Javier's School for Gifted Youngsters

Database Proposal 2016

By Lizeth Sánchez

3399 North Rd
Poughkeepsie, NY 12601

p. 575-3000 x2601
f. 575-3605

Javiers.school@gifted.edu
Jschool.edu

Table of Contents

Executive Summary

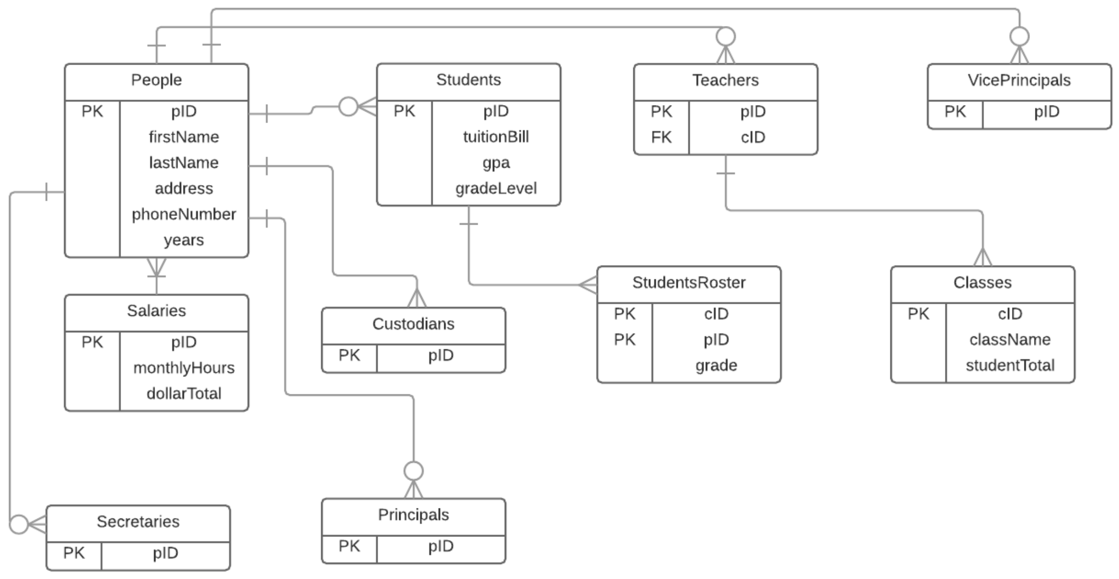
Overview

This proposal is made to model the structure of a high school; for this case, the fictitious Javier's School for Gifted Youngsters was created. In a high school setting, there is a number of roles that have to be separated and documented in order to grant the right permissions and right attributes to the right people.

Objectives

Some objectives of this database proposal are to keep track of students in the school, to keep a record of various faculty and their different permissions, to record and calculate salaries, and to generally organize all the necessary data for this school.

Entity Relation Diagram



Tables

People

Purpose: This table stores the various common information that is associated with each person in the school regardless of their role in the school.

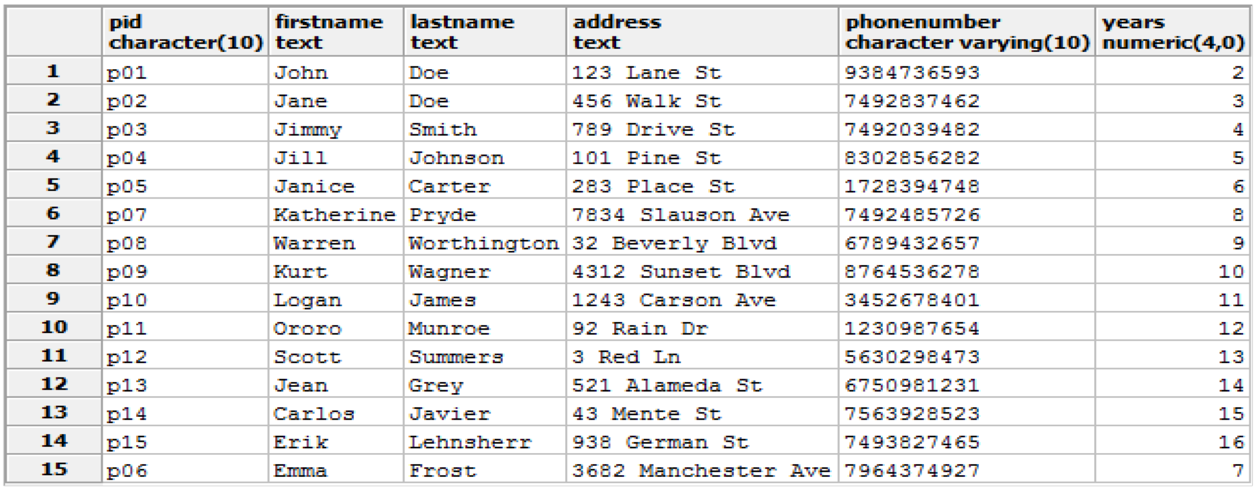
Create Statement:

```
CREATE TABLE People (
  pID char(10) NOT NULL,
  firstName TEXT,
  lastName TEXT,
  address TEXT,
  phoneNumber VARCHAR(10),
  years NUMERIC(4),
  PRIMARY KEY (pID)
);
```

Functional Dependencies:

pID → firstName, lastName, address, phoneNumber, years

Sample Data:



	pid character(10)	firstname text	lastname text	address text	phonenumber character varying(10)	years numeric(4,0)
1	p01	John	Doe	123 Lane St	9384736593	2
2	p02	Jane	Doe	456 Walk St	7492837462	3
3	p03	Jimmy	Smith	789 Drive St	7492039482	4
4	p04	Jill	Johnson	101 Pine St	8302856282	5
5	p05	Janice	Carter	283 Place St	1728394748	6
6	p07	Katherine	Pryde	7834 Slauson Ave	7492485726	8
7	p08	Warren	Worthington	32 Beverly Blvd	6789432657	9
8	p09	Kurt	Wagner	4312 Sunset Blvd	8764536278	10
9	p10	Logan	James	1243 Carson Ave	3452678401	11
10	p11	Ororo	Munroe	92 Rain Dr	1230987654	12
11	p12	Scott	Summers	3 Red Ln	5630298473	13
12	p13	Jean	Grey	521 Alameda St	6750981231	14
13	p14	Carlos	Javier	43 Mente St	7563928523	15
14	p15	Erik	Lehnsherr	938 German St	7493827465	16
15	p06	Emma	Frost	3682 Manchester Ave	7964374927	7

Students

Purpose: This table stores the students in the school so as to keep a record of their relevant information including their tuition bill, their grade point average, and their grade level.

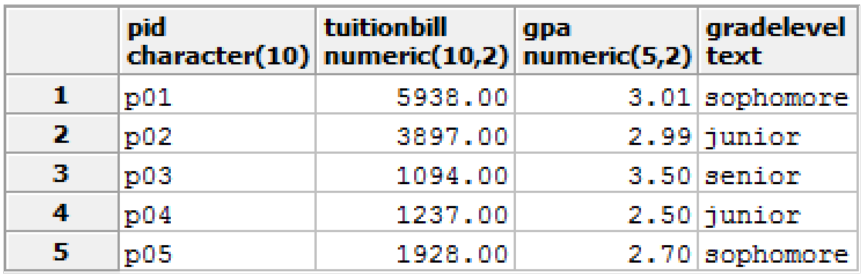
Create Statement:

```
CREATE TABLE Students (
  pID char(10) NOT NULL REFERENCES People,
  tuitionBill NUMERIC(10,2),
  gpa DECIMAL(5,2),
  gradeLevel TEXT,
  PRIMARY KEY (pID)
);
```

Functional Dependencies:

pID → tuitionBill, gpa, gradeLevel

Sample Data:



	pid character(10)	tuitionbill numeric(10,2)	gpa numeric(5,2)	gradelevel text
1	p01	5938.00	3.01	sophomore
2	p02	3897.00	2.99	junior
3	p03	1094.00	3.50	senior
4	p04	1237.00	2.50	junior
5	p05	1928.00	2.70	sophomore

Teachers

Purpose: This table stores the teachers in the school so as to keep a record of them and the ID of the class they teach.

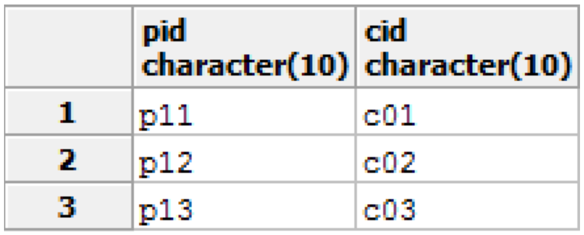
Create Statement:

```
CREATE TABLE Teachers (
  pID char(10) NOT NULL REFERENCES People,
  cID CHAR(10) NOT NULL REFERENCES Classes,
  PRIMARY KEY (pID)
);
```

Functional Dependencies:

(pID, cID) →

Sample Data:



	pid character(10)	cid character(10)
1	p11	c01
2	p12	c02
3	p13	c03

Principals

Purpose: This table stores the principal(s) of the school. This is useful in knowing who to grant permissions to in the database.

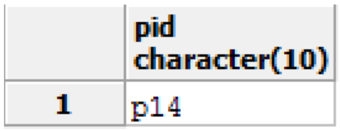
Create Statement:

```
CREATE TABLE Principals (
  pID CHAR(10) NOT NULL REFERENCES People,
  PRIMARY KEY (pID)
);
```

Functional Dependencies:

pID →

Sample Data:



	pid character(10)
1	p14

VicePrincipals

Purpose: This table stores the vice principal(s) of the school. This is useful in knowing who to grant permissions to in the database.

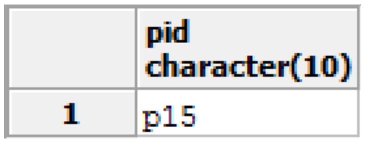
Create Statement:

```
CREATE TABLE VicePrincipals (
  pID CHAR(10) NOT NULL REFERENCES People,
  PRIMARY KEY (pID)
);
```

Functional Dependencies:

pID →

Sample Data:



	pid character(10)
1	p15

Secretaries

Purpose: This table stores the secretaries of the school. This is useful in knowing who to grant permissions to in the database.

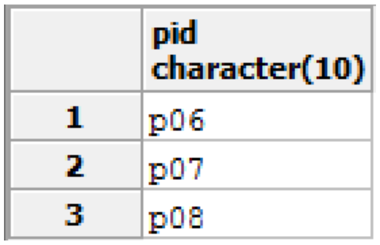
Create Statement:

```
CREATE TABLE Secretaries (
  pID char(10) NOT NULL REFERENCES People,
  PRIMARY KEY (pID)
);
```

Functional Dependencies:

pID →

Sample Data:



	pid character(10)
1	p06
2	p07
3	p08

Custodians

Purpose: This table stores the custodians of the school. This is useful in knowing who to grant permissions to in the database.

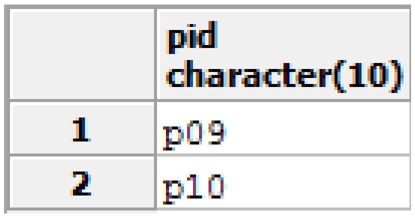
Create Statement:

```
CREATE TABLE Custodians (
  pID CHAR(10) NOT NULL REFERENCES People,
  PRIMARY KEY (pID)
);
```

Functional Dependencies:

pID →

Sample Data:



	pid character(10)
1	p09
2	p10

Classes

Purpose: This table stores the classes that are taught in the school by ID as well as the name of the class and the total number of students in each class.

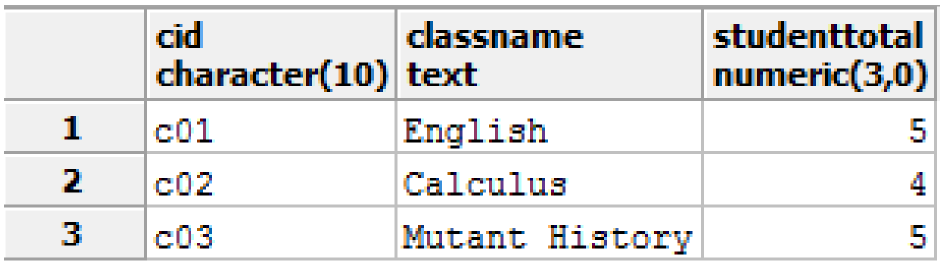
Create Statement:

```
CREATE TABLE Classes (
  cID CHAR(10) NOT NULL,
  className TEXT,
  studentTotal NUMERIC(3),
  PRIMARY KEY (cID)
);
```

Functional Dependencies:

cID → className, studentTotal

Sample Data:



	cid character(10)	classname text	studenttotal numeric(3,0)
1	c01	English	5
2	c02	Calculus	4
3	c03	Mutant History	5

StudentsRoster

Purpose: This table stores the IDs of the students with the ID of the class they are in. This helps keep in one place all the classes that all the students belong to.

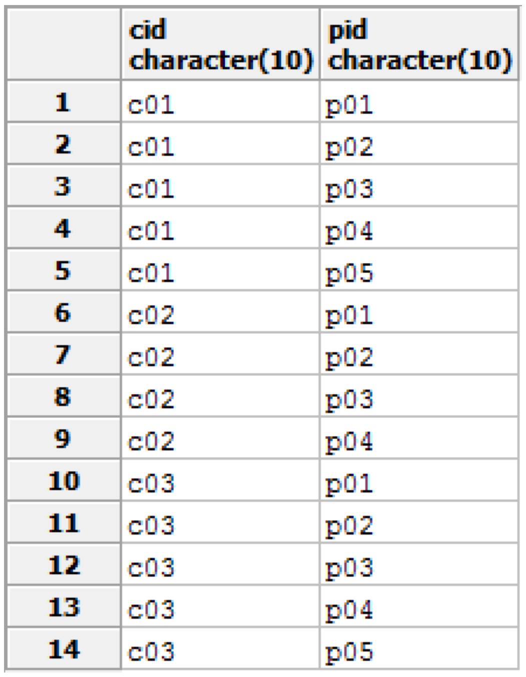
Create Statement:

```
CREATE TABLE StudentsRoster (
  cID CHAR(10) NOT NULL REFERENCES Classes,
  pID CHAR(10) NOT NULL REFERENCES People,
  grade CHAR(1),
  PRIMARY KEY (cID, pID)
);
```

Functional Dependencies:

(cID, pID) →

Sample Data:



	cid character(10)	pid character(10)
1	c01	p01
2	c01	p02
3	c01	p03
4	c01	p04
5	c01	p05
6	c02	p01
7	c02	p02
8	c02	p03
9	c02	p04
10	c03	p01
11	c03	p02
12	c03	p03
13	c03	p04
14	c03	p05

Salaries

Purpose: This table stores the monthly hours and total dollar sum of each person that earns a salary at the school.

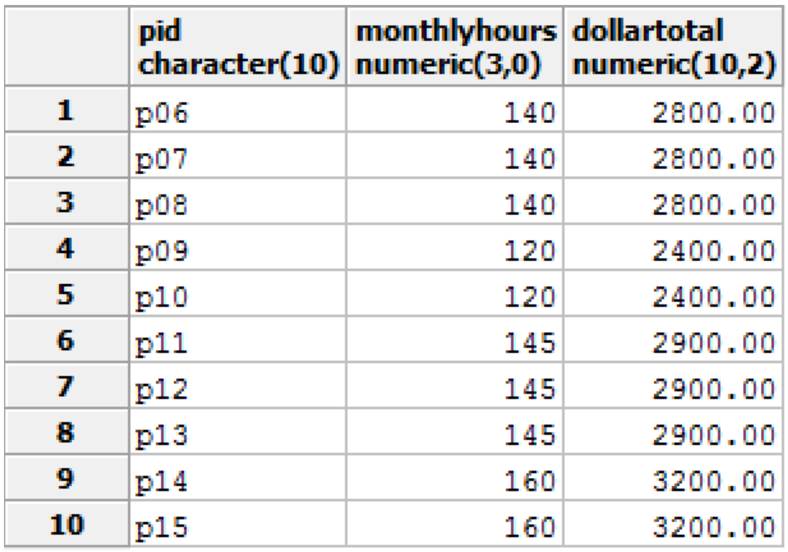
Create Statement:

```
CREATE TABLE Salaries (
  pID char(10) NOT NULL REFERENCES People,
  monthlyHours NUMERIC(3),
  dollarTotal NUMERIC (10,2),
  PRIMARY KEY (pID)
);
```

Functional Dependencies:

pID → monthlyHours, dollarTotal

Sample Data:



	pid character(10)	monthlyhours numeric(3,0)	dollartotal numeric(10,2)
1	p06	140	2800.00
2	p07	140	2800.00
3	p08	140	2800.00
4	p09	120	2400.00
5	p10	120	2400.00
6	p11	145	2900.00
7	p12	145	2900.00
8	p13	145	2900.00
9	p14	160	3200.00
10	p15	160	3200.00

Views

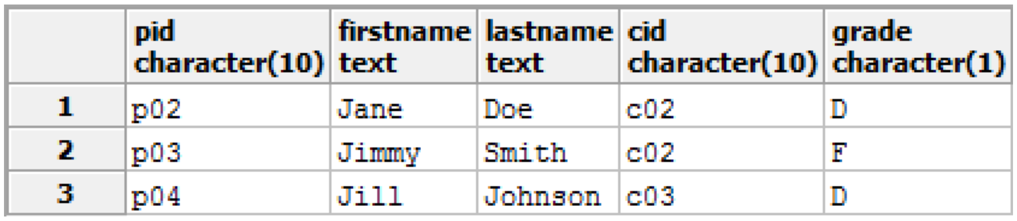
FailingStudents

Purpose: This view serves to highlight which students are failing any class so that they may receive any help they need to help them succeed.

Create Statement:

```
CREATE VIEW failingStudents AS
 SELECT s.pid,p.firstName,p.lastName,sr.cid,sr.grade
 FROM Students s
 LEFT OUTER JOIN studentsRoster sr ON s.pid = sr.pid
 RIGHT OUTER JOIN People p ON s.pid = p.pid
 WHERE sr.grade = 'D' OR sr.grade = 'F';
```

Sample:



	pid character(10)	firstname text	lastname text	cid character(10)	grade character(1)
1	p02	Jane	Doe	c02	D
2	p03	Jimmy	Smith	c02	F
3	p04	Jill	Johnson	c03	D

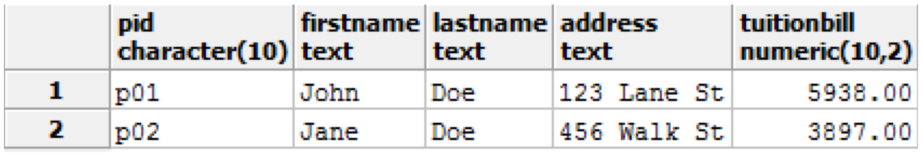
OutstandingDebt

Purpose: This view highlights all the students that have outstanding debts in their tuition bill over $3500. This is important because students with debt over $3500 are not allowed to take their finals and therefore will fail the courses they are enrolled in.

Create Statement:

```
CREATE VIEW outstandingDebt AS
  SELECT p.pid,p.firstName,p.lastName,p.address,s.tuitionBill
  FROM People p
  RIGHT OUTER JOIN Students s
  ON p.pid = s.pid
  WHERE s.tuitionBill > 3500
```

Sample:



	pid character(10)	firstname text	lastname text	address text	tuitionbill numeric(10,2)
1	p01	John	Doe	123 Lane St	5938.00
2	p02	Jane	Doe	456 Walk St	3897.00

Reports

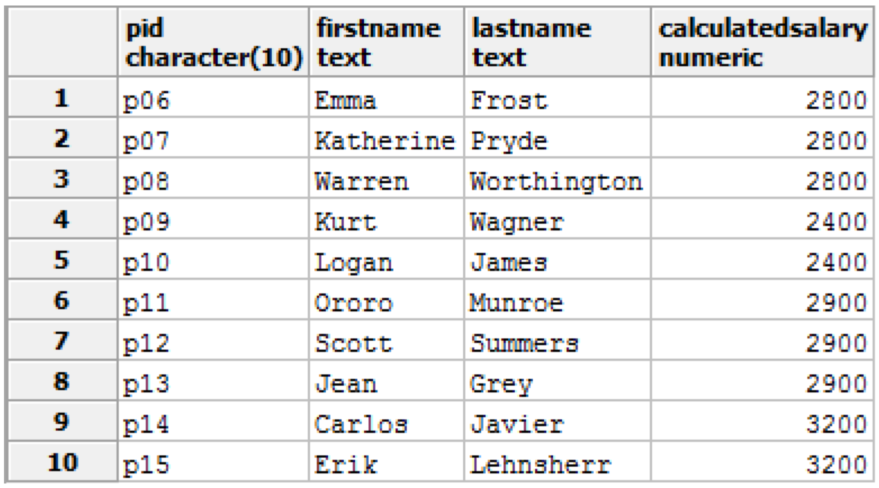
CalculatedSalary

Purpose: This table is similar to the Salaries table but it calculates the salary for each faculty member based on the input hours as opposed to relying solely on user input of the dollar total.

Query:

```
SELECT s.pid,p.firstName,p.lastName,(s.monthlyHours*20) AS
calculatedSalary
FROM Salaries s
LEFT OUTER JOIN People p on s.pid = p.pid;
```

Sample:



	pid character(10)	firstname text	lastname text	calculatedsalary numeric
1	p06	Emma	Frost	2800
2	p07	Katherine	Pryde	2800
3	p08	Warren	Worthington	2800
4	p09	Kurt	Wagner	2400
5	p10	Logan	James	2400
6	p11	Ororo	Munroe	2900
7	p12	Scott	Summers	2900
8	p13	Jean	Grey	2900
9	p14	Carlos	Javier	3200
10	p15	Erik	Lehnsherr	3200

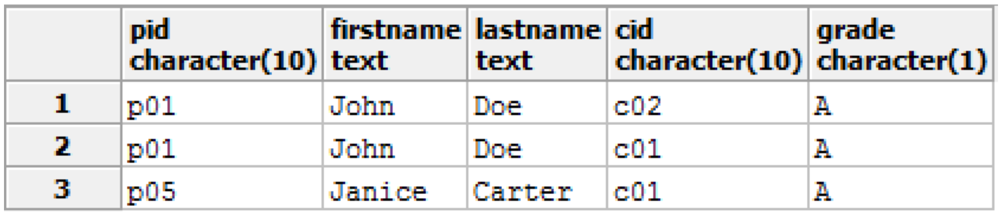
HonorRoll

Purpose: This table highlights students that are exceeding in classes by achieving a grade of “A” in the classes they are taking. By highlighting these students, an Honor Roll list can be complied.

Query:

```
SELECT p.pid, p.firstName, p.lastName, sr.cid, sr.grade
FROM studentsRoster sr
LEFT OUTER JOIN People p on p.pid = sr.pid
WHERE sr.grade = 'A';
```

Sample:



	pid character(10)	firstname text	lastname text	cid character(10)	grade character(1)
1	p01	John	Doe	c02	A
2	p01	John	Doe	c01	A
3	p05	Janice	Carter	c01	A

Stored Procedure

AddStudent

Purpose: This stored procedure is used to add a new student to the database. By using a stored procedure, one saves time in executing this common action.

Query:

```
CREATE OR REPLACE FUNCTION addStudent(CHAR(10), REFCURSOR)
RETURNS REFCURSOR AS
$$
DECLARE
        pID char(10) :=$1;
        resultset REFCURSOR :=$2;

BEGIN
        OPEN resultset for
                select pID
                from People p
                WHERE p.pID > 'p15';
        RETURN resultset;
END;
$$
language plpgsql;
```

Trigger

NewTuitionBill

Purpose: This trigger ensures that a new tuition bill field is added for each new student so as not to somehow skip this important field.

Query:

```
CREATE TRIGGER newTuitionBill
AFTER UPDATE ON Students
FOR EACH ROW EXECUTE PROCEDURE addStudent();
```

Security

Secretaries

The secretaries of the school need to enter information that has been submitted to them on paper, so they need access to certain tables. For example, they need to be able to manipulate the Students, Salaries, Teachers, and People tables.

```
CREATE ROLE secretaries;
GRANT SELECT, INSERT, UPDATE ON students TO secretaries;
GRANT SELECT, INSERT, UPDATE ON salaries TO secretaries;
GRANT SELECT, INSERT, UPDATE ON teachers TO secretaries;
GRANT SELECT, INSERT, UPDATE ON people TO secretaries;
```

Teachers

The teachers need to be able to manipulate students' grade.

```
CREATE ROLE teachers;
GRANT SELECT, INSERT, UPDATE ON people TO teachers;
```

Database Administrator

The database administrator has access to the entire database.

```
CREATE ROLE DBAdmin;
GRANT ALL PRIVILEGES ON ALL TABLES IN SCHEMA public TO DBAdmin;
```

Implementation Notes

As this database proposal is pretty basic, it is also pretty easy to implement. A pID for all the people in the school is required; after that, it is a matter of inserting each person into their corresponding table.

Known Problems

The stored procedure and trigger do not exactly work properly so they are currently not very useful in actually adding students to the database and creating a trigger for the tuition bill field. However, the trigger is not very important as it is not a good idea to use triggers in a database. Too many triggers constitutes a bad design.

Future Enhancements

The database could be expanded to include more complex functions such as a record of the guardians of the students and a more detailed record of expenses owed by the students as well as any scholarships they receive. It could also be expanded to include volunteers to the school and a log of visitors.