

ПРОБЛЕМЫ ХРАНЕНИЯ ИСТОРИИ ИЗМЕНЕНИЯ ДАННЫХ В КИС ОРГАНИЗАЦИИ

По мере развития организации разрастается и усложняется ее корпоративная информационная система. Под корпоративной информационной системой (КИС) здесь понимается единое хранилище информации и приложения, управляющие данными в соответствии с бизнес-логикой работы организации. Соответственно разрастание — это увеличение объема хранимых данных, а усложнение — увеличение разнообразия хранимой информации, взаимосвязей данных и усложнение приложений управления данными.

Вышесказанное неизбежно приводит к необходимости отслеживания изменений данных, с тем чтобы иметь возможность узнать, кто из сотрудников произвел конкретные изменения, когда эти изменения были сделаны и не были ли они ошибочными. Требования к хранению истории действий над объектами управления заложены в стандарты управления качеством ISO, что является еще одним подтверждением важности этой задачи.

Самый простой путь решения поставленной задачи — использование технологий резервного копирования. Резервное копирование заключается в сохранении срезов данных за определенные периоды времени, т.е. через регламентированные промежутки делаются копии всех важных данных, к которым при необходимости можно вернуться. Существует несколько основных вариантов резервного копирования:

- полное — сохраняются все данные;
- инкрементальное или разностное — копируются данные, сохраненные с момента последнего полного или инкрементального копирования.

Этот способ, несмотря на свою простоту, порождает множество проблем:

- полное копирование требует значительных ресурсов и не может выполняться часто;
- инкрементальное или разностное копирование при меньших объемах сохраняемых данных требует наличия у данных атрибута, позволяющего определить факт изменения и реализации достаточно быстро по этому атрибуту.

Такой способ хранения истории является единственно возможным для данных, хранимых в виде

файлов. Он также может применяться в некоторых базах данных (БД), имеющих атрибут изменения, например Lotus Notes / Domino, что и используется средствами архивирования для этой системы.

Другой способ заключается в ведении журналов изменений данных. Все современные СУБД и многие операционные системы используют журналирование изменений для обеспечения восстановления после сбоев. Суть этого метода в том, что наряду с самими данными пишется журнал, в котором указывается, какие изменения данных произошли. Такой способ позволяет восстановить картину данных на любой момент времени. Естественно, что ни СУБД, ни ОС не предоставляют разработчику доступ к своим собственным журналам изменений, поскольку журналы являются критичной составляющей надежности системы. Таким образом, реализация этого способа возможна только в системах, где разработчик может создавать свои обработчики события изменения данных. Любые серверы реляционных СУБД, соответствующие стандартам SQL, предоставляют пользователю такую возможность в виде триггеров на изменение данных. Существует набор программных пакетов ведения журналов изменений для собственных целей, например для осуществления репликации между серверами.

Одним из немаловажных преимуществ указанного способа является возможность расширения функций, возложенных на журнал изменений, поскольку журналы хранятся в самой БД в виде таблиц, что позволяет добавлять к ним свою собственную информацию, например автора изменений, время и различную дополнительную информацию. Таким образом, он является предпочтительным для хранения истории действий над данными КИС.

Кроме непосредственно хранения истории изменений, журналы можно использовать для следующих целей:

- репликации изменений между серверами в распределенной БД. Это позволяет распределять нагрузку между несколькими серверами: пользователи работают с одним сервером, статистический анализ, требующий больших ресурсов, ведется на другом, данные для отчетов и публикаций на Web берутся

с третьего. Кроме того, наличие реплик корпоративной БД на нескольких серверах повышает отказоустойчивость КИС;

- резервного копирования данных. В промежутках между полным резервным копированием данных производится резервное копирование журнала, которое требует гораздо меньшего времени и дискового пространства;

- синхронизации БД с различными внешними приложениями. Например, управления личными каталогами пользователей, информация о которых хранится в БД, синхронизация с другими БД. Если по соображениям производительности, безопасности или иным причинам нельзя создать триггер, производящий такую синхронизацию, то остается два пути — либо сканирование всех данных в БД и сравнение с данными внешнего приложения, что не всегда при-

емлемо с точки зрения производительности, либо анализ журнала изменений.

Использование журнала изменений для репликации и резервного копирования накладывает некоторые требования на процедуры каскадных изменений данных в целевой БД (БД, в которую идет репликация или происходит восстановление резервной копии). При журналировании как основного, так и каскадных изменений отработка журнала неизбежно приводит к повторной модификации одной и той же информации, что в свою очередь может привести к отказу в доступе из-за нарушения целостности данных, например, при каскадном добавлении новых записей в таблицу. Разработчик должен предусмотреть возможность отключения функций каскадирования при отработке журнала или предусмотреть отслеживание коллизий.