

## Лабораторная работа. Транзакции.

### Цель работы

Используя данные базы данных, подготовленной в предыдущей лабораторной работе, подготовить и реализовать серию запросов, связанных с выборкой информации и модификацией данных таблиц.

### Задание 1

Используя базу, полученную в лабораторной 2, создать транзакцию, произвести ее откат и фиксацию. Показать, что данные существовали до отката, удалились после отката, снова были добавлены, и затем были успешно зафиксированы. При необходимости используйте точки сохранения и вложенные транзакции.

#### 1. Изучаем Patient (Пациент) и Call (Вызов врача)

The screenshot displays a database management interface. At the top, the structure of two tables is shown:

- Call** table:
  - id (primary key)
  - appeal\_date
  - start\_time
  - end\_time
  - visit\_status
  - patient\_id
  - therapist\_id
- Patient** table:
  - id (primary key)
  - full\_name
  - street
  - house\_number
  - district\_id

Below the table structures, a SQL query window contains the following queries:

```
1 SELECT COUNT(*) AS 'кол-во пациентов' FROM Patient;
2 SELECT COUNT(*) AS 'кол-во вызовов' FROM Call;
3 SELECT TOP 5 id, full_name, street FROM Patient ORDER BY id DESC;
4
```

The results of these queries are shown in a table below the query window:

кол-во пациентов	
1	29

кол-во вызовов	
1	19

	id	full_name	street
1	38	Тестовый Сотрудник	Улица Рабочая
2	37	Тестовый Сотрудник	Улица Рабочая
3	36	Тестовый Сотрудник	Улица Рабочая
4	27	Шипицын Денис Александрович	Улица Цветочная
5	26	Шебета Сергей Иванович	Улица Садовая

Начинаем транзакцию и делаем точку сохранения. Делаем запись с тестовым Пациентов и проверяем его существование в Patient.

```

6
7      -- Начало транзакции
8      BEGIN TRANSACTION;
9      -- Сохраняем точку сохранения
10     SAVE TRANSACTION BeforeInsert;
11     -- Добавляем нового пациента
12     INSERT INTO Patient (full_name, street, house_number, district_id)
13     VALUES (N'Тестовый Пациент Иванов', N'Улица Тестовая', 99, 1);
14     SELECT COUNT(*) FROM Patient
15     WHERE full_name = N'Тестовый Пациент Иванов';
16
17

```

100 % Проблемы не найдены.

Результаты Сообщения

(Отсутствует имя столбца)	
1	1

```

17
18     -- Добавляем вызов для этого пациента
19     DECLARE @new_patient_id INT = SCOPE_IDENTITY();
20     INSERT INTO Call (appeal_date, start_time, end_time, visit_status, patient_id, therapist_id)
21     VALUES ('2025-12-12', '10:00', '10:30', 0, @new_patient_id, 1);
22
23     SELECT TOP 1 * FROM Call ORDER BY id DESC;
24

```

100 % Проблемы не найдены. Стр: 22 Симв

Результаты Сообщения

	id	appeal_date	start_time	end_time	visit_status	patient_id	therapist_id
1	42	2025-12-12	10:00:00.0000000	10:30:00.0000000	0	45	1

Откатываем транзакцию. Количество пациентов и вызовов осталось таким же как до начала транзакций.

```

27
28     -- ОТКАТ транзакции
29     ROLLBACK TRANSACTION;
30     SELECT COUNT(*) AS patients_after_rollback FROM Patient
31     WHERE full_name = N'Тестовый Пациент Иванов';
32     SELECT COUNT(*) AS total_patients_after_rollback FROM Patient

```

100 % Проблемы не найдены.

Результаты Сообщения

	patients_after_rollback
1	0

	total_patients_after_rollback
1	29

## 2. Изучаем Patient (Пациент) и Call (Вызов врача)

```
1  BEGIN TRANSACTION;
2  SAVE TRANSACTION BeforeNewData;
3
4  INSERT INTO Patient (full_name, street, house_number, district_id)
5  VALUES (N'Тестовый пациент Петров', N'Улица Ленина', 123, 1);
6
7  DECLARE @committed_patient_id INT = SCOPE_IDENTITY();
8  INSERT INTO Call (appeal_date, start_time, end_time, visit_status, patient_id, therapist_id)
9  VALUES ('2025-12-12', '11:00', '11:45', 1, @committed_patient_id, 2);
```

100 % Проблемы не найдены. Стр: 9 Симв: 7

Сообщения

(Затронута 1 строка)

(Затронута 1 строка)

Время выполнения: 2025-12-13T22:45:38.8523048+03:00

```
DECLARE @new_call_id INT = SCOPE_IDENTITY();
INSERT INTO Call_diagnosis (diagnosis_id, call_id, doctor_comment)
VALUES (1, @new_call_id, N'Транзакция успешно зафиксирована');
```

100 % Проблемы не найдены.

Сообщения

(Затронута 1 строка)

Показ данных перед коммитом.

```
15 SELECT p.full_name, p.street, c.appeal_date, c.start_time
16 FROM Patient p
17 LEFT JOIN Call c ON p.id = c.patient_id
18 WHERE p.id = @committed_patient_id;
19
```

100 % Проблемы не найдены.

Результаты Сообщения

	full_name	street	appeal_date	start_time
1	Тестовый пациент Петров	Улица Ленина	2025-12-12	11:00:00.0000000

Проверяем данные после коммита. “Тестовый пациент Петров” был записан в Patient.

```
20 COMMIT TRANSACTION;
21 SELECT COUNT(*) FROM Patient
22 WHERE full_name = N'Тестовый пациент Петров';
23
```

100 % Проблемы не найдены.

Результаты Сообщения

	(Отсутствует имя столбца)
1	1

1	SELECT TOP (1000) [id]
2	,[full_name]
3	,[street]
4	,[house_number]
5	,[district_id]
6	FROM [lab_3_test2].[dbo].[Patient] ORDER BY id D
7	

100 % Проблемы не найдены.

Результаты Сообщения

	id	full_name	street	house_number	district_id
1	48	Тестовый пациент Петров	Улица Ленина	123	1
2	38	Тестовый Сотрудник	Улица Рабочая	88	1
3	37	Тестовый Сотрудник	Улица Рабочая	88	1
4	36	Тестовый Сотрудник	Улица Рабочая	88	1
5	27	Шипицын Денис Александрович	Улица Цветочная	16	4

3. Пример с вложенными транзакциями и с несколькими сохранениями.

Результаты Сообщения

	(Отсутствует имя столбца)
1	0

```
SELECT COUNT(*) FROM Patient
WHERE full_name = N'Тестовый Пациент Сидоров';
```

Создаем внешнюю транзакцию.

4	
5	BEGIN TRANSACTION OuterTX;
6	PRINT 'Начало внешней транзакции, уровень: ' + CAST(@@TRANCOUNT AS NVARCHAR);
7	

0 % Проблемы не найдены.

Сообщения

Начало внешней транзакции, уровень: 1

Время выполнения: 2025-12-13T23:07:27.8507032+03:00

Создаем первое сохранение и добавляем тестового пациента.

6	PRINT 'Начало внешней транзакции, уровень: ' + CAST(@@TRANCOUNT AS NVARCHAR);
7	
8	SAVE TRANSACTION SP1;
9	INSERT INTO Patient (full_name, street, house_number, district_id)
10	VALUES (N'Тестовый Пациент Сидоров', N'Улица Тестовая', 111, 1);

% Проблемы не найдены.

Сообщения

(Затронута 1 строка)

Время выполнения: 2025-12-13T23:08:20.3691861+03:00

Создаем внутреннюю транзакцию.

```

11
12 DECLARE @PatientID INT = SCOPE_IDENTITY();
13
14 BEGIN TRANSACTION InnerTX;
15 PRINT 'Начало внутренней транзакции, уровень: ' + CAST(@@TRANCOUNT AS NVARCHAR);
16

```

Сообщения

Начало внутренней транзакции, уровень: 2

Создаем второе сохранение и добавляем запись в Call - вызов для нового пациента.

```

15
16 DECLARE @PatientID INT = SCOPE_IDENTITY();
17 SAVE TRANSACTION SP2;
18 INSERT INTO Call (appeal_date, start_time, end_time, visit_status, patient_id, therapist_id)
19 VALUES (GETDATE(), '09:00', '09:30', 1, @PatientID, 1);
20
21

```

Сообщения

(Затронута 1 строка)

Откатываем внутреннюю транзакцию. Последний пример не записывается.

```

21
22 SELECT TOP 1 * FROM Call ORDER BY id DESC
23 ROLLBACK TRANSACTION SP2;
24 SELECT TOP 1 * FROM Call ORDER BY id DESC
25

```

100 % 2 0 Стр: 24 Симв: 44 Табуляц

Результаты

	id	appeal_date	start_time	end_time	visit_status	patient_id	therapist_id
1	48	2025-12-13	09:00:00.0000000	09:30:00.0000000	1	50	1

	id	appeal_date	start_time	end_time	visit_status	patient_id	therapist_id
1	46	2025-12-12	11:00:00.0000000	11:45:00.0000000	1	48	2

```

26 COMMIT TRANSACTION InnerTX;
27 PRINT 'Внутренняя транзакция зафиксирована, уровень: ' + CAST(@@TRANCOUNT AS NVARCHAR);
28

```

Сообщения

Внутренняя транзакция зафиксирована, уровень: 1

Время выполнения: 2025-12-13T23:20:00.3500148+03:00

Пациент должен быть сохранен.

28	
29	DECLARE @PatientID INT = SCOPE_IDENTITY();
30	SELECT COUNT(*) FROM Call WHERE patient_id = @PatientID;
31	
00 % 2 0 ↑ ↓ ◀ ▶ Стр: 31 Симв: 1 Табуляция	
Результаты Сообщения	
	(Отсутствует имя столбца)
1	1

Откатываем внешнюю транзакцию, тестового пациента не должно быть.

32	ROLLBACK TRANSACTION;
33	SELECT COUNT(*) AS patients_count FROM Patient
34	WHERE full_name = N'Тестовый Пациент Сидоров';
100 % 2 0 ↑ ↓ ◀ ▶	
Результаты Сообщения	
	patients_count
1	0

## Задание 2

Подготовить SQL-скрипты для выполнения проверок изолированности транзакций. Ваши скрипты должны работать с одной из таблиц, созданных в лабораторной работе №2.

### *Выполнение работы*

1. Запустить клиента и соединиться с базой данных. Открыть второе окно для ввода текста запросов (Ctrl+N в первом окне).
2. Установить в обоих сеансах уровень изоляции READ UNCOMMITTED. Выполнить сценарии проверки:
  - потерянных изменений,
  - грязного чтения.
3. Записать протокол выполнения сценариев.
4. Установить в обоих сеансах уровень изоляции READ COMMITTED. Выполнить сценарии проверки:
  - грязного чтения.
  - неповторяющегося чтения.
5. Записать протокол выполнения сценариев.
6. Установить в обоих сеансах уровень изоляции REPEATABLE READ. Выполнить сценарии проверки:
  - неповторяющегося чтения,
  - фантома.
7. Записать протокол выполнения сценариев.
8. Установить в обоих сеансах уровень изоляции SERIALIZABLE. Выполнить сценария проверки
  - фантома.
9. Записать протокол выполнения сценария.

## Уровень изоляции READ UNCOMMITTED

### Потерянные изменения

SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED; - установка уровня изоляции.

#### Сеанс 1

```
1  -- Устанавливаем уровень изоляции
2  SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED;
3  BEGIN TRANSACTION;
4
5  -- Читаем данные
6  SELECT * FROM Patient WHERE id = 2;
7
8  WAITFOR DELAY '00:00:10';
9
10 -- Изменяем данные (не фиксируем)
11 UPDATE Patient
12 SET street = N'Улица Измененная 1'
13 WHERE id = 2;
14
15 -- Даем время сессии 2
16 WAITFOR DELAY '00:00:10';
17
18 -- Откатываем
19 ROLLBACK;
```

100 % Проблемы не найдены.

Результаты Сообщения

	id	full_name	street	house_number	district_id
1	2	Андрианов Эдуард Александрович	Улица Мира	25	1

#### Сеанс 2



```

1  SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED;
2  BEGIN TRANSACTION;
3
4  WAITFOR DELAY '00:00:05';
5
6  -- Пытаемся изменить те же данные
7  UPDATE Patient
8  SET street = N'Улица Измененная 2'
9  WHERE id = 2;
10
11 COMMIT;
12
13 -- Проверяем результат
14 SELECT * FROM Patient WHERE id = 2;

```

100 % Проблемы не найдены.

Результаты Сообщения

	id	full_name	street	house_number	district_id
1	2	Анрианов Эдуард Александрович	Улица Измененная 2	25	1

100 % Проблемы не найдены.

Результаты Сообщения

	id	full_name	street	house_number	district_id
1	2	Анрианов Эдуард Александрович	Улица Измененная 2	25	1
2	3	Быков Александр Владимирович	Улица Пушкина	15	2
3	4	Ворообев Юлий Павлович	Улица Чацкого	8	2

Изменения первой транзакции потеряны, вторая транзакция перезаписала данные до фиксации первой.

Грязное чтение:


```
1  SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED;
2  BEGIN TRANSACTION;
3
4  -- Изменяем данные
5  UPDATE Patient
6  SET street = N'Улица Грязная'
7  WHERE id = 3;
8
9  WAITFOR DELAY '00:00:10';
10
11 -- Откатываем
12 ROLLBACK;
13
14 SELECT * FROM Patient WHERE id = 3
```


100 % Проблемы не найдены.


Результаты Сообщения

	id	full_name	street	house_number	district_id
1	3	Быков Александр Владимирович	Улица Пушкина	15	2

```
1  SET TRANSACTION ISOLATION LEVEL READ UNCOMMIT
2  BEGIN TRANSACTION;
3
4  WAITFOR DELAY '00:00:05';
5
6  -- Читаем незафиксированные данные
7  SELECT * FROM Patient WHERE id = 3;
8
9  COMMIT;
```

100 %  Проблемы не найдены.

 Результаты

 Сообщения

	id	full_name	street	house_number	district_id
1	3	Быков Александр Владимирович	Улица Грязная	15	2

Грязное чтение предотвращено, транзакция читает только зафиксированные данные.

## Уровень изоляции READ COMMITTED

### Грязное чтение

```
1  SET TRANSACTION ISOLATION LEVEL READ COMMITTED;
2  BEGIN TRANSACTION;
3
4  -- Изменяем данные
5  UPDATE Patient
6  SET street = N'Улица Грязная'
7  WHERE id = 3;
8
9  WAITFOR DELAY '00:00:10';
10
11 -- Откатываем
12 ROLLBACK;
13
14 SELECT * FROM Patient WHERE id = 3
```

100 % Проблемы не найдены.


Результаты Сообщения



	id	full_name	street	house_number	district_id
1	3	Быков Александр Владимирович	Улица Пушкина	15	2

```

1  SET TRANSACTION ISOLATION LEVEL READ COMMITTED;
2  BEGIN TRANSACTION;
3
4  WAITFOR DELAY '00:00:05';
5
6  -- Читаем незафиксированные данные
7  SELECT * FROM Patient WHERE id = 3;
8
9  COMMIT;

```

100 %  Проблемы не найдены.

 Результаты  Сообщения

	id	full_name	street	house_number	district_id
1	3	Быков Александр Владимирович	Улица Пушкина	15	2

Решена проблема грязного чтения, так как UNCOMMITTED заменен на COMMITTED.  
Грязное чтение предотвращено

## Неповторяющееся чтение

```
1  SET TRANSACTION ISOLATION LEVEL READ COMMITTED;
2  BEGIN TRANSACTION;
3
4  -- Первое чтение
5  SELECT * FROM Patient WHERE id = 5;
6
7  WAITFOR DELAY '00:00:10';
8
9  -- Второе чтение (данные уже изменены сессией 2)
10 SELECT * FROM Patient WHERE id = 5;
11
12 COMMIT;
```

100 % Проблемы не найдены.

Результаты Сообщения

	id	full_name	street	house_number	district_id
1	5	Воробьева Мария Павловна	Улица Садовая	12	4

	id	full_name	street	house_number	district_id
1	5	Воробьева Мария Павловна	Улица Измененная при чтении	12	4

```
1  SET TRANSACTION ISOLATION LEVEL READ COMMITTED;
2  BEGIN TRANSACTION;
3
4  WAITFOR DELAY '00:00:05';
5
6  -- Изменяем данные
7  UPDATE Patient
8  SET street = N'Улица Измененная при чтении'
9  WHERE id = 5;
10
11 COMMIT;
```

Второе чтение показало измененные данные. Между двумя чтениями вторая транзакция изменила данные

## Уровень изоляции REPEATABLE READ

### Неповторяющееся чтение

```
1  SET TRANSACTION ISOLATION LEVEL REPEATABLE READ;
2  BEGIN TRANSACTION;
3
4  -- Первое чтение
5  SELECT * FROM Patient WHERE id = 5;
6
7  WAITFOR DELAY '00:00:10';
8
9  -- Второе чтение (данные уже изменены сессией 2)
10 SELECT * FROM Patient WHERE id = 5;
11
12 COMMIT;
```

100 % Проблемы не найдены.

Результаты Сообщения

	id	full_name	street	house_number	district_id
1	5	Воробьева Мария Павловна	Улица Измененная при чтении	12	4

	id	full_name	street	house_number	district_id
1	5	Воробьева Мария Павловна	Улица Измененная при чтении	12	4

```
1  SET TRANSACTION ISOLATION LEVEL READ COMMITTED;  
2  BEGIN TRANSACTION;  
3  
4  WAITFOR DELAY '00:00:05';  
5  
6  -- Изменяем данные  
7  UPDATE Patient  
8  SET street = N'Улица Измененная при чтении'  
9  WHERE id = 5;  
10  
11 COMMIT;
```

Неповторяющееся чтение предотвращено, блокировка строк не позволяет изменять прочитанные данные.



## Фантомное чтение

```

1  SET TRANSACTION ISOLATION LEVEL REPEATABLE READ;
2  BEGIN TRANSACTION;
3
4  -- Первое чтение диапазона
5  SELECT * FROM Patient
6  WHERE district_id = 1;
7
8  WAITFOR DELAY '00:00:10';
9
10 -- Второе чтение (появятся новые строки)
11 SELECT * FROM Patient
12 WHERE district_id = 1;
13
14 COMMIT;

```

100 % Проблемы не найдены.

Результаты

Сообщения

1	11	Лопаткин Даниил Евгеньевич	Улица Гагарина	9	1
3	12	Мерзун Иван Романович	Улица Ленина	31	1
4	13	Минаева Ксения Аркадьевна	Улица Мира	17	1
5	22	Середа Арсентий Станиславович	Улица Ленина	42	1
6	23	Тюменева Екатерина Андреевна	Улица Мира	29	1

	id	full_name	street	house_number	district_id
1	2	Андрианов Эдуард Александрович	Улица Измененная 2	25	1
2	11	Лопаткин Даниил Евгеньевич	Улица Гагарина	9	1
3	12	Мерзун Иван Романович	Улица Ленина	31	1
4	13	Минаева Ксения Аркадьевна	Улица Мира	17	1
5	22	Середа Арсентий Станиславович	Улица Ленина	42	1
6	23	Тюменева Екатерина Андреевна	Улица Мира	29	1
7	36	Тестовый Сотрудник	Улица Рабочая	88	1
8	37	Тестовый Сотрудник	Улица Рабочая	88	1
9	38	Тестовый Сотрудник	Улица Рабочая	88	1
10	48	Тестовый пациент Петров	Улица Ленина	123	1
11	1...	Новый Пациент	Улица Фантомная	99	1

```
1  SET TRANSACTION ISOLATION LEVEL REPEATABLE READ;
2  BEGIN TRANSACTION;
3
4  WAITFOR DELAY '00:00:05';
5
6  -- Вставляем новую строку
7  INSERT INTO Patient (full_name, street, house_number, district_id)
8  VALUES (N'Новый Пациент', N'Улица Фантомная', 99, 1);
9
10 COMMIT;
```

Появились новые строки при повторном чтении. Уровень Не блокирует вставку новых строк в диапазон.

Уровень изоляции SERIALIZABLE, проверка фантома.

```
1  SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;
2  BEGIN TRANSACTION;
3
4  SELECT * FROM Patient
5  WHERE district_id = 2;
6
7  WAITFOR DELAY '00:00:10';
8
9  SELECT * FROM Patient
10 WHERE district_id = 2;
11
12 COMMIT;
```

100 % Проблемы не найдены.

Результаты Сообщения

	id	full_name	street	house_number	district_id
1	3	Быков Александр Владимирович	Улица Пушкина	15	2
2	4	Воробев Юрий Палович	Улица Чехова	8	2
3	14	Немтинов Никита Сергеевич	Улица Пушкина	4	2
4	15	Петроченков Михаил Денисович	Улица Чехова	11	2
5	24	Хрулев Иван Сергеевич	Улица Пушкина	6	2
6	25	Чернышова Мария Ивановна	Улица Чехова	21	2
7	1049	Еще один Пациент	Улица Сериализуемая	100	2
8	1050	Еще один Пациент 2	Улица Сериализуемая	100	2

	id	full_name	street	house_number	district_id
1	3	Быков Александр Владимирович	Улица Пушкина	15	2
2	4	Воробев Юрий Палович	Улица Чехова	8	2
3	14	Немтинов Никита Сергеевич	Улица Пушкина	4	2
4	15	Петроченков Михаил Денисович	Улица Чехова	11	2
5	24	Хрулев Иван Сергеевич	Улица Пушкина	6	2
6	25	Чернышова Мария Ивановна	Улица Чехова	21	2
7	1...	Еще один Пациент	Улица Сериа...	100	2
8	1...	Еще один Пациент 2	Улица Сериа...	100	2

```
1  SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;
2  BEGIN TRANSACTION;
3
4
5  -- Эта вставка будет заблокирована
6  INSERT INTO Patient (full_name, street, house_number, district_id)
7  VALUES (N'Еще один Пациент 3', N'Улица Сериализуемая', 100, 2);
8
9  COMMIT;
```

SERIALIZABLE защитил нас от фантомного чтения, блокировка диапазона не позволяет вставлять новые строки.