8/23/2016

Cisse, Ibra

University of Houston Downtown

Synchronizing School Systems

Table of Contents

[*Abstract 2*](#_Toc468648687)

[*Introduction 3*](#_Toc468648688)

[*Database Description 4*](#_Toc468648689)

[*Database infrastructure and type of entities 5*](#_Toc468648690)

[*Database System Design and Blueprint “Prototype” 6*](#_Toc468648691)

[*Relational Database and ER diagram description 9*](#_Toc468648692)

[*Database use cases 12*](#_Toc468648693)

[*Relational Model to Database management Language 18*](#_Toc468648694)

[*Test Plan and Test management 22*](#_Toc468648695)

[*Time Table and Completion 23*](#_Toc468648696)

[*Conclusion 25*](#_Toc468648697)

[*References 26*](#_Toc468648698)

# 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 type of entities

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

**Students:**

**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.**

# Database System Design and Blueprint “Prototype”

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.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **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 |

# Relational Database and ER diagram description

The relation of the entities of the database will be build based on the access that each entity needs in order to access or perform a task. For example, students taking multiple courses with different teachers. There are also multiple teachers teaching same courses.

1. Members and students: Each member has the possibility to a student. Each student is a member already. Therefore, the relation will be a many to many. The correct term will be many members are either a student or not.
2. Members and principals: Each member can be a principal. In this scenario, we are using only one principal from the member table. Therefore, the relationship is one to one. One member is a principal.
3. Member and teachers: There are multiple members that will be considered as teachers. There will be many teachers that are already members. Therefore, the relationship is many to many.
4. Classes and teachers: There are multiple classes that a teacher can teach. There is also one teacher for every class. Therefore, the relationship between classes and teachers is many to one. Also there is a possibility of multiple teachers teaching the same subjects.
5. Classes and grades: There is grades for every single class. There is only one grade course for each class. Therefore, the grades and classes table relationship is one to one.
6. Classes and students: one class can contain multiple students. Each student can have multiple classes. The relationship becomes many to many.
7. Students and teachers: There are one to multiple teachers to a student. There are also multiple students to a teacher. Their relationship becomes many to many due to the fact that both entities might have multiple of the other.
8. Foreign keys: all the foreign keys will be used a method to gain access to other tables respectively and gain other information. Another aspect of the foreign keys can be used to form new tables as they will be the new relationship tables in order to make relationship easier to manage. For example, each class will trigger a subject table which will have the teacher as primary key, then the subject name, and all of the students enrolled in that with their primary key from the student table as foreign on this new table.
9. The foreign Keys: The foreign keys exist within the database in order to facilitate the information access on the other table. For example, rather than getting all of the information from one table to another by doing multiple IDs search and such, DBA can perform a query that will get all the students from a class and generate their names and last name by using the foreign key of the students which is the primary key on the student table. This will facilitate the information flow from table to table, data can be link also via primary key such as memebersID being foreign Key to StudentID to make sure that they are member and which role they perform on the database. Example of the following will be given towards the use cases as student is an actor of the dataset as well as the principals, teachers, and forth.
10. 

# Database use cases

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. Member 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. Member 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. Student request to enroll in a class
   1. Student log in
   2. Select the class he wants to enroll
   3. Submit the request
   4. Student is in class
   5. End of request
6. Teacher uploading grades for student
   1. Teacher logs in
   2. Select a specific class from view
   3. Teacher sees number of student
   4. Teacher select the specific student
   5. Teacher upload the grade for the exam
   6. Average is computed
   7. Save to the database
   8. End of request
7. 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.
8. 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.
9. Principal request to view all the teachers that are teaching specific material
   1. Principal logs in
   2. Principal request the list of all classes
   3. Selects a specific subjects
   4. And request to view all the teachers teaching that subject
   5. List of teachers are display as a view
   6. End the query request
10. Principal adding a student to a class
    1. Teacher logs in
    2. Request a specific class
    3. View the class list of enrolled student
    4. Add a new tuple for the student
    5. Student now is enrolled in this course
    6. End request
11. Student request to view grades
    1. Logs in
    2. Select a course that student is enrolled in
    3. Request his grade from the course
    4. Grade and retrieve and display as view
    5. End the query
12. Principal view the list of all teachers and students
    1. Log in as principal
    2. Select a subject
    3. View the list of teachers
    4. Request the list of students
    5. List of teachers and student are united
    6. Display the information as view
    7. End request
13. Student request the list of student enrolled in school
    1. Student log in
    2. Request the list of students
    3. Credentials will be checked
    4. And student is denied due to database restriction
14. Principals want to know how many students are currently taking physics one in the entire school
    1. Principal logs in
    2. View is display
    3. Request to view the list student taking physics
    4. Database performs the query of physics class,
       1. For each class of physics for that semester, do sum of the students.
       2. Join all the sums of the students
    5. Display the join sum of students
    6. End Query
15. Displaying the total number of teachers in the school
    1. Principal logs in
    2. View of the principal is displayed
    3. Request the total number of teachers in the school
       1. Database start the request by using foreign key of principal
       2. Uses the key to do a Count for the entire rows
       3. Then return the answer
    4. Result is return to the view
    5. Query is ended
16. Principal request the list of all student taking mathematics and physics which have grades c or better
    1. Principal logs in the database
    2. Principal view is display
    3. Principal request the view the list of student from physics classes
       1. For each physics class
       2. Select the name, last name, and grade of the student higher than c
       3. Return query
    4. Select all the students taking mathematical courses
       1. For each math course
       2. Select the first name, last name, and grade of all students higher than c
    5. Perform a join set of these two query by joining both result so information will not be repeated
    6. Display the result to the principal
    7. End the query

# Relational Model to Database management Language

Most users and actors has been defined earlier which remains the same. They are teachers, students, principals. With the increase of this project, it might go to other type of users such as parents of students which can only view their children’s grades. As for how to make this database possible, MySQL workbench provides a very good user interface for checking basic design and demonstration. With the start of the port 3306 which is dedicated to SQL, workbench 6E is able to allow the user to create database. From the database, tables which can be defined with different attributes and linked as the diagram shows. On this specific project, I started with defining the Database which is called “schoolSystem”. From the School System database, another command is ran to check just in case the SQL script is exported to another database. This is done by adding a simple if state to make sure if it does not exist, it will be created and starting to use. As for how the database is populated, first, we needed tables such as Members, Teachers, students, classes, Subjects, Grades, and Principals. For each table, their primary keys are defined along with their relationship with other table and how operations will be performed in order to support use cases and correct design if there is any flaw to it. As for populating the database, for testing purposes, information will be gathered based on the internet and filled in the tables. These information will save in a file where certain SQL statement will be written which will save the information to the proper table. When each table have enough data to perform certain test, aggregation and join operations will be performed and shown as example. The following SQL code will show how tables are being creating along with how their relationship is modified as new information is added:

1. Adding a new class:
   1. Principal can add a new class by performing a query that insert class with the proper subject name, subject id is automatically generated, the teacher teaching the course by adding its teacher id. From there, a trigger will start by automatically creating a new table for the subject, with blank student ID and primary ID as the teacher’s ID. This is the reason why class and subject has relationship along with class, teacher and subject.
2. Deleting a class:
   1. If a principal decide to delete a class, an entire table will be deleted by using drop table with subject name.
   2. By using the drop table, all of the information within the subject table such as teacher and student IDs as well as their grades will be deleted.
   3. After the table is drop, the delete class will be performed which will get rid of one row from the class table.

This is the reason why foreign keys are used in the class table, with teachers table, as well as the grade table to the principal table.

1. Adding a new Student:
   1. Student logs in and view is displayed.
   2. Student request to view list of subject being offered.
   3. Then database queries and brings the course information along with teachers names.
   4. Student chooses to enrolled in the course
   5. Database adds their name to the list of student taking that course
   6. A new row is added with the student grades as well as the number of absents being recorded as null and average as the average of all grades.
   7. Database ends query
2. Deleting new Student:
   1. Teacher logs in and view the list of classes he is teaching
   2. Choose a specific course the teacher is teaching
   3. From there, view the list of students and the number of absences
   4. Decide to erase a student from the role
   5. Database queries by eliminating the student from the class
   6. This triggers the delete function to delete the student record on grades, on the class table with by affecting a row.
3. Principal adding new teachers and classes:
   1. Principal logs in and view is display
   2. Principals request to add a new class
   3. Class subject is determine and added to the classes
   4. Class subject is assigned to a new teach or old teacher
   5. Subjects generates a new table with open student IDs as intake and absences, as well as grades.
4. Teaching filling the students’ grades and GPA being updated:
   1. Teacher logs in and view of his classes are showed
   2. Teacher clicks on a desired course.
   3. Courses shows all student currently enrolled on that course
   4. Along with a GPA of each student by row.
   5. Teach decided to click on specific student
   6. Student record is shown which includes test grades, GPA and absences.
   7. Teacher changes on row by input of a test grade either bigger or lower than its original
   8. GPA changed is triggered and it changes based on the grade input
   9. Update is triggered to save information on the related table
   10. Then teachers goes back to classes.
   11. End the query

# Test Plan and Test management

The test Plan goes according to the simplest to the hardest. It starts as actors as Students to the highest of all actors which is the principal. The tests are as followed:

1. Student Login:
2. Student logout
3. Student signup for class
4. Student view classes
5. Student view teachers
6. Students view average
7. Teachers logins
8. Teachers logs out
9. Teachers view classes they are teaching
10. Teachers add student to a class
11. Teachers remove student from a class
12. Teacher upload grades for a student
13. Teacher change student grades
14. Teachers check for the students with highest grade:
15. Teachers check with students lowest grade:
16. Teachers brings the list of all of his student from every class
17. Teachers brings the list of all students with a grade higher than 80 from all of his classes
18. Teacher shows all failing students from each classes

# Time Table and Completion

|  |  |  |
| --- | --- | --- |
| **Task** | **Status** | **Dates** |
| Documenting abstract and introduction for the database | Completed by Ibra Cisse | 7/27/2016 |
| Documenting database prototype and which language along with | Completed by Ibra Cisse | 07/28/2016 |
| Documenting and re-implementing requirement for database functionalities and ER Diagram | Completed by Ibra Cisse | 08/16/2016 |
| Designing and documenting use cases for the different actors and roles | Completed by Ibra Cisse | 10/20/2016 |
| Implementation of database prototype such as creating blank database with several blank tables. | Completed by Ibra Cisse | 10/28/2016 |
| Implementing relationship from diagram to tables and testing its functionalities by adding information to tables and view queries | Completed by Ibra Cisse | 11/05/2016 |
| Implementing student complete relationship with every table along with creating new entities to support relationships for every table | Completed by Ibra Cisse | 11/10/206 |
| Importing information to have a fully functional database | Completed by Ibra Cisse | 11/06/2016 |
| Testing phase 1: “student functionalities” | Completed by Ibra Cisse | 11/12/2016 |
| Testing phase2: “teacher and class attribute testing” | Completed by Ibra Cisse | 11/13/2016 |
| Testing phase 3: “Principal features and functionalities testing” | Completed by Ibra Cisse | 11/14/2016 |
| Testing phase 4: “Integration testing using path way to make sure all features and functionalities works as mentioned in the documentation | Completed by Ibra Cisse | From 11/15/2016 to 11/23/2016 |

# Conclusion

In the end, the project I have decided to work was out of proportion which I had to size in order to match the time and presentation. At this moment, the classes table could have been separated a new database and also could have been index with the school system database which would of improve performance and link the two databases. However, this is not the case. Please refer, to the demo file for procedure and such.

# References

1. <http://www.mysqltutorial.org/managing-sql-views.aspx>
2. <http://www.tutorialspoint.com/codingground.htm>
3. <http://www.cs.jhu.edu/~yarowsky/600.415.stored-procedures/mysql-storedprocedures.pdf>