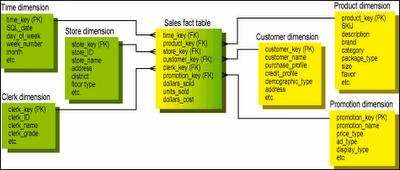
**Суррогатные ключи. Контролируйте идентификаторы строк формированием суррогатных ключей в хранилищах данных**

Автор оригинала: Ральф Кимбалл ([все статьи](http://lissianski.narod.ru/kimball/articles_kimball.html))  
Перевод на русский язык: [Михаил Замыслов](http://mishkaz.blogspot.com)

В соответствии с Webster's Unabridged Dictionary (словарь - прим. переводчика), суррогат - это "искусственный или синтетический продукт, который используется как заменитель натурального продукта". Это замечательное определение для суррогатных ключей мы применяем и для хранилищ данных. Суррогатный ключ - это искусственный или синтетический ключ, который используется в качестве замены натурального ключа.

Вообще говоря, суррогатный ключ в хранилище данных значительно большее чем просто замена натурального ключа. В хранилищах данных суррогатный ключ - это необходимое бобщение натурально произведенного ключа и является одним из основных лэементов проектирования хранилищ данных. Давайте однозначно определимся: Каждое объединение между таблицами размерности и таблицами фактов в хранилищах данных должны быть основаны на суррогатных ключах, а не натуральных. Следует заложить в логику изъятия данных (прим. перевод) систематический просмотр и замену каждого входящего натурального ключа суррогатным ключом хранилища данных при каждом перемещении записи размерности или факта в хранилище данных.

Другими словами, когда мы имеем размерность продукции соединенной с таблицей фактов, или размерность покупателей соединенной с таблицей фактов, или даже размерность времени и таблицу фактов, как показано на рисунке



физические ключи каждого из объединений не являются натуральными ключами , прямо взятыми из входящих данных. Предпочтительней, чтобы ключи, которые используются в качестве суррогатных, являлись обычными ни к чему не привзянными целочисленными значениями. Каждый такой ключ должен быть простым целочисленным integer, начинающимся с единицы и увеличивающимся на сколько это необходимо. Ключ продукта должен быть простым integer, ключ покупателя должен быть простым integer, и даже ключ времени должен быть простым integer. Ни один из ключей не должен быть:

* Значащим, когда вы можете что-то сказать о записи прочитав значение ключа.
* Сочетанием натуральных ключей, соединенных каким либо образом.
* Реализованным как множество параллельных объединений между таблицой размерности и таблицей фактов, так называемые двойные и тройные распределенные объединения (прим. перевод "double or triple barreled joins")

Если вы профессиональный администратор баз данных, я вероятно должен вас предупредить. Если вы новичок в построении хранилищ данных, вы вероятно ужаснетесь. Возможно вы скажете: "Но если я знаю, что мой базовый ключ значит, мой опыт подсказывает мне что я лишаю ключа принимаемые мной данные." Да, при обработке транзакции продукции, значение ключа продукции или ключа покупателя напрямую связаны с содержимым записей. Однако в хранилище данных ключ размерности должен быть обощением того, что найдено в записи.

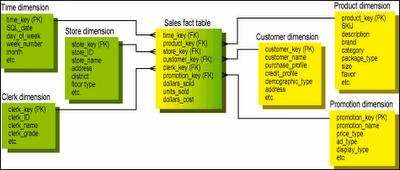
Как организатор хранилища данных, вы должны обеспечивать независимость ваших ключей от натуральных ключей продукции. Продукция имеет иные приоритеты для вас. Ключи продукции такие как ключи продукта, покупателя могут быть генерированы, форматированы, обновлены, удалены, переработаны или вновь использованы в зависимости от изменений продукции. Если вы будете использовать натуральные ключи продукции в качестве ключей хранилища данных, вы будете зависимы от малейших изменений, которые могут вас беспокоить, создать проблему и в худшем случае стать катострофическими для хранилища данных (прим. перевод). Представьте что вам надо хранить 3х летнюю историю продажи товаров в огромной таблице фактов продаж, но производство обновляет перечень продукции каждые 18 месяцев. Что вы будете делать? Давайте перечислим все пути по которым может пойти производство:

* Производство может использовать заново ключи с которыми вы работаете, как я описал.
* Производство может сделать ошибку и повторно использовать ключ даже не подозревая об этом. Это частенько случается в штрих-кодах при розничной торговле, иногда из лучших побуждений.
* Производство может пересмотреть пространство ключей с целью "сбора мусора" в системе. Одному из моих заказчиков недавно вручали целую ленту с повторно назначенными ключами покупателей продукции для загрузки в хранилище данных! (пирмерный перевод)
* Производство может вполне законно частично переписать описание товара или покупателя новыми значениями, но при этом не изменить ключ товара или покупателя новым значением. На вас сваливают ответственость, а вы не знаете что делать с измененными значениями атрибутов. Это проблема Медленно изменяющихся размерностей (SCD - Slowly Changig Dimensios), которую я сейчас, собственно, и раскрыл.
* Производство может изменить формат ключей для рашения ряд проблем системы транзакций. Теперь, например, целочисленные ключи стали текстовыми полями. Или представьте, что вы использовали 12ти байтный ключ, а теперь он стал 20ти байтным.
* Ваша компания только что претерпела слияние, и вам необходимо включитьболее миллиона новых клиентов в основной перечень заказчиков. Теперь вам необходимо извлекать данные из двух систем производства, но новая система имеет ключи клиентов неприводимые к вашим. (прим. перевод)

Проблема медленно изменяющихся размерностей, ранее упоминалась мной как широко известная ситуация в хранилищах данных. Вместо того чтобы проклинать производство за то что они не заботятся о ключах, лучше признать, что это сфера где интересы хранилища данных и производства расходятся. Обычно, когда администратор хранилища данных сталкивается с изменениями описания в записи размерности, типа товар или клиент, наиболее правильным является формирование новой записи размерности. Но для того чтобы это сделать, хранилище данных должно иметь более широкую систему ключей. Следовательно необходимо использовать суррогатные ключи. Медленно изменяющиеся размерности я описал в статье за апрель 1996 года (http://www.dbmsmag.com/9604d05.html). В следующей статье, я опишу низкоуровневую архитектуру для быстрого определения и обработки медленно изменяющихся последовательностей.

Еще есть много причн для использования суррогатных ключей. Одна из наиболее важных - необходимость идентифицировать неформализованные (случайные) сведения. Вам может понадобиться иметь ключ клиента для отображения транзакции, но клиент может оказаться анонимным. Возможно вы сформируете специальный ключ, который будет обозначать анонимных клиентов. Это веливо упоминается как "hack".

Если вы тщательно продумываете неизвестные ситуации типа "я не знаю", вы можете захотеть более чем просто один специальный ключ для анонимных клиентов. Вы также можете захотеть описать ситуацию где "идентификация клиента не имела пока места". Или, возможно, "это был покупатель, но система обработки данных некорректно вернула данные". А также "никакой клиент не возможен в этой ситуации". Все эти метаситуации вызывают ключ хранилища данных, который не может быть сформирован из ключей клиентов транзакции из производства. Не забудьте, что в хранилище данных вы должны сопоставить ключ покупателя для каждой записи фактов в схеме на рисунке.



Нулевой ключ автоматически вызовет предупреждение о нарушении целостности ссылочных данных в хранилище данных, т.к. внешний ключ (в таблице фактов) не может быть нулевым!  
Ситуация "я не знаю" возникает чаще всего с датами. Вероятно вы используете ключи образованные датой для объединения таблицы фактов и таблицы размерности. Еще раз, если вы так делаете, вы вынуждены использовать некоторую реальную дату даже для представления некоторой ситуации для которой определить дату не возможно. Я надеюсь, вы не используете 1 января 2000 года вместо неизвестной даты. Если вы так сделаете, вы получите ошибку 2000 года.

Возможно одна из причин, по которой Вы держитесь за ваши значащие ключи, созданные из реальных данных - то, что Вы думаете, что Вы захотите перемещаться по ключам непосредственно из приложения, избегая объединения с таблицей измерения. Время забыть об этой стратегии. Если 5й из 9ти символов в ключе объединения может быть интерпретирован как идентификатор производителя, то скопируйте эти символы и выделите их в нормальную колонку в таблице измерений. Еще лучше добавить наименование производителя отдельным текстовым полем. Финальным шагом следует обдумать исключение цифробуквенного идентификатора производителя в принципе. Единственная причина знать этот идентификатор конечным пользователям - использовать его для быстрого поиска в компьютерных системах.

Настояние на ключах в виде действительных дат также является стратегической ошибкой. Представьте, что вы имеете огромную таблицу фактов с миллиардом строк данных. В такой таблице каждый байт сэкономленный в строке экономит гигабайт носителя. Прелесть 4х байтного целочисленного ключа в том, что он может принять более 2 миллиардов различных значений. Этого достаточно для любой размерности, даже для супермегаразмерности, которая определит индивидуальность каждого человека. Таким образом мы сократим все наши длинные идентификаторы клиентов и все наши длинные идентификаторы товаров и все наши даты до 4х байтных ключей. Это сбережет гигабайты носителей.

Я считаю что конечной причиной для суррогатных ключей является то что я всегда во всем сомневаюсь и никогда не доказываю. Замещение громоздких безобразных натуральных и значащих композитных ключей прекрасными маленькими целочисленными суррогатными ключами максимально повышает операции объединения. Снизятся требования к носителям, а поиск по индексу упростится. Мне будет интересно услышать от кого-нибудь о реальном повышении производительности при замене больших ключей маленькими ключами типа integer.

Выбрав суррогатные ключи, теперь мы стоим перед проблемой их создания. В основном, каждый раз когда мы получаем натуральный ключ из входящего потока данных, мы должны найти корректное значениесуррогатного ключа и заменить натуральный ключ суррогатным. Мы должны максимально улучшить технологию для простого и скорейшего поиска данных, поэтому важнейшим шагом в ежедневном изъятии и преобразовании данных является этап подготовки данных. Следующую статью я посвящу искусству архитектуры суррогатных ключей.