SQL DEV. Основные вещи. Практика

*Дано 5 таблиц:*

CREATE TABLE dbo.SD\_Subscrs -- Лицевые счета

(

[LINK] [int] IDENTITY NOT NULL PRIMARY KEY, -- Ид

[C\_FirstName] [varchar] (150) NOT NULL, -- Имя

[C\_SecondName] [varchar] (150) NOT NULL, -- Фамилия

[C\_Address] [varchar](300) NULL, -- Адрес регистрации

[C\_Doc\_Serial] [varchar](4) NULL, -- Серия паспорта

[C\_Doc\_Number] [varchar](6) NULL, -- Номер паспорта

[D\_BirthDate] [date] NOT NULL, -- Дата рождения

)

GO

CREATE TABLE dbo.DD\_Docs -- Документы

(

[LINK] [int] IDENTITY NOT NULL PRIMARY KEY, -- Ид

[C\_Number] [varchar] (50) NULL, -- Номер

[F\_Subscr] [int] NOT NULL, -- Л/с

[C\_Doc\_Type] [varchar](50) NOT NULL, -- Тип

[D\_Date] [date] NOT NULL DEFAULT GETDATE(), -- Дата

[F\_Docs] [int] NULL, -- Родительский документ

CONSTRAINT FK\_DD\_Docs\_DD\_Docs FOREIGN KEY (F\_Docs) REFERENCES dbo.DD\_Docs (LINK),

CONSTRAINT FK\_DD\_Docs\_SD\_Subscrs FOREIGN KEY (F\_Subscr) REFERENCES dbo.SD\_Subscrs (LINK) ON DELETE CASCADE

)

GO

CREATE TABLE dbo.FD\_Bills -- Счета

(

[LINK] [int] IDENTITY NOT NULL PRIMARY KEY, -- Ид

[C\_Number] [varchar] (50) NULL, -- Номер

[F\_Subscr] [int] NOT NULL, -- Л/с

[C\_Sale\_Items] [varchar](50) NOT NULL, -- Услуга

[D\_Date] [date] NOT NULL DEFAULT GETDATE(), -- Дата

[N\_Amount] [money] NOT NULL, -- Сумма

[N\_Amount\_Rest] [money] NOT NULL, -- Остаток

CONSTRAINT FK\_FD\_Bills\_SD\_Subscrs FOREIGN KEY (F\_Subscr) REFERENCES dbo.SD\_Subscrs (LINK) ON DELETE CASCADE

)

GO

CREATE TABLE dbo.FD\_Payments -- Платежи

(

[LINK] [int] IDENTITY NOT NULL PRIMARY KEY, -- Ид

[C\_Number] [varchar] (50) NULL, -- Номер

[F\_Subscr] [int] NOT NULL, -- Л/с

[D\_Date] [date] NOT NULL DEFAULT GETDATE(), -- Дата

[N\_Amount] [money] NOT NULL, -- Сумма

CONSTRAINT FK\_FD\_Payments\_SD\_Subscrs FOREIGN KEY (F\_Subscr) REFERENCES dbo.SD\_Subscrs (LINK) ON DELETE CASCADE

)

GO

CREATE TABLE dbo.FD\_Payment\_Details -- Детализация платежей

(

[F\_Payments] [int] NOT NULL PRIMARY KEY, -- Ид платежа

[F\_Bills] [int] NOT NULL, -- Ид счета

[C\_Sale\_Items] [varchar](50) NOT NULL, -- Услуга

[N\_Amount] [money] NOT NULL -- Сумма

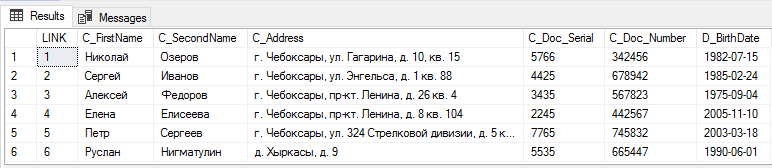
CONSTRAINT FK\_FD\_Payment\_Details\_FD\_Payments FOREIGN KEY (F\_Payments) REFERENCES dbo.FD\_Payments (LINK) ON DELETE CASCADE,

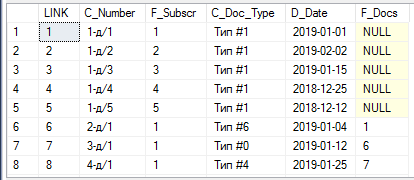
CONSTRAINT FK\_FD\_Payment\_Details\_FD\_Bills FOREIGN KEY (F\_Bills) REFERENCES dbo.FD\_Bills (LINK),

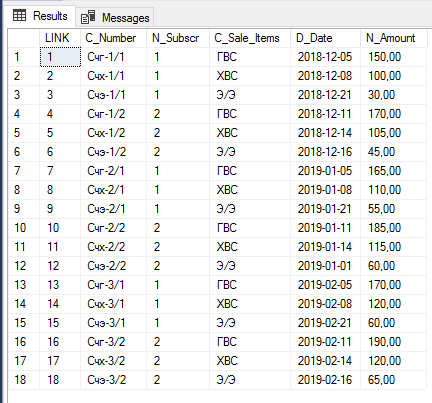
)

GO

*Со значениями соответственно:*





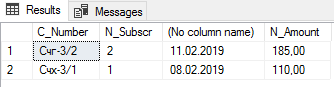


*Таблицы платежей - пустые.*

*Скрипт создания таблиц и БД «SQL Dev. Основные вещи. Практика.sql» прикладывается.*

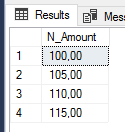
1. *Требуется написать запрос, который выведет номер, лицевой счет, дату в формате «день.месяц.год» и сумму ближайших по дате двух счетов, у которых день начисления в промежутке от 8 до 12 включительно.*

**Ожидаемый результат:**



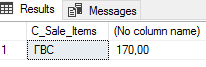
1. *Требуется вывести уникальные суммы начислений по услуге ХВС за все время*

**Ожидаемый результат:**



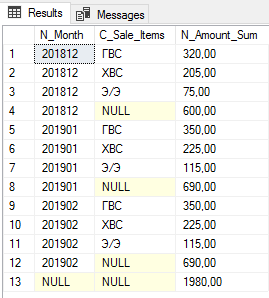
1. *Требуется вывести услугу и среднюю сумму начислений за все время, у которой средняя сумма превышает 120 р.*

**Ожидаемый результат:**



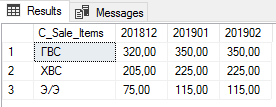
1. *Требуется вывести общие суммы начислений за каждый месяц по каждой услуге за все время с сортировкой по месяцам, с под итогами.*

**Ожидаемый результат:**



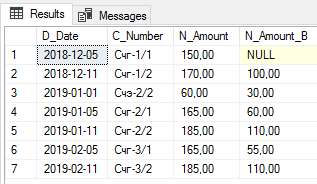
1. *Требуется вывести общие суммы начислений за каждый месяц по каждой услуге за все время, так чтобы услуги были строках, месяца в столбцах*

**Ожидаемый результат:**



1. *Требуется вывести дату номер и сумму начислений, для которых сумма превышает сумму предыдущего по дате начисления*

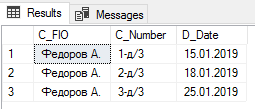
**Ожидаемый результат:**



1. *Требуется вывести список документов, совершенно летних абонентов, проживающих на проспекте Ленина.*

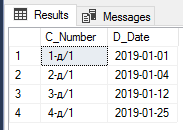
*В формате Фамилия инициал., Номер документа, Дата в формате день.месяц.год*

**Ожидаемый результат:**



1. *Требуется вывести номер и дату всех родительских документов, документа с номером "4-д/1", с сортировкой по дате*

**Ожидаемый результат:**



1. *Требуется написать хранимую процедуру dbo.UI\_FP\_Payment\_Split, которая по внесенным платежам в таблицу dbo.FD\_Payments будет расщеплять его на оплаты по конкретным счетам и услугам исходя из заполненных строк в таблице dbo.FD\_Bills.*

*Хранимая процедура так же должна пересчитывать остатки в таблице dbo.FD\_Bills.*

*Хранимая процедура будет иметь два параметра, ид платежа и тип расщепления.*

*Тип расщепления может принимать значения:*

1. *– По дате, начиная с самых старых счетов.*
2. *– Пропорционально по каждой услуге в месяце.*

**Проверка №1**:

BEGIN TRAN

DECLARE @LINK INT

INSERT dbo.FD\_Payments

SELECT 'П-123', 1, '20190105', 200

SET @LINK = SCOPE\_IDENTITY()

EXEC dbo.UI\_FP\_Payment\_Split @LINK = @LINK, @N\_Type = 0

INSERT dbo.FD\_Payments

SELECT 'П-124', 1, '20190105', 220

SET @LINK = SCOPE\_IDENTITY()

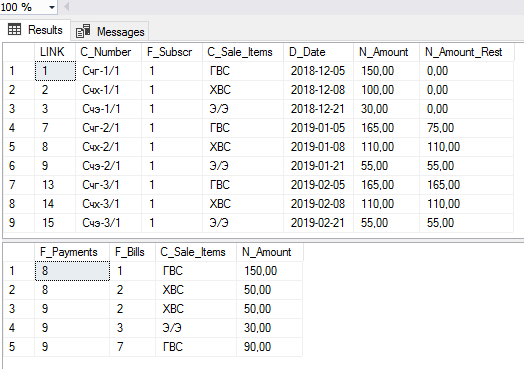
EXEC dbo.UI\_FP\_Payment\_Split @LINK = @LINK, @N\_Type = 0

SELECT \* FROM dbo.FD\_Bills WHERE F\_Subscr = 1

SELECT \* FROM dbo.FD\_Payment\_Details

ROLLBACK

**Ожидаемый результат:**



**Проверка №2**:

BEGIN TRAN

DECLARE @LINK INT

INSERT dbo.FD\_Payments

SELECT 'П-123', 1, '20190105', 200

SET @LINK = SCOPE\_IDENTITY()

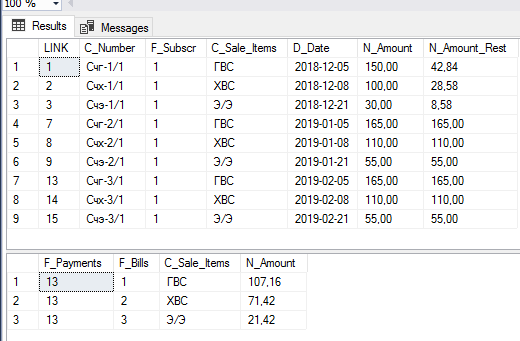
EXEC dbo.UI\_FP\_Payment\_Split @LINK = @LINK, @N\_Type = 0

SELECT \* FROM dbo.FD\_Bills WHERE F\_Subscr = 1

SELECT \* FROM dbo.FD\_Payment\_Details

ROLLBACK

**Ожидаемый результат:**



**Проверка №3**:

BEGIN TRAN

DECLARE @LINK INT

INSERT dbo.FD\_Payments

SELECT 'П-123', 1, '20190105', 200

SET @LINK = SCOPE\_IDENTITY()

EXEC dbo.UI\_FP\_Payment\_Split @LINK = @LINK, @N\_Type = 1

INSERT dbo.FD\_Payments

SELECT 'П-124', 1, '20190105', 220

SET @LINK = SCOPE\_IDENTITY()

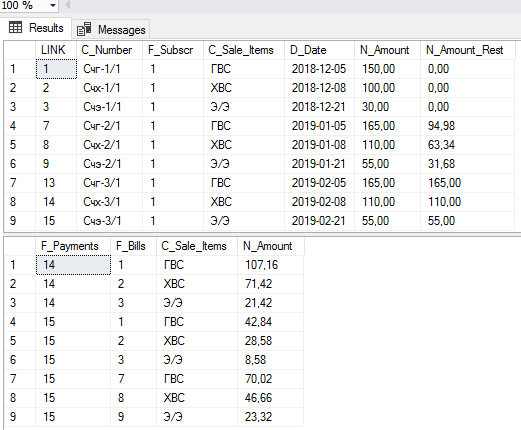
EXEC dbo.UI\_FP\_Payment\_Split @LINK = @LINK, @N\_Type = 1

SELECT \* FROM dbo.FD\_Bills WHERE F\_Subscr = 1

SELECT \* FROM dbo.FD\_Payment\_Details

ROLLBACK

**Ожидаемый результат:**



1. *Написать еще 3-и проверки, для хранимой процедуры dbo.UI\_FP\_Payment\_Split и объяснить почему выбрали именно эти тестовые случаи.*