**Федеральное агентство по образованию**

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**

**ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**Институт информационных технологий и управления**

**Кафедра компьютерных систем и программных технологий**

**ОТЧЕТ**

**по лабораторной работе №7**

# «Изучение работы транзакций»

**Студент гр.43501/1: Русаков Р.В.**

**Преподаватель: Мяснов А.В.**

**Санкт-Петербург**

**2014**

1. **Цели работы**

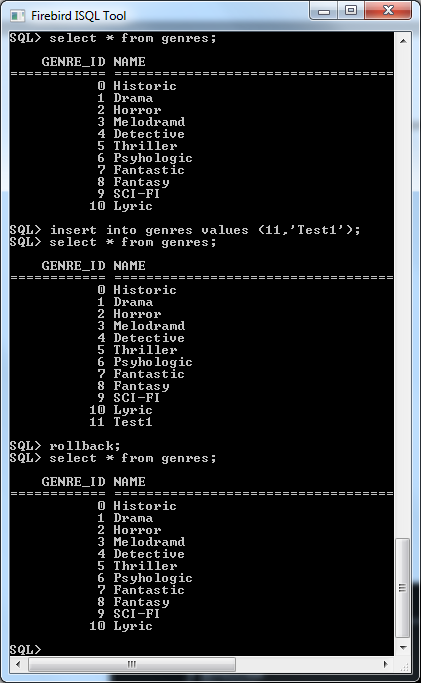
Познакомить студентов с механизмом транзакций, возможностями ручного управления транзакциями, уровнями изоляции транзакций.

1. **Основные принципы работы**

Все действия, выполняемые с базой данных — любые изменения как данных, так и метаданных, а также любая выборка данных — осуществляются в контексте какой-либо транзакции. Все изменения, выполненные в контексте транзакции, можно либо подтвердить (при отсутствии ошибок базы данных), либо все отменить. Если в любой операции, выполняемой в контексте транзакции, произошла ошибка, то подтвердить такую транзакцию нельзя. Можно только отменить все действия. При использовании операторов языка SQL для работы с базой данных для запуска транзакции и задания ее характеристик используется оператор SET TRANSACTION. Для подтверждения транзакции используется оператор SQL COMMIT, для отмены всех действий транзакции используется оператор ROLLBACK.

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

Запуск и откат транзакции:



Запуск транзакции и наблюдение изменений в другой сессии связи:

Для первой сессии связи выполним:

select \* from genres ;

insert into genres values (11,'Test1');

select \* from genres;



После этого для второй сессии выполним :

select \* from genres ;



Чтобы во второй сессии связи увидеть изменения, сделанные в первой, нужно подтвердить транзакции в обоих сессиях:

* 1. **Уровни изоляции:**

Таблица 2.2.1. Уровни изоляции транзакции в InterBase/Firebird

|  |  |
| --- | --- |
| **SQL** | **Значение** |
| READ COMMITTED | Чтение подтвержденных изменений. Транзакция может видеть самые последние подтвержденные изменения базы данных, выполненные другими транзакциями. |
| SNAPSHOT | Мгновенный снимок (образ). Значение по умолчанию. Другое название — повторяемое чтение (Repeatable Read). Дает состояние базы данных на момент старта транзакции. Изменения, выполненные другими транзакциями, в данной транзакции не видны. Естественно, транзакция «видит» все изменения, выполненные в контексте этой транзакции. |
| SNAPSHOT TABLE STABILITY | Изолированный образ или упорядочиваемый, сериализуемый (Serializable) образ. Аналогичен уровню SNAPSHOT с тем отличием, что другим транзакциям разрешено чтение данных из таблиц данной транзакции, однако они не могут вносить в них никаких изменений. |

Уровень изолированности транзакций — значение, определяющее уровень, при котором в транзакции допускаются несогласованные данные, то есть степень изолированности одной транзакции от другой.

При параллельном выполнении транзакций возможны следующие проблемы:

* потерянное обновление — при одновременном изменении одного блока данных разными транзакциями, одно из изменений теряется;
* «грязное» чтение — чтение данных, добавленных или изменённых транзакцией, которая впоследствии не подтвердится (откатится);
* неповторяющееся чтение — при повторном чтении в рамках одной транзакции, ранее прочитанные данные оказываются изменёнными;
* фантомное чтение — Одна транзакция в ходе своего выполнения несколько раз выбирает множество строк по одним и тем же критериям. Другая транзакция в интервалах между этими выборками добавляет или удаляет строки или изменяет столбцы некоторых строк, используемых в критериях выборки первой транзакции, и успешно заканчивается. В результате получится, что одни и те же выборки в первой транзакции дают разные множества строк.

Таблица 2.2.1. Предотвращение(+) проблем при разных уровнях изоляции

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Уровень изоляции** | **Фантомное чтение** | **Неповторяющееся чтение** | **«Грязное» чтение** | **Потерянное обновление** |
| SNAPSHOT TABLE STABILITY | + | + | + | + |
| SNAPSHOT | - | + | + | + |
| READ COMMITTED | - | - | + | + |

Уровень изоляции READ COMMITED

Эксперимент:

Первая сессия:

set transaction isolation level read committed;

Вторая сессия:

insert into genres values(11,’Temp1’);

Первая сессия:

select \* from genres;

Произошла блокировка, данные не прочитаны

Вторая сессия:

commit;

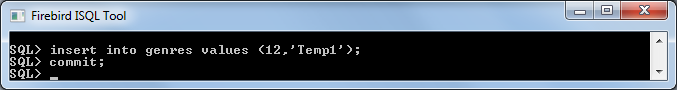
Первая сессия

Чтение произошло

Первая сессия:



Вторая сессия:



Эксперимент продемонстрировал блокировку на чтение неподтверждённых данных.

**Уровень изоляции SNAPSHOT**

Эксперимент:

Первая сессия

set transaction isolation level snapshot;

Вторая сессия

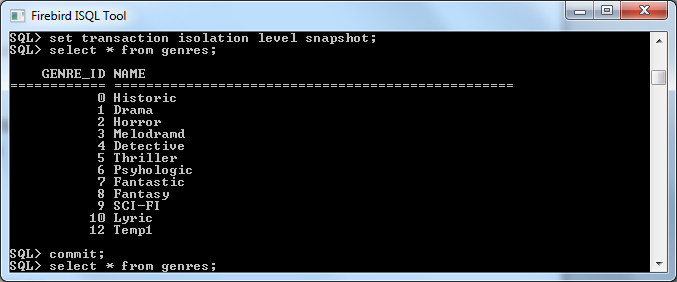
insert into VAGONS\_TYPES (VAGONS\_TYPES\_ID, VAGONS\_TYPES\_NAMES, NUMB\_OF\_PLACES)values (4, 'Lux-2', 100);

commit;

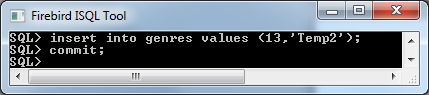
Первая сессия

select \* from vagons\_types;

Первая сессия



Вторая сессия



1 сессия:

Эксперимент продемонстрировал невозможность пронаблюдать неподтверждённые изменения, так как мы увидим старые значения на момент запуска транзакции

**Уровень изоляции SNAPSHOT TABLE STABILITY**

Эксперимент:

Первая сессия:

set transaction isolation level snapshot TABLE STABILITY;

Вторая сессия:

set transaction isolation level snapshot TABLE STABILITY;

Первая сессия:

update vagons\_types set numb\_of\_places=10 where vagons\_types\_id=4;

Вторая сессия:

select \* from vagons\_types

Произошла блокировка

Первая сессия:

Commit;

select \* from vagons\_types;

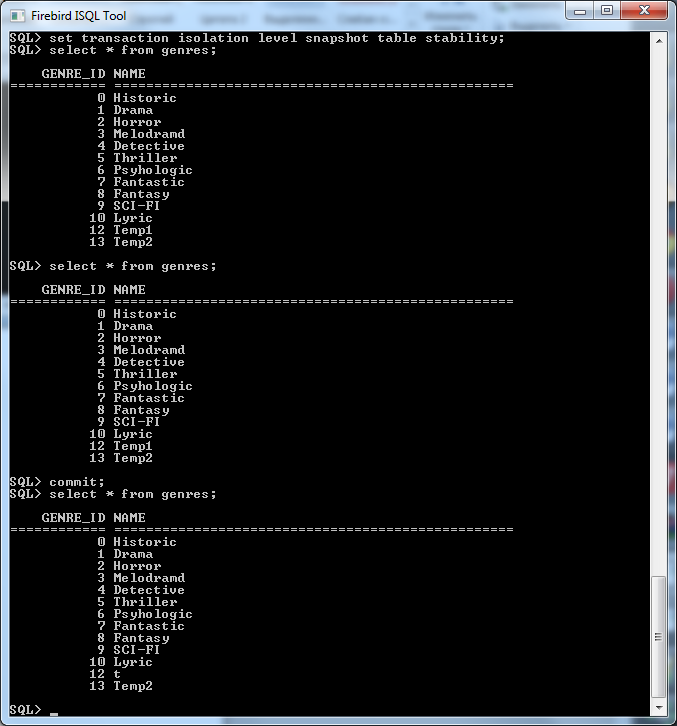
Вторая сессия:

select \* from vagons\_types;

1 Сессия



2 сессия:



Эксперимент продемонстрировал блокировку при обращении к данным, используемым транзакциями с уровнем изоляции **SNAPSHOT TABLE STABILITY.**

1. **Вывод**

В ходе работы были произведены эксперименты с заданием разного уровня изоляций транзакций . Изолирование необходимо для предотвращения ошибок, возникающих при параллельном доступе к данным. Более высокий уровень изолированности повышает точность данных, но при этом может снижаться количество параллельно выполняемых транзакций. С другой стороны, более низкий уровень изолированности позволяет выполнять больше параллельных транзакций, но снижает точность данных.