

ЛАБОРАТОРНАЯ РАБОТА № 6.

Управление транзакциями и блокировками в SQL Server.


Цель: изучить влияние различных уровней изоляции транзакций на выполнение запросов.

Программное обеспечение: SQL Server Management Studio Express

Задание:

1. Исследование влияния различных уровней изоляции транзакций на выполнение запросов.

Пример потерянных обновлений.

- 1) Откройте SQL Server Management Studio и подключитесь к серверу баз данных.
- 2) Войдите в узел
- 1) Щелкните на панели инструментов кнопку "Создать запрос" 2 раза.
- 2) В меню Окно выберите  Создать группу вертикальных вкладок
- пункт
- 3) В первом открывшемся окне введите следующий код на языке SQL, меняющий количество вакансий по специальности АСОИУ на 50:

```
declare @кс int
select @кс=код_специальности
from специальности
where название='АСОИУ'

update специальности set вакансий=50
where код_специальности=@кс
waitfor delay '00:00:05'
select * from специальности
where код_специальности=@кс
```

- 4) Во втором открывшемся окне введите следующий код на языке SQL, меняющий количество вакансий по специальности АСОИУ на 40:

```
declare @кс int
```

```

select @кс=код_специальности
from специальности
where название='АСОИУ'
update специальности set вакансий=40
where код_специальности=@кс
select * from специальности
where код_специальности=@кс

```

- 5) Исполните первый введенный код SQL и, не дожидаясь окончания выполнения первого запроса, исполните второй введенный код SQL (см. рисунок 6.1). Обратите внимание на результат выполнения первого запроса: последний оператор SELECT сообщает о том, что количество вакансий по специальности АСОИУ равно 40 и, таким образом, результат выполнения оператора UPDATE теряется.

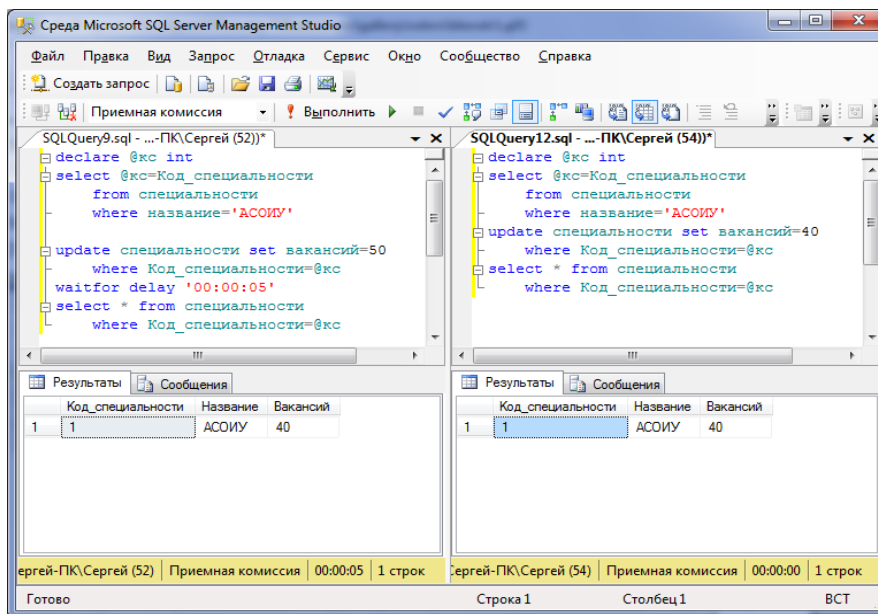


Рисунок 6.1 — Пример потерянных обновлений

- 6) Добавьте транзакцию в код на языке SQL, меняющий количество вакансий по специальности АСОИУ на 50:

```

begin tran

declare @кс int
select @кс=код_специальности
from специальности

```

```
where название='АСОИУ'

update специальности set вакансий=50
  where код_специальности=@кс
waitfor delay '00:00:05'
select * from специальности
  where код_специальности=@кс

commit tran
```

- 7) Добавьте транзакцию в код на языке SQL, меняющий количество вакансий по специальности АСОИУ на 40:

```
begin tran

declare @кс int
select @кс=код_специальности
  from специальности
  where название='АСОИУ'
update специальности set вакансий=40
  where код_специальности=@кс
select * from специальности
  where код_специальности=@кс

commit tran
```

- 8) Исполните первый введенный код SQL и, не дожидаясь окончания выполнения первого запроса, исполните второй введенный код SQL (см. рисунок 6.2). Обратите внимание на результат выполнения первого запроса: последний оператор SELECT сообщает о том, что количество вакансий по специальности АСОИУ равно 50 и, таким образом, результат выполнения оператора UPDATE уже не теряется.

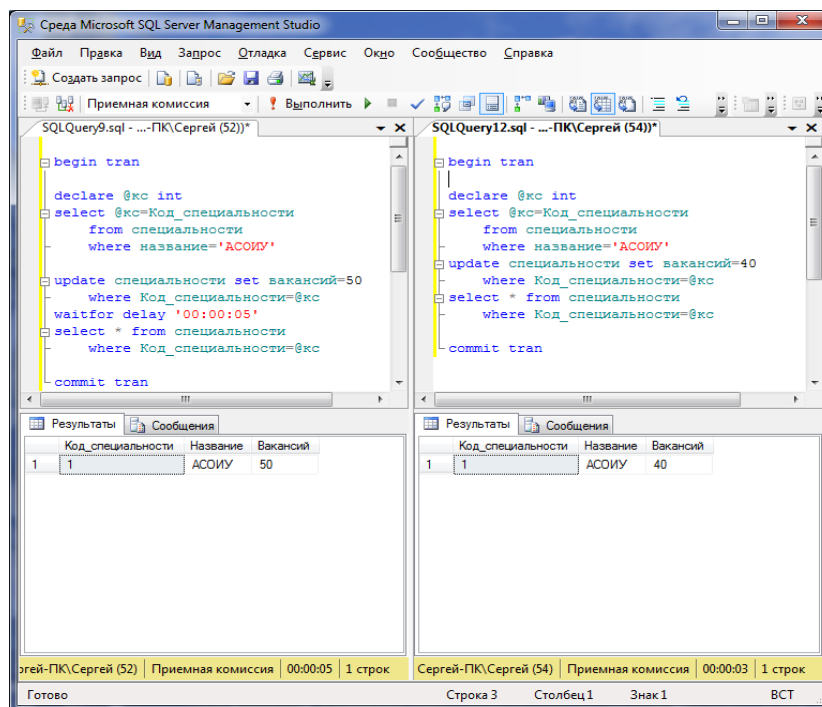


Рисунок 6.2 — Пример потерянных обновлений с использованием транзакции

Пример грязного чтения.

- 1) Откройте SQL Server Management Studio и подключитесь к серверу баз данных.
- 2) Щелкните на панели инструментов кнопку "Создать запрос" 2 раза.
- 3) В первом открывшемся окне введите следующий код на языке SQL, рассчитывающий количество зачисленных абитуриентов:

```
set transaction isolation level
    read uncommitted

select count(код_абитуриента)
    from абитуриенты
    where статус='зачислен'
```

С помощью оператора "set transaction isolation level read uncommitted" устанавливается уровень изоляции транзакции, допускающий грязное (неподтвержденное) чтение (read uncommitted).

- 4) Во втором открывшемся окне введите следующий код на языке SQL, меняющий статус абитуриента 2 на "зачислен":

```
begin tran
```

```
update абитуриенты
  set статус='зачислен'
  where код_абитуриента=2
waitfor delay '00:00:05'
```

```
rollback tran
```

В данной транзакции вначале происходит зачисление абитуриента, а затем происходит откат транзакции, отменяющий зачисление абитуриента.

5) Исполните второй введенный код SQL и, не дожидаясь окончания выполнения второго запроса, исполните первый введенный код SQL (см. рисунок 6.3). Обратите внимание на результат выполнения первого запроса: оператор SELECT сообщает о том, что количество зачисленных абитуриентов равно 4, хотя зачисление третьего абитуриента не подтверждается и фактически в базе данных есть только 3 подтвержденных абитуриента в статусе "зачислен".

6) Измените уровень изоляции в первом запросе до подтвержденного чтения (read committed):

```
set transaction isolation level
  read committed

select count(код_абитуриента)
  from абитуриенты
  where статус='зачислен'
```

7) Исполните второй введенный код SQL и, не дожидаясь окончания выполнения второго запроса, исполните первый введенный код SQL (см. рисунок 6.4). Обратите внимание на результат выполнения первого запроса: оператор SELECT сообщает о том, что количество зачисленных абитуриентов равно 3 и, таким образом, результат выполнения оператора UPDATE во втором запросе не учитывается.

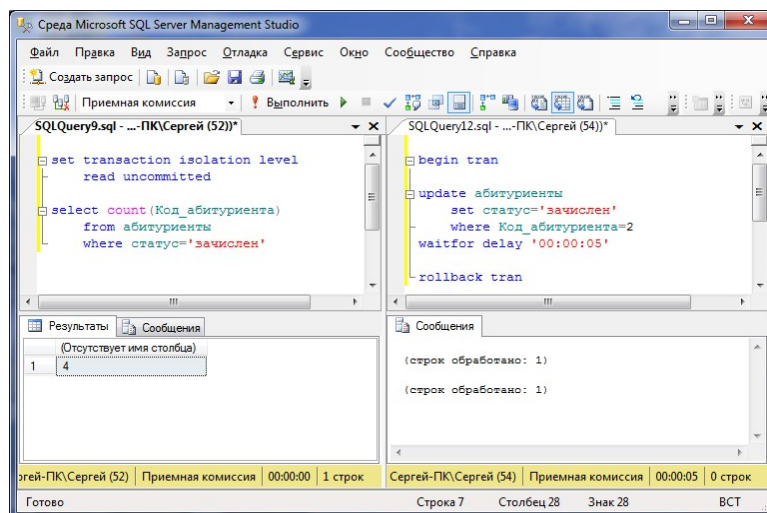


Рисунок 6.3 — Пример грязного чтения

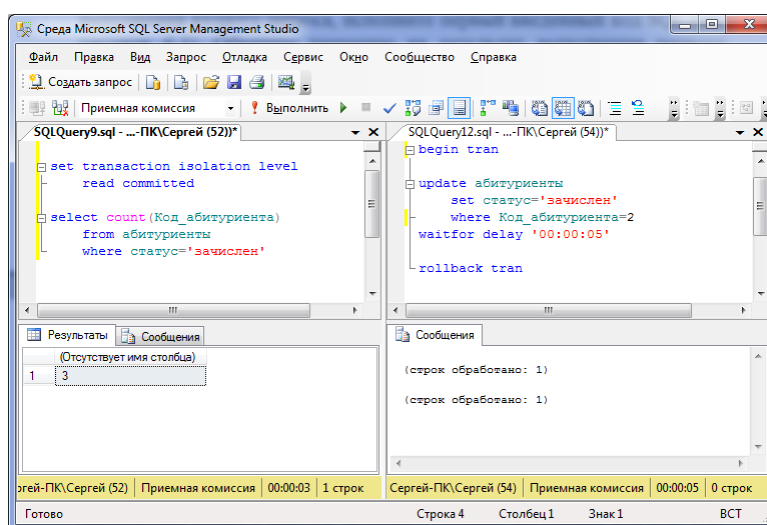


Рисунок 6.4 — Пример грязного чтения с измененным уровнем изоляции

Пример неповторяемого чтения.

- 1) Откройте SQL Server Management Studio и подключитесь к серверу баз данных.
- 2) Щелкните на панели инструментов кнопку "Создать запрос" 2 раза.
- 3) В первом открывшемся окне введите следующий код на языке SQL, вначале рассчитывающий количество зачисленных абитуриентов на специальность "АСОИУ", а затем выводящий на экран фамилии зачисленных на эту специальность абитуриентов:

```
begin tran
```

```
select count(код_абитуриента) as зачислено
from заявления з,
```

```
    специальности с
where з.код_специальности=с.код_специальности
    and с.название='АСОИУ'
    and з.статус='зачислен'

waitfor delay '00:00:05'

select фамилия
from заявления з,
    специальности с,
    абитуриенты а
where з.код_специальности=с.код_специальности
    and а.код_абитуриента]=з.код_абитуриента
    and с.название='АСОИУ'
    and з.статус='зачислен'

commit tran
```

- 4) Во втором открывшемся окне введите следующий код на языке SQL, меняющий статус заявления абитуриента "Иванов" (код абитуриента 1) на "не зачислен" для специальности "АСОИУ" (код специальности 1):

```
begin tran

update заявления
set статус='не зачислен'
where код_абитуриента=1
    and код_специальности=1

commit tran
```

В данной транзакции происходит отмена зачисления абитуриента "Иванов" (код абитуриента 1) на специальность "АСОИУ" (код специальности 1).

- 5) Исполните первый введенный код SQL и, не дожидаясь окончания выполнения первого запроса, исполните второй введенный код SQL (см. рисунок 6.5). Обратите внимание на результат выполнения первого запроса: первый оператор SELECT сообщает о том, что количество зачисленных абитуриентов равно 2, а второй оператор SELECT выводит в списке зачисленных только одну фамилию.

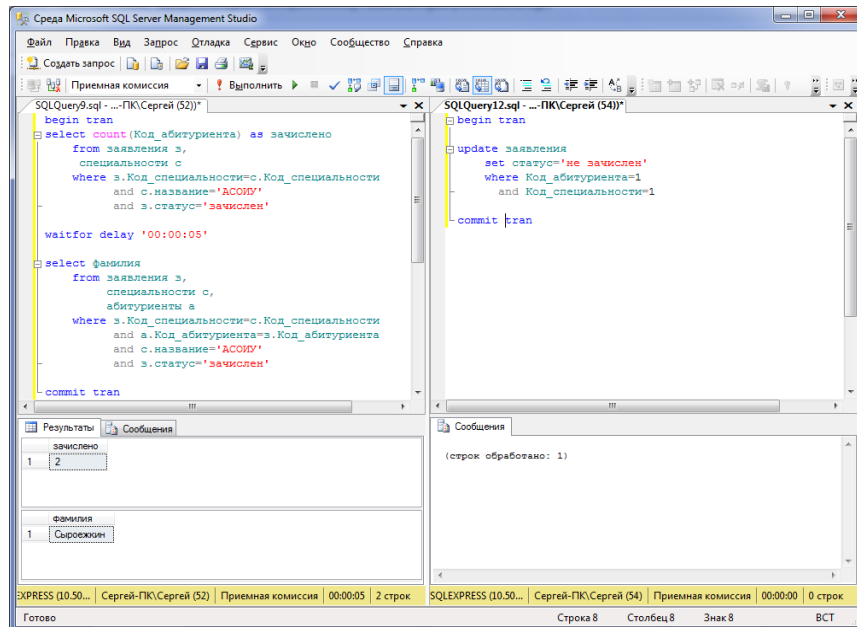


Рисунок 6.5 — Пример неповторяемого чтения

- 6) Откройте таблицу «Заявления» и измените статус абитуриента «Иванов» на «зачислен».
- 7) Измените уровень изоляции в первом запросе до повторяемого чтения (repeatable read):

```
begin tran
```

```
set transaction isolation level
```

```
repeatable read
```

```
select count(код_абитуриента) as зачислено
```

```
from заявления з,
```

```
специальности с
```

```
where з.код_специальности=с.код_специальности
```

```
and с.название='АСОИУ'
```



```
and з.статус='зачислен'

waitfor delay '00:00:05'

select фамилия
  from заявления з,
       специальности с,
       абитуриенты а
 where з.код_специальности=с.код_специальности
        and а.[код_абитуриента]=з.код_абитуриента
        and с.название='АСОИУ'
        and з.статус='зачислен'

commit tran
```

- 8) Исполните первый введенный код SQL и, не дожидаясь окончания выполнения первого запроса, исполните второй введенный код SQL (см. рисунок 6.6). Обратите внимание на результат выполнения первого запроса: первый оператор SELECT сообщает о том, что количество зачисленных абитуриентов равно 2, а второй оператор SELECT выводит список зачисленных из двух фамилий (изменение статуса абитуриента "Иванов" происходит только после завершения первой транзакции).

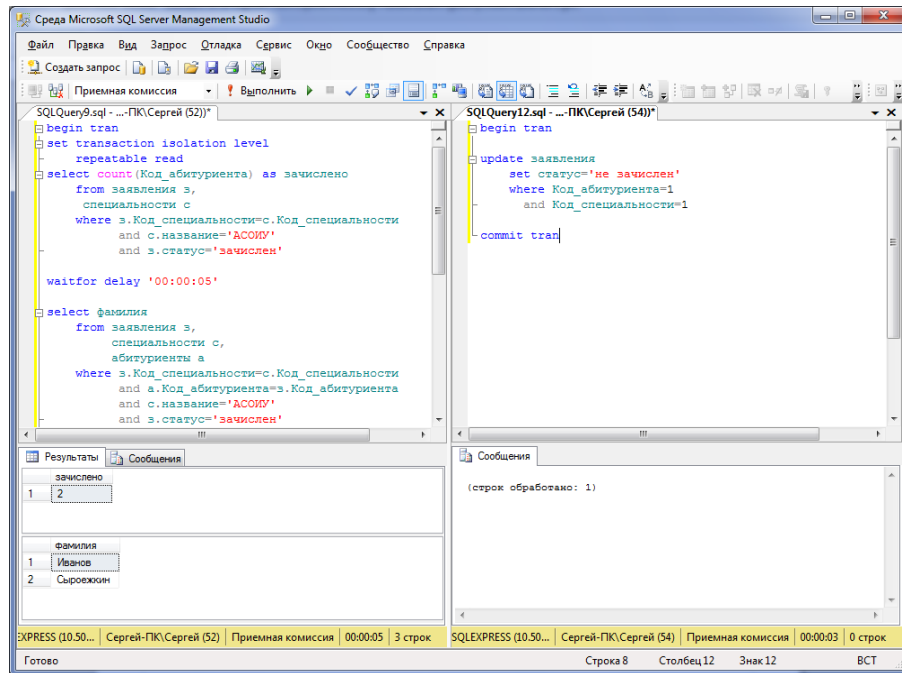


Рисунок 6.6 — Пример неповторяемого чтения с измененным уровнем изоляции

Пройдите **контрольный тест** по данной теме в системе тестирования Testing 2.1:

Разрешение экрана: 1024x768;

Название дисциплины — Разработка ИС;

Тема — Транзакции и блокировки.