**Базы данных**

**Отчет по лабораторной работе №11, Лешук Дмитрий ПОИБМС 7-2**

**Цель:** Изучить особенности использования транзакций.

Транзакция - Одна или несколько команд SQL, которые либо успешно выполняются как единое целое, либо отменяются как единое целое. Или же Логическая единица работы, обеспечивающая переход базы данных из одного согласованного состояния в другое согласованное состояние.

Основные свойства транзакции: атомарность (операторы изменения БД, включенные в транзакцию, либо выполнятся все, либо не выполнится ни один); согласованность (транзакция должна фиксировать новое согласованное состояние БД); изолированность (отсутствие взаимного влияния параллельных транзакций на результаты их выполнения); долговечность (изменения в БД, выполненные и зафиксированные транзакцией, могут быть отменены только с помощью новой транзакции).

Режим **неявной транзакции** может быть включен для текущего соединения с сервером БД с помощью специальной инструкции: **SET IMPLICIT\_TRANSACTIONS ON.**

Неявная транзакция начинается, если выполняется один из следующих операторов:

CREATE, DROP; ALTER TABLE; INSERT, DELETE, UPDATE, SELECT, TRUNCATE TABLE; OPEN, FETCH; GRANT (выдача разрешений), REVOKE (запрещение разрешений).

Неявная транзакция продолжается до тех пор, пока не будет выполнен оператор фиксации (COMMIT) или оператор отката (ROLLBACK) транзакции.

Рассмотрим на примере первого задания – рисунок 1.1. Неявная транзакция стартует при выполнении оператора CREATE TABLE и завершается фиксацией изменений с помощью оператора COMMIT. После этого осуществляется возврат в режим автофиксации (инструкция SET OFF). Созданная таблица c именем X сохранилась. Если в данном сценарии изменить значение переменной @flag, то будет выполняться оператор ROLLBACK, что приведет к отмене всех изменений в БД, осуществленных последней транзакцией (созданная таблица X не сохранится).

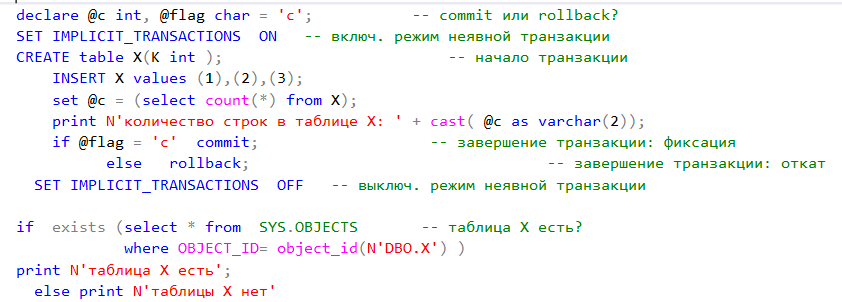


Рисунок 1.1 – Режим неявной транзакции

Явная транзакция — группа инструкций, начало и конец которой обозначаются инструкциями: BEGIN TRANSACTION, COMMIT, ROLLBACK. После завершения явной транзакции происходит возврат в исходный режим (автофиксации или неявной транзакции). Рассмотрим пример из второго задания – рисунок 1.2.

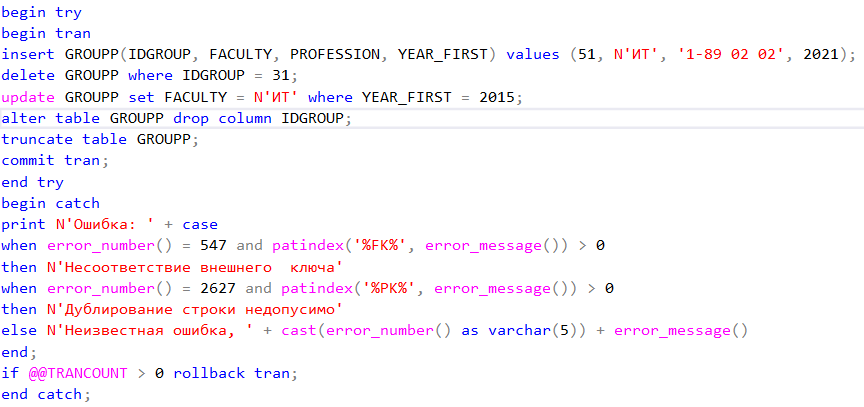


Рисунок 1.3 – Режим явной транзакции с блоком TRY…CATCH

Транзакция начинается внутри TRY-блока и в случае успешного выполнения завершается оператором COMMIT. Если при выполнении возникла ошибка, то в CATCH-блоке формируется и выводится соответствующее сообщение, а затем выполняется откат (ROLLBACK).

Системная функция @@TRANCOUNT возвращает уровень вложенности транзакции (если значение больше нуля, то транзакция не завершена). Изменения в таблице будут осуществлены только в случае отсутствия ошибок в операторах удаления и вставки.

Если транзакция состоит из нескольких независимых блоков операторов T-SQL, изменяющих базу данных, то может быть использован оператор SAVE TRANSACTION, формирующий контрольную точку транзакции. Инструкция ROLLBACK c такой же меткой вернёт базу данных в то состояние, которое зафиксировалось в этой точке. Задание 3 – рисунок 1.3.

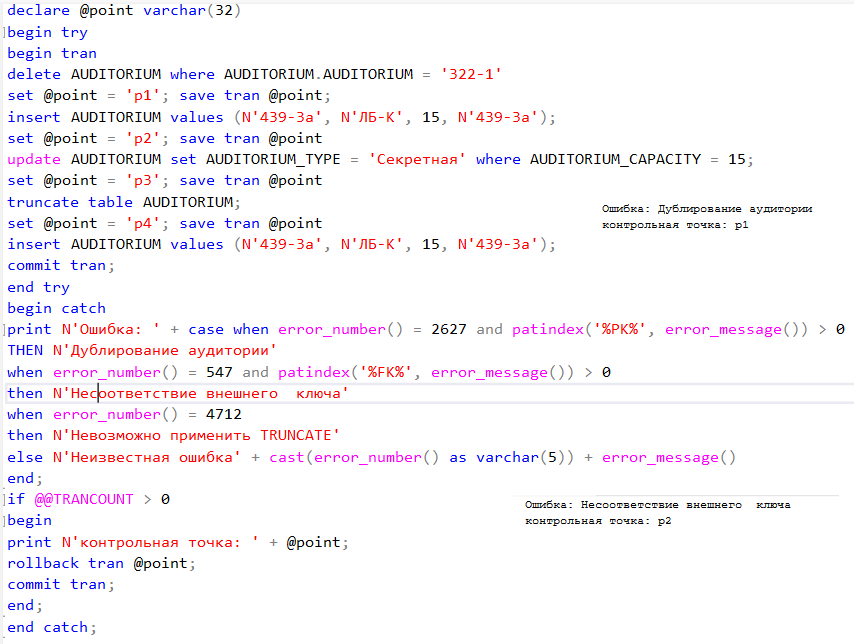


Рисунок 1.3 – Работа с точками сохранения

В данном примере, если происходит ошибка, выводится об этом сообщение, а также, в какой контрольной точке осталась база данных.

Уровень изоляции задает степень защищенности данных в транзакции от возможности изменения другими транзакциями. При параллельных транзакциях могут возникать три проблемы.

**Неподтвержденное чтение***.* До момента t1 транзакцией B выполняются два оператора: INSERT и UPDATE. Эти операторы изменяют таблицы БД, но до момента времени t2 не фиксируют и не откатывают эти изменения. После момента t1 транзакция A считывает содержимое таблиц, измененных транзакцией B и «видит» измененные или добавленные строки. При этом изменения остаются до момента t2 в неподтвержденном состоянии, т. е. могут быть как зафиксированными, так и отмененными.

**Неповторяющееся чтение***.* Одна транзакция читает данные несколько раз, а другая изменяет те же данные между двумя операциями чтения в первом процессе. По этой причине данные, прочитанные в различных операциях, будут разными.

**Фантомное чтение***.* Две последовательные операции чтения могут получать различные значения, т. к. дополнительные строки, называемые фантом­ными, могут добавляться другими транзакциями.

Чтобы такие проблемы не возникали, определяются различные уровни изоляции: READ COMMITED, REPEATABLE READ, SERIALIZABLE и др.

При **READ COMMITED** : не изолирует операции чтения других транзакций, транзакция не задает и не признает блокировок, допускает некоторые из перечисленных проблем.

**READ COMMITED** : транзакция выполняет проверку только на наличие монопольной блокировки для данной строки, является уровнем изоляции по умолчанию, но не все проблемы устранены.

**REPEATABLE READ** : Устанавливает разделяемые блокировки на все считываемые данные и удерживает эти блокировки до тех пор, пока транзакция не будет подтверждена или отменена, не препятствует другим инструкциям вставлять новые строки, из проблем остаётся лишь фантомное чтение.

**SERIALIZABLE** : Устанавливает блокировку на всю область данных, считываемых соответствующей транзакцией, предотвращает вставку новых строк другой транзакцией до тех пор, пока первая транзакция не будет подтверждена или отменена.

Установка осуществляется с помощью команды SET TRANSACTION ISOLATION LEVEL.

Транзакция, выполняющаяся в рамках другой транзакции, называется **вложенной**.

При работе с вложенными транзакциями нужно учитывать следующее:

− оператор COMMIT вложенной транзакции действует только на внутренние операции вложенной транзакции;

− оператор ROLLBACK внешней транзакции отменяет зафиксированные операции внутренней транзакции;

− оператор ROLLBACK вложенной транзакции действует на операции внешней и внутренней транзакции, а также завершает обе транзакции;

− уровень вложенности транзакции можно определить с помощью системной функции @@TRANCOUT.

**Вывод:**

В результате лабораторной работы были изучены : режимы явной и неявной транзакции и их особый синтаксис, работа с контрольными точками, уровни изоляции READ COMMITED, REPEATABLE READ, SERIALIZABLE, READ UNCOMMITED, а также проблемы, возникающие при выполнении параллельных транзакций.