

СОФИЙСКИ УНИВЕРСИТЕТ "СВ. КЛИМЕНТ ОХРИДСКИ"
ФАКУЛТЕТ ПО МАТЕМАТИКА И ИНФОРМАТИКА

Домашна работа

ПО

БАЗИ ОТ ДАННИ

СПЕЦ. ИНФОРМАТИКА, 3 КУРС, ЛЕТЕН СЕМЕСТЪР,

УЧЕБНА ГОДИНА 2018/19

15 април 2019 г.

Изготвил:

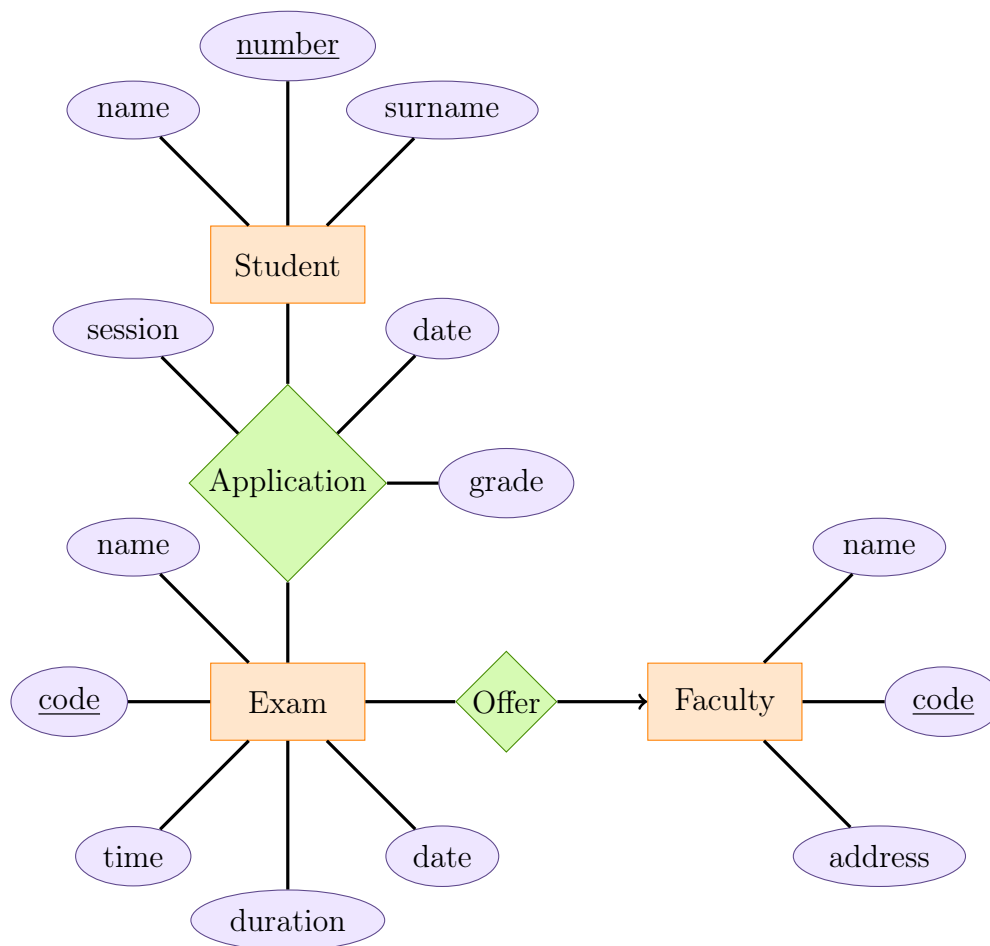
София

Иво Алексеев Стратев

Фак. номер: 45342

Група: 3

1 Е/Р диаграма



2 Преобразуване на Е/Р диаграмата до релационни схеми

Student(number, name, surname)
Exam(code, name, date, time, duration, faculty)
Faculty(code, name, address)
Application(number, code, date, session, grade)

2.1 Ограничения по референтна цялост

$$\begin{aligned}\pi_{faculty}(Exam) &\subseteq \pi_{code}(Faculty) \\ \pi_{number}(Application) &\subseteq \pi_{number}(Student) \\ \pi_{code}(Application) &\subseteq \pi_{code}(Exam)\end{aligned}$$

3 Функционални зависимости

3.1 За Student

Има единствена нетривиална функционална зависимост $number \rightarrow name, surname$. $number$ е първичния ключ на релацията и следователно релацията $Student$ е в НФБК, в частност е и в 3НФ.

3.2 За Exam

Има единствена нетривиална функционална зависимост $code \rightarrow name, date, time, duration$. $date, time, duration$ дори заедно не могат да породат нетривиални ФЗ, понеже в един ден, по едно и също време може да има повече от един изпит с еднаква продължителност. $name$ пък не може да породи нетривиална ФЗ, понеже е напълно възможно изпит с едно и също време да се проведе повече от веднъж. $code$ е първичния ключ на релацията и следователно релацията $Exam$ е в НФБК, в частност е и в 3НФ.

3.3 За Faculty

Нетривиалните функционални зависимости са:

$$\begin{aligned}code &\rightarrow name, address \\ name &\rightarrow code, address\end{aligned}$$

Понеже и $\{code\}$ и $\{name\}$ са кандидат ключове (в частност суперключове), то релацията е в НФБК, в частност е и в 3НФ.

3.4 За Application

Единствената нетривиална ФЗ е $number, code \rightarrow date, session, grade$ и понеже $\{number, code\}$ е първичен ключ, то $Application$ е в НФБК, в частност е и в 3НФ.

4 SQL код за създаване на Базата данни - таблици и ограничения

```
CREATE TABLE Student (  
    number INT NOT NULL,  
    name VARCHAR(30) NOT NULL,  
    surname VARCHAR(30) NOT NULL,  
    PRIMARY KEY (number)  
);  
  
CREATE TABLE Faculty (  
    code INT NOT NULL,  
    name VARCHAR(120) NOT NULL,  
    address VARCHAR(256) NOT NULL,  
    PRIMARY KEY (code)  
);  
  
CREATE TABLE Exam (  
    code INT NOT NULL,  
    name VARCHAR(60) NOT NULL,  
    date DATE NOT NULL,  
    time TIME NOT NULL,  
    duration SMALLINT NOT NULL,  
    faculty INT NOT NULL,  
    PRIMARY KEY (code),  
    FOREIGN KEY (faculty) REFERENCES Faculty(code)  
);  
  
CREATE TABLE Application (  
    number INT NOT NULL,  
    code INT NOT NULL,  
    date DATE NOT NULL,  
    session TINYINT NOT NULL,  
    grade FLOAT,  
    PRIMARY KEY (number, code),  
    FOREIGN KEY (number) REFERENCES Student(number),  
    FOREIGN KEY (code) REFERENCES Exam(code)  
);  
  
DELIMITER $$
```

```

CREATE TRIGGER Before_Insert_Application
  BEFORE INSERT ON Application FOR EACH ROW
  BEGIN
    IF NEW.grade < 2.0 OR NEW.grade > 6.0
    THEN SIGNAL SQLSTATE '45000'
      SET
        MESSAGE_TEXT = 'grade should be in interval [2.0, 6.0]',
        TABLE_NAME = 'Application',
        COLUMN_NAME = 'grade';
    END IF;
  END;

```

\$\$

```

CREATE TRIGGER Before_Update_Application
  BEFORE UPDATE ON Application FOR EACH ROW
  BEGIN
    IF NEW.grade < 2.0 OR NEW.grade > 6.0
    THEN SIGNAL SQLSTATE '45000'
      SET
        MESSAGE_TEXT = 'grade should be in interval [2.0, 6.0]',
        TABLE_NAME = 'Application',
        COLUMN_NAME = 'grade';
    END IF;
  END;

```

\$\$

```

DELIMITER ;

```

MySQL игнорира ограничения от тип CHECK. За това създаваме тригери, които се изпълняват преди добавяне и обновяване и служат за валидация, че стойността на grade ще е в интервала [2.0, 6.0].