# СПОСОБ ПОЛУЧЕНИЯ ИНФОРМАЦИИ ИЗ МНОЖЕСТВА ИНТЕГРИРОВАННЫХ БАЗ ДАННЫХ

**А.Ю. Куликов**

Рассматривается проблема интеграции баз данных (БД), которая состоит в том, чтобы обеспечить выполнение запросов на извлечение данных из их совокупности, в форме, не зависящей от физического расположения и способа хранения данных [1]. В настоящее время весьма важными стали виды профессиональной деятельности, для которых требуется оперативное получение достоверной информации из многочисленных функционирующих автономно и слабо связанных друг с другом источников. Такие виды деятельности обнаруживаются в сферах государственного и корпоративного управления, планирования и контроля, финансов, торговли, медицины и многих других.

Ввиду этих потребностей, востребованной представляется задача *массовой интеграции* – когда число интегрируемых баз данных велико (десятки и сотни единиц), и их состав динамически меняется. Существующие решения от ведущих производителей программного обеспечения нацелены на интеграцию относительно небольшого количества баз данных в рамках одной организации, поэтому не решают некоторых задач, которые возникают при интеграции в более крупных масштабах.

Существуют две основные архитектуры, которые чаще всего используются при построении систем интеграции БД: физическая и виртуальная (*федеративное объединение*) [2]. Сравнив недостатки и преимущества этих архитектур, можно сделать вывод о возможности использования федеративного объединения для построения системы, осуществляющей массовую интеграцию. При интеграции БД с различной структурой хранимой информации, возникает задача интеграции самих данных. Одно из возможных решений заключается в определении унифицированного представления данных и задании отображений, которые позволяют привести данные в различных БД к этому единому представлению. При этом предполагается, что поисковые запросы, которые оперируют с данными в унифицированном представлении (*унифицированные запросы*), система интеграции будет преобразовывать в запросы к конкретным базам данных (*конкретные запросы*), используя соответствующие *правила отображения*.

Обычно унифицированным представлением данных служит глобальная схема, а основными методами задания отображений для интегрируемых БД являются: Global As View (GAV) и Local As View (LAV) [3]. Однако оба этих метода имеют существенные недостатки. Основная проблема при использовании GAV — необходимость редактирования правил отображения для таблиц глобальной схемы при добавлении новой БД. Проблема LAV – это высокая вычислительная сложность процесса построения конкретных запросов по унифицированному запросу. Кроме того, согласно GAV и LAV методам, правила отображения связывают таблицы глобальной схемы с определённой совокупностью БД. Таким образом, в чистом виде классические способы интеграции данных не позволяют работать с произвольными подмножествами БД. Такая возможность работы с любым подмножеством источников данных не столь востребована при интеграции в небольших масштабах, но становится весьма важной в задаче массовой интеграции. Например, при сборе статистики по институтам в масштабах страны может потребоваться предоставлять данные как по всем интегрированным БД, так и по БД какого-либо региона или из отдельного учреждения. Для решения проблем, возникающих при массовой интеграции, предлагается использовать подход GAV с ограничением на вид отображений.

В подходе GAV правила отображения схем глобальных таблиц в схемы источников имеют вид $G \rightarrow Q$, где G – элемент глобальной схемы (таблица в реляционной модели), Q – запрос к некоторому числу баз данных. Предложение состоит в том, чтобы представить данные, соответствующие глобальным элементам, в виде совокупности разделов, каждый из которых содержит данные только из одной базы, а глобальный элемент является объединением разделов одного типа. При этом исчезает основной недостаток подхода GAV — необходимость редактирования правил отображения для таблиц глобальной схемы при добавлении новой БД. Поскольку наборы правил, относящиеся к разным разделам и к разным источникам, независимы друг от друга, включение в инфраструктуру нового источника сводится к добавлению правил отображения его разделов.

Следующим шагом для обеспечения возможности работы с произвольным множеством источников данных является введение понятия *группа баз данных*. Группа БД представляет собой произвольное подмножество интегрируемых баз данных. При этом в рамках каждой группы однозначным образом определены тела таблиц глобальной схемы. Если группа представляет собой множество Grp: $\{Di \mid i=1..N\}$, где Di – БД, N – количество БД в группе, то множество данных, соответствующих таблице глобальной схемы GT: $GT(Grp) = \cup Qi(Di)$, $i=1..N$, где Qi – запрос, который выбирает данные элементарного раздела из базы данных Di, соответствующие схеме таблицы GT, символ $\cup$ обозначает операцию Union.

Чтобы работать с произвольным множеством интегрируемых БД, нужно объединить их в группу, а затем использовать эту группу в запросах. Таким образом, составление запросов предлагается выполнять в два этапа:

1. определение множеств БД, из которых необходимо получить информацию (с помощью задания групп);
2. составление запроса с использованием этих групп.

Число интегрируемых источников в случае массовой интеграции велико, поэтому определяя группу, невозможно пользоваться сведениями об адресах отдельных БД, тем более что их состав может меняться. Поэтому для определения группы БД необходимо исходить только из содержательной модели информационного пространства. То есть осуществлять отбор БД группы по метаданным (метаатрибутам), которые задаются для каждой интегрируемой базы данных. Примерами таких метаатрибутов являются: название организации-владельца, адрес организации, тематика данных и т. д.

Для составления запросов с использованием групп баз данных используется несколько измененный язык SQL. Поскольку тела таблиц глобальной схемы определены лишь в рамках конкретной группы, то для доступа к таблицам используется конструкция <имя группы>.<имя глобальной таблицы>. Если задана группа SP={D1, D2..Dn}, и существует глобальная таблица GT, то имена вида SP.GT интерпретируются как объединение данных типа GT из всех баз данных группы SP: D1_GT ∪ D2_GT … ∪ Dn_GT. Например, запрос:

SELECT select_expressions FROM SP.GT

заменяется системой интеграции следующим образом:

SELECT select_expressions FROM (D1_GT ∪ D2_GT … ∪ Dn_GT),

где Di_GT адресует раздел таблицы GT в базе данных Di. Для каждого раздела Di_GT согласно подходу GAV задан соответствующий ему запрос к Di, который выбирает набор кортежей, удовлетворяющих схеме таблицы GT.

Приведём пример запроса с использованием групп. Пусть заданы группы баз данных SP1 и SP2, а так же таблица глобальной схемы people с полями (name, address, passport) и таблица cars с полями (regnumber, ownerpassport), содержащие информацию о людях и автомобилях соответственно. Тогда запрос, который выполняет поиск людей с фамилией ‘Сидоров’ из группы SP1 и их автомобилей из группы SP2 может быть записан следующим образом:

```
SELECT SP1.people.name, SP1.people.address, SP2.cars.regnumber
FROM SP1.people LEFT JOIN SP2.cars ON SP1.people.passport = SP2.cars.ownerpassport
WHERE SP1.people.name LIKE ‘%Сидоров%‘
```

Наряду обычными данными в запросах могут использоваться значения метаатрибутов БД в форме: <имя группы>.<имя метаатрибута> Если для базы данных DB группы SP задан метаатрибут M, который имеет значение V, то выражение SP.M для кортежей, которые выбираются из базы данных DB, будет иметь значение V.

Эта возможность позволяет, например, неявным образом задавать группы. Так, запрос:

SELECT SP1.persons.name FROM SP1.persons WHERE SP1.orgname = ‘ИПМ РАН’

выбирает данные из группы SP1, в которую попадают все интегрируемые БД, для которых метаатрибут orgname имеет значение ‘ИПМ РАН’.

В ИПМ им. М.В. Келдыша РАН разработан прототип системы, реализующий изложенные идеи.

ЛИТЕРАТУРА:


1. L. M. Haas, E. T. Lin, M. A. Roth. Data integration through database federation. IBM Systems Journal, Volume 41, Issue 4 (October 2002), p. 578 – 596. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.93.2247&rep=rep1&type=pdf
2. Wiederhold, G. (1992). Mediators in the Architecture of Future Information Systems. IEEE Computer, 25(3): 38-49.
3. M. Lenzerini. Data Integration: A Theoretical Perspective, PODS 2002. pp. 233–246. http://www.dis.uniroma1.it/~lenzerin/homepagine/talks/TutorialPODS02.pdf