

Домашнее задание 4

Отношение с атрибутами:

StudentId, StudentName, GroupId, GroupName, CourseId, CourseName, LecturerId, LecturerName, Mark.

Функциональные зависимости в данном отношении:

- *StudentId* → *StudentName*
- *GroupId* → *GroupName*
- *CourseId* → *CourseName*
- *LecturerId* → *LecturerName*
- *StudentId* → *GroupId*
- *GroupId, CourseId* → *LecturerId*
- *StudentId, CourseId* → *Mark*

Приведение к 5НФ

1. 1НФ

Отношение находится в 1НФ:

- В нем нет повторяющихся групп
- Атрибуты атомарны
- Ключ отношения {*StudentId, CourseId*} (доказано в домашнем задании 3)

2. 2НФ

Мы имеем 5 ФЗ который зависят от части ключа {*StudentId, CourseId*}

- *StudentId* → *StudentName*
- *GroupId* → *GroupName*
- *CourseId* → *CourseName*
- *LecturerId* → *LecturerName*
- *StudentId* → *GroupId*

После декомпозиции получим:

1. *StudentId*, *StudentName*
2. *GroupId*, *GroupName*
3. *CourseId*, *CourseName*
4. *LecturerId*, *LecturerName*
5. *StudentId*, *GroupId*
6. *StudentId, CourseId*, *GroupId, LecturerId, Mark*

Каждое отношение сохраняет свойство соответствующего ФЗ

3. 3НФ

Имеем транзитивную зависимость в отношении 6

$$StudentId \rightarrow GroupId; CourseId, GroupId \rightarrow LecturerId$$

Декомпозируем по последнему ФЗ

$$\pi_{StudentId, GroupId, CourseId, LecturerId, Mark}(R) = \pi_{CourseId, GroupId, LecturerId}(R) \wedge \pi_{StudentId, GroupId}(R) \wedge \pi_{StudentId, CourseId, Mark}(R)$$

После декомпозиции получим:

1. StudentId, StudentName
2. GroupId, GroupName
3. CourseId, CourseName
4. LecturerId, LecturerName
5. StudentId, GroupId
6. StudentId, CourseId, Mark
7. CourseId, GroupId, LecturerId

4. НФБК

Отношения находятся в НФБК, потому что все зависимости уз двух атрибутов уже находятся в НФБК, отношениям 6 и 7 соответствуют ФЗ и следовательно у них левые части являются надключом.

5. 4НФ

Проверим что отношения находятся в 4 НФ. Все отношения из 2 атрибутов уже находятся в 4НФ. Проверим отношения 6 и 7.

Имеем два отношения StudentId, CourseId, Mark и CourseId, GroupId, LecturerId
Рассмотрим одно произвольное без потери общности.

- $\forall X, M, Z. (X \rightarrow Y | Z \Rightarrow \forall A. X \rightarrow A)$ верно для данного отношения, поскольку в МЗ является ФЗ и её ключ StudentId, CourseId
- $\forall X, M, Z. (X \rightarrow Y | Z \Rightarrow X - \text{надключ})$ выполняется
- МЗ является ФЗ и отношение в НФБК

Аналогично для второго отношения. Следовательно, отношения уже в 4НФ.

5. 5НФ

Проверим что отношения уже в 5НФ

Отношения из двух атрибутов $A \rightarrow B$ уже в 5НФ:

- Имеет одну ФЗ которая МЗ
- $A \rightarrow B | \emptyset$ тогда по т. Фейгина $\{AB, A\}$ – единственная зависимость соединения при этом каждый её элемент – надключ.

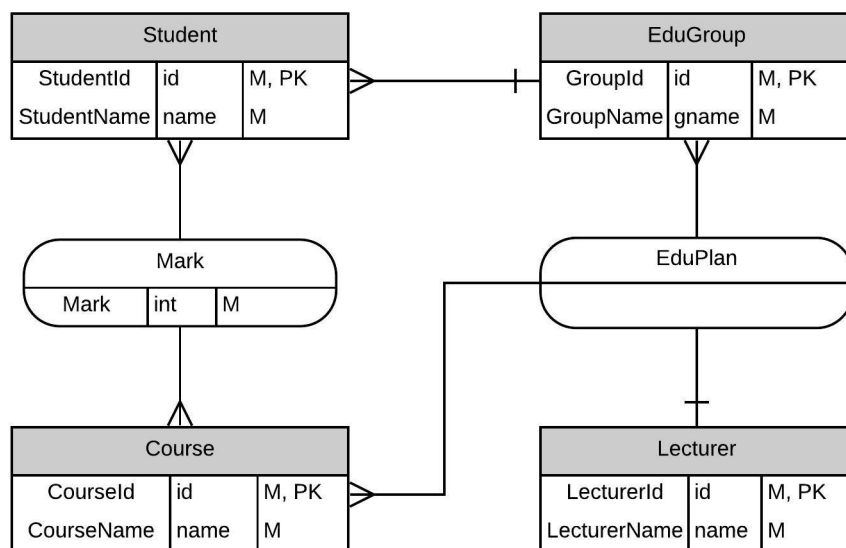
Рассмотрим отношения $\underline{StudentId}, \underline{CourseId}, Mark$ и $\underline{CourseId}, \underline{GroupId}, \underline{LecturerId}$ вида $AB \rightarrow C: AB \rightarrow C | \emptyset$ по т. Фейгина $\{ABC, AB\}$, каждый элемент – надключ.

Следовательно, отношения уже в 5НФ.

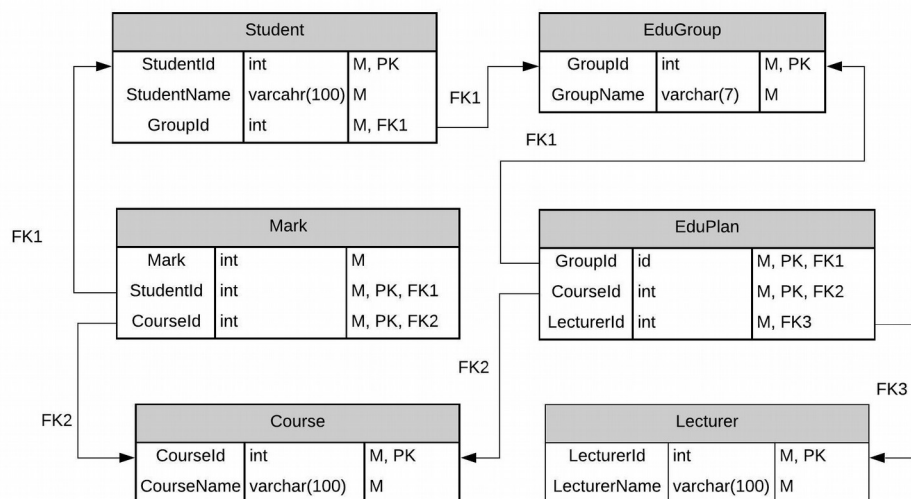
6. Итог

1. $\underline{StudentId}, StudentName$
2. $\underline{GroupId}, GroupName$
3. $\underline{CourseId}, CourseName$
4. $\underline{LecturerId}, LecturerName$
5. $\underline{StudentId}, \underline{GroupId}$
6. $\underline{StudentId}, \underline{CourseId}, Mark$
7. $\underline{CourseId}, \underline{GroupId}, \underline{LecturerId}$

Модель сущность-связь



Физическая модель



Создание

```
create table EduGroup (  
    GroupId int not null,  
    GroupName varchar(7) not null,  
    primary key (GroupId)  
);  
  
create table Student (  
    StudentId int not null,  
    StudentName varchar(100) not null,  
    GroupId int not null,  
    primary key (StudentId),  
    foreign key (GroupId) references EduGroup (GroupId)  
);  
  
create table Course (  
    CourseId int not null,  
    CourseName varchar(100),  
    primary key (CourseId)  
);  
  
create table Mark (  
    StudentId int not null,  
    CourseId int not null,  
    Mark int not null,  
    primary key (StudentId, CourseId),  
    foreign key (StudentId) references Student (StudentId),  
    foreign key (CourseId) references Course (CourseId)  
);  
  
create table Lecturer (  
    LecturerId int not null,  
    LecturerName varchar(100) not null,  
    primary key (LecturerId)  
);  
  
create table EduPlan (  
    GroupId int not null,  
    CourseId int not null,  
    LecturerId int not null,  
    primary key (GroupId, CourseId),  
    foreign key (GroupId) references EduGroup (GroupId),  
    foreign key (CourseId) references Course (CourseId),  
    foreign key (LecturerId) references Lecturer (LecturerId)  
);
```

Заполнение

```
insert into EduGroup  
    (GroupId, GroupName) values  
    (1, 'М3437'),  
    (2, 'М3435');  
  
insert into Student  
    (StudentId, StudentName, GroupId) values  
    (1, 'Владислав Сайфулин', 2),  
    (2, 'Егор Тимошенко', 1),  
    (3, 'Александр Волков', 2);  
  
insert into Course  
    (CourseId, CourseName) values  
    (1, 'Базы данных'),  
    (2, 'Алгоритмы и структуры данных');  
  
insert into Mark  
    (StudentId, CourseId, Mark) values  
    (1, 2, 95),  
    (3, 1, 80),  
    (2, 2, 60),  
    (1, 1, 80);  
  
insert into Lecturer  
    (LecturerId, LecturerName) values  
    (1, 'Георгий Корнеев'),  
    (2, 'Павел Маврин');  
  
insert into EduPlan  
    (GroupId, CourseId, LecturerId) values  
    (1, 1, 1),  
    (1, 2, 2),  
    (2, 1, 1),  
    (2, 2, 2);
```