8/23/2016

Cisse, Ibra

University of Houston Downtown

Synchronizing School Systems

Table of Contents

[***Abstract*** 2](#_Toc460188618)

[***Introduction*** 3](#_Toc460188619)

[***Database Description*** 4](#_Toc460188620)

[***Database infrastructure and Data*** 5](#_Toc460188621)

[*Database System Design and Blueprint* 7](#_Toc460188622)

[***Database use case and prototype ER Diagram*** 12](#_Toc460188623)

[Database prototype Diagram. 14](#_Toc460188624)

# Abstract

School is the most important part of our life as human being. We dedicate ten to twenty years of our life to learn and become a master of a subject based on the subjects that we choose. It is known from your early years to high school, to all the way to your masters or PHD, teachers and mentors will record how you perform on each subject and give you a performance’s mark. These marks known as grade in the American system is a way that most company use in order to determine your status and if they can give you an offer to work or choose what you want to do in your life. By looking into the school systems in the United States of America, each district have several schools. Each school have several teachers labeled as faculty, staff, principal, and then you have the students which makes up the most population which is the way that the school system is organizing the hierarchy. However, they hierarchy is somewhat digitized but not organized enough. This application is going to focus on the hierarchy of school from the lowest as of students, to faculty members with access point as well as centralized network for the entire district. The entire district will share data for students, faculty will not be able to share data. However, the district manager can go any school and pull any data that is within his/her district. Teachers will be able to view students in their class, view the classes they are teaching as well as the student grades which can be adjusted by teachers as well. As for students, they can view their grades, their classes they are currently enrolled and also the name of each teacher that is teaching the class.

# Introduction

The school systems in America has a hierarchy they utilize to have student and teachers in an organized way. The system allows teachers to have a role sheet which allows them to determine who is absent, who is present, what grades they have, and the foremost, what is their average. Each teacher have certain classes they teach, and each class have a role sheet of students. Most teachers rely on papers to also determine student grades or use a program such as excel to calculate the grade point average. However, with the database project that I am building, it will facilitate the entire district which consist of multiple schools to share data as of students in the entire district, teachers to manage their classes, have roster sheet to mark student present or absent, to give grades which can be computed easily and assigned a letter grade. Teachers will be able to view how many students are in her class or classes as well as the number of absent they have in the entire school year. As for higher ups such as principal, they can view all of the information as of all teachers along with the classes they are teaching, all students along with their grades and courses being taken by each student.

# Database Description

The database will use the same hierarchy as the school system. Within the school system, you have teachers that teachers multiple classes within the day. They usually have multiple role sheets to keep up with every single student as of present or absent. They also have to keep up with the student grades as well as their average. The teachers can change the grades of each student depending on the class, they can adjust average according to their respective desire. The higher up on the list such as the principal can change the student schedule. The principal can view what each student have as of classes, grades, absents, and grade point average. The principal has the right to also include a many students from the principal or dean list which is a functionality that only he can do. The list will be available for all students and teachers to view. The principal will not be able to delete the entire database, he/she can only make changes. However, if time allows, this project can connect entire districts with the board of education which can view all students’ information and make any changes within the school districts. By using MySQL as database and its DMBS, several tables and information can be neatly organize and restructure to follow this hierarchy and concept.

# Database infrastructure and Data

**The basic information being held on this project as well as the basic brief description of its functionalities are as followed:**

**User Groups:**

**This user group will have to view mostly. The user group will consist of students who will be able to have a single signed in view. The view will allow all students to see the current classes they are taking, the grades that have been posted for that class. As for restriction access, they can only see their own grade, they cannot change anything to the database, and they are single sign in forms.**

**Teacher:**

**Teachers have the same access students which allowed them to view grades posted, student of their classes as well as update grades. They can also request student grade point average or update anything they wish to desire or delete. However, they are not able to see students that are not listed in their classes.**

**Principal/Dean:**

**They have access to all the student records, they also have access to all of the teacher’s records. Principals have the access can modify, change student records, they can change the student enrolments, and they can also make changes on the status on principal/dean list. They are also allow to request a view of another school system that will give them the record of teachers and student. However, they cannot change any data on the other school. They can only view.**

**District manager of school system: The district manager is the highest authority towards the database. They can build the entire database, they can erase it, modify the table settings as well as create new aspect on a table. They can view and change anything they would like which can either effect the principals only or the entire school district.**

**The data will be gathered as follow:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **First name** | **Last name** | **Username** | **Password** | **Date of birth** | **Security question** | **Answer** | **Access Type** |
| **Andrew** | **Hiett** | **Ahiet** | **Something** | **MM/DD/YYYY** | **Blahhh** | **meow** | **S** |

**Regular user Group:**

**Teachers:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| First name | Last name | Username | Password | Date of birth | Security questions | answer | Access Type |
| Gail | Stephens | Gstephens | Something | **MM/DD/YYYY** | something | someting | T |

**Principals:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| First name | Last name | Username | Password | Date of birth | Security question | answer | Access Type |
| Lin | Hong | Lhong | Something | MM/DD/YYYY | Something | something | P |

**District:**

|  |  |  |  |
| --- | --- | --- | --- |
| School Name | School Principal | School location | Access type |
| Some District | Linh Hong | Houston, DownTown | S |

**State:**

|  |  |  |  |
| --- | --- | --- | --- |
| State | City | District | School Name |
| Texas | Houston | Some District | Some School Name |

# Database System Design and Blueprint

There are multiple software that provides good solution and ease of access to organize data. However, the software I decided to use in order to accomplish this project with minimal effort is MySQL. MySQL is a solution provided by oracle that has multiple support through the community. It is supported by major companies that provide software solutions such as Microsoft with visual studio, Linux and their gnu compiler “GCC”, apple programing software with Objective C or swift, last but not least Sun Oracle with Java. Other languages such as PHP, HTML, JavaScript, JQuery, and Perl, have several good driver and ways to work with the drivers provided by oracle or third party software companies in order to smooth the writing of any application or system based application. In this project, using MySQL workbench will be the user interface for any of the user groups mentioned earlier. MySQL provides a general user interface for different users to have a view of their tables and its content. This will also allow user to see their query as well as their result. For example, users can see when their access is denied or granted based on their restriction from the database administrator or their restriction due to their membership of a specific table. The database will have several concepts such as primary key, foreign keys, and also members that will inherit rights from one table to another. This concept might be called indexing in order to prevent information being repeated which will cost the developers or its client performance.

As for database performing, a downtime will not require due to being hosted locally. It is not stored in the cloud as or any other persona since it is a school project. The other aspect of the database is that the languages that will be used to build and query information will be MySQL workbench for user interface or there might be a possibility to use C# or a webserver to show a different user interface with views, access restriction, usernames and other aspect of the capabilities of the database. Information will be queried from multiple tables depending on how complex the user request is. Based on their request, clients/users will be able to perform certain duties or operations:

1. **Students**
   1. Log into database
   2. Log out from the database
   3. Request the list of classes
   4. View every class with their proper assignment with grades
   5. See their average for the class
2. **Teachers**
   1. Login into the database
   2. Logout from the database
   3. Request the classes they are teaching
   4. View the list of students enrolled into the professor’s class
   5. Assigned/replace/delete grades for students
   6. Able to assign attendance to each student based on their role sheets.
   7. GPA will be there and automatically calculated by the functionalities of the database.
3. **Principals**
   1. Principals can view the list of all teachers and students.
   2. View all classes taught by teachers along with the students enrolled in for that teacher.
   3. View student’s absences grades along with GPA.
   4. View students that have the highest GPA and are consider to be on the dean’s list.
   5. Add/remove/delete students from classes as well as the database.
   6. Add/remove/delete teachers on certain aspect such as classes or school if the teacher is no longer teaching.
   7. View the teachers’ personal information such as address and others.
   8. Principals can view other schools information such as address, distinguish student, as well as the information of the other principal. Hence, only basic information as of name, last name, and basic bio. None of the following will be display, address, username, or any other sort of information deemed to be private.
4. **District**
   1. District will be able to hold and restricted access all the schools located around its area.
   2. They can have the same access as the principal. Except, they cannot access detailed information principals.
   3. They will not have access to students’ personal information such as address, username, password or other content deemed too personal.
5. **State:**
   1. State holds all the districts of that state. For example, Houston, San Antonio, or Austin being in Texas.
   2. State will not see any user information such as students’ private information such as student address, username, and password. However, they can see the students enrolled in schools as well as their GPA.

The following table will show the Database will be modeled as followed:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **First name** | **Last name** | **Username** | **Password** | **Date of birth** | **Security question** | **Answer** | **Access Type** |
| **-1000** | **Andrew** | **Hiett** | **Ahiet** | **Something** | **MM/DD/YYYY** | **Blahhh** | **meow** | **S** |

**Student Table**:

**Teacher**:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Identification | First name | Last name | Username | Password | Date of birth | Security questions | answer | Access Type |
| -10000 | Gail | Stephens | Gstephens | Something | **MM/DD/YYYY** | something | someting | T |

**Principals:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ID | First name | Last name | Username | Password | Date of birth | Security question | answer | Access Type |
| 0 | Lin | Hong | Lhong | Something | MM/DD/YYYY | Something | something | P |

|  |  |  |  |
| --- | --- | --- | --- |
| School Name | School Principal | School location | Access type |
| Some District | Linh Hong | Houston | S |

**District:**

**State:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Identification | State | City | District | School Name |
| -1000 | Texas | Houston | Some District | Some School Name |

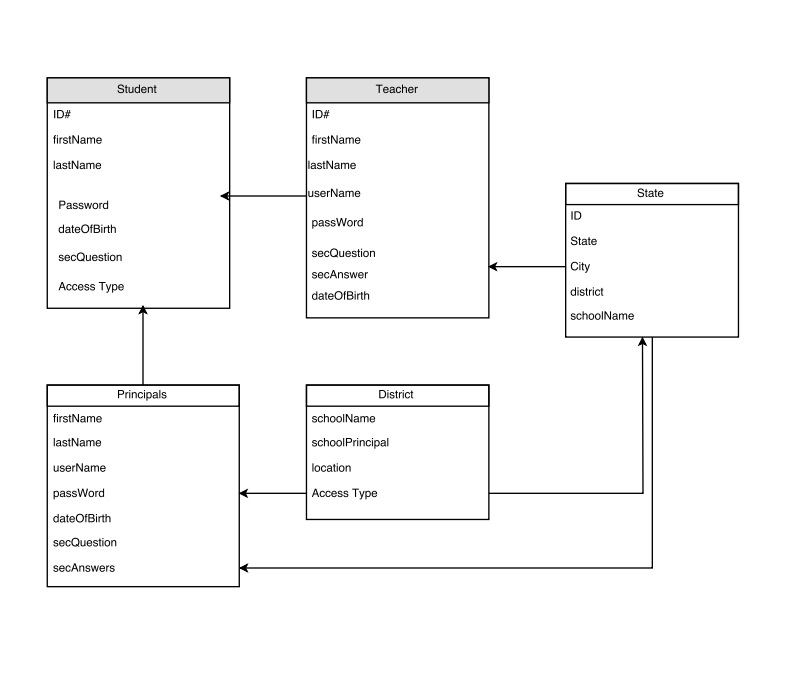
Certain aspect of the table such as identification will be automatically generated as the database is appending data to the table. The concept of the primary key, will be use through the entire database in order to link multiple tables with keys and ease access of information through the project.

# Database prototype use cases and prototype ER Diagram

The following diagram and simple use cases will provide basic use case and scenarios as for clients representing teachers, students, and principals, against the database management system:

1. Simple Use case Scenario between clients and DBMS
2. Client request to login
   1. Database asks for username and password
   2. Database checks for valid credentials
   3. Return with either access grand with a view or access denied with login screen
3. Client request to logout
   1. Client clicks the logout button
   2. If client is already login, database ends its session for that client
   3. Return to the login screen or if client was not logged in, returns to the log in screen.
4. Student request to view class enrolment:
   1. Student request to view class requirement.
   2. Database queries the request from the table list of classes
   3. Return the list if student is permitted, else, returns error message such as access denied.
5. Principal request to see GPA above 3.0
   1. Request to see all GPA higher than 3.0
   2. Database query by selecting all students with GPA higher than 3.0
   3. Returns the list of students with GPA higher than 3.0 along with the school they are enlisted as students.
6. Principal request to view another school’s Deans list.
   1. Request with the right credentials
   2. Database query the request by selecting the specific school along with the list of students with a specific GPA and display it to the user currently logged in.
7. Teacher changing the grade of a specific student.
   1. Teacher request specific class to query
   2. Database process the request and return the class along with students.
   3. Teacher select a student
   4. Teacher decided to change a student grade
      1. Database select specific from the class
      2. Database selects the specific student along with his record
      3. Then database updates the specific grade point that teacher requested
      4. Return from the student class
   5. Return from the teachers database
   6. Return the query information to the user
   7. End query.

# Database prototype Diagram.



# Database Modification

Analyzing the concept of the database structure for this project has a lot of repeats on several tables. Information such as using Identification, first name, last name, address, security question and answer are repeated through multiple field. This is one of the data redundancy mentioned earlier in database concept where we can separate the data and index that table in order to avoid repeating data. The new table will look as followed:

**Member Information**:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **fName** | **lName** | **username** | **password** | **secQuestion** | **secAnswer** | **role** |
| -1000 | Ibra | Cisse | Icisse | Something | someQuestion | someAnswer | s,p,t |

By having this table, all student and principal can be clustered with all the users in order to solve the repetitive data for the student, principal and teacher and synchronized it into one. The other aspect that will improve the database is by creating additional table such as teachers which will be linked with Member information with access type. With each access type, if they are teachers, they can have a different table that have the list of their classes and students. The information will be as followed:

**Student:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Role** | **Class** | **grades** | **Attendance** | **GPA** |
| -1000 | table | Table | table | Table | table |

By using the Id and access type as a primary key, those two fields can be used an entity to linked the other table with. This will make it easy for access to the other table access as well as to determine their roles for each member. From there, we can also optimize the principal table as well as well teacher’s table. The table will look as followed:

**Teacher:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Role** | **Classes** | **Grades** | **attendance** |
| -1000 | field | Table | Table | 0,1 |

**Principals:**

|  |  |  |
| --- | --- | --- |
| **ID** | **role** | **school** |
| -1000 | field | Table |

**Classes:**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** (class ID) | **Subject**(class ID) | **members** | **role** |
| -1000 | table | table | Table |

**Subject:**

|  |  |  |  |
| --- | --- | --- | --- |
| **classId** | **classSubject** | **members** | **role** |
| -1000 | Physics | Some student | s |

**Grades:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Subject** | **Test1** | **Test2** | **Test3** | **GPA** |
| -10000 | Physics | Int | Int | Int | (t1,t2,t3)/3 |

**HonorRollTable:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **student** | **school** | **District** | **subject** | **GPA** |
| -10000 | Table | Table | Table | Table | table |

**School**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **schoolID (different from user and other Tables)** | **address** | **zipcode** | **county** | **District** |
| -10000 | Some address | 77086 | Harris | Houston |

**District**:

|  |  |  |
| --- | --- | --- |
| **school** | **Principal** | **horrorRoll** |
| table | table | table |

**State:**

|  |  |  |
| --- | --- | --- |
| **School** | **State** | **District** |
| table | table | Table |

The following entities of all the following tables will accept only the specific inputs and which builds the new ER diagram as:

|  |  |
| --- | --- |
| **Entity** | **Input type** |
| ID | Integer |
| fName | Varchar “string” |
| lName | Varchar “string” |
| username | Varchar “string” |
| password | Varchar “string” |
| secQuestion | Varchar “string” |
| secAnswer | Varchar “string” |
| schoolID | Integer |

# Relational Database

The relationship database is something that is require in order to make sure information is not repeated over and over. By making sure that the database relationship is appropriate and well taken care of, we use several aspect such as many to many relationship or one to many relationship. One to many relationship is when a child replicate some of the parent entities and keep it for itself. Remember that the parent only have one child on this aspect and will re repeat it. For this example, we use multiple tables that does not need to be replicated as many time as the other such as the class table, teacher table, student table, and subject table as one to many relationship:

**District table**: District table has one to many relationship with its parent State. The state is what host all the district from the city in order to map schools with different principals and such.

**School table**: School table has one to many relationship with district. The school table will be linked to its parent which is the district. The school will have a unique ID for each school which will later on be used for each school system.

**Teacher Table**: The teacher table has one to many relationship with the member table. The teacher table will inherit memberID, role. From there, it will also inherit from the grade table as well as the attendance table in order to control access and altering the tables.

**Subject** **Table**: the subject table has a one to many relationship with class table. The subject table will inherit the classId, the classSubject, and also the member that is teaching it along with who is taking this subject.

**State**: state table has one to one relationship with other tables such as district, school, honored and principal table. These table will be members of the state which will be consider as one to many.

The other tables will have many to many relationships which copies multiple entities from other tables and co-relate it with either primary keys or foreign keys depends on how sensitive the information is. The following tables have many to many relationship with one another:

**Student Table**: The student table will inherit from member table by having the following entities: memberID, and role. From the class table, it will inherit the different subjects as well as their grades, and average.

**Class table**: The class table will have many to many relationship with several other table such as members in order to get certain entities, and also from the class subject in order to see grades and such. They will also contain the students are student and teacher in order to make different table for classes being though by teachers and table to hold all student grades.

**Principal**: The principal table has many to many relationship with members, school table, teachers, and students. This will only allow them to view the students of classes along with classes that are being thought. The principal table will be able view the subjects which is also another many to many relationship.

**HonorRoll**: Honor table has many to many relationship with students as well as principal, school, district, as well as state. The higher up table such as state and district can see the honor roll sheet for students.

With this summary of relationships, we look into scenarios for different type of users which brings different type of use case as well as their successful and failed queries.

# Database Scenarios and use cases