

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 4
по дисциплине «Разработка баз данных»

Выполнил: студент группы К33211 Швалов Д. А.

Проверил: ст. преподаватель Осетрова И.С.

Санкт-Петербург

2024

Лабораторная работа №4 «Обработка данных»

1. Цель работы

Обработка данных.

2. Задачи, решаемые при выполнении работы

1. Добавление данных.
2. Обновление данных.
3. Удаление данных.
4. Создание недостающих связей.

3. Объект исследования

Обработка данных в СУБД Microsoft SQL Server с помощью Microsoft SQL Server Management Studio (SSMS).

4. Исходные данные

- методические указания к лабораторной работе;
- СУБД Microsoft SQL Server;
- Microsoft SQL Server Management Studio;
- база данных ApressFinancial.

5. Выполнение работы

5.1. Первая задача

5.1.1. Создание сценария в Query Editor

С помощью контекстного меню, как это показано на рисунке 1, был открыт редактор запросов с предопределенным кодом. В соответствии с заданием, данный код был отредактирован таким образом, как показано на рисунке 2. Также была использована инструкция «SET QUOTED_IDENTIFIER OFF» для того, чтобы строки в двойных кавычках рассматривались как строковые литералы. После выполнения данного кода появилось сообщение в таблицу «Shares» была добавлена новая строка.

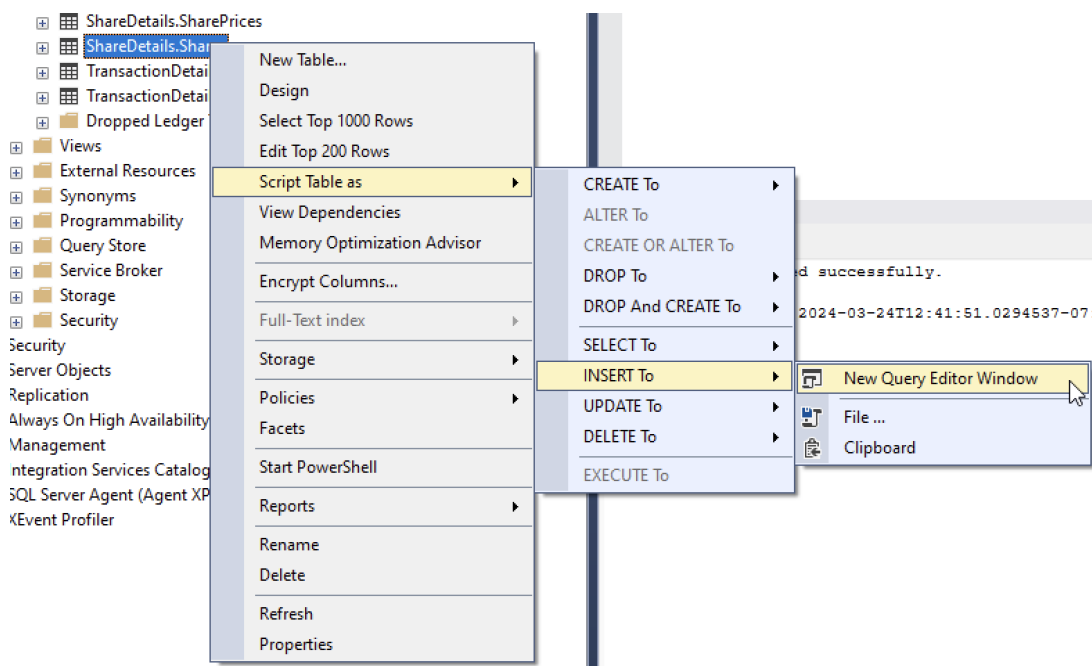


Рисунок 1 – Меню открытия редактора запросов

```

SET QUOTED_IDENTIFIER OFF
GO

USE [ApressFinancial]
GO

INSERT INTO [ShareDetails].[Shares]
    ([Description]
    ,[StockExchangeTicker]
    ,[CurrentPrice])
VALUES
    ('ACME'S HOMEBAKE COOKIES INC'
    ,'AHCI'
    ,2.34125)

GO

```

Рисунок 2 – Запрос добавления данных в таблицу «Shares»

5.1.2. Отметки NULL

В свойствах столбцов таблицы «Customers» было проверено, что для поля «CustomerId» свойство «Is Identity» установлено в «Yes» (рисунок 3), а также, что для поля «DateOpened» указано значение по умолчанию «getdate()» (рисунок 4). После этого с помощью контекстного меню (рисунок 5) было открыто окно редакти-

рования строк таблицы (6).

При попытке добавления строки с отсутствующими обязательными значениями для столбцов, возникает ошибка, как показано на рисунке 7. Все столбцы, которые были помечены при создании таблицы как недопускающие пустых значений, требуют обязательного заполнения. После корректного заполнения столбцов, как это показано на рисунке 8, новая строка была успешно сохранена в базу данных. Также ей был присвоен идентификатор «CustomerId», равный 6.

После этого в редакторе запросов был выполнен код, показанный на рисунке 9. В отличие от кода, выполненного в предыдущем задании, данный код не содержит квадратных скобок в идентификаторах схем, таблиц и столбцов.

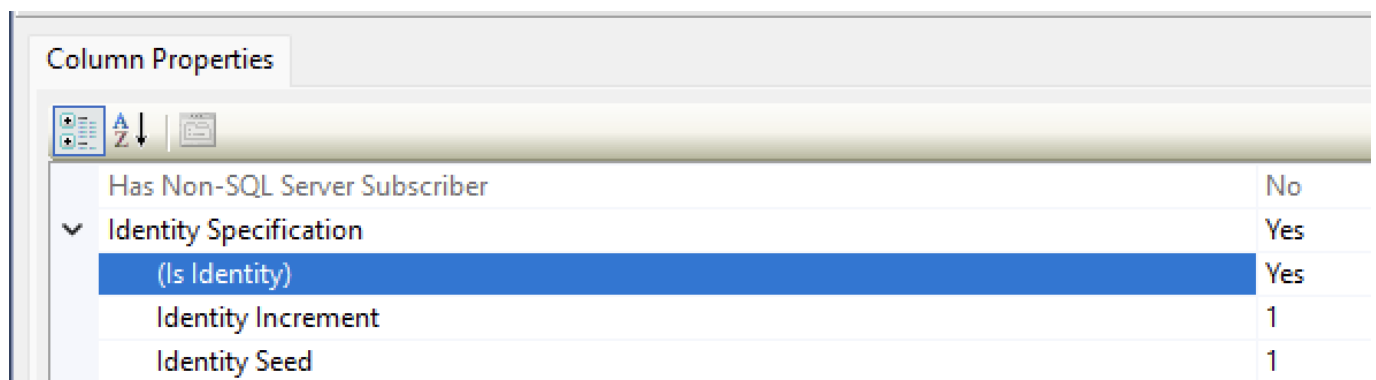


Рисунок 3 – Свойства столбца «CustomerId»

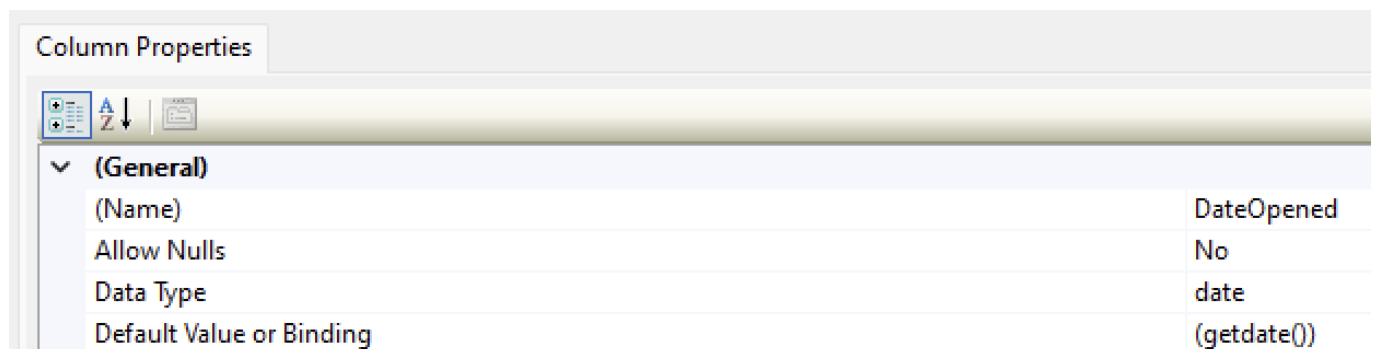


Рисунок 4 – Свойства столбца DateOpened

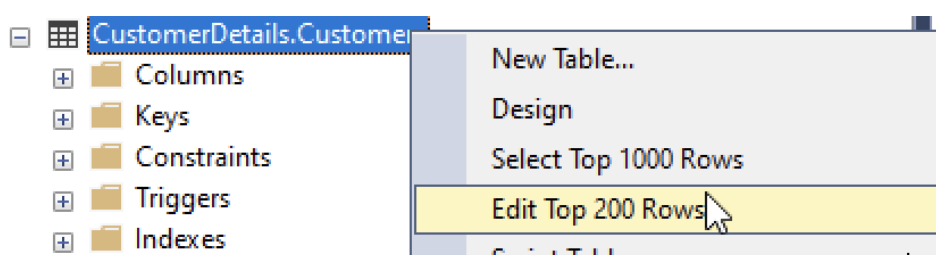


Рисунок 5 – Меню открытия окна редактирования строк

	CustomerId	Title	FirstName	OtherInitials	LastName	AddressLine1	AddressLine2	AddressLine3	TownOrCity	ZipCode	USState	AccountType
»	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Рисунок 6 – Окно редактирования строк

CustomerId	Title	FirstName	OtherInitials	LastName	AddressLine1
NULL	123	! 'Andrew' !	NULL	NULL	NULL
NULL	NULL	NULL	NULL	NULL	NULL

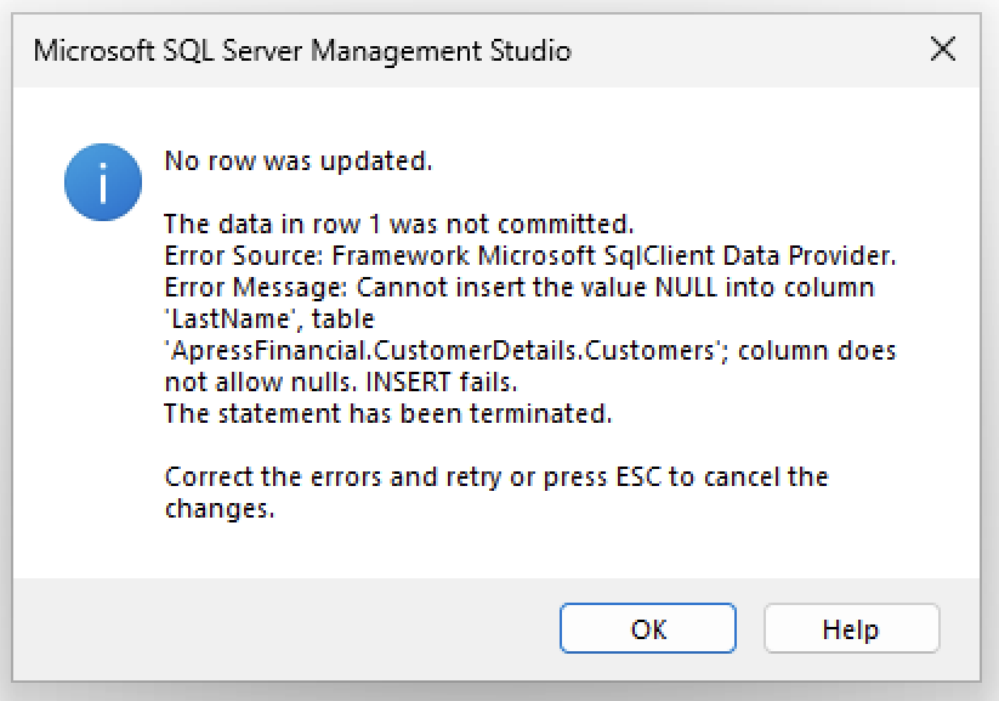


Рисунок 7 – Ошибка при добавлении строки с незаполненными обязательными столбцами

	CustomerId	Title	FirstName	OtherInitials	LastName	AddressLine1	AddressLine2	AddressLine3	TownOrCity	ZipCode	USState	AccountType	ClearedB
►	5	123	'Andrew'		'Goodman'	'Some address'	NULL	NULL	1	NULL	1	NULL	NULL
•	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Рисунок 8 – Результат добавления строки

```
INSERT INTO CustomerDetails.Customers (  
    Title,  
    LastName,  
    FirstName,  
    OtherInitials,  
    AddressLine1,  
    TownOrCity,  
    USState,  
    ZipCode,  
    AccountType,  
    ClearedBalance,  
    UnclearedBalance  
) VALUES (  
    3,  
    'Lomas',  
    'Aubrey',  
    NULL,  
    '11c Clerkenwell',  
    4,  
    8,  
    0,  
    2,  
    437.97,  
    -10.56  
)  
GO
```

Рисунок 9 – Запрос для добавления строки в таблицу

5.1.3. Инструкции проверки целостности DDBC

В запросе, показанном на рисунке 10, удаляются все строки таблицы, а затем с помощью «DBCC CHECKIDENT» идентификатору «CustomerId» устанавливается значение 0. Как видно на рисунке 10, запрос был успешно выполнен, обнуление идентификатора прошло успешно.

```
USE ApressFinancial
GO

DELETE FROM CustomerDetails.Customers
GO

DBCC CHECKIDENT ('CustomerDetails.Customers', RESEED, 0)
GO

INSERT INTO CustomerDetails.Customers (
    Title,
    LastName,
    FirstName,
    OtherInitials,
    AddressLine1,
    TownOrCity,
    USState,
    ZipCode,
    AccountType,
    ClearedBalance,
    UnclearedBalance
) VALUES
(1, 'Morgala', 'Noel', NULL, '87 Winnie Way', 4, 8, 112, 0, NULL, NULL),
(2, 'Lomas', 'Aubrey', NULL, '11c Clerkenwell', 2654, 0, 0, 2, 437.97, -10.56)
GO
```

100 %

Messages

(2 rows affected)
Checking identity information: current identity value '7'.
DBCC execution completed. If DBCC printed error messages, contact your system administrator.

(2 rows affected)

Completion time: 2024-03-25T09:34:44.4101372-07:00

Рисунок 10 – Запрос и результат его выполнения

5.1.4. Вставка значений в столбцы Identity

С помощью сценария из файла «SQLQuery_Insert_to_Identity.sql» (рисунок 11) во временную таблицу были вставлены несколько строк для проверки работы «IDENTITY_INSERT». С помощью данной директивы можно разрешить вставку идентификаторов в таблицу (вместо автоматического добавления). Как видно на рисунке 12, в таблицу были вставлены ровно те идентификаторы, которые были указаны.

```

USE ApressFinancial
GO

CREATE TABLE dbo.T1 (column_1 int IDENTITY, column_2 VARCHAR(30));

INSERT T1 VALUES ('Row #1');
INSERT T1 (column_2) VALUES ('Row #2');

SET IDENTITY_INSERT T1 ON;

INSERT INTO T1 (column_1, column_2)
VALUES (-99, 'Explicit identity value');

SELECT column_1, column_2 FROM T1;

SET IDENTITY_INSERT T1 OFF;|
INSERT T1 VALUES ('What Row?');

SELECT column_1, column_2 FROM T1;

SET IDENTITY_INSERT T1 ON;

INSERT INTO T1 (column_1, column_2)
VALUES (100, 'Explicit identity value');

SELECT column_1, column_2 FROM T1;

SET IDENTITY_INSERT T1 OFF;

INSERT T1 VALUES ('What Row?');

SELECT column_1, column_2 FROM T1;

DROP TABLE dbo.T1
GO

```

Рисунок 11 – Запрос для проверки работы «IDENTITY_INSERT»

Results			Messages		
	column_1	column_2			
1	1	Row #1			
2	2	Row #2			
3	-99	Explicit identity value			
	column_1	column_2			
1	1	Row #1			
2	2	Row #2			
3	-99	Explicit identity value			
4	3	What Row?			
	column_1	column_2			
1	1	Row #1			
2	2	Row #2			
3	-99	Explicit identity value			
4	3	What Row?			
5	100	Explicit identity value			
	column_1	column_2			
1	1	Row #1			
2	2	Row #2			
3	-99	Explicit i...			
4	3	What R...			
5	100	Explicit i...			
6	101	What R...			

Рисунок 12 – Результат выполнения запроса

5.1.5. Ограничения для столбцов

С помощью запроса, показанного на рисунке 13, для столбцов таблицы «CustomerProducts» были установлены ограничения:

- кластеризованный первичный ключ;
- ограничение, что «AmountToCollect» больше нуля;
- значение по умолчанию 0 для «Renewable».

После выполнения данного запроса данные ограничения появились в обозревателе (рисунок 14).

Затем, с помощью кнопки «Manage Check Constraints» (рисунок 15) было открыто окно изменения ограничений (рисунок 16). С помощью кнопки «Add» было добавлено новое ограничение, которое гарантирует, что дата в «LastCollection» больше или равна «LastCollected» (рисунок 17). После сохранения изменений дан-

ное ограничение было проверено запросом, представленном на рисунке 18. Как видно на рисунке 19, некорректные (с точки зрения ограничений) данные не были вставлены в таблицу.

```
USE ApressFinancial
GO

ALTER TABLE CustomerDetails.CustomerProducts
ADD CONSTRAINT PK_CustomerProducts
PRIMARY KEY CLUSTERED (CustomerFinancialProductId)
ON [SECONDARY]
GO

ALTER TABLE CustomerDetails.CustomerProducts
WITH NOCHECK
ADD CONSTRAINT CK_CustProds_AmtCheck
CHECK (AmountToCollect > 0)
GO

ALTER TABLE CustomerDetails.CustomerProducts
WITH NOCHECK
ADD CONSTRAINT DK_CustProds_Renewable
DEFAULT (0) FOR Renewable
GO
```

Рисунок 13 – Запрос для создания ограничений столбцов таблицы

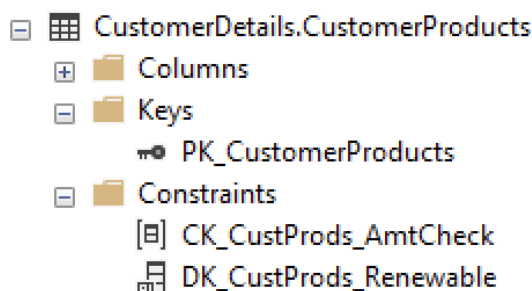


Рисунок 14 – Созданные ограничения

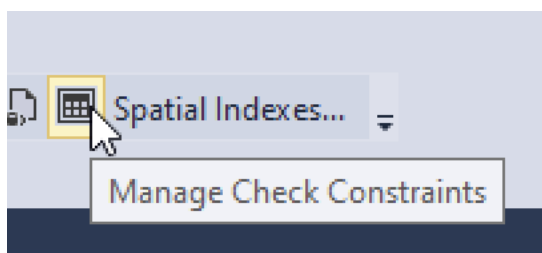


Рисунок 15 – Кнопка открытия окна изменения ограничений

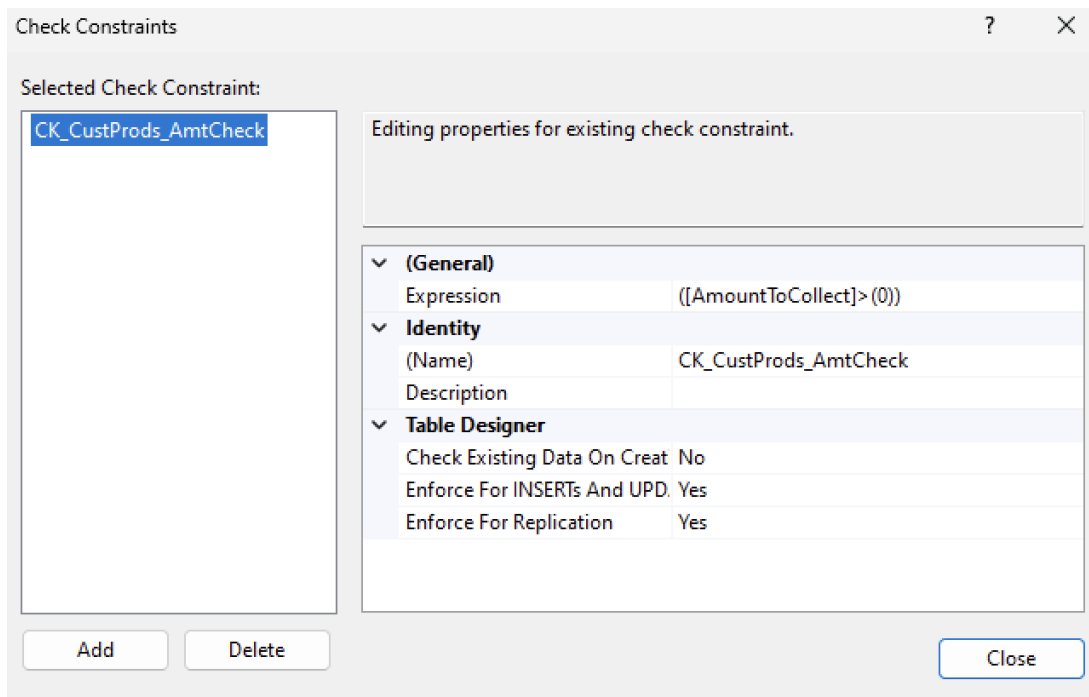


Рисунок 16 – Окно изменения ограничений таблицы

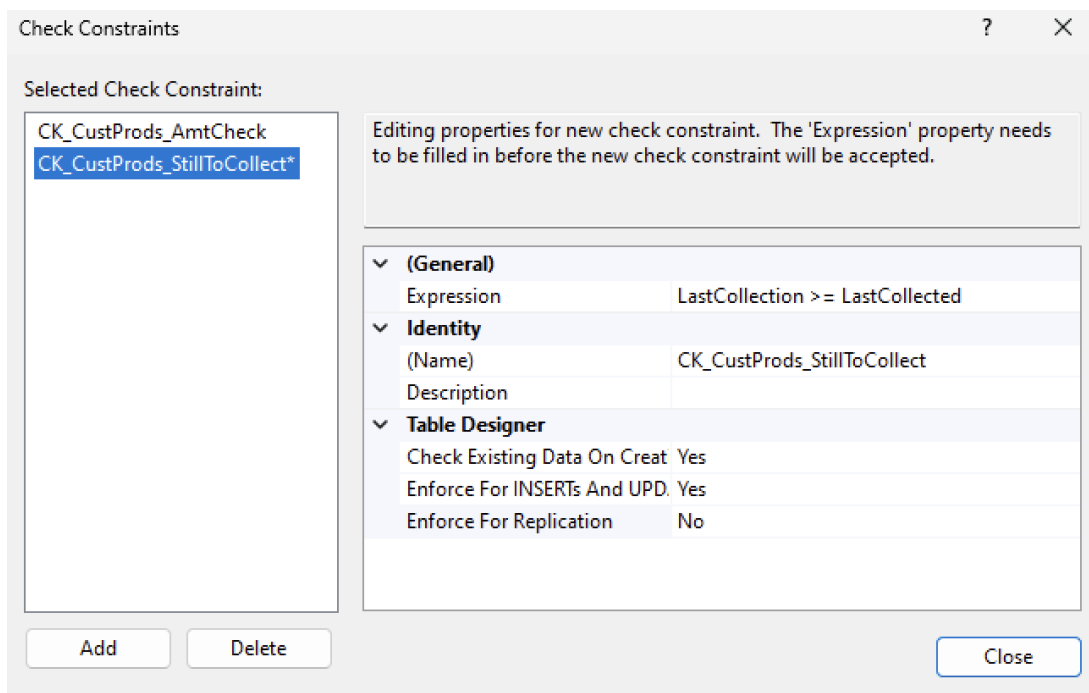


Рисунок 17 – Новое ограничение

```

SET DATEFORMAT dmy
GO

]INSERT INTO CustomerDetails.CustomerProducts (
    CustomerFinancialProductId,
    CustomerId,
    FinancialProductId,
    AmountToCollect,
    Frequency,
    LastCollected,
    LastCollection,
    Renewable
) VALUES
(1, 1, 1, -100, 0, '24-08-2014', '24-08-2017', 0)
GO

]INSERT INTO CustomerDetails.CustomerProducts (
    CustomerFinancialProductId,
    CustomerId,
    FinancialProductId,
    AmountToCollect,
    Frequency,
    LastCollected,
    LastCollection
) VALUES
(1, 1, 1, 100, 0, '24-08-2014', '23-08-2014')
GO

```

Рисунок 18 – Запрос для проверки ограничений

```

Msg 547, Level 16, State 0, Line 4
The INSERT statement conflicted with the CHECK constraint "CK_CustProds_AmtCheck". The conflict occurred in database "ApressFinancial", table "CustomerDetails.CustomerProducts".
The statement has been terminated.
Msg 547, Level 16, State 0, Line 17
The INSERT statement conflicted with the CHECK constraint "CK_CustProds_StillToCollect". The conflict occurred in database "ApressFinancial", table "CustomerDetails.CustomerPr".
The statement has been terminated.
Completion time: 2024-03-25T23:08:00.6031750-07:00

```

Рисунок 19 – Результат выполнения запроса

5.1.6. Одновременная вставка нескольких записей

С помощью запроса, показанного на рисунке 20, в таблицу «Customers» было вставлено несколько записей с помощью одного «INSERT INTO». Как видно на рисунке 21, данные были успешно вставлены в таблицу.

```

INSERT INTO CustomerDetails.Customers (
    Title,
    FirstName,
    OtherInitials,
    LastName,
    AddressLine1,
    AddressLine2,
    TownOrCity,
    USState,
    ZipCode,
    ClearedBalance,
    UnclearedBalance
) VALUES
(
    1, 'Bernie', 'I', 'McGee', '35 Baldrick Ave',
    'West Hampsted', 18651, 0, 0, 6653.11, 0.00
),
(
    3, 'Jane', 'L', 'Harper', '122 West Drive',
    'Tuala', 185111, 14, 111, 53.11, 900.00
),
(
    12, 'Mark', NULL, 'Vernon-Smith', '4155b Fredericks Street',
    'London', 157, 0, 97, -751, 900.00
)
GO

```

Рисунок 20 – Вставка нескольких записей

Results	Messages	CustomerId	Title	FirstName	OtherInitials	LastName	AddressLine1	AddressLine2	AddressLine3	TownOrCity	ZipCode	USState	Account Type	ClearedBalance	UnclearedBalance	DateOpened	DateClosed
1		1	Noel		NULL	Morgala	87 Winnie Way	NULL	NULL	4	112	8	0	NULL	NULL	2024-03-25	NULL
2		2	Aubrey		NULL	Lomas	11c Clerkenwell	NULL	NULL	2654	0	0	2	437.97	-10.56	2024-03-25	NULL
3		3	Bernie	I		McGee	35 Baldrick Ave	West Hampsted	NULL	18651	0	0	NULL	6653.11	0.00	2024-03-25	NULL
4		4	Jane	L		Harper	122 West Drive	Tuala	NULL	185111	111	14	NULL	53.11	900.00	2024-03-25	NULL
5		5	Mark		NULL	Vernon-Smith	4155b Fredericks Street	London	NULL	157	97	0	NULL	-751.00	900.00	2024-03-25	NULL

Рисунок 21 – Результат добавления данных

5.1.7. Заполнение таблиц учебной БД тестовыми данными

С помощью сценария «SQLQuery_BulkInsert.sql» в базу данных были вставлены тестовые данные для дальнейшего тестирования. При выполнении запроса ошибок не произошло, все данные были успешно добавлены.

5.1.8. Извлечение данных

С помощью запроса, показанного на рисунке 22, были извлечены данные из таблицы «Customers» (рисунок 23). После этого с помощью кнопки «Results To Text»

в меню «Query» представление было изменено на текстовое, как это показано на рисунке 24.

```
SELECT * FROM CustomerDetails.Customers
GO

]SELECT FirstName, LastName, ClearedBalance
FROM CustomerDetails.Customers
GO

]SELECT
    FirstName AS N'Имя',
    LastName AS N'Фамилия',
    ClearedBalance AS N'Баланс'
FROM CustomerDetails.Customers
GO
```

Рисунок 22 – Запрос для извлечения данных

Results															
	CustomerId	Title	FirstName	OtherInitials	LastName	AddressLine1	AddressLine2	AddressLine3	TownOrCity	ZipCode	USState	AccountType	ClearedBalance	UnclearedBalance	D
1	1	1	Noel	NULL	Morgala	87 Winnie Way	NULL	NULL	4	112	8	0	NULL	NULL	2
2	2	2	Aubrey	NULL	Lomas	11c Clerkenwell	NULL	NULL	2654	0	0	2	437.97	-10.56	2
3	3	1	Bernie	I	McGee	35 Baldrick Ave	West Hampstead	NULL	18651	0	0	NULL	6653.11	0.00	2
4	4	3	Jane	L	Harper	122 West Drive	Tuala	NULL	185111	111	14	NULL	53.11	27.50	2
5	5	12	Mark	NULL	Vernon-Smith	4155b Fredericks Street	London	NULL	157	97	0	NULL	-751.00	900.00	2
6	6	1	Terence	S	Madden	625 Second Blvd.	18 North Old Way	NULL	2107	56835	43	1	917.0126	NULL	2
7	7	0	Anne	J	Aguirre	45 South First Road	64 Oak Road	436 Old Freeway	1502	32457	50	19	NULL	NULL	2
8	8	4	Mickey	AB	Ferguson	228 Second Street	NULL	NULL	902	15476	47	21	187.2192	NULL	2

FirstName	LastName	ClearedBalance
115	Cameron Vang	23.9563
116	Cary Bird	303.0461
117	Kelli Villa	247.3405
118	Ruth Wong	929.0438
119	Shanna White	52.6566
120	Miranda Forbes	633.7092
121	Donna Mata	399.7733
122	Marilyn Farrell	41.3266

Имя	Фамилия	Баланс
1	Noel Morgala	NULL
2	Aubrey Lomas	437.97
3	Bernie McGee	6653.11
4	Jane Harper	53.11
5	Mark Vernon...	-751.00
6	Tere... Madden	917.0...
7	Anne Aguirre	NULL
8	Mickey Ferguson	187.2...
9	Dean... Palmer	148.0...
10	Anne Mann	299.8...
11	Carlos Liu	544.7...
12	Hector Ramirez	287.4...
13	Keri Steele	831.9...
14	Arturo Guerrero	453.9...
15	Molly Rivas	153.0...
16	Kristie Baker	34.01...
17	Alfred Bright	354.3...
18	Ruth McKinney	314.6...

Рисунок 23 – Результат выполнения запроса

CustomerId	Title	FirstName	OtherInitials	LastName	AddressLine1
1	1	Noel	NULL	Morgala	87 Winnie Way
2	2	Aubrey	NULL	Lomas	11c Clerkenwell
3	1	Bernie	I	McGee	35 Baldrick Ave
4	3	Jane	L	Harper	122 West Drive
5	12	Mark	NULL	Vernon-Smith	4155b Fredericks Street
6	1	Terence	S	Madden	625 Second Blvd.
7	0	Anne	J	Aguirre	45 South First Road
8	4	Mickey	AB	Ferguson	228 Second Street
9	4	Deanna	HG	Palmer	796 Hague Boulevard
10	1	Anne	HA	Mann	92 Rocky Oak Blvd.
11	2	Carlos	R	Liu	415 Green Cowley Parkway
12	4	Hector	CO	Ramirez	889 First Boulevard
13	1	Keri	ST	Steele	117 Rocky Nobel Parkway
14	4	Arturo	L	Guerrero	31 Hague Way
15	2	Molly	K	Rivas	285 Nobel Road
16	0	Kristie	A	Baker	216 North New Freeway
17	4	Alfred	QZ	Bright	937 White First St.
18	2	Ruth	O	McKinney	11 Oak Boulevard
19	2	Luz	N	Drake	66 Green Hague Avenue
20	1	Roderick	CA	Espinoza	84 Clarendon Avenue
21	0	Donnell	DN	Burgess	70 New Avenue

Рисунок 24 – Результат выполнения запроса в текстовом виде

5.2. Вторая задача

5.2.1. Простое обновление данных

С помощью запроса, показанного на рисунке 25, была изменена фамилия у строки с «CustomerId» равной 7. Как видно на рисунке 26, фамилия была успешно изменена.

```
USE ApressFinancial
GO

UPDATE CustomerDetails.Customers
SET LastName = 'Mather'
WHERE CustomerId = 7
GO
```

Рисунок 25 – Запрос на обновление данных

Results			Messages	
	FirstName	LastName		
1	Anne	Mather		

Рисунок 26 – Результат обновления данных

5.2.2. Обновление данных с помощью переменной

В запросе, который показан на рисунке 27, используется переменная, а также другие столбцы таблицы для изменения данных в таблице. Так, с помощью переменной «ValueToUpdate» изменяется фамилия клиента, а с помощью переменной «UnclearedBalance» обновляется баланс клиента. Как видно на рисунке 28, данные были успешно обновлены в соответствии с ожидаемым результатом.

```
USE ApressFinancial
GO

]SELECT FirstName, LastName, ClearedBalance, UnclearedBalance
FROM CustomerDetails.Customers
WHERE LastName = 'Booth';

DECLARE @ValueToUpdate NVARCHAR(50)
SET @ValueToUpdate = 'Prentice';

]UPDATE CustomerDetails.Customers
SET
    LastName = @ValueToUpdate,
    ClearedBalance = ClearedBalance + UnclearedBalance,
    UnclearedBalance = 0
WHERE LastName = 'Booth';

]SELECT FirstName, LastName, ClearedBalance, UnclearedBalance
FROM CustomerDetails.Customers
WHERE LastName = 'Prentice'
GO
```

Рисунок 27 – Запрос на обновление данных

Results		Messages		
	FirstName	LastName	ClearedBalance	UnclearedBalance
1	Misty	Booth	789.5141	NULL

	FirstName	LastName	ClearedBalance	UnclearedBalance
1	Misty	Prentice	NULL	0.00

Рисунок 28 – Результат обновления данных

5.2.3. Преобразование данных при обновлении

На рисунке 29 показан пример запроса, при выполнении которого происходит неявное преобразование типа данных. При выполнении данного запроса, что видно из рисунка 30, строковый тип был преобразован в денежный тип, поскольку, на самом деле, в строке лежало число. Однако, если в строке будет лежать что-то отличное от числа, например, как это показано на рисунке 31, то возникнет ошибка (рисунок 32).

```
USE ApressFinancial
GO

]SELECT FirstName, LastName, ClearedBalance, UnclearedBalance
FROM CustomerDetails.Customers
WHERE LastName = 'Prentice';

DECLARE @WrontDataType NVARCHAR(50)
SET @WrontDataType = '1422.26';

]UPDATE CustomerDetails.Customers
SET ClearedBalance = @WrontDataType
WHERE LastName = 'Prentice';

]SELECT FirstName, LastName, ClearedBalance, UnclearedBalance
FROM CustomerDetails.Customers
WHERE LastName = 'Prentice'

GO
```

Рисунок 29 – Запрос на обновление данных

Results		Messages		
	FirstName	LastName	ClearedBalance	UnclearedBalance
1	Misty	Prentice	NULL	0.00

	FirstName	LastName	ClearedBalance	UnclearedBalance
1	Misty	Prentice	1422.26	0.00

Рисунок 30 – Результат обновления данных

```

USE ApressFinancial
GO

]SELECT FirstName, LastName, ClearedBalance, UnclearedBalance
FROM CustomerDetails.Customers
WHERE LastName = 'Prentice';
-

DECLARE @WrontDataType NVARCHAR(50)
SET @WrontDataType = 'Invalid';

]UPDATE CustomerDetails.Customers
SET ClearedBalance = @WrontDataType
WHERE LastName = 'Prentice';
-

]SELECT FirstName, LastName, ClearedBalance, UnclearedBalance
FROM CustomerDetails.Customers
WHERE LastName = 'Prentice'
-
GO

```

Рисунок 31 – Некорректный запрос на обновление данных

```
(1 row affected)
Msg 235, Level 16, State 0, Line 11
Cannot convert a char value to money. The char value has incorrect syntax.
```

Рисунок 32 – Результат выполнения некорректного запроса

5.3. Третья задача

5.3.1. Создание временной таблицы

С помощью запроса, показанного на рисунке 33, была создана временная таблица данных «CustTemp», содержащая 5 первых строк из таблицы «Customers». Как видно на рисунке 34, запрос был успешно выполнен.

```
USE tempdb
GO

SELECT TOP (5) CustomerId, FirstName, OtherInitials, LastName
INTO dbo.CustTemp
FROM ApressFinancial.CustomerDetails.Customers
GO
```

Рисунок 33 – Запрос на создание временной таблицы

```
(5 rows affected)
```

Рисунок 34 – Результат выполнения запроса

5.3.2. Поведение счетчика идентификаторов при удалении строк

После выполнения запроса, показанного на рисунке 35, из временной таблицы была удалена строка с «CustomerId» равным 4. При выполнении запроса, показанного на рисунке 36, как видно на рисунке 37, новая строка получила «CustomerId» равный 6. Также, при полном удалении всех строк таблицы с помощью «DELETE FROM» и добавлении новой строки (рисунок 38), счетчик идентификаторов не обнуляется (рисунок 39).

```
USE tempdb
GO

DELETE FROM dbo.CustTemp
WHERE CustomerId = 4
GO
```

Рисунок 35 – Запрос на удаление строки

```
USE tempdb
GO

INSERT INTO dbo.CustTemp (
    FirstName,
    OtherInitials,
    LastName
) VALUES
('John', 'K.', 'Week')
GO
```

Рисунок 36 – Запрос на создание строки

Results		Messages		
	CustomerId	FirstName	OtherInitials	LastName
1	1	Noel	NULL	Morgala
2	2	Aubrey	NULL	Lomas
3	3	Bernie	I	McGee
4	6	John	K.	Week
5	5	Mark	NULL	Vernon-Smith

Рисунок 37 – Результат выполнения запросов

```

USE tempdb
GO

INSERT INTO dbo.CustTemp (
    FirstName,
    OtherInitials,
    LastName
) VALUES
('Mickey', 'K.', 'Mouse')
GO

```

Рисунок 38 – Запрос на создание строки после удаления всех строк

Results		Messages		
	CustomerId	FirstName	OtherInitials	LastName
1	7	Mickey	K.	Mouse

Рисунок 39 – Результат выполнения запроса

5.3.3. Очистка таблицы

При использовании «TRUNCATE TABLE», как это показано на рисунке 40, таблица будет полностью очищена, а также будет обнулен счетчик идентификатора. Это можно проверить с помощью запроса, показанного на рисунке 41. После выполнения запроса в таблицу была добавлена новая строка, идентификатор которой равен 1 (рисунок 42).

```

USE tempdb
GO

TRUNCATE TABLE dbo.CustTemp
GO

```

Рисунок 40 – Запрос для очистки таблицы

```

USE tempdb
GO

INSERT INTO dbo.CustTemp (
    FirstName,
    OtherInitials,
    LastName
) VALUES
('Misha', 'K.', 'Ivanov')
GO

```

Рисунок 41 – Запрос на создание новой строки

Results		Messages		
	CustomerId	FirstName	OtherInitials	LastName
1	1	Misha	K.	Ivanov

Рисунок 42 – Созданная строка

5.4. Четвертая задача

5.4.1. Создание недостающих связей

С помощью скрипта, показанного на рисунке 43, были созданы отсутствующие связи между таблицами. В итоге, получилась диаграмма, показанная на рисунке 44.

```
USE ApressFinancial
GO

ALTER TABLE CustomerDetails.CustomerProducts
ADD CONSTRAINT FK_CustomerProducts_Customers
FOREIGN KEY (CustomerId) REFERENCES CustomerDetails.Customers (CustomerId);

ALTER TABLE CustomerDetails.FinancialProducts
ADD CONSTRAINT PK_FinancialProducts
PRIMARY KEY CLUSTERED (ProductId)
ON [SECONDARY];

ALTER TABLE CustomerDetails.CustomerProducts
ADD CONSTRAINT FK_CustomerProducts_FinancialProducts
FOREIGN KEY (FinancialProductId) REFERENCES CustomerDetails.FinancialProducts (ProductId);

ALTER TABLE TransactionDetails.Transactions
ADD CONSTRAINT FK_Transactions_TransactionTypes
FOREIGN KEY (TransactionTypeId) REFERENCES TransactionDetails.TransactionTypes (TransactionTypeId);

ALTER TABLE TransactionDetails.Transactions
ADD CONSTRAINT FK_Transactions_CustomerProducts
FOREIGN KEY (RelatedProductId) REFERENCES CustomerDetails.CustomerProducts (CustomerFinancialProductId);

ALTER TABLE ShareDetails.SharePrices
ADD CONSTRAINT FK_SharePrices_Shares
FOREIGN KEY (ShareId) REFERENCES ShareDetails.Shares (ShareId);

GO
```

Рисунок 43 – Запрос на создания связей

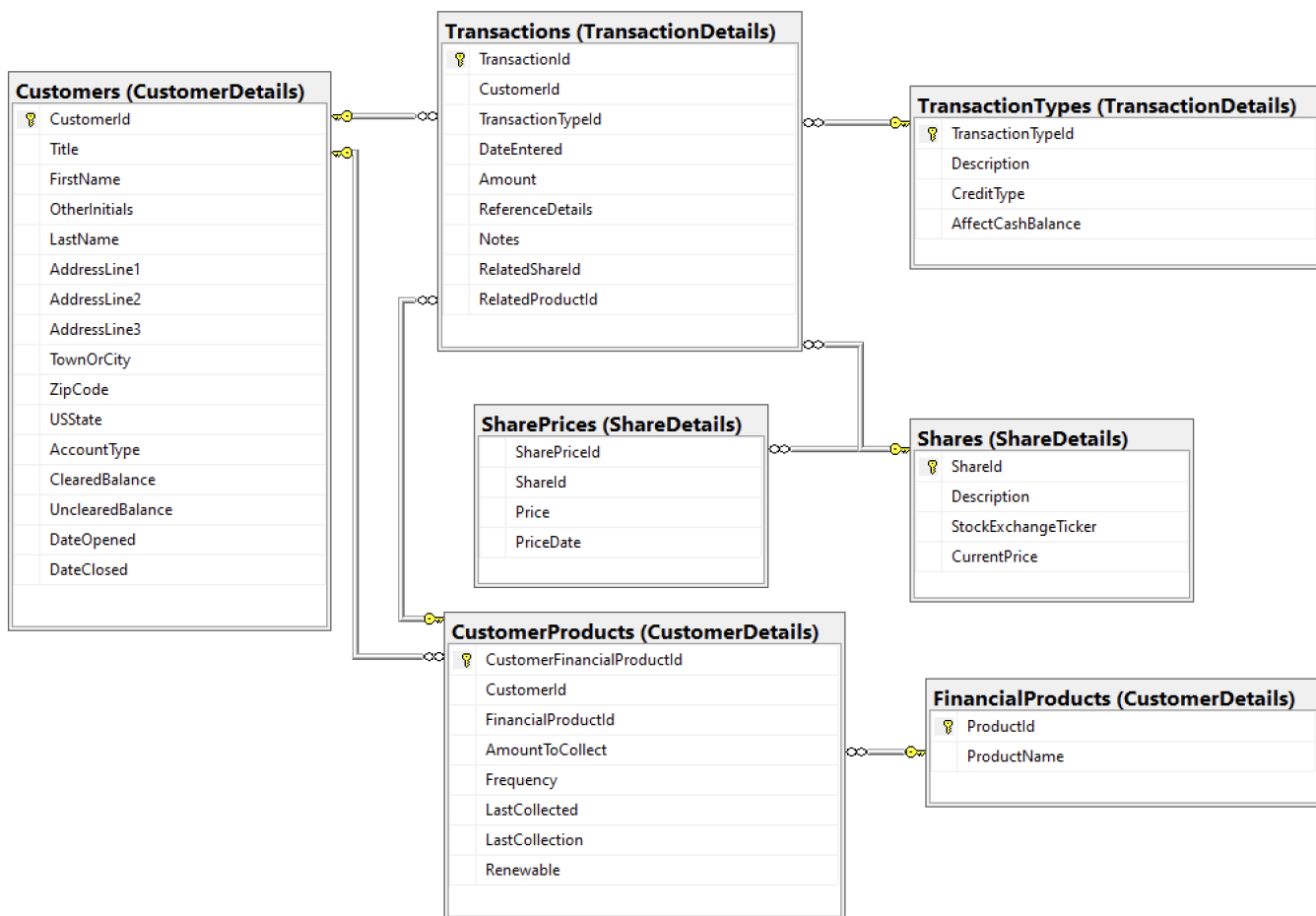


Рисунок 44 – Схема связей таблиц в базе данных

6. Выводы и анализ результатов работы

В данной лабораторной работе изучены способы обработки данных в SSMS. Цель, поставленная в начале работы, достигнута, задачи выполнены.