Jeffrey Kao
4/23/2018
IST 659 - Project

## Database for a High School

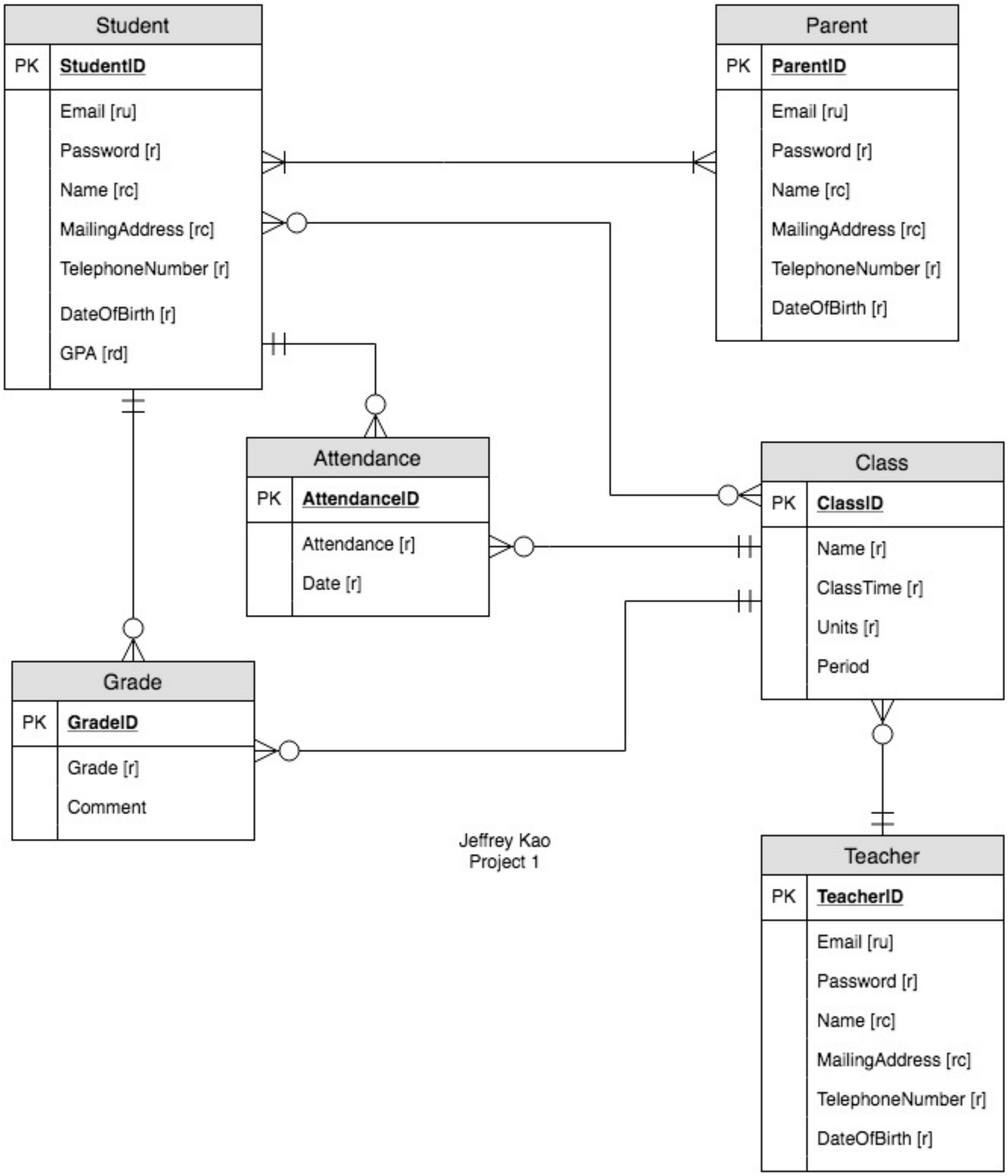
Project Summary

This project focuses on designing a database for a school management system for a high school. The management system is to maintain both attendance and grades for students at a school. It will also keep track of parents for each student. This project will solve questions like: Who are the students for each parent? How many students are in a class? Who has the highest GPAs at the school? What are grades for each student for each class? How many classes does each teacher teach? Which classes are the biggest at the school?

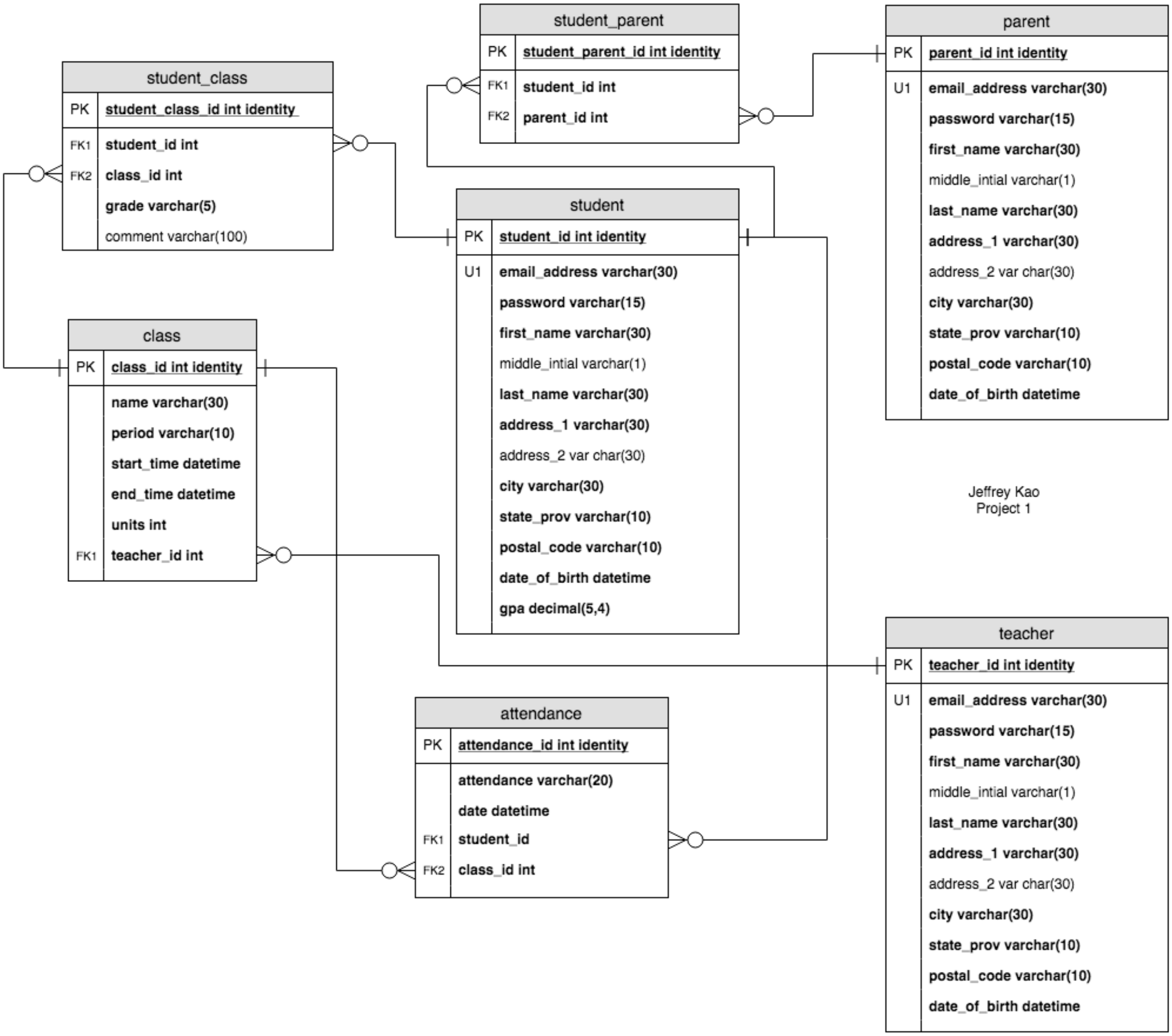
The database will store email, address, phone number for students, parents, and teachers. These will be the users so a password will be stored as well for the management system. Attendance stores an attendance status and date. While for every class, there is name, units, time of class, and period. All the grades will have both a letter grade and possibly a comment. Each student should have one or more than one parent. And each parent should have one or more student. Every student can take several classes and the classes can have many students. A student receives one grade for every class that he takes. Teachers can teach many classes. Every class and student has an attendance record for every day.

| **Entity** | **Attribute** |
|---|---|
| Student | Email [ru]<br>Password [r]<br>Name [rc]<br>Mailing Address [rc]<br>Telephone Number [r]<br>Date of Birth [r]<br>GPA [dr] |
| Parent | Email [ru]<br>Password [r]<br>Name [rc]<br>Mailing Address [rc]<br>Telephone Number [r]<br>Date of Birth [r] |
| Teacher | Email [ru]<br>Password [r]<br>Name [rc]<br>Mailing Address [rc]<br>Telephone Number [r]<br>Date of Birth [r] |
| Grade | Grade [r]<br>Comment |
| Attendance | Attendance [r]<br>Date [r] |

| | |
|---|---|
| Class | Name [r]<br>Class Time [r]<br>Units [r]<br>Period [r] |
| **Relationships** | |
| Each Student has 1 or more Parents, each Parent has 1 or more Students.<br>Each Student has 0 or more Classes, each Class can 1 or more Students.<br>Each Student has 0 or more Grades, each Grade has only 1 Student.<br>Each Student has 0 or more Attendances, each Attendance has only 1 Student.<br>Each Class has 0 or more Attendances, each Attendance has only 1 Class.<br>Each Class has 0 or more Grades, each Grade has only 1 Class.<br>Each Teacher has 0 or more Classes, each Class has only 1 Teacher. | |



Jeffrey Kao
Project 1



| student_class | |
|---|---|
| PK | **student_class_id int identity** |
| FK1 | **student_id int** |
| FK2 | **class_id int** |
| | **grade varchar(5)** |
| | comment varchar(100) |

| student_parent | |
|---|---|
| PK | **student_parent_id int identity** |
| FK1 | **student_id int** |
| FK2 | **parent_id int** |

| parent | |
|---|---|
| PK | **parent_id int identity** |
| U1 | **email_address varchar(30)** |
| | **password varchar(15)** |
| | **first_name varchar(30)** |
| | middle_intial varchar(1) |
| | **last_name varchar(30)** |
| | **address_1 varchar(30)** |
| | address_2 var char(30) |
| | **city varchar(30)** |
| | **state_prov varchar(10)** |
| | **postal_code varchar(10)** |
| | **date_of_birth datetime** |

| student | |
|---|---|
| PK | **student_id int identity** |
| U1 | **email_address varchar(30)** |
| | **password varchar(15)** |
| | **first_name varchar(30)** |
| | middle_intial varchar(1) |
| | **last_name varchar(30)** |
| | **address_1 varchar(30)** |
| | address_2 var char(30) |
| | **city varchar(30)** |
| | **state_prov varchar(10)** |
| | **postal_code varchar(10)** |
| | **date_of_birth datetime** |
| | **gpa decimal(5,4)** |

| class | |
|---|---|
| PK | **class_id int identity** |
| | **name varchar(30)** |
| | **period varchar(10)** |
| | **start_time datetime** |
| | **end_time datetime** |
| | **units int** |
| FK1 | **teacher_id int** |

Jeffrey Kao
Project 1

| teacher | |
|---|---|
| PK | **teacher_id int identity** |
| U1 | **email_address varchar(30)** |
| | **password varchar(15)** |
| | **first_name varchar(30)** |
| | middle_intial varchar(1) |
| | **last_name varchar(30)** |
| | **address_1 varchar(30)** |
| | address_2 var char(30) |
| | **city varchar(30)** |
| | **state_prov varchar(10)** |
| | **postal_code varchar(10)** |
| | **date_of_birth datetime** |

| attendance | |
|---|---|
| PK | **attendance_id int identity** |
| | **attendance varchar(20)** |
| | **date datetime** |
| FK1 | **student_id** |
| FK2 | **class_id int** |

SQL DDL Commands

```
-- Creating Student table
CREATE TABLE Student (
       StudentID int identity,
       EmailAddress varchar(50) not null,
       Passcode varchar(30) not null,
       FirstName varchar(30) not null,
       MiddleInitial varchar(1),
       LastName varchar(30) not null,
       Address1 varchar(30) not null,
       Address2 varchar(30),
       City varchar(30) not null,
       StateProv varchar(15) not null,
       PostalCode varchar(10) not null,
       DateOfBirth date not null,
       GPA decimal(5,4) not null,
       CONSTRAINT PK_Student PRIMARY KEY (StudentID),
       CONSTRAINT U1_Student UNIQUE(EmailAddress)
)

-- Creating Parent table
CREATE TABLE Parent (
       ParentID int identity,
       EmailAddress varchar(50) not null,
       Passcode varchar(30) not null,
       FirstName varchar(30) not null,
       MiddleInitial varchar(1),
       LastName varchar(30) not null,
       Address1 varchar(30) not null,
       Address2 varchar(30),
       City varchar(30) not null,
       StateProv varchar(15) not null,
       PostalCode varchar(10) not null,
       DateOfBirth date not null,
       CONSTRAINT PK_Parent PRIMARY KEY (ParentID),
       CONSTRAINT U1_Parent UNIQUE(EmailAddress)
)

-- Creating Teacher table
CREATE TABLE Teacher (
       TeacherID int identity,
       EmailAddress varchar(50) not null,
       Passcode varchar(30) not null,
       FirstName varchar(30) not null,
       MiddleInitial varchar(1),
       LastName varchar(30) not null,
       Address1 varchar(30) not null,
       Address2 varchar(30),
       City varchar(30) not null,
       StateProv varchar(15) not null,
       PostalCode varchar(10) not null,
       DateOfBirth date not null,
       CONSTRAINT PK_Teacher PRIMARY KEY (TeacherID),
       CONSTRAINT U1_Teacher UNIQUE(EmailAddress)
)

-- Creating Class table
CREATE TABLE Class (
       ClassID int identity,
       Name varchar(30) not null,
       Period varchar(10)  not null,
       StartTime time  not null,
       EndTime time  not null,
       units int  not null,
       TeacherID int  not null,
       CONSTRAINT PK_Class PRIMARY KEY (ClassID),
       CONSTRAINT FK1_Class FOREIGN KEY (TeacherID) REFERENCES Teacher(TeacherID)
)

-- Creating the Student Parent Table
CREATE TABLE StudentParent (
       StudentParentID int identity,
       StudentID int not null,
       ParentID int not null,
       CONSTRAINT PK_StudentParent PRIMARY KEY (StudentParentID),
       CONSTRAINT FK1_StudentParent FOREIGN KEY (StudentID) REFERENCES Student(StudentID),
       CONSTRAINT FK2_StudentParent FOREIGN KEY (ParentID) REFERENCES Parent(ParentID)
)

-- Creating the Student Class Table
CREATE TABLE StudentClass (
       StudentClassID int identity,
       StudentID int not null,
       ClassID int not null,
       Grade varchar(5) not null,
       Comment varchar(100),
       CONSTRAINT PK_StudentClass PRIMARY KEY (StudentClassID),
       CONSTRAINT FK1_StudentClass FOREIGN KEY (StudentID) REFERENCES Student(StudentID),
       CONSTRAINT FK2_StudentClass FOREIGN KEY (ClassID) REFERENCES Class(ClassID)
)

-- Creating Attendance Table
CREATE TABLE Attendance (
       AttendanceID int identity,
       AttendanceStatus varchar(20) not null,
       AttendanceDate datetime not null,
       StudentID int not null,
       ClassID int not null,
       CONSTRAINT PK_Attendance PRIMARY KEY (AttendanceID),
       CONSTRAINT FK1_Attendance FOREIGN KEY (StudentID) REFERENCES Student(StudentID),
       CONSTRAINT FK2_Attendance FOREIGN KEY (ClassID) REFERENCES Class(ClassID)
)
```

## SQL DML INSERT Statements

```
INSERT INTO Student
        (EmailAddress, Passcode, FirstName, MiddleInitial, LastName, Address1,
        City, StateProv, PostalCode, DateOfBirth, GPA)
VALUES
        ('jkao01@syr.edu', 'password1234', 'Jeffrey', 'L','Kao', '63 W Pamela Rd',
        'Arcadia', 'CA', '91007', '12/23/1987', 3.8875)

INSERT INTO Student
        (EmailAddress, Passcode, FirstName, MiddleInitial, LastName, Address1,
        City, StateProv, PostalCode, DateOfBirth, GPA)
VALUES
        ('tiffanylkao@gmail.com', 'password1234', 'Tiffany', 'L','Kao', '63 W Pamela Rd',
        'Arcadia', 'CA', '91007', '07/27/1992', 3.9875)

INSERT INTO Parent
        (EmailAddress, Passcode, FirstName, MiddleInitial, LastName, Address1,
        City, StateProv, PostalCode, DateOfBirth)
VALUES
        ('stevekao@yahoo.com', 'password1234', 'Steve', 'H','Kao', '63 W Pamela Rd',
        'Arcadia', 'CA', '91007', '01/31/1952')

INSERT INTO Teacher
```

```
        (EmailAddress, Passcode, FirstName, MiddleInitial, LastName, Address1,
        City, StateProv, PostalCode, DateOfBirth)
VALUES
        ('chad.harper@gmail.com', 'password1234', 'Chad', 'L','Harper', '900 South Crouse Ave',
        'Syracuse', 'NY', '13244', '02/13/1972')

INSERT INTO Class
        (Name, Period, StartTime, EndTime, units, TeacherID)
VALUES
        ('Biology', '1', '08:30', '09:15', 3, 1)

INSERT INTO StudentParent
        (StudentID, ParentID)
VALUES
        (1, 1)

INSERT INTO StudentParent
        (StudentID, ParentID)
VALUES
        (2, 1)

INSERT INTO StudentClass
        (StudentID, ClassID, Grade, Comment)
VALUES
        (1, 2, 'A', 'A pleasure to have in the class')

INSERT INTO StudentClass
        (StudentID, ClassID, Grade, Comment)
VALUES
        (2, 2, 'B', 'A good citizen')

INSERT INTO Attendance
        (AttendanceStatus, AttendanceDate, StudentID, ClassID)
VALUES
        ('Tardy', '07/02/2018', 1, 2)

INSERT INTO Attendance
        (AttendanceStatus, AttendanceDate, StudentID, ClassID)
VALUES
        ('Absent', '07/02/2018', 2, 2)

SELECT * FROM Student
SELECT * FROM Parent
SELECT * FROM Teacher
SELECT * FROM Class
SELECT * FROM StudentParent
SELECT * FROM StudentClass
SELECT * FROM Attendance
```

## More SQL DDL

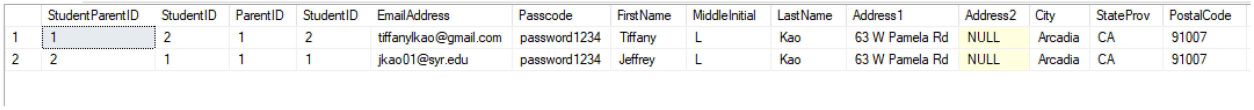
```
-- Function that returns how many students per class
CREATE FUNCTION dbo.StudentCount(@classID int)
RETURNS int as
BEGIN
        DECLARE @returnValue int
        SELECT @returnValue = COUNT(StudentID) FROM StudentClass
        WHERE StudentClass.ClassID = @classID
        RETURN @returnValue
END
GO
```

```
SELECT dbo.StudentCount(2)

-- Function that returns how many classes a Teacher teaches
CREATE FUNCTION dbo.ClassCount(@teacherID int)
RETURNS int as
BEGIN
        DECLARE @returnValue int
        SELECT @returnValue = COUNT(ClassID) FROM Class
        WHERE Class.TeacherID = @teacherID
        RETURN @returnValue
END
GO

SELECT dbo.ClassCount(1)

-- Create a view that shows Top 10 Students by GPA
CREATE VIEW HighestGPAs AS
        SELECT TOP 10
                *
        FROM Student
        ORDER BY GPA DESC
GO

SELECT * FROM HighestGPAs

-- Create Procedure that adds attendance records
CREATE PROCEDURE AddAttendanceRecord(@firstName varchar(30), @lastName varchar(30),
@attendanceStatus varchar(20), @className varchar(30))
AS
BEGIN
        DECLARE @studentID int
        SELECT @studentID = StudentID FROM Student
        WHERE FirstName = @firstName
        AND LastName = @lastName
        DECLARE @classID int
        SELECT @classID = ClassID FROM Class
        WHERE Name = @className
        INSERT INTO Attendance (AttendanceStatus, AttendanceDate, StudentID, ClassID)
        VALUES (@attendanceStatus, GETDATE(), @studentID, @classID)
        RETURN @@identity
END
GO


-- Create view that shows the 10 biggest classes
CREATE VIEW BiggestClasses AS
        SELECT TOP 10
        *,
        dbo.StudentCount(ClassID) as ClassCount
        FROM Class
        ORDER BY ClassCount DESC
GO

SELECT * FROM BiggestClasses
```

## SQL DML SELECT Statements

### Who are the students for a specific parent?

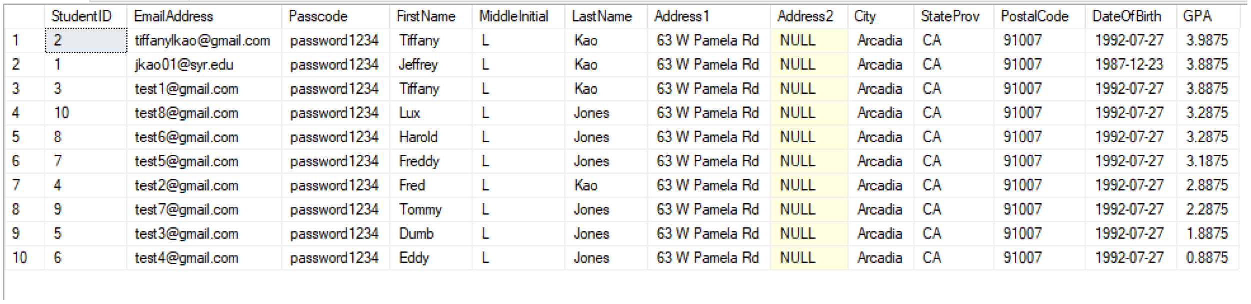
```
SELECT *
FROM StudentParent
JOIN Student ON StudentParent.StudentID = Student.StudentID
JOIN Parent ON StudentParent.ParentID = Parent.ParentID
WHERE Parent.FirstName = 'Steve'
AND Parent.LastName = ‘Kao'
```



| | StudentParentID | StudentID | ParentID | StudentID | EmailAddress | Passcode | FirstName | MiddleInitial | LastName | Address1 | Address2 | City | StateProv | PostalCode |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 2 | 1 | 2 | tiffanylkao@gmail.com | password1234 | Tiffany | L | Kao | 63 W Pamela Rd | NULL | Arcadia | CA | 91007 |
| 2 | 2 | 1 | 1 | 1 | jkao01@syr.edu | password1234 | Jeffrey | L | Kao | 63 W Pamela Rd | NULL | Arcadia | CA | 91007 |

Who has the highest GPAs at the school?

```
SELECT * FROM HighestGPAs
```



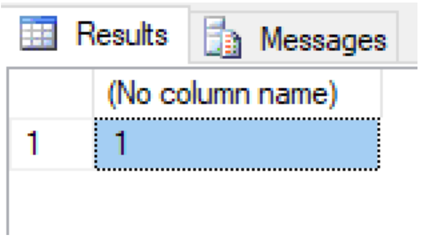
| | StudentID | EmailAddress | Passcode | FirstName | MiddleInitial | LastName | Address1 | Address2 | City | StateProv | PostalCode | DateOfBirth | GPA |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 2 | tiffanylkao@gmail.com | password1234 | Tiffany | L | Kao | 63 W Pamela Rd | NULL | Arcadia | CA | 91007 | 1992-07-27 | 3.9875 |
| 2 | 1 | jkao01@syr.edu | password1234 | Jeffrey | L | Kao | 63 W Pamela Rd | NULL | Arcadia | CA | 91007 | 1987-12-23 | 3.8875 |
| 3 | 3 | test1@gmail.com | password1234 | Tiffany | L | Kao | 63 W Pamela Rd | NULL | Arcadia | CA | 91007 | 1992-07-27 | 3.8875 |
| 4 | 10 | test8@gmail.com | password1234 | Lux | L | Jones | 63 W Pamela Rd | NULL | Arcadia | CA | 91007 | 1992-07-27 | 3.2875 |
| 5 | 8 | test6@gmail.com | password1234 | Harold | L | Jones | 63 W Pamela Rd | NULL | Arcadia | CA | 91007 | 1992-07-27 | 3.2875 |
| 6 | 7 | test5@gmail.com | password1234 | Freddy | L | Jones | 63 W Pamela Rd | NULL | Arcadia | CA | 91007 | 1992-07-27 | 3.1875 |
| 7 | 4 | test2@gmail.com | password1234 | Fred | L | Kao | 63 W Pamela Rd | NULL | Arcadia | CA | 91007 | 1992-07-27 | 2.8875 |
| 8 | 9 | test7@gmail.com | password1234 | Tommy | L | Jones | 63 W Pamela Rd | NULL | Arcadia | CA | 91007 | 1992-07-27 | 2.2875 |
| 9 | 5 | test3@gmail.com | password1234 | Dumb | L | Jones | 63 W Pamela Rd | NULL | Arcadia | CA | 91007 | 1992-07-27 | 1.8875 |
| 10 | 6 | test4@gmail.com | password1234 | Eddy | L | Jones | 63 W Pamela Rd | NULL | Arcadia | CA | 91007 | 1992-07-27 | 0.8875 |

Who has the biggest class sizes at the school?

```
SELECT * FROM BiggestClasses
```

How many classes does a teacher teach?

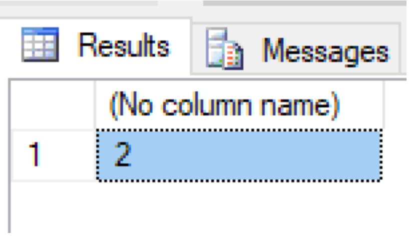
```
-- Find the TeacherID for a specific teacher
SELECT * FROM Teacher
WHERE Teacher.FirstName = ‘Chad’
AND Teacher.LastName = ‘Harper’

-- Use function to find out how many classes he teaches
SELECT dbo.ClassCount(1)
```



| | (No column name) |
|---|---|
| 1 | 1 |

How many students are in a particular class?

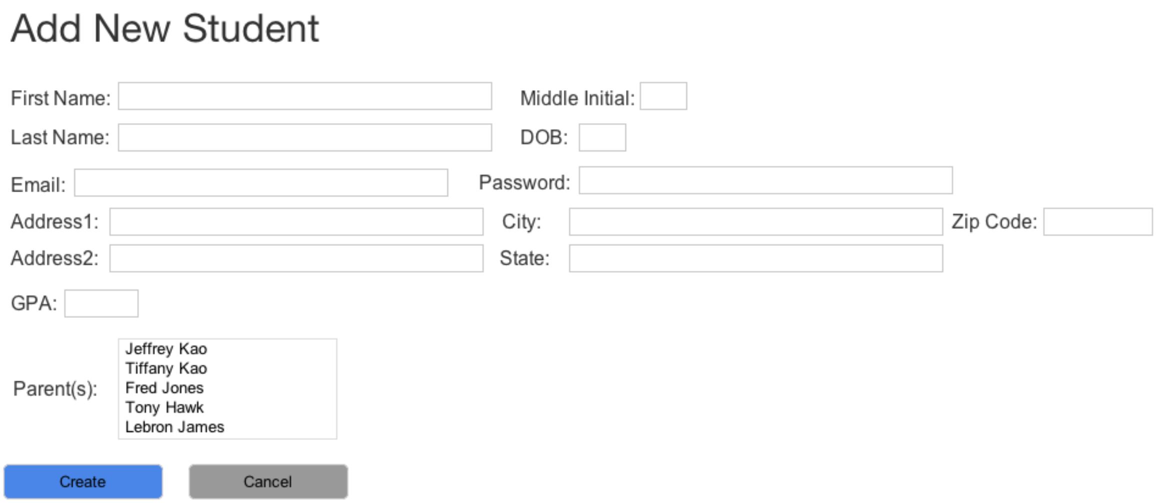
```
-- Find the ClassID for a specific class
SELECT * FROM Class
WHERE Class.Name = ‘Biology’

-- Use function to find out how many students are in the class
SELECT dbo.StudentCount(2)
```

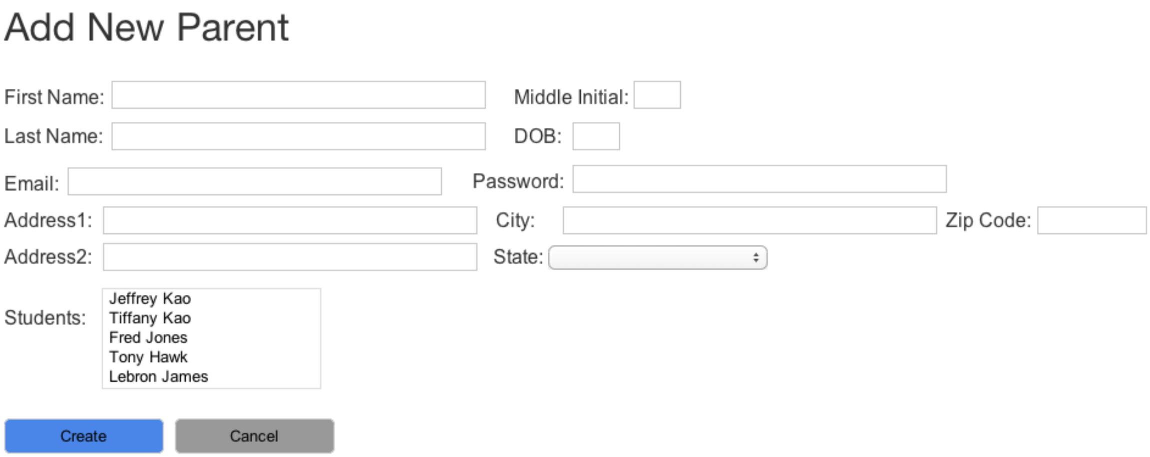


|   | (No column name) |
|---|---|
| 1 | 2 |

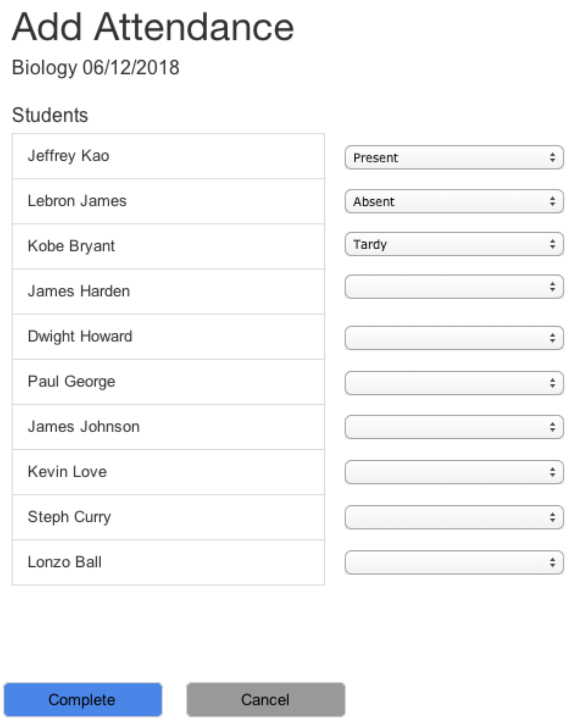
GUI Prototypes

Adding a new student and new parent views





Adding a new attendance record for a particular class the view for a teacher



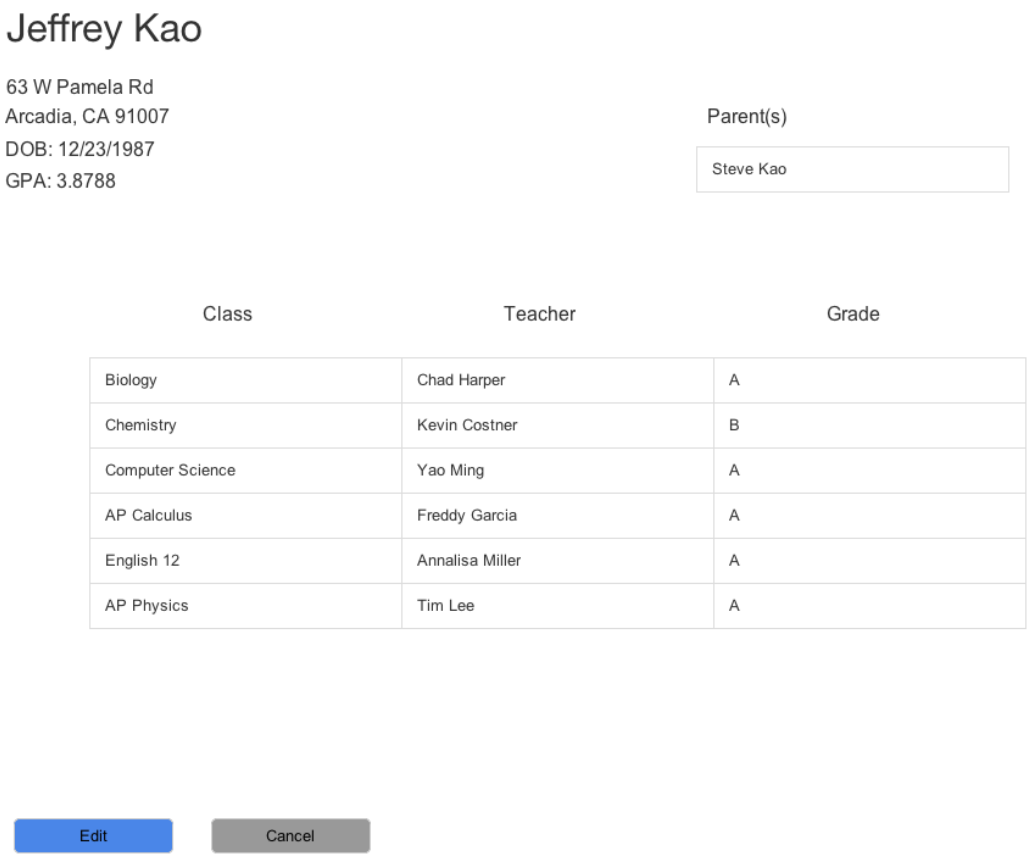
# Add Attendance

Biology 06/12/2018

Students

| | |
|---|---|
| Jeffrey Kao | Present |
| Lebron James | Absent |
| Kobe Bryant | Tardy |
| James Harden | |
| Dwight Howard | |
| Paul George | |
| James Johnson | |
| Kevin Love | |
| Steph Curry | |
| Lonzo Ball | |

Unique Student view



## Jeffrey Kao

63 W Pamela Rd
Arcadia, CA 91007
DOB: 12/23/1987
GPA: 3.8788

Parent(s)

Steve Kao

| Class | Teacher | Grade |
|---|---|---|
| Biology | Chad Harper | A |
| Chemistry | Kevin Costner | B |
| Computer Science | Yao Ming | A |
| AP Calculus | Freddy Garcia | A |
| English 12 | Annalisa Miller | A |
| AP Physics | Tim Lee | A |

Reflection

What assumptions did you have at the start of your project that changed by the end?

I assumed that the database would only need 3-4 tables at most before I started the project. I soon realized after I started that making more tables actually organized that data better and made it easier to query and find data that I needed.

The next time you do this, what will be different?

Next time I do this I would do this in a different database. I am not particularly fond of the Microsoft ecosystem and I would prefer to do this on a Unix based machine.

Regardless of whether you go through these steps again, how do you think it will inform your approach to data as an information professional?

I definitely got a better look a how data management and how databases are created. I know that even this small-scale database required a lot tedious steps and checking. Now imagine this on larger scale and it seems daunting at first by I believe that I am prepared.