Учреждение образования “БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ”

Лабораторная работа № 11. Особенности использования транзакций

Выполнил:

студент 2 курса, 7 группы

Бобрович Глеб Сергеевич

Минск 2022ПроТТеоре

|  |  |
| --- | --- |
| **Задание** |  |

|  |  |
| --- | --- |
| 1. Разработать сценарий, демонстрирующий работу в режиме *неявной* транзакции.  Проанализировать пример, приведенный справа, в котором создается таблица Х, и создать сценарий для другой таблицы. | **Задание 1**  USE ЛР11;  set nocount on  if exists (select \* from SYS.OBJECTS  where OBJECT\_ID=object\_id(N'DBO.TAB'))  drop table TAB;  declare @c int, @flag char = 'c'; -- если с->r, таблица не сохр  SET IMPLICIT\_TRANSACTIONS ON -- вкл режим неявной транзакции  create table TAB(K int );  insert TAB values (1),(2),(3),(4),(5);  set @c = (select count(\*) from TAB);  print 'кол-во строк в TAB: ' + cast(@c as varchar(2));  if @flag = 'c' commit -- фиксация  else rollback; -- откат  SET IMPLICIT\_TRANSACTIONS OFF -- действует режим автофиксации  if exists (select \* from SYS.OBJECTS  where OBJECT\_ID= object\_id(N'DBO.TAB')) print 'таблица TAB есть';  else print 'таблицы TAB нет' |
| 2.Разработать сценарий, демонстрирующий свойство *атомарности* *явной* транзакции на примере базы данных X\_UNIVER.  В блоке CATCH предусмотреть выдачу соответствующих сообщений об ошибках.  Опробовать работу сценария при использовании различных операторов модификации таблиц. | **Задание 2**  USE ЛР11;  begin try  begin tran -- начало явной транзакции  insert FACULTY values ('ДФ', 'Факультет других наук');  insert FACULTY values ('ПиМ', 'Факультет print-технологий');  commit tran; -- фиксация транзакции  end try  begin catch  print 'ошибка: '+ case  when error\_number() = 2627 and patindex('%FACULTY\_PK%', error\_message()) > 0 then 'дублирование ' --позиция 1-го вхождения  else 'неизвестная ошибка: '+ cast(error\_number() as varchar(5))+ error\_message()  end;  if @@trancount > 0 rollback tran; -- ур.вложенности тр.>0, транз не завершена  end catch;    DELETE FACULTY WHERE FACULTY = 'ДФ';  DELETE FACULTY WHERE FACULTY = 'ПиМ';  select \* from FACULTY; |
| 3.Разработать сценарий, демонстрирующий применение оператора SAVE TRAN на примере базы данных X\_UNIVER.  В блоке CATCH предусмотреть выдачу соответствующих сообщений об ошибках.  Опробовать работу сценария при использовании различных контрольных точек и различных операторов модификации таблиц. | **Задание 3**  USE ЛР11;  declare @point varchar(32);  begin try  begin tran  set @point = 'p1';  save tran @point; -- контрольная точка p1  insert STUDENT(IDGROUP, NAME, BDAY, INFO, FOTO) values  (20,'Екатерина', '1997-08-02', NULL, NULL),  (20,'Александра', '1997-08-06', NULL, NULL),  (20,'Елизавета', '1997-08-01', NULL, NULL),  (20,'Ольга', '1997-08-03', NULL, NULL);  set @point = 'p2';  save tran @point; -- контрольная точка p2  insert STUDENT(IDGROUP, NAME, BDAY, INFO, FOTO) values  (20, 'Особенный Студент', '1997-08-02', NULL, NULL);  commit tran;  end try  begin catch  print 'ошибка: '+ case  when error\_number() = 2627 and patindex('%STUDENT\_PK%', error\_message()) > 0 then 'дублирование студента'  else 'неизвестная ошибка: '+ cast(error\_number() as varchar(5)) + error\_message()  end;  if @@trancount > 0 -- если транзакция не завершена  begin  print 'контрольная точка: '+ @point;  rollback tran @point; -- откат к последней контр.точке  commit tran; -- фиксация изменений, выполн до контр.точки  end;  end catch;  select \* from STUDENT where IDGROUP=20;  delete STUDENT where IDGROUP=20; |
| 4.Разработать два сценария A и B на примере базы данных X\_UNIVER.  Сценарий A представляет собой явную транзакцию с уровнем изолированности READ UNCOMMITED, сценарий B – явную транзакцию с уровнем изолированности READ COMMITED (по умолчанию).  Сценарий A должен демонстрировать, что уровень READ UNCOMMITED допускает неподтвержденное, неповторяющееся и фантомное чтение. | **Задание 4**  USE ЛР11;  ------A---------  set transaction isolation level READ UNCOMMITTED  begin transaction  -----t1---------  select @@SPID, 'insert FACULTY' 'результат', \*  from FACULTY WHERE FACULTY = 'ИТ';  select @@SPID, 'update PULPIT' 'результат', \*  from PULPIT WHERE FACULTY = 'ИТ';  commit;  -----t2---------  -----B----------  begin transaction  select @@SPID  insert FACULTY VALUES  ('ИТ3','Информационных технологий');  update PULPIT set FACULTY = 'ИТ' WHERE PULPIT = 'ИСиТ'  -----t1----------  -----t2----------  ROLLBACK;  SELECT \* FROM FACULTY;  SELECT \* FROM PULPIT; |
| 5.Разработать два сценария A и B на примере базы данных X\_UNIVER.  Сценарии A и В представляют собой явные транзакции с уровнем изолированности READ COMMITED.  Сценарий A должен демонстрировать, что уровень READ COMMITED не допускает неподтвержденного чтения, но при этом возможно неповторяющееся и фантомное чтение. | **Задание** 5  USE ЛР11;  -----A--------  SELECT \* from PULPIT;  set transaction isolation level READ COMMITTED  begin transaction  select count(\*) from PULPIT  where FACULTY = 'ИТ'; --Указывает одно значение, видим результат 0  -----t1-------  -----t2-------  select 'update PULPIT' 'результат', count(\*) --здесь результат 1, т.к. произошло изменение  from PULPIT where FACULTY = 'ИТ'; --работает неповторяющееся чтение  commit;  ------B----  begin transaction  ------t1-----  update PULPIT set FACULTY = 'ИТ' where PULPIT = 'ИСиТ';  commit;  ------t2------ |
| 6.Разработать два сценария A и B на примере базы данных X\_UNIVER.  Сценарий A представляет собой явную транзакцию с уровнем изолированности REPEATABLE READ. Сценарий B – явную транзакцию с уровнем изолированности READ COMMITED.  Сценарий A должен демонстрировать, что уровень REAPETABLE READ не допускает неподтвержденного чтения и неповторяющегося чтения, но при этом возможно фантомное чтение. | **Задание 6**  USE ЛР11;  set transaction isolation level REPEATABLE READ  begin transaction  select TEACHER FROM TEACHER |
| 7.Разработать два сценария A и B на примере базы данных X\_UNIVER.  Сценарий A представляет собой явную транзакцию с уровнем изолированности SERIALIZABLE.  Сценарий B – явную транзакцию с уровнем изолированности READ COMMITED.  Сценарий A должен демонстрировать отсутствие фантомного, неподтвержденного и неповторяющегося чтения. | **Задание 7**  USE ЛР11;  select (select count(\*) from dbo.PULPIT where FACULTY = 'ИДиП') 'Кафедры ИДИПа',  (select count(\*) from FACULTY where FACULTY.FACULTY = 'ИДиП') 'ИДИП';  select \* from PULPIT  begin tran  begin tran  update PULPIT set PULPIT\_NAME='Кафедра ИДиПа' where PULPIT.FACULTY = 'ИДиП';  commit;  if @@TRANCOUNT > 0 rollback; |