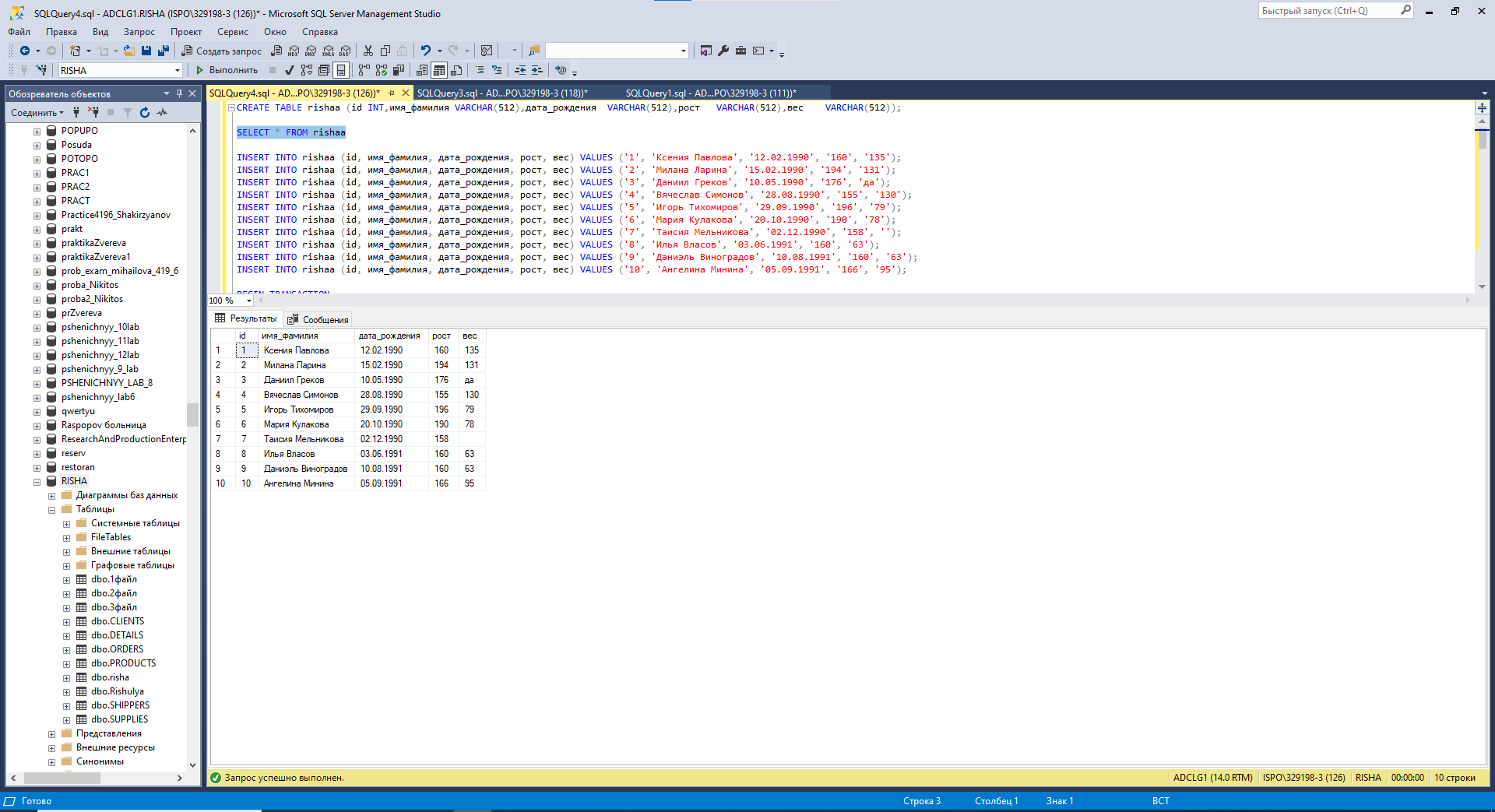
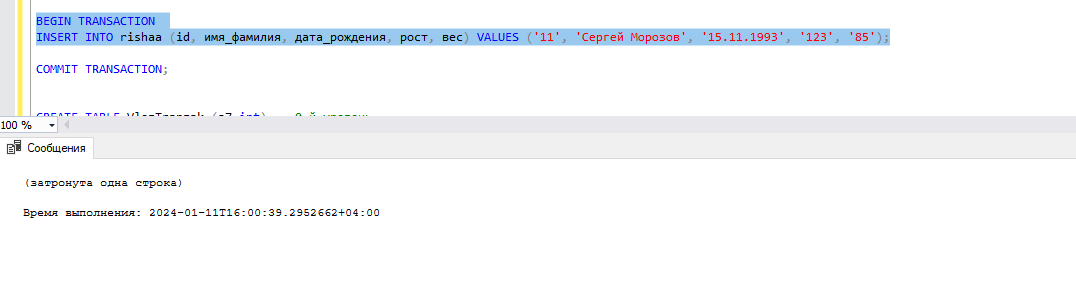
Лабораторная работа №13

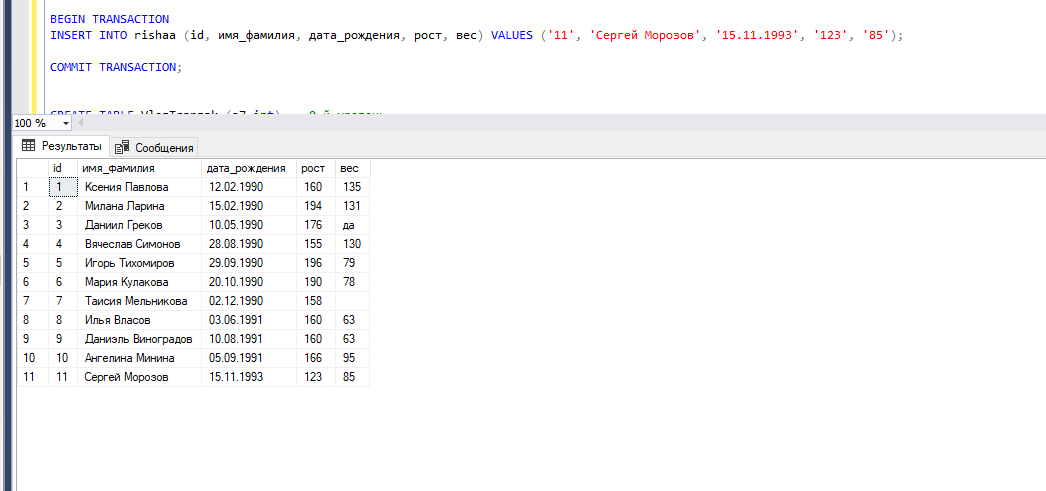
Цель: Изучение назначения и типов курсоров, синтаксиса языка T – SQL для создания и открытия курсоров, выборки данных из курсора и изменения строк таблиц с помощью курсоров, удаления данных, закрытия и освобождения курсоров, а также приобретения навыков их применения и управления с помощью команд и системных хранимых процедур SQL Server.

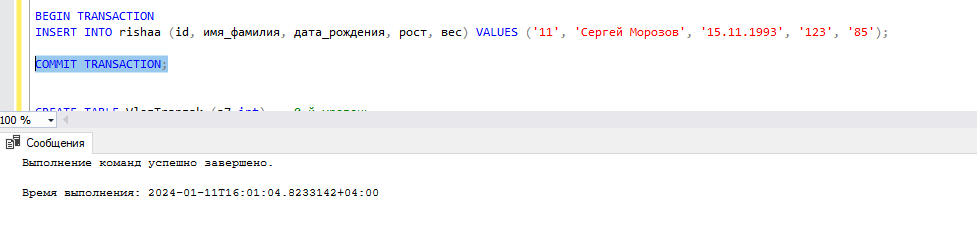
1.Проверьте работу транзакции следующим образом. Выберите любое отношение в вашей базе данных. Создайте транзакцию, в которой будет содержаться один запрос вставки. Запустите ее дважды. В первом случае завершите транзакцию командой COMMIT, второй раз ROLLBACK. Проанализируйте результаты. Вставьте скриншот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

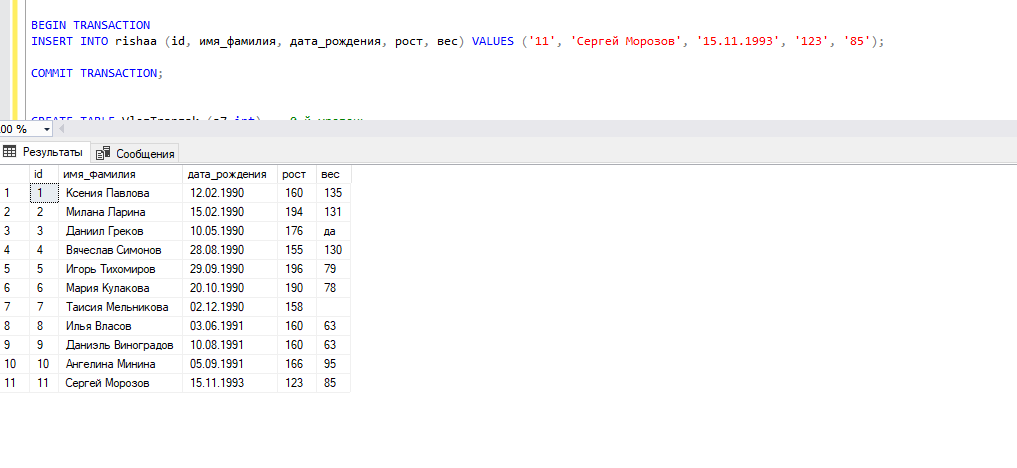


COMMIT:

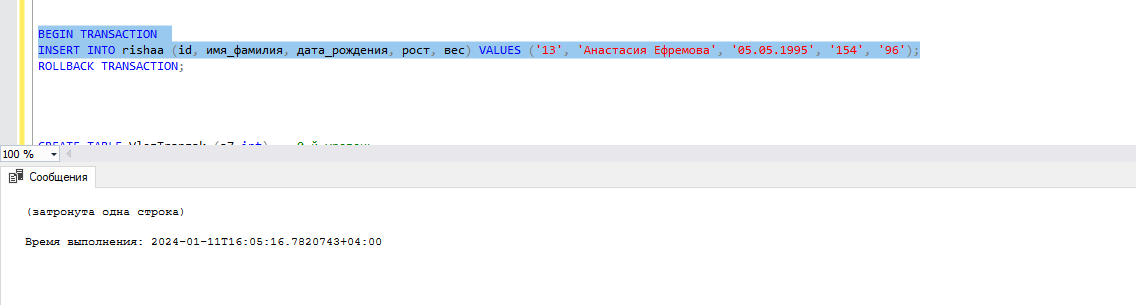


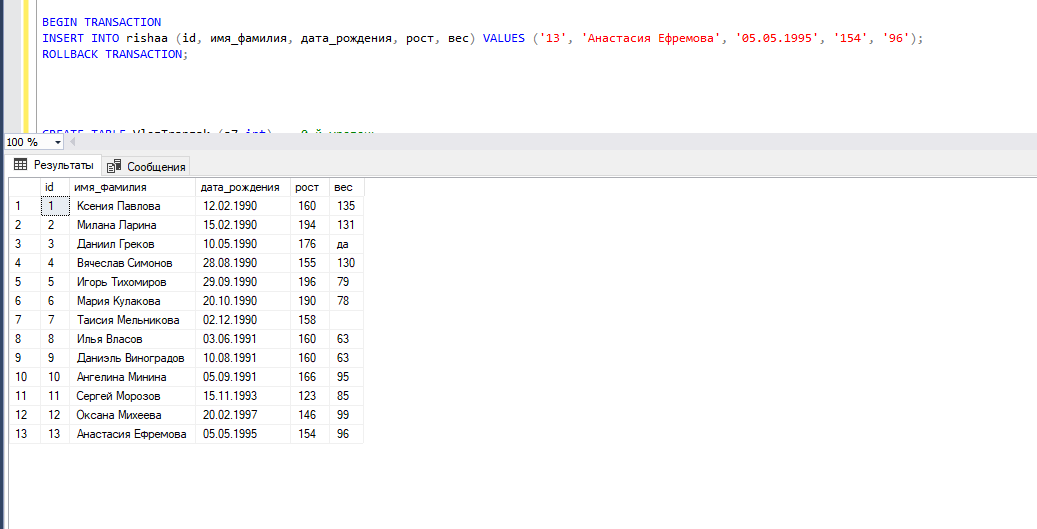


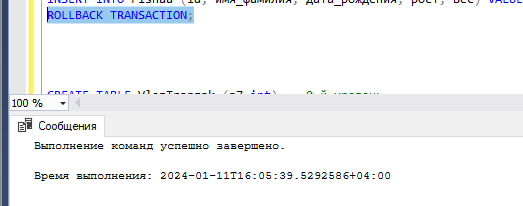


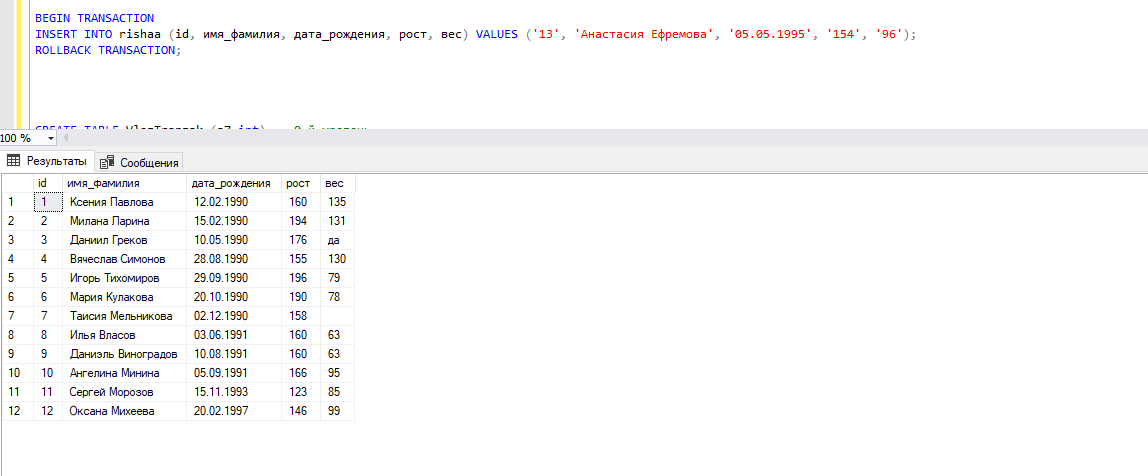


ROLLBACK:









2.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)

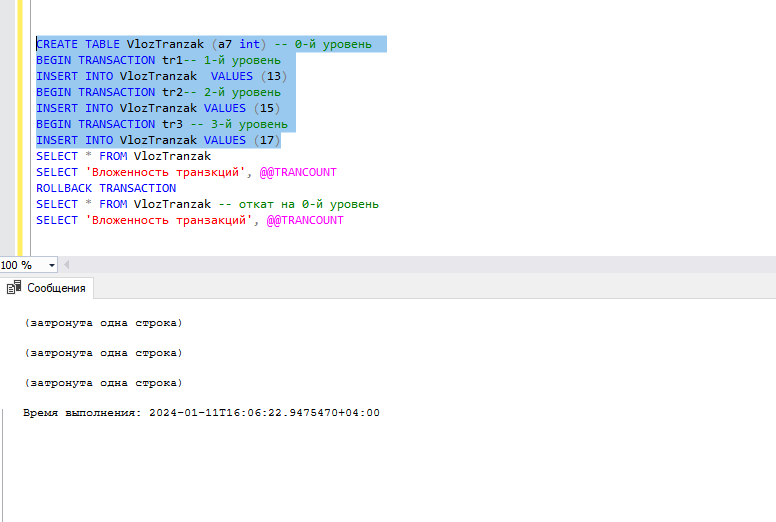
SELECT \* FROM table\_name

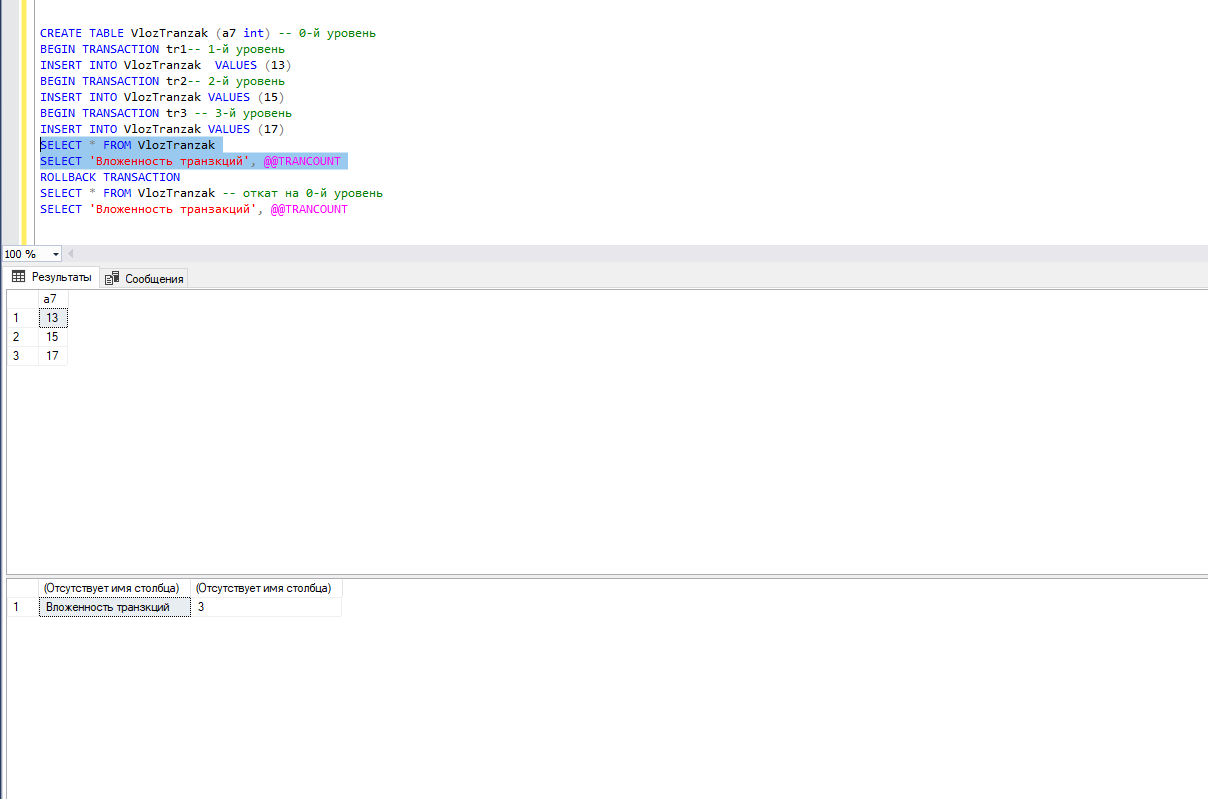
SELECT 'Вложенность транзкций', @@TRANCOUNT

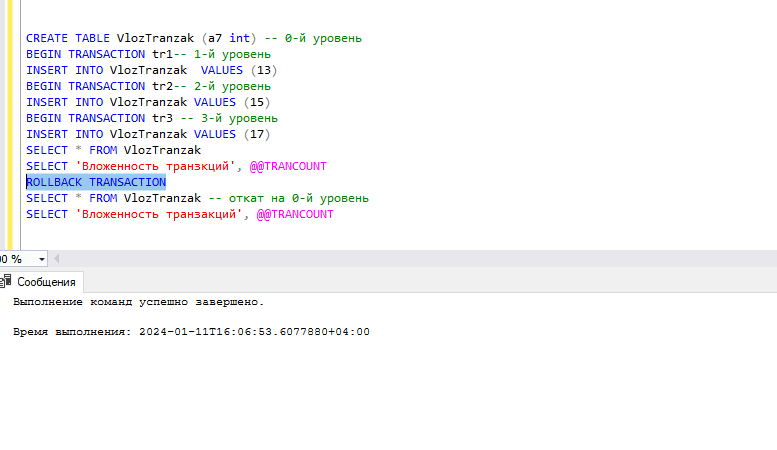
ROLLBACK TRANSACTION

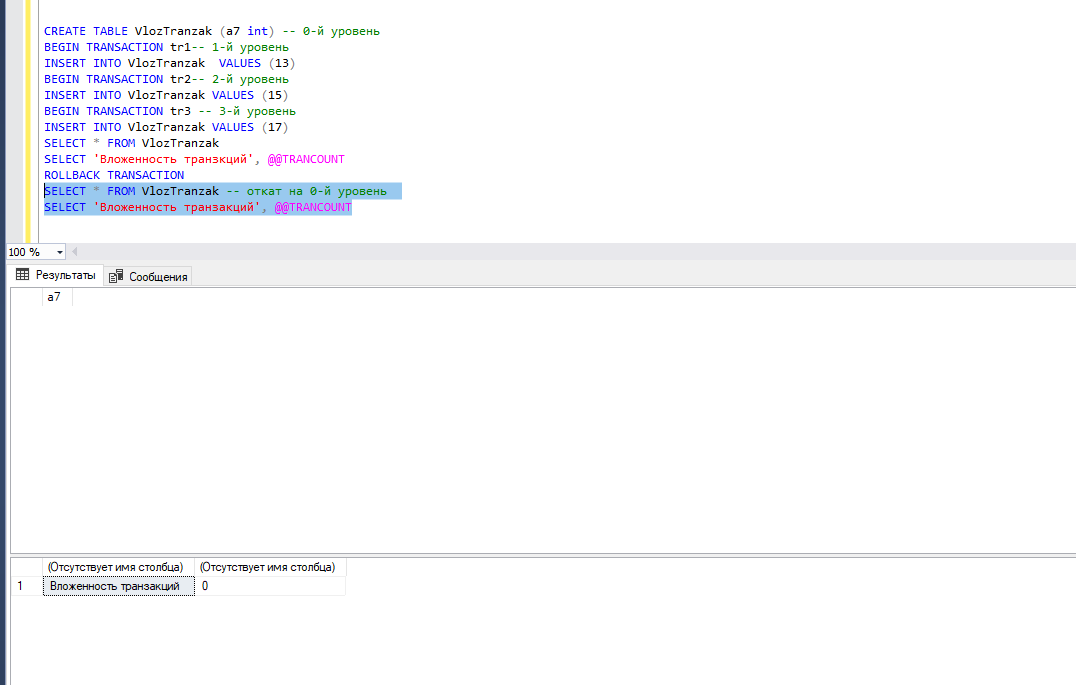
SELECT \* FROM table\_name -- откат на 0-й уровень

SELECT 'Вложенность транзакций', @@TRANCOUNT

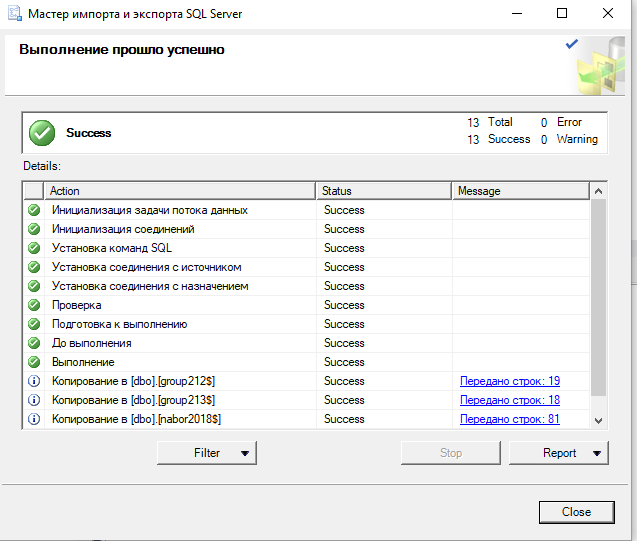


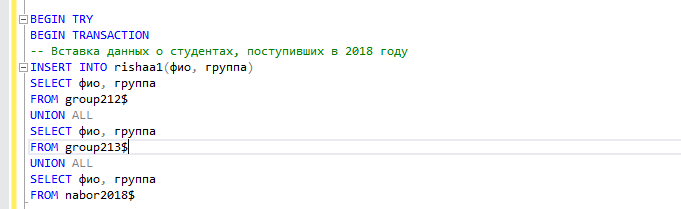


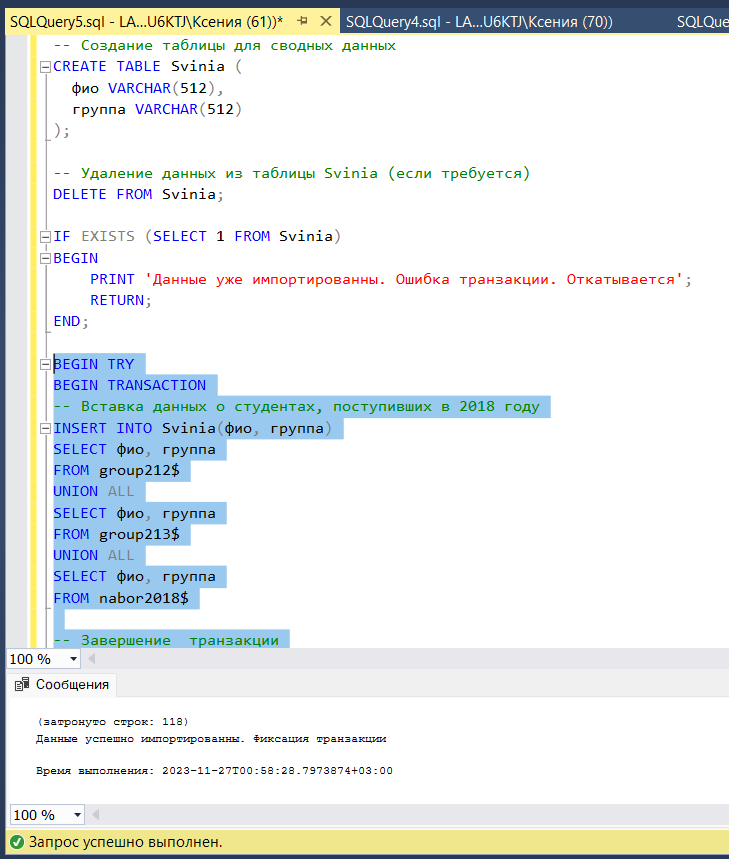


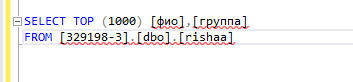


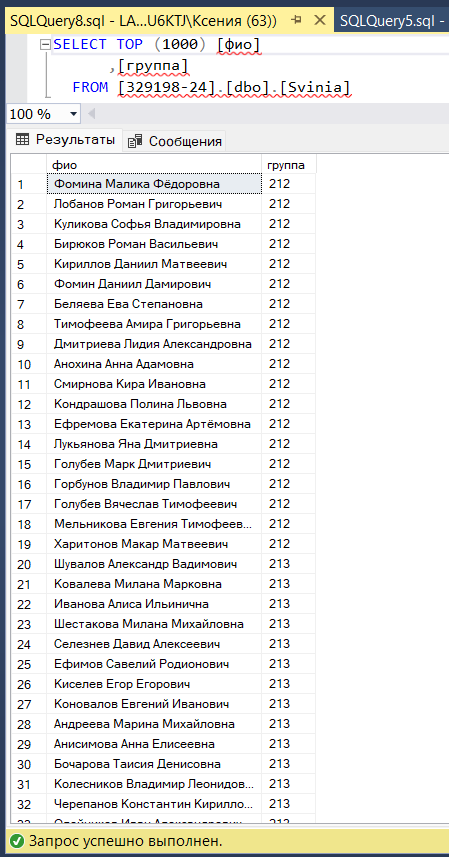
3. Работа с данными из файла groups. Импортируйте все листы. В первом листе находятся данные студентов, поступивших в 2018 году. В других листах содержаться списки групп. Задача импортировать данные из списков групп в сводную таблицу всех поступивших. Создайте явную транзакцию, в которой данные о студентах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

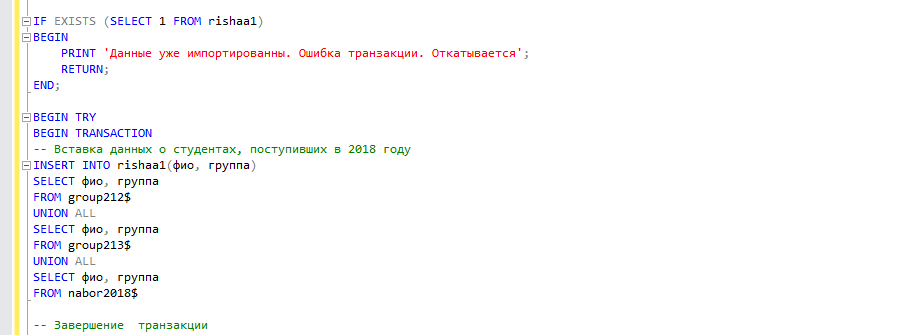


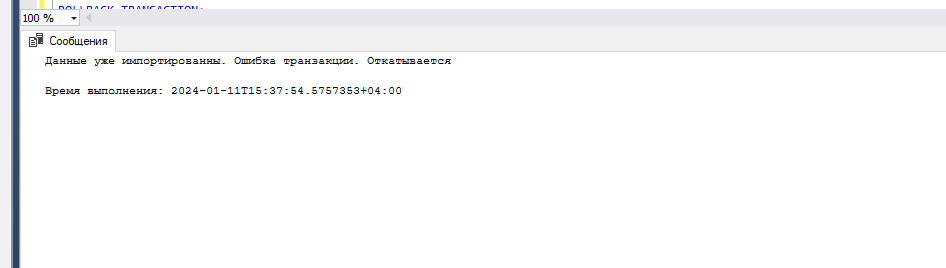








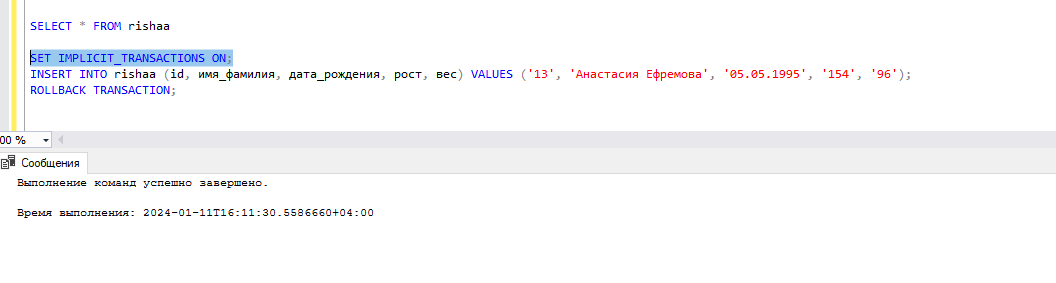


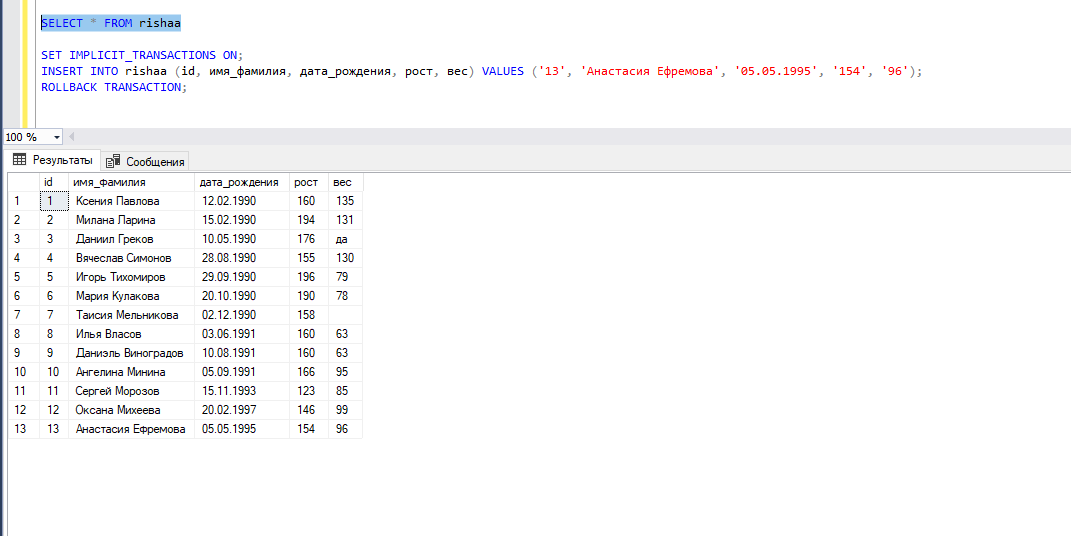


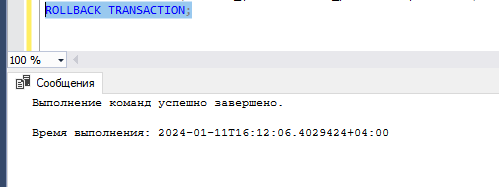
4.Изучите самостоятельно команду SET IMPLICIT\_TRANSACTIONS ON/OFF.

Продемонстрируйте разницу между этими режимами на примере любой таблицы в вашей базе данных.

SET IMPLICIT\_TRANSACTIONS ON:

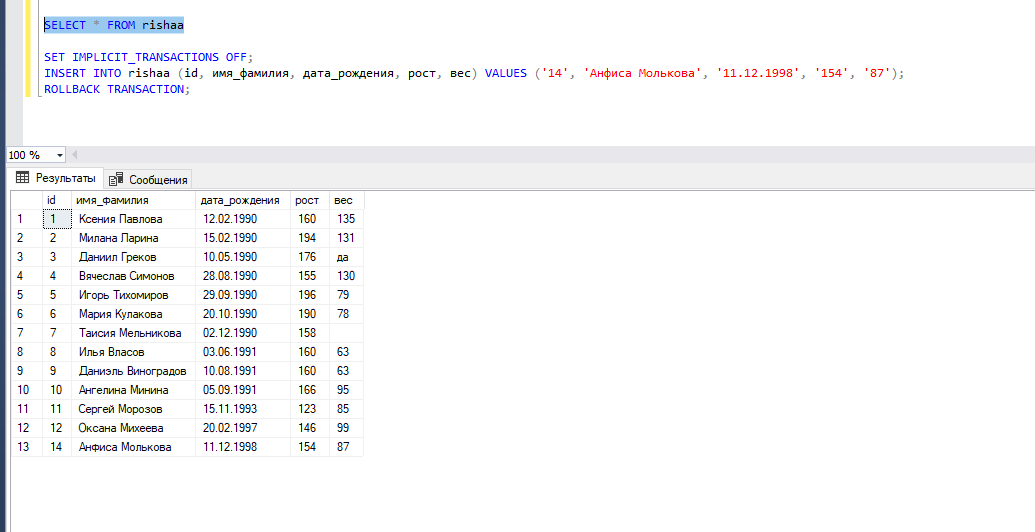


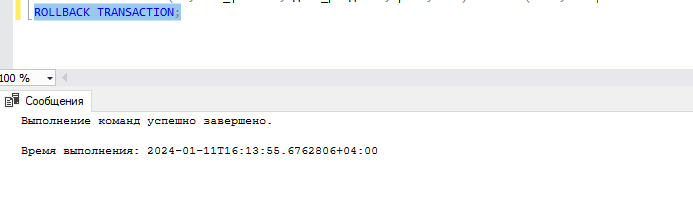






SET IMPLICIT\_TRANSACTIONS OFF:





Контрольные вопросы:

1. Зачем нужны транзакции?

Транзакции используются для обеспечения целостности и надежности операций на базе данных. Они гарантируют, что группа операций будет выполнена либо полностью, либо не будет выполнена вообще.

1. С какими режимами транзакции вы работали?

* Auto-Commit
* Явные транзакции
* Неявные транзакции

1. Какие команды являются признаком начала неявной транзакции?

INSERT

UPDATE

DELETE

CREATE

ALTER

DROP

Когда одна из этих команд выполняется, по умолчанию начинается неявная транзакция, и изменения будут зафиксированы при команде COMMIT или отменены при команде ROLLBACK.

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

Контрольные точки транзакций (Savepoints) позволяют установить точку внутри транзакции, до которой можно откатиться при необходимости. Они полезны, когда нужно временно сохранить состояние транзакции и потенциально откатиться до этой точки без полного отмены всей транзакции.

BEGIN; -- Начало транзакции

INSERT INTO table\_name (column1, column2) VALUES (value1, value2); -- Выполнение операции

SAVEPOINT savepoint1; -- Установка контрольной точки

DELETE FROM table\_name WHERE condition; -- Выполнение операции

ROLLBACK TO SAVEPOINT savepoint1; -- Откат до контрольной точки

COMMIT; -- Завершение транзакции

1. Можно ли узнать количество активных транзакций?

Чтобы узнать количество активных транзакций в СУБД, можно использовать системную представление или команду в зависимости от конкретной СУБД. Например, в PostgreSQL:

SELECT count(\*) FROM pg\_stat\_activity WHERE state = 'active';