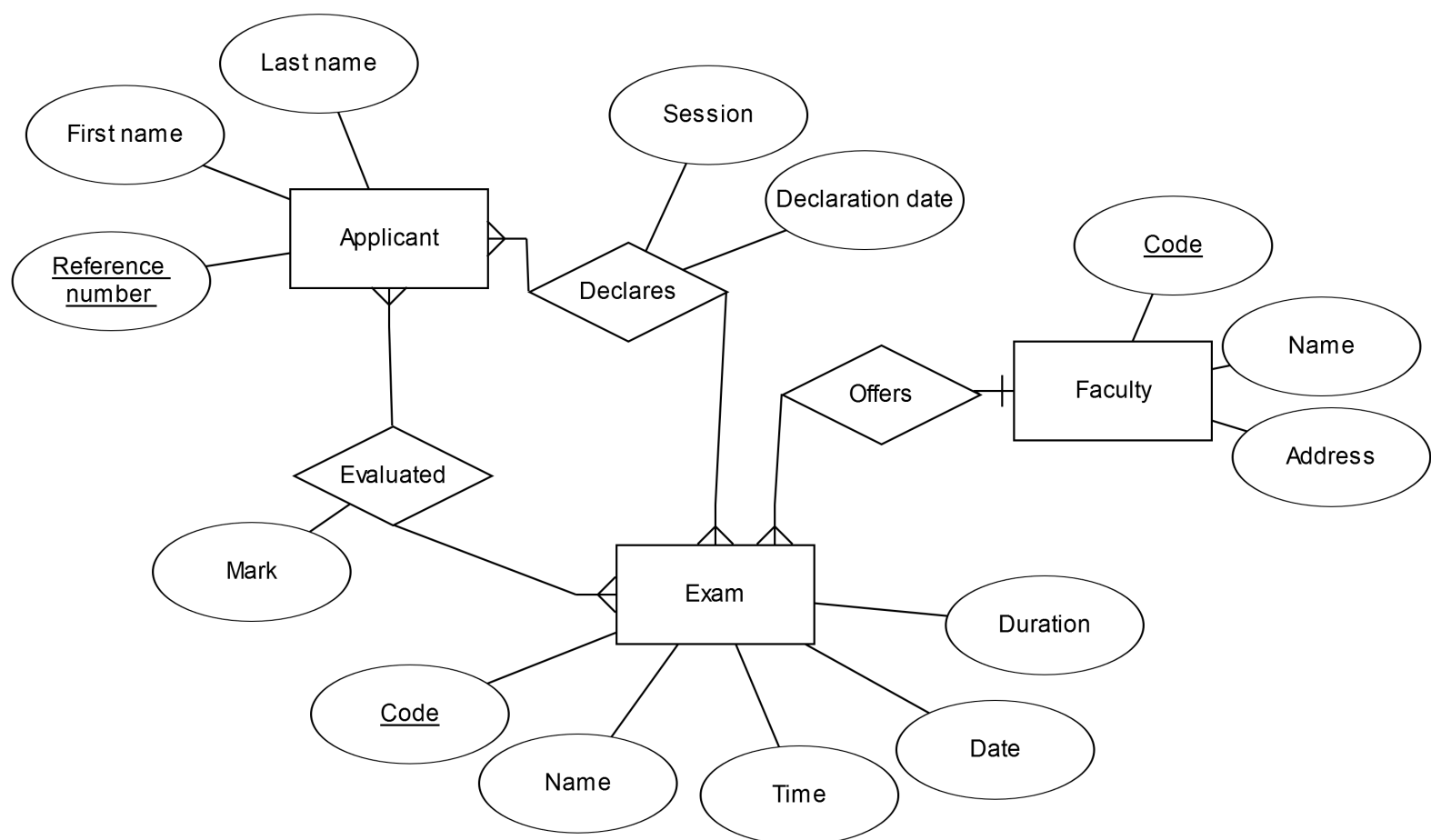


Бази данни

Домашна работа

Ростислав Стоянов
ф-н 45244

Зад.1 Е/Р модел представящ база от данни, съхраняваща информацията искана от условието на задачата, може да се илюстрира чрез Е/Р диаграмата показана на фиг.1. Приемаме, че един студент може да кандидатства за един изпит само по веднъж, т.е. за всеки изпит, на който кандидат-студентът се явява съществува само по едно заявление.



Фигура 1: Е/Р диаграма, описваща базата данни от условието

Зад.2 В следствие на преобразуването от Е/Р към релационен модел, се получават следните релационни схеми:

Applicant(refNumber, fName, lName),
Exam(examCode, name, time, date, duration),
Faculty(facultyCode, name, address),
Declares(applicantNumber, examCode, date, session),
Evaluated(applicantNumber, examCode, mark),
Offers(facultyCode, examCode).

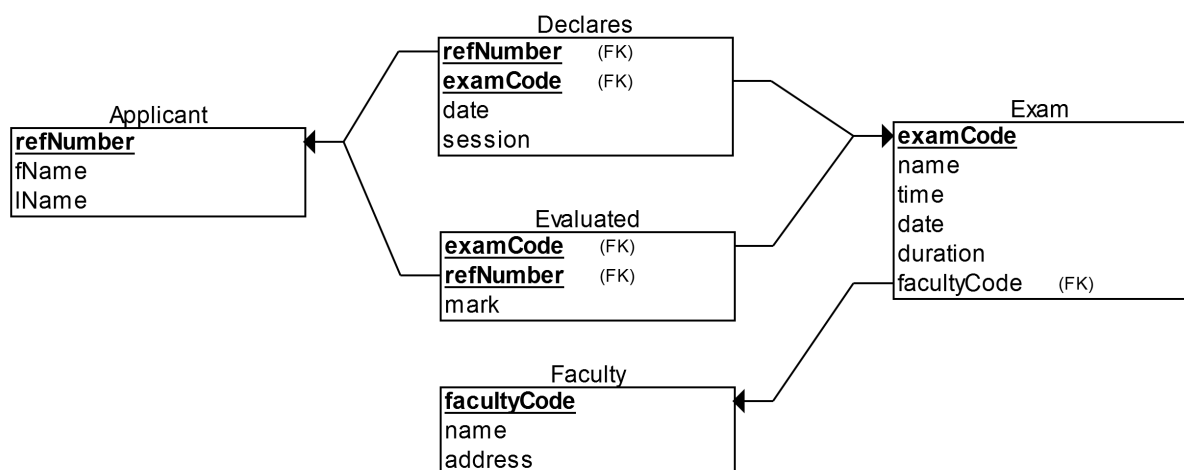
Поради факта, че Offers е от вида много-един, след оптимизация получаваме следния релационен модел:

Applicant(refNumber, fName, lName),
Exam(examCode, name, time, date, duration, facultyCode),
Faculty(facultyCode, name, address),
Declares(applicantNumber, examCode, date, session),
Evaluated(applicantNumber, examCode, mark).

Първичните ключове са:

{refNumber} за Applicant,
{examCode} за Exam,
{facultyCode} за Faculty,
{applicantNumber, examCode} за Declares,
{applicantNumber, examCode} за Evaluated.

Атрибутът applicantNumber в релациите Declares и Evaluated е външен ключ, свързващ ги с релацията Applicant, а атрибутът examCode в същите релации също е външен ключ, свързващ ги с релацията Exam. Външен ключ за Exam е facultyCode, който свързва релацията с Faculty.



Фигура 2: Релационен модел, описващ базата данни

Зад.3 Функционални зависимости:

1. Applicant:

- $refNumber \rightarrow fName, lName$
- $refNumber, fName \rightarrow lName$
- $refNumber, lName \rightarrow fName$

Всеки компонент в кортежите на релацията има атомарна стойност \rightarrow релацията е в 1НФ, но освен това и всеки атрибут е зависим от $refNumber$, който е единствения атрибут съставляващ първичния ключ \rightarrow релацията е в 2НФ. Тъй като и лявата част на всяка от функционалните зависимости (първата функционална зависимост може да се раздели на две ФЗ) е суперключ за релацията и тя е в 2НФ, то релацията е в 3НФ.

2. Faculty:

- $facultyCode \rightarrow name, address$
- $facultyCode, name \rightarrow address$
- $facultyCode, address \rightarrow name$

Аналогично тук също се извършват проверки (по подобие на извършените при релацията Applicant). Установява се, че и тази релация е в 3НФ.

3. Exam:

- $examCode \rightarrow name, time, date, duration, facultyCode$
- $examCode, name \rightarrow name, time, date, duration, facultyCode$
- $examCode, facultyCode \rightarrow name, time, date, duration$
- $examCode, facultyCode, name \rightarrow time, date, duration$
- $name, time, date, duration, facultyCode \rightarrow examCode$

Атомарната стойност на атрибутите и факта, че ключът на релацията се състои само от $examCode$, ни водят до 2НФ. За всички ФЗ, без последната, лявата част е суперключ, а за последната ФЗ е изпълнено, че дясната част е част (в частност целия) от ключа, т.е. релацията е в 3НФ.

4. Declares:

- $refNumber, examCode \rightarrow date, session$
- $refNumber, examCode, date \rightarrow session$
- $refNumber, examCode, session \rightarrow date$

Атомарност \rightarrow 1НФ; $\{refNumber, examCode\}$ - ключ, всички атрибути зависят от него, но не и от негово подмножество \rightarrow 2НФ. Лесно се проверява, че релацията е и в 3НФ.

5. Evaluated:

- $refNumber, examCode \rightarrow mark$

Очевидно релацията е в 1НФ и тъй като {refNumber,examCode} е ключ, а атрибутите поотделно не са ключ за релацията, то тя е във 2НФ. Оттук, тривиално релацията е и в 3НФ.

Зад.4

```
1  /*Create tables*/
2
3  create table Applicant (
4  refNumber INT NOT NULL,
5  fName CHAR(30),
6  lName CHAR(30)
7  );
8
9  create table Faculty (
10 facultyCode INT NOT NULL,
11 name CHAR(50),
12 address VARCHAR(255)
13 );
14
15 create table Exam(
16 examCode INT NOT NULL,
17 name CHAR(50),
18 time TIME,
19 date DATE,
20 duration INT,
21 facultyCode INT
22 );
23
24 create table Declares(
25 refNumber INT NOT NULL,
26 examCode INT NOT NULL,
27 date DATE,
28 session CHAR(30)
29 );
30
31 create table Evaluated(
32 refNumber INT NOT NULL,
33 examCode INT NOT NULL,
34 mark ENUM(2,3,4,5,6)
35 );
36
37 /*Create constraints*/
38 alter table Applicant add constraint PK_Applicant PRIMARY KEY(
    refNumber);
39 alter table Faculty add constraint PK_Faculty PRIMARY KEY(
    facultyCode);
```

```
40 alter table Exam add constraint PK_Exam PRIMARY KEY(examCode);
41 alter table Declares add constraint PK_Declares PRIMARY KEY(
    refNumber,examCode);
42 alter table Evaluated add constraint PK_Evaluated PRIMARY KEY(
    refNumber,examCode);
43
44 alter table Exam add constraint FK_Exam_Faculty FOREIGN KEY(
    facultyCode) references Faculty(facultyCode);
45 alter table Declares add constraint FK_Declares_Applicant FOREIGN
    KEY(refNumber) references Applicant(refNumber);
46 alter table Declares add constraint FK_Declares_Exam FOREIGN KEY
    (examCode) references Exam (examCode);
47 alter table Evaluated add constraint FK_Evaluated_Applicant
    FOREIGN KEY(refNumber) references Applicant(refNumber);
48 alter table Evaluated add constraint FK_Evaluated_Exam FOREIGN
    KEY (examCode) references Exam (examCode);
```