**Тема 7**

**Проект по база от данни за Електронни тестове**

**Изготвил: Иван Петров, ф.н. 3MI0700035**

1. **Обхват на модела. Дефиниране на задачата.**

Базата от данни за електронни тестове, ще съхранява информация за данните в университета нужни за тестовете. Разработената база от данни ще обслужва вътрешната система на университета, т.е. няма да се обслужва уеб приложение. Университета разполага с много факултети, но създаваните тестове не зависят от факултета. Всеки тест се определя еднозначно по идентификационен номер.

Тестовете се състоят от множество въпроси, за които се пази информация. Точките, които допринасят, когато са отговорени вярно са различни за всеки въпрос. Всеки въпрос допринася различен брой точки в различните тестове, в които участва. Въпросите се определят еднозначно от идентификационен номер. Въпросите се разделят йерархично в ООП модел на множество типове въпроси, всеки с различна характеристика.

Типовете въпроси, добавени вече в системата, са: Есета, Затворени въпроси и Изчислителни въпроси. Всички те се определят еднозначно с идентификационния номер на въпросите, и имат условие в текстово поле. Есетата имат допълнителни характеристики за максимална и минимална дължина на есето.

В системата се пази информация за отговорите, които еднозначно се определят от идентификационен номер и номер на въпроса за когото отговарят. В отговорите се пази информация за това дали са верен отговор на въпроса, към който сочат, както и стойността на отговора в текстов формат.

В базата данни се съхранява информация за всички студенти от различните факултети на университета. За тях се пази тяхното име и фамилия, факултет и курс. Те се определят еднозначно от десетсимволния си факултетен номер.

За всеки студент се пази информация, на коя дата се явява на определен електронен тест, както и всеки тест, който е взет в съчетание с получената от него оценка.

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

* Изпити - номер тест, заглавие, нужни точки за минаване, научна сфера
* Въпроси - номер въпрос, точки (поне една, с точност до един знак след десетичната запетая, конкретния брой зависи от изпита)
  + Есе - условие, минимална дължина на отговор, максимална дължина на отговор
  + Затворен въпрос - условие
  + Изчислителен въпрос - условие
* Отговори - номер отговор, текст
* Студенти - факултетен номер, първо име, фамилия, курс, факултет

1. **Домейн на атрибутите**

* Изпити - номер тест : цяло положително число, заглавие: низ, минимални точки за минаване : положително цяло число, максимални точки на изпита, научна сфера: низ
* Въпроси - номер въпрос : цяло положително число, точки: цяло число (по-голямо или равно на 1)
  + Есе - условие : низ, минимална дължина на отговор: цяло положително число, максимална дължина на отговор: цяло положително число
  + Затворен въпрос - условие : низ,
  + Изчислителен въпрос - условие: низ
* Отговори - номер отговор: цяло положително число, текст: низ
* Студенти - факултетен номер : низ, първо име : низ, фамилия: низ, курс : цяло положително число, факултет : низ

1. **Връзки**

* В един Изпит има много Въпроси. Въпросите участват в много Изпити.
* Всеки Изпит има 0 или 1 нужен Изпит.
* Въпросите имат различни точки във всеки изпит
  + Есето е Въпрос
  + Затвореният въпрос е Въпрос
  + Изчислителният въпрос е Въпрос
* Един Въпрос има много Отговори. Един Отговор се асоциира само с един Въпрос.
* Студентите могат да се явят на много Изпити. На един Изпит се явяват много Студенти.
* Студентите могат да вземат много Изпити. Един Изпит може да бъде взет от много Студенти.

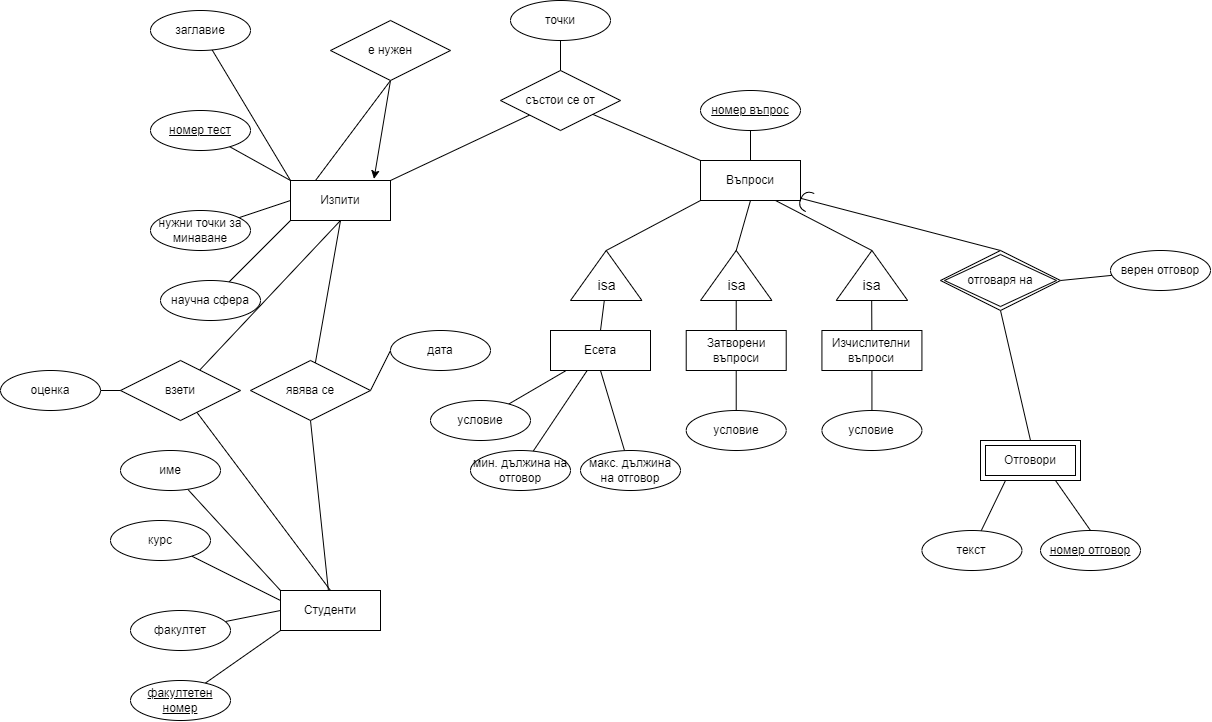
1. **Ключове**

* Изпити - номер тест : еднозначно определя изпита
* Въпроси - номер въпрос : еднозначно определя въпроса
* Отговори - номер отговор : еднозначно определя въпроса
* Студенти - факултетен номер: еднозначно определя студента

1. **Правила и проверки**

* За изпитите - минимални и максимални точки: точките трябва да са неотрицателни и максималните точки трябва да са повече или равни на минималните
* За въпросите - точки: се прави проверка дали са неотрицателни
* За есетата - минималната дължина и максималната дължина: минималната дължина се прави проверка дали е по-малка или равна от максималната дължина
* За студентите - курс: се прави проверка да е повече от 0
* За взетите изпити - оценка: се прави проверка оценката да е по-голяма или равна на 2

1. **E/R модел на данни**



1. **Релационен модел на данни**

Students(fn : char(10), fname : varchar(50), lname : varchar(50), faculty : string(100), course : smallint)   
PK: (fn, unique, not null)   
CK: (course > 0)  
Exams(nexam : int, title : varchar(255), minpoints : smallint, maxpoints : smallint, science\_field : varchar(255), needed\_nexam : int)   
PK: (nexam, unique, not null)   
FK: (needed\_nexam)   
CK: (minpoints > 0)

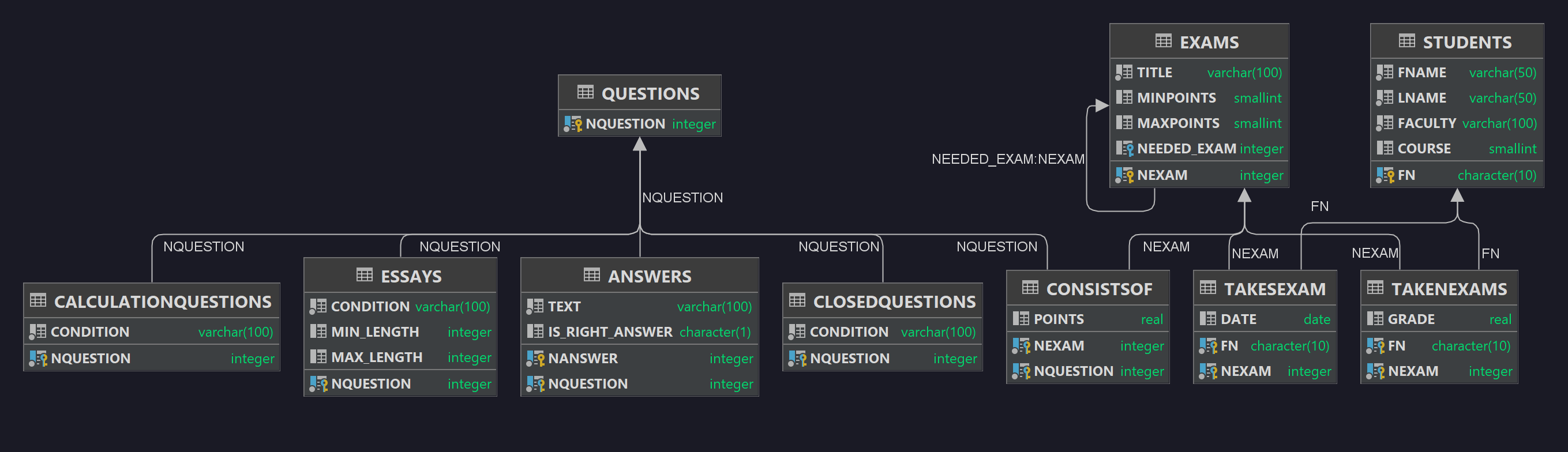
CK: (maxpoints>=minpoints)  
TakesExams(fn : char(10), nexam : int, date : date)   
PK: (fn, nexam, unique, not null)   
FK: (fn, nexam)  
TakenExams(fn : char(10), nexam : int, grade : decimal)   
PK: (fn, nexam, unique, not null)   
FK: (fn, nexam)   
CK: (grade >= 2)  
ConsistsOf(nexam : int, nquestion : int, points : decimal)   
PK: (nexam, nquestion , unique, not null)   
FK: (nexam, nquestion )   
CK: (points > 0)  
Answers(nanswer : int, text : varchar(100), nquestion : int, is\_right\_answer : char(1))   
Отговорите са слаби множества   
PK: (nanswer, text, nquestion, unique, not null)

FK: (nquestion)   
CK: (is\_right\_answer - ‘0’, ‘1’)  
Oкончателен вариант  
Questions(nquestion : int)   
PK:(nquestion, unique, not null)  
Essays(nquestion : int, condition : varchar(255), min\_length : int, max\_length : int)   
PK:(nquestion, unique, not null)   
FK:(nquestion)   
CK: (min\_length > 0)   
CK: (max\_length >= min\_length)  
ClosedQuestions(nquestion : int, condition : varchar(255))   
PK:(nquestion, unique, not null)   
FK:(nquestion)  
CalculationQuestions(nquestion : int, condition : varchar(255))   
PK:(nquestion, unique, not null)   
FK:(nquestion)

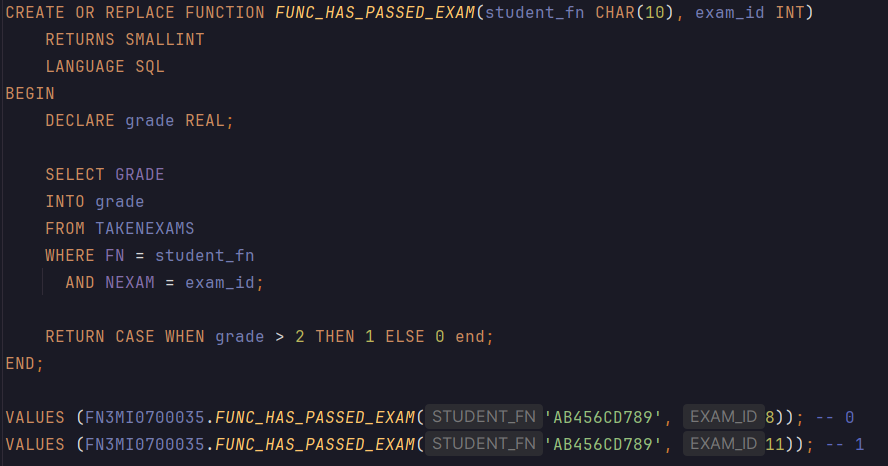
NULL-подход

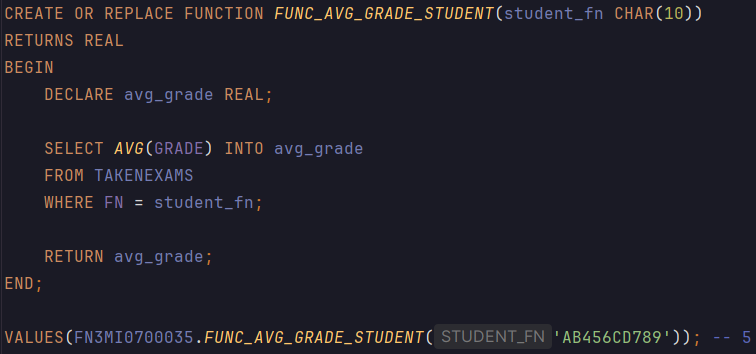
Questions(nquestion : int, condition : varchar(255), min\_length : int, max\_length : int)   
PK:(nquestion, unique, not null)   
CK: (min\_length > 0)   
CK: (max\_length >= min\_length)

1. **Схема на базата от данни**

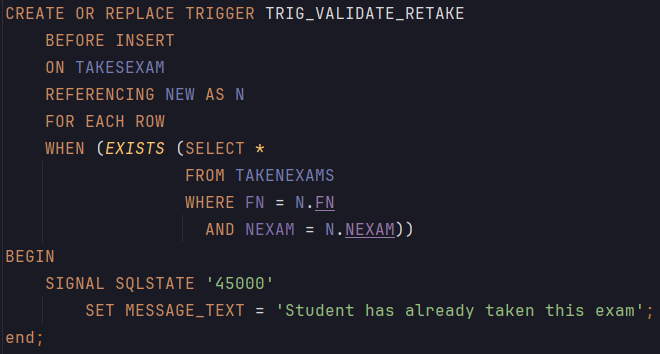


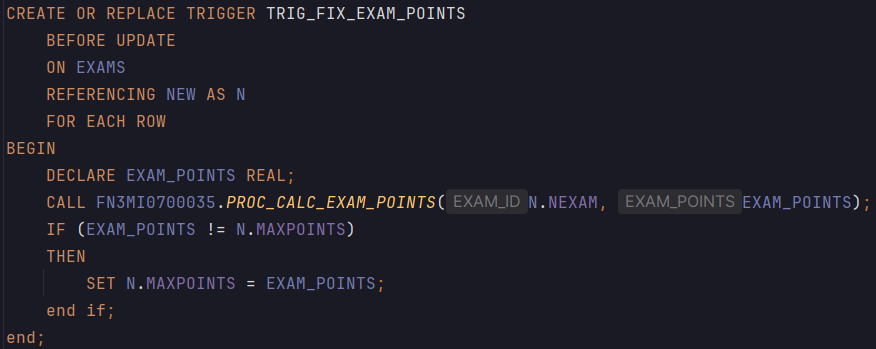
1. **Функции**

Функция за проверка дали студент е минал даден изпит

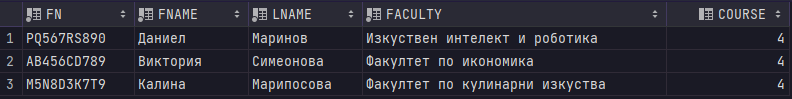
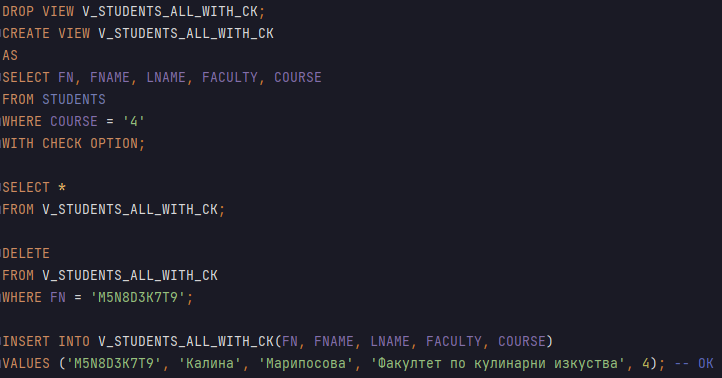
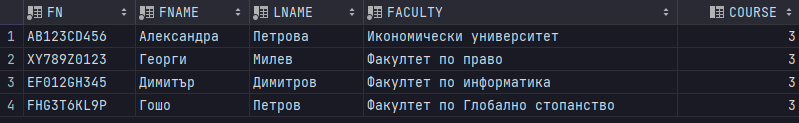
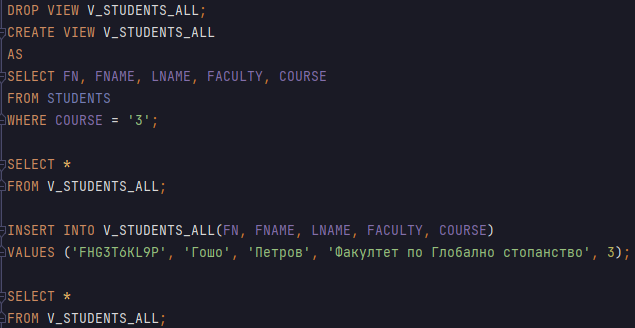
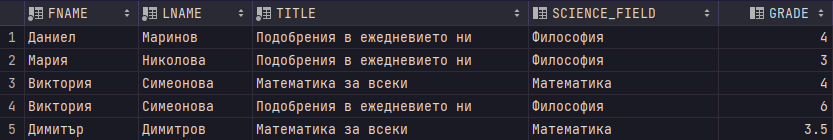
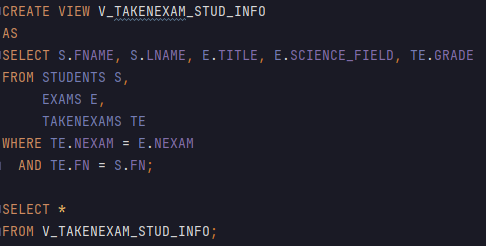
Функция за вземане средния успех на студент

1. **Тригери**

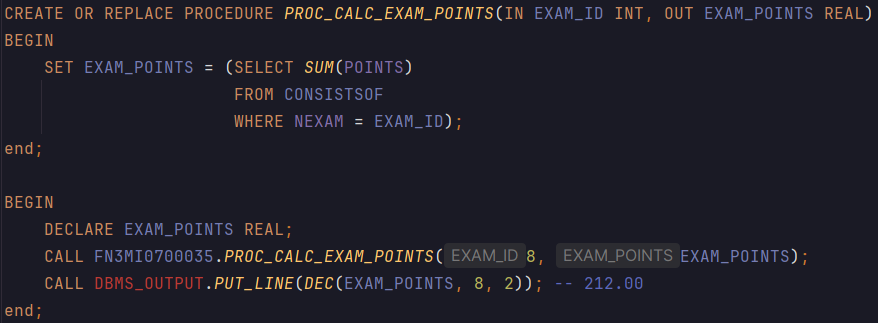
Тригер, който проверява дали студента си е взел изпита, преди вмъкване в таблицата за явяване на изпита

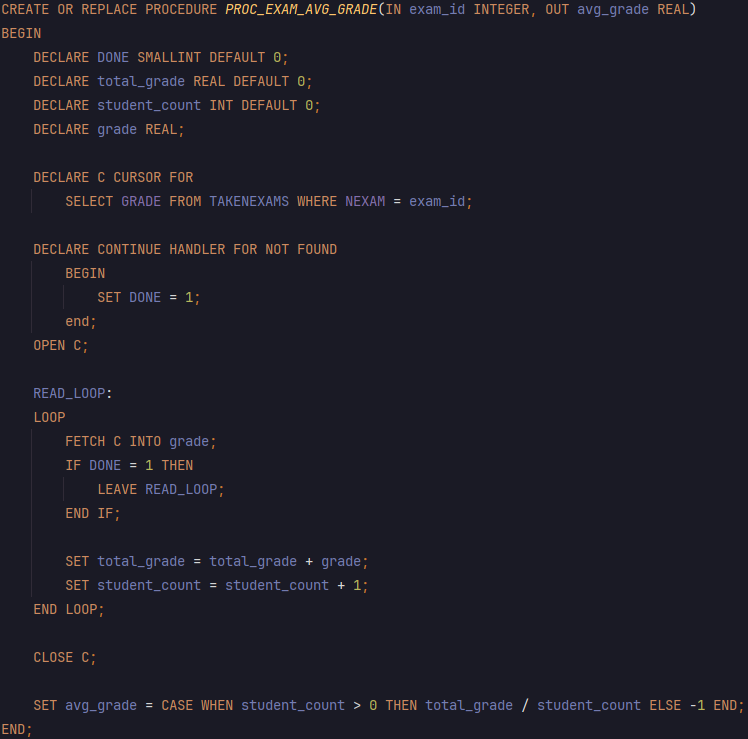
Тригер преди обновяване на Изпит с правилния брой максимален брой точки

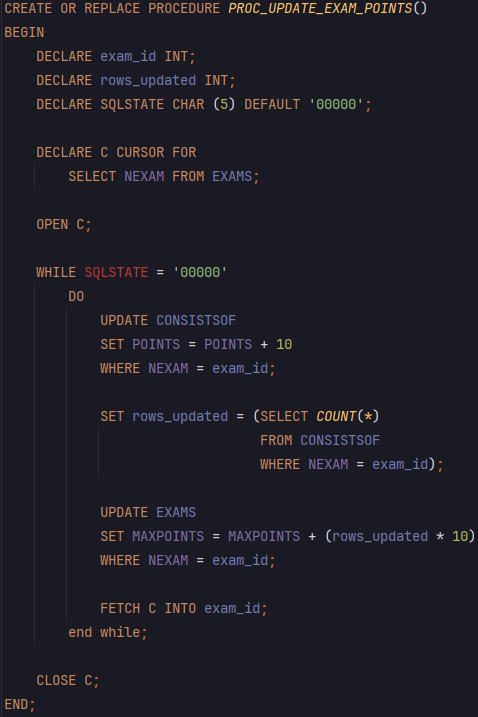
1. **Изгледи**

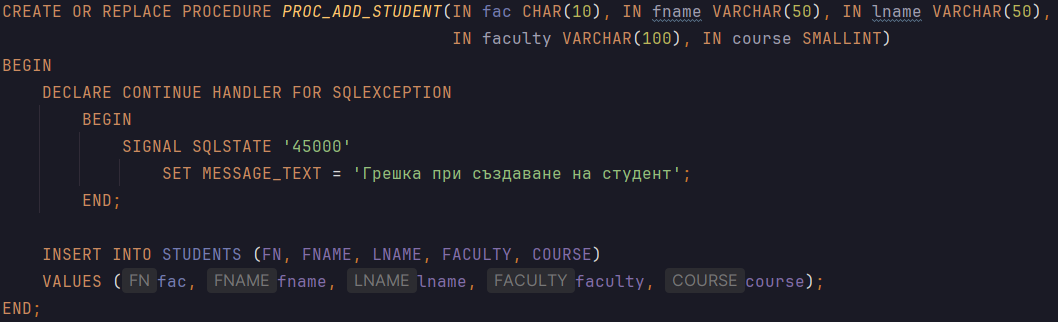


1. **Процедури**

Процедура за връщане на общ брой точки на изпит

Процедура за изчилсяване средна оценка за изпит

Процедура за увеличаване точките на отговорите и съответно изпита

Процедура за добавяне на студент

1. **Приложение за достъп до базата**

