МИНОБРНАУКИ РОССИИ

федеральное государственное бюджетное образовательное учреждение

высшего профессионального образования

«Череповецкий государственный университет»

**Лабораторная работа № 3**

**«Управление транзакциями. Журнализация изменений БД»**

**Выполнил:**

студент гр. 1ИВТпб-01-21оп

Климов А.Г.  
**Проверил:** преподаватель

Селяничев О.Л.  
Отметка о зачете:

Череповец

2017 год

**Цель работы:**

1. Рассмотреть порядок выполнения транзакций.

2. Изучить механизм ведения журнала изменений БД.

3. Исследовать способы поддержания целостности БД.

**Задания**

1. Исследовать работу с одной БД средствами СУБД Access. Изучить, каким образом осуществляется незаметная для пользователей работа для случаев:

- потерянных изменений;

- проблемы промежуточных данных;

- чтение грязных данных;

- неповторяющихся чтений;

- кортежей-фантомов.

Для каждой ситуации дать описание проведенного эксперимента, результатов его и сделать выводы.

2. На примере эксплуатации конкретной БД рассмотреть поведение СУБД, направленное на поддержание целостности БД, для случаев:

- отката транзакции;

- мягкого сбоя;

- жесткого сбоя (без его реализации).

**Выполнение заданий**

1. *Отсутствие потерянных изменений.*

Ситуация. Первая транзакция (Т1) изменяет объект БД А. До завершения Т1 вторая транзакция (Т2) также изменяет объект А. Т2 завершается оператором ROLLBACK Рис. 1. “Конфликт записи” (напр., из-за нарушения ограничения целостности). Тогда при повторном чтении объекта А Т1 не видит изменений объекта, произведенных ранее Рис. 2. “Отсутствие потерянных изменений”.

Диагноз. Ситуация потерянных изменений; противоречие требованию изолированности пользователей.

Лечение. До завершения Т1 никакая другая транзакция не может изменять объект А.

Профилактика: Отсутствие потерянных изменений – минимальное требование к СУБД по части синхронизации параллельно выполняемых транзакций.

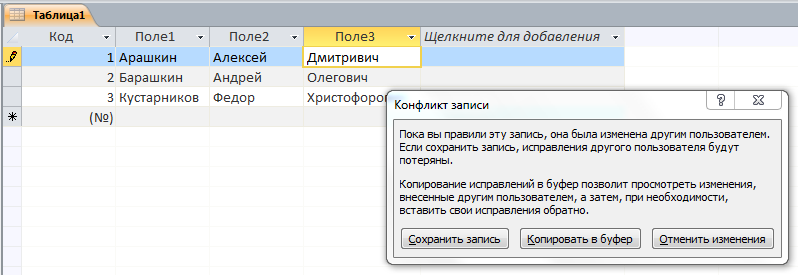


Рис. 1. Конфликт записи

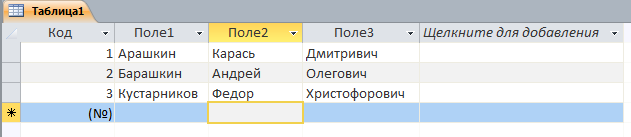


Рис. 2. Отсутствие потерянных изменений

2. *Отсутствие чтения «грязных данных».*

Ситуация. Т1 изменяет объект БД А Рис.3. Параллельно с этим Т2 читает объект А. Т.к. операция изменения еще не завершена, Т2 видит несогласованные «грязные» данные Рис. 4.

Диагноз. Несоответствие требованию изолированности пользователей: каждый пользователь начинает свою транзакцию при согласованном состоянии базы данных и вправе видеть согласованные данные.

Профилактика. До завершения Т1, изменившей объект А, никакая другая транзакция не должна читать объект А (минимальным требованием является блокировка чтения объекта А до завершения операции его изменения в Т1).

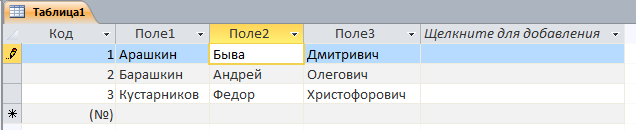


Рис. 3. Т1 изменяет объект БД А

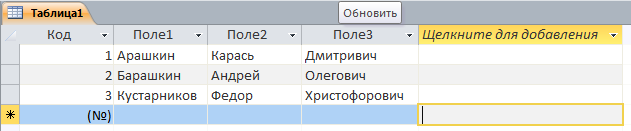


Рис. 4. Т2 видит несогласованные данные

3. *Отсутствие неповторяющихся чтений.*

Ситуация. Т1 читает объект БД А Рис. 5. До завершения Т1 Т2 изменяет объект А и успешно завершается оператором COMMIT Рис. 6. Т1 повторно читает объект А и видит его измененное состояние Рис. 7.

Диагноз. Неповторяющиеся чтения.

Профилактика. До завершения Т1 никакая другая транзакция не должна изменять объект А. В большинстве систем это является максимальным требованием к синхронизации транзакций.

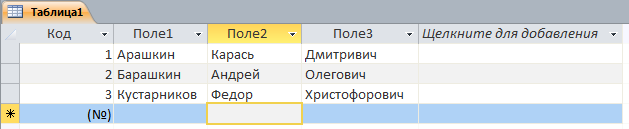


Рис. 5. Т1 читает объект БД А

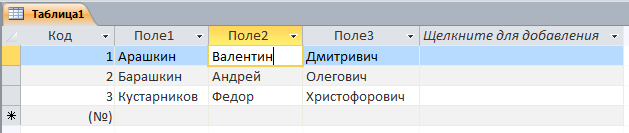


Рис. 6. Т2 изменяет объект А и успешно завершается оператором COMMIT

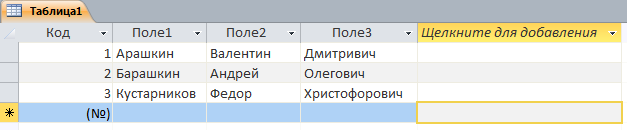


Рис. 7. Таблица базы данных с полями

4. *Проблема кортежей-«фантомов».*

Она вызывает ситуации, которые также противоречат изолированности пользователей.

Ситуация. Т1 выполняет оператор А выборки кортежей отношения R с условием выборки S Рис. 8. До завершения Т1 Т2 вставляет в отношение R новый кортеж К, удовлетворяющий условию S, и успешно завершается Рис. 9. Т1 повторно выполняет оператор А, и в результате появляется кортеж, который отсутствовал при первом выполнении оператора Рис. 10.

Диагноз. Противоречие идее изолированности транзакций.

Профилактика. Требуется более высокий уровень синхронизации транзакций. Идеи такой синхронизации (предикатные синхронизационные захваты) известны давно, но в большинстве систем не реализованы.

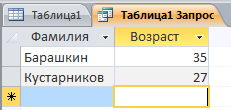


Рис. 8. Выборка по возрасту

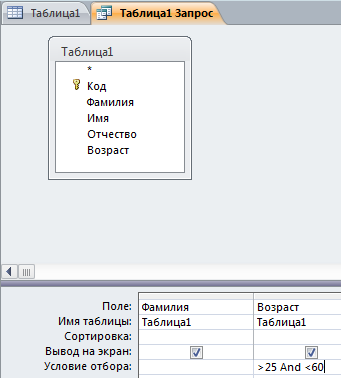


Рис. 9. Т2 добавляет новый кортеж

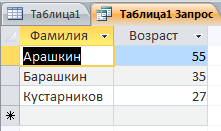


Рис. 10. Появляется новый кортеж

**Индивидуальный откат транзакции**. Тривиальной ситуацией отката является ее явное завершение оператором **ROLLBACK**. Возможны также ситуации, когда откат инициируется системой. Примерами могут быть возникновение исключительной ситуации в прикладной программе (например, деление на ноль) или выбор транзакции в качестве жертвы при обнаружении синхронизационного тупика. Для восстановления согласованного состояния БД при индивидуальном откате транзакции нужно устранить последствия операторов модификации БД, которые выполнялись в этой транзакции.

**Потеря содержимого оперативной памяти (мягкий сбой)**. Такая ситуация может возникнуть при аварийном выключении электрического питания, при возникновении неустранимого сбоя процессора (например, срабатывании контроля оперативной памяти). Ситуация характеризуется потерей той части БД, которая к моменту сбоя содержалась в буферах оперативной памяти.

**Поломка основного внешнего носителя БД (жесткий сбой)**. Эта ситуация при достаточно высокой надежности современных устройств внешней памяти может возникать сравнительно редко, но тем не менее, СУБД должна быть в состоянии восстановить БД даже и в этом случае. Основой восстановления является архивная копия и журнал изменений БД.

Во всех трех случаях основой восстановления является избыточное хранение данных. Эти избыточные данные хранятся в журнале, содержащем последовательность записей об изменении БД.

**Индивидуальный откат транзакции**. Для выполнения индивидуального отката транзакции по общему журналу, все записи в журнале от данной транзакции связываются в обратный список.

Для незакончившихся транзакций: началом списка является запись о последнем изменении БД, произведенном данной транзакцией. Для закончившихся транзакций начало списка - запись о конце транзакции, которая обязательно вытолкнута во внешнюю память журнала. Концом списка всегда служит первая запись об изменении БД, произведенном данной транзакцией.

**Восстановление после мягкого сбоя.** К числу основных проблем восстановления после мягкого сбоя относится то, что одна логическая операция изменения БД может изменять несколько физических блоков БД, например, страницу данных и несколько страниц индексов. Страницы БД буферизуются в оперативной памяти и выталкиваются независимо. Несмотря на применение протокола WAL, после мягкого сбоя набор страниц внешней памяти БД может оказаться несогласованным, т.е. часть страниц внешней памяти соответствует объекту до изменения, часть — после изменения. К такому состоянию объекта не применимы операции логического уровня.

Состояние внешней памяти БД называется физически согласованным, если страницы всех объектов соответствуют состоянию объекта либо после его изменения, либо до изменения. В журнале отмечаются точки физической согласованности БД — моменты времени, в которые во внешней памяти содержатся согласованные результаты операций, завершившихся до соответствующего момента времени, и отсутствуют результаты операций, которые не завершились, а буфер журнала вытолкнут во внешнюю память. Такие точки называют tpc (time of physical consistency - точки физической согласованности).

**Восстановление после жесткого сбоя**. Для восстановления согласованного состояния БД после жесткого сбоя журнала изменений БД недостаточно – нужны журнал и архивная копия БД. Восстановление начинается с обратного копирования БД из архивной копии. Затем для всех закончившихся транзакций выполняется redo, т.е. операции выполняются повторно.

Более точно, происходит следующее:

• по журналу в прямом направлении выполняются все операции;

• для транзакций, которые не закончились к моменту сбоя, выполняется откат.

Так как жесткий сбой не сопровождается утратой буферов оперативной памяти, можно восстановить БД до такого уровня, чтобы можно было продолжить даже выполнение незакончившихся транзакций. Но обычно это не делается, потому что восстановление после жесткого сбоя — это достаточно длительный процесс.

**КОНТРОЛЬНЫЕ ВОПРОСЫ**

1. *Опишите, какими средствами достигается изолированность пользователей в многопользовательских системах?*

Система должна обеспечивать изолированность пользователей, т.е. создание иллюзии того, что каждый работает с БД в одиночку. Тогда транзакция выступает единицей изолированности пользователей. Существуют несколько вариантов изолированности транзакций.

2. *Охарактеризуйте методы сериализации транзакций.*

Сериализация транзакций — это механизм их выполнения по некоторому сериальному плану. Способ выполнения набора транзакций называется сериальным планом, если результат совместного выполнения транзакций эквивалентен результату некоторого последовательного выполнения этих же транзакций. Только система с сериализацией транзакций обеспечивает изолированность пользователей.

Основная проблема реализации состоит в выборе метода сериализации набора транзакций, который не ограничивал бы их параллельность.

Практические методы сериализации транзакций основывается на учете трех конфликтов между транзакциями:

• W-W — Т2 пытается изменять объект, измененный не закончившейся Т1;

• R-W — Т2 пытается изменять объект, прочитанный не закончившейся Т1;

• W-R — Т2 пытается читать объект, измененный не закончившейся Т1.

Существуют два подхода к сериализации транзакций:

- метод синхронизационных захватов объектов БД;

- метод временных меток.

Суть обоих подходов состоит в обнаружении конфликтов транзакций и их устранении. Для каждого из подходов имеются две разновидности — пессимистическая и оптимистическая.

При применении пессимистических методов, ориентированных на ситуации, когда конфликты возникают часто, конфликты распознаются и разрешаются немедленно при их возникновении.

Оптимистические методы основываются на том, что результаты всех операций модификации БД сохраняются в рабочей памяти транзакций. Реальная модификация БД производится только на стадии фиксации транзакции. Тогда же проверяется, не возникают ли конфликты с другими транзакциями.

3. *Какие функции журнала изменений БД?*

Запись об изменении БД, которая должна поступить в журнал при выполнении любой операции модификации БД, не записывается немедленно во внешнюю память, так как это привело бы к существенному замедлению работы системы. Записи в журнал буферизуются: очередная страница выталкивается во внешнюю память журнала только при полном заполнении записями.

4. *Какими средствами поддерживается целостность БД?*

Корректное поддержание транзакций обеспечивает целостность баз данных и обеспечивают изолированность пользователей во многопользовательских системах.

5. *Опишите, как происходит взаимодействие с журналом при мягком и жестком сбоях.*

**Вывод:**

1. Рассмотрел порядок выполнения транзакций.

2. Изучил механизм ведения журнала изменений БД.

3. Исследовал способы поддержания целостности БД.