**Тема 3. Проектирование реляционной БД на основе принципов нормализации. Первичный и внешний ключ.**

**Нормализация БД**

Корректной назовем схему БД, в которой отсутствуют нежелательные зависимости между атрибутами отношений. Процесс разработки корректной схемы реляционной БД называется логическим проектированием БД. Проектирование схемы БД может быть выполнено двумя путями:

* путем декомпозиции (разбиения), когда исходное множество отношений, входящих в схему БД заменяется другим множеством отношений (число их при этом возрастает), являющихся проекциями исходных отношений;
* путем синтеза, то есть путем компоновки из заданных исходных элементарных зависимостей между объектами предметной области схемы БД.

Классическая технология проектирования реляционных баз данных связана с теорией нормализации, основанной на анализе функциональных зависимостей между атрибутами отношений. Понятие функциональной зависимости является фундаментальным в теории нормализации реляционных баз данных. Функциональные зависимости определяют устойчивые отношения между объектами и их свойствами в рассматриваемой предметной области. Именно поэтому процесс поддержки функциональных зависимостей, характерных для данной предметной области, является базовым для процесса проектирования.

Процесс проектирования с использованием декомпозиции представляет собой процесс последовательной нормализации схем отношений, при этом каждая последующая итерация соответствует нормальной форме более высокого уровня и обладает лучшими свойствами по сравнению с предыдущей.

Каждой нормальной форме соответствует некоторый определенный набор ограничений, и отношение находится в некоторой нормальной форме, если удовлетворяет свойственному ей набору ограничений. В теории реляционных БД обычно выделяется следующая последовательность нормальных форм:

* первая нормальная форма (1NF);
* вторая нормальная форма (2NF);
* третья нормальная форма (3NF);
* нормальная форма Бойса—Кодда (BCNF);
* четвертая нормальная форма (4NF);
* пятая нормальная форма, или форма проекции-соединения (5NF или PJNF).

Основные свойства нормальных форм:

* каждая следующая нормальная форма в некотором смысле улучшает свойства предыдущей;
* при переходе к следующей нормальной форме свойства предыдущих нормальных форм сохраняются.
* в основе классического процесса проектирования лежит последовательность переходов от предыдущей нормальной формы к последующей. Однако в процессе декомпозиции мы сталкиваемся с проблемой обратимости,то есть возможности восстановления исходной схемы. Таким образом, декомпозиция должна сохранять эквивалентность схем БД при замене одной схемы на другую.

Схемы БД называются эквивалентными, если содержание исходной БД может быть получено путем естественного соединения отношений, входящих в результирующую схему, и при этом не появляется новых кортежей в исходной БД.

При выполнении эквивалентных преобразований сохраняется множество исходных фундаментальных функциональных зависимостей между атрибутами отношений.

Функциональные зависимости определяют не текущее состояние БД, а все возможные ее состояния, то есть они отражают те связи между атрибутами, которые присущи реальному объекту, который моделируется с помощью БД.

Поэтому определить функциональные зависимости по текущему состоянию БД можно только в том случае, если экземпляр БД содержит абсолютно полную информацию (то есть никаких добавлений и модификации БД не предполагается). В реальной жизни это требование невыполнимо, поэтому набор функциональных зависимостей задает разработчик, системный аналитик, исходя из глубокого системного анализа предметной области.

Приведем ряд основных определений.

Функциональной зависимостью набора атрибутов В отношения R от набора атрибутов A того же отношения, обозначаемой как R.A -> R.B или A -> B называется такое соотношение проекций R[A] и R[B], при котором в каждый момент времени любому элементу проекции R[A] соответствует только один элемент проекции R[B], входящий вместе с ним в какой-либо кортеж отношения R.

Функциональная зависимость R.A -> R.B называется полной, если набор атрибутов B функционально зависит от A и не зависит функционально от любого подмножества A, то есть R.A -> R.B называется полной, если для любого A1, являющегося подмножеством А, R.B функционально не зависит от R.A, в противном случае зависимость R.A -> R.B называется неполной.

Функциональная зависимость R.A -> R.B называется транзитивной, если существует набор атрибутов С такой, что:

* С не является подмножеством А.
* С не включает в себя B.
* Существует функциональная зависимость R.A -> R.C.
* Не существует функциональной зависимости R.C -> R.A.
* Существует функциональная зависимость R.C -> R.B.

Возможным ключом отношения называется набор атрибутов отношения, который полностью и однозначно (функционально полно) определяет значения всех остальных атрибутов отношения, то есть возможный ключ — это набор атрибутов, однозначно определяющий кортеж отношения, и при этом при удалении любого атрибута из этого набора его свойство однозначной идентификации кортежа теряется.

А может ли быть ситуация, когда отношение не имеет возможного ключа? Давайте вспомним определение отношения: отношение — это подмножество декартова произведения множества доменов. И в полном декартовом произведении все наборы значений различны, тем более в его подмножестве. Значит, обязательно для каждого отношения всегда существует набор атрибутов, по которому можно однозначно определить кортеж отношения. В вырожденном случае это просто полный набор атрибутов отношения, потому что если мы зададим для всех атрибутов конкретные значения, то, по определению отношения, мы получим только один кортеж.

В общем случае в отношении может быть несколько возможных ключей. Среди всех возможных ключей отношения обычно выбирают один, который считается главным и который называют первичным ключом отношения.

Неключевым атрибутом называется любой атрибут отношения, не входящий в состав ни одного возможного ключа отношения.

Взаимно-независимые атрибуты — это такие атрибуты, которые не зависят функционально один от другого.

Если в отношении существует несколько функциональных зависимостей, то каждый атрибут или набор атрибутов, от которого зависит другой атрибут, называется детерминантом отношения.

Для функциональных зависимостей как фундаментальной основы проекта БД были проведены исследования, позволяющие избежать избыточного их представления. Ряд зависимостей могут быть выведены из других путем применения правил, названных аксиомами Армстронга, по имени исследователя, впервые сформулировавшего их. Это три основных аксиомы:

* Рефлексивность: если В является подмножеством А, то А -> B
* Дополнение: если А->B, то A.C -> B.C
* Транзитивность: если A -> B и B -> C, то A -> C.

Доказано, что данные правила являются полными и исчерпывающими, то есть, применяя их, из заданного множества функциональных зависимостей можно вывести все возможные функциональные зависимости.

Множество всех возможных функциональных зависимостей, выводимое из заданного набора исходных функциональных зависимостей, называется его замыканием.

Отношение находится в **первой нормальной форме** тогда и только тогда, когда на пересечении каждого столбца и каждой строки находятся только элементарные значения атрибутов.

В некотором смысле это определение избыточно, потому что собственно оно определяет само отношение в теории реляционных баз данных. Однако в силу исторически сложившихся обстоятельств и для преемственности такое определение первой нормальной формы существует и мы должны с ним согласиться. Отношения, находящиеся в первой нормальной форме, часто называют просто нормализованными отношениями. Соответственно, ненормализованные отношения могут интерпретироваться как таблицы с неравномерным заполнением, например таблица "Расписание", которая имеет вид:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Преподаватель** | **День недели** | **Номер пары** | **Название дисциплины** | **Тип занятий** | **Группа** |
| Петров В. И. | Понед. | 1 | Теор. выч. проц. | Лекция | 4906 |
| Вторник | 1 | Комп. графика | Лаб. раб.. | 4907 |
| Вторник | 2 | Комп. графика | Лаб. раб | 4906 |
| Киров В. А. | Понед. | 2 | Теор. информ. | Лекция | 4906 |
| Вторник | 3 | Пр-е на С++ | Лаб. раб. | 4907 |
| Вторник | 4 | Пр-е на С++ | Лаб. раб. | 4906 |
| Серов А. А. | Понед. | 3 | Защита инф. | Лекция. | 4944 |
| Среда | 3 | Пр-е на VB | Лаб. раб | 4942 |
| Четверг | 4 | Пр-е на VB | Лаб. раб. | 4922 |

Здесь на пересечении одной строки и одного столбца находится целый набор элементарных значений, соответствующих набору дней, перечню пар, набору дисциплин, по которым проводит занятия один преподаватель.

Для приведения отношения "Расписание" к первой нормальной форме необходимо дополнить каждую строку фамилией преподавателя.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Преподаватель** | **День недели** | **Номер пары** | **Название дисциплины** | **Тип занятий** | **Группа** |
| Петров В. И. | Понед. | 1 | Теор. выч. проц. | Лекция | 4906 |
| Петров В. И. | Вторник | 1 | Комп. графика | Лаб. раб.. | 4907 |
| Петров В. И. | Вторник | 2 | Комп. графика | Лаб. раб | 4906 |
| Киров В. А. | Понед. | 2 | Теор. информ. | Лекция | 4906 |
| Киров В. А. | Вторник | 3 | Пр-е на С++ | Лаб. раб. | 4907 |
| Киров В. А. | Вторник | 4 | Пр-е на С++ | Лаб. раб. | 4906 |
| Серов А. А. | Понед. | 3 | Защита инф. | Лекция. | 4944 |
| Серов А. А. | Среда | 3 | Пр-е на VB | Лаб. раб | 4942 |
| Серов А. А. | Четверг | 4 | Пр-е на VB | Лаб. раб. | 4922 |

Отношение находится во **второй нормальной форме** тогда и только тогда, когда оно находится в первой нормальной форме и не содержит неполных функциональных зависимостей непервичных атрибутов от атрибутов первичного ключа.

Рассмотрим отношение, моделирующее сдачу студентами текущей сессии. Структура этого отношения определяется следующим набором атрибутов:

(ФИО, Номер зач.кн., Группа, Дисциплина, Оценка)

Так как каждый студент сдает целый набор дисциплин в процессе сессии, то первичным ключом отношения может быть (Номер. зач.кн., Дисциплина), который однозначно определяет каждую строку отношения. С другой стороны, атрибуты ФИО и Группа зависят только от части первичного ключа — от значения атрибута Номер зач. кн., поэтому мы должны констатировать наличие неполных функциональных зависимостей в данном отношении. Для приведения данного отношения ко второй нормальной форме следует разбить его на проекции, при этом должно быть соблюдено условие восстановления исходного отношения без потерь. Такими проекциями могут быть два отношения:

(ФИО, Номер.зач.кн., Группа) (Номер зач.кн., Дисциплина, Оценка)

Этот набор отношений не содержит неполных функциональных зависимостей, и поэтому эти отношения находятся во второй нормальной форме.

А почему надо приводить отношения ко второй нормальной форме? Иначе говоря, какие аномалии или неудобства могут возникнуть, если мы оставим исходное отношение и не будем его разбивать на два? Давайте рассмотрим ситуацию, когда студент переведен из одной группы в другую. Тогда в первом случае (если мы не разбивали исходное отношение на два) мы должны найти все записи с данным студентом и в них изменить значение атрибута Группа на новое. Во втором же случае меняется только один кортеж в первом отношении. И конечно, опасность нарушения корректности (непротиворечивости содержания) БД в первом случае выше. Может получиться так, что часть кортежей поменяет значения атрибута Группа, а часть по причине сбоя в работе аппаратуры останется в старом состоянии. И тогда наша БД будет содержать записи, которые относят одного студента одновременно к разным группам. Чтобы этого не произошло, мы должны принимать дополнительные непростые меры, например организовывать процесс согласованного изменения с использованием сложного механизма транзакций. Если же мы перешли ко второй нормальной форме, то мы меняем только один кортеж. Кроме того, если у нас есть студенты, которые еще не сдавали экзамены, то в исходном отношении мы вообще не можем хранить о них информацию, а во второй схеме информация о студентах и их принадлежности к конкретной группе хранится отдельно от информации, которая связана со сдачей экзаменов, и поэтому мы можем в этом случае отдельно работать со студентами и отдельно хранить и обрабатывать информацию об успеваемости и сдаче экзаменов, что в действительности и происходит.

Отношение находится **в третьей нормальной форме** тогда и только тогда, когда оно находится во второй нормальной форме и не содержит транзитивных зависимостей.

Рассмотрим отношение, связывающее студентов с группами, факультетами и специальностями, на которых он учится.

(ФИО, Номер зач.кн., Группа, Факультет, Специальность, Выпускающая кафедра)

Первичным ключом отношения является Номер зач.кн., однако рассмотрим остальные функциональные зависимости. Группа, в которой учится студент, однозначно определяет факультет, на котором он учится, а также специальность и выпускающую кафедру. Кроме того, выпускающая кафедра однозначно определяет факультет, на котором обучаются студенты, выпускаемые по данной кафедре. Но если мы предположим, что одну специальность могут выпускать несколько кафедр, то специальность не определяет выпускающую кафедру. В этом случае у нас есть следующие функциональные зависимости:

Номер зач.кн. -> ФИО

Номер зач.кн. -> Группа

Номер зач.кн. -> Факультет

Номер зач.кн. -> Специальность

Номер зач.кн. -> Выпускающая кафедра

Группа -> Факультет

Группа -> Специальность

Группа -> Выпускающая кафедра

Выпускающая кафедра -> Факультет

И эти зависимости образуют транзитивные группы. Для того чтобы избежать этого, мы можем предложить следующий набор отношений:

(Номер.зач.кн., ФИО, Специальность, Группа)

(Группа, Выпускающая кафедра)

(Выпускащая кафедра, Факультет)

Первичные ключи отношений выделены. Полученный набор отношений находится в третьей нормальной форме.

Отношение R находится в **четвертой нормальной форме** (4NF) в том и только в том случае, если в случае существования многозначной зависимости A ->> B все остальные атрибуты R функционально зависят от A.

В отношении R (A, B, C) существует многозначная зависимость (multi valid dependence, MVD) R.A ->> R.B в том и только в том случае, если множество значений B, соответствующее паре значений A и C, зависит только от A и не зависит от С.

Когда мы рассматривали функциональные зависимости, то каждому значению детерминанта соответствовало только одно значение зависимого от него атрибута. При рассмотрении многозначных зависимостей мы выделяем случаи, когда одному значению некоторого атрибута соответствует устойчиво постоянное множество значений другого атрибута. Когда это может быть? Рассмотрим конкретную ситуацию, понятную всем студентам. Пусть дано отношение, которое моделирует предстоящую сдачу экзаменов на сессии. Допустим, оно имеет вид:

(Номер зач.кн., Группа, Дисциплина)

Перечень дисциплин, которые должен сдавать студент, однозначно определяется не его фамилией, а номером группы (то есть специальностью, на которой он учится).

В данном отношении существуют следующие две многозначные зависимости:

Группа ->> Дисциплина

Группа ->> Номер зач.кн.

Это означает, что каждой группе однозначно соответствует перечень дисциплин по учебному плану и номер группы определяет список студентов, которые в этой группе учатся.

Если мы будем работать с исходным отношением, то мы не сможем хранить информацию о новой группе и ее учебном плане — перечне дисциплин, которые должна пройти группа до тех пор, пока в нее не будут зачислены студенты. При изменении перечня дисциплин по учебному плану, например при добавлении новой дисциплины, внести эти изменения в отношение для всех студентов, занимающихся в данной группе, весьма затруднительно. С другой стороны, если мы добавляем студента в уже существующую группу, то мы должны добавить множество кортежей, соответствующих перечню дисциплин для данной группы. Эти аномалии модификации отношения как раз и связаны с наличием двух многозначных зависимостей.

В теории реляционных баз данных доказывается, что в общем случае в отношении R (A, B, C) существует многозначная зависимость R.A ->> R.B в том и только в том случае, когда существует многозначная зависимость R.A ->> R.C.

В нашем примере можно произвести декомпозицию исходного отношения в два отношения:

(Номер зач.кн., Группа)

(Группа, Дисциплина)

Оба эти отношения находятся в 4NF и свободны от отмеченных аномалий. Действительно, обе операции модификации теперь упрощаются: добавление нового студента связано с добавлением всего одного кортежа в первое отношение, а добавление новой дисциплины выливается в добавление одного кортежа во второе отношение, кроме того, во втором отношении мы можем хранить любое количество групп с определенным перечнем дисциплин, в которые пока еще не зачислены студенты.

Отношение находится в **нормальной форме Бойса—Кодда**, если оно находится в третьей нормальной форме и каждый детерминант отношения является возможным ключом отношения.

Рассмотрим отношение, моделирующее сдачу студентом текущих экзаменов. Предположим, что студент может сдавать экзамен по одной дисциплине несколько раз, если он получил неудовлетворительную оценку. Допустим, что во избежание возможных полных однофамильцев мы можем однозначно идентифицировать студента номером его зачетной книжки, но, с другой стороны, у нас ведется электронный учет текущей успеваемости студентов, поэтому каждому студенту-присваивается в период его обучения в вузе уникальный номер-идентификатор. Отношение, которое моделирует сдачу текущей сессии, имеет следующую структуру:

(Номер зач.кн., Идентификатор\_студента, Дисциплина, Дата, Оценка)

Возможными ключами отношения являются Номер\_зач.кн, Дисциплина, Дата и Идентификатор\_студента, Дисциплина, Дата.

Какие функциональные зависимости у нас имеются?

Номер\_зач.кн, Дисциплина, Дата -> Оценка;

Идентификатор\_студента, Дисциплина, Дата -> Оценка;

Номер зач.кн. -> Идентификатор\_студента;

Идентификатор\_студента -> Номер зач.кн.

Откуда взялись две последние функциональные зависимости? Но ведь мы предварительно описали, что каждому студенту ставится в соответствие один номер зачетной книжки и один Идентификатор\_студента, поэтому по значению Номер зач.кн. можно однозначно определить Идентификатор\_студента (это третья зависимость) и обратно (и это четвертая зависимость). Оценим это отношение.

Это отношение находится в третьей нормальной форме, потому что неполных функциональных зависимостей непервичных атрибутов от атрибутов возможного ключа здесь не присутствует и нет транзитивных зависимостей. А как же третья и четвертая зависимости, разве они не являются неполными? Нет, потому что зависимым не является непервичный атрибут, то есть атрибут, не входящий ни в один возможный ключ. Поэтому придраться к этому мы не можем. Но вот под четвертую нормальную форму наше отношение не подходит, потому что у нас есть два детерминанта Номер зач.кн.и Идентификатор\_студента, которые не являются возможными ключами отношения. Для приведения отношения к нормальной форме Бойса—Кодда надо разделить отношение, например, на два со следующими схемами:

(Идентификатор\_студента, Дисциплина, Дата, Оценка)

(Номер зач.кн., Идентификатор\_студента)

или наоборот:

(Номер зач.кн., Дисциплина, Дата, Оценка)

(Номер зач.кн., Идентификатор\_студента)

Эти схемы равнозначны с точки зрения теории нормализации, поэтому выбирать проектировщикам следует исходя из некоторых дополнительных рассуждений. Ну, например, если учесть, что зачетные книжки могут теряться, то как они будут восстанавливаться: если с тем же самым номером, то нет разницы, но если с новым номером, то тогда первая схема предпочтительней.

Последней нормальной формой является пятая нормальная форма 5NF, которая связана с анализом нового вида зависимостей, зависимостей "проекции соединения" (project-join зависимости,обозначаемые как PJ-зависимости). Этот вид зависимостей является в некотором роде обобщением многозначных зависимостей.

Отношение R находится в **пятой нормальной форме** (нормальной форме проекции-соединения — PJ/NF ) в том и только в том случае, когда любая зависимость соединения в R следует из существования некоторого возможного ключа в R.

Рассмотрим отношение R1:

R1(Преподаватель, Кафедра, Дисциплина)

Предположим, что каждый преподаватель может работать на нескольких кафедрах и на каждой кафедре может вести несколько дисциплин. В этом случае ключом отношения является полный набор из трех атрибутов. В отношении отсутствуют многозначные зависимости, и поэтому отношение находится в 4NF.

Введем следующие обозначения наборов атрибутов:

ПК (Преподаватель, Кафедра)

ПД (Преподаватель, Дисциплина)

КД (Кафедра, Дисциплина)

Допустим, что отношение R1 удовлетворяет зависимости проекции соединения (ПК, ПД, КД). Тогда отношение R1 не находится в NF/PJ, потому что единственным ключом его является полный набор атрибутов, а наличие зависимости PJ связано с наборами атрибутов, которые не составляют возможные ключи отношения R1. Для того чтобы привести это отношение к NF/PJ, его надо представить в виде трех отношений:

R2 (Преподаватель, Кафедра)

R3 (Преподаватель, Дисциплина)

R4 (Кафедра, Дисциплина)

Пятая нормальная форма редко используется на практике. В большей степени она является теоретическим исследованием. Очень тяжело определить само наличие зависимостей "проекции—соединения", потому что утверждение о наличии такой зависимости делается для всех возможных состояний БД, а не только для текущего экземпляра отношения R1. Однако знание о возможном наличии подобных зависимостей, даже теоретическое, нам все же необходимо.

**Внешний и первичный ключ**

Выше мы вспоминали: каждая строка (запись) БД должна быть уникальна. Именно первичный ключ в виде наборов определенных значений, максимально идентифицируют каждую запись. Можно определить по-другому. Первичный ключ: набор определенных признаков, уникальных для каждой записи. Обозначается первичный ключ, как primary key. Primary key (PK) очень важен для каждой таблицы, поскольку:

* Primary key не позволяет создавать одинаковых записей (строк) в таблице;
* PK обеспечивают логическую связь между таблицами одной базы данных (для реляционных БД);

На логической связи между таблицами, стоит остановиться подробнее.

**Ключ внешний**

Foreign key, кратко FK. Обеспечивает однозначную логическую связь, между таблицами одной БД.

Например, есть две таблицы А и В. В таблице А (обувь), есть первичный ключ: размер, в таблице В (цвет) должна быть колонка с названием размер. В этой таблице «размер» это и будет внешний ключ для логической связи таблиц В и А.

Более сложный пример.

Две таблицы данных: Люди и Номера телефонов.

Таблица: Люди

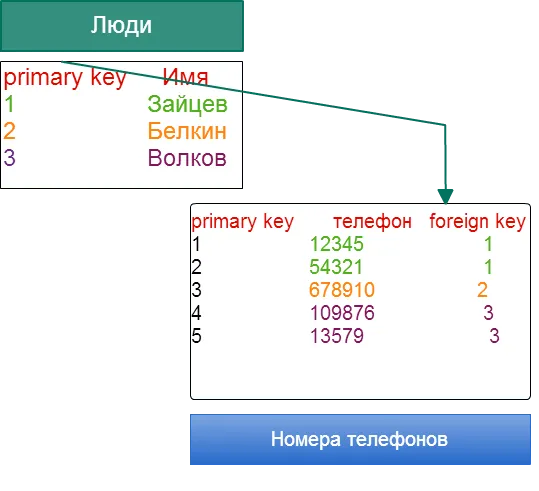
|  |  |
| --- | --- |
| primary key | Имя |
| 1 | Зайцев |
| 2 | Белкин |
| 3 | Волков |

Таблица: Номера телефонов

|  |  |  |
| --- | --- | --- |
| primary key | телефон | foreign key |
| 1 | 12345 | 1 |
| 2 | 54321 | 1 |
| 3 | 678910 | 2 |
| 4 | 109876 | 3 |
| 5 | 13579 | 3 |

В таблице Номера телефонов PK уникален. FK этой таблицы является PK таблицы Люди. Связь между номерами телефонов и людьми обеспечивает FK таблицы телефонов. То есть:

* У Зайцева два телефона;
* У Волкова два телефона;
* У Белкина один телефон.



первичный ключ и внешний ключ