

УДК 004.62

А.А. Гаганов (асп. каф. ПМ), В.Е. Клавдиев, к.т.н., доц.

НОВЫЕ ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА ПРИ СОЗДАНИИ СУБД

Широкое применение в самых различных областях деятельности сегодня находят информационные системы с многопользовательским доступом, реализованные на базе СУБД с архитектурой “клиент-сервер”.

Основными требованиями к развитым СУБД являются:

- обеспечение целостности данных. Задача СУБД обеспечить поддержание логической целостности базы данных, задаваемой на пользовательском уровне в процессе проектирования, т.е. при изменении данных БД должна переходить от одного целостного состояния к другому;

- надежность хранения баз данных;

- обеспечение изолированности пользователей;

- оперативное отслеживание изменений БД. Необходимо, чтобы возникновение некоторой ситуации в базе данных четко и оперативно влияло на ход выполнения прикладной программы, т.е. имело место оповещение прикладных программ обо всех происходящих в базе данных изменениях.

Если первым трем требованиям удовлетворяют многие СУБД то задача оперативного отслеживания на данный момент не имеет хорошего решения. Традиционное решение задач контроля за состоянием базы данных и уведомления прикладных программ обо всех происходящих в ней событиях опирается на механизмы опроса (polling) прикладными программами базы данных, которому присущи следующие недостатки: прикладная программа не может непрерывно опрашивать базу данных, так как это приведет к перегрузке сервера бесполезными запросами, опрос производится через интервалы времени, которые определяет программист. Следовательно, если в базе данных происходят какие-либо изменения, то они обнаруживаются не сразу, а через некоторый промежуток времени. Именно поэтому традиционное решение не обеспечивает оперативного оповещения, между тем как в прикладных программах, работающих в реальном времени, это требование является ключевым. Постоянный опрос базы данных сильно сказывается на производительности системы — программы, опрашивающие базу данных, перегружают своими запросами сервер и сеть. Громоздкие конструкции в тексте программ, реализующие опрос, серьезно затрудняют ее написание и понимание.

Более современным способом решения проблемы оперативного отслеживания изменений БД является механизм событий, который, к сожалению, воспроизведен далеко не во всех СУБД.

Механизм событий в базе данных (database events) позволяет прикладным программам и серверу базы данных уведомлять другие программы о наступлении в базе данных определенного события и тем самым синхронизировать их работу. Операторы языка SQL, обеспечивающие уведомление, часто называют сигнализаторами событий в базе данных (database event alerters). Функции управления событиями целиком ложатся на сервер базы данных.

СУБД InterBase - это сервер реляционных баз данных широкого назначения, поставляемый фирмой Borland для компьютеров Sun, Hewlett-Packard/9000, VAX, IBM RS/6000, под SCO Unix и других. СУБД InterBase предоставляет оптимальную производительность при обработке данных в сетевых установках с разнородными платформами, обеспечивает создание сложных приложений и удовлетворяет большинству требований, предъявляемых к процессам обработки данных в динамических и распределенных вычислительных средах. СУБД InterBase гарантирует целостность и

непротиворечивость своих баз данных, обеспечивая защиту данных такими средствами, как домены, критерии проверки достоверности, целостность на уровне ссылок и триггеры, а также восстановлением данных и запросов после аппаратурных сбоев.

InterBase основана на концепции "активного ядра", в соответствии с которой многие функции управления данными и событиями, а также обеспечение контроля целостности данных конкретных задач, возлагаются на ядро СУБД, а не на прикладные программы.

В InterBase реализован свой механизм событий в котором событие является сообщением, посылаемым из триггера или хранимой процедуры БД, "менеджеру событий" InterBase-а, для последующего оповещения клиентских приложений о наступлении специфичного события или действия. Обычно это изменения в БД, такие как вставка, изменение, удаление записи БД.

Программный модуль "Менеджер событий" составляющий ядро механизма событий, содержит сформированный в очередь список посланных событий из триггеров и хранимых процедур событий. Каждый раз, когда новое сообщение послано, "Менеджер событий" оповещает соответствующим сообщением клиентские приложения, причем только те из них, которые зарегистрировались в менеджере событий как заинтересованные в конкретных изменениях, происходящих в базе данных. После получения события приложения определяют какое из событий произошло, если таких событий несколько. Действия, которые должны быть выполнены приложением, получившим сообщения об интересующем его событии, определяются логикой работы этого приложения.

Концепция обмена сообщениями в СУБД Interbase, появилась совсем недавно. Она позволяет более эффективно организовать работу с СУБД, обеспечивая поддержание целостности программы. Возможно из-за того, что это совсем новая технология, у нее есть несколько существенных недостатков:

1. Отсутствие информационных полей. Единственным информационным полем служит название события, что очень сильно ограничивает возможность передачи полезной информации приложению;
2. В клиентском приложении невозможно зарегистрировать более 15 событий;
3. Имя события не может быть больше 15 символов;
4. В силу (2), (3) становится невозможным передача какой-либо смысловой информации (индексы, имена из базы и т.д.) в имени события;
5. Программный компонент ответственный за прием событий, работает крайне неустойчиво.

Цель данной работы является проектирование и реализация альтернативной системы оповещения, которая бы была аналогична по функциям системе оповещения СУБД InterBase, но была бы избавлена от ее недостатков и имела возможность интеграции в информационную систему использующую в качестве среды разработки СУБД Interbase.

Основные требования к системе:

1. Передача сообщений с расширенной структурой.
2. Возможность фильтрации и группировки по данным в дополнительных информационных полях в сообщении.
3. Взаимосвязь системы оповещения с СУБД InterBase.
4. Возможность приема и обработки событий в среде Delphi.
5. Устойчивость в работе.