

**Министерство науки и высшего образования Российской Федерации  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«Национальный исследовательский университет ИТМО»  
(Университет ИТМО)**

**Факультет  
Образовательная программа**

**Инфокоммуникационных технологий  
11.03.02 Программирование в  
инфокоммуникационных системах**

**ОТЧЕТ  
по лабораторной работе 8  
по дисциплине «Разработка баз данных»**

**Выполнил: студент группы К33202  
Рогозина Вероника Сергеевна  
Проверил: ст. преподаватель Осетрова И.С.**

**Санкт-Петербург  
2024**

## 1. Цель работы

Целью данной лабораторной работы является управление транзакциями и блокировками.

## 2. Задачи, решаемые при выполнении работы

- 2.1. Определение транзакции
- 2.2. Поиск и обнаружение блокирования
- 2.3. Уровни изоляции

## 3. Объект исследования

Объектом исследования данной лабораторной работы являются способы управления транзакциями и блокировками.

## 4. Исходные данные

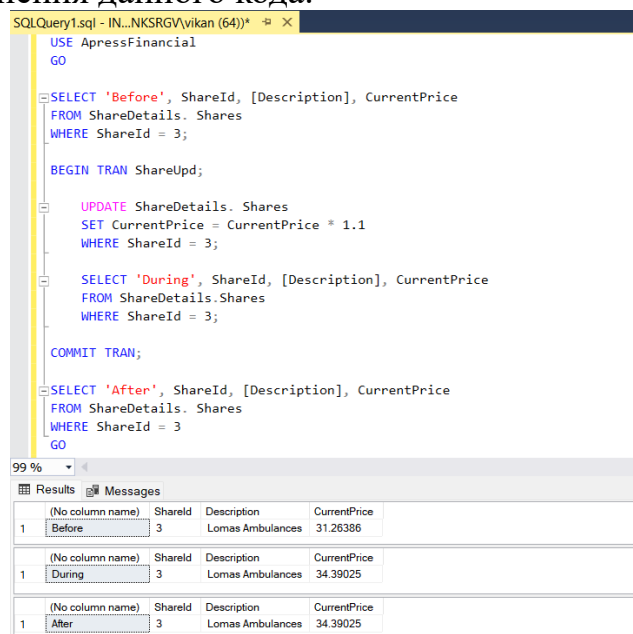
Инструкция к лабораторной работе, инструмент Microsoft SQL Server Management Studio 2019, база данных, таблицы, индексы, представления, процедуры, функции, триггеры, созданные в предыдущих лабораторных работах, файл для создания тестовой базы данных из приложения к лабораторной работе.

## 5. Выполнение работы

### 5.1. Определение транзакции.

В ходе выполнения данной задачи необходимо ознакомиться с кодом для создания транзакций, а также с инструкциями BEGIN TRAN, COMMIT TRAN и ROLLBACK TRAN.

5.1.1. На рисунке 1 представлен код запроса для создания транзакции, а также результат выполнения данного кода.



```
USE ApressFinancial
GO

SELECT 'Before', ShareId, [Description], CurrentPrice
FROM ShareDetails.Shares
WHERE ShareId = 3;

BEGIN TRAN ShareUpd;

UPDATE ShareDetails.Shares
SET CurrentPrice = CurrentPrice * 1.1
WHERE ShareId = 3;

SELECT 'During', ShareId, [Description], CurrentPrice
FROM ShareDetails.Shares
WHERE ShareId = 3;

COMMIT TRAN;

SELECT 'After', ShareId, [Description], CurrentPrice
FROM ShareDetails.Shares
WHERE ShareId = 3
GO
```

	(No column name)	ShareId	Description	CurrentPrice
1	Before	3	Lomas Ambulances	31.26386

	(No column name)	ShareId	Description	CurrentPrice
1	During	3	Lomas Ambulances	34.39025

	(No column name)	ShareId	Description	CurrentPrice
1	After	3	Lomas Ambulances	34.39025

Рисунок 1 – Код запроса для создания транзакции

5.1.2. Код запроса, использующего инструкцию ROLLBACK TRAN и результат его выполнения представлены на рисунке 2.

```

SQLQuery2.sql - IN...NKS...vikan (54))*
SELECT 'Before', ShareId, [Description], CurrentPrice
FROM ShareDetails. Shares
WHERE ShareId <= 3

BEGIN TRAN ShareUpd;
UPDATE ShareDetails.Shares
SET CurrentPrice = CurrentPrice * 1.1
WHERE ShareId <= 3;

SELECT 'Within the transaction', ShareId, [Description], CurrentPrice
FROM ShareDetails. Shares
WHERE ShareId <= 3;

ROLLBACK TRAN;

SELECT 'After', ShareId, [Description], CurrentPrice
FROM ShareDetails. Shares
WHERE ShareId <= 3
GO
  
```

	(No column name)	ShareId	Description	CurrentPrice
1	Before	1	ACME'S HOMEBAKE COOKIES INC	2.34125
2	Before	2	ACME'S HOMEBAKE COOKIES INC	2.34125
3	Before	3	Lomas Ambulances	34.39025

	(No column name)	ShareId	Description	CurrentPrice
1	Within the transaction	1	ACME'S HOMEBAKE COOKIES INC	2.57538
2	Within the transaction	2	ACME'S HOMEBAKE COOKIES INC	2.57538
3	Within the transaction	3	Lomas Ambulances	37.82928

	(No column name)	ShareId	Description	CurrentPrice
1	After	1	ACME'S HOMEBAKE COOKIES INC	2.34125
2	After	2	ACME'S HOMEBAKE COOKIES INC	2.34125
3	After	3	Lomas Ambulances	34.39025

Рисунок 2 – Код запроса, использующего инструкцию ROLLBACK TRAN

5.1.3. Код запроса, отражающего работу вложенных транзакций представлен на рисунке 3.

```

SQLQuery3.sql - IN...NKS...vikan (58))*
BEGIN TRAN ShareUpd
SELECT '1st TranCount', @@TRANCOUNT
BEGIN TRAN ShareUpd2
SELECT '2nd TranCount', @@TRANCOUNT
COMMIT TRAN ShareUpd2
SELECT '3rd TranCount', @@TRANCOUNT
COMMIT TRAN --It is at this point that data modifications will be committed
SELECT 'Last TranCount', @@TRANCOUNT
GO
  
```

	(No column name)	(No column name)
1	1st TranCount	1

	(No column name)	(No column name)
1	2nd TranCount	2

	(No column name)	(No column name)
1	3rd TranCount	1

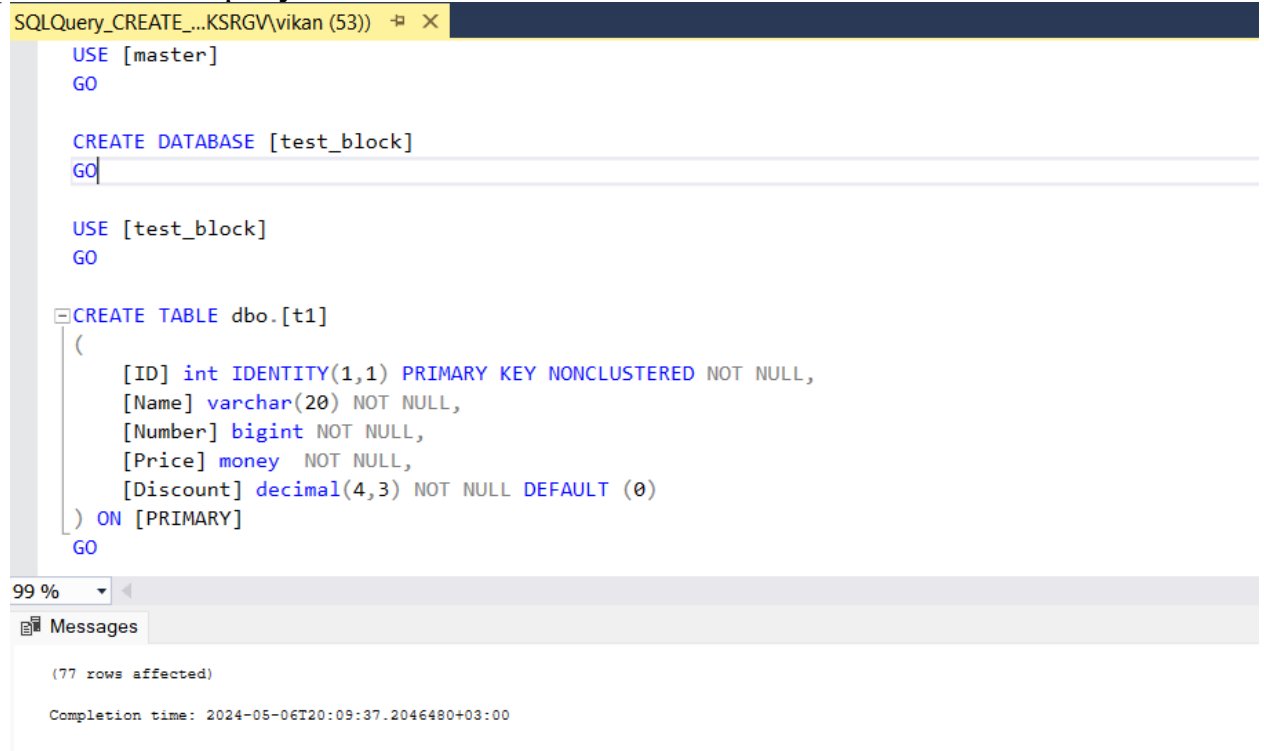
	(No column name)	(No column name)
1	Last TranCount	0

Рисунок 3 – Код запроса, отражающего работу вложенных транзакций

## 5.2. Поиск и обнаружение блокирования

В ходе выполнения данной задачи необходимо ознакомиться со способами поиска и обнаружения блокирования.

5.2.1. Код, необходимый для создания тестовой базы данных представлен на рисунке 4.



```
SQLQuery_CREATE_...KSRGV\vikan (53)  X
USE [master]
GO

CREATE DATABASE [test_block]
GO

USE [test_block]
GO

CREATE TABLE dbo.[t1]
(
    [ID] int IDENTITY(1,1) PRIMARY KEY NONCLUSTERED NOT NULL,
    [Name] varchar(20) NOT NULL,
    [Number] bigint NOT NULL,
    [Price] money NOT NULL,
    [Discount] decimal(4,3) NOT NULL DEFAULT (0)
) ON [PRIMARY]
GO
```

99 %

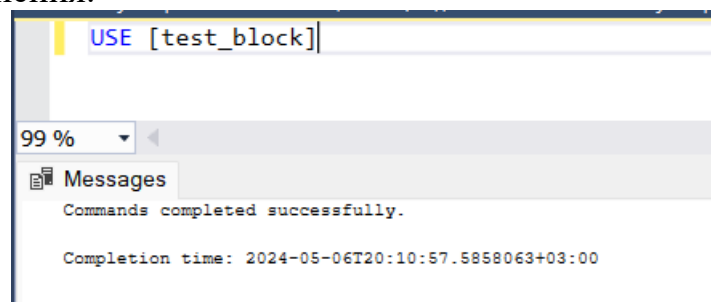
Messages

(77 rows affected)

Completion time: 2024-05-06T20:09:37.2046480+03:00

Рисунок 4 – Код для создания тестовой БД

5.2.2. Код, необходимый для подключения соединения 1 к тестовой базе данных представлен на рисунке 5. Аналогичным образом были подключены остальные соединения.



```
USE [test_block]
```

99 %

Messages

Commands completed successfully.

Completion time: 2024-05-06T20:10:57.5858063+03:00

Рисунок 5 – Код для подключения Соединения 1 к тестовой БД

5.2.3. Код запроса, выполняемый в окне первого соединения и необходимый для обновления строки в таблице тестовой базы данных представлен на рисунке 6.

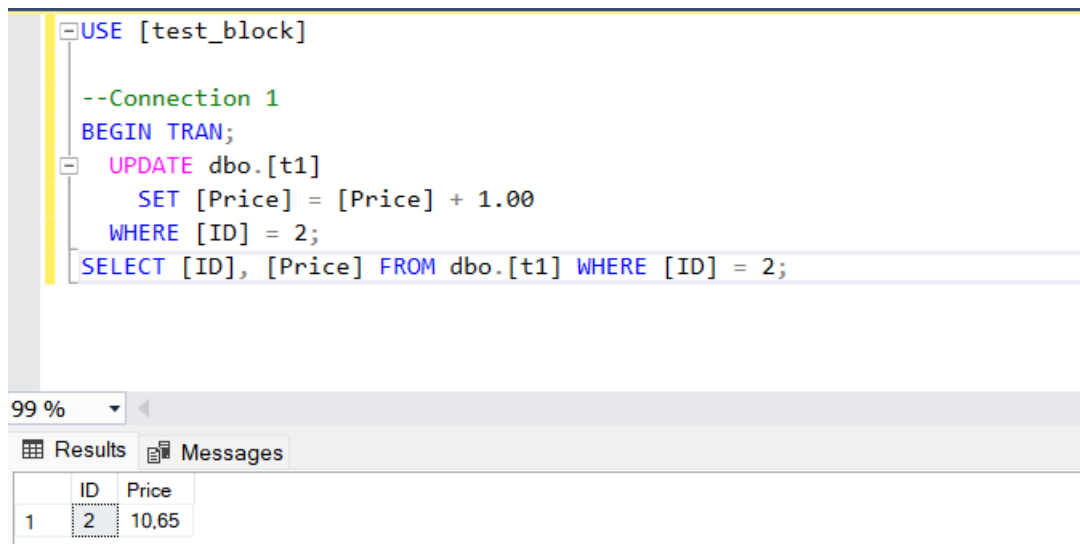


Рисунок 6 – Код запроса для обновления строки

5.2.4. Код запроса, выполняемого в окне второго соединения и необходимого для запроса той же строки из таблицы тестовой базы данных представлен на рисунке 7.

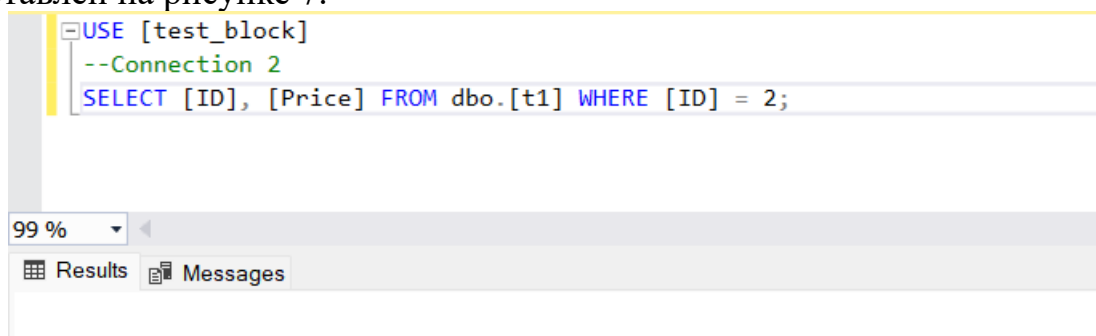


Рисунок 7 – Код запроса для получения строки

5.2.5. Код, выполняющий запросы к объектам динамического управления для обнаружения ситуации блокирования в окне третьего соединения, представлен на рисунке 8.

SQLQuery3.sql - IN...NKSRGV\vikan (79))* SQLQuery2.sql - INK...n (77)) Executing...*									
USE [test_block]									
-- Connection 3									
-- Lock info									
SELECT									
request_session_id AS spid,									
resource_type AS restype,									
resource_database_id AS dbid,									
DB_NAME(resource_database_id) AS dbname,									
resource_description AS res,									
resource_associated_entity_id AS resid,									
request_mode AS mode,									
request_status AS status									
FROM sys.dm_tran_locks;									
99 %									
Results Messages									
	spid	restype	dbid	dbname	res	resid	mode	status	
1	79	DATABASE	6	test_block		0	S	GRANT	
2	77	DATABASE	6	test_block		0	S	GRANT	
3	75	DATABASE	6	test_block		0	S	GRANT	
4	53	DATABASE	6	test_block		0	S	GRANT	
5	77	PAGE	6	test_block	1:240	72057594045726720	IS	GRANT	
6	75	PAGE	6	test_block	1:240	72057594045726720	IX	GRANT	
7	77	PAGE	6	test_block	1:256	72057594045792256	IS	GRANT	
8	75	RID	6	test_block	1:240:1	72057594045726720	X	GRANT	
9	77	RID	6	test_block	1:240:1	72057594045726720	S	WAIT	
10	77	KEY	6	test_block	(61a06abd401c)	72057594045792256	S	GRANT	
11	77	OBJECT	6	test_block		901578250	IS	GRANT	
12	75	OBJECT	6	test_block		901578250	IX	GRANT	

Рисунок 8 – Код запроса для обнаружения ситуации блокирования

5.2.6. Код запроса для получения сведений о подключениях, связанных с процессами, входящими в цепочку блокировок, представлен на рисунке 9.

SQLQuery3.sql - IN...NKSRGV\vikan (79))* SQLQuery2.sql - INK...n (77)) Executing...* SQLQuery1.sql - IN...NKSRGV\vikan (75)									
USE [test_block]									
-- Connection 3									
-- Connection info									
SELECT --use * to explore									
session_id AS spid,									
connect_time,									
last_read,									
last_write,									
most_recent_sql_handle									
FROM sys.dm_exec_connections									
WHERE session_id IN (75, 77);									
99 %									
Results Messages									
	spid	connect_time	last_read	last_write	most_recent_sql_handle				
1	75	2024-05-06 20:10:12.750	2024-05-06 20:14:58.407	2024-05-06 20:14:58.443	0x02000000ACDB8C0FFFC7EB49A92A8729BA60465E2661D1...				
2	77	2024-05-06 20:10:48.810	2024-05-06 20:16:38.377	2024-05-06 20:11:27.250	0x0200000086656431F4E25CA19EEB6E26B18F44350F6F3C9...				

Рисунок 9 – Код запроса для получения сведений о подключениях

5.2.7. Код запроса для получения результата, отображающего последний пакет или фрагмент программного кода, запускаемого каждым подключением, входящим в цепочку блокировок, представлен на рисунке 10.

```
USE [test_block]
-- Connection 3
-- SQL text
SELECT session_id, text
FROM sys.dm_exec_connections
CROSS APPLY sys.dm_exec_sql_text(most_recent_sql_handle) AS ST
WHERE session_id IN(75, 77);
```

99 %

Results

session_id	text
75	USE [test_block]

```
--Connection 1
BEGIN TRAN;
UPDATE dbo.[t1]
SET [Price] = [Price] + 1.00
WHERE [ID] = 2;
SELECT [ID], [Price] FROM dbo.[t1] WHERE [ID] = 2;
77      (@1 tinyint)SELECT [ID],[Price] FROM [dbo].[t1] WHERE [ID]=@1

(2 rows affected)

Completion time: 2024-05-06T20:33:15.2860275+03:00
```

Рисунок 10 – Код запроса для получения результата, отображающего последний пакет кода

5.2.8. Код запроса для получения активных процессов, включая заблокированные, представлен на рисунке 11.

```
SQLQuery3.sql - IN...NKSRGV\vikan (79))* SQLQuery2.sql - INK...n (77)) Executing...* SQLQuery1.sql - IN...NKSRGV\vikan (75))*
USE [test_block]
-- Connection 3
--Blocking
SELECT --use * to explore
session_id AS spid,
blocking_session_id,
command,
sql_handle,
database_id,
wait_type,
wait_time,
wait_resource
FROM sys.dm_exec_requests
WHERE blocking_session_id > 0;
```

99 %

Results Messages

	spid	blocking_session_id	command	sql_handle	database_id	wait_type	wait_time	wait_resource
1	77	75	SELECT	0x0200000086656431F4E25CA19EEB6E26B18F44350F6F3C...	6	LCK_M_S	1136305	RID: 6:1:240:1

Рисунок 11 – Код запроса для получения активных процессов

5.2.9. Запрос, выполнявшийся в пункте 5.2.4 останавливается (Рисунок 12). На рисунке 13 представлен код запроса, в котором время ожидания блокировки определено.

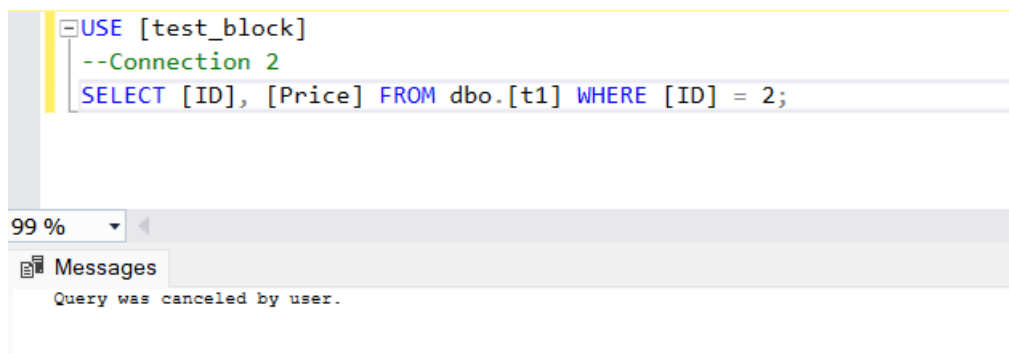


Рисунок 12 – Выход из выполнения запроса

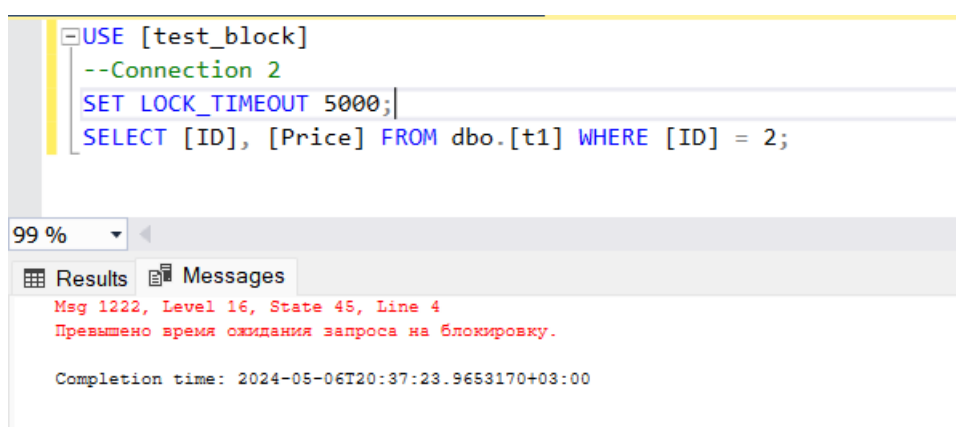


Рисунок 13 – Код запроса с определенным временем ожидания блокировки

5.2.10. На рисунке 14 представлен код запроса, в котором время ожидания блокировки установлено в значения по умолчанию.

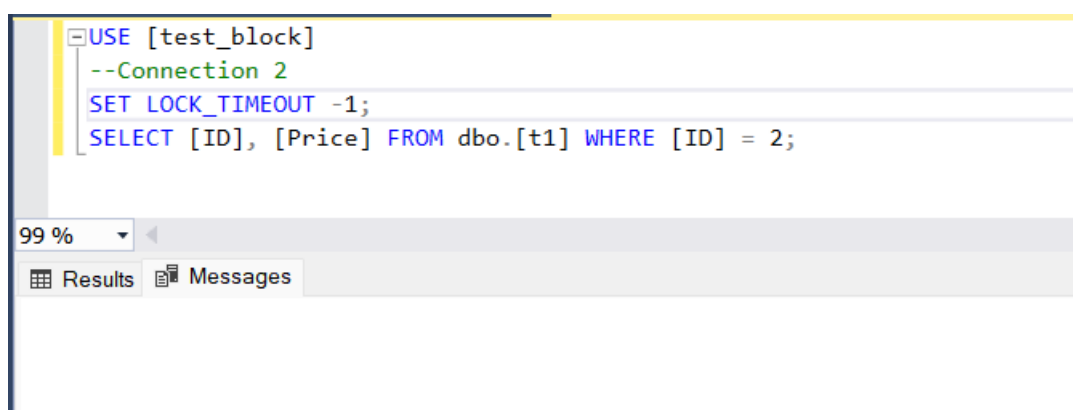


Рисунок 14 – Код запроса с временем ожидания блокировки по умолчанию



5.2.11. Код запроса для завершения транзакции представлен на рисунке 15.

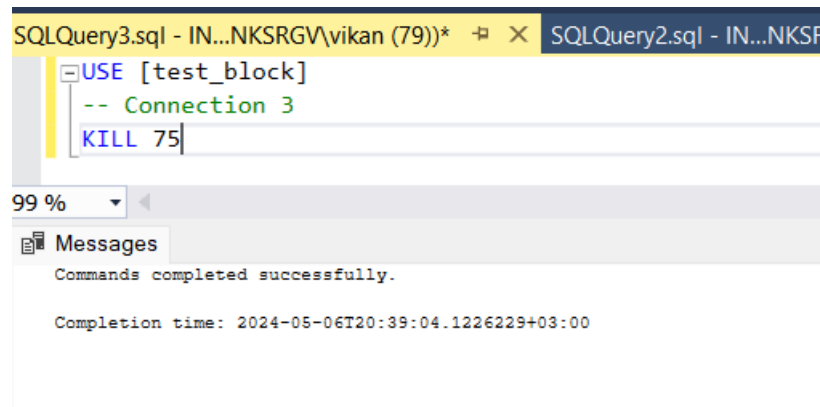


Рисунок 15 – Код запроса для завершения транзакции

5.2.12. На рисунке 16 представлен результат запроса после отката транзакции в сеансе первого соединения (изменение цены товара 2 отменяется, монопольная блокировка снимается).

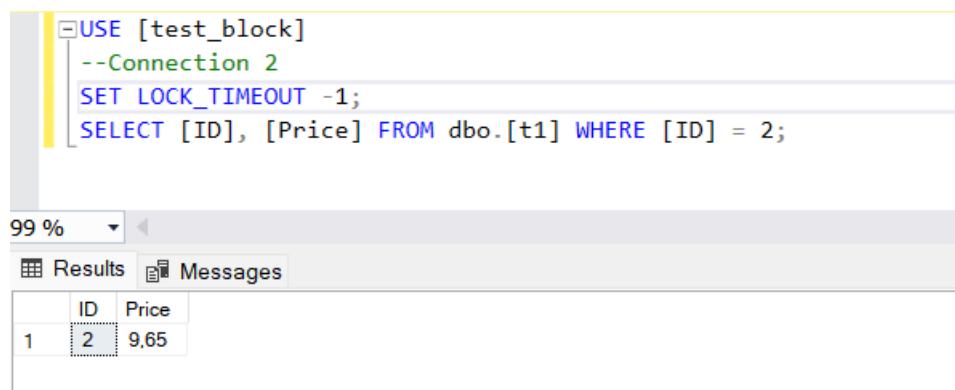


Рисунок 16 – Результат запроса после отката транзакции

### 5.3. Уровни изоляции

В ходе выполнения данной задачи необходимо ознакомиться с уровнями изоляции: READ UNCOMMITTED, READ COMMITTED, REPEATABLE READ, SERIALIZABLE.

#### Уровень изоляции READ UNCOMMITTED

5.3.1. Код запроса для подключения соединения к базе данных представлен на рисунке 17.

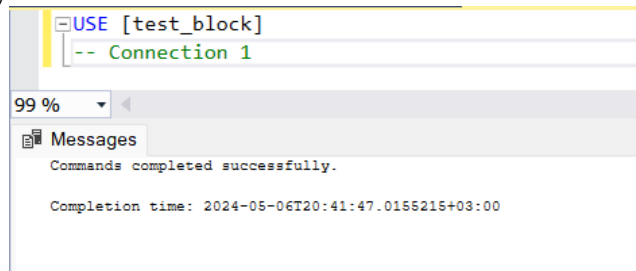


Рисунок 17 – Код запроса для подключения соединения к БД

5.3.2. Код запроса для запуска транзакции и обновления строки в таблице `dbo.[t1]` представлен на рисунке 18.

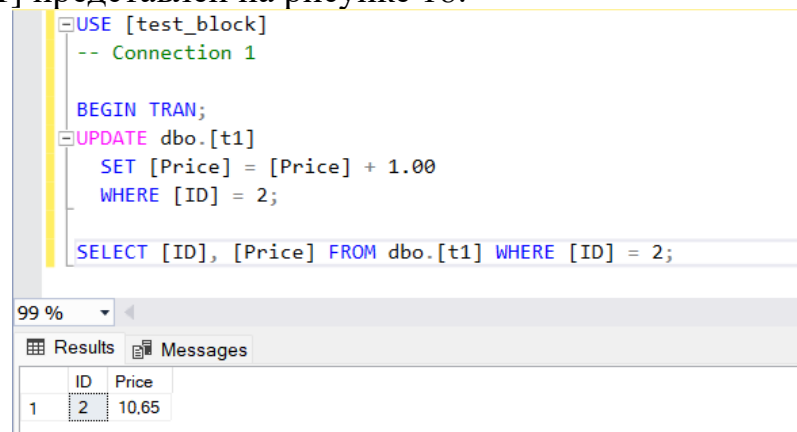


Рисунок 18 - Код запроса для запуска транзакции и обновления строки в таблице

5.3.3. Код запроса для установки уровня изоляции READ UNCOMMITTED в окне второго соединения представлен на рисунке 19.

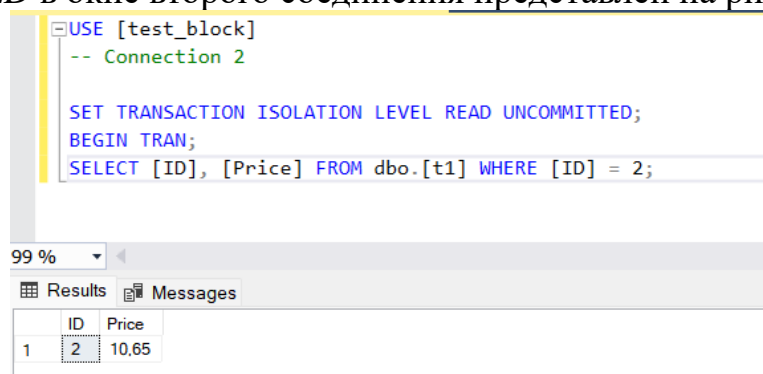


Рисунок 19 – Код запроса для установки уровня изоляции READ UNCOMMITTED

5.3.4. Код запроса для отката транзакции представлен на рисунке 20. После выполнения данного кода обновление данных было отменено.

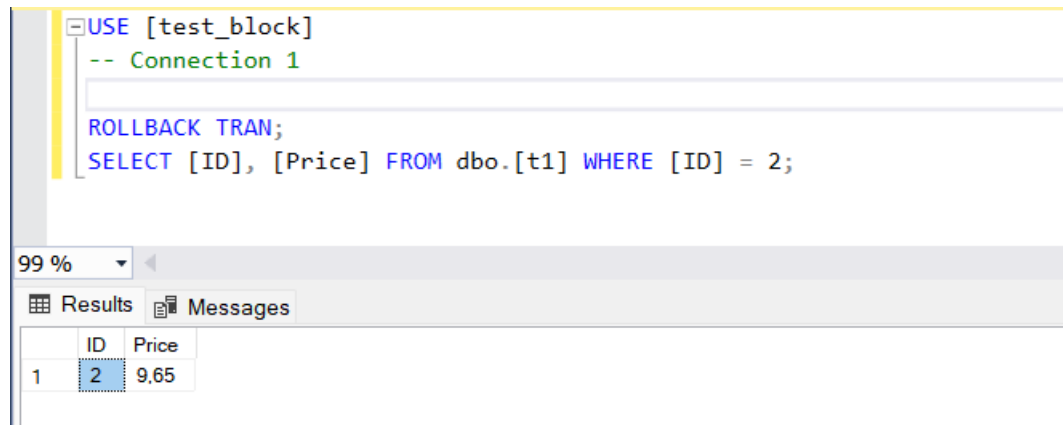


Рисунок 20 – Код запроса для отката транзакции

### Уровень изоляции READ COMMITTED

5.3.5. Перед началом выполнения данного пункта было выполнено подключение соединений к базе данных. На рисунке 21 представлен запуск транзакции.

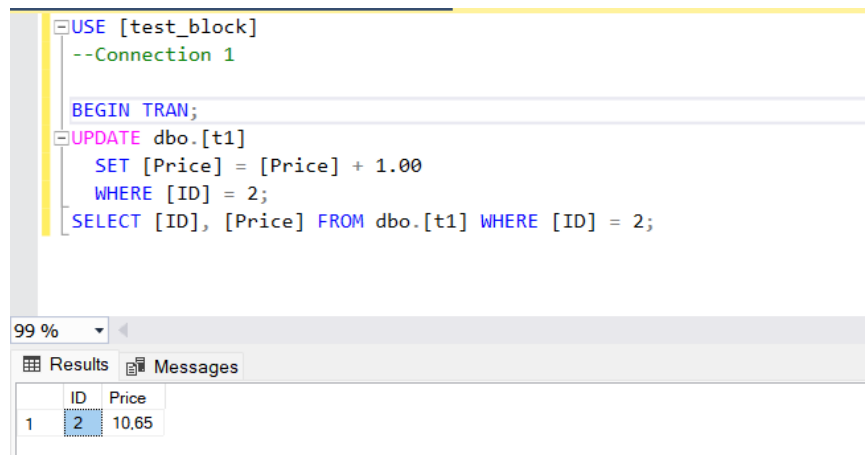


Рисунок 21 – Запуск транзакции

5.3.6. Установка уровня изоляции READ COMMITTED в окне второго соединения представлена на рисунке 22.

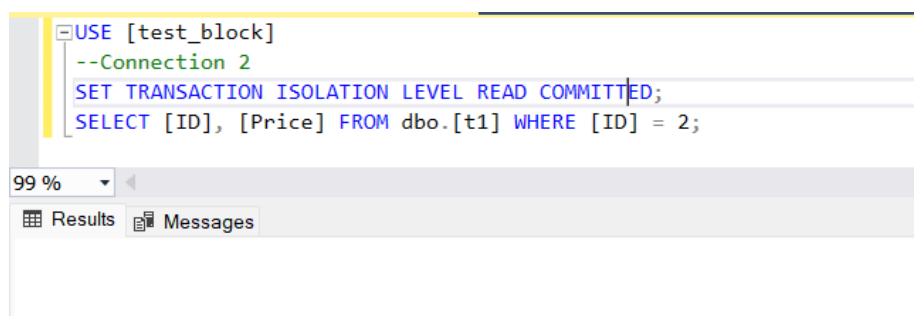


Рисунок 22 – Установка уровня изоляции READ COMMITTED

5.3.7. Код запроса для отката транзакции в окне первого соединения представлен на рисунке 23.

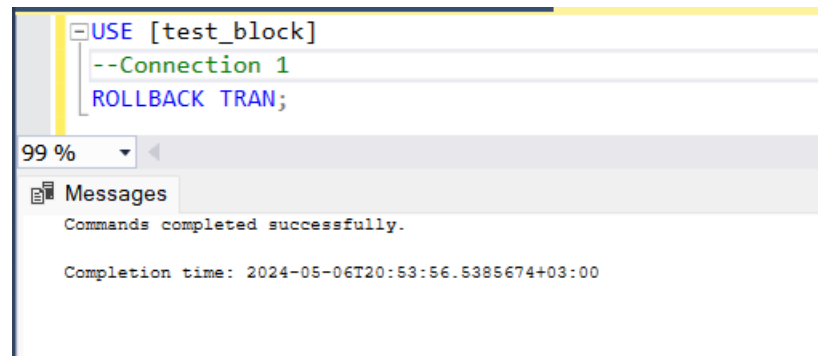


Рисунок 23 – Код запроса для отката транзакции

### Уровень изоляции REPEATABLE READ

5.3.8. Перед началом выполнения данного пункта было выполнено подключение соединений к базе данных. На рисунке 24 представлена установка уровня изоляции REPEATABLE READ в окне первого соединения.

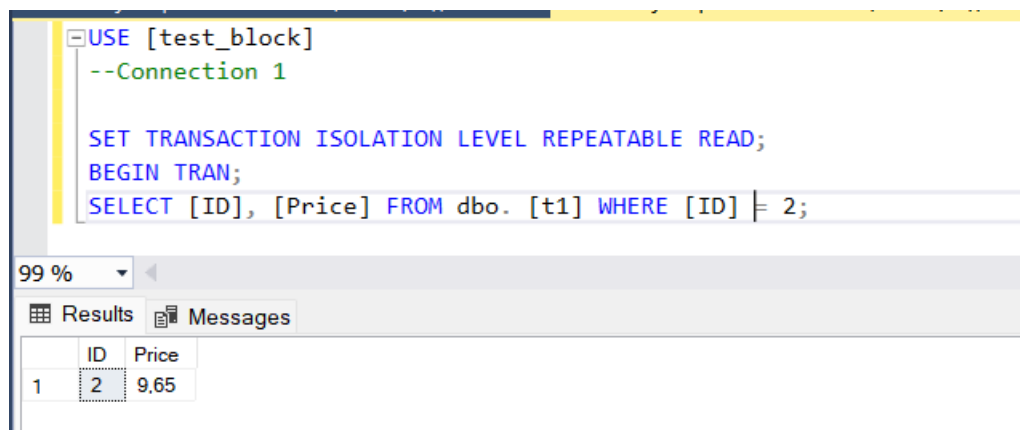


Рисунок 24 – Установка уровня изоляции REPEATABLE READ

5.3.9. Код запроса для изменения цены товара 2 в окне второго соединения представлен на рисунке 25. Данная попытка будет заблокирована.

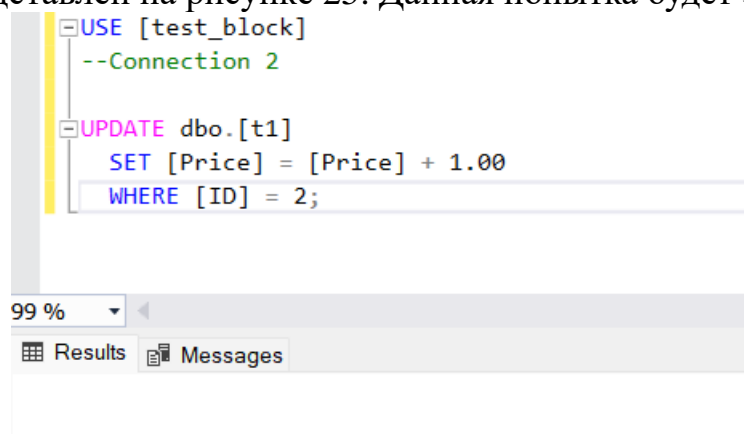


Рисунок 25 – Код запроса для изменения цены товара 2

5.3.10. Код запроса для повторного считывания данных и сохранения транзакции в окне первого соединения представлен на рисунке 26.

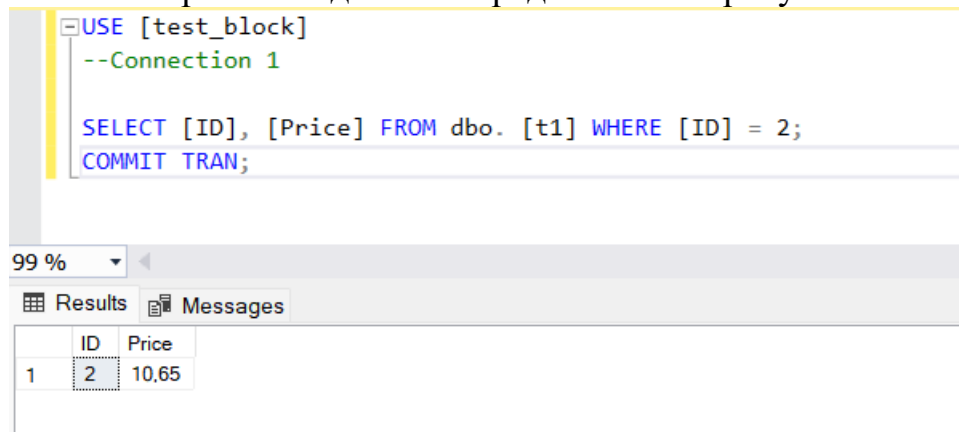


Рисунок 26 – Код запроса для повторного считывания данных и сохранения транзакции

5.3.11. Результат сохранения транзакции в окне второго соединения представлен на рисунке 27. Цена товара изменилась.

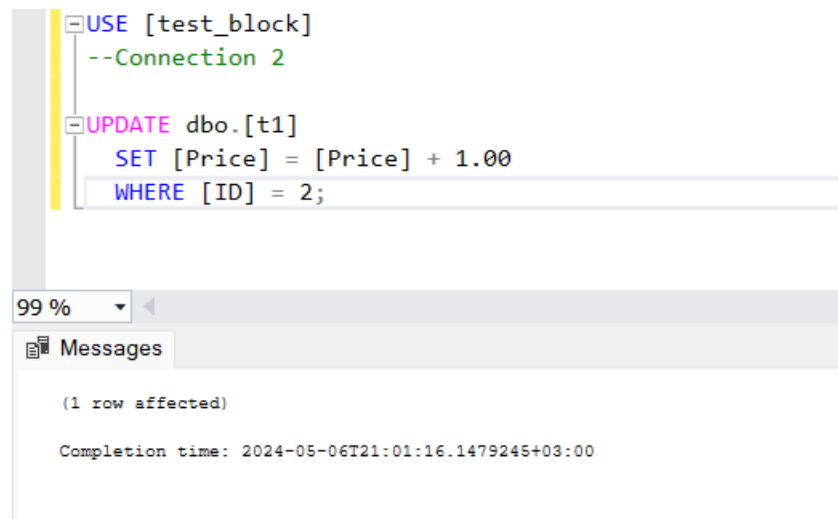


Рисунок 27 – Результат сохранения транзакции

### Уровень изоляции SERIALIZABLE

5.3.12. Перед началом выполнения данного пункта было выполнено подключение соединений к базе данных. На рисунке 28 представлена установка уровня изоляции SERIALIZABLE и запуск транзакции в окне первого соединения.

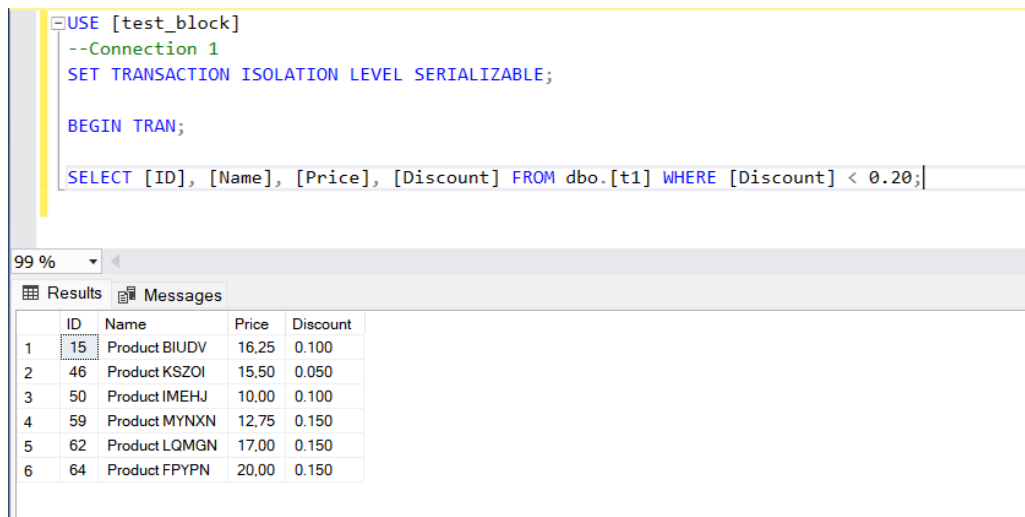


Рисунок 28 - Установка уровня изоляции SERIALIZABLE и запуск транзакции

5.3.13. Код запроса для добавления нового товара представлен на рисунке 29. Данная попытка будет заблокирована.

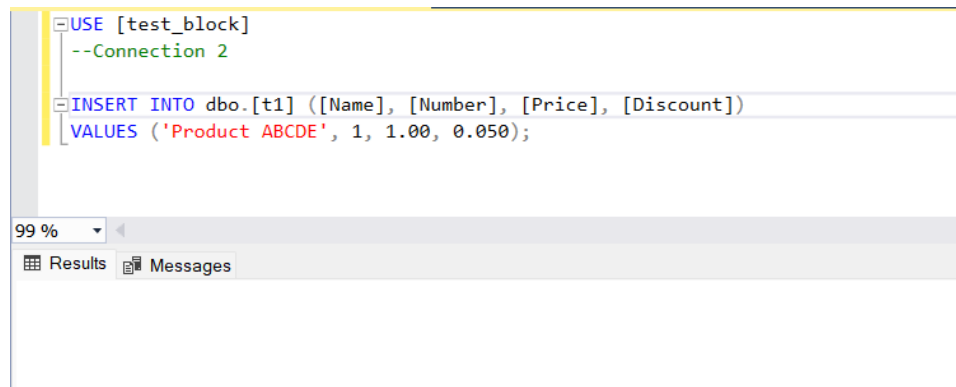


Рисунок 29 – Код запроса для добавления товара

5.3.14. Код запроса для извлечения данных и фиксирования транзакции представлен на рисунке 30.

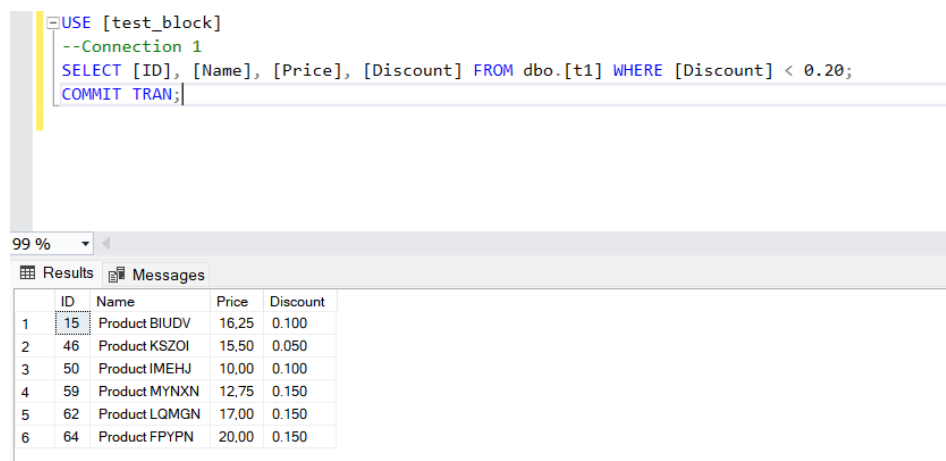


Рисунок 30 - Код запроса для извлечения данных и фиксирования транзакции

5.3.15. Результат получения монопольной блокировки и добавления строки в окне соединения 2 представлен на рисунке 31.

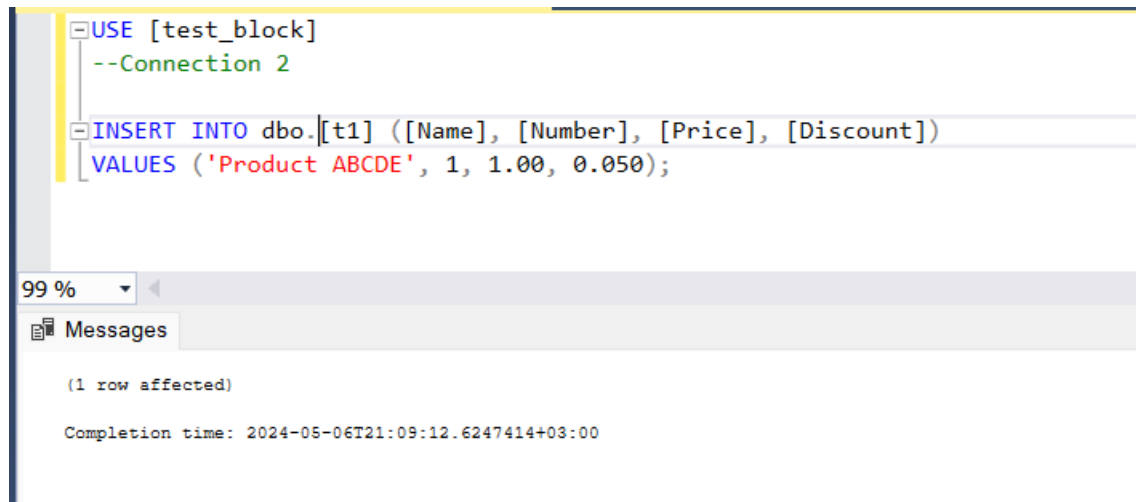


Рисунок 31 - Результат получения монопольной блокировки и добавления строки

5.3.16. Код запроса для возврата уровня изоляции к значению, принятому по умолчанию, представлен на рисунке 32.

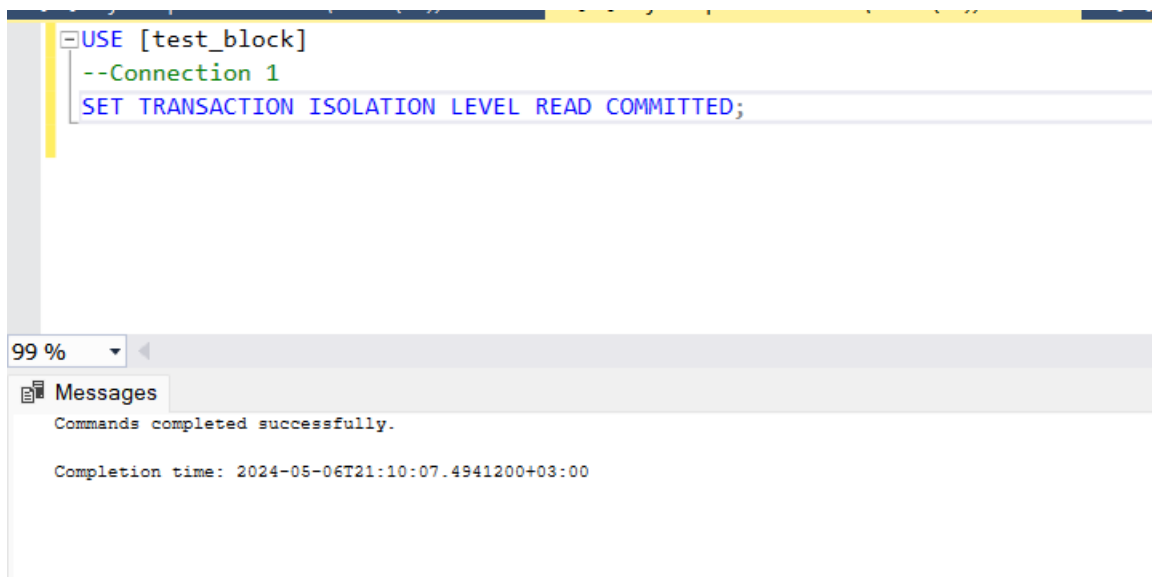


Рисунок 32 - Код запроса для возврата уровня изоляции к значению, принятому по умолчанию

## **6. Выводы и анализ результатов работы**

В ходе выполнения данной лабораторной работы были получены практические и теоретические навыки по созданию и управлению транзакциями с помощью инструкций BEGIN TRAN, COMMIT TRAN и ROLLBACK TRAN, по поиску и обнаружению блокирования, а также по уровням изоляции: READ UNCOMMITTED, READ COMMITTED, REPEATABLE READ, SERIALIZABLE.

Все поставленные задачи были выполнены с помощью подробной инструкции, представленной в тексте лабораторной работы, однако в ходе выполнения лабораторной работы возникли некоторые трудности. В ходе выполнения пункта 5.2.6 были неверно определены подключения, связанные с процессами, входящими в цепочку блокировок. В связи с этими дальнейшее выполнение данной задачи было неверным. Данная трудность была связана с неверным пониманием работы блокировок. Для решения данной проблемы была прочитана дополнительная литература по данной теме и задание было выполнено заново. В ходе повторного выполнения задания трудностей не возникало.