Декларативный подход к описанию трансформации данных с разрешением конфликтов обновления

Ю.В.Стадник

Хмельницкий государственный университет, 29016,г.Хмельницький ул.Институтская,11, т.(03822)2-13-56,e-mail: yura@dpa.km.ua

Преобразование данных является одной из основных проблем при наличие множественных источников данных. В работе рассматривается решение задачи однозначного сопоставление данных из нескольких баз источников одной базе получателю с использованием правил разрешения конфликтов обновления. Для описания схем баз данных и процедур трансформации используется декларативный подход. Предлагается алгоритм генерации обновлений целевой базы данных.

Многие программные системы используют информацию из различных источников данных, в которых связанные данные представляются по-разному. Например, в систему хранения данных могут добавляться структуры данных, которые отражают новые аспекты представления существующих в системе сущностей. Данные в новом источнике могут не соответствовать существующей схеме хранения. Новые данные могут представлять информацию избыточно, может отличаться формат хранения. Кроме этого, бывает необходимость представлять данные в формате, необходимом определенным приложениям. Но наиболее распространенным примером необходимости трансформации данных является наличие нескольких оперативных баз, которые служат источником для формирования срезов оперативной информации представленные только отдельными, наиболее значимыми, атрибутами.

Обзор основных решений задачи трансформации данных приведен в [4].

Цель данной работы показать реализацию части концептуальной схемы данных, процесса трансформации данных, используя язык логики предикатов, а так же показать возможность хранения правил формирования реляционных предикатов в базе данных.

Под преобразованием данных будем понимать изменение формата данных, слияние или разбиение переменных отношений базы данных для формирования необходимого представления. Преобразование структуры базы данных и самих данных будем рассматривать как этапы единого процесса.

Задача трансформации данных состоит в отображении одного экземпляра базы данных источника на целевую базу данных. Схемы баз данных источника и приемника различны. При этом может существовать несколько экземпляров баз данных источников. Отображение между двумя базами должно однозначно определить соответствие экземпляров переменных.

Для определения преобразования базы данных представим схему и данные целевой базы данных и баз данных источников в виде общей модели (метаданных). В этой модели должны отражаться основные понятия реляционной базы данных: схемы отношений, значения отношений, ограничения, домены. В модель включим правила вывода, служащие для данных целевых отношений.

Понятие отношения реляционных и дедуктивных баз данных отождествляется n-арным предикатом P[1]:

$$P(t_1,...t_n),$$

где P – некоторый реляционный предикат, t – значения атрибутов отношений - значения простого типа данных – строка, число – принадлежащих домену D.

Схема отношения описывает информацию , которая не зависит от значения отношения . эта информация состоит из имени отношения и множества пар атрибут/домен.

$$r(A_1 / D_1,...A_n / D_n),$$

где $A = (A_1,...A_n)$ – атрибуты отношения. D – домены, необязательно разные.

Ограничения (Constraints) являются частью определения и оптимизации преобразования, в то же время ограничения используются при реализации преобразования. Используя [1], определим следующие типы ограничений, которые используются для реализации трансформации данных.

Первичный ключ определим как

$$Pk(P,X) \sim \forall x \forall y \forall z (P_1(x_1,..x_n, y_1,..y_n) \wedge P_1(x_1,..x_n, z_1,..z_n) \supset (y_1 = z_1) \wedge (y_2 = z_2)..(y_n = z_n) \wedge (x_1 \neq \varpi) \wedge (x_2 \neq \varpi)..(x_n \neq \varpi))',$$

где символ ϖ, представляет некоторое неизвестное значение (необязательно одно и то же). Т.е. множество значений атрибутов $x_1,..x_n$ - уникально определяет предикатную переменную Р, при этом значения атрибутов $x_1,..x_n$ известно.

Ссылочное ограничение (внешний ключ)

$$Fk(P_1, X, P_2) \sim \forall x \exists y (P_1(x_1,..x_n) \supset ((P_2(y_1,..y_n, z) \wedge (x_1 = y_1) \wedge (x_2 = y_2)..$$
$$(x_i = y_i) \wedge Pk(P_2, y_1,..y_n))) \vee ((x_1 = \varpi) \wedge (x_2 = \varpi)..(x_i = \varpi))$$

Если атрибуты $x_1,..x_n$ отношения P_1 ссылаются на атрибуты отношения P_2, то совокупность этих атрибутов в отношении P_2 является первичным ключом. Атрибуты $x_1,..x_n$ отношения P_1 могут принимать значения определенные в первичном ключе отношения P_2, или неопределенные значения.

Ограничения на значения по умолчанию.

$$Ch(P_1, x_i, Const) \sim \forall x (P_1(x_i) \wedge x_i \notin D_i \supset P(Const / D_i))$$

Если значение x_i атрибута A_i не равно ни одному значению из домена, определенного для этого атрибута, то по умолчанию значение равно $Const$.

Частным случаем ограничением по значению может являться ограничение на невозможность внесения пустого (неизвестного) значения ϖ.

Указанные выше понятия имеют соответствие с сущностями, поддерживаемыми реляционными системами управления базами данных (СУБД). Так, в соответствие понятию предикат ставится таблица, ограничению - ограничение целостности базы данных (integrity constraint), Символу ϖ, представляющему неизвестное значение – константа $Null$, представляющая отсутствующее, неприемлемое или пустое значение.

Обновление экстенсиональной части базы – операций добавления, изменения , модификации значений в отношениях – происходит гораздо чаще чем обновление ее схемы. Обновление базы данных приводит к переходу базы из одного состояния в другое. При этом изменение значений в одной группе отношений (базе-источнике) вызывает изменение значений в другой группе (целевой базе). Эти изменения определяются правилами, заданными в интенсиональной части базы данных. Для осуществления обновления базы данных система должна осуществить вывод всех необходимых для текущего состояния предложений, присоединяя к текущему состоянию предложения, выражающие изменения в предметной области.

В рамках рассматриваемой задачи не каждое обновление базы-источника влияет на целевую базу данных. Значения атрибутов отношений целевой базы зависят от значений атрибутов отношений базы источника. Будем говорить, что множество значений атрибутов отношения P_1 определяет множество атрибутов отношения P_2, если существует хотя бы один атрибут в отношении P_1, от которого зависит значение хотя бы одного атрибута отношения P_2.

$P_1 \xrightarrow[a=b]{} P_2$, Значение атрибута a предиката P_1 определяет значение атрибута b предиката P_2 .

Определим событие изменения (events) при необходимости изменения целевой базы данных.

$$Events(P_1, P_2) \sim \forall x \exists y (P_1(x_1..x_n) \supset P_2(y_1..y_i) \wedge ((x_1^b \neq x_1^a) \vee$$
$$(x_2^b \neq x_2^a)..(x_n^b \neq x_n^a)) \wedge P_1 \xrightarrow[x_n = y_i]{} P_2)$$,

т.е. если отношение P_1 определяет отношение P_2, и значения какого либо атрибута x_i^b отношения P_1 до модификации не равно значению этого атрибута x_i^a после модификации, то система генерирует событие. При каждом обновлении базы источника система проверяет, затрагивались или нет атрибуты отношений, которые влияют на целевые отношения. Только в случае если затрагивались – система начинает трансформацию данных.

Наличие нескольких отношений источников вызывает необходимость указать правила формирования целевых атрибутов. Эти правила должны однозначно определять отношение баз данных источников, из которого должен быть получен целевой атрибут.

Рассмотрим возможные преобразования атрибутов. Преобразования атрибутов определяются мощностью сопоставления [4]. В [4] определены следующие типы преобразования атрибутов: 1:1, n:1, 1:n и n:m. Преобразование 1:1 имеет место, когда атрибуту целевого отношения строго соответствует атрибут отношения источника. Преобразование 1:n, n:1 возникает тогда, когда существует функциональная зависимость между атрибутом целевого отношения и несколькими атрибутами отношения источника или нескольких атрибутов

целевого отношения и атрибутом отношения источника соответственно. Преобразование n:m возникает при необходимости одновременного получения нескольких атрибутов целевого отношения из нескольких атрибутов отношения источника. Преобразования такого рода, требующие дополнительной информации о структуре исходного и целевого отношений, будут рассмотрены ниже.

Использование информации об ограничениях, которые накладываются на отношения, позволяет получить дополнительную информацию для преобразования данных. Информация о первичном ключе дает возможность определить перечень полей для определения типа операции – добавления, удаления, или обновления – целевого отношения. Использование ограничений по умолчанию дает возможность определить несформированные значения атрибутов первичного ключа. Использование информации, извлеченной из описания ограничений ссылочной целостности, позволяет корректно реализовывать соответствие между несколькими отношениями-источниками и целевым отношением. Построения текста SQL запросов для извлечения данных из нескольких отношений предлагается использовать информацию о ссылочных ограничениях. Определим $Link$ как последовательность ссылочных ограничений между отношениями:

$$\forall A \forall B \forall X\,(Fk(A, X, B) \supset Link(A, B));$$
$$\forall A \forall B \forall C \forall X\,(Link(A, C) \wedge Fk(C, X, B) \supset Link(A, B));$$

где A,B,C – некоторые отношения, X – атрибуты, участвующие в ссылочном ограничении.

Для построения запросов к нескольким отношениям используются метаданные, описывающие схему базы.

Рассмотрим следующие варианты формирования данных целевого отношения с учетом различий в структуре исходного и целевого отношений.

1) Формирование нескольких переменных целевого отношения из одной переменной отношения источника представлено на рисунке 1. В целевое отношение вводится дополнительный атрибут (Pk2), входящий в первичный ключ. Правило вывода для примера на Рис.1.

$$Td(Pk, f1, f2, c1) \leftarrow Ts(Pk, f1, f2, f3, f4)$$
$$Td(Pk, f3, f4, c2) \leftarrow Ts(Pk, f1, f2, f3, f4)$$

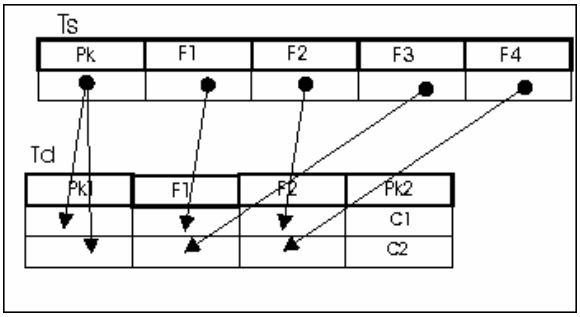

Рис.1. Схема формирование нескольких переменных целевого отношения из одной переменной отношения источника

2) Для формирования целевого отношения используются несколько переменных отношения источника. В целевое отношение попадает часть первичного ключа отношения

$$Td(Pk, f1, f2, f3, f4) \leftarrow (Ts(Pk, Pk2, f1, f2) \wedge (Pk2 = c2)) \wedge (Ts(Pk, Pk2, f3, f4) \wedge (Pk2 = c3))$$

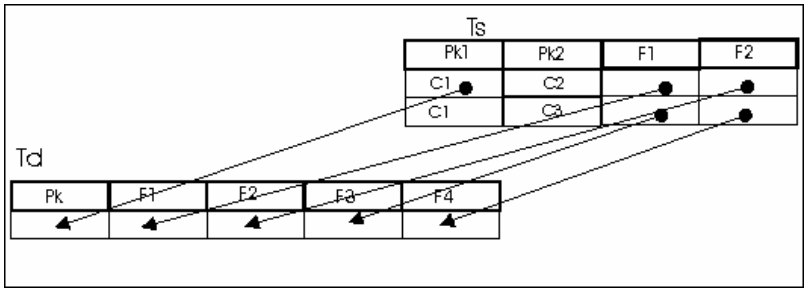

Рис.2. Схема формирования целевого отношения используются несколько переменных отношения источника

источника. Оставшаяся часть первичного ключа источника используется для выбора соответствия полей между источником и получателем.

3) Формирование целевого отношения из данных нескольких отношений источников подразумевает объединение переменных отношений источников (рис.3).

$$Td(Pk,\ f1,\ f2) \leftarrow Ts1(Pk, k1, k2) \vee Ts2(Pk, l1, l2)$$

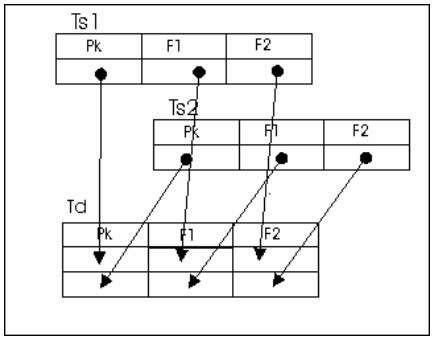

Рис.3. Схема формирование целевого отношения
из данных нескольких отношений источников

Каждый из вариантов отличается формированием ключевых атрибутов целевого отношения. При формировании ключевых атрибутов целевого отношения необходимо руководствоваться дополнительной информацией, которая извлекается из ограничений, накладываемых на отношения .

Для преобразования данных необходимо привести все данные к единому формату. При этом часть данных может быть утеряна. Например: в отношении-источнике атрибут, предназначенный для хранения информации о семейном статусе физического лица, может быть представлен символьным, логическим или строковым типом данных. В целевом отношении тип соответствующего атрибута может не совпадать с типом источника, что, в общем случае, приведет к ошибке преобразования типа. Более детально вопросы очистки данных рассмотрены в [5].

База данных эволюционирует во времени в результате операций обновления схемы (например, операций порождения и удаления отношений и ограничений) . Для описания базы данных введем следующие отношения: Ent(Tn) – описание отношений, Tn – название отношения, Attrib(A,D,Tn,FTn,FAt) – описание атрибутов отношений, A – название атрибута, D – тип данных, Tn – название отношения, которому принадлежит атрибут, FTn – название отношения, на которое ссылается атрибут, FAt – название атрибута в отношении FTn. Описание ограничений, накладываемых на отношения - Constr(Cn,Tn,Tc,A,NA), Cn – название ограничения, Tn – отношение, которому принадлежит ограничение, Tc – тип ограничения – первичный ключ или ограничение на значение атрибута, A – название атрибута в отношении, NA – порядковый номер атрибута в ограничении.

Для описания соответствия атрибутов для выполнения трансформации введем в систему отношение $\mathrm{Trans(Tn,At,\ IdP,P,U)}$,где Tn – идентификатор отношения, At - атрибут отношения, IdP – идентификатор, указывающий на какие атрибуты отображается атрибут At отношения Tn, P - некоторое правило, необходимость в котором возникает если существует несколько источников формирования целевого атрибута. Это правило позволяет сгенерировать ранг для построения порядка перебора источников. В наиболее простом случае в атрибуте P задается приоритет источника для получения атрибута целевого отношения. U - является неквантифицированой формулой первого порядка, а переменные, входящие в эту формулу, являются атрибутами переменных отношения Tn. Пример использования отношения Trans: получение перечня отношений, атрибутов и условий, которые участвуют в формировании некоторого атрибута A отношения T, определяется формулой реляционной алгебры

$$\Pi_{\mathrm{Tn,At,P,U}}((\Pi_{IdP}(\sigma_{(Tn=T)\wedge(At=A)}\mathrm{Trans})) \underset{IdP=IdP}{\rhd\lhd} \mathrm{Trans}),$$

где $\Pi, \sigma, \rhd\lhd$ - операторы проекции, выбора и соединения соответственно.

Или в формате SQL

```
SELECT Tn,At,P
FROM
(SELECT IdP FROM TRANS WHERE TN=T and At=A) A,
TRANS B
WHERE A.IdP=B.IdP
```

Зависимости между отношениями, описывающими схему базы данных и правила преобразования представлены на рисунке 4 в формате UML.

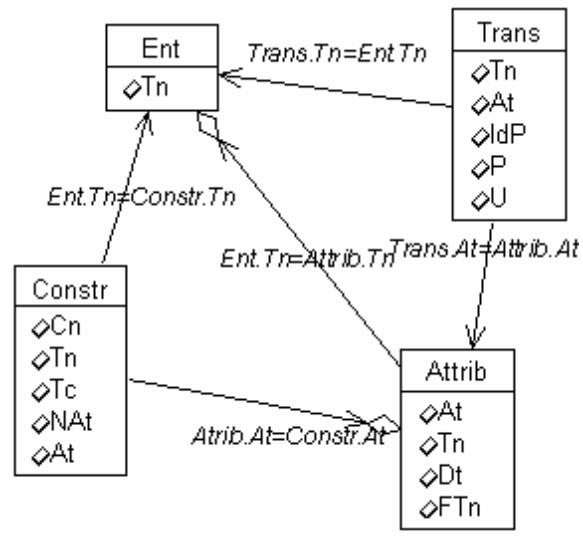

Рис.4. Зависимости между отношениями, описывающими схему базы данных

Используя вышеизложенное как метаданные, т.е. данные, которые описывают предметную область, покажем реализацию трансформации данных из базы источника в целевую базу данных.

Рассматриваются отношения находящиеся в третей нормальной форме (ЗНФ), т.е. отношения, у которых каждый кортеж состоит из значений первичного ключа, которое идентифицирует некоторую сущность, и набора значений взаимно независимых атрибутов (каждый атрибут может быть обновлен независимо от остальных), некоторым образом описывающих эту сущность. Если отношения-источники менее нормализованы, возникает необходимость расширять отношение Trans дополнительными атрибутами, указывающими на зависимости между атрибутами в отношениях-источниках и целевом отношении.

Решение задачи предполагает два варианта: формирование каждой переменной целевого отношения и формирования текста запроса в формате SQL или в терминах реляционной алгебры для получения набора переменных целевого отношения. Безусловно, второй подход является более эффективным, т.к. носит описательный характер и предполагает меньшее количество обращений к данным.

Алгоритм формирования целевых наборов данных.

Входными данными является $S(A_1 / c_1, A_2 / c_2 .. A_n / c_n)$ переменная отношения базы источника. Выходными данными являются команды в формате SQL для обновления, добавления, удаления данных из целевых отношений.

1. Используя Trans, определяем какие целевые отношения затрагивает отношение источник, в которо е внесли изменение. Список целевых отношений сохраним в TblList

2. For i:=1 to TblList.count do

2.1. Используя Trans, определяем отношения-источники, атрибуты которых влияют на формирование первичного ключа TblList[i]. Список отношений сохраним в TblAtList

2.2. Сортируем источники формирования первичных ключей TblAtList в зависимости от условия P для определения наиболее приоритетных источников формирования.

2.3. For j:=1 to TblAtList.count do

2.3.1.Проверка наличия данных в отношениях-источниках, используя ограничения ссылочной целостности и входные данные отношения источника;

2.3.2. If данные найдены then

2.3.2.1. Формируем строку SQL запроса на извлечение атрибутов, влияющих на поля входящие в первичный ключ и атрибутов, которые извлекаются из TblAtList[j] отношения;

2.3.2.2. Выход из цикла 2.3;

2.4. Определяем источники формирования остальных атрибутов, формируя строку SQL таким образом, чтобы получить корректный запрос с учетом соответствия структур.

2.5. После выполнения запроса и сравнения полученного набора данных с данными, имеющимися в целевом отношении с учетом ключевых реквизитов, генерируем обновление целевого отношения (insert, update,delete).

3. End

Использование алгоритма подразумевает обновление данных целевого отношения сразу после обновления данных отношения источника, или групповую отложную обработку всех обновлений отношений источников за определенный период времени.

Следует отметить, что предложенный алгоритм преобразования требует минимальной настройки, если схемы баз источника и получателя имеют иерархическую структуру. Значительное количество схем баз данных в действующих системах имеют такую топологию.

Использование указанного подхода с использованием метаописания схем баз данных позволяет производить трансформацию между разнородными источниками данных, в том числе и такими, которые не поддерживают ограничения первичных ключей и ссылочной целостности. Таким образом, можно корректно проводить обновление целевой базы данных информацией из форматированных текстовых источников, файлов в XML формате и пр.

1. Андон П.И.,Яшунин А.Е., Резниченко В.А. .Логические модели интеллектуальных систем. Киев.Наукова думка.1999.
2. Андре Тейз,Паскаль Грибомонд Логический подход к искусственному интеллекту. «Мир» Москва 1998.
3. Susan B. Davidson and Anthony S. Kosky. Specifying Database Transformations in WOL Data Engineering 1999 Vol22No1 p.25-31.
4. Erhard Rahm ,Philip A.Bernstein. A survey of approaches to automatic schema matching, The VLDB Journal 2001, 334 350.
5. Erhard Ram,Hong Hai Do. Очистка данных: проблемы и актуальные подходы http://www.olap.ru/ .