Basics of database systems

# Project – School Database

Lappeenranta-Lahti University of Technology LUT
Software Engineering

Basics of database systems
Spring 2024

Tomi Vilpponen & Aleksi Natunen

## Table of contents

# 1 DEFINITION

**School database**

In this project ‘School database,’ the database is developed for a school who manages its students, professors and their courses. Within the database, the school's information is stored as well as the students, professors and courses inside it.

It is important to be able to find the courses that the school provides. Also it is important to be able to see which courses which student is attending in addition to which course which professor is currently teaching.

The following database queries have to be implemented to the database: (1) List a certain field's students. (2) List a certain student's courses. (3) List a certain course’s information. (4) List courses that a professor teaches. (5) List a certain field’s events.

# 2 MODELING

## 2.1 Conceptual model

In Figure 1 is the ER model of the designed database. The ER model contains 6 different entities and 7 different relationships. 5 Of the entities containing their own primary key, that functions as an ID to help the identification and to eliminate copies. One of the entities is also a weak entity for it cannot live without its strong entity "parent". This weak entity in the ER model is the Events entity. There is in addition to all of this one N:M relationship between Course and Student.

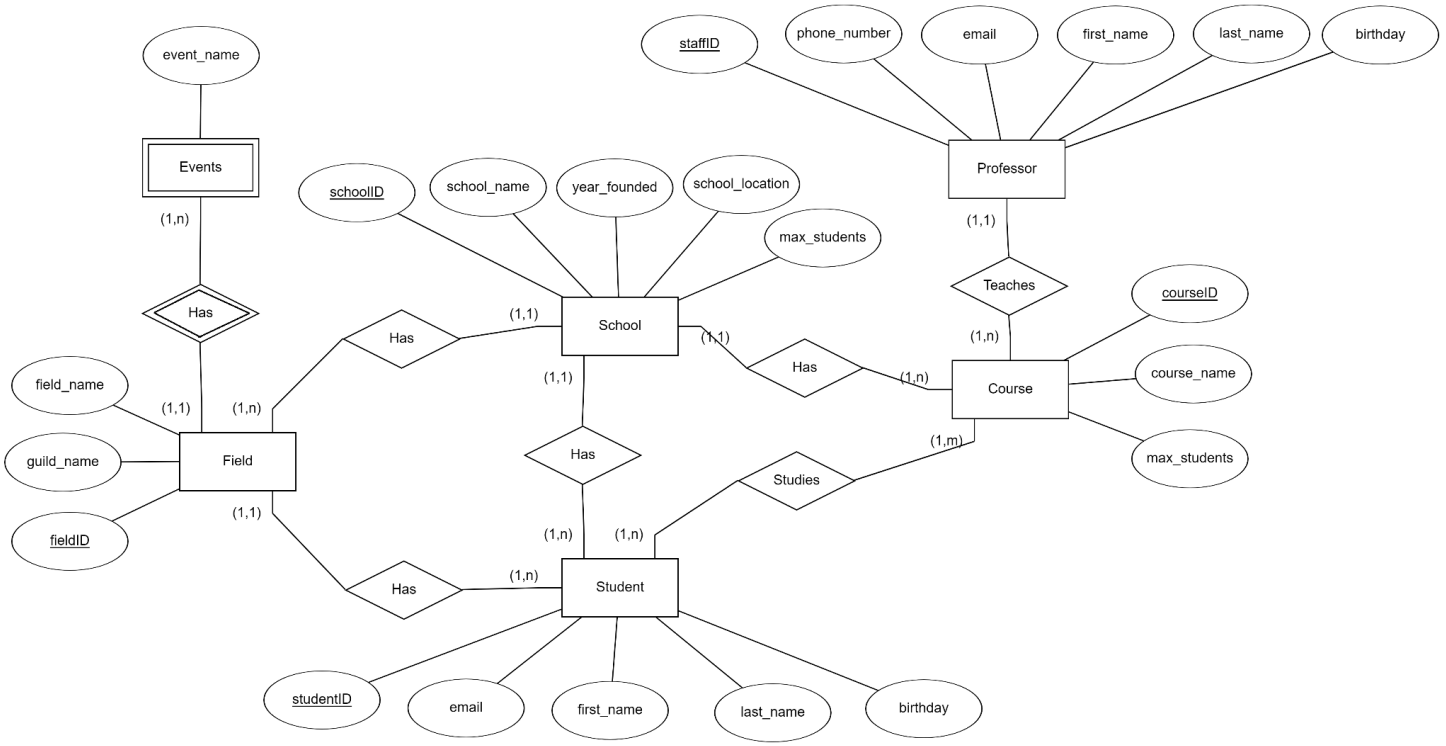


**Figure 1:** ER model

## 2.2 Logical model

Figure 2 shows the logical model of the project. The logical model of the project was created based on the ER model and the actual implementation of the database. Due to the N:M relationship between entities Courses and Students, an interim relation CourseInfomation was created between Courses and Students entities. While doing the actual implementation of the database, it was noticed that the CourseInformation could be used to easen the Professor entity also, so it was also added to the interim in addition to entities Course and Student.

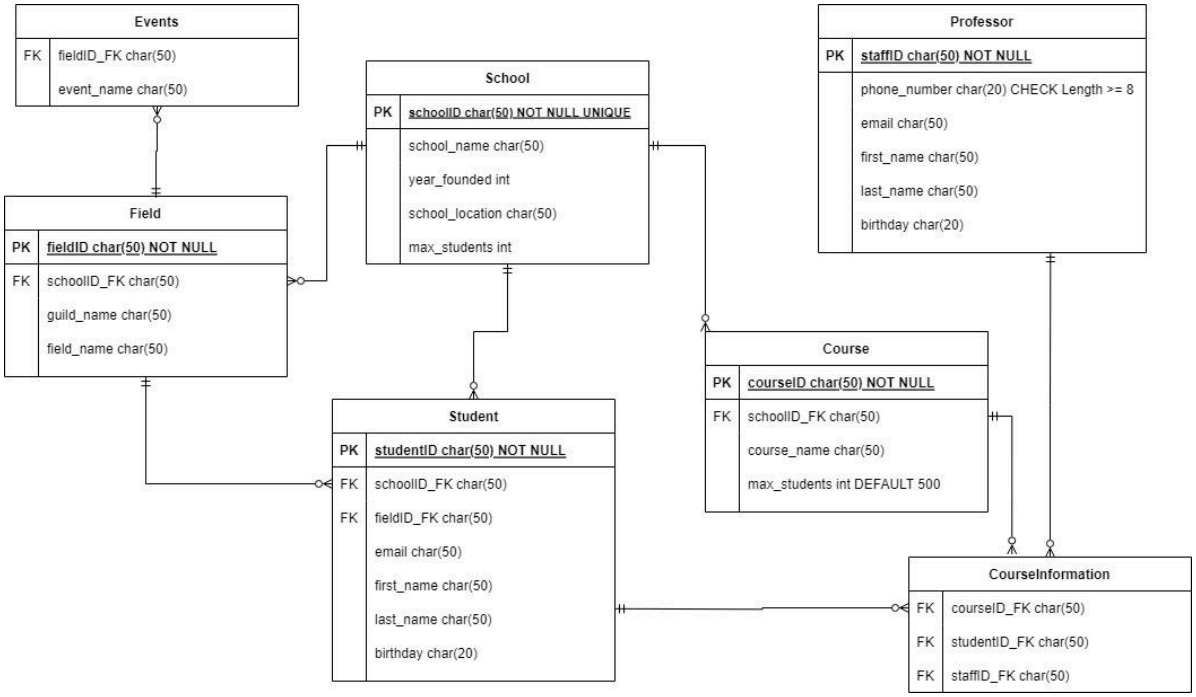


**Figure 2:** Logical model from the conceptual model

## 3 DATABASE IMPLEMENTATION

During implementation, the following constraints are created for the relations:

- **School**:
  - schoolID cannot be null(NOT NULL) and is unique (UNIQUE)
- **Student**:
  - Foreign key reference to field and school.
  - studentID cannot be null (NOT NULL)
  - ON UPDATE CASCADE
  - ON DELETE CASCADE
- **Course**:
  - courseID cannot be null (NOT NULL)
  - Foreign key reference to school.
  - max_students is defaults to 500 (DEFAULT)
  - ON UPDATE CASCADE
  - ON DELETE CASCADE
- **Professor**:
  - staffID cannot be null (NOT NULL)
  - phone number has to be over 8 digits (CHECK)
- **Field**:
  - Foreign key reference to field
  - ON UPDATE CASCADE
  - ON DELETE CASCADE
- **Events**:
  - fieldID cannot be null (NOT NULL)
  - Foreign key reference to school.
  - ON UPDATE CASCADE
  - ON DELETE CASCADE
- **CourseInformation**:
  - Foreign key reference to course, student and staff.
  - ON UPDATE CASCADE

- ON DELETE CASCADE

In addition to the integrity constraints listed above, a python user interface will be implemented. It will run in a loop and can showcase the 5 queries, and also will have the feature to add a course to the database, modify courses and delete courses.

## 4 DISCUSSION

### Challenges faced

While implementing the database there were a fair amount of challenges. The first challenge we found as a project team was that there might be situations where codes and ideas were crossed, but after a brief plan those were fixed. The second challenge was that at first the actual implementation was challenging in a browser based software, but after changing to an actual code editor the implementation improved a lot.

### Future Enhancements

In the future the UI will be enhanced to be a GUI (Graphical User Interface), to easen the use of the database for a new user. In addition to that a user documentation will be added to easen it more.