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 выполнила

студентка гр. 43501/3

К.В. Ардашова

Проверил преподаватель

А.В. Мяснов

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

Санкт - Петербург

2016

**Цель работы**

Познакомиться с механизмом транзакций, возможностями ручного управления транзакциями, уровнями изоляции транзакций.

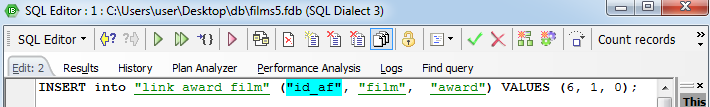
**Программа работы**

1. Провести эксперименты по запуску, подтверждению и откату транзакций.
2. Спланировать и провести эксперименты, показывающие основные возможности транзакций с различным уровнем изоляции

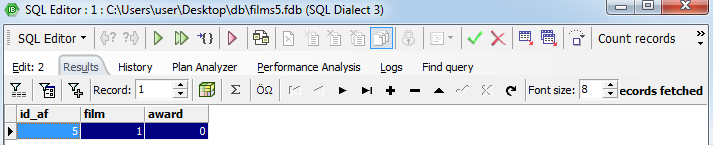
**Ход работы**

**1. Эксперименты по запуску, подтверждению и откату транзакций**

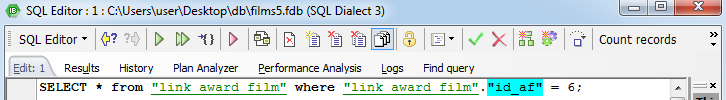
**Запуск транзакции:** добавим строку в таблицу.



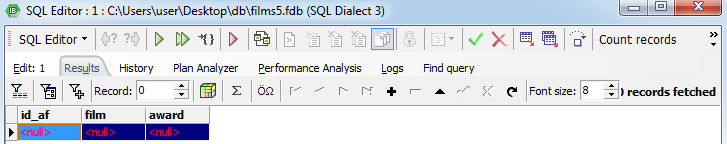
Видим добавленную строку:



В другом окне осуществим поиск строки:

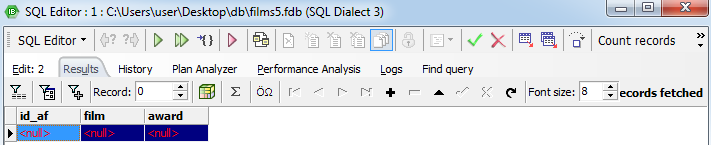


Строка не найдена:

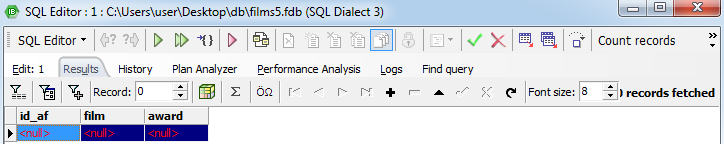


Совершим **откат транзакции** (rollback). Тогда из обоих окон поиск данных с id = 5 ничего не даст:

Окно 1:

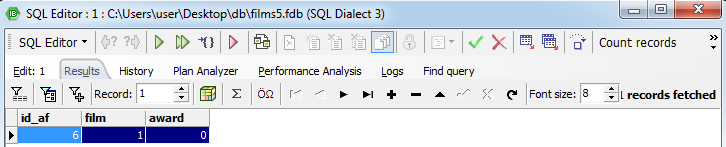


Окно 2:



Добавим такую же строку ещё раз в окне 1 и **подтвердим** транзакцию (commit).

Запросив строку с id = 12 в окне 2 увидим:



**2. Уровни изоляции транзакций в Firebird**

В Firebird есть три уровня изоляции транзакций:

1. READ COMMITTED
2. SNAPSHOT
3. SNAPSHOT TABLE STABILITY

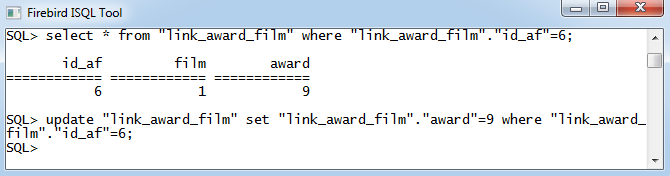
Данная часть работы выполняется в консоли для гарантии установки уровня изоляции с использованием нескольких терминалов.

**READ COMMITTED** – видны только результаты операций, для которых сделан commit.

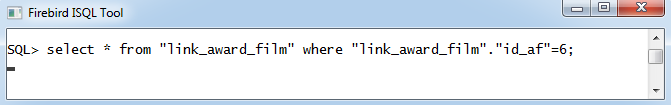
В терминале 1 установим нужный нам уровень:



В терминале 2 изменим строку:

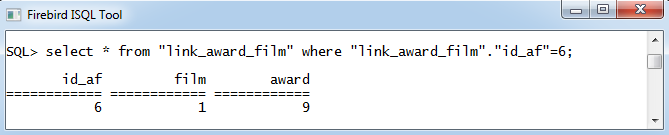


В терминале 1 запросим данную строку:



Видим, что терминал, как и ожидалось, перешел в режим ожидания.

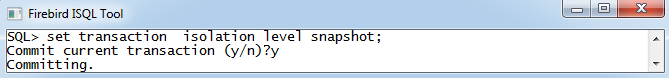
В терминале 2 завершим транзакцию (commit), после чего в терминале 1 сразу же появился результат:



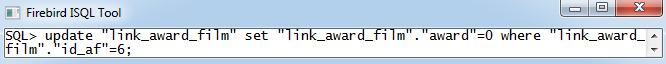
Видим, что при уровне изоляции read committed при запросе данных, изменяемых в ходе другой транзакции, начинается ожидание до того момента, как та транзакция будет завершена.

**SNAPSHOT** – создается «снимок» базы данных, и все операции чтения в рамках транзакции с данным уровнем изоляции будут видеть только состояние базы на момент начала операции.

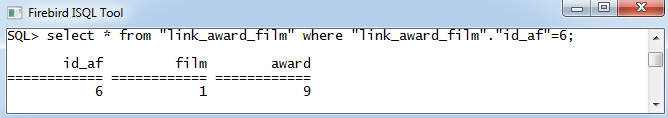
Терминал 1:



Терминал 2:

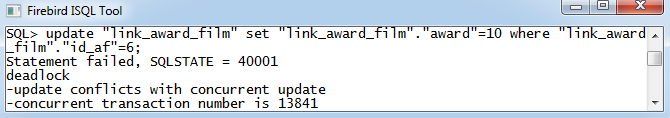


Терминал 1:



Осуществлена выборка «старых» значений.

Терминал 1:



Изменение записи вызвало ошибку.

Видим, что при уровне изоляции snapshot:

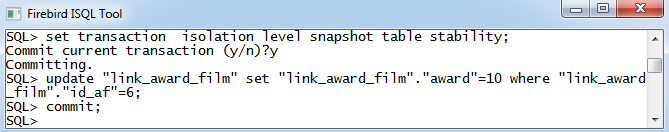
• при запросе данных, изменяемых в ходе другой транзакции получаем состояние запрашиваемых данных на момент начала выполнения текущей транзакции,

• при попытке обновления уже измененных в другой транзакции данных получаем сообщение об ошибке.

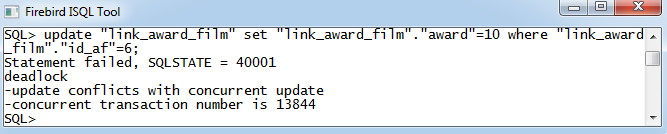
**SNAPSHOT TABLE STABILITY** – также создается «снимок» базы данных на момент начала транзакции. При этом блокируются на запись данные, задействованные в операции. Таблицы, изменяемые данной транзакцией, недоступны другим транзакциям, а изменяемые другими транзакциями – недоступны в рамках данной транзакции.

Попытка обновить одну и ту же запись с разных терминалов (ожидается конфликт).

Терминал 1:

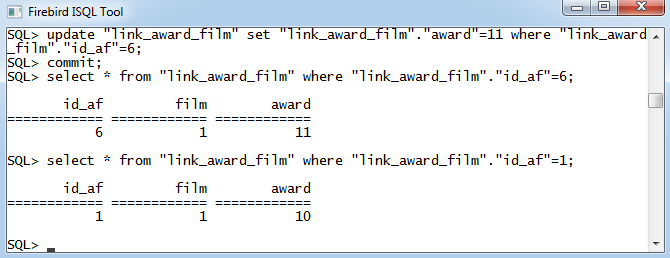


Терминал 2:

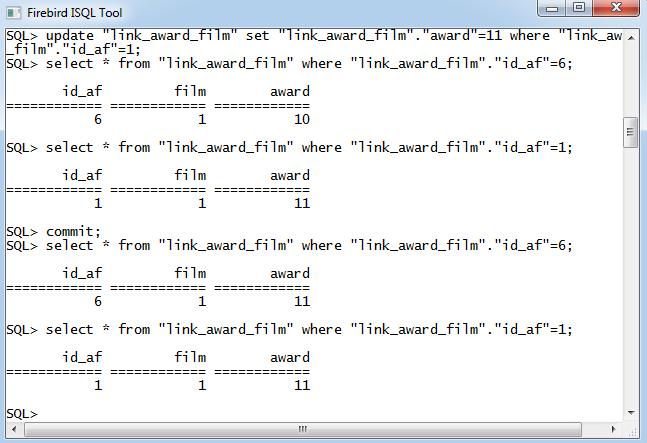


Попытка обновить разные записи с разных терминалов (не ожидается конфликт).

Терминал 1:



Терминал 2:



Видим, что результаты соответствуют ожиданиям. Изменяемая в транзакции с уровнем изоляции snapshot table stability таблица целиком блокируется для всех остальных транзакций до окончания выполнения транзакции.

**Выводы**

Транзакция – неделимая группа последовательных операций с базой данных. Механизм транзакций случит для обеспечения целостности данных в БД – позволяет предотвратить ошибки, возникающие при смешивании данных в разных сессиях, и отследить конфликты.

Важными свойствами транзакций являются:

* Атомарность, т.е. транзакция может быть либо выполнена целиком, либо не выполнена вовсе.
* Изолированность: параллельно выполняющиеся транзакции не оказывают влияния на результат выполнения друг друга.
* Надежность: в случае, если транзакция вообще выполнена, гарантировано сохранение результатов транзакции независимо от внешних факторов.
* Согласованность: транзакция переводит БД из одного согласованного состояния в другое согласованное состояние, обеспечивая более высокий уровень целостности данных.

Транзакции могут иметь несколько уровней изолированности. Чем выше уровень, тем меньше транзакций могут выполняться одновременно, но тем лучше обеспечивается целостность данных. Чем ниже этот уровень, тем ниже точность данных, но тем больше транзакций можно выполнять параллельно.

Недостаток транзакций – замедление работы с БД, к примеру, при одновременной работе нескольких пользователей с одними и теми же данными (за счет ожидания), а так же возможность возникновения тупиков (блокировок транзакций) (deadlock). Такие ситуации требуется либо выявлять перед запуском транзакций и предотвращать, либо разрешать с помощью СУБД при возникновении.