

Санкт-Петербургский государственный политехнический университет
Кафедра компьютерных систем и программных технологий

Отчёт по лабораторной работе

Дисциплина: Базы данных

Тема: Изучение работы транзакций

Выполнил студент гр. 43501/3

(подпись) В.Е. Бушин

Руководитель

(подпись) А.В. Мяснов

“ _ ” _____ 2015 г.

Санкт-Петербург

2015

1. Цель работы

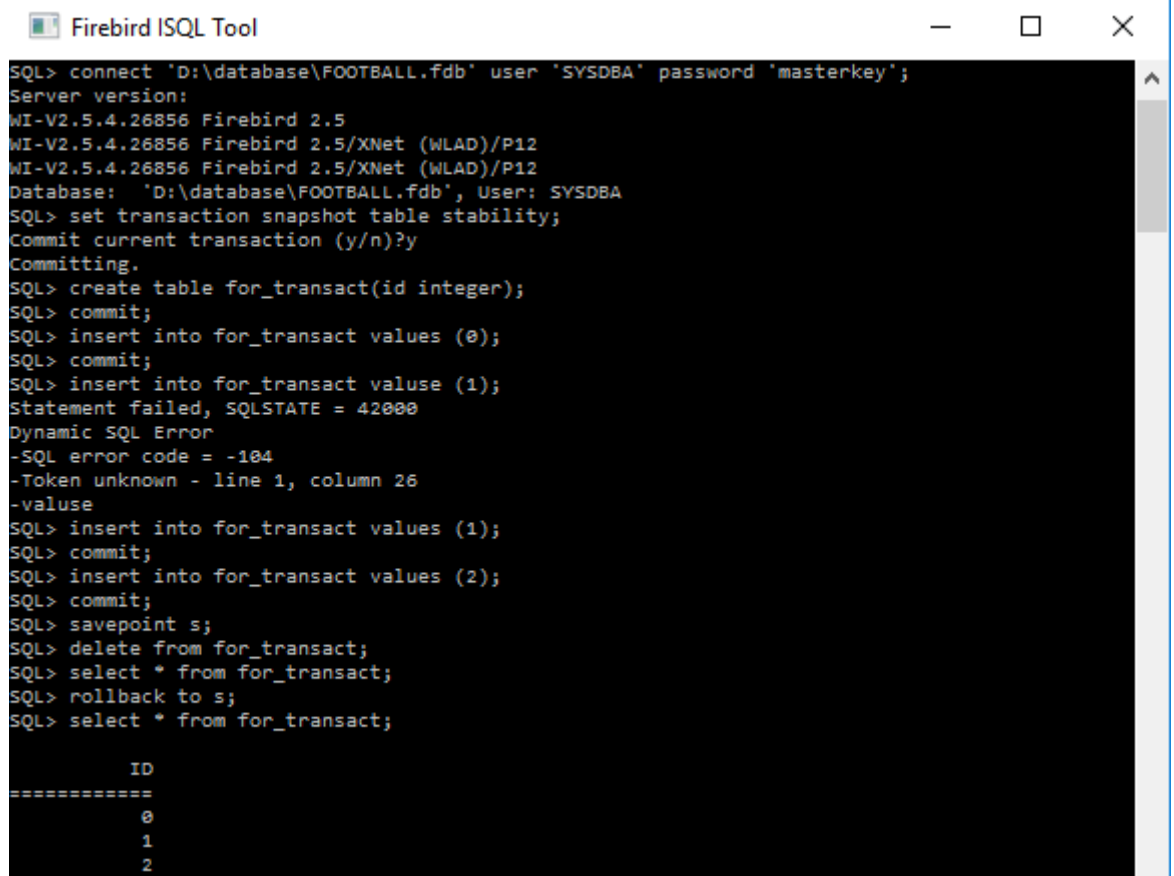
Познакомить студентов с механизмом транзакций, возможностями ручного управления транзакциями, уровнями изоляции транзакций.

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.

3. Ход работы

- 1) Изучены основные принципы работы транзакций.
- 2) Проведены эксперименты по запуску, подтверждению и откату транзакций:



```
SQL> connect 'D:\database\FOOTBALL.fdb' user 'SYSDBA' password 'masterkey';
Server version:
WI-V2.5.4.26856 Firebird 2.5
WI-V2.5.4.26856 Firebird 2.5/XNet (WLAD)/P12
WI-V2.5.4.26856 Firebird 2.5/XNet (WLAD)/P12
Database: 'D:\database\FOOTBALL.fdb', User: SYSDBA
SQL> set transaction snapshot table stability;
Commit current transaction (y/n)?y
Committing.
SQL> create table for_transact(id integer);
SQL> commit;
SQL> insert into for_transact values (0);
SQL> commit;
SQL> insert into for_transact value (1);
Statement failed, SQLSTATE = 42000
Dynamic SQL Error
-SQL error code = -104
-Token unknown - line 1, column 26
-value
SQL> insert into for_transact values (1);
SQL> commit;
SQL> insert into for_transact values (2);
SQL> commit;
SQL> savepoint s;
SQL> delete from for_transact;
SQL> select * from for_transact;
SQL> rollback to s;
SQL> select * from for_transact;

      ID
=====
      0
      1
      2
```

Рис.1. Опыты с запуском, подтверждением и откатом транзакций.

- 3) Изучены уровни изоляции транзакций в Firebird.
- 4) Проведены эксперименты, показывающие основные возможности транзакций с различным уровнем изоляции.

Опыты с уровнем изоляции snapshot: позволяет видеть только те изменения, фиксация которых произошла не позднее момента старта этой транзакции. Любые подтверждённые изменения, сделанные другими конкурирующими транзакциями, не будут видны в такой транзакции в процессе её активности без перезапуска (рис. 2).

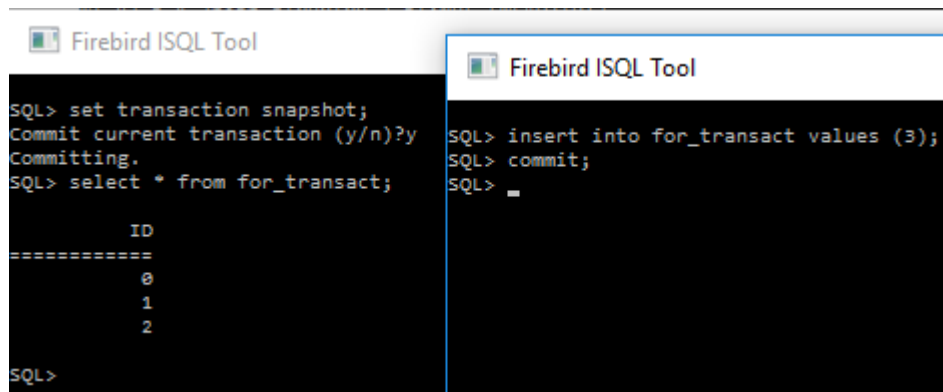


Рис.2.

Опыты с уровнем изоляции snapshot table stability: позволяет видеть только те изменения, фиксация которых произошла не позднее момента старта этой транзакции. При этом после старта такой транзакции в других клиентских транзакциях невозможно выполнение изменений ни в каких таблицах этой БД, уже каким-либо образом изменённых первой транзакцией (рис. 3).

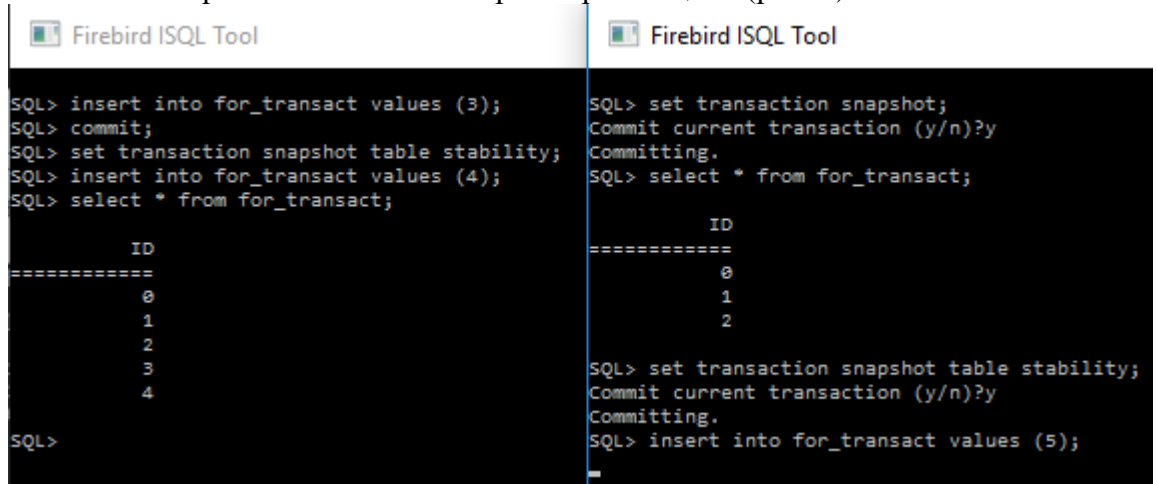


Рис.3.

Опыты с уровнем изоляции read committed: позволяет в транзакции без её перезапуска видеть все подтверждённые изменения данных БД, выполненные в других параллельных транзакциях. Неподтверждённые изменения не видны в транзакциях и этого уровня изоляции. С опцией record_version: транзакция всегда читает последнюю подтверждённую версию записей таблиц. В этом случае режим разрешения блокировок (WAIT или NO WAIT) никак не влияет на поведение транзакции при её старте (рис. 4).

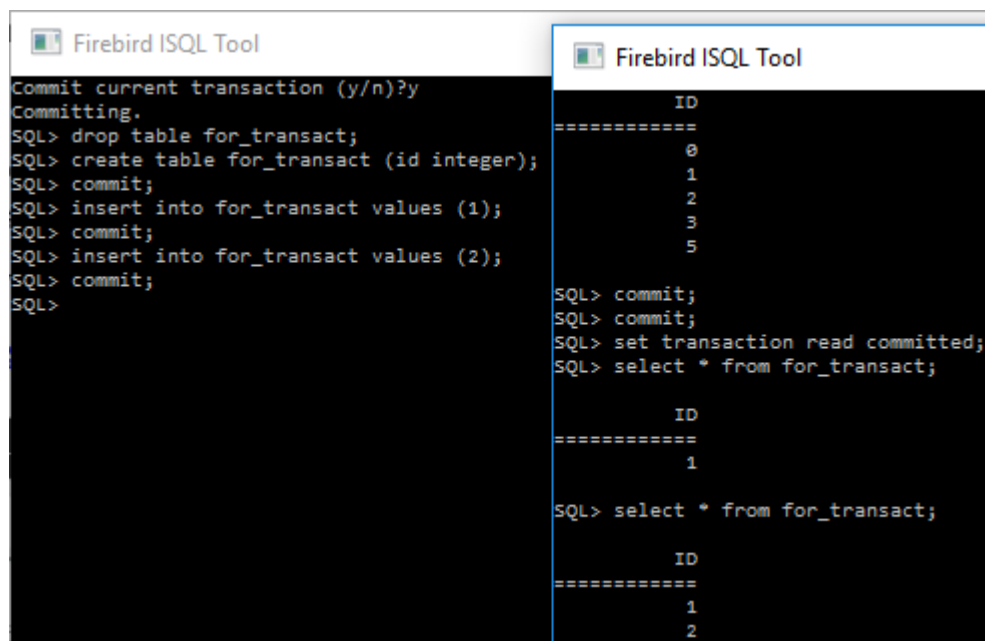


Рис.4

С опцией no record_version no wait: при обращении к таблице, изменённой в другой неподтверждённой транзакции, база выбросит исключение (рис. 5).

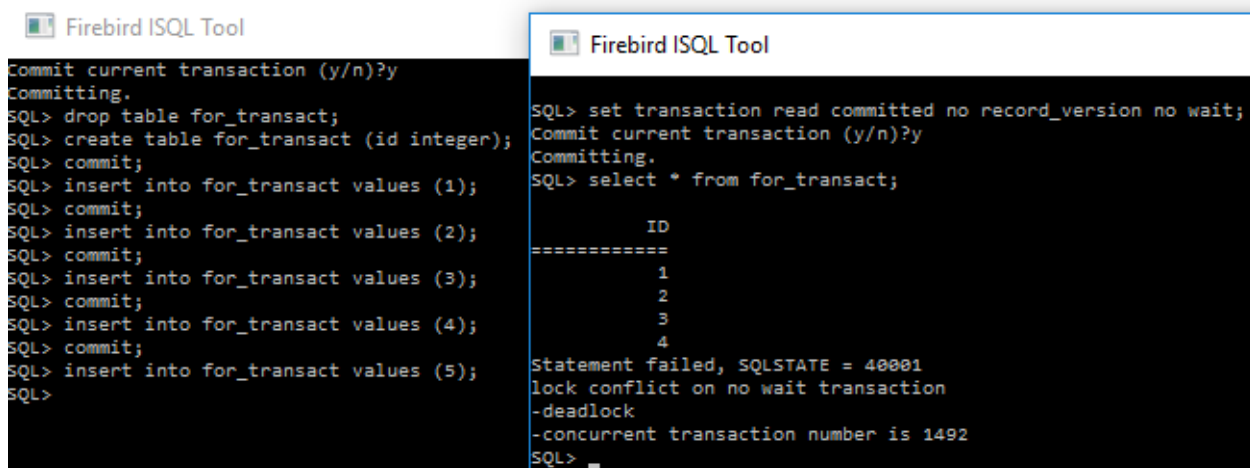


Рис.5

4. Вывод

Во время работы изучен механизм транзакций, возможности управления транзакциями и уровни изоляции транзакций в Firebird. Транзакции позволяют сохранять целостность БД при подключении к ней нескольких клиентов. Задавая различные типы разрешения конфликтов и уровни изоляции, можно управлять видимостью изменений, произошедших в базе, для разных пользователей.