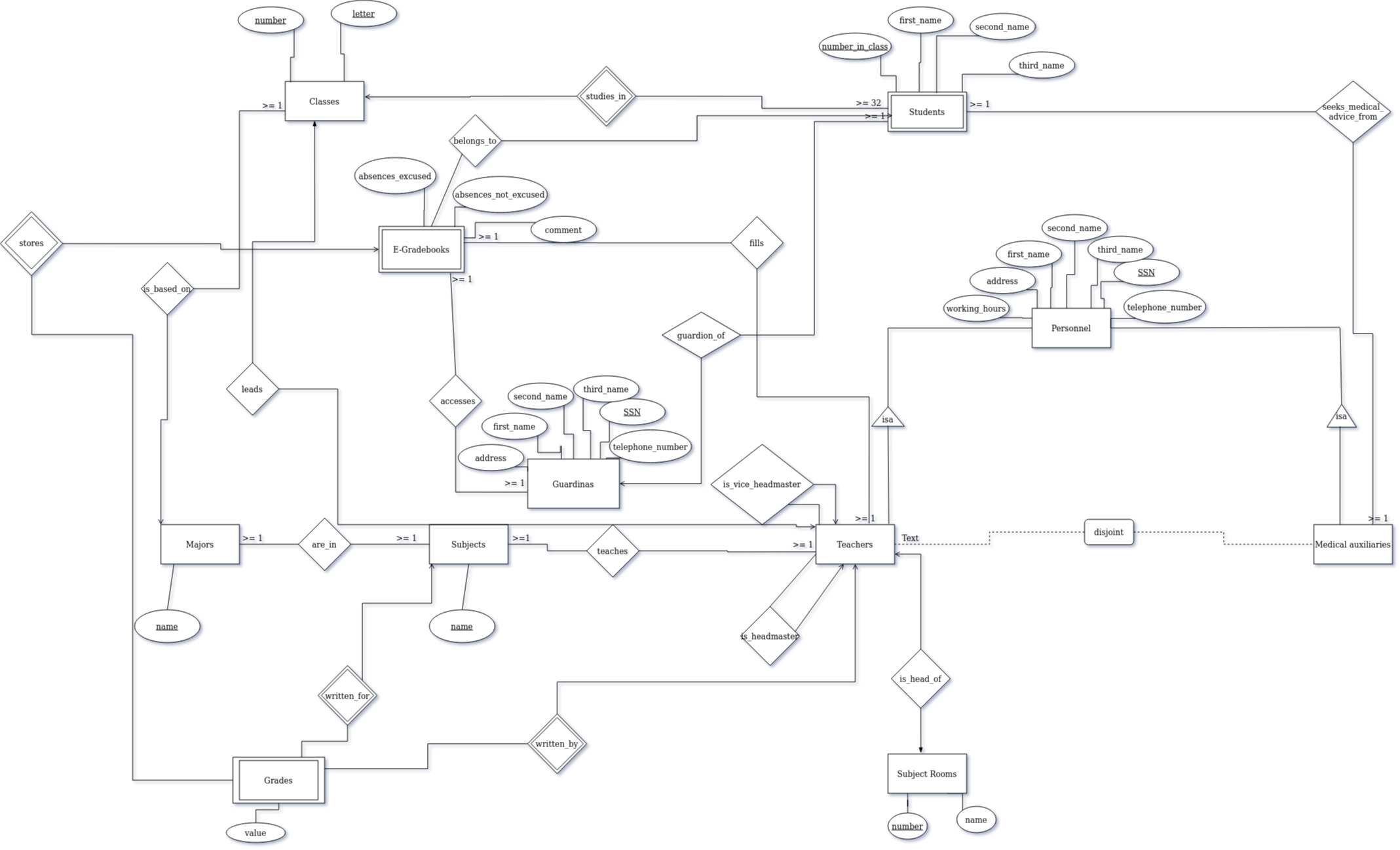
**Екип 8**

**Проект на тема: “База от данни за държавна профилирана гимназия”**

**Йоан Цвятков  
Симеон Христов**

1. **Множества от същности и техните атрибути**

Базата от данни се фокусира върху учениците, електронните дневници (в частност - оценките) и учителите. В тази връзка всеки ученик се характеризира с име, паралелка, номер в клас и родител-наставник, електронният дневник съдържа информация за номера на ученика, неговата паралелка, нанесени забележки и отсъствия. Оценките са слабо множество от същности, което пази информация за ЕГН-то на учителя, преподавания от него предмет, номера и паралелката на ученика, както и самата оценка. Учителят, има ЕГН, три имена, адрес, работно време, телефон, както и връзки към директора, зам.-директора и главния учител. Класът се характеризира с паралелката, към която принадлежи, профилът, от който е част и класен ръководител. Родителите имат отново ЕГН, три имена, адрес и телефонен номер, но имат и връзка със студентите. Фелдшерите поддържат същите атрибути, но имат и връзка с учениците, която се изразява в това да пазят тяхните медицински документи. Всички други служители (санитари, охрана, техници, счетоводители) имат ЕГН, три имена, адрес, работно време и телефон за връзка. Предметите имат име и задават важни връзки с учителите, оценките и паралелките.

1. **E/R диаграма на модела на БД**
2. **Преобразуване от E/R модел към релационен модел**

**Is-a heirarchy (using the OO method)**

Personnel (SSN, firstName, secondName, thirdName, address, telephoneNumber, workingHours)

PersonnelMedicalAuxiliaries (SSN, firstName, secondName, thirdName, address, telephoneNumber, workingHours)

PersonnelTeachers (SSN, firstName, secondName, thirdName, address, telephoneNumber, workingHours, headmasterSSN, viceHeadmasterSSN, subjectRoomNumber)

**Strong sets**

SubjectRooms (number, name)

Classes (number, letter, majorName, teacherSSN)

Majors (name)

Subjects (name)

Guardians (SSN, firstName, secondName, thirdName, address, telephoneNumber)

**Weak sets**

Students (numberInClass, firstName, secondName, thirdName, classNumber, classLetter, guardianSSN)

E-Gradebooks (absencesNotExcused, absencesExcused, comment, studentNumber, studentClassNumber, studentClassLetter)

Grades (value, studentNumber, studentClassNumber, studentClassLetter, subjectName, teacherSSN)

**Many-many relationships**

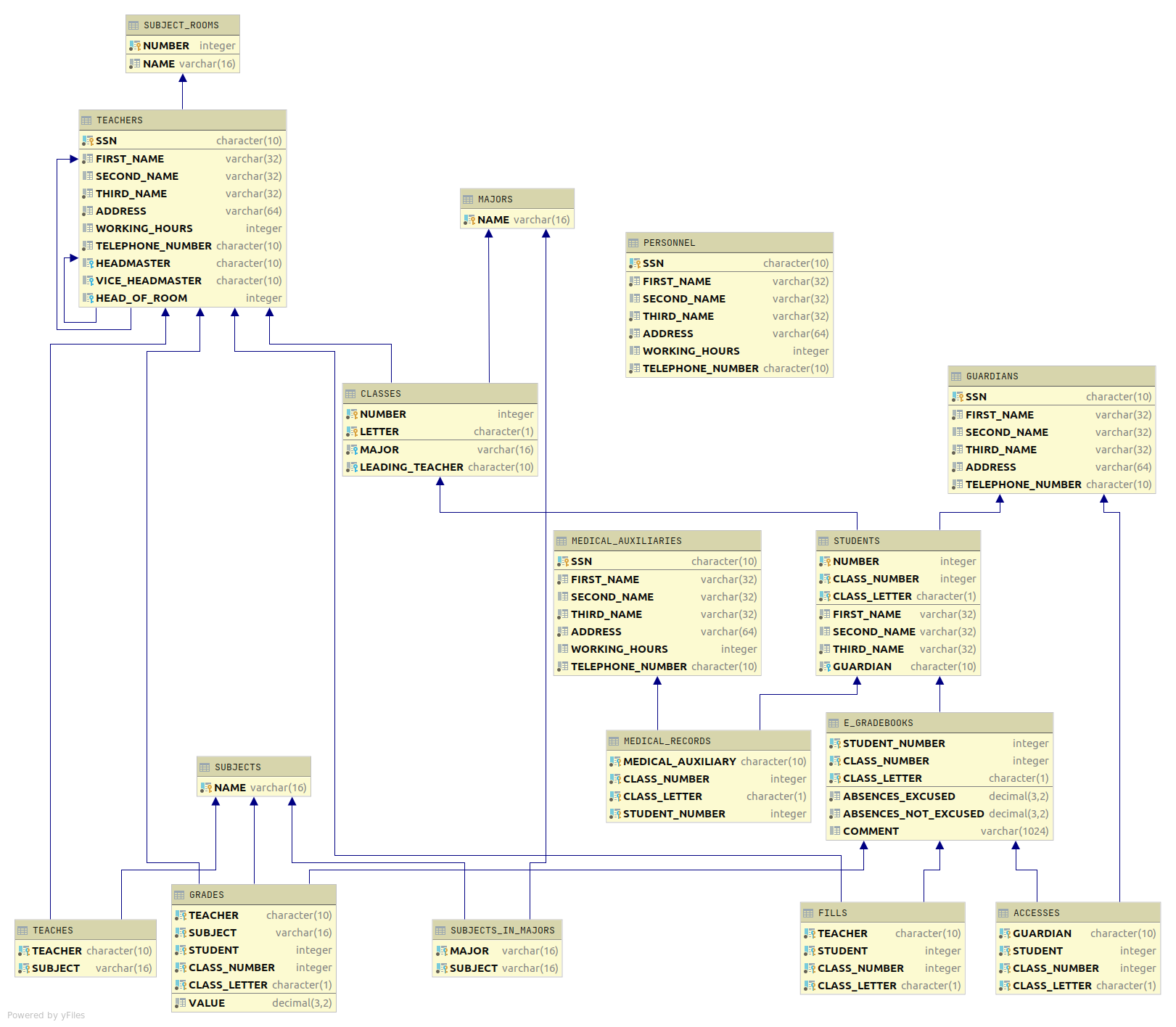
SeeksMedicalAdvice(studentNumber, studentClassNumber, studentClassLetter, medicalAuxiliarySSN)

Fills (studentNumber, studentClassNumber, studentClassLetter, teacherSSN)

Accesses (studentNumber, studentClassNumber, studentClassLetter, guardianSSN)

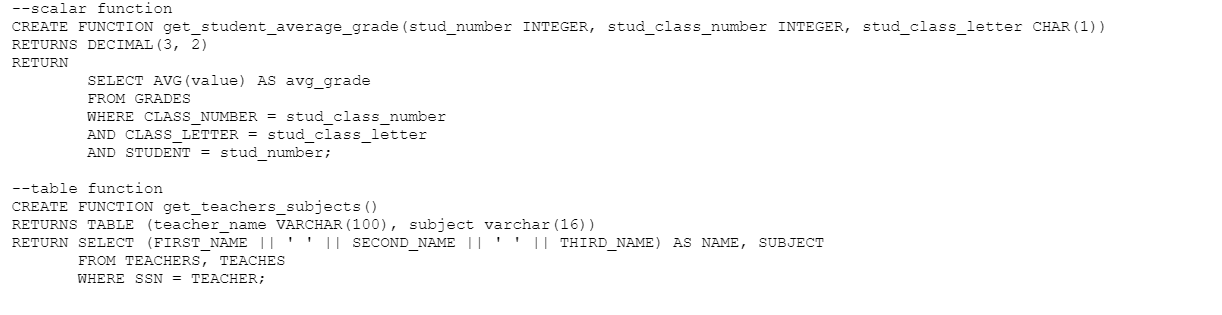
AreIn (majorName, subjectName)

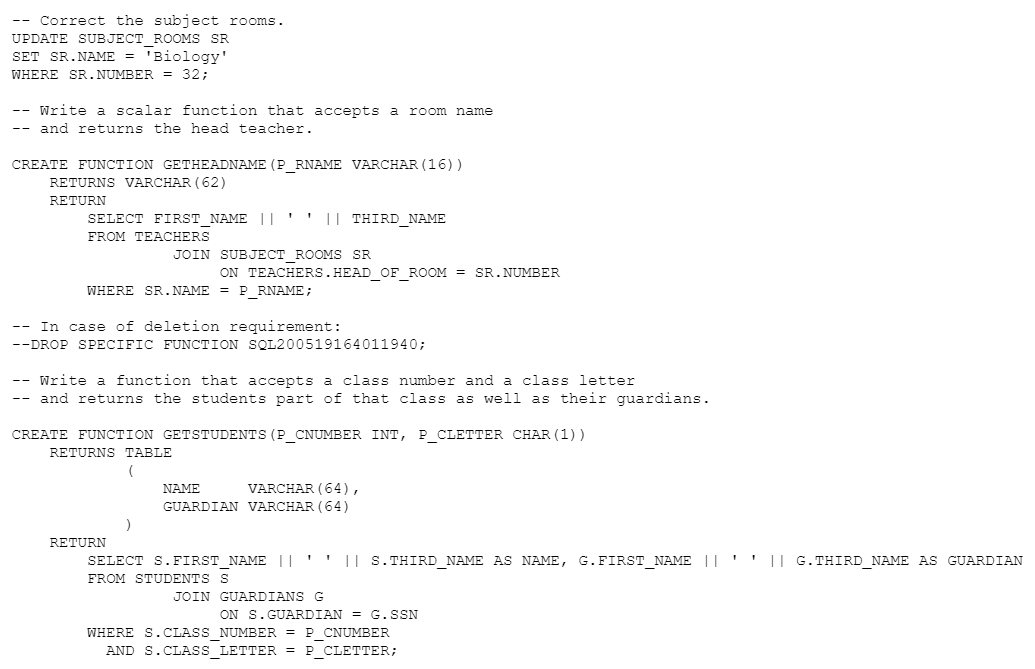
Teaches (subjectName, teacherSSN)

1. **Картинка на релационния модел**
2. **Ограниченията към БД.**

* Ако ученик има двеста извинени часа получава предупреждение за преместване в друга паралелка, при двеста и петдесет бива премесет, а при триста - изключен от училището.
* Ако ученик получи десет неизвинени получава предупреждение за преместване за изключване, при петнадесет бива преместен.
* Ученик може да бъде записан само в един профил, който в последствие НЕ може да се променя.
* Училището има точно един (зам.-)директор.

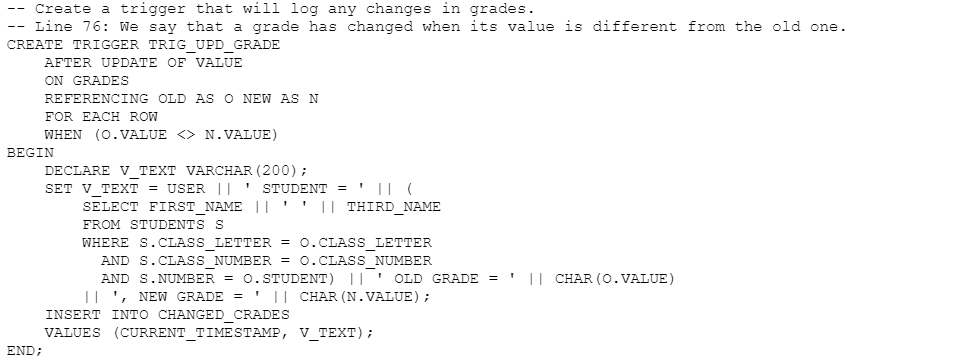
1. **Описание на функциите**

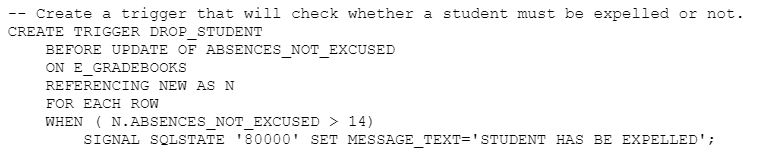
****

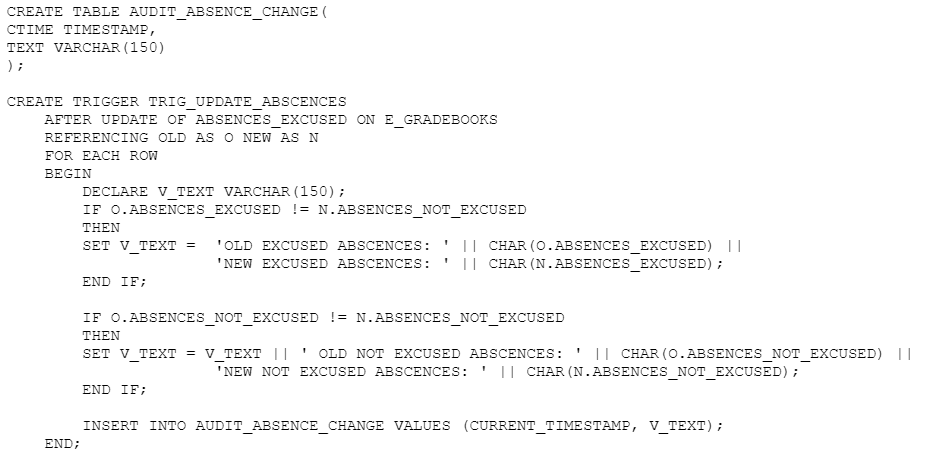


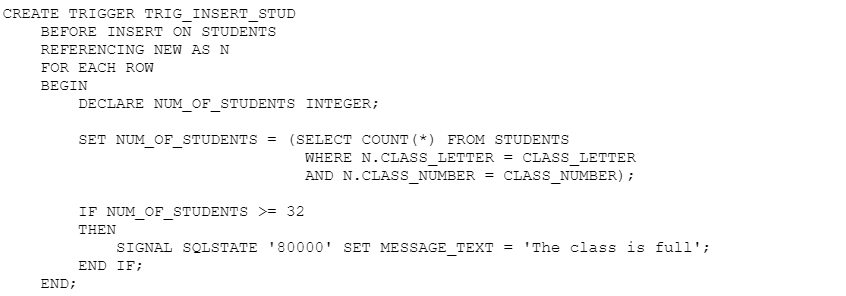
Имаме два вида функции: скаларни и таблични. Първата скаларна функция по подадени номера на ученик, номер в клас и символ за клас връща средния успех на ученика от този клас. Следващата скаларна функция по подадено име на стая връща главният учителя, който преподава в тази стая. Първата функция, която връща таблица, има за резултат таблица с имената на учителите и предмета, по който преподават. И последната функция, която връща таблица, приема номер и символ на клас и връща имената на учениците и името на техните настойници в този клас.

1. **Описание на тригерите**

****

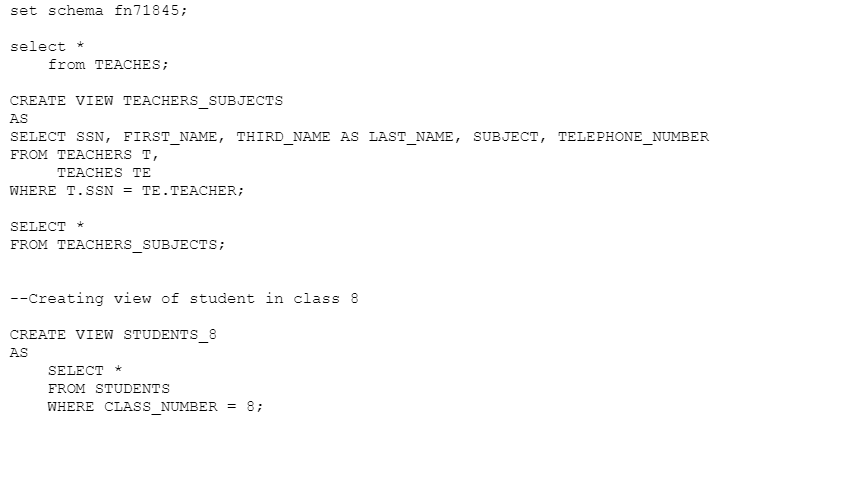
****

****

****

Имаме два вида тригери: BEFORE и AFTER. За първият AFTER тригер ни е необходима таблица с време, в което е сменена оценката на ученик, и символен низ. Целта на този тригер е да записва промените, направени върху таблицата с оценките, в таблицата CHANGED\_GRADES. За вторият AFTER тригер също ни е необходима нова таблица, в която ще записваме информация за промените на отсъствията на учениците. Този тригер създава символен низ, като в него ще пазим извинените и неизвинените отсъствия, както и времето, в което са били променени. Тези промени ще записваме в таблицата AUDIT\_ABSCENCE\_CHANGE. Първият BEFORE тригер проверява дали преди промени в таблицa E\_GRADEBOOKS на неизвинените отсъствия има ученик с повече от 14 такива отсъствия. Ако това е така, то хвърля изключени, което показва че ученика трябва да бъде изключен. Последният BEFORE тригер при добавяне на ученик в даден клас, проверява дали този клас не е запълнен. Ако това е така то хвърляме изключение за грешка, което ни информира че калсът вече е запълнен.

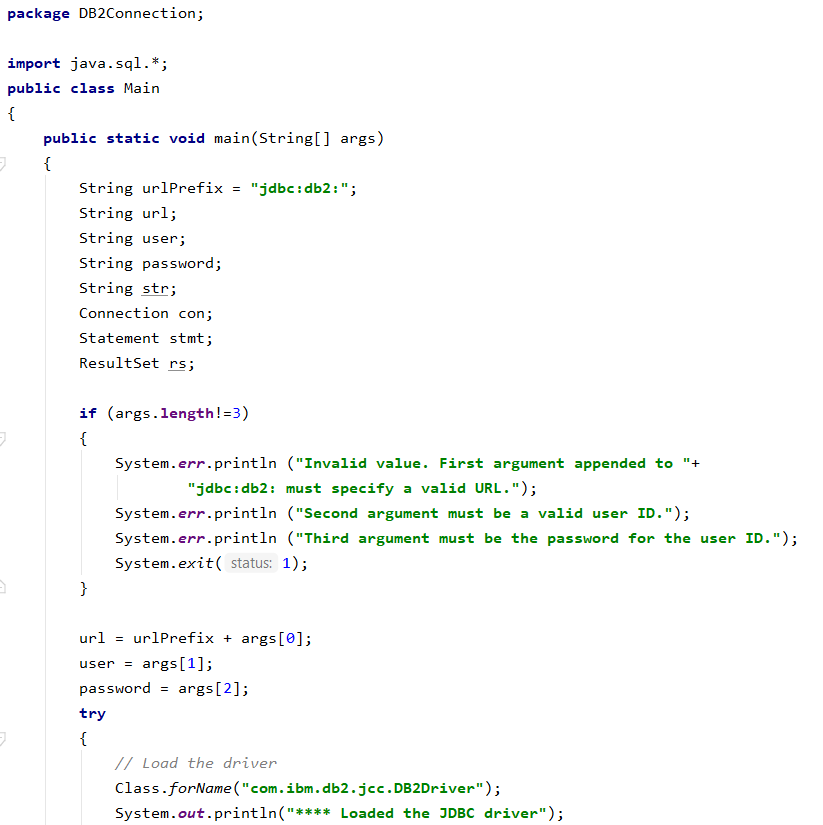
1. **Описание на изгледите**

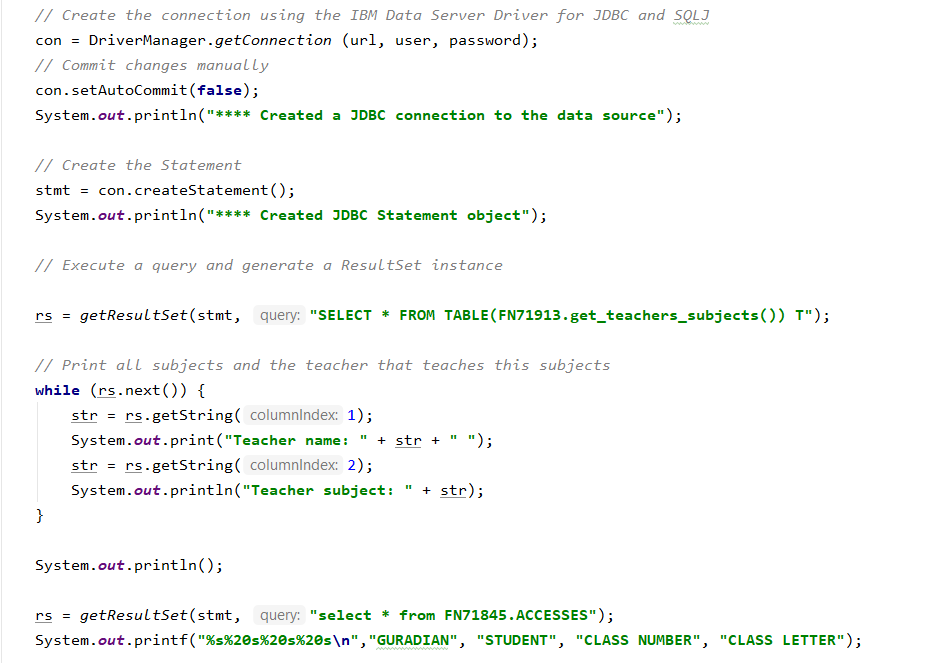


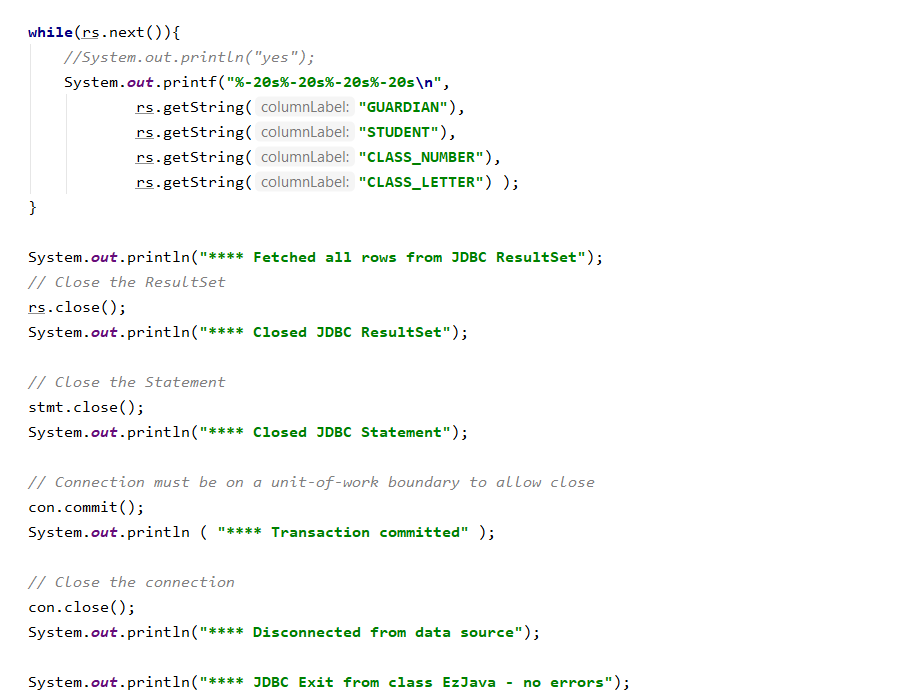
Първият изглед се състои от резултата на заявка, която прави съединение между таблиците TEACHERS и TEACHES. Атрибутите на този изглед са имената на учителите и предметите, по които преподават, и телефонните им номера.

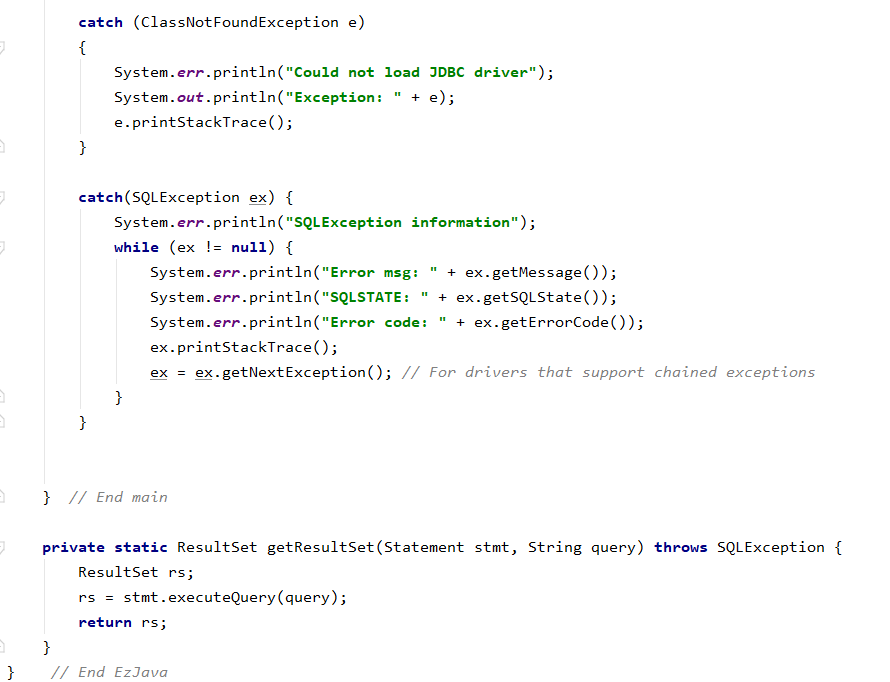
Вторият изглед се състои от резултата на заявка, която намира всички ученици в 8 клас.

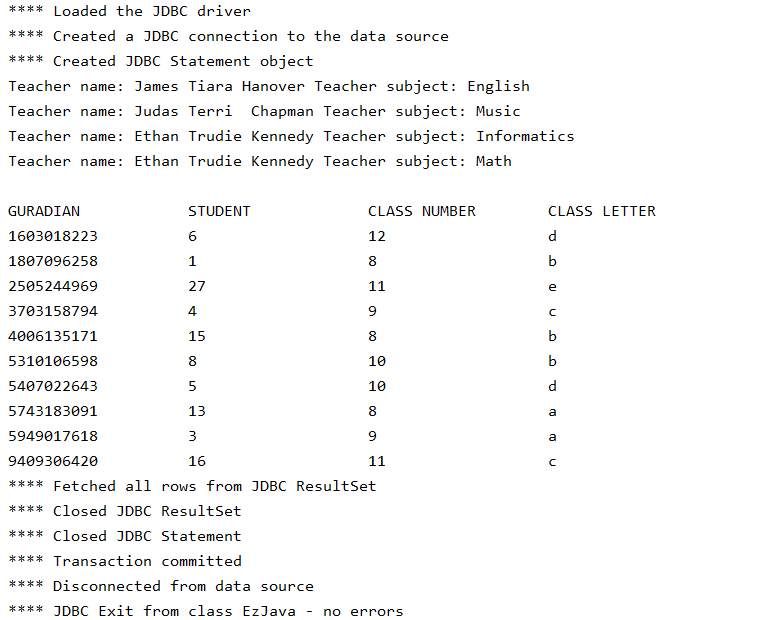
1. **Описание на приложението за достъп с резултатите от заявката**

****

****

****

****

****

За да може да направим заявките към базата от данни първо трябва да направим връзката между нея и java. За целта използваме статичния метод getConnection на класа DriverManager. Този метод приема като аргумент url, name, password. За да направим заявка към базата от данни трябва да създадем обект от тип Statement. След това ни трябва обект от тип ResultSet чрез който ще може да съхраняваме резултатите от нашата заявка. За създаването на инстанция от тип ResultSet използваме метода getResultSet, подавайки му обект от тип Statement и SQL заявка. Едната заявка в нашето приложение извиква функцията, която връща таблица с учителите и предмета, който те ръководят, а другата извежда информацията от таблицата ACCESSES. Итерираме резултатите от съответните заявки и извеждаме всеки един атрибут на стандартния изход.