**Лабораторная работа № 13**

**Создание и управление транзакциями**

**Цель:** 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.

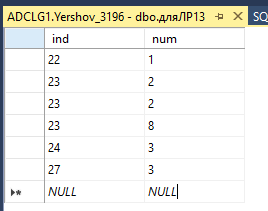
1. Проверка работы транзакции. Выбрал отношение базе данных. Создал транзакцию, в которой содержится один запрос вставки. Запустил ее дважды. В первом случае завершил транзакцию командой COMMIT, второй раз ROLLBACK.

BEGIN TRANSACTION lr13

INSERT INTO dbo.дляЛР13 (ind, num)

VALUES (27,3);

COMMIT TRANSACTION lr13

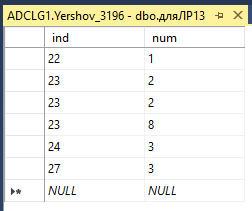


BEGIN TRANSACTION lr13

INSERT INTO dbo.дляЛР13 (ind, num)

VALUES (27,3);

ROLLBACK TRANSACTION lr13



Выводы: в результате использования команды COMMIT транзакция выполняется, а в результате команды ROLLBACK происходит откат изменений транзакции.

1. Работа с вложенными транзакциями. Создать вложенные транзакции, выполнив следующие команды:

CREATE TABLE [lr13-2] (a1 int)

BEGIN TRANSACTION tr1

INSERT INTO [lr13-2] VALUES (11)

BEGIN TRANSACTION tr2

INSERT INTO [lr13-2] VALUES (22)

BEGIN TRANSACTION tr3

INSERT INTO [lr13-2] VALUES (33)

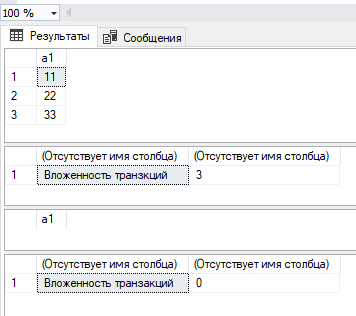
SELECT \* FROM [lr13-2]

SELECT 'Вложенность транзкций', @@TRANCOUNT

ROLLBACK TRANSACTION

SELECT \* FROM [lr13-2]

SELECT 'Вложенность транзакций', @@TRANCOUNT



Выводы: происходит создание таблицы lr13-2, далее код начал 3 транзакции, в которых добавил в таблицу по строчке данных (11,22,33). Затем код выбрал данные из этой таблицы, и стали видны 3 строки данных. Далее с помощью функции @@TRANCOUNT произошел вывод количества активных транзакций – 3. Командой ROLLBACK TRANSACTION закончил все транзакции с откатом изменений (все, тк не указано имя конкретной транзакции). После вывода на экран повторно списка активных транзакций результатом является - 0.

1. Работа с данными из файла groups. Импортировал 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.

BEGIN TRANSACTION [lr13-3]

INSERT INTO dbo.набор2018 (ФИО, группа)

SELECT \* FROM dbo.group212 WHERE фио NOT IN (SELECT ФИО FROM dbo.набор2018);

SELECT \* FROM dbo.group213 WHERE фио NOT IN (SELECT ФИО FROM dbo.набор2018);

COMMIT TRANSACTION [lr13-)]



1. SET IMPLICIT\_TRANSACTIONS {ON | OFF}

Присвоение параметру SET IMPLICIT\_TRANSACTIONS значения ON устанавливает для соединения режим неявных транзакций.

Для транзакций, автоматически открываемых в результате присвоения этому параметру значения ON, пользователю необходимо явно выполнить в конце транзакции фиксацию или откат. Иначе после отключения пользователя будет выполнен откат транзакции и будут потеряны все сделанные ей изменения данных.

Значение OFF возвращает соединение в режим с автоматической фиксацией транзакций.

После фиксации транзакции выполнение одной из указанных инструкций (см. таблицу) начнет новую транзакцию.

Когда соединение находится в режиме неявных транзакций, выполнение любой из следующих инструкций начнет транзакцию.

**Контрольные вопросы:**

1.Зачем нужны транзакции?

Для повышения надежности работы системы. Выполнение транзакций не нарушает целостность данных БД и откатывает их в случае сбоя.

2.С какими режимами транзакции вы работали?

Режим автофиксации – каждый оператор – транзакция. По завершению оператора транзакция завершается.

Явный режим – указание начала и конца транзакции.

Неявный режим – транзакция начинается автоматически при использовании определённых операторов T-SQL. Конец транзакции обозначается COMMIT или ROLLBACK.

3.Какие команды являются признаком начала неявной транзакции?

Операторы, для начала транзакции в неявном режиме:

* ALTER TABLE
* CREATE
* DELETE
* DROP
* FETCH
* GRANT
* INSERT
* OPEN
* REVOKE
* SELECT
* TRUNCATE TABLE
* UPDATE

4.Зачем нужны контрольные точки транзакций? Продемонстрируйте их работу.

Точка транзакции позволяет выполнять откат только до определенной точки в транзакции, а не до самого начала транзакции. Все модификации, выполненные до точки сохранения, остаются в силе и не подвергаются откату, но для всех операторов, выполняемых после точки сохранения вплоть до оператора ROLLBACK, будет выполнен откат. Затем начнут выполняться операторы, следующие за оператором ROLLBACK. Если вы затем зададите откат транзакции без указания точки сохранения, то, как обычно, будет выполнен откат всех модификаций вплоть до начала данной транзакции, т.е. будет отменена вся транзакция.

5.Можно ли узнать количество активных транзакций?

Функция @@TRANCOUNT предназначена для определения количества активных транзакций, начатых в активном соединении.