**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ**

**РОССИЙСКОЙ ФЕДЕРАЦИИ**

Федеральное государственное автономное образовательное учреждение

высшего образования

«Санкт-Петербургский политехнический университет Петра Великого»

(ФГАОУ ВО «СПбПУ»)

Изображение выглядит как текст, старый, винтажный, нарисованный

Автоматически созданное описание**Институт среднего профессионального образования**

**ОТЧЕТ**

**По выполнению лабораторных работ**

**по предмету «МДК 11.01 Технология разработки и защиты БД»**

**Выполнил:** Лелюков Матвей Алексеевич

**Группа №** 32919/22

**Преподаватель:** Сынкова Алина Денисовна

**« »** 20 Оценка:

г. Санкт-Петербург

2025

**Цель работы:** изучение способов обеспечения надежной работы SQL Server с помощью механизма транзакций и контрольных точек, приобретение навыков управления транзакциями, а также ознакомление с физической и логической архитектурой журнала транзакций.

1. **Проверьте работу транзакции следующим образом. Выберите любое отношение в вашей базе данных. Создайте транзакцию, в которой будет содержаться один запрос вставки. Запустите ее дважды. В первом случае завершите транзакцию командой COMMIT, второй раз ROLLBACK. Проанализируйте результаты. Вставьте скриншот результата в отчет.**

Для проверки возьмем таблицу «Owner» из ипподрома

Код транзакции:

BEGIN TRANSACTION;

INSERT INTO Owner (NAME, CITY, PHONE)

VALUES ('Матвеев', 'Новосибирск', 1234);

COMMIT;

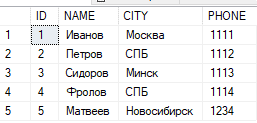


Рис 1 – запись была добавлена

Транзакция была проведена, а теперь попробуем провести транзакцию с ROLLABACK:

Код транзакции:

BEGIN TRANSACTION;

INSERT INTO Owner (NAME, CITY, PHONE)

VALUES ('Свинов', 'Тула', 9999);

ROLLBACK;

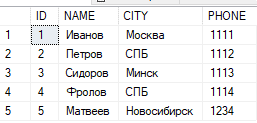


Рис 2 – запись не была добавлена

1. **Работа с вложенными транзакциями.**

**Создать вложенные транзакции, выполнив следующие команды:**

**CREATE TABLE table\_name (a1 int) –– 0-й уровень**

**BEGIN TRANSACTION tr1-– 1-й уровень**

**INSERT INTO table\_name VALUES (11)**

**BEGIN TRANSACTION tr2-– 2-й уровень**

**INSERT INTO table\_name VALUES (22)**

**BEGIN TRANSACTION tr3 –– 3-й уровень**

**INSERT INTO table\_name VALUES (33)**

**SELECT \* FROM table\_name**

**SELECT 'Вложенность транзкций', @@TRANCOUNT**

**ROLLBACK TRANSACTION**

**SELECT \* FROM table\_name –– откат на 0-й уровень**

**SELECT 'Вложенность транзакций', @@TRANCOUNT**

**Проанализируйте результаты.**

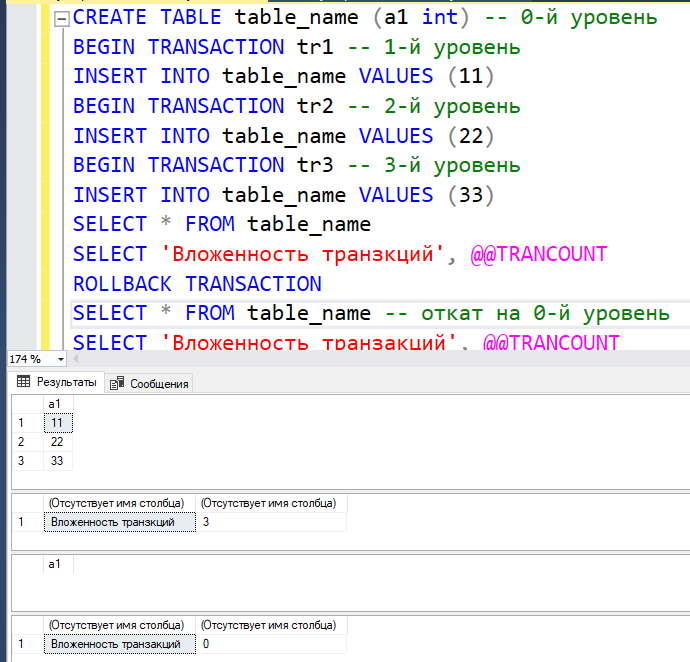


Рис 3 – вложенные транзакции

В этом запросе были созданы три транзакции вносящие записи в таблицу, а после произошел откат всей транзакции полностью, несмотря на вложенность.

1. **Работа с данными из файла groups. Импортируйте все листы. В первом листе находятся данные студентов, поступивших в 2018 году. В других листах содержаться списки групп. Задача импортировать данные из списков групп в сводную таблицу всех поступивших. Создайте явную транзакцию, в которой данные о студентах будут вноситься, а, если эти данные уже были внесены раннее, то транзакция будет откатываться. Результаты продемонстрируйте в отчете**.

Для начала импортируем все данные из файла:

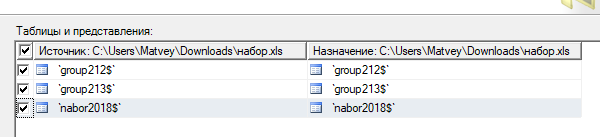


Рис 4 – листы документа

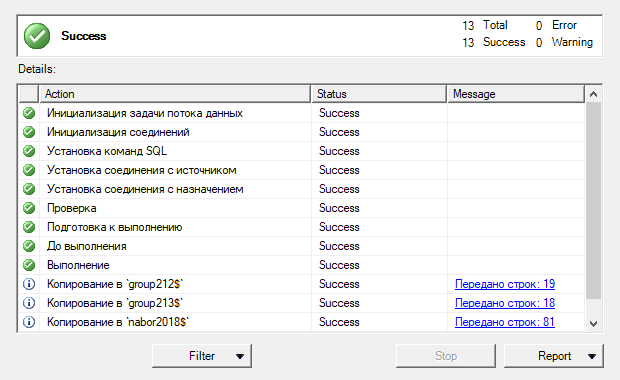


Рис 5 – успешное импортирование

Код транзакции:

BEGIN TRANSACTION Groups;

CREATE TABLE NewTable (ФИО NVARCHAR(255), группа INT);

INSERT INTO NewTable

SELECT \* FROM group212$;

INSERT INTO NewTable

SELECT \* FROM group213$;

INSERT INTO NewTable

SELECT \* FROM nabor2018$;

IF EXISTS (

SELECT 1 FROM NewTable

GROUP BY ФИО, группа

HAVING COUNT(\*) > 1

)

BEGIN

PRINT 'Обнаружены одинаковые записи, произошел откат транзакции.';

ROLLBACK TRANSACTION insertGroups;

END

ELSE

BEGIN

PRINT 'Транзакция прошла успешно.';

COMMIT TRANSACTION insertGroups;

END

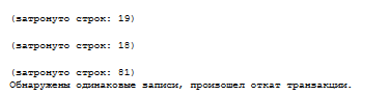


Рис 6 – результат выполнения транзакции «Groups»

1. **Изучите самостоятельно команду SET IMPLICIT\_TRANSACTIONS ON/OFF Продемонстрируйте разницу между этими режимами на примере любой таблицы в вашей базе данных.**

Чтобы продемонстрировать разницу создадим тестовую таблицу:

Код запроса:

CREATE TABLE Test (

ID INT PRIMARY KEY,

Name NVARCHAR(MAX)

);

SET IMPLICIT\_TRANSACTIONS OFF – стоит в транзакциях по умолчанию. Происходит коммит каждой команды, каждая отдельная команда автоматически выполняется как самостоятельная транзакция

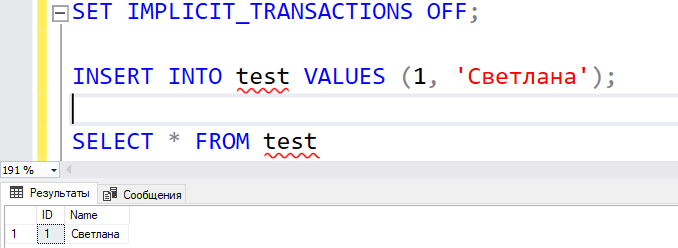


Рис 6 – код транзакции с IMPLICIT\_TRANSACTIONS OFF

SET IMPLICIT\_TRANSACTIONS ON – требует явного указания ROLLBACK/COMMIT, начинается автоматически при выполнении команд DML.

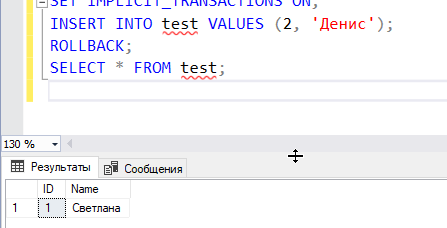


Рис 7 – код транзакции с IMPLICIT\_TRANSACTIONS ON