

Лабораторная работа №3

Задание 1 (7 вариант)

а) Добавьте в таблицу **dbo.PersonPhone** поле **City** типа **nvarchar(30)**

```
/*
а) добавьте в таблицу dbo.PersonPhone поле City типа nvarchar(30);
*/
ALTER TABLE dbo.PersonPhone
    ADD City NVARCHAR(30);
GO
```

Результат выполнения:

The screenshot shows the SQL Server Enterprise Manager interface. On the left, the 'Object Explorer' pane displays the 'dbo.PersonPhone' table structure. The 'Columns' folder is expanded, and the 'City (nvarchar(30), null)' column is highlighted with a red rectangle. On the right, the 'SQL Server Enterprise Manager' window shows the execution of a T-SQL script. The script includes a comment in Russian, the 'ALTER TABLE' statement to add the 'City' column, and a 'GO' command. The 'Results' pane at the bottom indicates that the commands were completed successfully.

б) Объявите табличную переменную с такой же структурой как **dbo.PersonPhone** и заполните ее данными из **dbo.PersonPhone**. Поле **City** заполните значениями из таблицы **Person.Address** поля **City**, а поле **PostalCode** значениями из **Person.Address** поля **PostalCode**. Если поле **PostalCode** содержит буквы — заполните поле значением по умолчанию.

```
/*
б) объявите табличную переменную с такой же структурой как dbo.PersonPhone и заполните ее данными из dbo.PersonPhone.
Поле City заполните значениями из таблицы Person.Address поля City, а поле PostalCode
значениями из Person.Address поля PostalCode. Если поле PostalCode содержит буквы – заполните поле значением по умолчанию;
*/
DECLARE @PersonPhoneTableVar TABLE (
    BusinessEntityId INT NOT NULL,
    PhoneNumber NVARCHAR(25) NOT NULL,
    PhoneNumberTypeId BIGINT,
    ModifiedDate DATETIME NOT NULL,
    PostalCode NVARCHAR(15),
    City NVARCHAR(30)
);
```

```
-- Заполнение созданной табличной переменной данными из dbo.PersonPhone
INSERT INTO @PersonPhoneTableVar(
    BusinessEntityId,
    PhoneNumber,
    PhoneNumberTypeId,
    ModifiedDate,
    PostalCode,
    City
)
SELECT
    PersonPhone.BusinessEntityID,
    PersonPhone.PhoneNumber,
    PersonPhone.PhoneNumberTypeID,
    PersonPhone.ModifiedDate,
    CASE
        WHEN Address.PostalCode LIKE '[A-Za-z]%' THEN '0'
        ELSE Address.PostalCode
    END,
    Address.City
FROM dbo.PersonPhone AS PersonPhone
INNER JOIN Person.BusinessEntityAddress
ON PersonPhone.BusinessEntityID = Person.BusinessEntityAddress.BusinessEntityID
INNER JOIN Person.Address AS Address
ON Address.AddressID = Person.BusinessEntityAddress.AddressID;
```

Результат выполнения:

```
SELECT * from @PersonPhoneTableVar;
GO
```

83 %

	BusinessEntityId	PhoneNumber	PhoneNumberTypeId	ModifiedDate	PostalCode	City
1	4780	1 (11) 500 555-0175	1	2008-06-25 00:00:00.000	0	Oxford
2	4782	1 (11) 500 555-0131	1	2007-03-22 00:00:00.000	60323	Frankfurt
3	4784	1 (11) 500 555-0137	1	2007-09-21 00:00:00.000	2580	Goulburn
4	4785	149-555-0146	1	2007-08-19 00:00:00.000	0	N. Vancouver
5	4788	1 (11) 500 555-0174	1	2008-01-11 00:00:00.000	2060	Lavender Bay
6	4790	1 (11) 500 555-0144	1	2008-01-09 00:00:00.000	0	Peterborough
7	4791	1 (11) 500 555-0138	1	2007-05-14 00:00:00.000	0	London
8	4792	703-555-0112	1	2008-04-17 00:00:00.000	92118	Coronado
9	4793	921-555-0176	1	2005-10-13 00:00:00.000	90802	Long Beach
10	4796	1 (11) 500 555-0159	1	2007-12-04 00:00:00.000	75010	Paris
11	4798	1 (11) 500 555-0181	1	2006-12-12 00:00:00.000	2055	North Sydney

Query executed successfully. DESKTOP-L39DT9G (15)

с) Обновите данные в полях **PostalCode** и **City** в **dbo.PersonPhone** данными из табличной переменной.

Также обновите данные в поле **PhoneNumber**. Добавьте код '1 (11)' для тех телефонов, для которых этот код не указан;

Состояние таблицы **dbo.PersonPhone** до ее обновления:

Results		Messages				
	BusinessEntityID	PhoneNumber	PhoneNumberTypeID	ModifiedDate	PostalCode	City
1	1	697-555-0142	1	2003-02-08 00:00:00.000	0	NULL
2	3	212-555-0187	1	2001-12-05 00:00:00.000	0	NULL
3	4	612-555-0100	1	2001-12-29 00:00:00.000	0	NULL
4	5	849-555-0139	1	2002-01-30 00:00:00.000	0	NULL
5	8	815-555-0138	1	2003-01-23 00:00:00.000	0	NULL
6	9	185-555-0186	1	2003-02-10 00:00:00.000	0	NULL
7	11	719-555-0181	1	2004-12-29 00:00:00.000	0	NULL
8	14	465-555-0156	1	2005-01-23 00:00:00.000	0	NULL
9	15	970-555-0138	1	2005-02-11 00:00:00.000	0	NULL
10	17	150-555-0189	1	2001-02-19 00:00:00.000	0	NULL
11	19	124-555-0114	1	2005-03-10 00:00:00.000	0	NULL

Обновление данных в полях таблицы **dbo.PersonPhone**:

```

/*
с) обновите данные в полях PostalCode и City в dbo.PersonPhone данными из табличной переменной.
Также обновите данные в поле PhoneNumber. Добавьте код '1 (11)' для тех телефонов, для которых этот код не указан;
*/
UPDATE dbo.PersonPhone
SET
    dbo.PersonPhone.PostalCode = PersonPhoneTableVar.PostalCode,
    dbo.PersonPhone.City = PersonPhoneTableVar.City,
    dbo.PersonPhone.PhoneNumber =
    CASE
        WHEN PATINDEX('%1 (11)%', dbo.PersonPhone.PhoneNumber) = 0 THEN
            '1 (11)' + dbo.PersonPhone.PhoneNumber
        ELSE
            dbo.PersonPhone.PhoneNumber
    END
FROM dbo.PersonPhone
INNER JOIN @PersonPhoneTableVar AS PersonPhoneTableVar
ON dbo.PersonPhone.BusinessEntityID = PersonPhoneTableVar.BusinessEntityID;
GO

SELECT * FROM dbo.PersonPhone;

```

Результат обновления:

83 %

Results

Messages

	BusinessEntityID	PhoneNumber	PhoneNumberTypeID	ModifiedDate	PostalCode	City	
1	1	1 (11)697-555-0142	1	2003-02-08 00:00:00.000	98006	Newport Hills	
2	3	1 (11)212-555-0187	1	2001-12-05 00:00:00.000	98052	Redmond	
3	4	1 (11)612-555-0100	1	2001-12-29 00:00:00.000	55402	Minneapolis	
4	5	1 (11)849-555-0139	1	2002-01-30 00:00:00.000	98004	Bellevue	
5	8	1 (11)815-555-0138	1	2003-01-23 00:00:00.000	98201	Everett	
6	9	1 (11)185-555-0186	1	2003-02-10 00:00:00.000	98004	Bellevue	
7	11	1 (11)719-555-0181	1	2004-12-29 00:00:00.000	98028	Kenmore	
8	14	1 (11)465-555-0156	1	2005-01-23 00:00:00.000	98055	Renton	
9	15	1 (11)970-555-0138	1	2005-02-11 00:00:00.000	98055	Renton	
10	17	1 (11)150-555-0189	1	2001-02-19 00:00:00.000	98201	Everett	
11	19	1 (11)124-555-0114	1	2005-03-10 00:00:00.000	98004	Bellevue	

Query executed successfully.

DESKTOP-L39DT9G (15.0 RTM)

DESKTOP

	BusinessEntityID	PhoneNumber	PhoneNumberTypeID	ModifiedDate	PostalCode	City
2037	4777	1 (11) 500 555-01...	1	2007-08-23 00:00:00.000	92150	Suresnes
2038	4779	1 (11) 500 555-01...	1	2007-11-15 00:00:00.000	6105	Cloverdale
2039	4780	1 (11) 500 555-01...	1	2008-06-25 00:00:00.000	0	Oxford
2040	4782	1 (11) 500 555-01...	1	2007-03-22 00:00:00.000	60323	Frankfurt
2041	4784	1 (11) 500 555-01...	1	2007-09-21 00:00:00.000	2580	Goulburn
2042	4785	1 (11) 149-555-0146	1	2007-08-19 00:00:00.000	0	N. Vancouver
2043	4788	1 (11) 500 555-01...	1	2008-01-11 00:00:00.000	2060	Lavender Bay
2044	4790	1 (11) 500 555-01...	1	2008-01-09 00:00:00.000	0	Peterborou...
2045	4791	1 (11) 500 555-01...	1	2007-05-14 00:00:00.000	0	London
2046	4792	1 (11) 703-555-0112	1	2008-04-17 00:00:00.000	92118	Coronado
2047	4793	1 (11) 921-555-0176	1	2005-10-13 00:00:00.000	90802	Long Beach

Query executed successfully.

DESKTOP-L39DT9G (15.0 RTM) | DE

Посмотрев на скриншот с предыдущего задания для табличной переменной мы можем увидеть, что значения полей PostalCode и City были действительно обновлены значениями из табличной переменной:

```
SELECT * from @PersonPhoneTableVar;
GO
```

	BusinessEntityId	PhoneNumber	PhoneNumberTypeId	ModifiedDate	PostalCode	City
1	4780	1 (11) 500 555-0175	1	2008-06-25 00:00:00.000	0	Oxford
2	4782	1 (11) 500 555-0131	1	2007-03-22 00:00:00.000	60323	Frankfurt
3	4784	1 (11) 500 555-0137	1	2007-09-21 00:00:00.000	2580	Goulburn
4	4785	149-555-0146	1	2007-08-19 00:00:00.000	0	N. Vancouver
5	4788	1 (11) 500 555-0174	1	2008-01-11 00:00:00.000	2060	Lavender Bay
6	4790	1 (11) 500 555-0144	1	2008-01-09 00:00:00.000	0	Peterborough
7	4791	1 (11) 500 555-0138	1	2007-05-14 00:00:00.000	0	London
8	4792	703-555-0112	1	2008-04-17 00:00:00.000	92118	Coronado
9	4793	921-555-0176	1	2005-10-13 00:00:00.000	90802	Long Beach
10	4796	1 (11) 500 555-0159	1	2007-12-04 00:00:00.000	75010	Paris
11	4798	1 (11) 500 555-0181	1	2006-12-12 00:00:00.000	2055	North Sydney

Query executed successfully.

DESKTOP-L39DT9G (15

d) Удалите данные из **dbo.PersonPhone** для сотрудников компании, то есть где **PersonType** в **Person.Person** равен 'EM'.

```
/*
d) удалите данные из dbo.PersonPhone для сотрудников компании, то есть где PersonType в Person.Person равен 'EM';
*/
SELECT * from Person.Person;
SELECT * from dbo.PersonPhone;

DELETE FROM dbo.PersonPhone
WHERE EXISTS(
    SELECT BusinessEntityID, PersonType
    FROM Person.Person
    WHERE Person.BusinessEntityID = dbo.PersonPhone.BusinessEntityID
    AND Person.PersonType = 'EM'
);
GO
```

Содержимое таблицы dbo.PersonPhone до удаления данных:

	BusinessEntityID	PhoneNumber	PhoneNumberTypeID	ModifiedDate	PostalCode	City
1	1	1 (11)697-555-0142	1	2003-02-08 00:00:00.000	98006	Newport Hills
2	3	1 (11)212-555-0187	1	2001-12-05 00:00:00.000	98052	Redmond
3	4	1 (11)612-555-0100	1	2001-12-29 00:00:00.000	55402	Minneapolis
4	5	1 (11)849-555-0139	1	2002-01-30 00:00:00.000	98004	Bellevue
5	8	1 (11)815-555-0138	1	2003-01-23 00:00:00.000	98201	Everett
6	9	1 (11)185-555-0186	1	2003-02-10 00:00:00.000	98004	Bellevue
7	11	1 (11)719-555-0181	1	2004-12-29 00:00:00.000	98028	Kenmore
8	14	1 (11)465-555-0156	1	2005-01-23 00:00:00.000	98055	Renton
9	15	1 (11)970-555-0138	1	2005-02-11 00:00:00.000	98055	Renton
10	17	1 (11)150-555-0189	1	2001-02-19 00:00:00.000	98201	Everett
11	19	1 (11)124-555-0114	1	2005-03-10 00:00:00.000	98004	Bellevue

Query executed successfully. DESKTOP-L39DT9G (15.0

Содержимое таблицы Person.Person:

	BusinessEntityID	PersonType	NameStyle	Title	FirstName	MiddleName	LastName	Suffix	EmailPromotion	AdditionalContactInfo	Demogr
1	1	EM	0	NULL	Ken	J	Sánchez	NULL	0	NULL	<Individ
2	2	EM	0	NULL	Terri	Lee	Duffy	NULL	1	NULL	<Individ
3	3	EM	0	NULL	Roberto	NULL	Tamburello	NULL	0	NULL	<Individ
4	4	EM	0	NULL	Rob	NULL	Walters	NULL	0	NULL	<Individ
5	5	EM	0	Ms.	Gail	A	Erickson	NULL	0	NULL	<Individ
6	6	EM	0	Mr.	Jossef	H	Goldberg	NULL	0	NULL	<Individ
7	7	EM	0	NULL	Dylan	A	Miller	NULL	2	NULL	<Individ
8	8	EM	0	NULL	Diane	L	Margheim	NULL	0	NULL	<Individ
9	9	EM	0	NULL	Gigi	N	Matthew	NULL	0	NULL	<Individ
10	10	EM	0	NULL	Michael	NULL	Raheem	NULL	2	NULL	<Individ

Query executed successfully. DESKTOP-L39DT9G (15.0 RTM) DESKTOP-L39

	BusinessEntityID	PersonType	NameStyle	Title	FirstName	MiddleName	LastName	Suffix	EmailPromotion	AdditionalContactInfo	Der
268	268	EM	0	NULL	Ramesh	V	Meyyappan	NULL	1	NULL	<In
269	269	EM	0	NULL	Dan	K	Bacon	Jr.	0	NULL	<In
270	270	EM	0	NULL	François	P	Ajenstat	NULL	0	NULL	<In
271	271	EM	0	NULL	Dan	B	Wilson	NULL	0	NULL	<In
272	272	EM	0	NULL	Janaina	Barreiro Ga...	Bueno	NULL	0	NULL	<In
273	273	EM	0	Mr.	Brian	S	Welcker	NULL	0	NULL	<In
274	274	SP	0	NULL	Stephen	Y	Jiang	NULL	0	NULL	<In
275	275	SP	0	NULL	Michael	G	Blythe	NULL	1	NULL	<In
276	276	SP	0	NULL	Linda	C	Mitchell	NULL	0	NULL	<In
277	277	SP	0	NULL	Jillian	NULL	Carson	NULL	1	NULL	<In

Query executed successfully. DESKTOP-L39DT9G (15.0 RTM) DESKTOP-

Результат удаления данных из таблицы **dbo.PersonPhone**:

Results		Messages				
	BusinessEntityID	PhoneNumber	PhoneNumberTypeID	ModifiedDate	PostalCode	City
1	274	1 (11)238-555-0197	1	2005-01-28 00:00:00.000	98052	Redmond
2	275	1 (11)257-555-0154	1	2005-06-24 00:00:00.000	48226	Detroit
3	280	1 (11)340-555-0193	1	2005-06-24 00:00:00.000	97205	Portland
4	281	1 (11)330-555-0120	1	2005-06-24 00:00:00.000	94109	San Francisco
5	286	1 (11) 500 555-0190	1	2007-06-24 00:00:00.000	3000	Melbourne
6	288	1 (11) 500 555-0140	1	2007-06-24 00:00:00.000	14111	Berlin
7	290	1 (11) 500 555-0117	1	2006-06-24 00:00:00.000	33000	Bordeaux
8	291	398-555-0132	1	2009-05-16 16:33:33.060	0	NULL
9	293	747-555-0171	1	2009-05-16 16:33:33.077	0	NULL
10	297	599-555-0127	1	2009-05-16 16:33:33.090	0	NULL
11	299	1 (11) 500 555-0132	1	2009-05-16 16:33:33.090	0	NULL

Query executed successfully. DESKTOP-L39DT9G (15.0 R

е) Удалите поле **City** из таблицы, удалите все созданные ограничения и значения по умолчанию.

Состояние таблицы **dbo.PersonPhone** до удаления поля **City**:

dbo.PersonPhone
Columns
BusinessEntityID (PK, int, not null)
PhoneNumber (PK, nvarchar(25), not null)
PhoneNumberTypeID (bigint, null)
ModifiedDate (datetime, not null)
PostalCode (nvarchar(15), null)
City (nvarchar(30), null)
Keys
Constraints
CHK_PostalCode
DF_PersonPhone_PostalCode

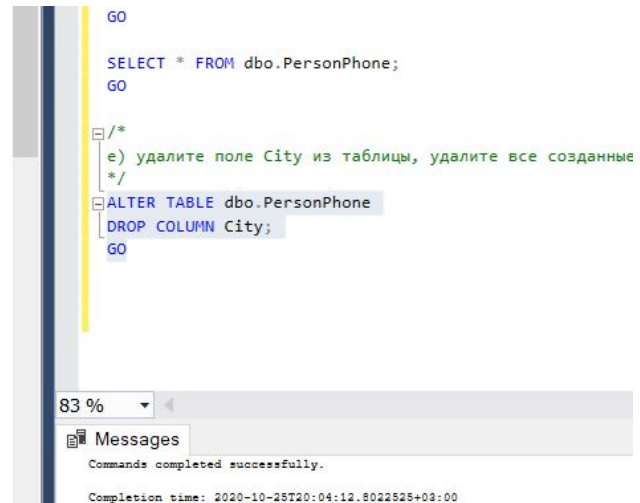
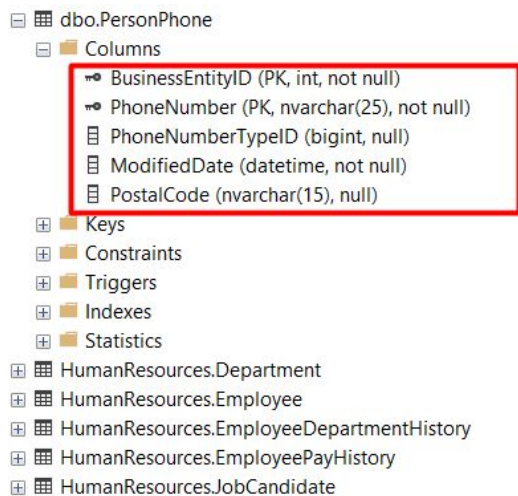
Удаление поля **City**:

```

/*
е) удалите поле City из таблицы, удалите все созданные ограничения и значения по умолчанию.
*/
ALTER TABLE dbo.PersonPhone
DROP COLUMN City;
GO

```


Результат удаления поля **City**:

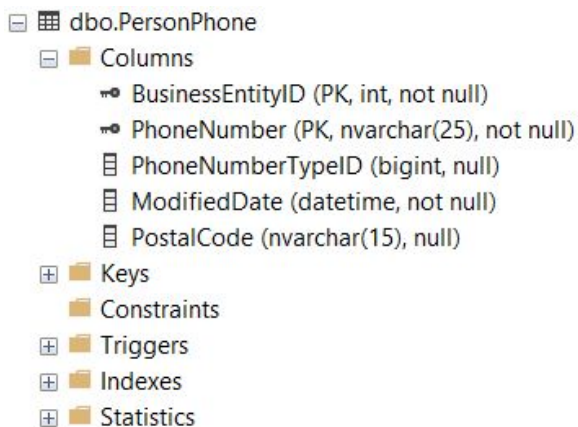


Удаление ограничений и значений по умолчанию:

```
ALTER TABLE dbo.PersonPhone
DROP CONSTRAINT CHK_PostalCode;
GO

ALTER TABLE dbo.PersonPhone
DROP CONSTRAINT DF_PersonPhone_PostalCode;
GO
```

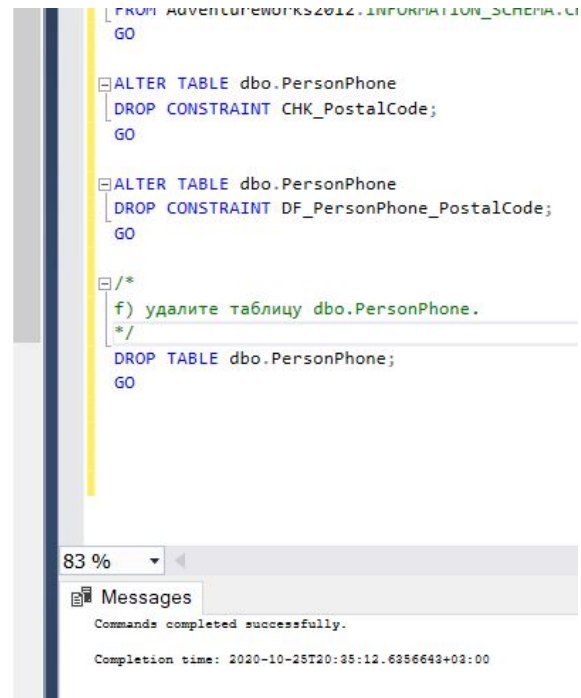
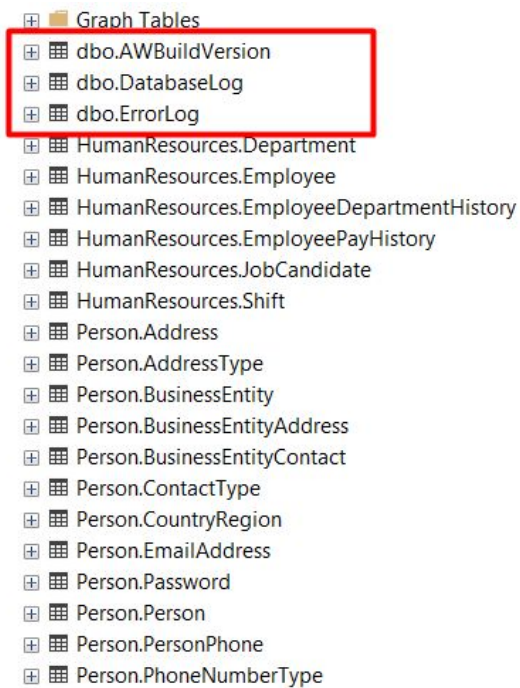
Результат выполнения:



f) Удалите таблицу **dbo.PersonPhone**.

```
/*
f) удалите таблицу dbo.PersonPhone.
*/
DROP TABLE dbo.PersonPhone;
GO
```

Результат выполнения:



Задание 2 (7 вариант)

а) Выполните код, созданный во втором задании второй лабораторной работы. Добавьте в таблицу **dbo.PersonPhone** поля **OrdersCount** **INT** и **CardType** **NVARCHAR(50)**. Также создайте в таблице вычисляемое поле **IsSuperior**, которое будет хранить 1, если тип карты **'SuperiorCard'** и 0 для остальных карт.

```

/*
а) Добавьте в таблицу dbo.PersonPhone поля OrdersCount INT и CardType NVARCHAR(50).
Также создайте в таблице вычисляемое поле IsSuperior, которое будет хранить 1, если тип карты 'SuperiorCard' и 0 для остальных карт.
*/
ALTER TABLE dbo.PersonPhone
ADD
    [OrdersCount] INT,
    [CardType] NVARCHAR(50),
    [IsSuperior] AS
    CASE
        WHEN [CardType] = 'SuperiorCard' THEN 1
        ELSE 0
    END;
GO

```


Состояние таблицы до выполнения кода:

dbo.PersonPhone
Columns
BusinessEntityID (PK, int, not null)
PhoneNumber (PK, nvarchar(25), not null)
PhoneNumberTypeID (bigint, null)
ModifiedDate (datetime, not null)
PostalCode (nvarchar(15), null)
Keys
Constraints
CHK_PostalCode
DF_PersonPhone_PostalCode
Triggers
Indexes
Statistics

Результат выполнения кода:

Tables

System Tables

FileTables

External Tables

Graph Tables

dbo.AWBUILDVersion

dbo.DatabaseLog

dbo.ErrorLog

dbo.PersonPhone

Columns

BusinessEntityID (PK, int, not null)

PhoneNumber (PK, nvarchar(25), not null)

PhoneNumberTypeID (bigint, null)

ModifiedDate (datetime, not null)

PostalCode (nvarchar(15), null)

OrdersCount (int, null)

CardType (nvarchar(50), null)

IsSuperior (Computed, int, not null)

Keys

Constraints

Triggers

```

/*
а) Добавьте в таблицу dbo.PersonPhone поля OrdersCount INT
Также создайте в таблице вычисляемое поле IsSuperior, котор
*/
ALTER TABLE dbo.PersonPhone
ADD
[OrdersCount] INT,
[CardType] NVARCHAR(50),
[IsSuperior] AS
CASE
WHEN [CardType] = 'SuperiorCard' THEN 1
ELSE 0
END;
GO

```

83 %

Messages

Commands completed successfully.

Completion time: 2020-10-28T21:52:55.1915920+03:00

Results Messages

	BusinessEntityID	PhoneNumber	PhoneNumberTypeID	ModifiedDate	PostalCode	OrdersCount	CardType	IsSuperior
1	1	697-555-0142	1	2003-02-08 00:00:00.000	0	NULL	NULL	0
2	3	212-555-0187	1	2001-12-05 00:00:00.000	0	NULL	NULL	0
3	4	612-555-0100	1	2001-12-29 00:00:00.000	0	NULL	NULL	0
4	5	849-555-0139	1	2002-01-30 00:00:00.000	0	NULL	NULL	0
5	8	815-555-0138	1	2003-01-23 00:00:00.000	0	NULL	NULL	0
6	9	185-555-0186	1	2003-02-10 00:00:00.000	0	NULL	NULL	0
7	11	719-555-0181	1	2004-12-29 00:00:00.000	0	NULL	NULL	0
8	14	465-555-0156	1	2005-01-23 00:00:00.000	0	NULL	NULL	0
9	15	970-555-0138	1	2005-02-11 00:00:00.000	0	NULL	NULL	0
10	17	150-555-0189	1	2001-02-19 00:00:00.000	0	NULL	NULL	0
11	19	124-555-0114	1	2005-03-10 00:00:00.000	0	NULL	NULL	0
12	21	138-555-0118	1	2003-03-27 00:00:00.000	0	NULL	NULL	0
13	27	632-555-0129	1	2002-03-23 00:00:00.000	0	NULL	NULL	0

б) Создайте временную таблицу **#PersonPhone**, с первичным ключом по полю **BusinessEntityID**. Временная таблица должна включать все поля таблицы **dbo.PersonPhone** за исключением поля **IsSuperior**.

```

/*
b) создайте временную таблицу #PersonPhone, с первичным ключом по полю BusinessEntityID.
Временная таблица должна включать все поля таблицы dbo.PersonPhone за исключением поля IsSuperior.
*/
CREATE TABLE #PersonPhone(
    [BusinessEntityId] INT NOT NULL,
    [PhoneNumber] NVARCHAR(25) NOT NULL,
    [PhoneNumberTypeId] BIGINT,
    [ModifiedDate] DATETIME NOT NULL,
    [PostalCode] NVARCHAR(15),
    [OrdersCount] INT,
    [CardType] NVARCHAR(50)
);
GO

ALTER TABLE #PersonPhone
    ADD CONSTRAINT PK_PersonPhones_BusinessEntityID
    PRIMARY KEY ([BusinessEntityID]);
GO

```

Результат выполнения:

Временная таблица должна включать все поля таблицы dbo.PersonPhone за исключением поля IsSuperior.

```

CREATE TABLE #PersonPhone(
    [BusinessEntityId] INT NOT NULL,
    [PhoneNumber] NVARCHAR(25) NOT NULL,
    [PhoneNumberTypeId] BIGINT,
    [ModifiedDate] DATETIME NOT NULL,
    [PostalCode] NVARCHAR(15),
    [OrdersCount] INT,
    [CardType] NVARCHAR(50)
);
GO

ALTER TABLE #PersonPhone
    ADD CONSTRAINT PK_PersonPhones_BusinessEntityID
    PRIMARY KEY ([BusinessEntityID]);
GO

SELECT * FROM #PersonPhone;
GO

```

82 %

Messages

Commands completed successfully.

Completion time: 2020-10-25T22:18:25.1705622+03:00

с) Заполните временную таблицу данными из **dbo.PersonPhone**. Поле **CardType** заполните данными из таблицы **Sales.CreditCard**. Посчитайте количество заказов, оплаченных каждой картой (**CreditCardID**) в таблице **Sales.SalesOrderHeader** и заполните этими значениями поле **OrdersCount**. Подсчет количества заказов осуществите в Common Table Expression (CTE).

Определение Common Table Expression (CTE) для подсчета количества заказов:

```

-- Определение Common Table Expression (CTE) для подсчета количества заказов
WITH SalesCount_CTE(CreditCardID, OrdersCount) AS
(
    SELECT SalesOrderHeader.CreditCardID, COUNT(SalesOrderHeader.CreditCardID) AS OrdersCount
    FROM Sales.SalesOrderHeader AS SalesOrderHeader
    GROUP BY SalesOrderHeader.CreditCardID
)

```

Заполнение временной таблицы необходимыми данными:

```
INSERT INTO #PersonPhone(
    BusinessEntityID,
    PhoneNumber,
    PhoneNumberTypeID,
    ModifiedDate,
    PostalCode,
    OrdersCount,
    CardType
)
SELECT
    PersonPhone.BusinessEntityID,
    PersonPhone.PhoneNumber,
    PersonPhone.PhoneNumberTypeID,
    PersonPhone.ModifiedDate,
    PersonPhone.PostalCode,
    SalesCount_CTE.OrdersCount,
    Sales.CreditCard.CardType
FROM dbo.PersonPhone AS PersonPhone
LEFT JOIN Sales.PersonCreditCard
    ON PersonPhone.BusinessEntityID = Sales.PersonCreditCard.BusinessEntityID
LEFT JOIN Sales.CreditCard
    ON Sales.PersonCreditCard.CreditCardID = Sales.CreditCard.CreditCardID
LEFT JOIN SalesCount_CTE
    ON Sales.CreditCard.CreditCardID = SalesCount_CTE.CreditCardID;
GO
```

Результат выполнения выборки данных из временной таблицы:

```
SELECT * FROM #PersonPhone;
GO
```

	BusinessEntityId	PhoneNumber	PhoneNumberTypeId	ModifiedDate	PostalCode	OrdersCount	CardType
1	3730	409-555-0193	1	2006-05-09 00:00:00.000	0	2	Vista
2	3731	947-555-0187	1	2007-12-29 00:00:00.000	0	1	ColonialVoice
3	3734	366-555-0119	1	2007-06-19 00:00:00.000	0	2	ColonialVoice
4	3735	1 (11) 500 555-0164	1	2007-10-14 00:00:00.000	0	1	ColonialVoice
5	3739	1 (11) 500 555-0133	1	2006-09-17 00:00:00.000	0	2	Distinguish
6	3740	124-555-0115	1	2008-02-29 00:00:00.000	0	1	Vista
7	3741	597-555-0116	1	2007-11-30 00:00:00.000	0	1	ColonialVoice
8	3742	364-555-0152	1	2008-01-30 00:00:00.000	0	1	SuperiorCard
9	3744	624-555-0184	1	2008-06-22 00:00:00.000	0	1	Vista

Query executed successfully. DESKTOP-L39DT9G (15.0 RTM) DESKTOP-L39DT9G\marta ...

Проверим полученные данные для **BusinessEntityId** = 3730.

Исходя из выборки значений из таблицы **dbo.PersonPhone**, видим, что данные для полей **BusinessEntityID**, **PhoneNumber**, **PhoneNumberTypeID**, **ModifiedDate**, **PostalCode** выбраны корректно:

```
SELECT * FROM dbo.PersonPhone
WHERE BusinessEntityID = 3730;
GO
```

	BusinessEntityID	PhoneNumber	PhoneNumberTypeID	ModifiedDate	PostalCode	OrdersCount	CardType	IsSuperior
1	3730	409-555-0193	1	2006-05-09 00:00:00.000	0	NULL	NULL	0

Получим **CreditCardID** по **BusinessEntityID**:

```
SELECT * FROM Sales.PersonCreditCard
WHERE BusinessEntityID = 3730;
GO
```

	BusinessEntityID	CreditCardID	ModifiedDate
1	3730	3092	2006-05-09 00:00:00.000

Выведем все заказы, оплаченные картой **CreditCardID = 3092** (как мы можем увидеть их два, как и в нашей выборке с временной таблицы):

```
SELECT * FROM Sales.SalesOrderHeader
WHERE CreditCardID = 3092;
GO
```

	OrderID	SalesPersonID	TerritoryID	BillToAddressID	ShipToAddressID	ShipMethodID	CreditCardID	CreditCardApprovalCode	CurrencyRateID	SubTotal	TaxAmt	Freight	TotalDue	Com
1	36	NULL	4	12759	12759	1	3092	532306V16271	NULL	3578.27	286.2616	89.4568	3953.9884	NUL
2	36	NULL	4	12759	12759	1	3092	537591V16271	NULL	2341.97	187.3576	58.5493	2587.8769	NUL

Теперь получим тип карточки по её **CreditCardID** для того, чтобы убедиться, что он совпадает с полученным нами во временной таблице:

```
SELECT * FROM Sales.CreditCard
WHERE CreditCardID = 3092;
GO
```

	CreditCardID	CardType	CardNumber	ExpMonth	ExpYear	ModifiedDate
1	3092	Vista	11111175812840	5	2005	2006-05-09 00:00:00.000

d) Удалите из таблицы **dbo.PersonPhone** одну строку (где **BusinessEntityID = 297**)

Вывод содержимого данной строки:

```
SELECT * FROM dbo.PersonPhone
WHERE BusinessEntityID = 297;
GO
```

	BusinessEntityID	PhoneNumber	PhoneNumberTypeID	ModifiedDate	PostalCode	OrdersCount	CardType	IsSuperior
1	297	599-555-0127	1	2009-05-16 16:33:33.090	0	NULL	NULL	0

Результат удаления:

```
DELETE FROM dbo.PersonPhone
WHERE BusinessEntityID = 297;
GO
```

82 %

Messages

(1 row affected)

Completion time: 2020-10-26T22:14:06.8321863+03:00

```
*/
SELECT * FROM dbo.PersonPhone
WHERE BusinessEntityID = 297;
GO
```

```
DELETE FROM dbo.PersonPhone
```

82 %

Results Messages

BusinessEntityID	PhoneNumber	PhoneNumberTypeID	ModifiedDate	PostalCode	OrdersCount	CardType	IsSuperior
------------------	-------------	-------------------	--------------	------------	-------------	----------	------------

е) Напишите **Merge** выражение, использующее **dbo.PersonPhone** как **target**, а временную таблицу как **source**. Для связи **target** и **source** используйте **BusinessEntityID**. Обновите поля **OrdersCount** и **CardType**, если запись присутствует в **source** и **target**. Если строка присутствует во временной таблице, но не существует в **target**, добавьте строку в **dbo.PersonPhone**. Если в **dbo.PersonPhone** присутствует такая строка, которой не существует во временной таблице, удалите строку из **dbo.PersonPhone**.

Для того, чтобы проверить корректно ли написано **Merge** выражение, добавим в таблицу **dbo.PersonPhone** запись, которая впоследствии после выполнения выражения **Merge** должна быть удалена из нее:

```
INSERT INTO dbo.PersonPhone(
    [BusinessEntityID],
    [PhoneNumber],
    [ModifiedDate],
    [PostalCode]
)
VALUES (
    99999999,
    '111-111-111',
    CURRENT_TIMESTAMP,
    '220034'
);
GO
```

```
SELECT * FROM dbo.PersonPhone
WHERE [BusinessEntityID] = 99999999;
GO
```

82 %

Results Messages

	BusinessEntityID	PhoneNumber	PhoneNumberTypeID	ModifiedDate	PostalCode	OrdersCount	CardType	IsSuperior
1	99999999	111-111-111	NULL	2020-10-28 21:56:10.170	220034	NULL	NULL	0

Merge выражение:

```

MERGE INTO dbo.PersonPhone AS [target]
USING #PersonPhone AS [source]
ON [target].[BusinessEntityID] = [source].[BusinessEntityID]
WHEN MATCHED THEN
    UPDATE
        SET [OrdersCount] = [source].[OrdersCount],
            [CardType] = [source].[CardType]
WHEN NOT MATCHED BY TARGET THEN
    INSERT (
        [BusinessEntityID],
        [PhoneNumber],
        [PhoneNumberTypeID],
        [ModifiedDate],
        [PostalCode],
        [OrdersCount],
        [CardType]
    )
    VALUES (
        [source].[BusinessEntityID],
        [source].[PhoneNumber],
        [source].[PhoneNumberTypeID],
        [source].[ModifiedDate],
        [source].[PostalCode],
        [source].[OrdersCount],
        [source].[CardType]
    )
WHEN NOT MATCHED BY SOURCE THEN DELETE;
GO

```

82 %

Messages

(10096 rows affected)

Completion time: 2020-10-28T21:26:12.4394186+03:00

82 %

✓ Query executed successfully.

До выполнения операции Merge две таблицы отличались только строками с **BusinessEntityId = 297** (была удалена из **dbo.PersonPhone**, но присутствовала во временной таблице) и строкой с **BusinessEntityId = 99999999**, которая была добавлена в **dbo.PersonPhone** до выполнения выражения. Согласно заданию, строка с **BusinessEntityId = 297** должна быть добавлена в **dbo.PersonPhone** с временной таблицы и запись с **BusinessEntityId = 99999999** должна быть удалена из таблицы **dbo.PersonPhone**.

```

-- Проверка того, что после merge операции в таблице dbo.PersonPhone вновь появилась запись с BusinessEntityId = 297
-- и запись с BusinessEntityId = 99999999 удалена
SELECT * FROM dbo.PersonPhone
WHERE [BusinessEntityID] = 297 OR [BusinessEntityID] = 99999999;
GO

```

82 %

Results Messages

	BusinessEntityID	PhoneNumber	PhoneNumberTypeID	ModifiedDate	PostalCode	OrdersCount	CardType	IsSuperior
1	297	599-555-0127	1	2009-05-16 16:33:33.090	0	11	Vista	0