

Базы данных

Нормализация

Белова О.П., Казнин А.А.

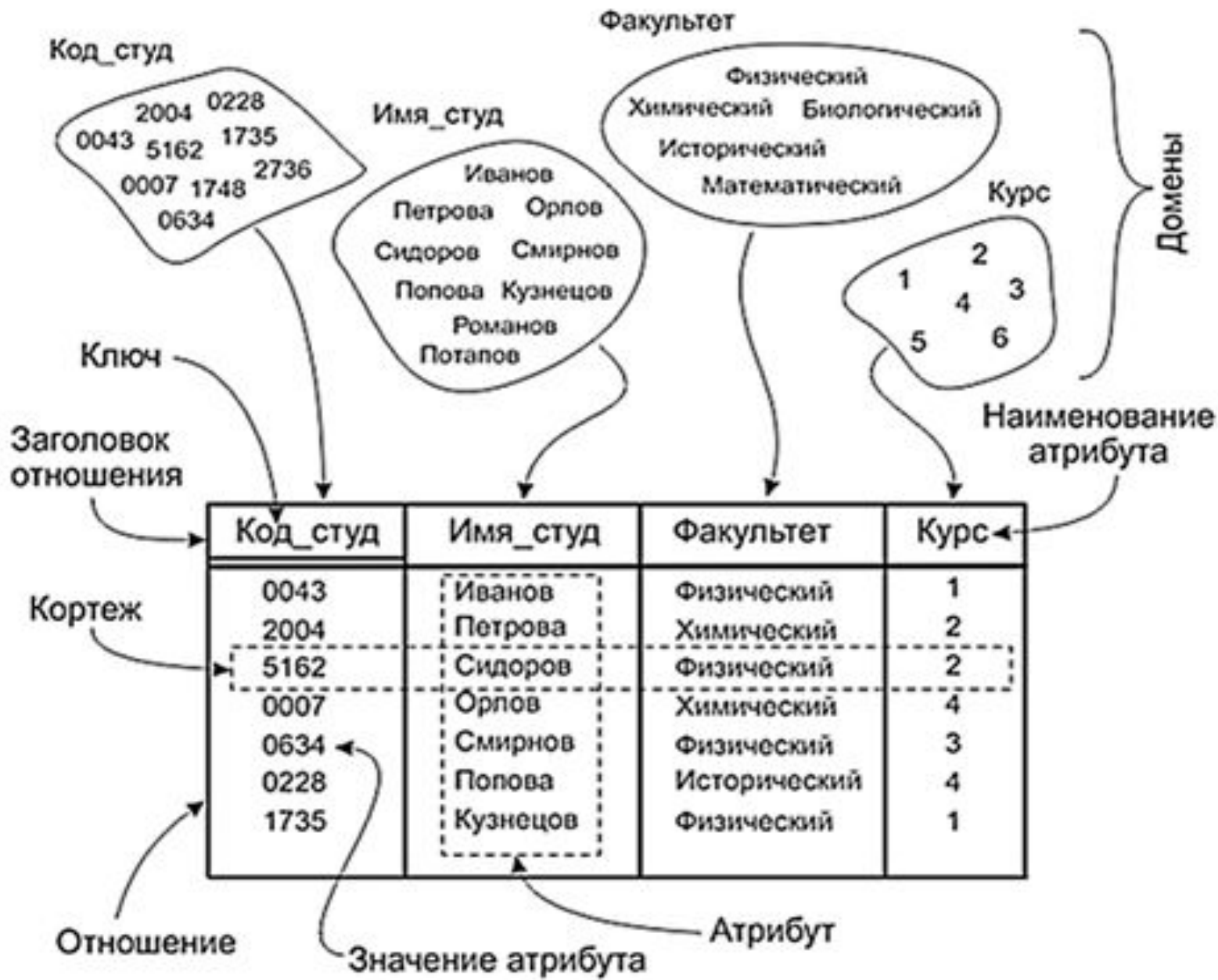
ИСТОЧНИКИ

<http://www.intuit.ru/studies/courses/508/364/info> -
базы данных, концептуальное
проектирование

<https://habrahabr.ru/post/193284/> -
проектирование БД, ключи

<http://www.intuit.ru/studies/courses/1095/191/info>
- основы проектирования реляционных БД

<https://habrahabr.ru/post/254773/> -
нормализация БД



Свойства реляционных БД

- каждый элемент таблицы соответствует одному элементу данных (атомарны и не делимы)
- все столбцы в таблице однородные, т.е. все элементы в столбце имеют одинаковый тип и длину
- каждый столбец имеет уникальное имя
- одинаковые строки в таблице отсутствуют
- порядок следования строк и столбцов может быть произвольным

ERD

ER-диаграмма (*entity – relationship*) – диаграмма сущность-связь

Сущность (Entity) или объект – то, о чем будет накапливаться информация в информационной системе (нечто такое, за чем пользователь хотел бы наблюдать)

Атрибут – поименованное свойство (характеристика) сущности

Экземпляр сущности (запись, кортеж)- это конкретный представитель данной сущности

Ключи

Первичный ключ – это значение, которое уникально для каждой записи в таблице

Номер заказа при покупке в интернет-магазине может быть первичным ключом таблицы заказов в БД этого магазина, т.к. он является уникальным значением

Первичный ключ служит для идентификации записей

first_name	last_name
vasya	pupkin
vasya	pupkin

Id	first_name	last_name
1	vasya	pupkin
2	vasya	pupkin

Ключи

Обычно первичный ключ – числовое значение.

Часто первичный ключ состоит из одного поля, но может быть и комбинацией нескольких столбцов. Но нужно помнить, что первичный ключ всегда уникален, а значит нужно, чтобы комбинация n-го количества полей, например, 2-х, была уникальна.

Поле первичного ключа часто, но не всегда, обрабатывается самой БД. Использование автоинкрементных ключей – хороший способ для задания уникальных первичных ключей. Классическое название такого ключа – суррогатный первичный ключ. Такой ключ не содержит полезной информации, относящейся к сущности (объекту), информация о которой хранится в таблице, поэтому он и называется суррогатным.

Внешние ключи

Проектирование базы данных – это решение двух вопросов:

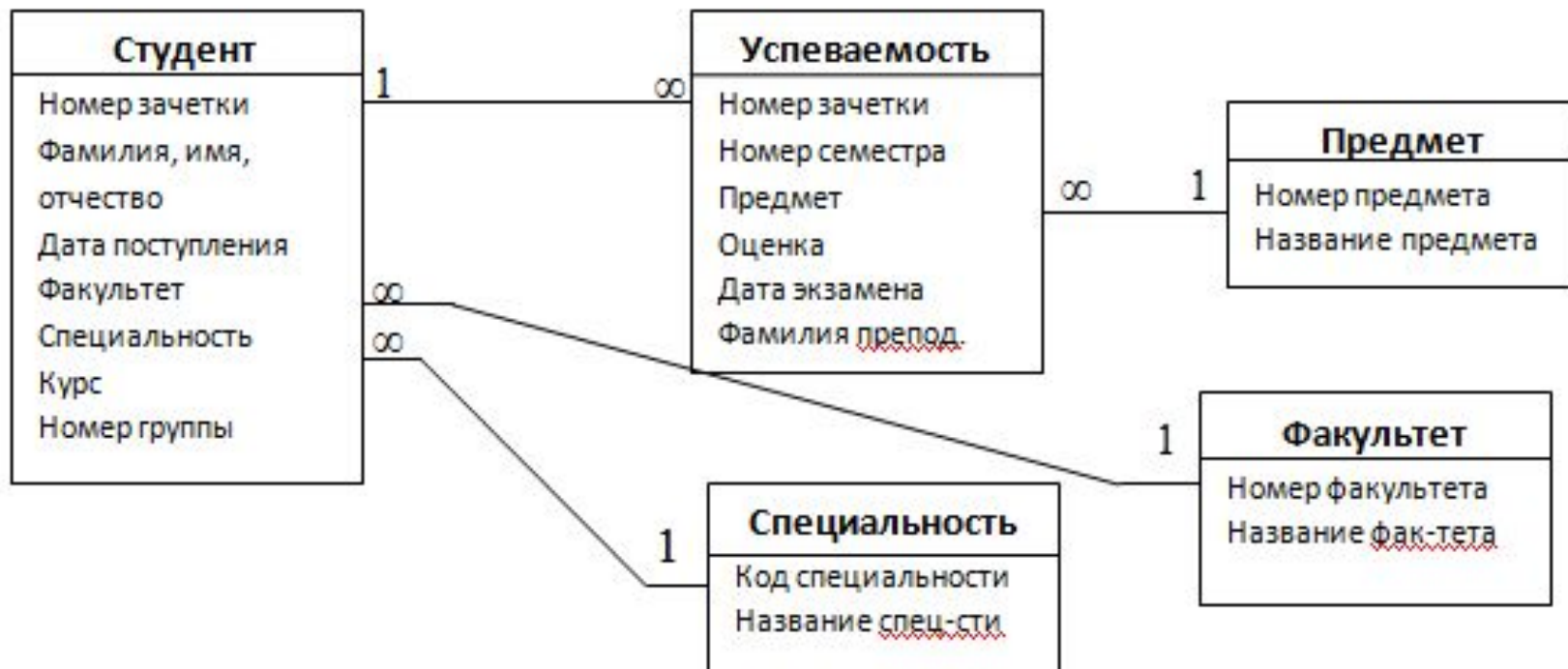
- определение того, какие сущности вы хотите хранить в ней
- какие связи между этими сущностями существуют

Если один или несколько столбцов, в которых находится первичный ключ для одной таблицы, упоминается в одном или нескольких столбцах другой таблицы, то в ссылке внешнего ключа создается связь между двумя таблицами. Этот столбец становится **внешним ключом** во второй таблице.

Типы связей

- **один-к-одному** - один экземпляр первой сущности связан с одним экземпляром второй сущности. Чаще всего свидетельствует о том, что на самом деле мы имеем всего одну сущность, неправильно разделенную на две
- **мноее-ко-многим** - каждый экземпляр первой сущности может быть связан с несколькими экземплярами второй сущности, и каждый экземпляр второй сущности может быть связан с несколькими экземплярами первой сущности. Тип связи **много-ко-многим** является временным типом связи, допустимым на ранних этапах разработки модели. В дальнейшем этот тип связи заменяют двумя связями типа **один-ко-многим** путем создания промежуточной сущности
- **один-ко-многим** - один экземпляр первой сущности связан с несколькими экземплярами второй сущности. Наиболее часто используемый тип связи. Левая сущность (со стороны «один») называется **родительской**, правая (со стороны «много») — **дочерней**





Нормализация

Избыточная функциональная зависимость – включает в себя информацию, которая может быть получена на основе других зависимостей, имеющих в БД

Корректной считается БД, в которой отсутствуют избыточные функциональные зависимости. В противном случае приходится прибегать к процедуре декомпозиции (разложения) имеющегося множества отношений

Нормализация - разделение одной таблицы базы данных на несколько таблиц

Цель: исключить избыточное дублирование данных, которое является причиной аномалий, возникших при добавлении, редактировании и удалении кортежей (строк таблицы)

Аномалии

Ситуации в таблице БД, которые приводят к противоречию в БД либо существенно усложняют обработку БД. Причиной является излишнее дублирование данных в таблице, которое вызывается наличием функциональных зависимостей от не ключевых атрибутов.

Аномалии-модификации - изменение одних данных может повлечь просмотр всей таблицы и соответствующее изменение некоторых записей таблицы.

Аномалии-удаления - при удалении какого либо кортежа из таблицы может пропасть информация, которая не связана напрямую с удаляемой записью.

Аномалии-добавления - информацию в таблицу нельзя поместить, пока она не полная, либо вставка записи требует дополнительного просмотра таблицы.

Шесть форм нормализации

Нормальная форма — требование, предъявляемое к структуре таблиц в теории реляционных БД для устранения из базы избыточных функциональных зависимостей между атрибутами (полями таблиц)

- первая нормальная форма (First Normal Form — 1NF)
- вторая нормальная форма (Second Normal Form — 2NF)
- третья нормальная форма (Third Normal Form — 3NF)
- нормальная форма Бойса—Кодда (Boyce—Codd Normal Form — BCNF)
- четвертая нормальная форма (Fourth Normal Form — 4NF)
- пятая нормальная форма, или нормальная форма проекции-соединения (Fifth Normal Form — 5NF, или PJ/NF)
- шестая нормальная форма (6NF)

Функциональная зависимость

Атрибут Y некоторого отношения функционально зависит от X (атрибуты могут быть составными), если в любой момент времени каждому значению X соответствует одно значение Y . Функциональная зависимость обозначается $X \rightarrow Y$.

Пример

Известно, что:

- каждому рейсу соответствует определенное время вылета
- для каждого пилота, даты и времени вылета возможен только один рейс
- на определенный день и рейс назначается определенный пилот

Следовательно:

"Рейс" -> "Время_{} вылета" ;

{"Пилот", "Дата_вылета", "Время_вылета"} -> "Рейс" ;

{"Рейс", "Дата_вылета"}: {"Рейс", "Дата_вылета"} -> "Пилот".

Полная функциональная зависимость

Неключевой атрибут функционально полно зависит от составного ключа, если он функционально зависит от всего ключа в целом, но не находится в функциональной зависимости от какого-либо из входящих в него атрибутов

Многозначная функциональная зависимость

Атрибут А многозначно определяет атрибут В, если для каждого значения А существует «хорошо определенное множество» соответствующих значений атрибута В. Например, если рассматривать таблицу успеваемости учащихся в школе, включающую в себя поля «Предмет» (поле А) и «Оценка» (поле В), то атрибут В имеет «хорошо определенное множество» допустимых значений: 1, 2, 3, 4, 5

Транзитивная функциональная зависимость

Пусть X, Y, Z - три атрибута некоторого отношения. При этом $X \rightarrow Y$ и $Y \rightarrow Z$, но обратное соответствие отсутствует, т.е. $Z \not\rightarrow Y$ и $Y \not\rightarrow X$. Тогда Z транзитивно зависит от X

Взаимная независимость

Два или более атрибута взаимно независимы, если ни один из этих атрибутов не является функционально зависимым от других

Первая нормальная форма

Таблица находится в первой нормальной форме тогда и только тогда, когда ни одно из полей не содержит более одного значения и любое ключевое поле не пусто.

Любая таблица в реляционной базе данных автоматически находится в первой нормальной форме, иное невозможно по определению. В такой таблице не должно содержаться полей (признаков), которые можно было бы разделить на несколько полей (признаков).

1HΦ

Фирма	Модели
BMW	M5, X5M, M1
Nissan	GT-R



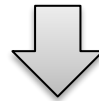
Фирма	Модели
BMW	M5
BMW	X5M
BMW	M1
Nissan	GT-R

Вторая нормальная форма

Отношение находится во 2НФ, если оно находится в 1НФ и каждый не ключевой атрибут неприводимо зависит от Первичного Ключа(ПК).

Неприводимость означает, что в составе потенциального ключа отсутствует меньшее подмножество атрибутов, от которого можно также вывести данную функциональную зависимость.

<u>Модель</u>	<u>Фирма</u>	Цена	Скидка
M5	BMW	5500000	5%
X5M	BMW	6000000	5%
M1	BMW	2500000	5%
GT-R	Nissan	5000000	10%



<u>Модель</u>	<u>Фирма</u>	Цена
M5	BMW	5500000
X5M	BMW	6000000
M1	BMW	2500000
GT-R	Nissan	5000000

<u>Фирма</u>	Скидка
BMW	5%
Nissan	10%

Третья нормальная форма

Отношение находится в 3НФ, когда находится во 2НФ и каждый не ключевой атрибут нетранзитивно зависит от первичного ключа. Требуется выносить все не ключевые поля, содержимое которых может относиться к нескольким записям таблицы в отдельные таблицы.

Транзитивная ФЗ

Пусть X, Y, Z - три атрибута некоторого отношения. При этом $X \rightarrow Y$ и $Y \rightarrow Z$, но обратное соответствие отсутствует, т.е. $Z \not\rightarrow Y$ и $Y \not\rightarrow X$. Тогда Z транзитивно зависит от X .

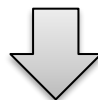
<u>Модель</u>	Магазин	Телефон
BMW	Риал-авто	87-33-98
Audi	Риал-авто	87-33-98
Nissan	Некст-Авто	94-54-12

Личных телефонов у автомобилей нет, телефон зависит исключительно от магазина.

Модель \rightarrow Магазин, Магазин \rightarrow Телефон, Модель \rightarrow Телефон.

Зависимость Модель \rightarrow Телефон является транзитивной

<u>Модель</u>	Магазин	Телефон
BMW	Риал-авто	87-33-98
Audi	Риал-авто	87-33-98
Nissan	Некст-Авто	94-54-12



<u>Модель</u>	Магазин
BMW	Риал-авто
Audi	Риал-авто
Nissan	Некст-Авто

<u>Магазин</u>	Номер
Риал-авто	87-33-98
Некст-Авто	94-54-12