

**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 7  
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 создание триггеров.

## 2. Задачи, решаемые при выполнении работы

- 2.1. Создание триггера DML.
- 2.2. Создание триггера DDL.
- 2.3. Удаление триггера DDL.

## 3. Объект исследования

Объектом исследования в данной лабораторной работе являются способы создания и удаления триггеров.

## 4. Исходные данные

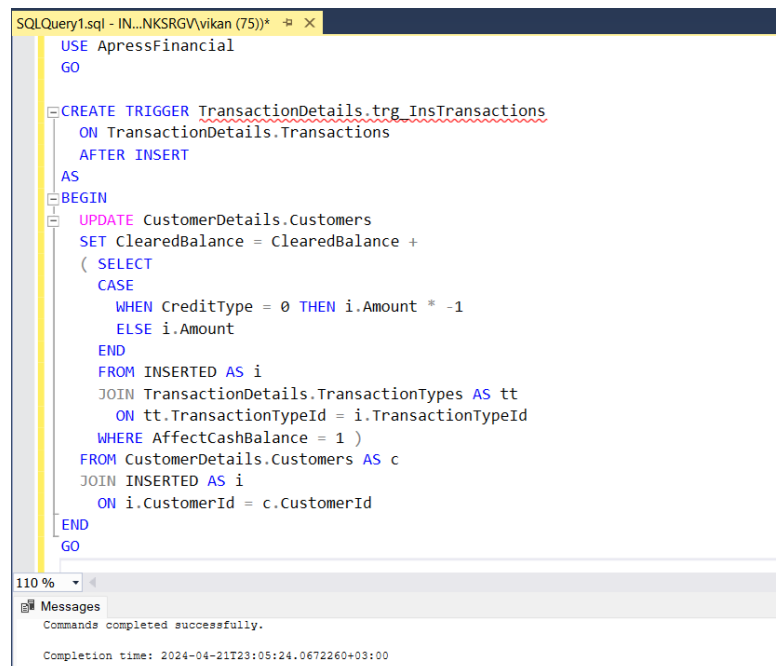
Инструкция к лабораторной работе, инструмент Microsoft SQL Server Management Studio 2019, база данных, таблицы, индексы, представления, процедуры, созданные в предыдущих лабораторных работах

## 5. Выполнение работы

### 5.1 Создание триггера DML.

В ходе выполнения данной задачи необходимо ознакомиться со способом создания DML триггера с помощью инструмента Microsoft SQL Server Management Studio.

5.1.1. Код запроса для создания триггера для вставки данных представлен на рисунке 1.



```
SQLQuery1.sql - IN...NKSRGV\vikan (75)*
USE ApressFinancial
GO

CREATE TRIGGER TransactionDetails.trg_InsTransactions
ON TransactionDetails.Transactions
AFTER INSERT
AS
BEGIN
    UPDATE CustomerDetails.Customers
    SET ClearedBalance = ClearedBalance +
    (
        SELECT
        CASE
            WHEN CreditType = 0 THEN i.Amount * -1
            ELSE i.Amount
        END
        FROM INSERTED AS i
        JOIN TransactionDetails.TransactionTypes AS tt
        ON tt.TransactionTypeId = i.TransactionTypeId
        WHERE AffectCashBalance = 1
    )
    FROM CustomerDetails.Customers AS c
    JOIN INSERTED AS i
    ON i.CustomerId = c.CustomerId
END
GO
```

110 %

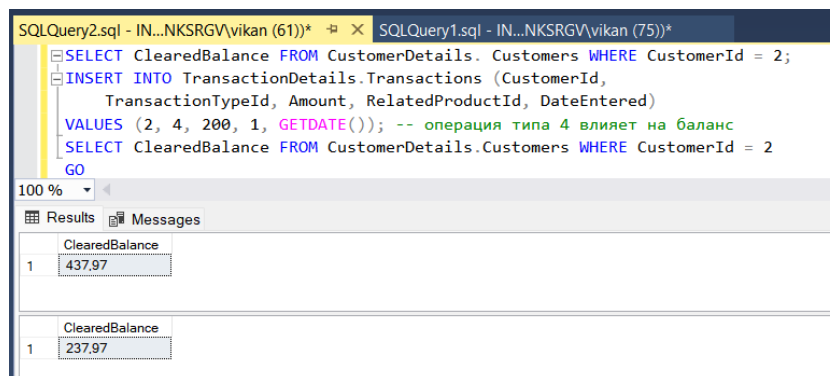
Messages

Commands completed successfully.

Completion time: 2024-04-21T23:05:24.0672260+03:00

Рисунок 1 – Код запроса для создания триггера

5.1.2. Код запроса, реализующего пакет для проверки триггера, созданного в пункте 5.1.1, представлен на рисунке 2. Результаты проверки отображают ожидаемое уменьшение баланса клиента на 200\$.



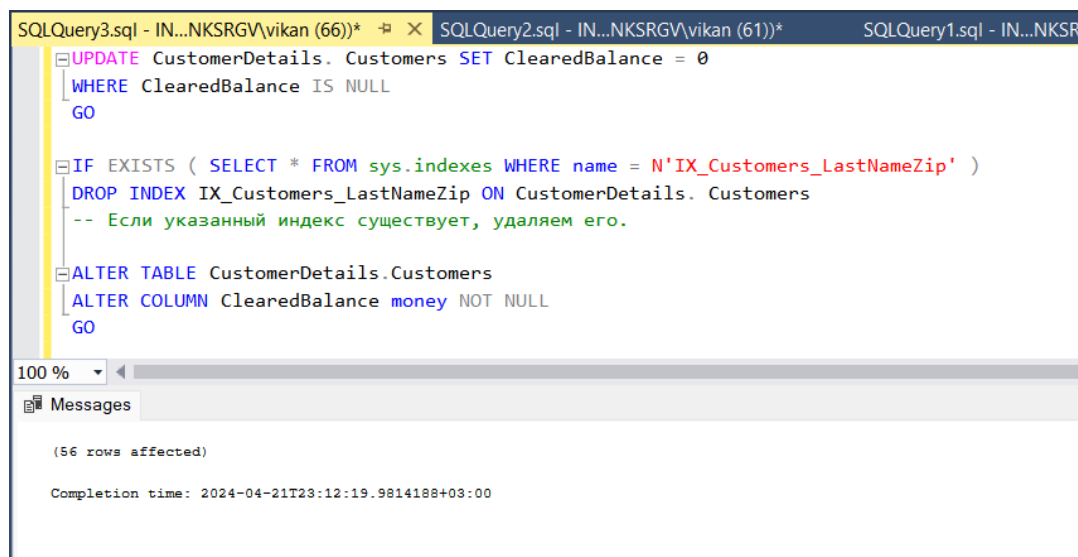
The screenshot shows a SQL query window with the following code:

```
SELECT ClearedBalance FROM CustomerDetails.Customers WHERE CustomerId = 2;  
INSERT INTO TransactionDetails.Transactions (CustomerId,  
TransactionTypeId, Amount, RelatedProductId, DateEntered)  
VALUES (2, 4, 200, 1, GETDATE()); -- операция типа 4 влияет на баланс  
SELECT ClearedBalance FROM CustomerDetails.Customers WHERE CustomerId = 2  
GO
```

The results pane shows two tables. The first table has one row with 'ClearedBalance' 437.97. The second table has one row with 'ClearedBalance' 237.97.

Рисунок 2 – Код запроса для проверки триггера

5.1.3. Код запроса, изменяющего столбец ClearedBalance, для повторного тестирования триггера представлен на рисунке 3.



The screenshot shows a SQL query window with the following code:

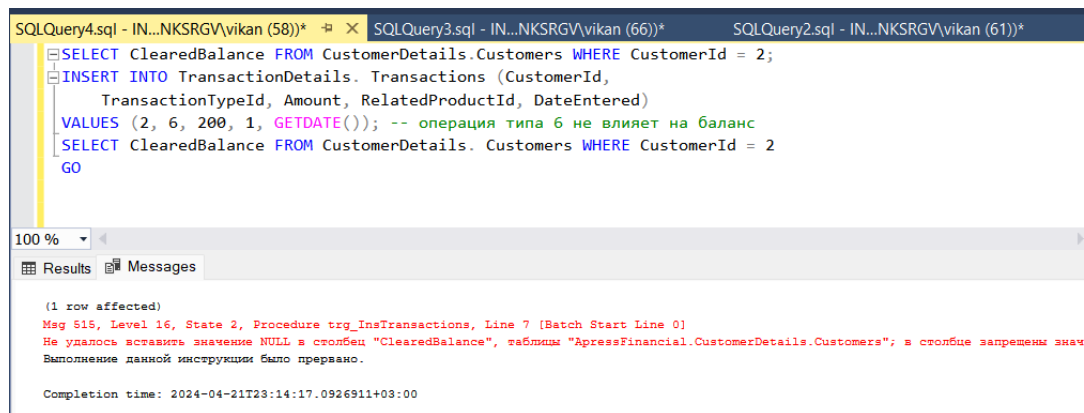
```
UPDATE CustomerDetails.Customers SET ClearedBalance = 0  
WHERE ClearedBalance IS NULL  
GO  
  
IF EXISTS ( SELECT * FROM sys.indexes WHERE name = N'IX_Customers_LastNameZip' )  
DROP INDEX IX_Customers_LastNameZip ON CustomerDetails.Customers  
-- Если указанный индекс существует, удаляем его.  
  
ALTER TABLE CustomerDetails.Customers  
ALTER COLUMN ClearedBalance money NOT NULL  
GO
```

The messages pane shows the following output:

```
(56 rows affected)  
  
Completion time: 2024-04-21T23:12:19.9814188+03:00
```

Рисунок 3 – Код запроса для повторного тестирования триггера

5.1.4. На рисунке 4 представлен код запроса, моделирующего транзакцию, которая не должна влиять на баланс клиента. В ходе выполнения данного запроса появляется сообщение об ошибке.



```
SQLQuery4.sql - IN...NKS...vikan (58))* SQLQuery3.sql - IN...NKS...vikan (66))* SQLQuery2.sql - IN...NKS...vikan (61))*
SELECT ClearedBalance FROM CustomerDetails.Customers WHERE CustomerId = 2;
INSERT INTO TransactionDetails.Transactions (CustomerId,
TransactionTypeId, Amount, RelatedProductId, DateEntered)
VALUES (2, 6, 200, 1, GETDATE()); -- операция типа 6 не влияет на баланс
SELECT ClearedBalance FROM CustomerDetails.Customers WHERE CustomerId = 2
GO
```

100 %

Results Messages

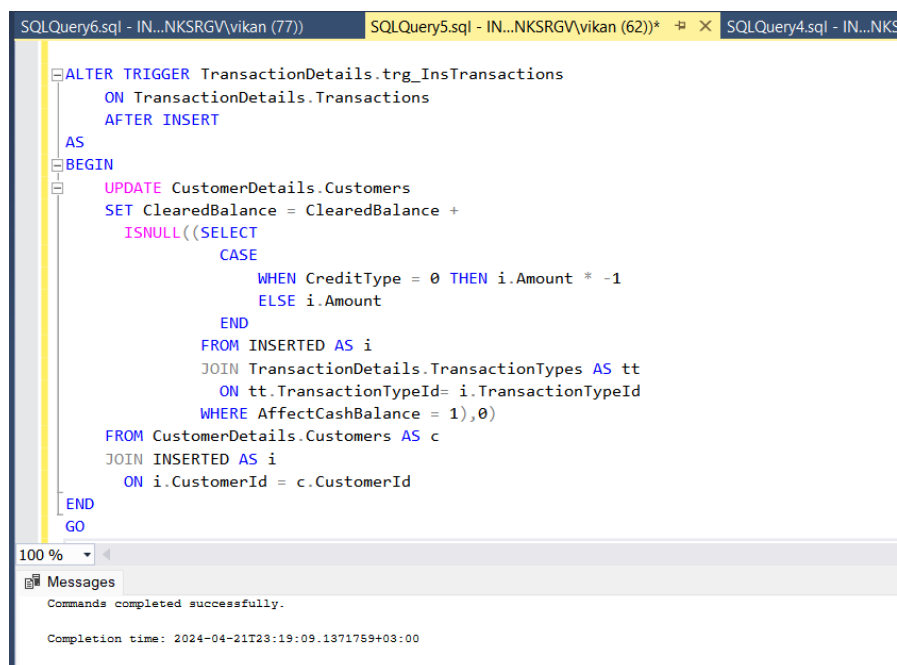
(1 row affected)

Msg 815, Level 16, State 2, Procedure trg\_InsTransactions, Line 7 [Batch Start Line 0]  
Не удалось вставить значение NULL в столбец "ClearedBalance", таблиц "ApressFinancial.CustomerDetails.Customers"; в столбце запрещены знач  
Выполнение данной инструкции было прервано.

Completion time: 2024-04-21T23:14:17.0926911+03:00

Рисунок 4 – Код запроса, моделирующего транзакцию

5.1.5. Код запроса, в котором представлено изменение триггера с помощью инструкции ALTER TRIGGER, представлен на рисунке 5.



```
SQLQuery6.sql - IN...NKS...vikan (77)) SQLQuery5.sql - IN...NKS...vikan (62))* SQLQuery4.sql - IN...NKS...vikan (58))*
ALTER TRIGGER TransactionDetails.trg_InsTransactions
ON TransactionDetails.Transactions
AFTER INSERT
AS
BEGIN
UPDATE CustomerDetails.Customers
SET ClearedBalance = ClearedBalance +
ISNULL((SELECT
CASE
WHEN CreditType = 0 THEN i.Amount * -1
ELSE i.Amount
END
FROM INSERTED AS i
JOIN TransactionDetails.TransactionTypes AS tt
ON tt.TransactionTypeId= i.TransactionTypeId
WHERE AffectCashBalance = 1),0)
FROM CustomerDetails.Customers AS c
JOIN INSERTED AS i
ON i.CustomerId = c.CustomerId
END
GO
```

100 %

Messages

Commands completed successfully.

Completion time: 2024-04-21T23:19:09.1371759+03:00

Рисунок 5 – Код запроса для изменения триггера

5.1.6. Код запроса для повторного моделирования транзакции, которая привела к ошибке и отмене вставки, представлен на рисунке 6. Теперь транзакция выполняется без ошибок, а баланс счета не изменяется.

```

SQLQuery7.sql - IN...NKSRGV\vikar (67))*
SELECT TransactionId, TransactionTypeId, Amount, DateEntered
FROM TransactionDetails.Transactions
WHERE CustomerId = 2 ORDER BY TransactionId DESC;
SELECT ClearedBalance FROM CustomerDetails.Customers WHERE CustomerId = 2;
INSERT INTO TransactionDetails.Transactions (CustomerId,
TransactionTypeId, Amount, RelatedProductId, DateEntered)
VALUES (2, 6, 200, 1, GETDATE()); -- операция типа 6 не влияет на баланс
SELECT TransactionId, TransactionTypeId, Amount, DateEntered
FROM TransactionDetails.Transactions
WHERE CustomerId = 2 ORDER BY TransactionId DESC;
SELECT ClearedBalance FROM CustomerDetails.Customers WHERE CustomerId = 2
GO

Results
TransactionId TransactionTypeId Amount DateEntered
1 1005 4 200,00 2024-04-21 23:09:14.823
2 607 2 34215,274 2012-03-06 14:20:18.770
3 244 1 14149,7153 2012-03-22 22:04:05.440

ClearedBalance
1 237,97

TransactionId TransactionTypeId Amount DateEntered
1 1007 6 200,00 2024-04-21 23:21:28.787
2 1005 4 200,00 2024-04-21 23:09:14.823
3 607 2 34215,274 2012-03-06 14:20:18.770
4 244 1 14149,7153 2012-03-22 22:04:05.440

ClearedBalance
1 237,97

```

Рисунок 6 – Код запроса для повторного моделирования транзакции

## 5.2. Создание триггера DDL.

В ходе выполнения данной задачи необходимо ознакомиться со способом создания DDL триггера с помощью инструмента Microsoft SQL Server Management Studio.

5.2.1. Код запроса для создания триггера, который будет выполняться при создании хранимой процедуры, ее изменении и удалении, представлен на рисунке 7. При выполнении любого из вышеперечисленных действий триггер проверит время дня, и, если это рабочие часы, действие будет отменено.

```

SQLQuery8.sql - IN...NKSRGV\vikar (68))*
CREATE TRIGGER trg_Sprocs
ON DATABASE
FOR CREATE_PROCEDURE, ALTER_PROCEDURE, DROP_PROCEDURE
AS
IF DATEPART(hh, GETDATE()) > 9 AND DATEPART(hh, GETDATE()) < 17
BEGIN
DECLARE @Message nvarchar(max)
SELECT @Message = 'Completing work during core hours. Trying to release -'
+ EVENTDATA().value
('(/EVENT_INSTANCE/TSQLCommand/CommandText) [1]', 'nvarchar(max)')
RAISERROR (@Message, 16, 1)
ROLLBACK
EXEC msdb.dbo.sp_send_dbmail
@profile_name = 'SQL Server Database Mail Profile',
@recipients = 'exam@limtu.spb.ru', @body = 'A stored procedure change',
@subject = 'A stored procedure change has been initiated
and rolled back during core hours'
END
GO

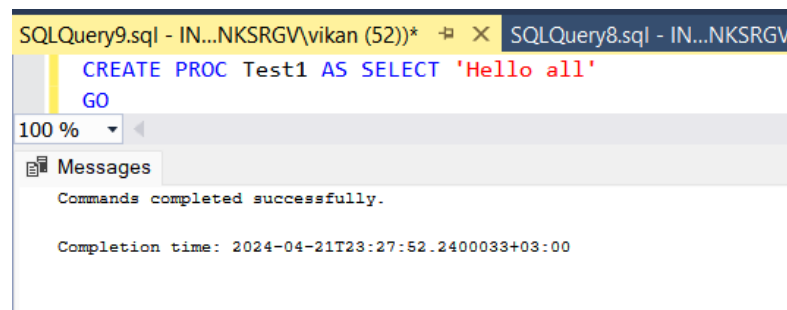
Messages
Commands completed successfully.

Completion time: 2024-04-21T23:26:54.1573978+03:00

```

Рисунок 7 – Код запроса для создания триггера

5.2.2. Код запроса для тестирования триггера путем создания хранимой процедуры представлен на рисунке 8. Запроса был выполнен, так как процедура была создана в нерабочее время.

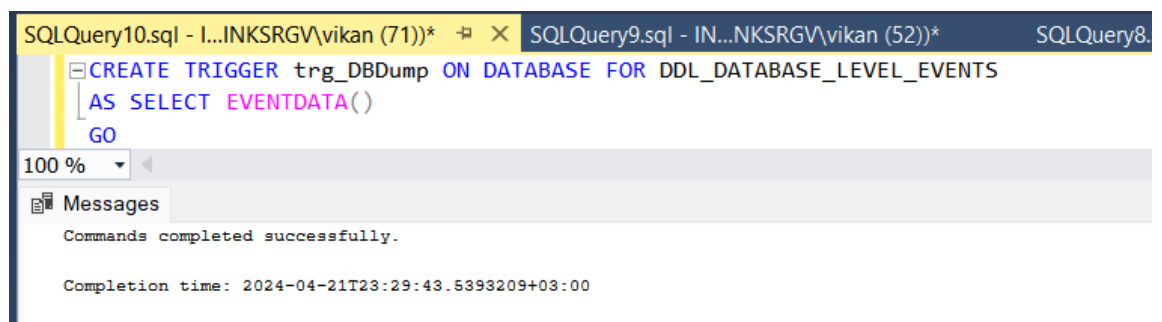


The screenshot shows a SQL query window with the following code: `CREATE PROC Test1 AS SELECT 'Hello all'` followed by `GO`. The Messages pane below indicates that the commands were completed successfully. The completion time is 2024-04-21T23:27:52.2400033+03:00.

```
SQLQuery9.sql - I...NKSRGV\vikan (52))* X SQLQuery8.sql - I...NKSRGV\vikan (52))* X
CREATE PROC Test1 AS SELECT 'Hello all'
GO
100 %
Messages
Commands completed successfully.
Completion time: 2024-04-21T23:27:52.2400033+03:00
```

Рисунок 8 – Код запроса для тестирования триггера

5.2.3. Код запроса для создания еще одного триггера представлен на рисунке 9. Данный триггер будет выполняться по любому действию, осуществляемому в БД.

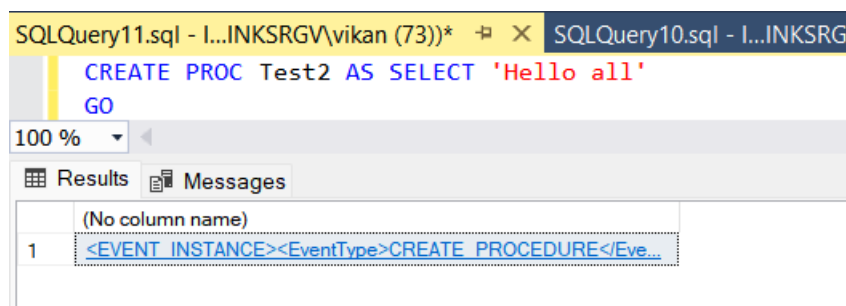


The screenshot shows a SQL query window with the following code: `CREATE TRIGGER trg_DBDump ON DATABASE FOR DDL_DATABASE_LEVEL_EVENTS` followed by `AS SELECT EVENTDATA()` and `GO`. The Messages pane below indicates that the commands were completed successfully. The completion time is 2024-04-21T23:29:43.5393209+03:00.

```
SQLQuery10.sql - I...INKSRGV\vikan (71))* X SQLQuery9.sql - I...NKSRGV\vikan (52))* X SQLQuery8.sql - I...INKSRGV\vikan (52))* X
CREATE TRIGGER trg_DBDump ON DATABASE FOR DDL_DATABASE_LEVEL_EVENTS
AS SELECT EVENTDATA()
GO
100 %
Messages
Commands completed successfully.
Completion time: 2024-04-21T23:29:43.5393209+03:00
```

Рисунок 9 – Код запроса для создания триггера

5.2.4. Код запроса для тестирования триггера путем создания хранимой процедуры представлен на рисунке 10.



The screenshot shows a SQL query window with the following code: `CREATE PROC Test2 AS SELECT 'Hello all'` followed by `GO`. The Results pane below shows the execution results, including the event instance details.

```
SQLQuery11.sql - I...INKSRGV\vikan (73))* X SQLQuery10.sql - I...INKSRGV\vikan (71))* X
CREATE PROC Test2 AS SELECT 'Hello all'
GO
100 %
Results Messages
(No column name)
1 <EVENT_INSTANCE><EventType>CREATE PROCEDURE</Event...
```

Рисунок 10 – Код запроса для тестирования триггера

5.2.5. Полученные после выполнения запроса в пункте 5.2.4 данные в формате XML представлены на рисунке 11.

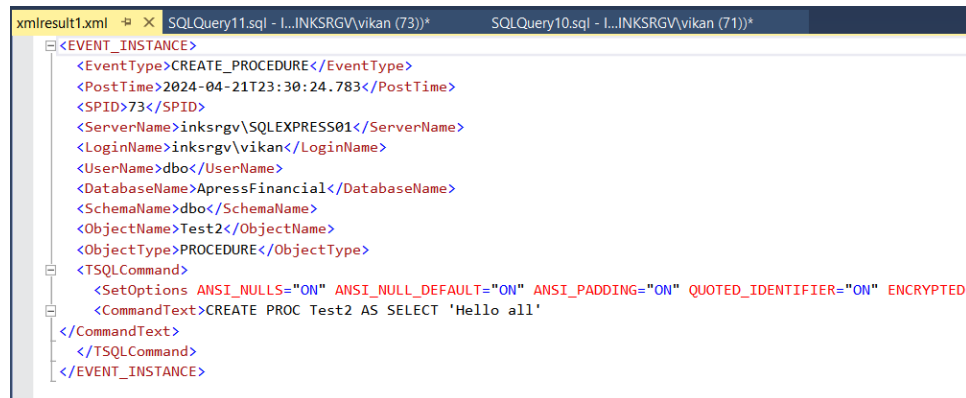


Рисунок 11 – Полученные данные

### 5.3. Удаление триггера DDL.

В ходе выполнения данной задачи необходимо ознакомиться со способом удаления триггеров и процедур с помощью инструмента Microsoft SQL Server Management Studio.

5.3.1. Код запроса для удаления триггеров и результат его выполнения представлены на рисунке 12. В ходе его выполнения триггеры не были удалены, так как не были найдены.

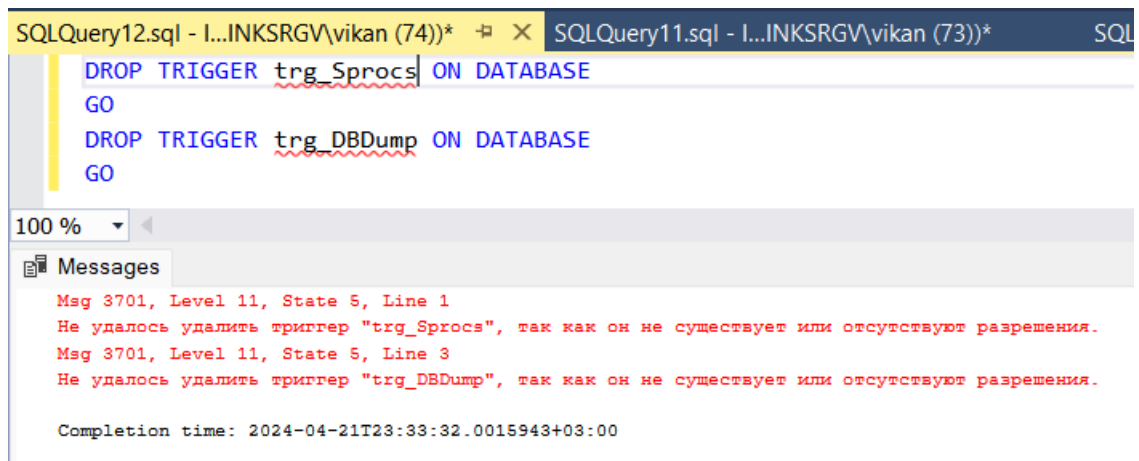


Рисунок 12 – Код запроса для удаления триггеров

5.3.2. Код запроса для успешного удаления триггеров представлен на рисунке 13.

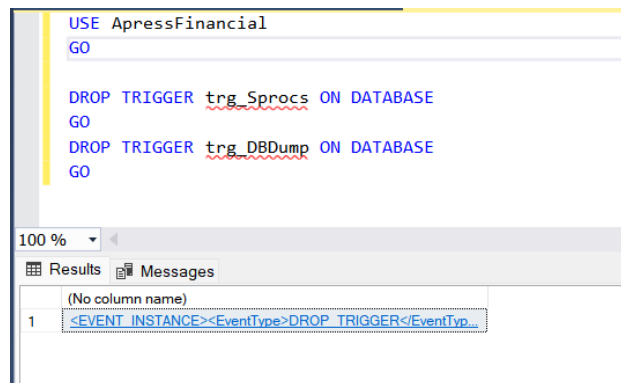


Рисунок 13 – Код запроса для удаления триггеров

5.3.3. Код запроса для успешного удаления ранее созданных процедур представлен на рисунке 14.

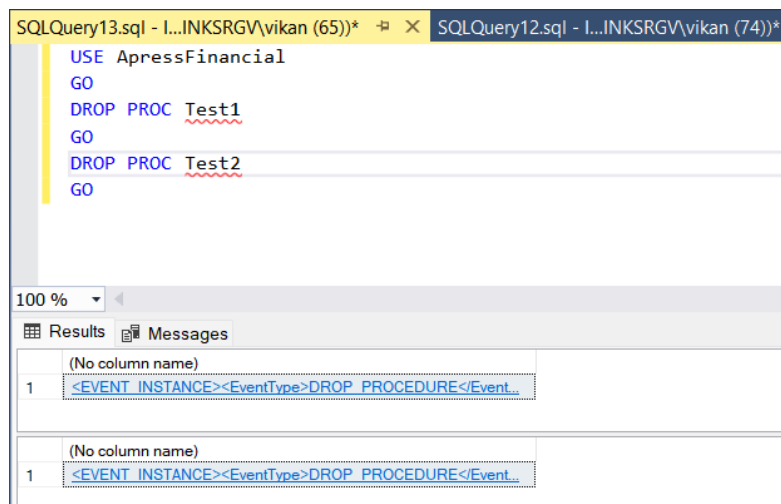


Рисунок 14 – Код запроса для удаления процедур

5.3.4. Результат успешного удаления триггеров и процедур представлен на рисунке 15.

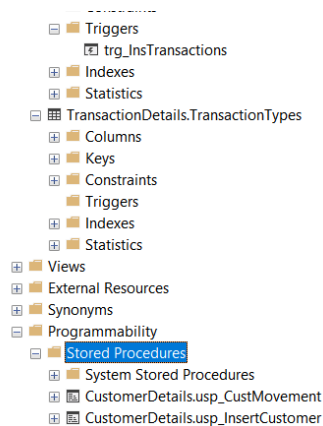


Рисунок 15 – Удаление триггеров и процедур



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

В ходе выполнения данной лабораторной работы были получены практические и теоретические навыки по созданию, изменению и удалению триггеров в инструменте SQL Server Management Studio, а также были закреплены знания по хранимым процедурам, полученные в ходе выполнения лабораторной работы №6.

Все поставленные задачи были выполнены с помощью подробной инструкции, представленной в тексте лабораторной работы, однако в ходе выполнения лабораторной работы возникли некоторые трудности. Первая трудность возникла в ходе выполнения пункта 5.1.1. Данная трудность была связана с тем, что триггер должен был создаваться для одной из таблиц БД ApressFinancial, однако она не была использована в начале выполнения запроса. Для устранения ошибки в ходе выполнения запроса было добавлено использование БД для создания в ней триггера. Вторая трудность возникла в ходе выполнения задачи 5.3. Ошибка возникала при удалении ранее созданных триггеров и процедур. Данная ошибка также была связана с отсутствием использования БД, в которую были записаны триггеры и процедуры. После добавления использования БД в начало кода ошибка была устранена и удаление триггеров и процедур прошло успешно. Во всех остальных пунктах трудностей не возникло.