Виды ключей

**Потенциальный ключ**

В реляционной модели данных подмножество атрибутов отношения, удовлетворяющее требованиям уникальности и минимальности (несократимости).

* **Уникальность** означает, что нет и не может быть двух кортежей данного отношения, в которых значения этого подмножества атрибутов совпадают (равны). Свойство уникальности определяется не для конкретного значения в таблице в тот или иной момент времени, а по всем возможным значениям, то есть следует из внешнего знания о природе и закономерностях данных, которые могут находиться в таблице.
* **Минимальность** (несократимость) означает, что в составе потенциального ключа отсутствует меньшее подмножество атрибутов, удовлетворяющее условию уникальности. Иными словами, если из потенциального ключа убрать любой атрибут, он утратит свойство уникальности.

Поскольку все кортежи в таблице по определению уникальны, в нём всегда существует хотя бы один потенциальный ключ (например, включающий все атрибуты отношения).

**Первичный ключ (primary key)**

В реляционной модели данных один из потенциальных ключей отношения, выбранный в качестве основного ключа (или ключа по умолчанию).

Если в отношении имеется единственный потенциальный ключ, он является и первичным ключом. Если потенциальных ключей несколько, один из них выбирается в качестве первичного, а другие называют «альтернативными».

Критерии выбора первичного ключа

* Удобство для создания внешних ключей в других таблицах либо для создания кластерного индекса. Поэтому в качестве первичного ключа, как правило, выбирают тот, который имеет наименьший размер (физического хранения) и/или включает наименьшее количество атрибутов;
* Сохранение уникальности со временем;

**Внешний ключ (англ. foreign key)**

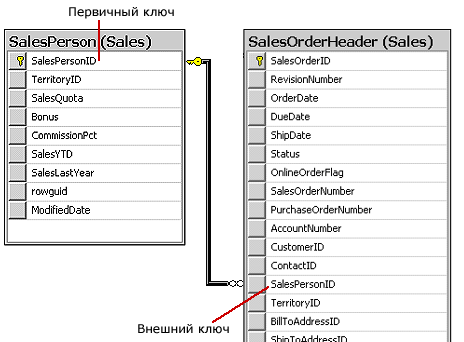
Внешний ключ представляет собой подмножество атрибутов некоторой таблицы R2, значения которых должны совпадать со значениями некоторого потенциального ключа некоторой таблицы R1.

Внешние ключи используются для организации связей между таблицами базы данных (родительскими и дочерними) и для поддержания ограничений ссылочной целостности данных.

Ссылочная целостность проверяется при:

* Удалении записей родительской таблицы;
* Модификация значений полей родительской таблицы, на которые ссылаются поля внешнего ключа дочерней таблицы;

Основная цель ссылочной целостности заключается в недопустимости наличия "висячих" ссылок из дочерних таблиц на родительскую таблицу.



Нормальные формы

**Нормальная форма** - требование, предъявляемое к структуре таблиц в теории реляционных баз данных для устранения из базы избыточных функциональных зависимостей между атрибутами (полями таблиц).

Процесс преобразования отношений (таблиц) базы данных к виду, отвечающему нормальным формам, называется нормализацией.

Нормализация предназначена для приведения структуры БД к виду, обеспечивающему минимальную логическую избыточность, и не имеет целью уменьшение или увеличение производительности работы, или же уменьшение или увеличение физического объёма базы данных.

Ниже приведены математически нестрогие определения, но достаточные для практического проектирования баз данных.

На практике нормальные формы выше третей используются редко. Несоблюдение требований нормальных форм выше третей может привести к ухудшению архитектуры базы данных, но на функциональности это сказаться не должно.

**Первая нормальная форма (1NF)**

**Основные критерии**

* Все строки должны быть различными (не должно быть повторений строк в таблице);
* Все элементы внутри ячеек должны быть атомарными (не списками). Элемент является атомарным, если его нельзя разделить на части, которые могут использовать в таблице независимо друг от друга;

**Пример не 1NF таблицы**

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

M5, X5M, M1 – список, не атомарное значение.

**Исправить нужно так**

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

**Методы приведения к 1NF**

* Устраните повторяющиеся группы в отдельных таблицах (одинаковые строки);
* Создайте отдельную таблицу для каждого набора связанных данных;
* Идентифицируйте каждый набор связанных данных с помощью первичного ключа (добавить уникальный идентификатор для каждой строки);

**Вторая нормальная форма (2NF)**

**Основные критерии**

* Таблица должна находиться в первой нормальной форме;
* Любое её поле, не входящее в состав первичного ключа, функционально полно зависит от первичного ключа;

**Пример не 2NF таблицы**

|  |  |  |  |
| --- | --- | --- | --- |
| Категория | Дата | Скидка | Товар |
| Книги | 10.10.2008 | 10% | PHP for dummies |
| Ноутбуки | 10.10.2008 | 20% | Acer |
| Книги | 10.10.2008 | 10% | Windows XP |

В этой таблице первичный ключ составляют первые два столбца (Категория и Дата)

Скидка функционально полно зависит от ключа, так как определяется одновременно категорией и датой. Товар зависит только от категории и не зависит от даты. Поэтому, говорят, что он зависит функционально не полно от всего первичного ключа, а только зависит только от его части (Категория). Это нарушает требования второй нормальной формы.

**Исправить это нужно разделением на две таблицы**

|  |  |  |
| --- | --- | --- |
| Категория | Дата | Скидка |
| Книги | 10.10.2008 | 10% |
| Ноутбуки | 10.10.2008 | 20% |
| Книги | 10.10.2008 | 10% |

|  |  |
| --- | --- |
| Категория | Товар |
| Книги | PHP for dummies |
| Ноутбуки | Acer |
| Книги | Windows XP |

**Методы приведения к 2NF**

* Создайте отдельные таблицы для наборов значений, относящихся к нескольким записям (Выше мы это сделали);
* Свяжите эти таблицы с помощью внешнего ключа (в нашем случае это поле Категория);

**Третья нормальная форма (3NF)**

**Основные критерии**

* Таблица находится во второй нормальной форме;
* Любой её не ключевой атрибут функционально зависит только от первичного ключа.

Проще говоря, третье правило требует выносить все не ключевые поля, содержимое которых может относиться к нескольким записям таблицы в отдельные таблицы.

**Пример не 3NF таблицы**

|  |  |
| --- | --- |
| Имя шпиона | Государство |
| Джеймс Бонд | Великобритания |
| Ким Филби | СССР |
| Штирлиц | СССР |

В этой таблице ключом является имя шпиона. А не ключевым полем – государство. В этой таблице государства могут быть одинаковыми для нескольких шпионов.

И для того, чтобы эта таблица находилась в третей нормальной форме, не обходимо ее разделить на две:

|  |  |
| --- | --- |
| ID | Государство |
| 1 | Великобритания |
| 2 | СССР |

|  |  |
| --- | --- |
| Имя шпиона | Государство |
| Джеймс Бонд | 1 |
| Ким Филби | 2 |
| Штирлиц | 2 |

**Методы приведения к 3NF**

* Удаление полей, не зависящих от ключа

**Нормальная форма Бойса-Кодда (BCNF)**

Эта форма – частный случай третьей нормальной формы. Для отношений, имеющих один потенциальный ключ (первичный), BCNF является 3NF.

**Основные критерии**

* Таблица находится в третьей нормальной форме;
* Таблица находится в нормальной форме Бойса-Кодда тогда и только тогда, когда детерминанты всех ее функциональных зависимостей являются потенциальными ключами;

Другими словами, в таблице должен быть только один первичный ключ и не должно быть других потенциальных вариантов (например, набор не ключевых полей этой таблицы).

**Пример не BCNF таблицы**

|  |  |  |  |
| --- | --- | --- | --- |
| **Номер корта** | **Время начала** | **Время окончания** | **Тариф** |
| 1 | 09:30 | 10:30 | «Бережливый» |
| 1 | 11:00 | 12:00 | «Бережливый» |
| 1 | 14:00 | 15:30 | «Стандарт» |
| 2 | 10:00 | 11:30 | «Премиум-B» |
| 2 | 11:30 | 13:30 | «Премиум-B» |
| 2 | 15:00 | 16:30 | «Премиум-А» |

Тариф имеет уникальное название и зависит от выбранного корта и членства в клубе:

* «Бережливый»: корт 1 для членов клуба;
* «Стандарт»: корт 1 для не членов клуба;
* «Премиум-А»: корт 2 для членов клуба;
* «Премиум-B»: корт 2 для не членов клуба.

Таким образом, возможны следующие составные первичные ключи: {Номер корта, Время начала}, {Номер корта, Время окончания}, {Тариф, Время начала}, {Тариф, Время окончания}.

Отношение соответствует второй (2NF) и третьей (3NF) нормальной форме. Требования второй нормальной формы выполняются, так как все атрибуты входят в какой-то из потенциальных ключей, а не ключевых атрибутов в отношении нет. Также нет и транзитивных зависимостей, что соответствует требованиям третьей нормальной формы.

Тем не менее, существует функциональная зависимость Тариф → Номер корта, в которой левая часть (детерминант) не является потенциальным ключом отношения, то есть отношение не находится в нормальной форме Бойса — Кодда.

Недостатком данной структуры является то, что, например, по ошибке можно приписать тариф «Бережливый» к бронированию второго корта, хотя он может относиться только к первому корту.

Можно улучшить структуру с помощью декомпозиции отношения на два и добавления атрибута **"Является членом клуба"**, получив отношения, удовлетворяющие BCNF (подчёркнуты атрибуты, входящие в первичный ключ):

|  |  |  |
| --- | --- | --- |
| **Тариф** | **Номер корта** | **Является членом клуба** |
| «Бережливый» | 1 | Да |
| «Стандарт» | 1 | Нет |
| «Премиум-А» | 2 | Да |
| «Премиум-B» | 2 | Нет |

|  |  |  |
| --- | --- | --- |
| **Тариф** | **Время начала** | **Время окончания** |
| «Бережливый» | 09:30 | 10:30 |
| «Бережливый» | 11:00 | 12:00 |
| «Стандарт» | 14:00 | 15:30 |
| «Премиум-B» | 10:00 | 11:30 |
| «Премиум-B» | 11:30 | 13:00 |
| «Премиум-А» | 15:00 | 16:30 |

**Методы приведения к BCNF**

* Вынести в отдельную таблицу потенциальные первичные ключи;
* Добавить дополнительные поля при необходимости;

**Четвертая нормальная форма**

Отношение находится в 4NF, если оно находится в BCNF и все нетривиальные многозначные зависимости являются функциональными зависимостями от ее потенциальных ключей.

Предположим, что рестораны производят разные виды пиццы, а службы доставки ресторанов работают только в определенных районах города. Составной первичный ключ соответствующей переменной отношения включает три атрибута: {Ресторан, Вид пиццы, Район доставки}.

Такая таблица не соответствует 4NF, так как существует следующая многозначная зависимость: {Ресторан} → {Вид пиццы}, {Ресторан} → {Район доставки}.

То есть, например, при добавлении нового вида пиццы придется внести по одному новому кортежу для каждого района доставки. Возможна логическая аномалия, при которой определенному виду пиццы будут соответствовать лишь некоторые районы доставки из обслуживаемых рестораном районов.  
  
Для предотвращения аномалии нужно декомпозировать отношение, разместив независимые факты в разных отношениях. В данном примере следует выполнить декомпозицию на {Ресторан, Вид пиццы} и {Ресторан, Район доставки}.

Однако, если к исходной переменной отношения добавить атрибут, функционально зависящий от потенциального ключа, например, цену с учётом стоимости доставки ({Ресторан, Вид пиццы, Район доставки} → Цена), то полученное отношение будет находиться в 4NF и его уже нельзя подвергнуть декомпозиции без потерь.

**Пятая нормальная форма (5NF)**

В некоторых предыдущих формах, для разрешения требований, мы производили декомпозицию таблицы (выделение некоторых полей в отдельную таблицу) на две другие. Иногда такого рода декомпозицию нельзя без потерь произвести (на две таблицы именно), но зато можно произвести декомпозицию на 3 и более таблицы.

Пятая форма требует, чтобы все возможные декомпозиции были произведены.

**Доменно-ключевая нормальная форма (DKNF)**

Таблица находится в ДКНФ тогда и только тогда, когда каждое наложенное на неё ограничение является логическим следствием ограничений доменов и ограничений ключей, наложенных на данную таблицу.

**Ограничение домена** - ограничение, предписывающее использовать для определённого атрибута значения только из некоторого заданного домена. Ограничение по своей сути является заданием перечня (или логического эквивалента перечня) допустимых значений типа и объявлением о том, что указанный атрибут имеет данный тип.

**Ограничение ключа** - ограничение, утверждающее, что некоторый атрибут, или комбинация атрибутов, является потенциальным ключом.

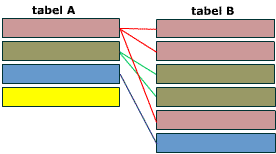
Любая переменная отношения, находящаяся в ДКНФ, обязательно находится в 5НФ. Однако не любую переменную отношения можно привести к ДКНФ.

**Шестая нормальная форма (6NF)**

Переменная отношения находится в шестой нормальной форме тогда и только тогда, когда она удовлетворяет всем нетривиальным зависимостям соединения. Из определения следует, что переменная находится в 6NF тогда и только тогда, когда она неприводима, то есть не может быть подвергнута дальнейшей декомпозиции без потерь. Каждая переменная отношения, которая находится в 6NF, также находится и в 5NF.

Виды отношений между таблицами

**Связь один-ко-многим**

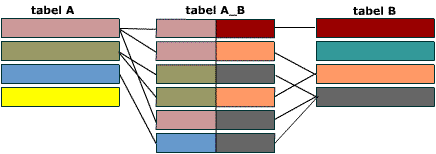
Когда одна запись в таблице А может быть связана с 0, 1 или множеством записей в таблице B, вы имеете дело со связью **один-ко-многим**. В реляционной модели данных связь один-ко-многим использует две таблицы.  


Примеры связи один-ко-многим:

* Машина и ее части. Каждая часть машины единовременно принадлежит только одной машине, но машина может иметь множество частей.
* Дома и улицы. На улице может быть несколько домов, но каждый дом принадлежит только одной улице.

**Связь многие-ко-многим**

Связь многие-ко-многим – это связь, при которой множественным записям из одной таблицы (A) могут соответствовать множественные записи из другой (B).

Связь многие-ко-многим создается с помощью трех таблиц. Две таблицы – “источника” и одна соединительная таблица. Первичный ключ соединительной таблицы A\_B – **составной**. Она состоит из двух полей, двух внешних ключей, которые ссылаются на первичные ключи таблиц A и B.  
  
  
  
Все первичные ключи должны быть уникальными. Это подразумевает и то, что комбинация полей A и B должна быть уникальной в таблице A\_B.

Примеры связи многие-ко-многим:

* Школа, где учителя обучают учащихся. Каждый учитель обучает многих учащихся, а каждый учащийся может обучаться несколькими учителями;

**Связь один-к-одному.**

В связи один-к-одному каждый блок сущности A может быть ассоциирован с 0, 1 блоком сущности B.

Связь один-к-одному можно смоделировать:

* **В одной таблице**. Связь один-к-одному легко моделируется в одной таблице. Записи таблицы содержат данные, которые находятся в связи один-к-одному с первичным ключом или записью;
* **В отдельных таблицах**. В редких случаях связь один-к-одному моделируется с использованием двух таблиц. Такой вариант иногда необходим, чтобы преодолеть ограничения РСУБД или с целью увеличения производительности. Наличие двух таблиц в связи один-к-одному считается плохой практикой;

Примеры связи многие-ко-многим:

* Люди и их паспорта. Каждый человек в стране имеет только один действующий паспорт и каждый паспорт принадлежит только одному человеку;

1. <http://i-novice.net/6-normalnyx-form-bd/>
2. <https://habrahabr.ru/post/254773/>
3. <https://ru.wikipedia.org/wiki/%D0%9D%D0%BE%D1%80%D0%BC%D0%B0%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D1%84%D0%BE%D1%80%D0%BC%D0%B0>
4. <http://habrahabr.ru/post/193380/>