**Суррогатные ключи. Конвейерная обработка суррогатных ключей**

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

**Хорошая система суррогатных ключей стоит трудов**

В прошлом месяце я заострил внимание на необходимости использования суррогатных ключей для каждой опреации объединения в хранилищах данных. Иными словами каждый ключ объединения между таблицей фактов и таблицей размерностей должен быть суррогатным ключом или независимым целочисленным числом, но не натуральным или значащим (зависящим) ключом. Значение суррогатного ключа должно начинаться единицей, второе значение должно быть два и так далее. Не должно быть никакой связи. Взглянув на суррогатный ключ вы не должны иметь возможности представить себе какие данные содержит идентифицированная им запись. Все суррогатные ключи представляются 4х байтным целочисленным (integer, int - прим. переводчика) числом (иногда даже 2х байтным для небольших размерностей), т.к. 4мя байтами можно представить более 2 млрд. записей размерности. В своей практике я не встречал таблиц размерностей соизмеримых с двумя миллиардаим записей.

В [прошлой статье](http://gocrm.ru/zamyslov/sk1.html) я отметил, что суррогатные ключи разрешают проблему администратора хранилищ данных в представлении медленно изменяющихся размерностей, в принципе как и представление неизвестных (unknown) или еще неопределенных (not-yet-recorded) значений размерности. В завершении следует отметить, что применение суррогатных ключей позволяет полностью контролировать хранилище данных, изолируя его от изменений в системе производства.

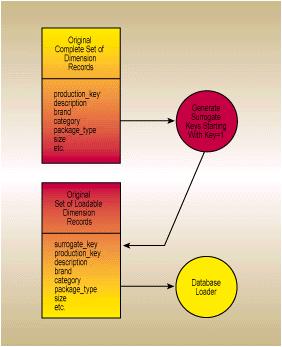
С момента создания записи в таблице размерности с корректным суррогатным ключом, каждая таблица фактов, ссылающаяся на это измерение должна содержать соответствующий суррогатный ключ. В таком случае суррогатный ключ в таблице фактов становится внешним ключом. Определите для себя таблицу размерностей задающей значения ключей ("keymaster") для измерения. Таблица измерения определяет верность суррогатного ключа, т.к. он является первичным ключом таблицы измерения.

Когда каждый суррогатный ключ в таблице фактов является корректным внешним ключом соответствующим первичному ключу таблицы размерности, тогда устанавливается ссылочная целостность между таблицей фактов и таблицей измерения. Таблица фактов с четырьмя и 12ю внешними ключами к ореолу (окружению) таблиц размерностей представлют интересную задачу импорта данных: Вам придется перехватить все входящие записи фактов и заменить все элементы внешних ключей корректными суррогатными ключами, причем с высокой скоростью.

**Ключи для таблиц измерений**

Давайте займемся разбором задачи создания первичных суррогатных ключей сначала в таблицах размерности, а затем разрешением быстрой генерации ключей для таблицы фактов.

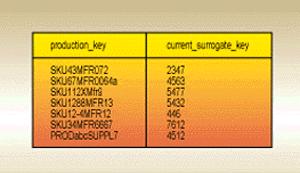
Когда мы говорим о создании ключей таблиц размерностей, мы должны выделить первичную загрузку (импорт) таблицы размерности от последующих. Первичная загрузка (импорт) скорее всего создаст число записей точно совпадающих с числом записей необзодимых для определения данных производства. Преположим что данные производства точны, достоверны и не содержат дублирующихся записей. Исходя из этого, просто добавляем последовательсноть суррогатных ключей по мере первичной загрузки размерности. Этот процесс является простым последовательным чтением входящих данных как показано на рисунке.



Интересное начинается после первичной загрузки размерности. Следующий раз, как и все последующие, вы читаете данные, определяющие размерность и вам придется принимать некоторые сложные решения. Простейшим решением является распознание новых записей, добавленных в исмерение в систме производства, и формирование для них новых натуральных ключей. Проще говоря представьте что данные производства имеют корректное тодельное поле назваемое ключом продукции. (Смысл не меняется, если уникальность продукции определяется рядом отдельных полей) Каждый раз при импорте данных размерности продукции, вам достаточно проверить все ключи продукции на предмет того, обрабатывались ли они ранее. Строго говоря вам достаточно просмотреть таблицу рамерности хранилища данных где натуральный ключ хранится просто как обычное поле. Сразу рекомендую использовать отдельную таблицу поиска для этих целей, а не использовать реальную таблицу размерностей в хранилище данных.

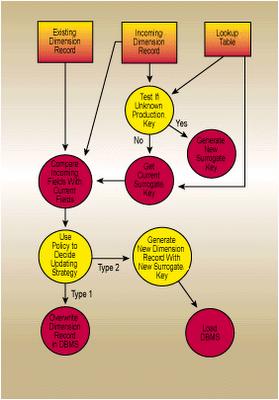
При анализе импортируемых данных определяющих размерности, наиболее сложным является случай, когда запись с данным натуральным ключом уже импортирована, но в системе производства некоторые поля этой записи изменились. Это классический пример медленно изменяющихся размерностей; например товар может претерпеть небольшие изменения описывающих упаковку или состав, но артикул товара (клюя товара) не изменился, или клиент имеет тот же идентификатор, но некторые сведения о нем, такие как семейное положение, изменились. Для разрешения этой проблемы у вас должна быть сформирована политика медленно изменяющихся измерений. Политика гласит что если определенное (достоверное или неизменяющееся - прим. переводчика) описательное поле размерности изменяется, данные в записи полностью переписываются (так называемые изменения Типа 1) (SCD-1 - прим. переводчика). Но если изменяются другие описательные поля, такие как семейное положение, формируется новая запись размерности (изменения Типа 2) (SCD-2). Политика определяет какие поля должны переписываться а какие должны формировать новую запись. В результате политика должны быть реализована в механизме импорта в хранилище данных.

Наиболее быстрый способ определить встречался ли натуральный ключ ранее - сформировать специальную сортированную и индексированную таблицу ипортированных натуральных ключей для быстрого поиска при импорте.



Эта таблица будет содержать меньше строк чем в таблице измерения хранилища данных, т.к. будет содержать только одну строку для каждого импортированного натурального ключа. Кроме натурального ключа, таблица поиска будет содержвать наиболее свежий суррогатный ключ для сопоставления этому натуральному.

Каждое утро когда приходят данные для импорта в таблицу размерности следует использовать таблицу поиска для определения новых и старых ключей. Если ключ оказался новым, вы однозначно знаете, что вам необходимо сформировать новую запись размерности. Если достигли максимального значения суррогатного ключа ранее использованного в текущем измерении, вам достаточно добавить единицу к этому значению при создании новой записи измерения. Затем сравните значения каждого поля записи текущей версии из хранилища данных с входящей записью. Если все совпадает, то это не медленно изменяющаяся размерность (далее SCD - прим. переводчика), и можно переходить к следующей записи. Если одно или несколько полей были изменены, необходимо применить политику SCD для решения изменить или создать новую запись в таблице измерений. Если вы создаете новую запись, следует внести коррективы в таблицу поиска, т.к. сформировался новый суррогатный ключ для представления текущего натурального ключа.



Если вам повезло и данные производства поступают с штампом времени (timestamp), то можно заменить трудоемкий процесс сравнения полей на контроль даты. Однако остальная часть логики процесса останется неизменной.

Существует еще одна интересная техника обработки SCD-2, которая заключается в том, что мы имеем три поля в таблице измерения в дополнение к суррогатному ключу, необходимому для основных операций. Эти три поля: дата вступления в силу (Effective datetime), следующее изменение (NextChange datetime), Текущая транзакция (Effective transaction). Эти поля позволяют делать чрезвычайно эффективные срезы таблицы измерений для представления размерности в любой момент времени. Я раскрыл использование этих полей в статье за февраль 1998 года Human Resources DataMarts (http://www.dbmsmag.com/9802d05.html).

**Ключи таблицы фактов**

Перед тем как перейти к обработке таблицы фактов, следует сначала обработать таблицы измерений. После обновления таблицы измерений, кроме того что все записи измерений корректны, но и поисковая таблица связывающая натуральный ключ с текущим суррогатным ключом хранилища данных должна быть также корректно обновлена. Эта маленькая моисковая табличка чрезвычайно важна для быстрой обработки фактов.

Задача для обработки входящих фактов предельно проста. Берем каждый ключ размерности в таблице фактов и заменяем его корректным текущим суррогатным ключом. Обратите внимание, я сказал "заменяем". Не храните значение натурального ключа внешней системы обработки данных непосредственно в таблице фактов. Если вас беспокоит проблема определения натурального ключа для этой записи, вы всегда сможете его определить из связанной с фактами таблицы размерностей.

Если у вас от 4 до 12ти натуральных ключей в записи факта, это может потребовать соответственно от 4 до 12 отдельных запросов поиска для определения суррогатных ключей.

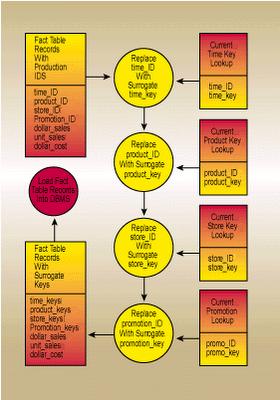


Рисунок показывает способ который я использую для ускорения этой процедуры. Создайте многопоточное приложение, которое проведет все записи по шагам, которые указаны в рисунке. Когда я сказал многопоточное, я имел в виду, что пока обрабатывается запись №1 на предмет замены ключей, запись №2 сразу за записью №1 и так далее. М ы не будем обрабатывать все входящие записи за первый шаг поиска а затем целиком отправлять на следующий шаг. Для повышения производительности важно, чтобы входящие записи не записывались до тех пор, пока не пройдут все шаги обработки. Они могут как бы "пролетать" по памяти не касаясь земли (диска) пока не будут до конца обработаны.

Если это возможно, следует все поисковые таблички поместить в памятьчтобы можно было моментально получить доступ к любой записи и вернуть суррогатный ключ для замены. Это, кстати говоря, одна из причин, по которой стоит отделять поисковые таблицы от реальных таблиц размерностей хранилища данных. Представьте вы имеете поисковую таблицу на миллион строк для таблицы измерений. Ключ натуральный ключ состоит из 20ти байт и суррогатный 4 байта, в таком случае вам необходимо примерно 24Мб ОЗУ для хранения поисковой таблицы. В случае если у вас есть возможность использовать для обработки ЭВМ с ОЗУ 256Мб и больше, вы вполне можете разместить все ваши поисковые таблицы в памяти.

В некоторых важных больших таблицах фактов вы можете столкнуться с супер размерностями ("monster dimension") такими как размерности коммунальных потребителей колличество которых может составлять десятки миллионов. Если у нас только одно такое гигантское измерение, то всеравно можно построить производительную систему обработки (подстановки) суррогатных ключей, не смотря на то что придется считывать данные с диска. Главным образом благодаря предварительной сортировке как входящего потока фактов, так и поисковой таблицы натуральных ключей. Теперь подстановка суррогатных ключей сведена к однопроходной задаче подстановки данных из двух отсортированных файлов. Это должно работать довольно быстро без особых затрат ОЗУ. Если же у вас два таких суперизмерения, следует консультироваться с экспертами.

Над конструированием быстрой суррогатной системы очевидно придется поразмыслить, но это принесет свои плоды. Качественная система суррогатных ключей сэкономит драгоценное место в таблице фактов, избавит вас от административных проблем, связанных с изменениями на производстве, потенциально обезопасит вас от таких сюрпризов как раздел или слияние компании, предоставит гибкий механизм для обработки медленно изменяющихся измерений и избавит от неуверенности в натуральных ключах.