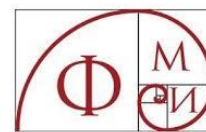


Софийски университет “Св. Климент  
Охридски”  
Факултет по математика и информатика



# Курсов проект по Базис Данни - практикум летен семестър 2020/2021

Тема - Университет  
Изработил: Веселин Славов Тодоров  
Специалност: Информационни системи  
Курс: 2  
Факултетен номер: 71923

Гр. София  
Май 2021

# Съдържание:

1. Дефиниране на задачата (множества от същности, връзки, ограничения, ...)
2. Представяне на базата от данни чрез E/R модел
3. Представяне на базата от данни чрез релационен модел
4. Нормализация
5. Създаване на релациите чрез DataGrip и SQL (DDL)
6. Вмъкване на информация в базата от данни (DML)
7. Създаване на Java приложение за използване на базата от данни
8. Използвана литература

# 1. Дефиниране на задачата

Базата от данни за университет, ще съхранява информация за данните на даден университет. Разработената база от данни ще бъде използвана за система, използвана от администрацията на университета, тоест няма да бъде общодостъпна. Университета разполага с факултети, които са в един и същи град. Всеки факултет се определя еднозначно от своя номер. Също така за всеки факултет пази информация относно тяхното наименование и адреса на който се намират.

Всеки факултет съдържа множество катедри, за които се пази информация. За тях се съхранява единствено тяхното наименование, което ги определя еднозначно. Един факултет може да има много катедри, но всяка катедра е точно към един факултет. В факултета учат студенти, които са определени еднозначно по ЕГН. За студентите се пази информация за име, ЕГН, факултетен номер, специалност и курс на обучение. Факултетния номер на всеки студент трябва да бъде строго положителен, а курса на обучение трябва да бъде в интервала от 1 до 6, като той не е задължителен и може да приема стойност NULL. Във всеки факултет могат да се обучават множество студенти, но всеки студент учи в точно един факултет.

Студентите посещават Курсове. За тези Курсове се пази информация за техния номер, година на провеждане, вид, име и кратко описание. Всеки курс се определя еднозначно от неговия номер и година на провеждане. Много студенти могат да посещават даден курс, тоест са записани на него, и всеки курс може да бъде посещаван от много студенти.

Базата съхранява информация и за Преподавателите в университета. За всеки преподавател се пази информация за неговите име, ЕГН и научно звание. ЕГН на преподавател го определя еднозначно, тъй като, въпреки че преподавателите са по-малко от студентите, може да има преподаватели с едни и същи

имена. Преподавателя притежава Научни степени. За Научните степени се пази информация за съкращение (акроним) и име. Всеки преподавател може да има много Научни степени и всяка Научна степен може да бъде притежавана от много преподаватели.

Преподавателя има право да води Курсове. Всички Преподаватели могат да водят много курсове (включително 0), но всеки от Курсовете може да бъде воден от точно един Преподавател. Катедрите имат Преподаватели към тях. Всяка Катедра може да има много Преподаватели, но всеки Преподавател може да бъде към точно една Катедра. Въпреки че университета има администрация, всяка Катедра трябва да има ръководител. Една Катедра може да има точно един Преподавател ръководител и всеки Преподавател може да бъде ръководител на точно една Катедра или на нито една (точно тази към която е). Преподавателите могат освен ръководители на Катебри, да бъдат и наставници на Студенти. Всеки преподавател може да бъде наставник на много Студенти, но всеки Студент има точно един наставник от Преподавателите.

Накратко информацията като множества от същности се систематизира по следния начин:

- Факултети – номер, наименование, адрес • Катебри – наименование
- Студенти – име, ЕГН, факултетен номер, специалност, курс на обучение
- Преподаватели – име, ЕГН, научно звание
- Научна степен – съкращение, име
- Курсове – име, номер, година на провеждане, вид, описание

Информацията за връзките:

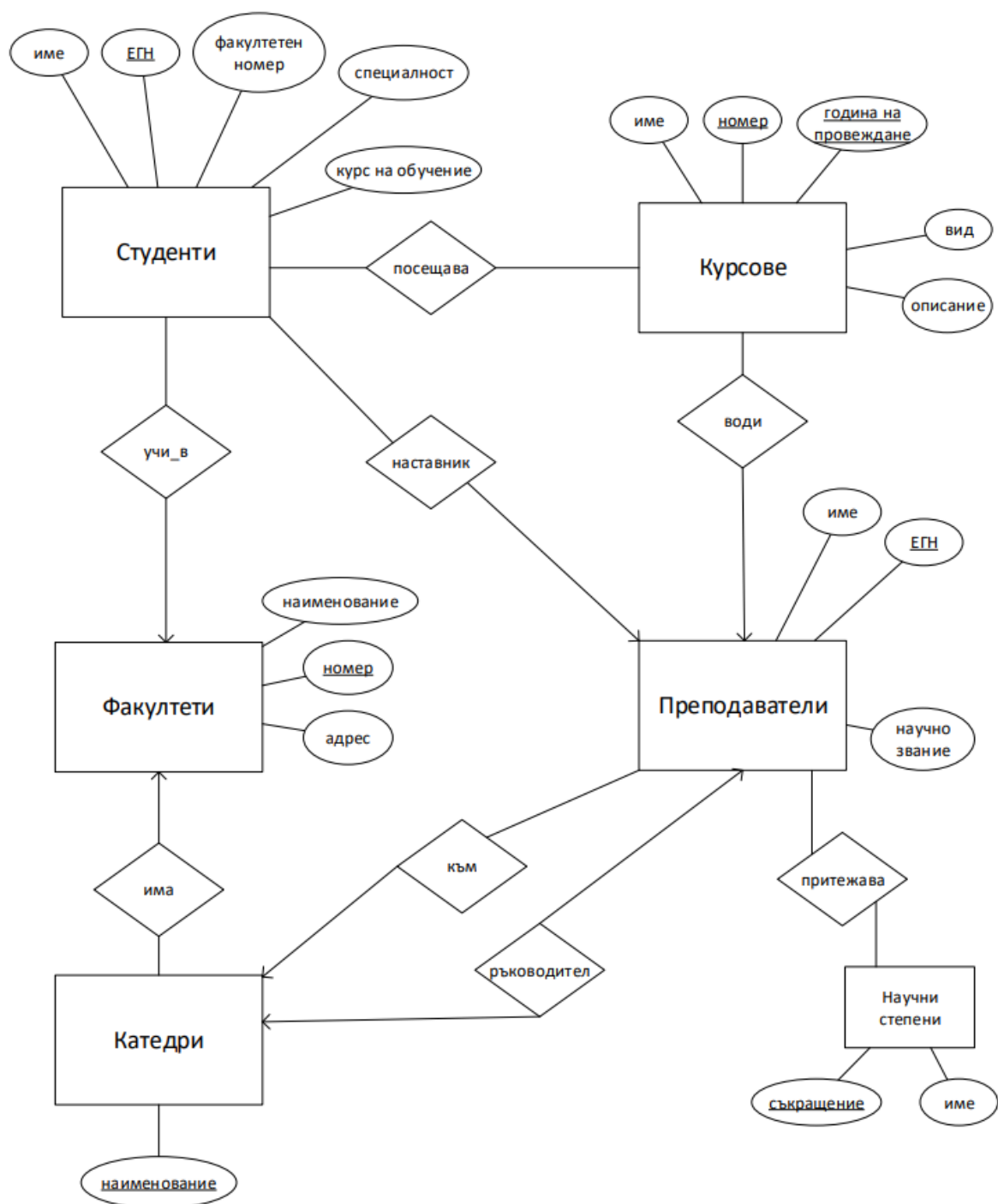
- Един Факултет има много Катебри. Катебриите са в точно един Факултет
- В един Факултет учат много Студенти. Студентите учат в точно един факултет

- Един Студент посещава много Курсове. Курсовете се посещават от много Студенти
  - Един Преподавател е наставник на много Студенти. Студентите са наставлявани от точно един Преподавател
- Един Преподавател води много Курсове. Курсовете са водени от точно един Преподавател
- Един Преподавател притежава много Научни степени. Научните степени могат да бъдат притежавани от много Преподаватели
  - Една Катедра има много Преподаватели към нея.
- Преподавателите са към точно една Катедра.
- Всяка Катедра има точно един ръководител Преподавател. Един Преподавател може да е ръководител на точно една катедра

Ограничения по единствена стойност, референтна цялостност и друг тип ограничения

- Факултети – номер: еднозначно определя факултета
- Катедри – наименование: еднозначно определя катедрата
- Студенти –ЕГН: еднозначно определя студента, факултетен номер: проверка за валидност, специалност: низ, курс на обучение: проверка за валидност
- Преподаватели –ЕГН: еднозначно определя преподавателя
- Научна степен – съкращение: еднозначно определя научната степен
- Курсове –номер и година на провеждане: еднозначно определят курса

## 2. E/R модел



### 3. Релационен модел

След преобразуване на всички множества от същности и връзки от E/R към релационен модел получаваме следния релационен модел:

Students(name, EGN, facultyNumber, major, yearOfStudying)

Courses(name, courseNumber, year, type, description)

Faculties(facultyName, facultyID, address)

Departments(name)

Teacher (name, teacherEGN, scientificTitle)

Degrees (abbreviation, degreeName)

Attends(studentEGN, courseNumber, courseYear)

HasDegree(teacherEGN, abbreviation)

StudiesIn(studentEGN, facultyID)

Mentor(studentEGN, teacherEGN)

InFaculty(facultyID, departmentName)

Teaches(teacherEGN, courseNumber, courseYear)

WorksIn(teacherEGN, departmentName)

HeadTeacher(teacherEGN, departmentName)

След оптимизиране на модела при връзките много към един и един към един получаваме следния релационен модел:

Students(name, EGN, facultyNumber, major, yearOfStudying, facultyID, mentor)

Courses(name, courseNumber, year, type, description, teacherEGN)

Faculties(facultyName, facultyID, address)

Departments(name, facultyID, teacherEGN)

Teacher (name, teacherEGN, scientificTitle, departmentName)

Degrees (abbreviation, degreeName)

Attends(studentEGN, courseNumber, courseYear)

HasDegree(teacherEGN, abbreviation)

## 4. Нормализация

За схемата на БД по-горе са в сила следните ФЗ:

ФЗ-1 EGN -> name, facultyNumber, major, yearOfStudying, facultyID, mentor (Students)

ФЗ-2 courseNumber, year -> name, type, description, teacherEGN (Course)

ФЗ-3 facultyID -> facultyName, address (Faculties)

ФЗ-4 name -> facultyID, teacherEGN (Departments)

ФЗ-5 teacherEGN -> name, scientificTitle, departmentName (Teachers)

ФЗ-6 abbreviation -> degreeName (Degrees)

## 5. Създаване на релациите чрез DataGrip и SQL (DDL)

Този скрипт описва създаването на таблиците в Datagrip:

```
SET SCHEMA FN71923;

--CREATING TABLES

--Table with faculty
CREATE TABLE FACULTIES
(
    facultyName VARCHAR(30) NOT NULL,
    facultyID CHAR(2) NOT NULL PRIMARY KEY,
    faddress VARCHAR(100)
);

--Table with department
CREATE TABLE DEPARTMENTS
(
    name VARCHAR(30) NOT NULL PRIMARY KEY,
    facultyID CHAR(2) NOT NULL,
    teacherEGN CHAR(10)
);

--Table with degree
CREATE TABLE DEGREE
(
    abbreviation VARCHAR(5) NOT NULL PRIMARY KEY,
    degreeName VARCHAR(30) NOT NULL
);

CREATE TABLE TEACHER
(
    name VARCHAR(50) NOT NULL,
    teacherEGN CHAR(10) NOT NULL PRIMARY KEY,
    scientificTitle VARCHAR(30),
    departmentName VARCHAR(30) NOT NULL
);

--Table with student
CREATE TABLE STUDENTS
(
    name VARCHAR(50) NOT NULL,
    EGN CHAR(10) NOT NULL PRIMARY KEY,
    facultyNumber CHAR(5) NOT NULL,
    major VARCHAR(30) NOT NULL,
    yearOfStudying INTEGER,
    facultyID CHAR(2) NOT NULL,
    mentor CHAR(10) NOT NULL
);

--Table with courses
CREATE TABLE COURSES
(
    name VARCHAR(50) NOT NULL,
    courseNumber INTEGER NOT NULL,
    year INTEGER NOT NULL,
    type VARCHAR(30),
    description VARCHAR(100),
    teacherEGN CHAR(10) NOT NULL,
    PRIMARY KEY (courseNumber, year)
```



```

CREATE TABLE ATTENDS
(
    studentEGN CHAR(10) NOT NULL,
    courseNumber INTEGER NOT NULL,
    courseYear INTEGER NOT NULL,
    PRIMARY KEY (studentEGN, courseNumber, courseYear)
);

--Table with has degree
CREATE TABLE HASDEGREE
(
    teacherEGN CHAR(10) NOT NULL,
    abbreviation VARCHAR(5) NOT NULL,
    PRIMARY KEY (teacherEGN, abbreviation)
);

-- FOREIGN KEY CONSTRAINTS
ALTER TABLE STUDENTS
ADD CONSTRAINT FK_STUDENTS_FACULTIES
FOREIGN KEY (facultyID) REFERENCES FACULTIES(facultyID); -- CHECKS

ALTER TABLE STUDENTS
ADD CONSTRAINT FK_STUDENTS_TEACHERS
FOREIGN KEY (mentor) REFERENCES TEACHER(teacherEGN);

ALTER TABLE COURSES
ADD CONSTRAINT FK_COURSES_TEACHERS
FOREIGN KEY (teacherEGN) REFERENCES TEACHER(teacherEGN);

ALTER TABLE DEPARTMENTS
ADD CONSTRAINT FK_DEPARTMENTS_TEACHERS
FOREIGN KEY (teacherEGN) REFERENCES TEACHER(teacherEGN);

ALTER TABLE ATTENDS
ADD CONSTRAINT FK_ATTENDS_STUDENTS
FOREIGN KEY (studentEGN) REFERENCES STUDENTS(EGN);

ALTER TABLE ATTENDS
ADD CONSTRAINT FK_ATTENDS_COURSES
FOREIGN KEY (courseNumber, courseYear) REFERENCES COURSES(courseNumber, year);

ALTER TABLE HASDEGREE
ADD CONSTRAINT FK_HASDEGREE_TEACHERS
FOREIGN KEY (teacherEGN) REFERENCES TEACHER(teacherEGN);

ALTER TABLE HASDEGREE
ADD CONSTRAINT FK_HASDEGREE_DEGREES
FOREIGN KEY (abbreviation) REFERENCES DEGREE(abbreviation);

ALTER TABLE STUDENTS
ADD CONSTRAINT CHECK_YEAR
CHECK (yearOfStudying IN (1,2,3,4) OR yearOfStudying IS NULL);

ALTER TABLE STUDENTS
ADD CONSTRAINT CHECK_FN
CHECK (facultyNumber > 0);

```

Първо се създават таблиците и се залага техния първичен ключ. След това са описани ограниченията по външен ключ и накрая са описани CHECK ограниченията

## 6. Вмъкване на информация в базата от данни (DML)

```

SET SCHEMA FN71923;

-- DEGREES
INSERT INTO DEGREE VALUES ('DMS', 'Doctor Math Sciences');
INSERT INTO DEGREE VALUES ('DIS', 'Doctor Informatic Sciences');
INSERT INTO DEGREE VALUES ('DL', 'Doctor Languages');
INSERT INTO DEGREE VALUES ('DCS', 'Doctor Chemistry Sciences');

-- FACULTIES
INSERT INTO FACULTIES VALUES ('FMI', '01', 'James Boucher No.5');
INSERT INTO FACULTIES VALUES ('FHF', '02', 'James Boucher No.5');

-- DEPARTMENTS
INSERT INTO DEPARTMENTS VALUES ('Algebra', '01', NULL);
INSERT INTO DEPARTMENTS VALUES ('Probability and statistics', 1, NULL);
INSERT INTO DEPARTMENTS VALUES ('Geometry', '01', NULL);
INSERT INTO DEPARTMENTS VALUES ('Analysis', '01', NULL);
INSERT INTO DEPARTMENTS VALUES ('Computer informatics', '01', NULL);
INSERT INTO DEPARTMENTS VALUES ('Software technologies', '01', NULL);
INSERT INTO DEPARTMENTS VALUES ('Differential Equations', '01', NULL);

```

```
--TEACHERS
INSERT INTO TEACHER VALUES ('Evgenia Velikova', '7012253415', 'PHD', 'Algebra');
INSERT INTO TEACHER VALUES ('Asen Bojilov', '6205233312', 'PROF', 'Algebra');
INSERT INTO TEACHER VALUES ('Maya Stoyanova', '7103043415', 'PHD', 'Algebra');
INSERT INTO TEACHER VALUES ('Dontcho Dontchev', '5812253415', 'PHD', 'Probability and statistics');
INSERT INTO TEACHER VALUES ('Deyan Palejev', '7605233312', 'DOC', 'Probability and statistics');
INSERT INTO TEACHER VALUES ('Bogdan Aleksandrov', '6301018632', 'PHD', 'Geometry');
INSERT INTO TEACHER VALUES ('Milko Takev', '6204018147', 'PHD', 'Analysis');
INSERT INTO TEACHER VALUES ('Radoslava Hristova', '8805123415', 'PHD', 'Computer informatics');
INSERT INTO TEACHER VALUES ('Vladimir Dimitrov', '5703113719', 'PROF', 'Computer informatics');
INSERT INTO TEACHER VALUES ('Dimitar Dimitrov', '9105223961', 'PHD', 'Computer informatics');
INSERT INTO TEACHER VALUES ('Kalin Georgiev', '8003083211', 'PHD', 'Computer informatics');
INSERT INTO TEACHER VALUES ('Olga Georgieva', '7410233920', 'DOC', 'Software technologies');
INSERT INTO TEACHER VALUES ('Tsvetan Hristov', '7808233022', 'PHD', 'Differential Equations');

--ADD MENTOR TO DEPARTMENTS
UPDATE DEPARTMENTS SET TEACHEREGN = '6205233312' WHERE NAME = 'Algebra';
UPDATE DEPARTMENTS SET TEACHEREGN = '5812253415' WHERE NAME = 'Probability and statistics';
UPDATE DEPARTMENTS SET TEACHEREGN = '6301018632' WHERE NAME = 'Geometry';
UPDATE DEPARTMENTS SET TEACHEREGN = '6204018147' WHERE NAME = 'Analysis';
UPDATE DEPARTMENTS SET TEACHEREGN = '8805123415' WHERE NAME = 'Computer informatics';
UPDATE DEPARTMENTS SET TEACHEREGN = '7410233920' WHERE NAME = 'Software technologies';
UPDATE DEPARTMENTS SET TEACHEREGN = '7808233022' WHERE NAME = 'Differential Equations';
```

## Данни за катедри и преподаватели

```
INSERT INTO STUDENTS VALUES ('Cvetelina Kirilova', '0000000001', '00001', 'Information systems', 2, '01', '6204018147');
INSERT INTO STUDENTS VALUES ('Dimitar Dimitrov', '0000000002', '00002', 'Information systems', 2, '01', '6204018147');
INSERT INTO STUDENTS VALUES ('Dimitar Sotirov', '0000000003', '00003', 'Information systems', 2, '01', '6204018147');
INSERT INTO STUDENTS VALUES ('Ivailo Stoyanov', '0000000004', '00004', 'Information systems', 2, '01', '6204018147');
INSERT INTO STUDENTS VALUES ('Kaloyan Sharkov', '0000000005', '00005', 'Information systems', 2, '01', '6204018147');
INSERT INTO STUDENTS VALUES ('Kristiana Atanasova', '0000000006', '00006', 'Information systems', 2, '01', '6204018147');
INSERT INTO STUDENTS VALUES ('Lachezar Peshterliev', '0000000007', '00007', 'Information systems', 2, '01', '6204018147');
INSERT INTO STUDENTS VALUES ('Rositsa Deyanova', '0000000008', '00008', 'Information systems', 2, '01', '6204018147');
INSERT INTO STUDENTS VALUES ('Veselin Todorov', '0000000009', '00009', 'Information systems', 2, '01', '6204018147');
INSERT INTO STUDENTS VALUES ('Yoanna Vasileva', '0000000010', '00010', 'Information systems', 2, '01', '6204018147');
INSERT INTO STUDENTS VALUES ('Aleksandar Filisyan', '0000000011', '00011', 'Information systems', 2, '01', '6204018147');
INSERT INTO STUDENTS VALUES ('Andon Denkov', '0000000012', '00012', 'Information systems', 2, '01', '6204018147');
INSERT INTO STUDENTS VALUES ('Atanas Ivanov', '0000000013', '00013', 'Information systems', 2, '01', '6204018147');
INSERT INTO STUDENTS VALUES ('Betina Mladenova', '0000000014', '00014', 'Information systems', 2, '01', '6204018147');
INSERT INTO STUDENTS VALUES ('Damiana Dimova', '0000000015', '00015', 'Information systems', 2, '01', '6204018147');
INSERT INTO STUDENTS VALUES ('Evelina Popova', '0000000016', '00016', 'Information systems', 2, '01', '6204018147');
INSERT INTO STUDENTS VALUES ('Nikolay Aleksiev', '0000000017', '00017', 'Information systems', 2, '01', '6204018147');
INSERT INTO STUDENTS VALUES ('Petar Zemqrski', '0000000018', '00018', 'Information systems', 2, '01', '6204018147');
INSERT INTO STUDENTS VALUES ('Mariya Qneva', '0000000019', '00019', 'Information systems', 2, '01', '6204018147');
INSERT INTO STUDENTS VALUES ('Aleksandar Ganchev', '0000000020', '00020', 'Information systems', 2, '01', '6204018147');

-- COURSES
INSERT INTO COURSES VALUES ('Linear algebra', 1, 2020, 'Applied maths', NULL, '7012253415');
INSERT INTO COURSES VALUES ('Fractals', 2, 2020, 'Maths', NULL, '6204018147');
INSERT INTO COURSES VALUES ('Databases', 3, 2020, 'Computer science core', NULL, '8805123415');
INSERT INTO COURSES VALUES ('Web technologies', 4, 2020, 'Computer science core', NULL, '8003083211');
INSERT INTO COURSES VALUES ('Intro to programming', 5, 2020, 'Computer science basics', NULL, '9105223961');

-- ATTENDS
INSERT INTO ATTENDS VALUES ('0000000002', 1, 2020);|
```

## Данни за студенти и курсове

```
INSERT INTO ATTENDS VALUES ('0000000002', 1, 2020);
INSERT INTO ATTENDS VALUES ('0000000005', 1, 2020);
INSERT INTO ATTENDS VALUES ('0000000011', 1, 2020);
INSERT INTO ATTENDS VALUES ('0000000013', 1, 2020);
INSERT INTO ATTENDS VALUES ('0000000016', 1, 2020);
INSERT INTO ATTENDS VALUES ('0000000001', 2, 2020);
INSERT INTO ATTENDS VALUES ('0000000005', 2, 2020);
INSERT INTO ATTENDS VALUES ('0000000010', 2, 2020);
INSERT INTO ATTENDS VALUES ('0000000014', 2, 2020);
INSERT INTO ATTENDS VALUES ('0000000017', 2, 2020);
INSERT INTO ATTENDS VALUES ('0000000003', 3, 2020);
INSERT INTO ATTENDS VALUES ('0000000009', 3, 2020);
INSERT INTO ATTENDS VALUES ('0000000011', 3, 2020);
INSERT INTO ATTENDS VALUES ('0000000018', 3, 2020);
INSERT INTO ATTENDS VALUES ('0000000019', 3, 2020);
INSERT INTO ATTENDS VALUES ('0000000001', 4, 2020);
INSERT INTO ATTENDS VALUES ('0000000004', 4, 2020);
INSERT INTO ATTENDS VALUES ('0000000011', 4, 2020);
INSERT INTO ATTENDS VALUES ('0000000019', 4, 2020);
INSERT INTO ATTENDS VALUES ('0000000020', 4, 2020);
INSERT INTO ATTENDS VALUES ('0000000016', 5, 2020);
INSERT INTO ATTENDS VALUES ('0000000007', 5, 2020);
INSERT INTO ATTENDS VALUES ('0000000008', 5, 2020);
INSERT INTO ATTENDS VALUES ('0000000010', 5, 2020);
INSERT INTO ATTENDS VALUES ('0000000011', 5, 2020);
```

Данни за посещаване на курсове от студенти

```

INSERT INTO HASDEGREE VALUES ('5812253415', 'DMS');
INSERT INTO HASDEGREE VALUES ('5812253415', 'DIS');
INSERT INTO HASDEGREE VALUES ('6204018147', 'DMS');
INSERT INTO HASDEGREE VALUES ('6301018632', 'DMS');
INSERT INTO HASDEGREE VALUES ('7808233022', 'DMS');
INSERT INTO HASDEGREE VALUES ('5703113719', 'DIS');
INSERT INTO HASDEGREE VALUES ('8805123415', 'DIS');
INSERT INTO HASDEGREE VALUES ('8003083211', 'DIS');
INSERT INTO HASDEGREE VALUES ('7410233920', 'DIS');
INSERT INTO HASDEGREE VALUES ('9105223961', 'DIS');

-- VIEWS
CREATE VIEW ATTENDEDCOURSES AS
SELECT S.NAME AS STUDENTNAME, C.NAME AS COURSENAME
FROM STUDENTS S, ATTENDS A, COURSES C
WHERE S.EGN = A.STUDENTEGN AND A.COURSEYEAR = C.YEAR AND A.COURSENUMBER = C.COURSENUMBER;

SELECT * FROM ATTENDEDCOURSES;

CREATE VIEW MENTOROFDEPARTMENTS AS
SELECT D.NAME AS DEPARTMENT, T.NAME AS TEACHER
FROM DEPARTMENTS D, TEACHER T
WHERE D.TEACHEREGN = T.TEACHEREGN;

SELECT * FROM MENTOROFDEPARTMENTS;

```

Данни за притежаване на научни степени от преподаватели.

```

CREATE VIEW ATTENDEDCOURSES AS
SELECT S.NAME AS STUDENTNAME, C.NAME AS COURSENAME
FROM STUDENTS S, ATTENDS A, COURSES C
WHERE S.EGN = A.STUDENTEGN AND A.COURSEYEAR = C.YEAR AND A.COURSENUMBER = C.COURSENUMBER;

SELECT * FROM ATTENDEDCOURSES;

CREATE VIEW MENTOROFDEPARTMENTS AS
SELECT D.NAME AS DEPARTMENT, T.NAME AS TEACHER
FROM DEPARTMENTS D, TEACHER T
WHERE D.TEACHEREGN = T.TEACHEREGN;

SELECT * FROM MENTOROFDEPARTMENTS;

```

Изглед показващ името на студент и записаните от него курсове.  
Изглед показващ име и ръководител на катедра.

## 7. Създаване на Java приложение за използване на базата от данни

Създадох конзолно Java приложение, което трябва да достъпва информация за студенти, преподаватели и курсове, тяхните взаимодействия и добавяне на данни за тях.

```
Menu:
1 - Find student by:
2 - Show attended courses
3 - Show students mentor
4 - Find teacher by:
5 - Show departments headteacher
6 - Enroll student
7 - Hire teacher
8 - Create course
9 - Exit
Your choice: 1
Choose criteria (course/egn): course
Choose course: 2
SELECT * FROM FN71923.STUDENTS WHERE yearOfStudying = 2
Executing query: SELECT * FROM FN71923.STUDENTS WHERE yearOfStudying = 2

Result output

-----

Cvetelina Kirilova, 0000000001, 00001, Information systems, 2, 1 , 6204018147
Dimitar Dimitrov, 0000000002, 00002, Information systems, 2, 01, 6204018147
Dimitar Sotirov, 0000000003, 00003, Information systems, 2, 01, 6204018147
Ivailo Stoyanov, 0000000004, 00004, Information systems, 2, 01, 6204018147
Kaloyan Sharkov, 0000000005, 00005, Information systems, 2, 01, 6204018147
Kristiana Atanasova, 0000000006, 00006, Information systems, 2, 01, 6204018147
Lachezar Peshterliev, 0000000007, 00007, Information systems, 2, 01, 6204018147
Rositsa Deyanova, 0000000008, 00008, Information systems, 2, 01, 6204018147
Veselin Todorov, 0000000009, 00009, Information systems, 2, 01, 6204018147
Yoanna Vasileva, 0000000010, 00010, Information systems, 2, 01, 6204018147
Aleksandar Filisyan, 0000000011, 00011, Information systems, 2, 01, 6204018147
```

фиг. 1 Информация за студенти по номер на курс на обучение

```

Your choice: 2
Enter egn: 0000000001
Executing query: SELECT S.NAME AS STUDENTNAME, C.NAME AS COURSENAME
FROM FN71923.STUDENTS S, FN71923.ATTENDS A, FN71923.COURSES C
WHERE S.EGN = A.STUDENTEGN AND A.COURSEYEAR = C.YEAR AND A.COURSENUM
Result output
-----
Cvetelina Kirilova, Fractals
Cvetelina Kirilova, Web technologies

```

фиг 2. Информация за записани курсове от студент по ЕГН на студента

```

Your choice: 3
Enter egn: 0000000007
Executing query: SELECT S.NAME AS
FROM FN71923.STUDENTS S
WHERE S.EGN = 0000000007
Result output
-----
Lachezar Peshterliev, 6204018147

```

фиг 3. Информация за ЕГН на наставника на даден студент

```
Your choice: 4
Enter egn: 6204018147
Executing query: SELECT *
FROM FN71923.TEACHER
WHERE teacherEGN = 6204018147

Result output

-----

Milko Takev, 6204018147, PHD, Analysis
```

фиг 4. Информация за преподавател по ЕГН

```
Your choice: 5
Executing query: SELECT * FROM FN71923.MENTOR

Result output

-----

Algebra, Asen Bojilov
Probability and statistics, Dontcho Dontchev
Geometry, Bogdan Aleksandrov
Analysis, Milko Takev
Computer informatics, Radoslava Hristova
Software technologies, Olga Georgieva
Differential Equations, Tsvetan Hristov
```

фиг 5. Информация за ръководителите на катедри



```
Your choice: 6
Enter name: Ivan
Enter egn (10 digits): 2901903290
Enter faculty number (5 digits): 32193
Enter major: Software Engineering
Enter course: 1
Enter mentor (10 digits): 6204018147
Successfully inserted!
```

фиг 6. Добавяне на студент

```
Your choice: 7
Enter name: Aleksandra Soskova
Enter egn (10 digits): 6629123909
Enter scientific title: PHD
Enter department: Algebra
Successfully inserted!
```

фиг 7. Добавяне на преподавател

```
Your choice: 9
Thank you for using our program

Process finished with exit code 0
```

фиг 8. Изход

## 8. Използвана литература

- *Ръководство по бази от данни - Радослава Христова и Димитър Димитров*
- *Упражнения по БД - практикум*