

1. Функциональные зависимости

(Функциональные зависимости должны быть заданы по одной на строке в формате «A, B -> D, E». Названия атрибутов должны быть как в условии ДЗ.)

- StudentId -> StudentName, GroupId
- GroupId -> GroupName
- CourseId -> CourseName
- LecturerId -> LecturerName
- GroupId, CourseId -> LecturerId
- StudentId, CourseId, LecturerId -> Mark

2. Ключи

2.1 Процесс определения ключей

Возьмём в качестве надключа множество всех атрибутов и будем пытаться его минимизировать, удаляя атрибуты и проверяя, остался ли он надключом. Как только после какого-то этапа мы не сможем удалить ни один атрибут, мы получим минимальный по включению надключ, он и будет являться ключом по определению.

2.2 Полученные ключи

{StudentId, CourseId}

3. Замыкания множества атрибутов

(Последовательность построения замыкания, по одному множеству на строке. В последней строке должно быть указано само замыкание.)

3.a. GroupId, CourseId

{GroupId, CourseId}

{GroupId, CourseId, GroupName}

{GroupId, CourseId, GroupName, CourseName}

{GroupId, CourseId, GroupName, CourseName, LecturerId}

{GroupId, CourseId, GroupName, CourseName, LecturerId, LecturerName}

{GroupId, CourseId, GroupName, CourseName, LecturerId, LecturerName} = {GroupId, CourseId}+

3.b. StudentId, CourseId

{StudentId, CourseId}

{StudentId, CourseId, StudentName}

{StudentId, CourseId, StudentName, GroupId}

{StudentId, CourseId, StudentName, GroupId, GroupName}

{StudentId, CourseId, StudentName, GroupId, GroupName, CourseName}

{StudentId, CourseId, StudentName, GroupId, GroupName, CourseName, LecturerId}

{StudentId, CourseId, StudentName, GroupId, GroupName, CourseName, LecturerId, Mark}

{StudentId, CourseId, StudentName, GroupId, GroupName, CourseName, LecturerId, Mark, LecturerName}

{StudentId, CourseId, StudentName, GroupId, GroupName, CourseName, LecturerId, Mark, LecturerName}
= {StudentId, CourseId}+

3.c. StudentId, LecturerId

{StudentId, LecturerId}

{StudentId, LecturerId, StudentName}

{StudentId, LecturerId, StudentName, GroupId}

{StudentId, LecturerId, StudentName, GroupId, GroupName}

{StudentId, LecturerId, StudentName, GroupId, GroupName, LecturerName}

{StudentId, LecturerId, StudentName, GroupId, GroupName, LecturerName} = {StudentId, LecturerId}+

4. Неприводимое множество функциональных зависимостей

4.1d. Первый этап

Сделаем так, чтобы каждая правая часть ФЗ содержала ровно один атрибут (по правилу расщепления).

4.1r. Результаты первого этапа

(В формате из пункта 1)

- StudentId -> StudentName
- StudentId -> GroupId
- GroupId -> GroupName
- CourseId -> CourseName
- LecturerId -> LecturerName
- GroupId, CourseId -> LecturerId
- StudentId, CourseId, LecturerId -> Mark

4.2d. Второй этап

Добьемся, чтобы каждая левая часть ФЗ стала минимальной по включению. Пытаемся удалить по одному атрибуту из левой части каждого правила, повторяем до тех пор, пока не смогли удалить ни одного.

Получилось удалить LecturerId из последнего правила, т. к. LecturerId содержится в замыкании $A = \{StudentId, CourseId\}$ над S .

4.2r. Результаты второго этапа

- StudentId -> StudentName
- StudentId -> GroupId
- GroupId -> GroupName
- CourseId -> CourseName
- LecturerId -> LecturerName
- GroupId, CourseId -> LecturerId
- StudentId, CourseId -> Mark

4.3d. Третий этап

Пытаемся удалить правила из множества ФЗ. Удаляем таким образом: возьмем какое-то правило (A -> B), выкинем его из S (получим S') и проверим, содержится ли B в замыкании A над S'. Если да, то удалим это правило, иначе удалять нельзя.

В нашем случае ни одного правила удалить нельзя.

4.3r. Результаты третьего этапа

- StudentId -> StudentName
- StudentId -> GroupId
- GroupId -> GroupName
- CourseId -> CourseName
- LecturerId -> LecturerName
- GroupId, CourseId -> LecturerId
- StudentId, CourseId -> Mark

Пример данных

StudentId	StudentName	GroupId	GroupName	CourseId	CourseName	LecturerId	LecturerName	Mark
1	Комаров	2	M3337	17	DB	2	Георгий Корнеев	B
1	Комаров	2	M3337	1	Диффуры	1	Басов	D
2	Козырев	2	M3337	17	DB	2	Георгий Корнеев	A
3	Хала	666	M3939	1	Диффуры	1	Басов	F