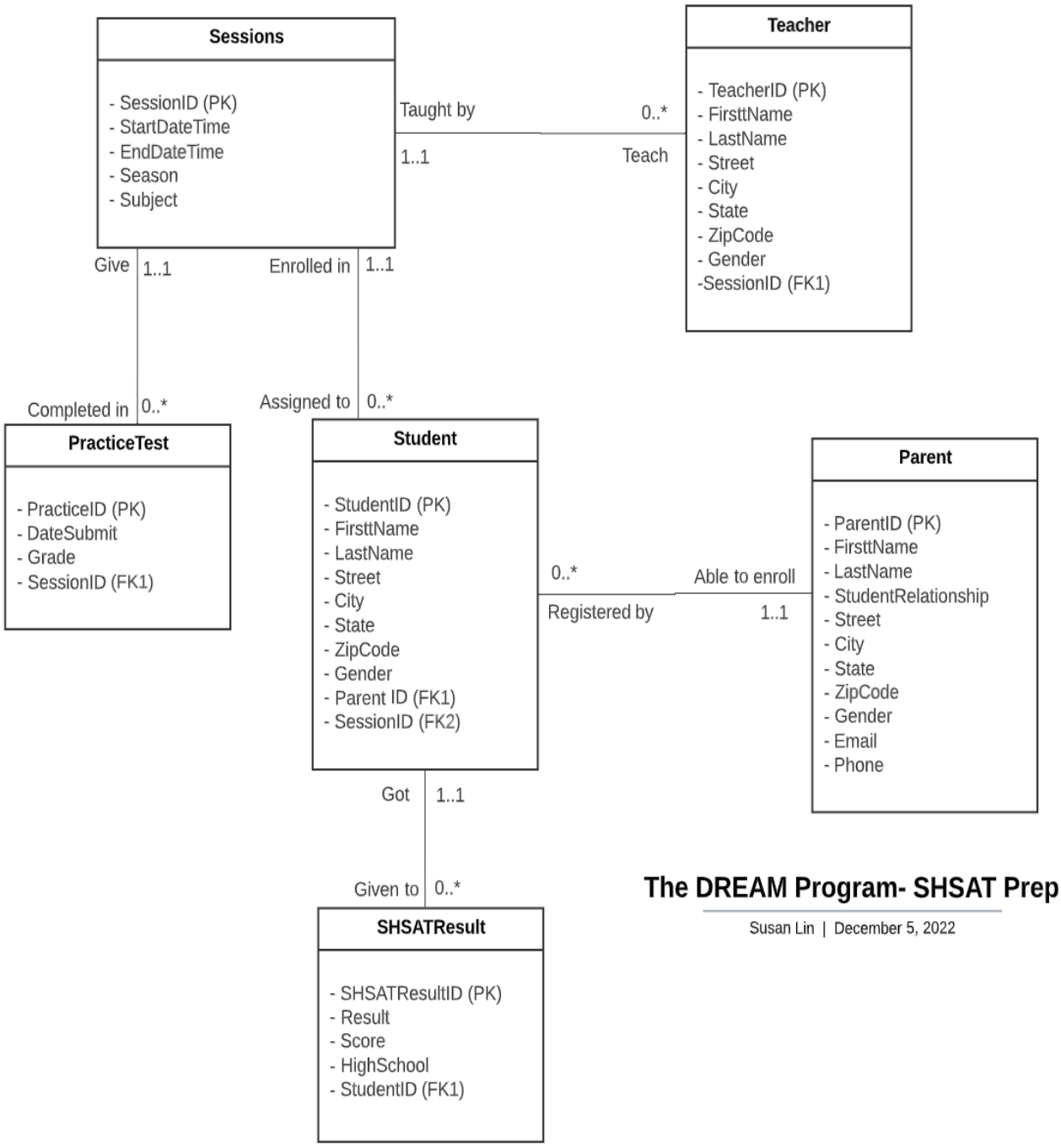
Student (StudentID (key), FirstName, LastName, Street, City, State, ZipCode, Gender, ParentID (fk), SessionID (fk))

Teacher (TeacherTeacherID (key), FirstName, LastName, Street, City, State, ZipCode, Gender, SessionID (fk))

SHSATResult (SHSATResultID (key), Result, Grade, HighSchool, StudentID (fk))

Practice Test (PracticeID (key), DateSubmit, Grade)PracticeID (key), DateSubmit, Grade, SessionID (fk))

Session (SessionID (key), StartDateTime, EndDateTime, Season)



The DREAM Program- SHSAT Prep

Susan Lin | December 5, 2022

Normalization:

1) **Parent** (ParentID (key), FirstName, LastName, Street, City, State, ZipCode, Gender)

ParentID	FirstName	LastName	Street	City	State	ZipCode	Gender
P100	Elisa	Smith	1745 Broadway	Manhattan	NY	10019	F
P101	Michelle	Taylor	921 Post Ave	Staten Island	NY	10302	F
P102	Joe	King	31 Saint Johns	Brooklyn	NY	11217	M
P103	John	Lee	197 Ashland	Brooklyn	NY	11217	M
P104	Lisa	Wood	94 Madsen Ave	Staten Island	NY	10302	F
P105	Mika	White	806 Washington	Brooklyn	NY	11217	M
P106	Fiona	Walker	1857 Broadway	Manhattan	NY	10019	F

Key: ParentID

FD1: ParentID → FirstName, LastName, Street, City, State, ZipCode, Gender

FD2: ZipCode → City, State

1NF: Meets the definition of a relation

2NF: No partial Key dependencies

3NF: Transitive dependency exists:

ParentID → FirstName, LastName, ZipCode (fk), Street, Gender

ZipCode → City, State

Solution: Split Parent relation into two new relations named Parent1 and ZipCode:

Parent1 (ParentID (key), FirstName, LastName, Street, ZipCode (fk), Gender)

ParentID	FirstName	LastName	Street	ZipCode	Gender
P100	Elisa	Smith	1745 Broadway	10019	F
P101	Michelle	Taylor	921 Post Ave	10302	F
P102	Joe	King	31 Saint Johns	11217	M
P103	John	Lee	197 Ashland	11217	M
P104	Lisa	Wood	94 Madsen Ave	10302	F
P105	Mika	White	806 Washington	11217	M
P106	Fiona	Walker	1857 Broadway	10019	F

Key: ParentID

FD1: ParentID → FirstName, LastName, PhoneNumber, Street, ZipCode (fk), Gender

1NF: Meets the definition of a relation

2NF: No partial Key dependencies

3NF: No Transitive dependencies

ZipCode (ZipCode (key), City, State)

City	State	ZipCode
Staten Island	NY	10302
Brooklyn	NY	11217
Manhattan	NY	10019

Key: ZipCode

FD1: ZipCode → City, State

1NF: Meets the definition of a relation

2NF: No partial Key dependencies

3NF: No Transitive dependencies

2) **Student** (StudentID (key), FirstName, LastName, Street, City, State, ZipCode, Gender, ParentID (fk), SessionID (fk))

StudentID	FirstName	LastName	Street	City	State	ZipCode	Gender	ParentID	SessionID
S100	Rose	Smith	1745 Broadway	Manhattan	NY	10019	F	P100	1
S101	Bella	Taylor	921 Post Ave	Staten Island	NY	10302	F	P101	2
S102	David	King	31 Saint Johns	Brooklyn	NY	11217	M	P102	3
S103	Johnny	Lee	197 Ashland	Brooklyn	NY	11217	M	P103	1
S104	Lina	Wood	94 Madsen Ave	Staten Island	NY	10302	F	P104	2
S105	Li	White	806 Washington	Brooklyn	NY	11217	F	P105	5
S106	Ben	Walker	1857 Broadway	Manhattan	NY	10019	M	P106	6

Key: StudentID

FD1: StudentID → FirstName, LastName, Street, City, State, ZipCode, Gender, ParentID (fk), SessionID (fk)

FD2: ZipCode → City, State

1NF: Meets the definition of a relation

2NF: No partial Key dependencies

3NF: Transitive dependency exists:

StudentID → FirstName, LastName, Street, ZipCode (fk), Gender, ParentID (fk), SessionID (fk)

ZipCode → City, State

Solution: Split Student relations into two new relations named Student1 and ZipCode:

Student1 (StudentID (key), FirstName, LastName, Street, ZipCode (fk), Gender, ParentID (fk), SessionID (fk))

StudentID	FirstName	LastName	Street	ZipCode	Gender	ParentID	SessionID
S100	Rose	Smith	1745 Broadway	10019	F	P100	1
S101	Bella	Taylor	921 Post Ave	10302	F	P101	2
S102	David	King	31 Saint Johns	11217	M	P102	3
S103	Johnny	Lee	197 Ashland	11217	M	P103	1
S104	Lina	Wood	94 Madsen Ave	10302	F	P104	2
S105	Li	White	806 Washington	11217	F	P105	5
S106	Ben	Walker	1857 Broadway	10019	M	P106	6

Key: StudentID

FD1: StudentID → FirstName, LastName, Street, ZipCode (fk), Gender, ParentID (fk), SHSATID (fk), SessionID (fk)

1NF: Meets the definition of a relation

2NF: No partial Key dependencies

3NF: No Transitive dependencies

Note: ZipCode relation is in Parent1 relation so there is no need to create a second ZipCode relation.

3) **Teacher** (TeacherID (key), FirstName, LastName, Street, City, State, ZipCode, Gender, SessionID (fk))

TeacherID	FirstName	LastName	Street	City	State	ZipCode	Gender	SessionID
T100	Mary	Green	42 Saint Place	Manhattan	NY	10900	F	1
T101	James	Li	123 Hay St	Staten Island	NY	10314	F	1
T102	Jay	Anson	56 Kitty Place	Staten Island	NY	10314	M	2
T103	Danial	Bell	95 St 8av	Brooklyn	NY	11220	M	1
T104	Tina	Hope	480 Burn Ave	Manhattan	NY	10900	F	3
T105	Berry	Douglas	48 St 6av	Brooklyn	NY	11220	M	5
T106	Kriti	Jones	75 Hyden Ct	Staten Island	NY	10314	F	6

Key: TeacherID

FD1: TeacherID → FirstName, LastName, Street, City, State, ZipCode, Gender, SessionID (fk)

FD2: ZipCode → City, State

1NF: Meets the definition of a relation

2NF: No partial Key dependencies

3NF: Transitive dependency exists:

TeacherID → TeacherID (key), FirstName, LastName, Street, ZipCode (fk), Gender, SessionID (fk))

ZipCode → City, State

Solution: Split Teacher relations into two new relations named Teacher1 and ZipCode:

Teacher1 (TeacherID (key), FirstName, LastName, Street, ZipCode (fk), Gender, SessionID (fk))

TeacherID	FirstName	LastName	Street	ZipCode	Gender	SessionID
T100	Mary	Green	42 Saint Place	10900	F	1
T101	James	Li	123 Hay St	10314	F	1
T102	Jay	Anson	56 Kitty Place	10314	M	2
T103	Danial	Bell	95 St 8av	11220	M	1
T104	Tina	Hope	480 Burn Ave	10900	F	3
T105	Berry	Douglas	48 St 6av	11220	M	5
T106	Kriti	Jones	75 Hyden Ct	10314	F	6

Key: TecherID

FD1: TeacherID → FirstName, LastName, PhoneNumber, Street, ZipCode (fk), Gender, SessionID (fk)

1NF: Meets the definition of a relation

2NF: No partial Key dependencies

3NF: No Transitive dependencies

Note: ZipCode relation is in Parent1 relation so there is no need to create a second ZipCode relation.

4) **SHSATResult** (SHSATResultID (key), Result, Grade, HighSchool, StudentID (fk))

SHSATResultID	Result	Grade	HighSchool	StudentID
R100	Accepted	750	Staten Island Tech	S100
R101	Rejected	540	CSI High School	S101
R102	Rejected	650	Midwood High School	S102
R103	Accepted	725	Brooklyn Tech	S103
R104	Accepted	710	Staten Island Tech	S104
R105	Accepted	760	Brooklyn Tech	S105
R106	Rejected	400	Midwood Highschool	S106

Key: SHSATReultID

FD1: SHSATResultID → Result, Grade, HighSchool, StudentID (fk)

FD2: Result → HighSchool

1NF: Meets the definition of a relation

2NF: No partial Key dependencies

3NF: No Transitive dependencies

5) **Practice Test** (PracticeID (key), DateSubmit, Grade, SessionID (fk))

PracticeID	DateSubmit	Grade	SessionID
100	January 1, 2022	80	1
101	January 15, 2022	60	5
102	February 15, 2022	100	6
103	November 1, 2022	100	1
104	March 3, 2022	90	2
105	April 5, 2022	70	2
106	June 10, 2022	40	3

Key: PracticeID

FD1: PracticeID → Grade, DateSubmit, Grade, SessionID (fk)

1NF: Meets the definition of a relation

2NF: No partial Key dependencies

3NF: No Transitive dependencies

6) **Session** (SessionID (key), StartDateTime, EndDateTime, Season)

SessionID	StartDateTime	EndDateTime	Season
1	9 AM	12 PM	Summer
3	1 PM	2 PM	Summer
2	10 AM	1 PM	Fall
4	2 PM	3 PM	Fall
5	9:20 AM	12:30 PM	Summer, Fall
6	1:30 PM	2:30 PM	Summer, Fall

Key: SessionID

FD1: SessionID → StartDateTime, EndDateTime, Season

1NF: Season may be treated as a multi-valued attribute. In this case, Session is not in 1NF.

Solution: Split out Season into separate relations. For this example, however, we will keep Season as an attribute of the Session relation.

SessionID	StartDateTime	EndDateTime	Season
1	9 AM	12 PM	Summer
3	1 PM	2 PM	Summer
2	10 AM	1 PM	Fall
4	2 PM	3 PM	Fall
5	9:20 AM	12:30 PM	Summer
6	1:30 PM	2:30 PM	Summer
5	9:20 AM	12:30 PM	Fall
6	1:30 PM	2:30 PM	Fall

1NF: Meets the definition of a relation

2NF: No partial Key dependencies

3NF: No Transitive dependencies

Final Set of Relations:

Parent1 (ParentID (key), FirstName, LastName, Street, ZipCode (fk), Gender)

Student1 (StudentID (key), FirstName, LastName, Street, ZipCode (fk), Gender, ParentID (fk), SessionID (fk))

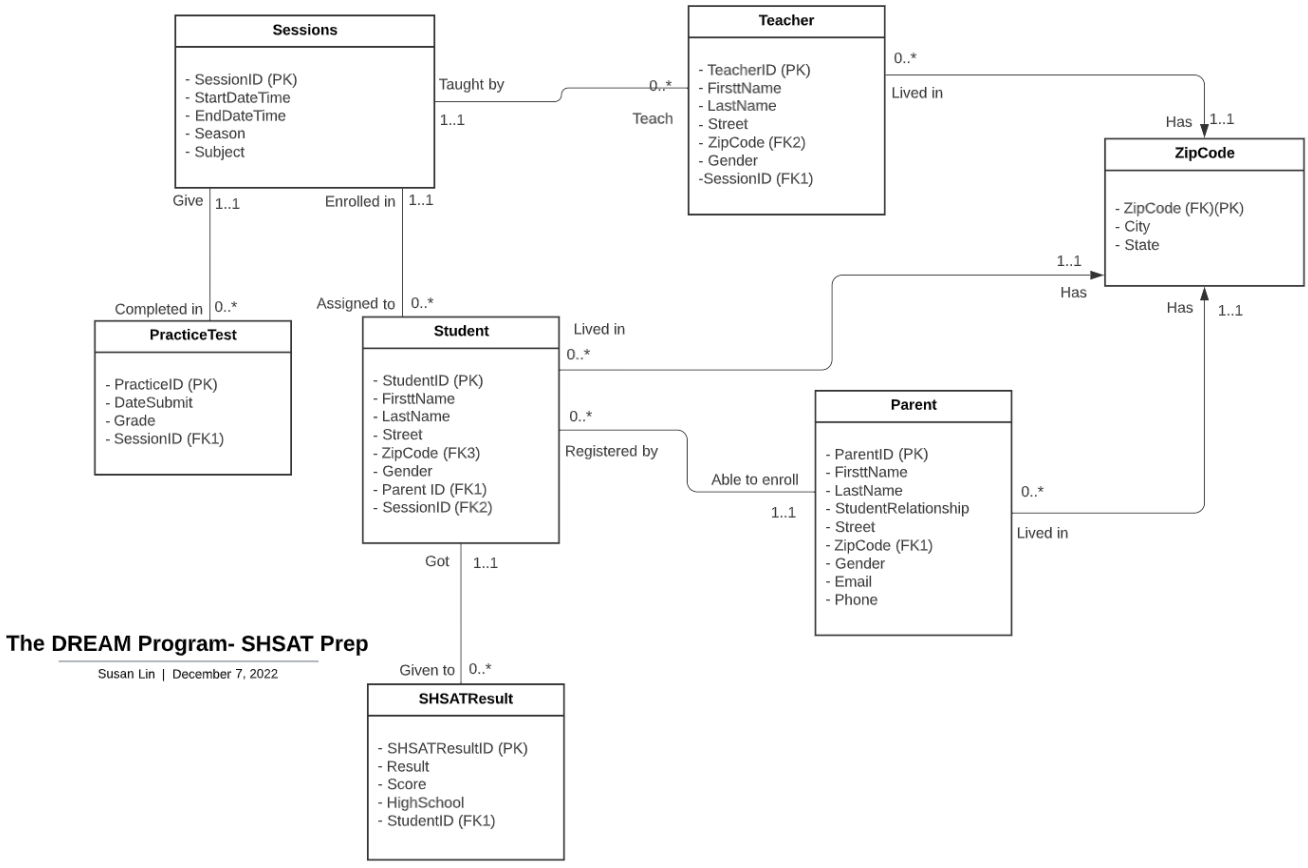
Teacher1 (TeacherID (key), FirstName, LastName, Street, ZipCode (fk), Gender, SessionID (fk))

SHSATResult1 (SHSATID (key), Result, Grade, HighSchool, StudentID (fk))

Practice Test (PracticeID (key), DateSubmit, Grade, SessionID (fk))

Session (SessionID (key), StartDateTime, EndDateTime, Season)

ZipCode (ZipCode (key), City, State)



Structured Query Language (SQL) to Create the Schema:

The following SQL code creates seven tables with the PRIMARY KEY and FOREIGN KEY constraints:

Parent1 (ParentID (key), FirstName, LastName, Street, ZipCode (fk), Gender)

```
CREATE TABLE Parent1(
ParentID VARCHAR (10) NOT NULL,
FirstName VARCHAR(35),
LastName VARCHAR(35),
Street VARCHAR(35),
ZipCode VARCHAR(12),
Gender VARCHAR(2),
CONSTRAINT pk_parent1
PRIMARY KEY (ParentID),
CONSTRAINT FK_Zipcode1 foreign key (ZipCode)
references ZipCode (ZipCode)
ON UPDATE CASCADE ON DELETE NO ACTION
);
```

Student1 (StudentID (key), FirstName, LastName, Street, ZipCode (fk), Gender, ParentID (fk), SessionID (fk))

```
CREATE TABLE Student1(
StudentID VARCHAR (10) NOT NULL,
FirstName VARCHAR(35),
LastName VARCHAR(35),
Street VARCHAR(35),
ZipCode VARCHAR(12),
Gender VARCHAR(2),
ParentID VARCHAR(10),
SessionID VARCHAR(15),
CONSTRAINT pk_student1
```

```
PRIMARY KEY (StudentID),
CONSTRAINT FK_Zipcode2 foreign key (ZipCode)
references ZipCode (ZipCode),
CONSTRAINT FK_Parent foreign key (ParentID)
references Parent1 (ParentID),
CONSTRAINT FK_Session foreign key (SessionID)
references Sessions (SessionID)
ON UPDATE CASCADE ON DELETE NO ACTION
);
```

Teacher1 (TeacherID (key), FirstName, LastName, Street, ZipCode (fk), Gender, SessionID (fk))

```
CREATE TABLE Teacher1(
TeacherID VARCHAR (10) NOT NULL,
FirstName VARCHAR(35),
LastName VARCHAR(35),
Street VARCHAR(35),
ZipCode VARCHAR(12),
Gender VARCHAR(2),
SessionID VARCHAR(15),
CONSTRAINT pk_teacher1 PRIMARY KEY (TeacherID),
CONSTRAINT FK_Zipcode3 foreign key (ZipCode)
references ZipCode (ZipCode),
CONSTRAINT FK_Session1 foreign key (SessionID)
references Sessions (SessionID)
ON UPDATE CASCADE ON DELETE NO ACTION
);
```

SHSATResult1 (SHSATID (key), Result, Grade, StudentID (fk))

```
CREATE TABLE SHSATResult (
SHSATID VARCHAR (15) NOT NULL,
Result VARCHAR (15),
```

```
Grade INTEGER,
HighSchool VARCHAR (25),
StudentID VARCHAR (10),
CONSTRAINT pk_SHSAT PRIMARY KEY (SHSATID),
CONSTRAINT FK_Student5 foreign key (StudentID)
references Student1 (StudentID)
);
```

Practice Test (PracticeID (key), DateSubmit, Grade, SessionID (fk))

```
CREATE TABLE PracticeTest(
PracticeID VARCHAR (15) NOT NULL,
DateSubmit DATE,
Grade NUMBER,
SessionID VARCHAR(15),
CONSTRAINT pk_Practice PRIMARY KEY (PracticeID),
CONSTRAINT FK_Session2 foreign key (SessionID)
references SessionSession (SessionID)
);
```

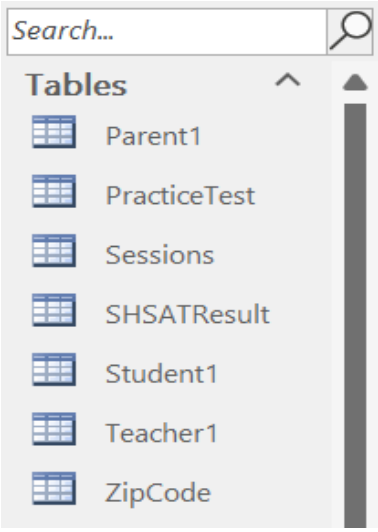
Session (SessionID (key), StartDateTime, EndDateTime, Season)

```
CREATE TABLE Sessions(
SessionID VARCHAR (15) NOT NULL,
StartDateTime DATE,
EndDateTime DATE,
Season VARCHAR (10) ,
CONSTRAINT pk_Session1
PRIMARY KEY (SessionID)
);
```

ZipCode (ZipCode (key), City, State)

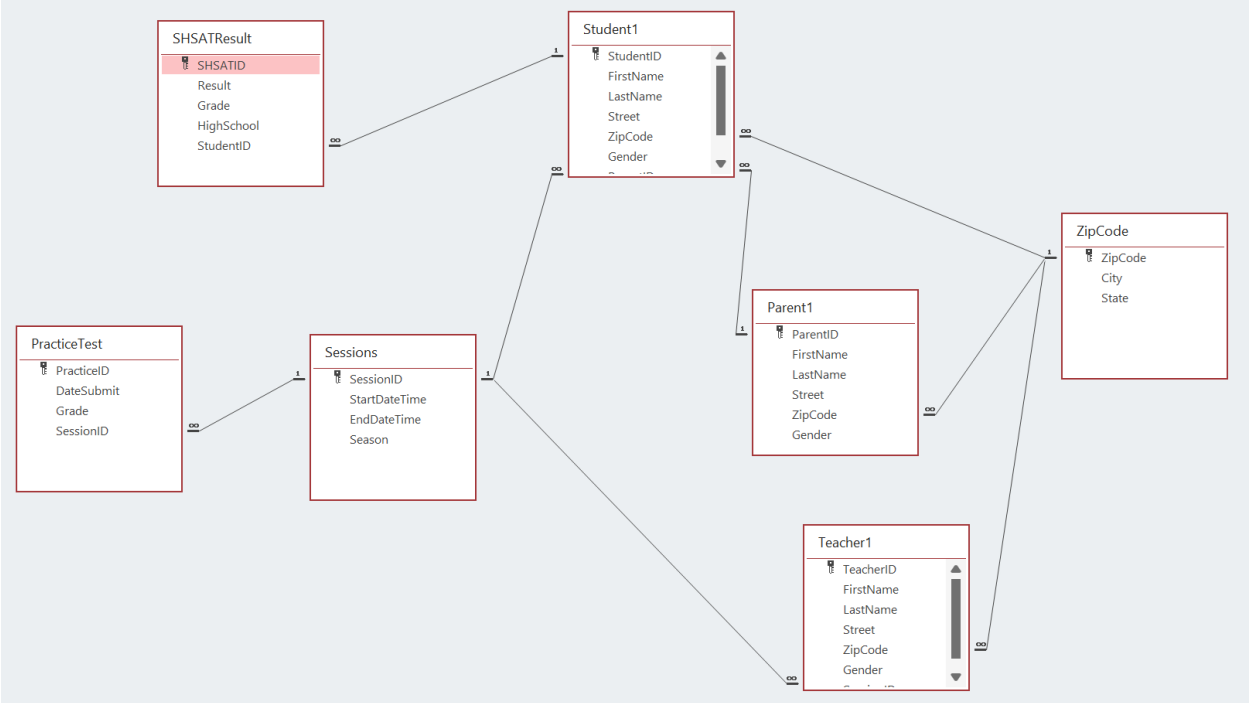
```
CREATE TABLE ZipCode(
ZipCode VARCHAR (12) NOT NULL,
City VARCHAR (36),
State VARCHAR (4),
CONSTRAINT pk_zipcode
PRIMARY KEY (zipcode)
);
```

After creating the tables and adding the foreign key constraints, the database schema now looks like the following:



Relationship View

Using the Relationship View under Database Tools, we can see the relationships (foreign keys) between the tables:



Adding Data to the Tables using SQL INSERT Statements For Some Tables:

```
INSERT INTO ZipCode (ZipCode, City, State) VALUES ('10302',
'Staten Island', 'NY');
INSERT INTO ZipCode (ZipCode, City, State) VALUES ('11217',
'Brooklyn', 'NY');
INSERT INTO ZipCode ( ZipCode, City, State ) VALUES ('10019',
'Manhattan', 'NY');
INSERT INTO ZipCode ( ZipCode, City, State ) VALUES ('10900',
'Manhattan', 'NY');
INSERT INTO ZipCode ( ZipCode, City, State ) VALUES ('10314',
'Staten island', 'NY');
INSERT INTO ZipCode ( ZipCode, City, State ) VALUES ('11220',
'Brooklyn', 'NY');
```

```
INSERT INTO Sessions ( SessionID, StartDateTime, EndDateTime,
Season ) VALUES ('1', '9 AM', '12 PM', 'Summer');
INSERT INTO Sessions ( SessionID, StartDateTime, EndDateTime,
Season ) VALUES ('2', '10 AM', '1 PM', 'Fall');
INSERT INTO Sessions ( SessionID, StartDateTime, EndDateTime,
Season ) VALUES ('3', '1 PM', '2 PM', 'Summer');
INSERT INTO Sessions ( SessionID, StartDateTime, EndDateTime,
Season ) VALUES ('4', '2 PM', '3 PM', 'Fall');
INSERT INTO Sessions ( SessionID, StartDateTime, EndDateTime,
Season ) VALUES ('5', '9:20 AM', '12:30 PM', 'Summer, Fall');
INSERT INTO Sessions ( SessionID, StartDateTime, EndDateTime,
Season ) VALUES ('6', '1:30 PM', '2:30 PM', 'Summer, Fall');

INSERT INTO Parent1 ( ParentID, FirstName, LastName, Street,
ZipCode, Gender ) VALUES ('P100', 'Elisa', 'Smith', '1745
Broadway', '10019', 'F');

INSERT INTO Student1 ( StudentID, FirstName, LastName, Street,
ZipCode, Gender, ParentID, SessionID ) VALUES ('S102', 'David',
'King', '31 Saint Johns', '11217', 'M', 'P102', '3');
```

Adding Different Queries using SQL SELECT:

```
SELECT StudentID, FirstName, LastName
FROM Student1;
```

StudentID	FirstName	LastName
S100	Rose	Smith
S101	Bella	Taylor
S102	David	King
S103	Johnny	Lee
S104	Lina	Wood
S105	Li	White
S106	Ben	Walker

This shows the student's ID, first, and last name.

```
SELECT FirstName, LastName, Result, HighSchool
FROM Student1, SHSATResult
WHERE Student1.StudentID=SHSATResult.StudentID
AND Result = 'Accepted';
```

FirstName	LastName	Result	HighSchool
Rose	Smith	Accepted	SI Tech
Johnny	Lee	Accepted	Brooklyn Tech
Lina	Wood	Accepted	SI Tech
Li	White	Accepted	Brooklyn Tech

This shows the students who got accepted to Specialized High School

```
SELECT FirstName, LastName, Result, HighSchool
FROM Student1, SHSATResult
WHERE Student1.StudentID=SHSATResult.StudentID
AND Result = 'Rejected';
```

FirstName	LastName	Result	HighSchool
Ben	Walker	Rejected	Midwood High
Bella	Taylor	Rejected	CSI High School
David	King	Rejected	Midwood High

This shows the students who got rejected Specialized High School

```
SELECT StudentID, FirstName, LastName
FROM Student1
WHERE SessionID='1';
```

StudentID	FirstName	LastName
S100	Rose	Smith
S103	Johnny	Lee

This shows the students who are in SessionID 1

```
SELECT *
FROM Teacher1
ORDER BY LastName;
```

TeacherID	FirstName	LastName	Street	ZipCode	Gender	SessionID
T102	Jay	Anson	56 Kitty Place	10314	M	2
T103	Danial	Bell	95 St 8av	11220	M	1
T105	Berry	Douglas	48 St 6av	11220	M	5
T100	Mary	Green	42 Saint Place	10900	F	1
T104	Tina	Hope	480 Burn Ave	10900	F	3
T106	Kriti	Jones	75 Hyden Ct	10314	F	6
T101	James	Li	123 Hay St	10314	F	1

The table is organized based on the teacher's last name.

```
SELECT COUNT(*) AS NumStudent, AVG(SHSATResult.Grade) AS
SHSATGrade
FROM SHSATResult
GROUP BY SHSATResult.Grade
HAVING (((AVG(SHSATResult.Grade) < 700)))
ORDER BY SHSATResult.Grade;
```

NumStudent	SHSATGrade
1	540
2	650

This shows the number of students who got less than 700.

```
SELECT SessionID, AVG(Grade) AS AverageGrade
FROM PracticeTest
GROUP BY SessionID;
```

SessionID	AverageGrac
1	90
2	80
3	40
5	60
6	100

This shows the average practice grade in each session.

```
SELECT StudentID, FirstName, LastName
FROM Student1
WHERE ZipCode IN
(Select ZipCode
From ZipCode
Where City='Staten Island');
```

StudentID	FirstName	LastName
S101	Bella	Taylor
S104	Lina	Wood

This shows students that are living in Staten Island.

```
SELECT FirstName, LastName, HighSchool
FROM Student1 INNER JOIN SHSATResult
ON Student1.StudentID = SHSATResult.StudentID
```

FirstName	LastName	HighSchool
Rose	Smith	SI Tech
Bella	Taylor	CSI High School
David	King	Midwood High
Johnny	Lee	Brooklyn Tech
Lina	Wood	SI Tech
Li	White	Brooklyn Tech
Ben	Walker	Midwood High
Yuan	Li	SI Tech

This shows which school the students went

```
SELECT FirstName, LastName
FROM Teacher1
WHERE SessionID = '1'
UNION
SELECT FirstName, LastName
FROM Teacher1, Sessions
WHERE Teacher1.SessionID = Sessions.SessionID
ORDER BY FirstName ASC;
```

FIrstName	LastName
Berry	Douglas
Danial	Bell
James	Li
Jay	Anson
Kriti	Jones
Mary	Green
Tina	Hope

This shows the teachers and students who are in sessionID 1

```
SELECT Student1.FirstName, Student1.LastName,
Teacher1.FirstName, Teacher1.LastName
FROM Student1, Sessions, Teacher1
WHERE Student1.SessionID = Sessions.SessionID
AND Sessions.SessionID = Teacher1.SessionID
ORDER BY Student1.FirstName;
```

Student1.Firs	Student1.Las	Teacher1.Fir	Teacher1.Las
Bella	Taylor	Jay	Anson
Ben	Walker	Kriti	Jones
David	King	Tina	Hope
Johnny	Lee	Danial	Bell
Johnny	Lee	James	Li
Johnny	Lee	Mary	Green
Li	White	Berry	Douglas
Lina	Wood	Jay	Anson
Rose	Smith	Danial	Bell
Rose	Smith	James	Li
Rose	Smith	Mary	Green
Yuan	Li	Danial	Bell
Yuan	Li	James	Li
Yuan	Li	Mary	Green

This shows the teachers that taught the students.

```
SELECT ParentID, FirstName, LastName
FROM Parent1
WHERE ZipCode IN
(SELECT ZipCode
FROM ZipCode
WHERE City = 'Brooklyn');
```

ParentID	FirstName	LastName
P102	Joe	King
P103	John	Lee
P105	Mika	White

This shows the parents that lived in Brooklyn

```
SELECT ParentID, FirstName, LastName
FROM Parent1;
```

ParentID	FirstName	LastName
P100	Elisa	Smith
P101	Michelle	Taylor
P102	Joe	King
P103	John	Lee
P104	Lisa	Wood
P105	Mika	White
P106	Fiona	Walker

This shows parent's information

```
SELECT FirstName, LastName, State, City
FROM Teacher1 INNER JOIN ZipCode
ON Teacher1.ZipCode = ZipCode.ZipCode;
```

FirstName	LastName	State	City
James	Li	NY	Staten Island
Jay	Anson	NY	Staten Island
Kriti	Jones	NY	Staten Island
Mary	Green	NY	Manhatten
Tina	Hope	NY	Manhatten
Danial	Bell	NY	Brooklyn
Berry	Douglas	NY	Brooklyn

This shows the teacher's state and city

```
SELECT *
FROM ZipCode;
```

ZipCode	City	State
10019	Manhattan	NY
10302	Staten Island	NY
10314	Staten Island	NY
10900	Manhatten	NY
11217	Brooklyn	NY
11220	Brooklyn	NY

This shows the zipcode

```
SELECT FirstName, LastName
FROM Student1
WHERE SessionID = '3'
UNION
SELECT FirstName, LastName
FROM Student1, Sessions
WHERE Student1.SessionID = Sessions.SessionID
ORDER BY FirstName;
```

FirstName	LastName
Bella	Taylor
Ben	Walker
David	King
Johnny	Lee
Li	White
Lina	Wood
Rose	Smith
Yuan	Li

Students in SessionID 3