

AdventureWorks2012

Database Diagrams

Tables

System Tables

FileTables

dbo.Address

Columns

AddressID (int, null)

AddressLine1 (nvarchar(64), null)

AddressLine2 (nvarchar(64), not null)

City (nvarchar(32), null)

StateProvinceID (int, null)

PostalCode (nvarchar(16), null)

ModifiedDate (datetime, null)

ID (int, not null)

PersonName (varchar(100), null)

Keys

Constraints

/*

а) добавьте в таблицу dbo.Address поле PersonName
типа nvarchar размерностью 100 символов;

*/

ALTER TABLE dbo.Address ADD PersonName VARCHAR(100)

б) объявите табличную переменную с такой же структурой
как dbo.Address и заполните ее данными из dbo.Address,
где StateProvinceID = 77. Поле AddressLine2 заполните
значениями из CountryRegionCode таблицы Person.CountryRegion,
Name таблицы Person.StateProvince и City из Address. Разделите значения запятыми;

*/

```
DECLARE @TableVar TABLE(AddressID int NULL,
                          AddressLine1 nvarchar(64) NULL,
                          AddressLine2 nvarchar(64) NOT NULL,
                          City nvarchar(32) NULL,
                          StateProvinceID int NULL,
                          PostalCode nvarchar(15) NULL,
                          ModifiedData datetime NULL,
                          ID int identity(1,1) unique not null,
                          PersonName varchar(100) NULL);

INSERT INTO @TableVar
(AddressID,
 AddressLine1,
 AddressLine2,
 City,
 StateProvinceID,
 PostalCode,
 ModifiedData)
SELECT AddressID,
 AddressLine1,
 (SELECT CntrRgn.CountryRegionCode + ', ' + SttPrvnc.Name + ', ' + Addr.City
  FROM Person.CountryRegion AS CntrRgn
  JOIN Person.StateProvince AS SttPrvnc
  ON CntrRgn.CountryRegionCode = SttPrvnc.CountryRegionCode
  JOIN Person.Address AS PAddr
  ON SttPrvnc.StateProvinceID = PAddr.StateProvinceID
  WHERE PAddr.AddressID = Addr.AddressID),
 City,
 StateProvinceID,
 PostalCode,
 ModifiedDate
FROM dbo.Address as Addr
where StateProvinceID = 77;
SELECT * FROM @TableVar --проверка
```

	Address...	AddressLine1	AddressLine2	City	StateProvinc...	PostalCo...	ModifiedData	ID	PersonNa...
1	36	34 Waterloo Road	AU, Victoria, Melbourne	Melbourne	77	3000	2007-06-24 00:00:00.000	1	NULL
2	416	Level 100	AU, Victoria, South Melbourne	South Melbourne	77	3205	2007-08-01 00:00:00.000	2	NULL
3	418	Licensing Account	AU, Victoria, Seaford	Seaford	77	3198	2007-09-01 00:00:00.000	3	NULL
4	419	Level 99	AU, Victoria, Melbourne	Melbourne	77	3000	2007-09-01 00:00:00.000	4	NULL
5	433	Level 25	AU, Victoria, Melbourne	Melbourne	77	3000	2007-08-01 00:00:00.000	5	NULL
6	437	Level 992	AU, Victoria, Melbourne	Melbourne	77	3000	2007-07-01 00:00:00.000	6	NULL
7	442	45671 Queens ...	AU, Victoria, Melbourne	Melbourne	77	3000	2007-07-01 00:00:00.000	7	NULL
8	11437	3852 Northridge...	AU, Victoria, Geelong	Geelong	77	3220	2008-04-02 00:00:00.000	8	NULL
9	11441	8426 Easley Dr.	AU, Victoria, Melton	Melton	77	3337	2007-11-03 00:00:00.000	9	NULL
10	11463	2111 Freedom ...	AU, Victoria, Sunbury	Sunbury	77	3429	2005-12-05 00:00:00.000	1..	NULL
11	11474	2665 B Wildbro...	AU, Victoria, Warrnambool	Warrnambool	77	3280	2006-08-08 00:00:00.000	1..	NULL
12	11482	5636 Barcelona	AU, Victoria, Cranbourne	Cranbourne	77	3977	2006-04-03 00:00:00.000	1..	NULL
13	11493	793 Crawford St...	AU, Victoria, South Melbourne	South Melbourne	77	3205	2005-12-11 00:00:00.000	1..	NULL
14	11516	9549 Roxbury D...	AU, Victoria, Bendigo	Bendigo	77	3550	2007-04-15 00:00:00.000	1..	NULL
15	11517	593 Willow Pass...	AU, Victoria, Melbourne	Melbourne	77	3000	2007-06-14 00:00:00.000	1..	NULL

```

/*
с) обновите поле AddressLine2 в dbo.Address данными из
табличной переменной. Также обновите данные в поле PersonName
данными из Person.Person, соединив значения полей FirstName и LastName;
*/
UPDATE dbo.Address
SET AddressLine2 = tmp.AddressLine2 --почему ругается если использовать
-- @TableVar без псевдонима
FROM @TableVar AS tmp;

UPDATE dbo.Address
SET PersonName = PPrsn.FirstName + ' ' + PPrsn.LastName
FROM Person.Person AS PPrsn
JOIN Person.BusinessEntityAddress AS PBisness
ON PBisness.BusinessEntityID = PPrsn.BusinessEntityID
WHERE PBisness.AddressID = dbo.Address.AddressID;
/*
) %

```

Messages

(901 row(s) affected)

(1007 row(s) affected)

(1001 row(s) affected)

```

/*
d) удалите данные из dbo.Address, которые относятся к типу
'Main Office' из таблицы Person.AddressType;
*/
DELETE dbo.Address
FROM dbo.Address AS Addr
JOIN Person.BusinessEntityAddress AS PBisness
ON Addr.AddressID = PBisness.AddressID
JOIN Person.AddressType AS AddrT
ON PBisness.AddressTypeID = AddrT.AddressTypeID
WHERE AddrT.Name = 'Main Office';

```

100 %

Messages

(6 row(s) affected)

- System Databases
- AdventureWorks2012
 - Database Diagrams
 - Tables
 - System Tables
 - FileTables
 - dbo.Address
 - Columns
 - Keys
 - Constraints
 - Triggers
 - Indexes
 - Statistics
 - dbo.AWBuildVersion
 - dbo.DatabaseLog
 - dbo.ErrorLog

```

/*
e) удалите поле PersonName из таблицы, удалите все
созданные ограничения и значения по умолчанию;
*/
ALTER TABLE dbo.Address DROP COLUMN PersonName;
ALTER TABLE dbo.Address DROP CONSTRAINT Odd
ALTER TABLE dbo.Address DROP CONSTRAINT Unknown
ALTER TABLE dbo.Address DROP CONSTRAINT UQ_Address__3214EC26388ACA75

SELECT *
FROM AdventureWorks2012.INFORMATION_SCHEMA.CONSTRAINT_TABLE_USAGE
WHERE TABLE_SCHEMA = 'dbo' AND TABLE_NAME = 'Address';

```

100 %

Messages

Command(s) completed successfully.

- Databases
 - System Databases
 - AdventureWorks2012
 - Database Diagrams
 - Tables
 - System Tables
 - FileTables
 - dbo.AWBuildVersion
 - dbo.DatabaseLog
 - dbo.ErrorLog
 - HumanResources.Department
 - HumanResources.Employee
 - HumanResources.EmployeeDepartmentHistory

```

/*
f) удалите таблицу dbo.Address.
*/
drop table dbo.Address

```

100 %

Messages

Command(s) completed successfully.

- HumanResources.JobCandidate
- HumanResources.Shift
- Person.Address
 - Columns
 - AddressID (PK, int, not null)
 - AddressLine1 (nvarchar(60), not null)
 - AddressLine2 (nvarchar(60), null)
 - City (nvarchar(30), not null)
 - StateProvinceID (FK, int, not null)
 - PostalCode (nvarchar(15), not null)
 - SpatialLocation (geography, null)
 - rowguid (uniqueidentifier, not null)
 - ModifiedDate (datetime, not null)
 - AccountNumber (nvarchar(15), null)
 - MaxPrice (money, null)
 - AccountID (Computed, nvarchar(17), null)
 - Keys
 - Constraints
 - Triggers

```

/*
a) выполните код, созданный во втором задании второй лабораторной работы.
Добавьте в таблицу dbo.Address поля AccountNumber NVARCHAR(15) и MaxPrice MONEY.
Также создайте в таблице вычисляемое поле AccountID, которое будет добавлять к
значению в поле AccountNumber приставку 'ID'.
*/
--Добавляю в Person.Address тк в задании c) нельзя потом сдвоинить таблицы с dbo.Address
ALTER TABLE Person.Address
ADD AccountNumber NVARCHAR(15),
MaxPrice MONEY,
AccountID AS 'ID' + AccountNumber;

```

100 %

Messages

Command(s) completed successfully.

HumanResources.EmployeePayHistory
HumanResources.JobCandidate
HumanResources.Shift
Person.Address
Columns
AddressID (PK, int, not null)
AddressLine1 (nvarchar(60), not null)
AddressLine2 (nvarchar(60), null)
City (nvarchar(30), not null)
StateProvinceID (FK, int, not null)
PostalCode (nvarchar(15), not null)
SpatialLocation (geography, null)
rowguid (uniqueidentifier, not null)
ModifiedDate (datetime, not null)
AccountNumber (nvarchar(15), null)
MaxPrice (money, null)
AccountID (Computed, nvarchar(17), null)
Keys
Constraints
Triggers
Indexes
Statistics
Person.AddressType
Person.BusinessEntity
Person.BusinessEntityAddress

```

b) создайте временную таблицу #Address, с первичным ключом по полю ID.
Временная таблица должна включать все поля таблицы dbo.Address за исключением поля AccountID

/*
--Делаю идентичную Person.Address тк в задании с) нельзя потом сдвойнить таблицы с dbo.Address
CREATE TABLE #Address(
    ID INT IDENTITY(1, 1) PRIMARY KEY,
    AddressID INT NOT NULL,
    AddressLine1 NVARCHAR(60) NOT NULL,
    AddressLine2 NVARCHAR(60),
    City NVARCHAR(30),
    StateProvinceID INT NOT NULL,
    PostalCode NVARCHAR(16),
    SpatialLocation GEOGRAPHY,
    rowguid UNIQUEIDENTIFIER NOT NULL,
    ModifiedDate DATETIME,
    AccountNumber NVARCHAR(15),
    MaxPrice MONEY;
)
SELECT * FROM #Address --проверка что всё создалось

/*
c) заполните временную таблицу данными из dbo.Address.

```

Results
Spatial results
Messages

ID	Address...	AddressLine...	AddressLine...	City	StateProvinc...	PostalCo...	SpatialLocati...	rowg...	ModifiedD...	AccountNum...	MaxPrice
----	------------	----------------	----------------	------	-----------------	-------------	------------------	---------	--------------	---------------	----------

```

/*
c) заполните временную таблицу данными из dbo.Address.
Поле AccountNumber заполните данными из таблицы Purchasing.Vendor.
Определите максимальную цену продукта (StandardPrice),
поставляемого каждым поставщиком (BusinessEntityID) в таблице Purchasing.ProductVendor
и заполните этими значениями поле MaxPrice. Подсчет максимальной цены осуществите в Common Table Expression (CTE).
*/
--Заполняю данными из Person.Address
WITH MaxPriceCTE
AS (
SELECT      BusinessEntityID,
            MAX(StandardPrice) AS MaxPrice
FROM        Purchasing.ProductVendor
GROUP BY    BusinessEntityID
)

```

```

INSERT INTO #Address (
    AddressID,
    AddressLine1,
    AddressLine2,
    City,
    StateProvinceID,
    PostalCode,
    SpatialLocation,
    rowguid,
    ModifiedDate,
    AccountNumber,
    MaxPrice
)
SELECT      PA.AddressID,
            AddressLine1,
            AddressLine2,
            City,
            StateProvinceID,
            PostalCode,
            SpatialLocation,
            PA.rowguid,
            PA.ModifiedDate,
            PV.AccountNumber,
            MPCTE.MaxPrice
FROM        Person.Address AS PA
JOIN        Person.BusinessEntityAddress AS PBEA
ON          PBEA.AddressID = PA.AddressID
JOIN        Purchasing.Vendor AS PV
ON          PV.BusinessEntityID = PBEA.BusinessEntityID
JOIN        MaxPriceCTE AS MPCTE
ON          MPCTE.BusinessEntityID = PBEA.BusinessEntityID
SELECT * FROM #Address --проверка что всё создалось

```

Results
Spatial results
Messages

ID	Address...	AddressLine1	AddressLine2	City	StateProvinc...	PostalCo...	SpatialLocation	rowguid	
7	8	345	6441 Co Road	NULL	Lemon G...	6	85252	0xE6100000010CD42F158283BA40402EBA15288B045C...	CC52592C...
8	9	324	50 Via Del Sol	NULL	Lynnwood	79	98036	0xE6100000010C6D8B22A02BEE474010B051BF2B9C5...	89E1D63E...
9	1..	357	683 Larch Ct.	NULL	Salt Lak...	74	84101	0xE6100000010C9683D5A842594440C7485C1A92F95B...	B870F6AC...
10	1..	396	6 Dancing Road	NULL	Burien	79	98168	0xE6100000010C82611C8E6CB447406B2E9982EE925E...	CC2EC84...

```

/*
    d) удалите из таблицы dbo.Address одну строку (где ID = 293)
*/
DELETE FROM dbo.Address WHERE ID = 293;

```

91 %

Messages

(1 row(s) affected)

```

/*
    e) напишите Merge выражение, использующее dbo.Address как target,
    а временную таблицу как source. Для связи target и source используйте ID.
    Обновите поля AccountNumber и MaxPrice, если запись присутствует в source и target.
    Если строка присутствует во временной таблице, но не существует в target,
    добавьте строку в dbo.Address. Если в dbo.Address присутствует такая строка,
    которой не существует во временной таблице, удалите строку из dbo.Address.
*/

```

```

MERGE dbo.Address AS TARGET
USING #Address AS SOURCE
ON TARGET.ID = SOURCE.ID
WHEN MATCHED
THEN UPDATE
SET AccountNumber = source.AccountNumber,
    MaxPrice = source.MaxPrice
WHEN NOT MATCHED BY TARGET
THEN INSERT (
    AddressID,
    AddressLine1,
    AddressLine2,
    City,
    StateProvinceID,
    PostalCode,
    ModifiedDate,
    AccountNumber,
    MaxPrice)
VALUES (
    source.AddressID,
    source.AddressLine1,
    source.AddressLine2,
    source.City,
    source.StateProvinceID,
    source.PostalCode,
    source.ModifiedDate,
    source.AccountNumber,
    source.MaxPrice)
WHEN NOT MATCHED BY SOURCE
THEN DELETE;
SELECT * FROM dbo.Address --проверка

```

5 %

Results Messages

	Address...	AddressLine1	AddressLin...	City	StateProvinc...	PostalCo...	ModifiedDate	ID	AccountNumber	MaxPrice	AccountID
1	36	34 Waterloo Road	Unknown	Melbourne	77	3000	2007-06-24 00:00:00.000	1	AUSTRALI0001	47,33	IDAUSTRA
2	416	Level 100	Unknown	South Melbourne	77	3205	2007-08-01 00:00:00.000	2	ALLENSON0001	15,32	IDALLENS