Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и технологий

Кафедра компьютерных систем и программных технологий

**Отчет по лабораторной работе №6**

по дисциплине «Базы данных»

«Изучение работы транзакций»

Работу выполнил

студент гр. 43501/1

В.В.Сякаев

Проверил преподаватель

А.В. Мяснов

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_2016г

Санкт - Петербург

2016

**1. Цель работы**

Познакомить студентов с механизмом транзакций, возможностями ручного управления транзакциями, уровнями изоляции транзакций.

**2.Программа работы**

1. Изучить основные принципы работы транзакций.

2. Провести эксперименты по запуску, подтверждению и откату транзакций.

3. Разобраться с уровнями изоляции транзакций в Firebird.

4. Спланировать и провести эксперименты, показывающие основные возможности транзакций с различным уровнем изоляции.

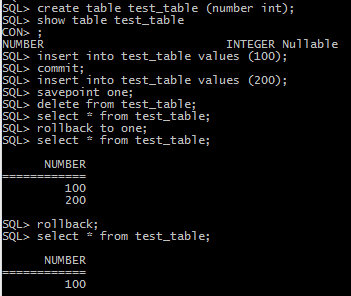
5. Продемонстрировать результаты преподавателю, ответить на контрольные вопросы.

Работа проводится в IBExpert. Для проведения экспериментов параллельно запускается несколько сессий связи с БД, в каждой сессии настраивается уровень изоляции транзакций. Выполняются конкурентные операции чтения/изменения данных в различных сессиях, а том числе приводящие к конфликтам.

**3. Ход работы**

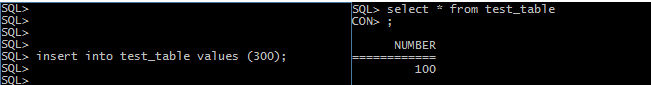
**3.1 Эксперименты по запуску, подтверждению и откату транзакций**

Создадим тестовую таблицу с одним полем, внесем в него запись и произведем commit. После чего внесем еще одну запись, но уже создадим точку сохранения. Удалили данные — при выборке таблица пуста. Вернулись к точке сохранения — при выборке на экран выведены прежние данные. После возвращения к последнему подтверждению транзакции в таблице осталась только первая запись.

Рис. 1

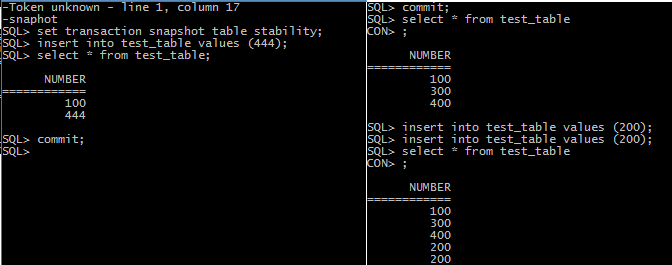
**3.2 Различные уровни изоляции**

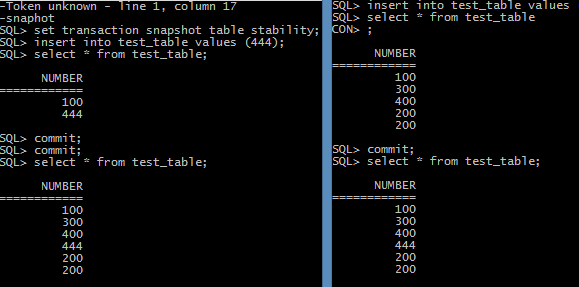
**Snapshot** – Уровень изоляции по умолчанию. Все операции в транзакции с данным уровнем изоляции видят только те данные, которые существовали (committed) на момент старта этой транзакции (создается «снимок» БД), даже если они впоследствии были изменены или удалены другими транзакциями.

Рис. 2

Как видно из рис. 2 , при подключении двух клиентов к базе изменения, произведенные одним из них (в данном случае вставка нового значения в тестовую таблицу) не видны другому клиенту. Второй клиент видит только те данные, которые существовали на момент его подключения к базе.

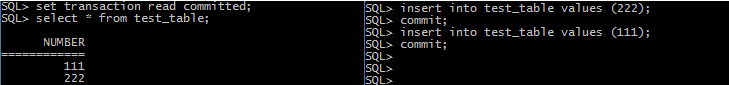
**Snapshot table stability -** Это уровень изоляции также создает "моментальный" снимок базы данных, но одновременно блокирует на запись данные, задействованные в операциях, выполняемые данной транзакцией. Это означает, что если транзакция с уровнем SNAPSHOT TABLE STABILITY изменила данные в какой-нибудь таблице, то после этого данные в этой таблице уже не могут быть изменены в других параллельных транзакциях. Если при запуске клиентом транзакции с таким уровнем изоляции какая-нибудь другая транзакция выполнила неподтвержденное изменение данных любой таблицы из базы данных, то запуск транзакции с уровнем изоляции snapshot table stability приведет к ошибкам (или база повиснет, что было в моем случае)

 Рис.3

Рис. 4

Видно, что произведенные изменения для обоих клиентов видны только после подтверждения обеих транзакций.

**Read commited-** Уровень подразумевает видимость только подтвержденных действий (транзакции, завершившие операцию COMMIT). Транзакция может видеть самые последние подтвержденные изменения базы данных, выполненные другими транзакциями.(неподтвержденные транзакции увидеть нельзя, система переходит в состояние ожидания подтверждения транзакции)

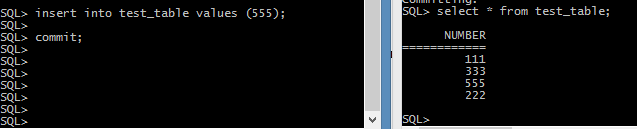
Рис.5

При этом уровне изоляции используются еще два взаимоисключающих параметра:

По умолчанию NO RECORD\_VERSION (isc\_tpb\_no\_rec\_version в TPB) требует, чтобы было выполнено подтверждение всех измененных другими транзакциями данных.

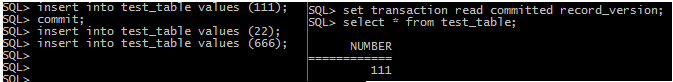
Рис.6

На рис 6. видно, что без подтверждения изменений в другой транзакции, из транзакции с уровнем изоляции read committed no record\_version wait нельзя просмотреть содержимое изменяемой таблицы .

Рис. 7

После подтверждения первым клиентом транзакции второй смог увидеть содержимое тестовой таблицы (завершить выполнение команды select).

RECORD\_VERSION (isc\_tpb\_rec\_version в TPB) позволяет читать самую последнюю подтвержденную версию изменений, даже если существуют другие неподтвержденные версии.

Рис. 8

На рис. 8 видно, что второй клиент увидел только подтвержденные изменения из транзакции первого клиента

**4. Выводы**

Транзакция – набор логически связанных операций, работающих с данными базы данных, и либо переводящий базу данных в из одного целостного состояния в другое, либо нет( «все» — или «ничего»). Они реализуют механизм обеспечения целостности данных в базе. Позволяют отслеживать конфликты операций с данными в таблицах и предотвращать смешивание данных в разных сессиях. Конфликты могут возникать по многим причинам: одновременное обновление данных в одной и той же таблице, удаление данных и одновременно попытка обращения к ним, занесение строк с одинаковым первичным ключом и так далее.

Транзакции гарантируют надежность, т.е. сохранение результатов транзакции вне зависимости от других факторов.

Наличие различных уровней изолированности позволяет параллельным транзакциям выполняться без оказания влияния на результат выполнения друг друга.

Вместе с тем механизм транзакций имеет свои недостатки:. Например, замедляет работу с базой данных при одновременной работе нескольких пользователей, так как транзакции должны выполняться последовательно и не могут быть прерваны.

Кроме того, возможно возникновение так называемых deadlock`ов – тупиков при исполнении транзакций. Они должны обнаруживаться заранее и предотвращаться или разрешаться при