Database final project

Natnael Kidane
naki21@student.bth.se
28-05-2024

Program description

The program I want to write is for school. The idea is that for many parents it is hard to go to their children's schools and monitor their development and progress. This can happen when both parents work full time or if the parents travel because of work for many hours.

The program will fix this problem by making it possible for the parent to check the progress of their children from home or from work. This will be possible when the teachers register the students points and completed subjects using the same program, and the parents will see the change from distance by logging in to the program.

Features

The program offers a login account first that will be given to both parents and teachers. And when they log in the teachers will be able to register grades of students on their subject and also be able to see already registered grades.

And on the other hand parents will be offered options to see what grades their children have in all subjects or in a specific subject.

Logical model

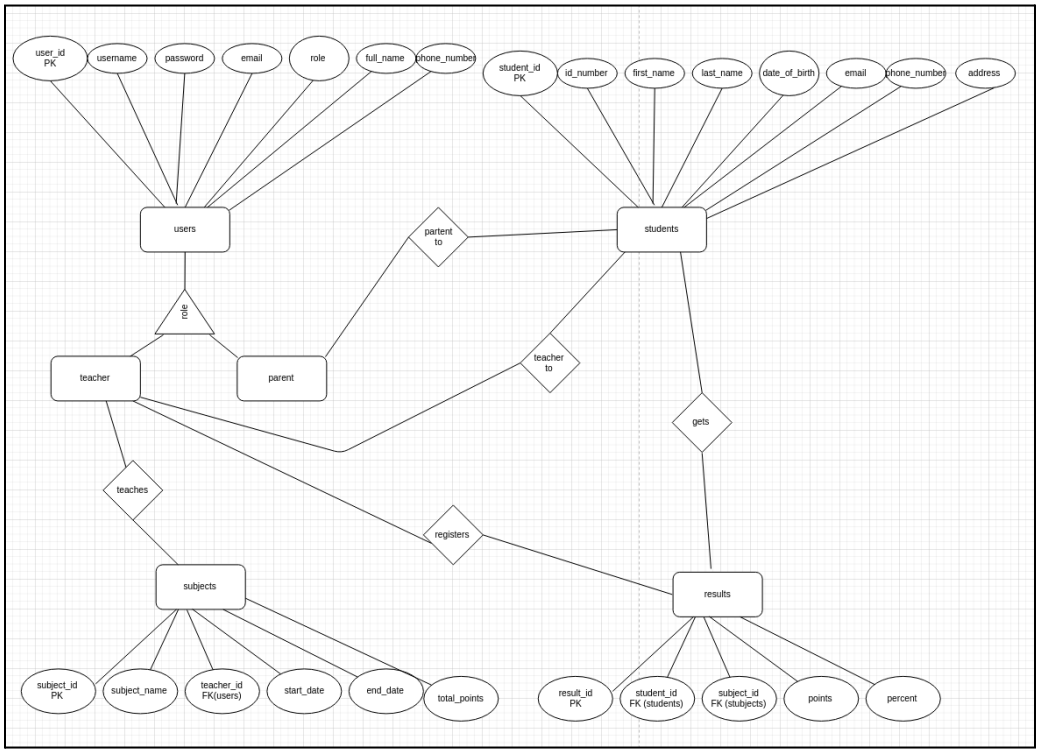

Diagram description

As the diagram shows there are 4 entities:

- Users:
    - Teachers
    - Parents
- Students
- Subjects
- Results

Users table

The users table will store the teachers accounts. They have an attribute “role”. Role describes their role as teachers and parents. Role will help the program identify the user and give them options what they can do when they are logged in accordingly.

- Teachers will register points in the results table.
- Parents will read points of students from the results table.

Students table

This table will store the students and it will have a relation to the results table because the results of the students will be registered there. This result will be identified because there will be a foreign key in the results table that connects students table to it.

Subjects table

This table contains the subjects that teachers teach and the students studies. It also have relation to the results table because the results are which student to which subject. This will be needed to connect the students with the result on a specific subject.

Results table

This table contains the results of students on a specific subject. This table connects all the other entities mentioned above because:

- Teachers inserts results
- Parents reads results
- Students and subjects are connected to it with foreign key because it has to specify which student and subject belongs to which point.

SQL queries

There are different sql queries in the program to make all the above functionalities to work, because the data and accounts are stored in an sql database. The queries are as follows:

Read results from database

This is done in a python function called fetch_results(). This query uses INNER JOIN to connect results and subjects and then it gets the target student using subquery (select in select). By doing this it gets the grades of students in different subjects registered in the database.

Insert result to database

This query is both update and insert queries optionally found in one python function. Both the insert and update use subqueries (select inside insert and select inside update) to insert or update results of students in the results table. Only users with a teacher's role can do this. Insert is used if there is no entry in the table with the same student and subject combination because it will first time registration. And update is used if old data is found in the database with the same student and subject.

Create trigger

In the results table there is column named percent. It records the result in percent by calculating grade the student have divided by the total possible grade in the subject. This is done using a trigger updates this entry when ever an insert or update is done in any row in the results table. In that table there are two triggers called "*insert_percent*" and "*insert_percent*" that are triggered when insert or update happens in the table.

Create function

In the database there is a function that calculates total average of a student from all the registered grades by using data from percent column. This calculated total average will be used by teachers to sort the students according by their average (highest to lowest grades scored).

Aggregate functions

AVG()

The function total_avgerage in the database takes all entries of a student from the percent column and uses AVG() function to calculate the total average.

COUNT()

There is a python function called "*total_ave*" in the program that uses the database function "*total_avgerage*". But to know how many subjects this total average is based on it first uses a select query with function COUNT() to calculate how many subjects are graded for that student in the results database. The idea is that there could be subjects that are not graded for that student yet.

Implementation

The program is implemented as a command line program that runs in the terminal. Before using the functionalities mentioned above it offers a login page first to identify the user and their role. And it gives options to choose according to their role recorded in the database.

Github link

The program can be found in the github link below:
https://github.com/natnaelbth21/database_final_project