

Ключи: классификация и применение при проектировании баз данных.

Тема суррогатных и естественных ключей баз данных обсуждалась многократно, и тем не менее каждый раз вызывает оживленную дискуссию, каждый раз демонстрирующую терминологическую несогласованность. На взгляд автора, точнее говорить о трех видах ключей с точки зрения БД в целом, а также отдельно рассматривать способ миграции ключей внутри БД. Приведен краткий обзор преимуществ и недостатков естественных и суррогатных ключей. В дополнение к общеизвестным аргументам рассматривается случай ограничения целостности “тот же родитель”.

1. Основные понятия

Мы рассматриваем базы данных SQL типа, т.е в той или иной мере поддерживающих реляционную модель данных (РМД), поэтому будем использовать термины “таблица”, “строка” и “атрибут”. Под *ключом таблицы* как обычно понимается набор атрибутов, который однозначно идентифицирует строку таблицы - никакие две строки не обладают одинаковыми значениями ключа. Ключ не может включать NULL значений. В РМД принято требовать минимальности набора атрибутов, но здесь это несущественно. Определенный таким образом ключ является кандидатом на роль первичного ключа таблицы, который и используется для ссылок на таблицу во внешних ключах других таблиц и в других конструкциях. Некоторые СУБД допускают внешние ключи, ссылающиеся на любой не NULL ключ. Но для определенности будем считать, что ключ, используемый для построения внешних ключей, является первичным.

Атрибуты определены на доменах базы данных. При семантическом моделировании домены часто называют элементами данных. Домен, на котором определен один из атрибутов ключа какой-либо таблицы, является *ключевым доменом*, а набор доменов атрибутов ключа - *ключом базы данных*. Внешние ключи таблиц очевидно определены также на ключах базы данных. Обычно классификация суррогатных и естественных ключей дается относительно таблиц. Это приводит к следующей проблеме: если суррогатный ключ А таблицы Т1 входит в ключ (А,В) таблицы Т2, то остается ли А суррогатом в Т2 ? Ведь он определен не в данной таблице. Правильнее все же различать суррогатные домены и бизнес-домены БД, а миграция ключей таблиц – это отдельный вопрос.

2. Классификация

Тип ключа базы данных определяется взаимосвязями домена с внешним по отношению к базе данных миром.

Замечание. Часто говорят, что база данных содержит сведения о реальном мире. Автору кажется неудачным термин “реальный мир” в данном контексте. Во-первых БД не менее реальна чем остальные произведения человеческих рук, является одним из реальных инструментов ведения бизнеса, научного исследования или иной деятельности, и в этой роли может быть не только отражением, но также источником информации (см. далее искусственные ключи). Во-вторых, объектами внешнего мира могут быть в свою очередь какие-то информационные системы и базы данных. Невозможно считать одну базу данных реальным миром а другую – не реальным.

Для нас важно лишь, что мы вполне определенно очертили границы нашей БД, - что к ней относится, а что нет. Если эти границы меняются, то меняется и тип ключа. Разумеется, нас интересуют только информационные процессы внешнего мира – документы, знания людей, электронные информационные системы, другие базы данных, ресурсы Интернет и т.д. Взаимосвязи ключей базы данных и соответствующей им информации во внешнем мире важны в двух аспектах – **область использования (Scope)** и **источник значения (Source)**.

2.1. Область использования ключа

Область использования ключа может быть либо ограничена базой данных, либо охватывать как БД так и внешний мир. Данные, имеющиеся во внешнем мире, но отсутствующие в БД не интересны по определению. Итак по области использования мы делим ключи БД на **внутренние** и **бизнес-ключи**.

Внутренний ключ за пределами БД бессмыслен или лучше сказать его смысл непредсказуем. Если запомнить значение внутреннего ключа, а затем через некоторое время (в другой транзакции) обратиться к БД с запросом информации по запомненному ключу, то результат непредсказуем. В этом отношении внутренний ключ сходен с физическим адресом. По прошествии времени никоим образом не гарантировано, что нужная информация лежит по прежнему адресу, и что данный адрес не был использован для каких-то других целей. Соответственно внутренние ключи не интересны пользователям (за исключением администраторов системы при анализе проблем) как средство идентификации конкретного объекта.

Заметим однако, что само свойство уникальности внутреннего ключа вполне может быть использовано пользователем БД в запросах. Например, можно запросить среднее число различных значений ключа работников в таблице состава подразделений.

К внутренним ключам БД относятся прежде всего системные атрибуты - адреса записей во внутреннем адресном пространстве СУБД (атрибут Rowid в ORACLE). Rowid не подконтролен приложению, СУБД может изменить его например при переносе таблицы в другое табличное пространство. Такие ключи назовем встроенными ключами СУБД, и далее не рассматриваем. Мы

можем использовать некоторые обычные с точки зрения СУБД поля как собственные внутренние ключи БД. Именно их принято называть **суррогатными ключами (СК)**. По замыслу данное значение суррогатного ключа всегда будет соответствовать данному объекту, выражая его идентичность при любом изменении информации об объекте. Например, человек может сменить фамилию, паспорт, внешность (фотографию в БД), но благодаря сохранению значения СК мы “знаем”, что это тот же самый человек. Но верно и то, что значение СК для данного объекта может измениться при реорганизации данных. Также существует практика повторного использования данного значения СК, когда например при увольнении сотрудника запись с данным СК помечается как логически удаленная, а через некоторое время разрешается ее использовать для вновь принятого сотрудника. Поэтому с точки зрения автора правильнее рассматривать СК не как идентификатор объекта, а как адрес логического адресного пространства БД. Доменом СК является не набор объектов внешнего мира, а отведенный (с помощью тех или иных механизмов СУБД) администратором базы данных набор значений.

Так или иначе, важно, что соответствие СК объекту внешнего мира существует лишь в пределах БД и не существует никаких обязательств БД перед внешним миром сохранять это соответствие, т.е. область использования СК ограничена данной БД. Заметим, что внешние ключи таблиц, ссылающиеся на данный СК, очевидно определены на том же суррогатном домене – суррогатном ключе БД.

Замечание. Используя пример с БД о сотрудниках отметим, что применяя только СК без других ограничений целостности невозможно отличить изменение сведений о человеке от подмены данных одного человека данными другого. Это еще раз говорит о том, что логически СК – адрес, а не идентификатор.

Бизнес-ключ используется как во внешнем мире так и в БД, для него характерна согласованность смысла между БД и внешним миром.

Замечание. Неоднократно уже употреблялось выражение “смысл ключа”. Естественно, единственный смысл, который здесь имеется ввиду – это соответствие между значением ключа и объектом внешнего мира. Мы считаем, что смысл данного значения ключа сохраняется при любых изменениях идентифицируемой этим ключом информации до тех пор, пока эта информация относится все к тому же объекту.

Эта согласованность требует некоторых специальных мер и усилий, здесь не обсуждаемых. Пример - справочник валют. Мы вполне можем использовать общепринятые трехсимвольные коды валют как первичный ключ справочника валют нашей БД. Часто можно встретить утверждение что бизнес-ключи “существуют в реальном мире”, но в настоящее время все чаще в бизнесе используются идентификаторы, порождаемые в некоторой информационной системе. Тем не менее они должны быть классифицированы именно как бизнес-ключи. Важно, что их значения согласованы с внешним миром, а что касается источника значений – это

следующий вопрос. Бизнес ключ имеет одинаковый смысл и домен значений в БД и во внешнем мире, откуда бы не бралось его значение.

2.2. Источник значений

Источником значения внутреннего ключа по определению является БД (даже в том случае, когда предполагаемое значение суррогатного ключа вводится с клавиатуры. Процедуру ввода, проверки уникальности и подтверждения тогда следует рассматривать как часть реализации БД). Первоисточником значения бизнес-ключа может быть либо БД либо внешний мир. Соответственно бизнес-ключи БД делятся на **искусственные** и **естественные**.

Значение **искусственного** ключа формируется в БД и публикуется для использования во внешнем мире. Обратное, **естественный** ключ формируется неким сторонним для БД источником. Понятно, что относительно этого источника ключ будет искусственным. Т.е. если в одной БД ключ естественный, то где-то существует информационная система, где он является искусственным, ибо никто и ничто не рождается и не возникает с готовым именем или номером. Люди, компании, планеты, показания датчиков, получают имена и иные «естественные» ключи посредством регистрации некоторой признанной организацией. По своему масштабу этой организацией может быть международное агентство по поддержке стандарта ISO 3166 (<http://www.iso.org/iso/en/prods-services/iso3166ma/index.html>), или группа одноклассников, давших прозвище своему товарищу, или система управления технологическим оборудованием, и т.д.

Хотя как искусственные так и суррогатные ключи могут формироваться с помощью идентичных программных механизмов (скажем сиквенса ORACLE, identity поля в MS SQL), следует различать понятия суррогатного и автоматически генерируемого ключа. Практически все суррогатные ключи генерируются автоматически, но некоторые автоматически генерируемые ключи не являются суррогатными. Например, если автогенерированный номер заказа (ключ БД) сообщается пользователю для последующих ссылок, то по нашей классификации он считается искусственным ключом.

Чтобы отличить суррогатные и искусственные ключи можно применить мысленный тест «Сдвиг ключа». Предположим для простоты, что тестируемый ключ имеет числовой тип. Если остановить БД и изменить каждое значение тестируемого ключа, включая его использование во внешних ключах, прибавив к нему единицу, а затем запустить систему, то в случае суррогатного ключа пользователи не смогут заметить никакой разницы в результатах работы системы. В случае искусственного ключа это приведет к нарушению согласованности ключей в БД и внешнем мире.

Источники значений ключа могут быть далее конкретизированы в описании домена в словаре данных системы. Например домен «Код страны» описывается как «Двухсимвольный код страны согласно стандарта ISO 3166-1».

2.3. Итоговая классификация и альтернативы.

В таблице приведена итоговая классификация ключей

Область использования	Источник значений		
	СУБД	БД	Внешний мир
Внутренние ключи: только внутри БД	Встроенный	Суррогатный (Surrogate)	-
Бизнес-ключи: в БД и во внешнем мире	-	Искусственный (Artificial)	Естественный (Natural)

3. Преимущества и недостатки

Для обсуждения используем пример БД. На рис 1 и 2 показаны соответственно варианты БД с бизнес-ключами и с суррогатными ключами. В данной БД все бизнес-ключи являются естественными.

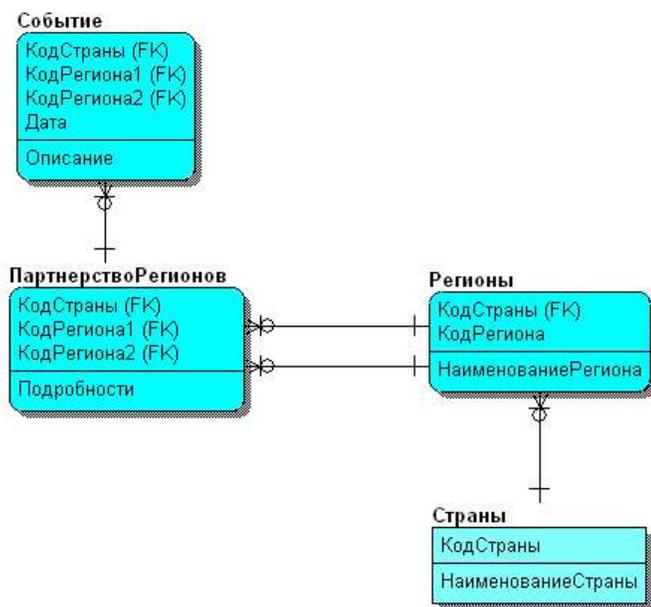


Рисунок 1 Бизнес-ключи

3.1. Преимущества естественных ключей

Реляционная парадигма заключается в отказе от использования на логическом уровне дисковых адресов, указателей и других механизмов управления памятью. Вместо этого следует концентрироваться на собственно данных. Использование бизнес ключей, естественных и искусственных, позволяет в максимальной степени отобразить логику предметной области в структуре базы данных. Бизнес-ключ знаком пользователю, и скажем следующее отображение

вполне понятно пользователю и не требует расшифровок – ему вполне достаточно общепринятых кодов стран и регионов:

Страна	КодРегиона	Наименование
RU	01	Алтайский край
AF	BDG	Badghis

Это важное преимущество естественных ключей к сожалению не всегда удается использовать.

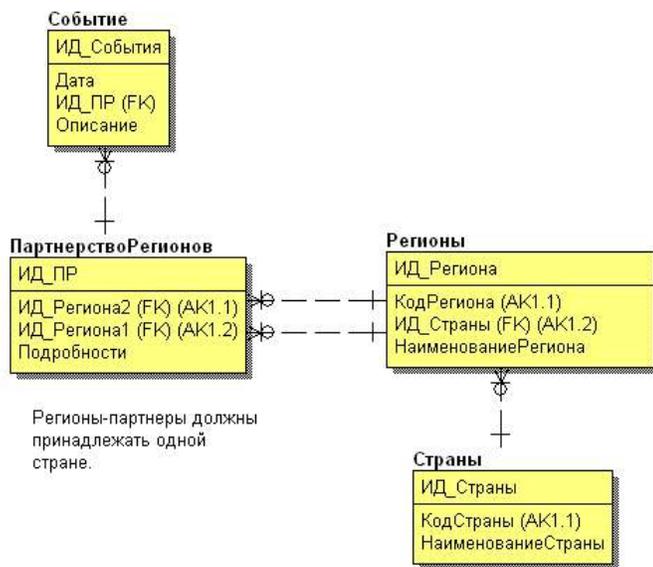


Рисунок 2 Суррогатные ключи

3.2. Сложности использования естественного ключа

1) Некоторые таблицы могут просто не иметь естественного ключа.. Несомненно, проектировщик обязан выявить все возможные потенциальные ключи сущности, ибо это влияет на сам перечень сущностей и на состав бизнес-правил. Однако сложно найти естественный ключ для таких сущностей как «Человек» или «Адрес». Даже КГБ и ЦРУ терпели неудачу в идентификации некоторых людей.

2) По определению естественные ключи находятся вне контроля системы. Мы должны полагаться на то, что источник значений обеспечит стабильность и уникальность ключа, а во-вторых быть уверенны, что имеем необходимый доступ к этому источнику и будем его иметь и впредь. Стабильность включает неизменность структуры ключа (типа значения) и неизменность собственно значений. Любая нестабильность чревата серьезными переделками.

3) Естественный ключ может оказаться громоздким, что вызывает во-первых снижение производительности системы и во-вторых усложнение программного кода, особенно при миграции такого ключа в качестве внешнего во многие другие таблицы.

4) В некоторых случаях использование естественного ключа в подчиненной таблице ведет к тому, что миграция родительского ключа обязана быть идентифицирующей, и первичный ключ подчиненной таблицы становится громоздким. Для примера рассмотрим подчиненную таблицу «Регионы», содержащую информацию о регионах различных стран, перечисленных в вышеупомянутой таблице «Страны». Предположим, что для наших целей ISO 3166-2 недостаточно детален и мы используем национальные классификаторы, например для России "ОБЩЕРОССИЙСКИЙ КЛАССИФИКАТОР ОБЪЕКТОВ АДМИНИСТРАТИВНО-ТЕРРИТОРИАЛЬНОГО ДЕЛЕНИЯ" ОК 019-95, утв. Постановлением Госстандарта РФ от 31.07.1995 N 413 <http://www.mosreestr.ru/118/OKATO.nsf/>. Тогда чтобы застраховаться от случайных совпадений кодов в национальных классификаторах естественный ключ таблицы «Регионы» должен включать первичный ключ родительской таблицы «Страны».

5) Сложно унифицировать программный код из-за различной структуры ключей в различных таблицах.

Все эти недостатки привели к тому, что многие авторы рекомендуют применять исключительно первичные суррогатные ключи. Однако это тоже имеет некоторые негативные последствия.

3.3. Проблемы суррогатных ключей

1) Требуется больше операций соединения чтобы получить полезную конечному пользователю информацию. В сложной системе большее число операций соединения может означать существенное замедление работы, хотя это не является абсолютным правилом. Более того, если бизнес ключ существенно длиннее суррогатного, то в больших подчиненных таблицах можно достигнуть значительной экономии памяти и увеличение числа соединений будет компенсировано меньшим числом обращений к диску. С другой стороны, индексы на основе битовых карт а также партиционирование по бизнес-ключу позволяют получить выигрыш уже при применении бизнес-ключей. Короче, в каждом конкретном случае оптимальное решение может быть разным.

2) При введении суррогатного ключа необходимо оставлять в БД ограничения уникальности, задаваемые бизнес-ключами. Значит потребуются дополнительная память на суррогатный ключ и индекс по нему наряду с индексом по бизнес-ключу. Конечно разумный выбор типа суррогатного ключа существенно сокращает эти затраты.

3) Если нужно объединять данные из разных баз, возникает конфликт ключей. Две разных сущности могут иметь одинаковые суррогатные ключи в разных базах, и наоборот, записи, относящихся к одной и той же сущности, могут иметь разные ключи. Конечно, если эти базы заранее известны, то можно заранее принять меры, применив функцию Win32 API UuidCreate(...), которая почти гарантирует различные значения ключа на различных компьютерах, что снимает первую проблему. Решение второй без участия бизнес-ключей невозможно.

4. Идентифицирующая миграция ключа

Под идентифицирующей миграцией понимается включение в первичный ключ дочерней таблицы внешнего ключа (ссылки) на родительскую таблицу. Считается, что применение суррогатных ключей исключает идентифицирующую миграцию, и что в этом случае каждая таблица должна иметь ключ из единственного атрибута. Однако это не верно. Вопрос об идентифицирующей миграции - это самостоятельное проектное решение, в значительной мере независимое от решения по применению суррогатных ключей. См. например модель корпоративной базы данных в области метеорологии и океанографии министерства обороны США http://pao.cnmoc.navy.mil/scripts/public_JMCDM/home_pwd.pl?JMCDM где широко использована идентифицирующая миграция как суррогатных так и бизнес-ключей.

Выбор идентифицирующей или не идентифицирующей миграции также не однозначен, преимущества и недостатки есть у каждого варианта.

4.1. Преимущества

- 1) Идентифицирующая миграция позволяет избежать длинных цепочек соединений.
- 2) Идентифицирующая миграция упрощает некоторые ограничения целостности (ограничение “тот же родитель”) на следующих уровнях цепочки родитель/ дочка. Остановимся на этом моменте подробнее, так как он практически нигде не упоминается.

Например, пусть сведения о связях регионов одной страны записываются в таблицу «Партнерство регионов» и применена идентифицирующая миграция кода страны в таблицу «Регионы». Тогда внешние ключи для первого и второго региона в таблице «Партнерство регионов» содержат одно и тоже поле «Код страны» и автоматически обеспечивается принадлежность регионов одной стране. В случае, если для таблицы «Регионы» применен собственный суррогатный ключ, это ограничение нужно реализовывать программно, например триггером. Интересный прием, основанный на создании UNIQUE ограничения, включающего и первичный и внешний ключи промежуточной таблицы (в нашем случае «Регионы»), предложил Том Кайт для ORACLE (

http://asktom.oracle.com/pls/ask/f?p=4950:8::::F4950_P8_DISPLAYID:1235228851558#31460269085028 “how to implement this model” January 09, 2005).

4.2. Недостатки

1) Возможная нестабильность естественного ключа родителя распространяется на дочерние таблицы.

2) Громоздкие ключи.

3) Сложность унификации программного обеспечения.

Последние два пункта полностью аналогичны недостаткам естественных ключей. Возможно поэтому их считают недостатками собственно естественных ключей, но в данном случае они возникают именно из-за идентифицирующей миграции ключей, неважно каких – суррогатных или естественных.

5. Ключи в ERP системах

В ERP системах, и не только, применяются на уровне БД обобщенные структуры хранения типа Объект-Свойство-Значение (ОСЗ). Они могут иметь и другую конструкцию. Общее для них то, что набор атрибутов, характерных для класса объектов, задается не набором колонок физических таблиц, а фиксируется в собственном словаре данных системы и контролируется на уровне сервера приложений. Там же осуществляется преобразование в физическую структуру хранения. Все вышесказанное в принципе применимо и к модели данных, задаваемой словарем данных ERP системы, если считать типы объектов логическими таблицами. Для ключей же физических структур применяются только суррогаты.

6. Выводы

Классификации суррогатный/естественный/искусственный подлежат домены базы данных, а не атрибуты таблиц. Атрибут естественно наследует соответствующую характеристику домена. Идентифицирующая миграция возможна для любого вида ключей. Поэтому при характеристике дизайна БД нужно описать и политику применения доменов и политику применения идентифицирующей миграции. В частности популярный дизайн “Каждой таблице – суррогатный ключ из единственного атрибута” описывается как применение суррогатных доменов и только не идентифицирующей миграции ключей.

По мнению автора следует всегда, когда это возможно, применять бизнес-ключи. Слова “когда возможно” означают абсолютную уверенность в отсутствии всех вышеуказанных рисков.

Надо признать, такая возможность – редкое явление. Как правило она имеется для наиболее фундаментальных и мало изменяемых справочных таблиц типа справочников стран или валют.

Однако можно ожидать, что ситуация будет меняться. Все большее число источников данных в настоящее время – это в свою очередь автоматизированные системы. Можно ожидать, что посредством WEB-сервисов или иных средств многие из них будут предоставлять сервис поддержки определенных справочников, и что этот сервис будет отвечать требованиям надежности и стабильности. Поэтому область применения бизнес-ключей также может расширяться.

В любом случае необходимо сохранять естественные ключи в качестве ограничений уникальности для поддержки бизнес-правил, а также отслеживать “выпадающие” при переходе к суррогатным ключам и не идентифицирующей миграции естественные ограничения целостности.

Сергей Левин

sergn-n@yandex.ru