Presentation + characteristics :

- prezence list application

- used in universities (most part)

- used to mark the prezence of a student

- teacher creates account :

- selects the subjects he teaches

- compleats the from (id, first name, last name, password, email, degree)

- recives a confirmation mail to activate the account

- student creates account :

- compleats the from (id, first name, last name, password, emai, year, group)

- recives a confirmation mail to activate the account

- the teacher connects first

- the app generates a code that only the teacher has access to

- the students connect

- students will use this code in order to mark their prezence

- the code will expire in a short time (10-15 minutes)

Functions :

* mark\_prezence(id\_student,id\_course,prezences)

(marks the students prezence at that class)

* class\_precentage\_prezence(prezences)

(shows in real time the average number of students that are at that class using by percentage)

* most\_absences(id\_student, prezences, id\_course)

(shows in real time the students with the most absence)

* most\_prezences(id\_student,prezences,id\_course)

(shows in real time the students with the most prezence)

* students\_promoting(prezences, min\_prezences)

(shows in real time the number of prezences the student needs to pass)

* student\_precentage\_prezence(id\_course,id\_student)

(shows in real time the prezence situation for each student using percentage)

* delete\_prezence(id\_student,prezences,code)

(delete a prezence)

* add\_prezence(id\_course,id\_student,prezences)

(add a prezence - only used by the teacher)

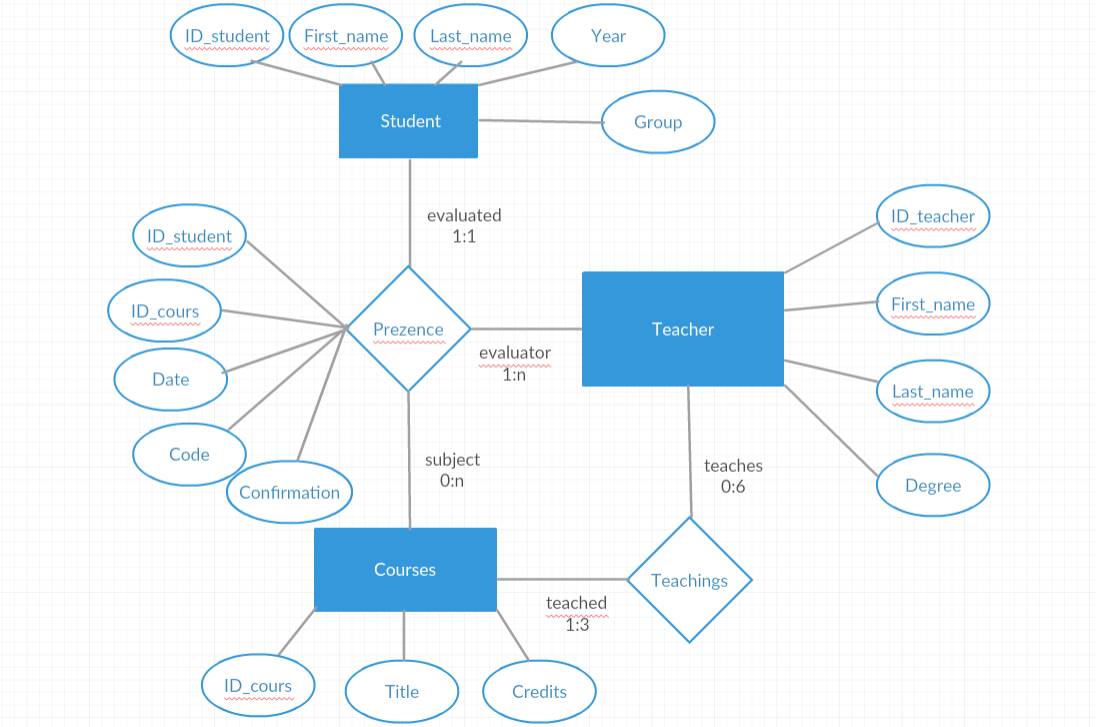
* delete\_all\_prezences(code,prezences)

(delete all prezences)

* generate\_code(id\_course)

(generate a numeric code for the students)

Data base schema :



Data stuctures :

* we use indexes to improve the performance of selectin queries, if the SELECT query uses an indexed column in any of the WHERE conditions, the query uses the index segment instead of performing a full table scan. This results in enhanced query performance
* we use hashes because we want to select only specific clusters from the table

Demonstration for 4NF :

Student = (ID\_student, First\_name, Last\_name, Year, Group)

R1 = ( A B C D E )

SIGMA = ( A -> BCDE, ABC -> DE, BC -> ADE )

L M R

----------------------------------------

- ABC DE

A, BC, ABC are our keys (actually they are all superkeys). And because they are all superkeys => our schema is in 4NF. Because they are in 4NF => that they are also in 3NF and 2NF.

-----------------------------------------------------------------------------------------------------------------

Course = (ID\_course, Title, Credits)

R2 = ( X Y Z )

SIGMA = ( X -> YZ, Y ->XZ )

L M R

--------------------------------------

- XY Z

X and Y are our keys (actually they are all superkeys). And because they are all superkeys => our schema is in 4NF. Because they are in 4NF => that they are also in 3NF and 2NF.

------------------------------------------------------------------------------------------------

Teacher = (ID\_teacher, First\_name, Last\_name, Degree)

R3 = ( Q W E R )

SIGMA = ( Q -> WER, WE -> QR )

L M R

--------------------------------------

- QWE R

Q and WE are our keys (actually they are all superkeys). And because they are all superkeys => our schema is in 4NF. Because they are in 4NF => that they are also in 3NF and 2NF.