

Дрождин В.В., Масленников А.А., Спиринов А.В. Темпоральность в эволюционных базах данных. // Проблемы информатики в образовании, управлении, экономике и технике: Сб. статей VIII Всерос. научно-техн. конф. – Пенза: ПДЗ, 2008. – С. 27-30.

ТЕМПОРАЛЬНОСТЬ В ЭВОЛЮЦИОННЫХ БАЗАХ ДАННЫХ

В.В. Дрождин, А.А. Масленников, А.В. Спиринов

Пензенский государственный педагогический университет
им. В.Г. Белинского,
г. Пенза

Во многих программных приложениях, особенно в различных автоматизированных информационных системах, существует объективная необходимость сохранения исторической информации, а также получения к ней доступа. Поддержка исторической информации в базе данных как в хранилище программного приложения может осуществляться посредством реализации темпоральности в отношении изменения значений (состояний) элементов данных (операции добавления, модификации и удаления элементов данных), а также в отношении изменения структуры данных (операции преобразования схемы базы данных).

Поддержка темпоральности заключается в реализации возможности получения состояния любого объекта базы данных на любой момент времени его существования. То есть если объект базы данных был создан в момент времени t_1 и уничтожен в момент времени t_2 , то должна существовать возможность получения состояния этого объекта на каждый момент времени $t \in [t_1, t_2]$.

Эволюционная база данных представляет собой базу данных, способную оперативно изменяться (эволюционировать) в течение всего времени своего существования. Технически это реализуется, в частности, посредством поддержки эволюции схемы базы данных. Такая поддержка заключается в возможности создания новых и изменения существующих отношений (схем отношений) после определения схемы базы данных.

Поддержка темпоральности в отношении элементов данных в эволюционной базе данных может осуществляться централизованно, путем регистрации сведений обо всех изменениях, связанных с элементами данных (добавление, удаление, модификация), в специальном реестре (темпоральной таблице). Например, при создании объекта (элемента данных, кортежа) в темпоральную таблицу могут заноситься сведения о времени его создания (с соответствующей ссылкой на объект), при изменении объекта помимо времени изменения может сохраняться значение предыдущего состояния этого объекта, при удалении объекта может фиксироваться время удаления и значение состояния объекта на момент удаления. Описанный способ поддержки темпоральности аналогичен журнализации изменений базы данных, реализованной практически во всех современных системах управления базами данных (СУБД).

Альтернативой централизованной поддержке темпоральности в отношении элементов данных является децентрализованная поддержка темпоральности. Примером реализации такого способа решения является СУБД Postgres. В СУБД Postgres при модификации объекта (кортежа) непосредственное изменение исходного объекта не производится. Вместо этого создается новая запись,

содержащая измененные поля, а также сведения о транзакции, инициировавшей изменение (включая время завершения транзакции, т.е. собственно время изменения). Созданная таким образом запись ставится в соответствие исходному объекту посредством добавления ее в конец списка записей о предыдущих изменениях этого объекта [2].

Поддержка темпоральности в отношении схемы базы данных представляет собой серьезную проблему, так как затрагивает вопросы внешнего представления хранилища программного приложения. Поскольку эволюционная база данных допускает оперативное изменение своей схемы (схемы базы данных), то пользователи (клиенты) такой базы данных должны согласовывать формулировку своих запросов с состоянием схемы базы данных на требуемый момент времени. Несогласованность запроса пользователя со схемой базы данных может вызвать, например, попытку обращения к несуществующей в момент запроса (или на момент, указанный в запросе) сущности (таблице) или к несуществующему в указанный момент времени атрибуту (полю).

Для упрощения согласования запросов пользователя со схемой базы данных можно предложить три конвенции внешнего представления структуры хранилища.

1. Согласование запроса пользователя с состоянием структуры хранилища в начальный момент времени. В этом случае происходит фиксация внешнего представления структуры хранилища на момент создания схемы базы данных, т.е. состояние схемы базы данных в любой момент существования отображается через представление ее состояния в начальный момент времени (в момент ее создания). Такой способ согласования скрывает от пользователя факт эволюции схемы базы данных, однако сами эволюционные возможности при этом ограничены зафиксированным внешним представлением структуры хранилища.

2. Согласование запроса пользователя с состоянием структуры хранилища в текущий момент времени. В этом случае любая запрашиваемая историческая информация отображается через представление текущего состояния схемы базы данных. Такой способ согласования не накладывает принципиальных ограничений на эволюционные возможности схемы базы данных, однако приводит к необходимости отслеживания со стороны пользователя изменений в текущем состоянии внешнего представления структуры хранилища. Кроме того, доступ к исторической информации на моменты времени, когда представление схемы базы данных отличалось от ее текущего представления, будет осуществляться с неизбежными искажениями.

3. Согласование запроса пользователя с состоянием структуры хранилища на момент времени, указанный в запросе. В этом случае любая запрашиваемая историческая информация отображается в соответствии с представлением схемы базы данных на требуемый момент времени. Такой способ согласования позволяет адекватно представлять историческую информацию на любой момент времени ее существования, однако приводит к необходимости формулировки запроса пользователя в точном соответствии с особенностями представления схемы базы данных на каждый конкретный исторический момент.

Целесообразность использования эволюционной базы данных в программном приложении обусловлена, главным образом, необходимостью адаптации хранилища к изменениям во внешней среде, а также потребностью в повышении

эффективности хранения и обработки данных при стабильных внешних условиях [1]. Проведение тех или иных эволюционных преобразований базы данных может быть вызвано из внешней среды (например, соответствующими правилами бизнес-логики приложения) или инициировано механизмом самоорганизации самой базы данных (в том случае, если применяемая при проектировании хранилища модель данных предусматривает самоорганизацию).

Использование в качестве хранилища программного приложения эволюционной базы данных, поддерживающей темпоральность, позволит упростить автоматизацию решения задач, предусматривающих хранение и обработку исторической информации. Поддержка способности программной системы к эволюционному развитию будет способствовать уменьшению затрат на сопровождение системы и продлению срока ее эксплуатации.

Библиографический список

1. Дрождин, В.В. Системный подход к построению модели данных эволюционных баз данных // Программные продукты и системы. – 2007. – № 3. – С. 52 – 55.
2. Кузнецов, С.Д. Основы современных баз данных // Информационно-аналитические материалы Центра информационных технологий. – <http://www.citforum.ru/database/osbd/contents.shtml>