

Домашна работа по Бази от Данни

Условие:

Информационна система съхранява информация за кандидат-студентска кампания (КСК) на университет. Съхранява се информация за кандидат-студенти с данни за техния входящ номер — уникален за всеки кандидат-студент, име и фамилия на кандидат-студента. Съхранява се също информация и за изпити с данни за код на изпита — уникален за всеки изпит, име на изпита, продължителност на изпита и дата и час на провеждане на изпита. Един кандидат-студент може да заяви участие за много изпити и за един изпит може да са заявили участие много кандидат-студенти. Във всяко заявление на кандидат-студента, трябва да се пази и информация за дата на подаване на заявлението и сесията, на която кандидат студента се явява. В информационната система се съхраняват и данни за факултети с код на факултета — уникален за всеки факултет, име на факултет и адрес на факултета. Един факултет може да предлага много изпити, но всеки изпит се предлага точно от един факултет. За всеки кандидат-студент се пази и информация за оценка — подробно число от 2.0 до 6.0, което кандидат-студентът е получил за всеки изпит, на който се е явил (може да бъде NULL в случай, че кандидат-студентът не се е явил на изпит).

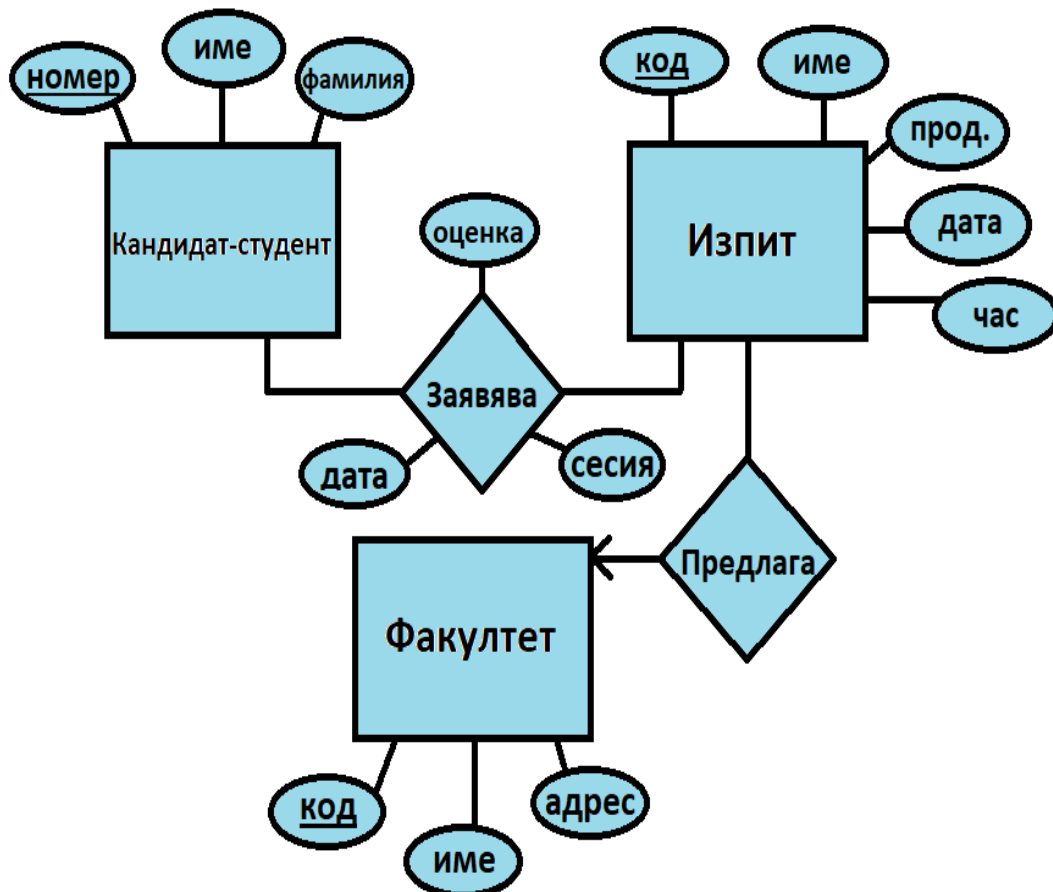
Зад 1. Да се направи E/R модел на БД, която съхранява гореописаната информация. Начертайте E/R диаграма на модела.

Зад 2. Преобразувайте E/R диаграмата към релационни схеми. Отбележете първичните и външни ключове.

Зад 3. Определете ФЗ за всяка релация от схемата на БД. Проверете дали полученият от вас модел е в ЗНФ, ако не е — нормализирайте го до ЗНФ.

Зад 4. Напишете SQL скрипт за създаване на моделиранта от вас БД — таблици и ограничения.

1. E/R модел на БД



2. Преобразуване към релационни схеми

- Кандидат-студент(номер,име,фамилия)
- Изпит(код на изпит,код на факултет,име,продължителност,дата,час)
- Факултет(код,име,адрес)
- Заявление(номер на студент,код на изпит,дата,сесия,оценка от изпит)

3. Функционални зависимости

- НомерС-> име , фамилия
- КодИ -> име,продължителност,час,дата,кодФ
- КодФ -> имеФ,адресФ
- НомерС,кодИ -> дата, сесия, озенка

Заклучение: Очевидно всички функционални зависимости съдържат в лявата си страна част от ключа на покриващото множество. Следователно релацията е в 3НФ и допълнителна нормализация не е необходима.

4. Скрипт за създаване на моделираната база от данни

```
SET SCHEMA FN45297;
```

- Create tables

```
CREATE TABLE CandidateStudent (  
  Id INT NOT NULL,  
  FirstName CHAR(30),  
  LastName CHAR(30)  
);
```

```
CREATE TABLE Faculty (  
  Id INT NOT NULL,  
  Name CHAR(30),  
  Address VARCHAR(255)  
);
```

```
CREATE TABLE Exam (  
  IdExam INT NOT NULL,  
  IdFaculty INT,  
  Name CHAR(30),  
  Duration Int ,  
  StartDare Date,  
  StartHour Time  
);
```

```
CREATE TABLE Request (  
  IdStudent INT NOT NULL,  
  IdExam INT NOT NULL,  
  DateSub Date,
```

```
Mark Decimal(4,3),  
Session Int);
```

- **Create Constraints**

```
ALTER TABLE CandidateStudent ADD CONSTRAINT PK_CandidateStudent PRIMARY KEY (Id);
```

```
ALTER TABLE Faculty ADD CONSTRAINT PK_Faculty PRIMARY KEY (Id);
```

```
ALTER TABLE Exam ADD CONSTRAINT PK_Exam PRIMARY KEY (IdExam);
```

```
ALTER TABLE Request ADD CONSTRAINT PK_Request PRIMARY KEY (IdStudent,IdExam);
```

```
ALTER TABLE Exam ADD CONSTRAINT FK_Exam FOREIGN KEY (IdFaculty)  
REFERENCES CandidateStudent(Id);
```

```
ALTER TABLE Request ADD CONSTRAINT FK_Request1 FOREIGN KEY (IdStudent)  
REFERENCES Faculty(Id);
```

```
ALTER TABLE Request ADD CONSTRAINT FK_Request2 FOREIGN KEY (IdExam)  
REFERENCES Exam(IdExam);
```

```
ALTER TABLE Request ADD CONSTRAINT FK_Request2 FOREIGN KEY (IdExam)  
REFERENCES Exam(IdExam);
```

```
ALTER TABLE Request ADD CONSTRAINT CHECK_Request CHECK (Mark in (2.00,6.00));
```