

База от данни за училище

Проектираме модел на база от данни, който ще описва училище.

Имаме множество от същности Student, чрез което се описва всеки ученик с помощта на следните атрибути: идентификационен номер (ID), който е уникален и еднозначно определя всеки един ученик, име (firstName), фамилия (lastName), рождена дата (birthdate), пол (gender), адрес (addr), телефон (phoneNumber) и ел.поща (e-mail).

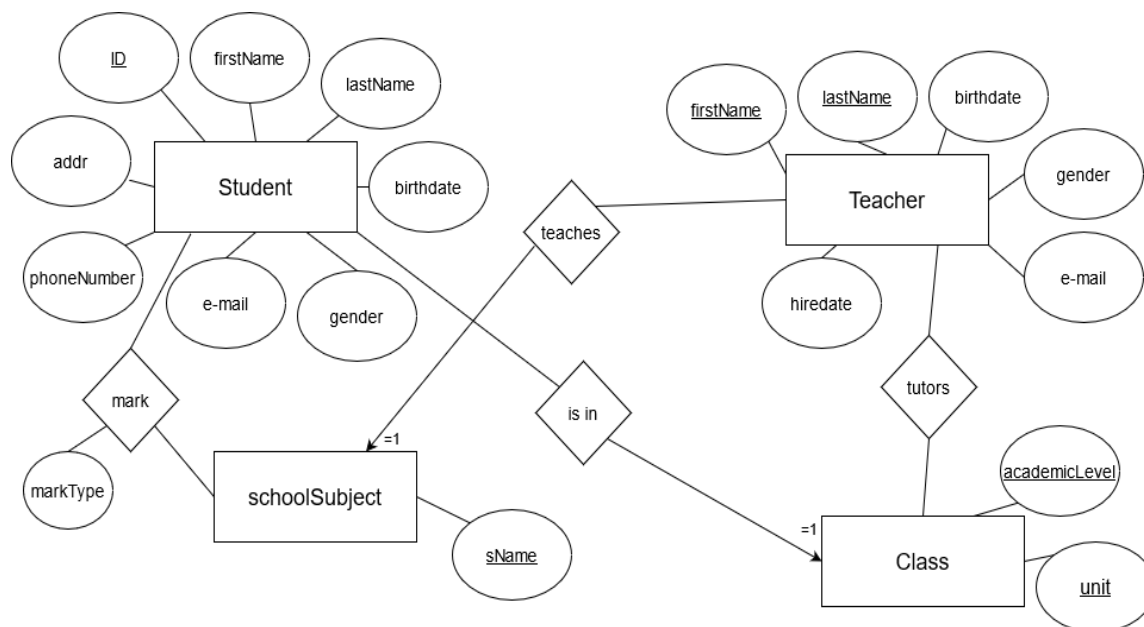
За учителите създаваме отделно множество от същности с атрибути име (firstName), фамилия (lastName), които в комбинация еднозначно определят всеки преподавател, рождена дата (birthdate), пол (gender), дата на назначаване на работа (hiredate) и ел.поща (e-mail).

Всеки учебен предмет е част от множеството от същности schoolSubject, което има единствен атрибут sName, явяващ се първичния му ключ.

Последното множество от същности Class служи за характеризация на класовете в училището, които еднозначно се определят от двата атрибута в съвкупност - академично ниво (academicLevel) и паралелка (unit).

Известно ни е, че всеки ученик принадлежи точно към един клас, а във всеки клас има много ученици (връзката is in). Всеки учител преподава точно един предмет и всеки за предмет има множество от учители, които могат да го преподават (връзката teaches). Всеки клас има екип учители, които му водят занятия, и е възможно един учител да преподава на няколко различни класа (връзката tutors). Всеки ученик има много оценки по различните предмети и по един предмет могат да бъдат нанесени много оценки на всеки ученик (връзката mark с атрибута markType, чиито стойности са ограничени между 2-6).

При моделиране с E-R диаграма горните твърдения придобиват вида:



След преобразуване на E-R модела към реляционен модел и оптимизиране на връзките, получаваме следните релации:

Student (ID, firstName, lastName, birthdate, gender, e-mail, classAcademicLevel, classUnit)

Teacher (firstName, lastName, birthdate, gender, hiredate, e-mail, subjectName)

schoolSubject (sName)

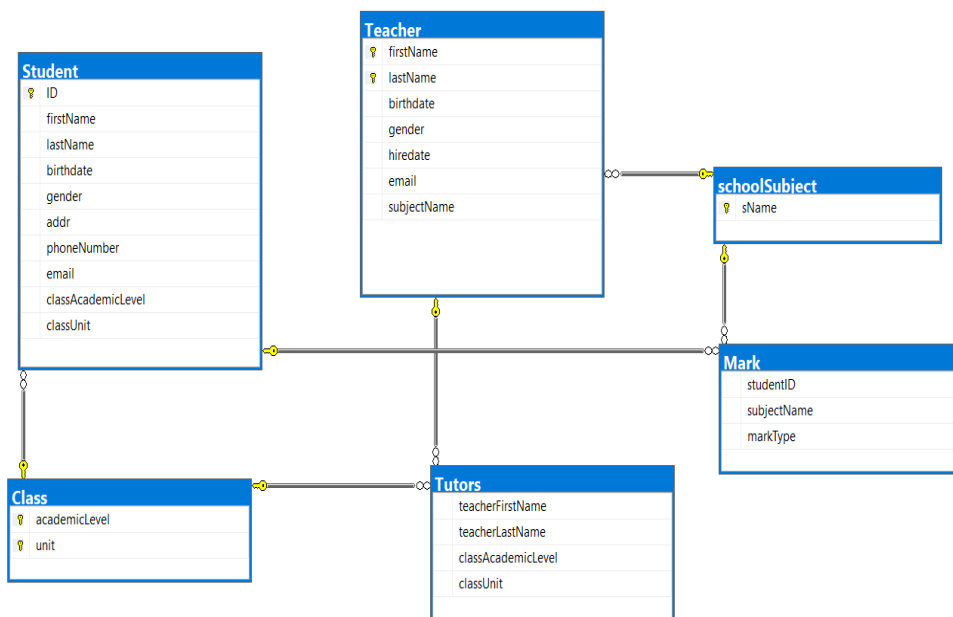
Class (academicLevel, unit)

Mark (studentID, subjectName, marktype)

Tutors (teacherFirstName, teacherLastName, classAcademicLevel, classUnit)

Подчертаните атрибути в релациите са техните първични ключове, а маркираните се явяват външи ключове. При създаване на базата от данни създаваме последователно таблиците със ъответните им атрибути.

В релациите Student, Teacher, schoolSubject и Class добавяме ограниченията NOT NULL PRIMARY KEY съответно за колоните ID на Student, комбинацията (firstName, lastName) на Teacher, sName на schoolSubject и комбинацията (academicLevel, unit) на Class, с което определяме тези атрибути като първични ключове за съответните таблици. Допълнително check ограничение добавяме и за колоната gender на Student и Teacher чрез CHECK gender in ('M', 'F'). Дефинираме и ограничението CHECK markType >= 2 AND markType <= 6 OR markType IS NULL за атрибута markType в Mark, за да осигурим действителни стойности на оценки в системата. Външните ключове за всяка от релациите дефинираме стандартно чрез REFERENCES, като добавяме опциите ON UPDATE CASCADE ON DELETE CASCADE, които гарантират, че при промяна или изтриване на реферирани стойности, това ще се отрази надолу по съответната връзка.



В реализацията на базата от данни са дефинирани два изгледа: Students_11C и AllMarks_11C.

Students_11C ни позволява в табличен вид да видим всички ученици на клас 11C. Понеже е изглед дефиниран върху една таблица, чрез него можем да променяме данните за Student, като опцията with check option лимитира тези промени само до атрибутите, които изгледа „вижда“ – ID, firstName, lastName.

AllMarks_11C извежда всички оценки по всички предмети на всички ученици от клас 11C, създавайки по този начин своеобразен дневник за този клас. Това е изглед дефиниран върху две релации – Student и Mark и е от вида read only.

Функциите, дефинирани в базата от данни, са AllSubjectTeachers и AverageMark.

AllSubjectTeachers е таблична функция, която връща таблица с имената на всички учители, които преподават подадения чрез аргумента @subject учебен предмет.

AverageMark е скаларна функция на два аргумента - @studentID, @subject. Спрямо подадените стойности на идентификационен номер и име на предмет, функцията връща средната аритметична стойност на оценката на ученика с подадения номер по подадения предмет.

Съхранената процедура в базата от данни е getAllMarksOfStudent. Позволява ни по подаден идентификационен номер чрез аргумента @ID да извлечем всички нанесени оценки на съответния ученик.

Тригерът HiringAge се задейства при добавяне на нов кортеж в релацията Teacher. Ако новодобавеният учител е на възраст под 21 години, т.е. разликата между hiredate и birthdate в години е по-малка от 21, за стойност на атрибута hiredate се задава NULL.