Р. М. Ганопольский, Д. Б. Кепещук

ПРЕДСТАВЛЕНИЕ ЗНАНИЙ В ГЕТЕРОГЕННЫХ РАСПРЕДЕЛЕННЫХ БАЗАХ ДАННЫХ НА ПРИМЕРЕ ИНГРИС ТюмГУ

Рассматриваются проблемы разработки и функционирования распределенных баз данных с гетерогенными составляющими. Предлагаются решения на основе специального сервера-коммутатора. Формулируются задача и план научного исследования.

Состояние проблемы

Современные вузы активно внедряют информационные технологии для поддержки образовательной деятельности. По мере наращивания информационных ресурсов встает задача их взаимной интеграции во избежание избыточной работы по наполнению систем информацией, сходной по содержанию, а также для повышения оперативности наполнения ресурсов информацией [10]. Таким образом, разрозненные информационные системы и базы данных объединяются в распределенные базы данных (РБД). Так как структура баз данных, входящих в состав РБД, различна, будем называть последние гетерогенными распределенными базами данных.

Аналогичная ситуация наблюдается и в нефтегазовом комплексе [4]. Предприятиями нефтегазовой промышленности и сторонними компаниями разработано либо находится в стадии разработки или модернизации огромное количество систем по сбору, накоплению и анализу разнообразных данных. Многие системы работают с информацией сходного содержания, и эффективность применения подобных систем значительно повышается при их взаимной интеграции. Как и в рассмотренном выше случае, происходит объединение разрозненных систем в распределенные базы данных с гетерогенными составляющими.

Выделим основные проблемы гетерогенных РБД [5, 10]:

— синхронизация данных;

— обеспечение ссылочной целостности;

— независимость разработчиков в изменении отдельных компонентов распределенной среды.

Как правило, первые две из указанных проблем решаются обеспечением репликации данных с помощью стандартных средств СУБД. Но при изменении одного из компонентов гетерогенной РБД необходимо обновление сценариев репликации для всех компонентов-потребителей данных из измененного поставщика. До тех пор пока все сценарии не будут обновлены, не будет обеспечена правильная работа всей гетерогенной РБД.

РБД ИНГРИС

В данной статье рассмотрим информационную систему ИНГРИС, разрабатываемую для поддержки образовательного процесса в Тюменском государственном университете. ИНГРИС (аббревиатура от «интегрированная геораспределенная информационная система») основана на распределенной базе данных со следующими гетерогенными составляющими:

- БД «ПреАбитуриент» — информация об абитуриентах, зарегистрировавшихся самостоятельно через Интернет.
- БД «Абитуриент» — информация об абитуриентах, зарегистрированных в приемной комиссии, а также импортированная из базы данных «ПреАбитуриент».

• БД «Реестр студентов» — информация о контингенте студентов, обучающемся или обучавшемся в Тюменском государственном университете и его филиалах, Институте дистанционного образования, Институте дополнительного профессионального образования, а также информация о приказах по контингенту студентов.

• БД «Выпускник» — информация о выпускниках ТюмГУ.

• БД «УМК» — данные о специальностях, учебных планах и учебно-методических пособиях.

• БД «Нагрузка кафедр» — информация о распределении нагрузки на преподавателей кафедр.

• БД «ГИС: Кадастр» — информация о географической привязке корпусов ТюмГУ.

• БД «Система тестирования ТюмГУ» — информация о тестах и результатах тестирования студентов.

• БД «Биллинг» — система учета квот и оплаты использования Интернета.

• Системная база данных ИНГРИС — информация о пользователях ИНГРИС, их правах и подразделениях ТюмГУ.

Перечисленные базы данных разрабатываются, администрируются и наполняются различными подразделениями Тюменского государственного университета и НИИ Интеллектуальных информационных систем ТюмГУ.

Потоки данных

Между отдельными компонентами распределенной среды ИНГРИС существует взаимодействие. Подсистемы, накапливающие данные одной сферы, используются другими подсистемами как источники информации, исключающие дополнительную работу по ручному вводу данных.

К рассмотренному выше списку программ следует добавить информационные системы «1С: Предприятие», «Парус», «Дело» — коммерческие системы, разрабатываемые сторонними предприятиями-разработчиками и используемые для поддержки образовательного процесса в ТюмГУ. Данные программы также планируется интегрировать в общую циркуляцию данных.

В настоящий момент данные передаются по следующим потокам (рис. 1):

1. Данные об абитуриентах вводятся в подсистему «ПреАбитуриент» абитуриентами самостоятельно с помощью web-сайта.

2. Далее информация передается (копируется) в систему «Абитуриент», где проходит проверку работниками приемной комиссии. По результатам проверки приемной комиссии создаются приказы о зачислении в университет отдельных абитуриентов.

3. Приказы о зачислении и данные об абитуриентах, указанных в этих приказах, передаются (копируются) в подсистему «Реестр студентов».

4. Данные о студентах копируются из «Реестра студентов» в подсистему «Биллинг» для предоставления студентам доступа в Интернет.

5. Деканаты получают доступ к данным о студентах из БД «Реестр студентов» через интерфейс «Деканат».

6. Информация о структуре университета (формах и видах обучения, факультетах, специальностях) передается в подсистему «УМК».

7. Данные о выпускниках университета передаются из «Реестра студентов» в подсистему «Выпускник».

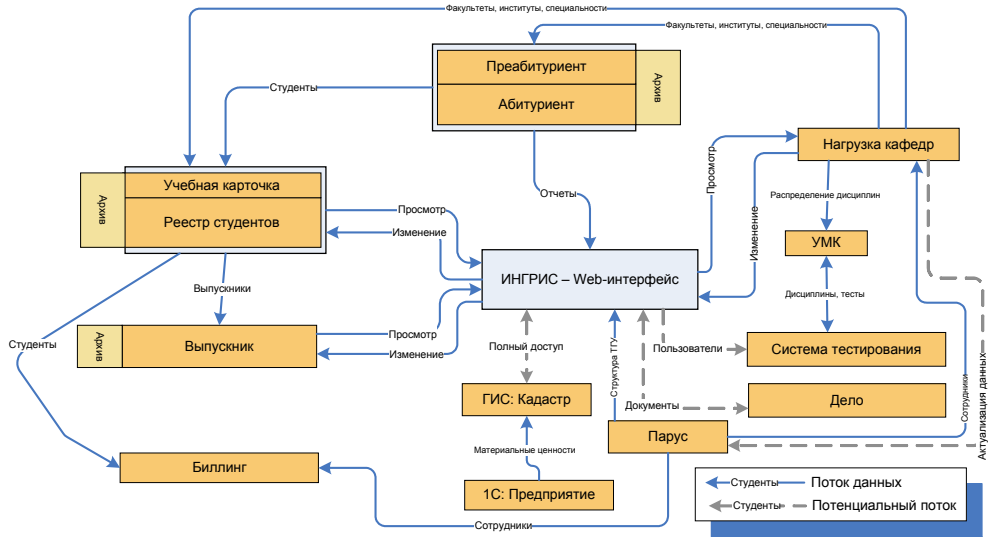

Рис. 1. Потоки данных в информационной системе ИНГРИС

На рис. 1 видно, что, если, например, модифицировать структуру базы данных «Реестр студентов», возможно, перестанут получать информацию подсистемы «Выпускник», «УМК», «Биллинг» и «ИНГРИС — Web-интерфейс». То есть работа этих систем возобновится только тогда, когда и они будут модернизированы в соответствии с изменением структуры поставщика данных.

На практике для обеспечения бесперебойной работы распределенной базы данных при необходимости изменения одним из разработчиков отдельного компонента гетерогенной РБД требуется обязательное согласование всех изменений со всеми разработчиками, сопровождающими компоненты РБД, зависящие от данного. Необходимо также согласование сроков изменения всех связанных компонентов и практически одновременный запуск полученных компонентов. Таким образом, время, необходимое для модернизации отдельного компонента, может увеличиться в десятки раз, в зависимости от количества подсистем-потребителей данных.

Существующие решения

Существует несколько готовых решений проблемы независимой модернизации отдельных компонентов распределенных гетерогенных баз данных. Некоторые являются методологическими и подразумевают специальные принципы разработки гетерогенных распределенных СУБД, например использование представлений (views) для преобразования структур данных из физической, для данного компонента, в логическую форму, принятую в разработке. В этом случае, при модификации структуры, разработчику достаточно модифицировать представления.

Интересное техническое решение предложено разработчиками СУБД Progress. В ней каждое приложение работает со словарем базы данных (Data Dictionary), в котором описывается структура базы данных, даны характеристики отдельных таблиц, полей, индексы, определены триггеры, выполняемые при работе с данными. Все остальные компоненты среды разработки используют по умолчанию информацию, хранимую в словаре, и любые изменения,

сделанные в Data Dictionary, наследуются всеми компонентами приложения, в результате приложения не зависят ни от физического расположения данных в распределенных и гетерогенных сетях, ни даже от формата источника данных [8].

Однако подобные решения требуют переноса всех составляющих гетерогенной распределенной базы данных на единую (определенную) программную платформу и не подходят для интеграции уже существующих гетерогенных составляющих.

Предложенное решение

В информационной системе ИНГРИС для решения данной проблемы предложено использовать специальный сервер-коммутатор, хранящий информацию о структурах всех баз данных — поставщиков, участвующих в обмене данными, и преобразующий информацию в вид, необходимый подсистеме-потребителю.

При такой схеме работы подсистемы-потребители не взаимодействуют непосредственно с подсистемами-поставщиками, а получают всю необходимую им информацию в нужном виде от сервера-коммутатора (рис. 2).

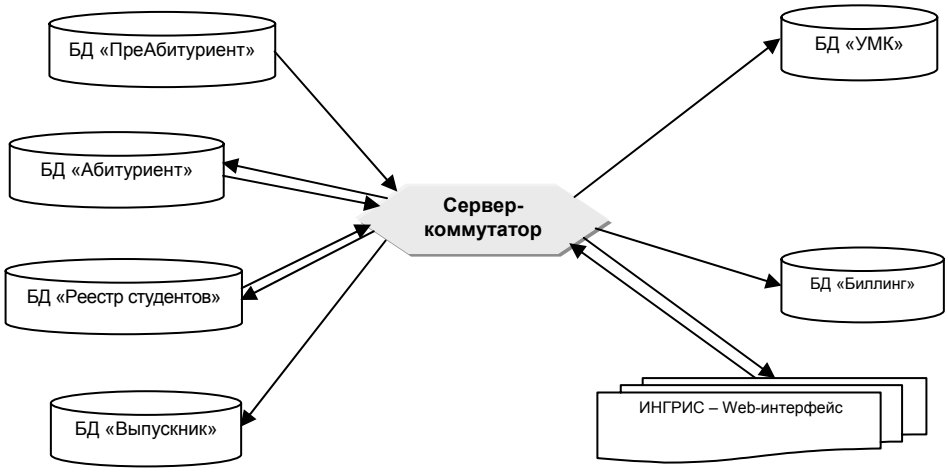

Рис. 2. Схема потоков данных
при использовании специального сервера-коммутатора

Сервер-коммутатор решает следующие задачи:

1. Хранение прав доступа отдельных подсистем к данным и контроль доступа.
2. Хранение прав доступа разработчиков подсистем для администрирования структур баз данных и контроль доступа.
3. Хранение и изменение метаданных о структурах баз данных — поставщиков и метаданных преобразования структуры.
4. Хранение и изменение информации об используемых драйверах и способах подключения к базам данных — поставщикам.
5. Обработка запросов от подсистем-потребителей и возврат результатов запросов (по протоколу SOAP).
6. Автоматическое присвоение глобальных уникальных идентификаторов (GUID) записям таблиц баз данных — поставщиков, выполнение запросов с использованием присвоенных глобальных уникальных идентификаторов.

Архитектурно сервер-коммутатор подразделяется на следующие составляющие (рис. 3):

1. Анализатор запросов — разбивает блок запросов на отдельные запросы, производит их синтаксический разбор и запускает преобразователь запросов.

2. Преобразователь запросов — обращается к хранилищу метаданных структуры, строит запрос к базе данных — поставщику, посылает запрос в пул соединений, преобразует результаты запроса в вид, необходимый потребителю, возвращает их потребителю.

3. Хранилище метаданных структуры — база данных, хранящая метаданные о структурах баз данных — поставщиков.

4. Пул соединений — управляет соединениями со всеми зарегистрированными базами данных — поставщиками с помощью соответствующего компонента доступа к базам данных, выполняет построенный запрос, возвращает данные преобразователю запросов.

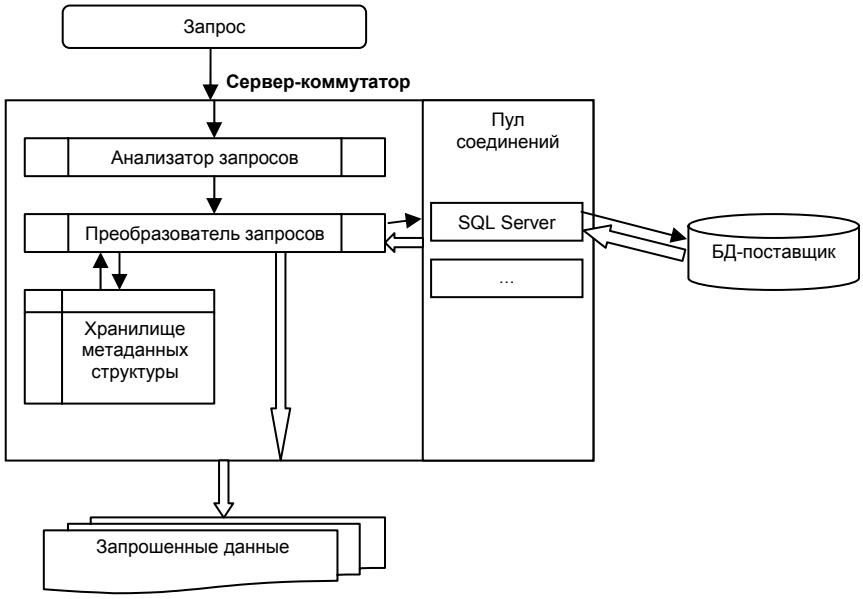

Рис. 3. Архитектура сервера-коммутатора

За представление структуры данных в виде метаданных сервера-коммутатора отвечают сами разработчики поставщиков данных. При изменении структуры базы данных — поставщика разработчик приводит в соответствие новой структуре метаданные на сервере-коммутаторе, и подсистемы-потребители продолжают получать данные в том же виде, что и раньше. Таким образом, достигается гораздо меньшая зависимость разработчиков баз данных — поставщиков от разработчиков баз данных — потребителей.

Для абстрагирования от структур баз данных поставщиков используется подход выделения ключевых сущностей, при котором, в отличие от реляционной структуры данных поставщика, присутствует только небольшое число основных (главных) сущностей (например, студенты, приказы) с большим набором атрибутов, полученных из справочников. При этом атрибут присоединяется к сущности независимо от того, через сколько отношений «таблица — внешний ключ — таблица» необходимо пройти, чтобы достигнуть данного атрибута.

Рассмотрим простой пример. Пусть в реляционной базе данных существуют таблицы студентов, групп, специальностей, факультетов, связанных последовательно отношениями «один ко многим» (рис. 4).

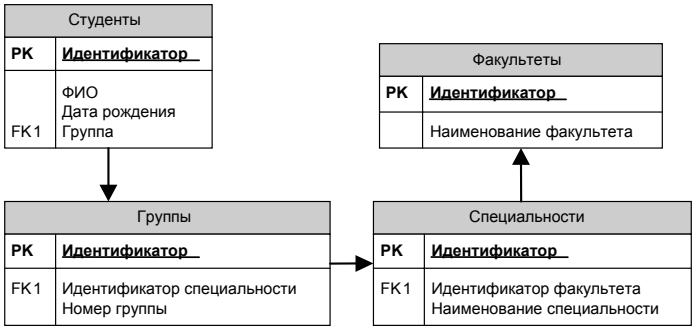

Рис. 4. Пример: схема реляционной базы данных по студентам

Для того чтобы компоненту-потребителю запросить факультет определенного студента, при обычном подходе необходимо соединить последовательно соответствующие записи всех четырех таблиц с помощью операции соединения (join). Если же структура базы данных изменилась и добавилась еще одна промежуточная таблица, разработчику компонента-потребителя необходимо исправить запрос, включив в него соединение с дополнительной таблицей.

При использовании сервера-коммутатора, для выяснения факультета определенного студента достаточно запросить значение соответствующего атрибута сущности «студенты» (рис. 5).

Рис. 5. Пример: сущность «студенты», представляемая сервером-коммутатором

Для подсистемы-потребителя данных структура сущности «студенты» будет оставаться неизменной даже после добавления промежуточных справочников, изменения структуры хранения данных в БД-поставщике.

Заключение

Таким образом, рассмотрено объединение в РБД с гетерогенными составляющими разрозненных информационных систем на примере информационной системы ИНГРИС ТюмГУ.

Предлагается следующий план научного исследования:

1. Разработка функционала сервера-коммутатора.
2. Разработка спецификаций обмена данными между компонентами РБД и сервером-коммутатором.
3. Реализация протокола обмена данными.
4. Апробация полученной системы.

ЛИТЕРАТУРА

1. *Вагин В. И.* Дедукция и обобщение в системах принятия решений. — М.: Наука, 1988. — 384 с.

2. *Гаврилова Т. А., Хорошевский В. Ф.* Базы знаний интеллектуальных систем. —. СПб.: Питер, 2000. — 384 с.

3. *Гаврилова Т. А., Червинская К. Р.* Извлечение и структурирование знаний для экспертных систем. — М.: Радио и связь, 1992. — 200 с.

4. *Коровин С. Я., Артамонов Р. А., Назаров В. Ю.* Информационная нефтепромысловая система нефтегазовой компании // Нефтяное хозяйство. — 2002. — № 8. — С. 113–118.

5. *Кульба В. В., Ковалевский С. С., Косяченко С. А., Сиротюк В. О.* Теоретические основы проектирования оптимальных структур распределенных баз данных. — М.: СИНТЕГ, 1999. — 660 с.

6. *Перегудов Ф. И., Тарасенко Ф. П.* Основы системного анализа. — Томск: НТЛ, 1997. — 396 с.

7. *Ревунков Г. И., Самохвалов Э. Н., Чистов В. В.* Базы и банки данных и знаний. — М.: Высш. шк., 1992. — 367 с.

8. *Твердова О.* СУБД Progress // DBMS. — 1997. — № 2. — Режим доступа: http://dhm.mstu.edu.ru/e_library/internet/in_inter/www-sbras.nsc.ru/win/docs/db/sql/18.htm, свободный.

9. *Шапот М.* Интеллектуальный анализ данных в системах поддержки принятия решений // Открытые системы. — 1998. — № 1. — С. 30–35.

10. *Шокин Ю. И., Федотов А. М., Жижимов О. Л., Мазов Н. А.* Интегрированная распределенная система (ИРИС) Сибирского отделения РАН // Выездное заседание научно-координационного совета по целевой программе «Информационно-телекоммуникационные ресурсы СО РАН», Иркутск, 29–30 авг. — Иркутск: ИДСТУ СО РАН, 2003.

R. M. Ganopolsky, D. B. Kepeschuk

KNOWLEDGE REPRESENTATION IN HETEROGENEOUS DISTRIBUTED DATA BASE USING INGDIS (“INTEGRATED GEODISTRIBUTED INFORMATIONAL SYSTEM”)

TYUMEN STATE UNIVERSITY

The problems in distributed data base with heterogeneous components are considered. Solutions based on special server-commutator are suggested. Problem statement and scientific research schedule are enunciated.