



" "

**U13469**

-

2008

-

« »

-

« »

2008



⋮  
/  
« », 2008. 32 .

220300,

Transact-SQL.

1

:  
SELECT  
( ) Library.

1. I - SQL Server Management Studio Library.
2. , title title\_no
3. title.
3. , title no = 10.
4. loanhist, -  
( fine\_assessed) ,  
8.00 9.00 \$.
5. title  
Charles Dickens, Jane Austen.
6. , title title no title ,  
«Adventures».
7. , NULL  
fine\_paid. loanhist -  
2 -

DISTINCT:

Adult;

2) title.

2. ;

1) loanhist

: member\_no, isbn fine\_assessed

fineassessed, NULL;

2) fine assessed.

'double fine'.

3.

:

1) ———, firstname, middleinitial

lastname member,

Anderson;

2) email\_name

;

3) :

SUBSTRING

LOWER

,

4. :

1) title title no

title;

2) :

'The title is: Poems, title number 7'.

:

- The title is: - ;
- title.title - ;
- title number - ;
- title.title\_no - .

3) CONVERT ti-

tle.title\_no .



3 -

:

-

sp\_who

.

2.

:

SELECT @@spid

3.

,

,

:

EXEC sp\_who

4.

:

SELECT @@version

5.

:

SELECT USER\_NAME(), DB\_NAME(), @@servername

6.

-

:

USE library

SELECT \*

FROM information\_schema.tables

WHERE table\_type = 'base table'

2

:

GROUP BY HAVING,

AdventureWorks.

1 -

SELECT.

SELECT

WITH TIES

1.

Sales.Salesperson

SalesPersonID

Bonus.

Bonus

.

2.

,

4

( )

.

5



3 -

:

-

sp\_who

.

2.

:

SELECT @@spid

3.

,

,

:

EXEC sp\_who

4.

:

SELECT @@version

5.

:

SELECT USER\_NAME(), DB\_NAME(), @@servername

6.

-

:

USE library

SELECT \*

FROM information\_schema.tables

WHERE table\_type = 'base table'

2

:

GROUP BY HAVING,

AdventureWorks.

1 -

SELECT.

SELECT

WITH TIES

1.

Sales.Salesperson

SalesPersonID

Bonus.

Bonus

.

2.

,

4

( )

.

5

5

2 —

GROUP BY HAVING.

1. :  
1) Employee  
HumanResources;  
2) ,  
( ManagerID).

2. GROUP BY:

- 1) Sales.SalesOrderDetail,  
( OrderQty)  
( ProductID);
  - 2) -
  - 3) ;
- .2000.

3. GROUP BY

- 1) :  
Sales.SalesOrderDetail, SELECT  
ProductID, SpecialOfferID,  
( UnitPrice)  
LineTotal. ;
- 2) ProductID -  
SELECT, -
- 3) ;

3 -

ROLLUP CUBE:

1. ROLLUP :



```

Sales.SalesPerson.      SELECT      -
SalesQuota              SalesYTD.
                          ;
TotalSalesYTD
2)                      ,
                          GROUPING;
3)                      .
                          NULL.      NULL?
SalesQuota TotalSalesYTD

NULL      1533087,5999
250000,00      33461260,59
300000,00      9299677,9445
NULL      44294026,1344
(4 row(s) affected)

2.      CUBE      :
1)      Sales.SalesOrderDetail.      SELECT
      ProductID      LineTotal.
      ,      UnitPrice < $5.00.
      ProductID;
2)      ,      CUBE,      -
      OrderQty;
3)      :
ProductID Total

NULL 86579.210714
923      7425.120000
923      7425.120000
(119 row(s) affected)
      4 -      COMPUTE      COMPUTE
BY      SELECT      :
1.      Sales.SalesOrderHeader.      SELECT
      SalesPersonID, CustomerID, OrderDate, SubTotal, TotalDue.
2.      SalesPersonID, OrderDate.

```



COMPUTE  
SubTotal TotalDue.

3

I —  
“JOIN”. \*

1. SQL Server Management Studio Library.
2. member adult,  
firstname, middleinitial, lastname, street, city, state zip.  
firstname, iniddleinitial lastname  
name.

3. , , -

name	street	city	state	zip
Amy A Anderson	Bowery Estates	Montgomery	AL	36100
Brian A Anderson	Dogwood Drive	Sacramento	CA	94203
Daniel A Anderson	Fir Street	Washington	DC	20510
Eva A Anderson	The Highlands	Atlanta	GA	30026
Gary A Anderson	James Road	Springfield	IL	62700

5000 row(s) affected

2 -

title, item , -

isbn, copy\_no, on\_loan, title, translation cover,

ISBN 1, 500 1000.

isbn.

1. SELECT.



COMPUTE  
SubTotal TotalDue.

3

I —  
“JOIN”. \*

1. SQL Server Management Studio Library.
2. member adult,  
firstname, middleinitial, lastname, street, city, state zip.  
firstname, iniddleinitial lastname  
name.

3. , , -

name	street	city	state	zip
Amy A Anderson	Bowery Estates	Montgomery	AL	36100
Brian A Anderson	Dogwood Drive	Sacramento	CA	94203
Daniel A Anderson	Fir Street	Washington	DC	20510
Eva A Anderson	The Highlands	Atlanta	GA	30026
Gary A Anderson	James Road	Springfield	IL	62700

5000 row(s) affected

2 -

title, item , -

isbn, copy\_no, on\_loan, title, translation cover,

ISBN 1, 500 1000.

isbn.

1. SELECT.

FROM,

INNER JOIN

title isbn.  
INNER JOIN,  
item isbn.

3.

4.

WHERE ,

: ISBN ( ) 1,  
500 1000.

5.

ORDER BY ISBN -

3 - OUTER JOIN.

member\_no member, isbn log date reservation -  
250, 341 1675. memberno.  
,

1.

SELECT:

- name lastname,  
firstname middleinitial;
- date, log date char(8).

2.

FROM, OUTER JOIN

member reservation.

3.

WHERE: member

250, 341 1675.

4.

CONVERT.

5.

member.member\_no.

4 - UNION -  
:

1.

, ,  
:

1)

SELECT, member\_no

. -  
( , numkids). -

member no adult;

2)

1) CTRL+N

2)

3)

4)

1) CTRL+N;

2)

3)

4)

UNION

4

Library.

I -

INSERT.

INSERT

Library.

1. item,

1) SQL Server Management Studio;

2) item 8,

The Cherry Orchard.

	isbn	Titleno	Cover	Loanable	Translation
1	10001	8	HARDBACK	Y	ENGLISH
2	10101	8	SOFTBACK	Y	ENGLISH

1) CTRL+N

2)

3)

4)

1) CTRL+N;

2)

3)

4)

UNION

4

Library.

I -

INSERT.

INSERT

Library.

1. item,

1) SQL Server Management Studio;

2) item 8,

The Cherry Orchard.

	isbn	Titleno	Cover	Loanable	Translation
1	10001	8	HARDBACK	Y	ENGLISH
2	10101	8	SOFTBACK	Y	ENGLISH

item, -

2. ,  
:  
1) ,

: Isbn 10001, copy no 1, title no 8, Onloan N;

2)  
3. , -  
:

1) , ( translation)

, ;

2) -

2 — INSERT

DEFAULT:

1. , null.  
sp\_help ,  
title null.  
default -  
IDENTITY

2. title. title  
*The Art of Lawn Tennis* William T. Tilden.  
DEFAULT , null  
default. title no,  
IDENTITY.

3. IDENTITY'.  
title no , -

4. title.  
title.

	title.	title
	<i>Riders of the Purple Sage</i>	Zane Grey.
		,
6.	default.	title.
		-
		.
	3 —	INSERT
	DEFAULT VALUES.	-
1.	sample 1.	
	sample 1	library:
	USE LIBRARY	
	CREATE TABLE sample 1 (	
	Custjd int NOT NULL IDENTITY( 100,5)	
	.Name char( 10) NULL )	
2.	co	sample 1:
1)	,	sample 1.
		-
		DEFAULT VALUES;
2)	,	.
	4 -	DELETE:
1.	,	item.
	, isbn 10101,	: — «The Cherry Orchard», title no 8.
2.	.	
	5 -	UPDATE:
1.	,	.
		507 member.
2.	.	,
	.	
	6 -	
	:	
1.	.	-
	.	SET IDENTITY INSERT



## IDENTITY.

USE library

BEGIN TRANSACTION

SET IDENTITY INSERT member ON

INSERT member (member\_no, lastname, firstname, middleinitial)

VALUES (16101, 'Walters', 'B.', 'L')

SET IDENTITY INSERT member OFF

INSERT juvenile

VALUES ( 16101, 1, DATEADD(YY, 8, DATEADD(DD, -1,

GETDATEO)))

COMMIT TRANSACTION

2. , juvenile

adult:

1) , member no -

juvenile street, city, state, zip phone no adult.

1 . -

: DATEADD (YY, 1, GETDATEO);

2) juvenile adult member: ju-

venile.adultmemberno = adult.memberno;

3) WHERE,

juvenile, 18 .

DATE ADD.

3. juvenile adult:

1) , adult;

2) ,

16101 adult.

4. juvenile.

, juvenile adult

: juvenile.member\_no = adult.member\_no.

5. juvenile:

1) , ;

2) , 16101 juvenile.

;

I -

Library.

1. , -  
juve-  
nile.adult\_member\_no -  
( )  
( ). SELECT -  
adult\_member\_no No\_Of\_Child-  
ren exprdate adult.

2. ,  
,  
,  
adult member no,  
juvenile 3.

expr date adult.  
JOIN.

2 -

1. (single-value). -  
mem-  
ber.firstname, member, lastname, loanhist.isbn loanhist.fine\_paid -  
:

1)  
loanhist.fine\_paid;

2) .

```

1)      ,
      firstname, lastname, isbn  fine_paid
2)      1
      WHERE,
      ;
3)      DISTINCT;
4)      :
      Firstname lastname isbn fine_paid
      Michael Nash 883 8.0000
      Robert Rothenberg 330 8.0000
      (2 row(s) affected)

```

```

3.      .
      title, loan  reservation,
      4      : title no, title, isbn  Total Reserved.  Total Reserved
      .
      , 50, 5.

```

```

1)      , isbn  reservations
      , 50 ;
2)      .
4.      -
      , title no, title, isbn  Total Reserved,
      Total Reserved -
      . :

```

```

1)      , -
      5;
2)      IN, WHERE
      , 1
      3 -
      ,
      , -
      , 5 .

```



member\_no

member.

2. , loanhist.

3. member.member\_no loan-  
hist.member\_no .

4. WHERE.

5. .

6

AdventureWorks.

1 -

:

1. HumanResources.JobCandidateHistory.

:

- JobCandidateID. int.

;

- Resume. xml,

;

- Rating. int,

1... 10, - 5;

- RejectedDate. datetime,

;

- ContactID. int,

ContactID Person.Contact.

```

1.      member_no
      member.
2.
      loanhist.
3.      member.member_no      loan-
      hist.member_no
4.      WHERE.
5.

```

## 6

```

      AdventureWorks.

1.      HumanResources.JobCandidateHistory.
      :
      • JobCandidateID.      int.
      ;
      • Resume.      xml,
      ;
      • Rating.      int,
      1... 10,      - 5;
      • RejectedDate.      datetime,
      ;
      • ContactID.      int,
      ContactID      Person.Contact.

```

AdventureWorks, Tables  
HumanResources.JobCandidateHistory.

3. JobCandidateHistory :
  - 1) File SQL Server Management Studio  
TestConstraints.sql, C:\Labfiles;
  - 2) , This should fail,  
Execute. INSERT  
, Rating CHECK;
  - 3) This should succeed -  
Execute. INSERT .  
Microsoft® SQL Server™ Management Studio -

2 - :

1. :  
USE [AdventureWorks]  
GO  
ALTER TABLE HumanResources. JobCandidateHistory  
NOCHECK CONSTRAINT [CK\_JobCandidateHistory\_Rating]  
GO  
ALTER TABLE HumanResources. JobCandidateHistory  
NOCHECK CONSTRAINT [DF\_JobCandidateHistory\_Rating]  
GO
2. InsertTestData.sql C:\Labfiles.
3. HumanResources.JobCandidateHistory ,  
:  
ALTER TABLE HumanResources.JobCandidateHistory  
CHECK CONSTRAINT [CKJobCandidateHistoryRating]  
GO  
ALTER TABLE HumanResources. JobCandidateHistory  
CHECK CONSTRAINT [DF JobCandidateHistory Rating]  
GO
4. ,

AdventureWorks.

1 —

HumanResources.vEmployeeDetails,  
SELECT:

```
SELECT
e.[EmployeeID]
, . [Title]
,c.[FirstName]
,c.[MiddleName]
,c.[LastName]
,c. [Suffix]
,e.[Title] AS [JobTitle]
,c.[Phone]
,c. [Email Address]
,c. [Email Promotion]
,a.[AddressLine1]
,a. [ AddressLine2]
,a.[City]
,sp.[Name] AS [StateProvinceName]
,a.[PostalCode]
,cr.[Name] AS [Country RegionName]
,c. [ AdditionalContactInfo]
FROM [HumanResources].[Employee] e
INNER JOIN [Person].[Contact] c
ON c.[ContactID] = e.[ContactID]
INNER JOIN [HumanResources].[EmployeeAddress] ea
```

ON e.[EmployeeID] = ea.[EmployeeID]

INNER JOIN [Person].[Address] a

ON ea.[AddressID] = a.[AddressID]

INNER JOIN [Person].[StateProvince] sp

ON sp.[StateProvinceLD] = a.[StateProvinceID]

INNER JOIN [Person].[CountryRegion] cr

ON cr.[CountryRegionCode] = sp.[CountryRegionCode]

SCHEMABINDING.

2 —

:

USE AdventureWorks

GO

CREATE UNIQUE CLUSTERED INDEX IX\_vEmployeeDetails

ON HumanResources.vEmployeeDetails (EmployeeID)

3 —

AdventureWorks

sp\_depends.

sp\_helptext.

8

AdventureWorks.



ON e.[EmployeeID] = ea.[EmployeeID]

INNER JOIN [Person].[Address] a

ON ea.[AddressID] = a.[AddressID]

INNER JOIN [Person].[StateProvince] sp

ON sp.[StateProvinceLD] = a.[StateProvinceID]

INNER JOIN [Person].[CountryRegion] cr

ON cr.[CountryRegionCode] = sp.[CountryRegionCode]

SCHEMABINDING.

2 —

:

USE AdventureWorks

GO

CREATE UNIQUE CLUSTERED INDEX IX\_vEmployeeDetails

ON HumanResources.vEmployeeDetails (EmployeeID)

3 —

AdventureWorks

sp\_depends.

sp\_helptext.

8

:

AdventureWorks.

```

1 -
    GetDiscounts Sales,
    Sales. Specialoffer: Description, DiscountPct,
Type, Category, StartDate, EndDate, MinQty MaxQty.
    , StartDate EndDate.
    :

EXEC Sales.GetDiscounts

2 —
    GetDiscountsForCategory Sales.
    @Category,
nvarchar 50
    GetDiscounts,
    @Category.
    :
EXEC Sales.GetDiscountsForCategory 'Reseller'

3 -
    Sales
GetDiscountsForCategoryAndDate.
    @Category, GetDiscountsForCategory,
    @DateToCheck datetime.
    @DateToCheck
NULL. NULL @DateToCheck,
    ,
GETDATE().
EXEC Sales.GetDiscountsForCategoryAndDate 'Reseller'
    ,
DECLARE @DateToCheck datetime
SET @DateToCheck = DateAdd(month, 1, GetDate())
EXEC Sales.GetDiscountsForCategoryAndDate 'Reseller', @DateToCheck

4 -
    AddDiscount Sales,
    Sales.SpecialOffer.
    :

```

Description	nvarchar(255)
@DiscountPct	smallmoney
@Type	nvarchar(50)
@Category	nvarchar(50)
@StartDate	Datetime
@EndDate	Datetime
@MinQty	int
@MaxQty	int
@NewProductID	int (output)

INSERT

```

,
                                dbo.ErrorLog.
                                , @NewProductID
SCOPE_IDENTITY.
-

```

Try... Catch.

```

DECLARE @StartDate datetime, @EndDate datetime
SET @StartDate = GetDate()
SET @EndDate = DateAdd(month, 1, @StartDate)
DECLARE @NewId int
EXEC Sales. AddDiscount
'Half price off everything',
0.5,
'Seasonal Discount',
'Customer',
@StartDate,
@EndDate,
0,
20,
@NewID OUTPUT
SELECT @NewID

```

### Transact-SQL:

```

DECLARE @StartDate datetime, @EndDate datetime
SET @StartDate = GetDate()
SET @EndDate = DateAdd(month, 1, @StartDate)
DECLARE @NewId int, @ReturnValue int
EXEC @ReturnValue = Sales.AddDiscount
'Half price off everything',
-0.5,          - UNACCEPTABLE VALUE
'Seasonal Discount',
'Customer!',
@StartDate,
@EndDate,
0,
20,
@NewID OUTPUT
IF (@ReturnValue = 0)
SELECT @NewID
ELSE
SELECT TOP 1 * FROM dbo.ErrorLog ORDER BY ErrorTime DESC

```

ErrorProcedure, AddDiscount.

9

### UDF

:

I (UDF).  
I : AdventureWorks Adventure WorksDW.  
t InitializeData.sql

### Transact-SQL:

```

DECLARE @StartDate datetime, @EndDate datetime
SET @StartDate = GetDate()
SET @EndDate = DateAdd(month, 1, @StartDate)
DECLARE @NewId int, @ReturnValue int
EXEC @ReturnValue = Sales.AddDiscount
'Half price off everything',
-0.5,          - UNACCEPTABLE VALUE
'Seasonal Discount',
'Customer!',
@StartDate,
@EndDate,
0,
20,
@NewID OUTPUT
IF (@ReturnValue = 0)
SELECT @NewID
ELSE
SELECT TOP 1 * FROM dbo.ErrorLog ORDER BY ErrorTime DESC

```

ErrorProcedure, AddDiscount.

9

### UDF

:

I (UDF).  
I : AdventureWorks Adventure WorksDW.  
t InitializeData.sql

1 -

Sales.GetMaximumDiscountForCategory,  
( DiscountPct),

@Category nvarchar(50)

GETDATE()

StartDate EndDate.

SELECT Sales.GetMaximumDiscountForCategory('Reseller')

2 —

(In-Line Table-valued UDF).

Sales.GetDiscountsForDate ,

GetDiscounts (

. 8, . 1).

@DateToCheck

datetime.

Adventure Works ,

:

SELECT \*

FROM Sales.GetDiscountsForDate(GetDate())

ORDER BY DiscountPct DESC

3 -

(Multi-Statement Table-valued UDF).

GetDiscountedProducts

Sales.

: Sales.SpecialOfferProduct, Sales.SpecialOffer Production.Product.

: ProductID, Name,

ListPrice Production.Product,

Description DiscountPct

Sales.SpecialOffer,

ListPrice DiscountPct;

ListPrice

ListPrice DiscountPct.

@IncludeHistory bit,

ProductID	int
Name	nvarchar(50)
ListPrice	money
DiscountDescription	nvarchar(255)
Discountpercentage	smallmoney
DiscountAmount	money
DiscountedPrice	money

SELECT \* FROM Sales.GetDiscountedProducts(O)

SELECT \* FROM Sales.GetDiscountedProducts(I)

4 -

ExecutionContext.sql.

1. GetCur-  
rencyRate Sales,  
@CurrencyCode nchar(3),  
( float).  
@CurrencyRate float;  
DimCurrency FactCurrencyRate.  
Adventure WorksDW,  
MIAMNAdam.  
DimCurrency Adventure-  
WorksDW,

CurrencyKey (PK)	int, not null
CurrencyAlternateKey	nchar(3), not null
CurrencyName	nvarchar(50), not null

FactCurrencyRate, -

TimeKey (FK)	int, not null
CurrencyKey (FK)	int, not null
EndOfDayRate	float, not null

TimeKey FactCurrencyRate -  
DimTime, .

TimeKey (PK)	int, not null
DayNumberOfYear	smallint, null

:

SELECT Sales.GetCurrencyRate('GBP')

2.

GetCurrencyRate

, AdventureWorks Adventure-

WorksDW.

:

1)

AdventureWorksDW

USE Adventure WorksDW

IF NOT EXISTS (SELECT • FROM sys.database\_principals WHERE  
name = Adam' AND type = 'U')

CREATE USER Adam FOR LOGIN [MIAMNAdam]

2)

AUTHENTICATE

Adventure WorksDW:

GRANT AUTHENTICATE TO Adam

3)

SELECT

Adam:

GRANT SELECT TO Adam

4)

TRUSTWORTHY

AdventureWorks:

ALTER DATABASE AdventureWorks SET TRUSTWORTHY ON

3.

GetCurrencyDiscountedProducts.

GetCurrencyDiscountedProducts

Sales,

-

GetDiscountedProducts,

@CurrencyCode

nchai(3).

:



ProductID	int
Name	nvarchar(50)
ListPrice	money
Currencyprice	money
DiscountDescription	nvarchar(255)
Discountpercentage	smallmoney
DiscountAmount	money
DiscountedPrice	money
DiscountedCurrencyPrice	money

@CurrencyRate float.

```

Sales.GetCurrencyRate.      @CurrencyCode
    GetCurrencyRate.
        (      CurrencyPrice      ),
        ListPrice      @CurrencyRate.
    ,
    @CunencyRate      ListPrice
ListPrice      DiscountPct.
SELECT * FROM Sales.GetCurrencyDiscountedProducts('GBP')

```

10

AdventureWorks.

1. Trani.sql.  
 Person.Contact. SELECT  
 @@trancount ,  
 START TRANSACTION HERE  
 BEGIN TRANSACTION.
2. Execute,

ProductID	int
Name	nvarchar(50)
ListPrice	money
Currencyprice	money
DiscountDescription	nvarchar(255)
Discountpercentage	smallmoney
DiscountAmount	money
DiscountedPrice	money
DiscountedCurrencyPrice	money

@CurrencyRate float.

```

Sales.GetCurrencyRate.      @CurrencyCode
    GetCurrencyRate.
        (      CurrencyPrice      ),
        ListPrice      @CurrencyRate.
    ,
    ,
    @CunencyRate      ListPrice
ListPrice      DiscountPct.
SELECT * FROM Sales.GetCurrencyDiscountedProducts('GBP')

```

10

:

AdventureWorks.

I - :

1. Trani.sql.

Person.Contact. SELECT

@@trancount , .

START TRANSACTION HERE

BEGIN TRANSACTION.

2. Execute,

END TRANSACTION HERE,  
COMMIT TRANSACTION

1. Tran2.sql.B UPDATE  
Person.Contact.  
SELECT  
@@trancount ,

2. END TRANSACTION HERE,  
ROLLBACK TRANSACTION.

3 — :

1. SQL Server Management Studio  
Management , Activity Monitor  
View Proc-  
ess. , «  
» (Process Info), « » (Locks by Process)  
« » (Locks by Objects).

```
2. SELECT resource_type, request_mode, request_type, request_status,
    request_session_id
    FROM sys.dm_tran_locks
```

3.	Lock1.sql.	BEGIN
TRANSACTION,	ROLLBACK TRANSACTION	COMMIT
TRANSACTION.		

4. Execute, SPID.  
sys.dm\_tran\_locks.  
SPID

Lock1.sql.

ROLLBACK TRANSACTION.

sys.dm\_tran\_locks.

4 -

1. Lock2.sql.

SERIALIZABLE,

WHERE.

2. LockList.sql

request\_status

WAIT,

3. Lock2.sql

Cancel Exe-

cuting Query.

LockList.sql

4. Lock2.sql

BEGIN TRANSACTION:

SET lock timeout 5000

5.

6. SQL Server Management.

11

AdventureWorks.

Lock1.sql.

ROLLBACK TRANSACTION.

sys.dm\_tran\_locks.

4 -

1. Lock2.sql.

SERIALIZABLE,

WHERE.

2. LockList.sql

request\_status

WAIT,

3. Lock2.sql

Cancel Exe-

cuting Query.

LockList.sql

4. Lock2.sql

BEGIN TRANSACTION:

SET lock timeout 5000

5.

6. SQL Server Management.

11

AdventureWorks.

```
USE [AdventureWorks]

GO

CREATE TABLE [HumanResources].[JobCandidateHistory](
    [JobCandidateID] [int] NOT NULL UNIQUE,
    [Resume] [xml] NULL,
    [Rating] [int] NOT NULL CONSTRAINT
[DFJobCandidateHistoryRating] Default (5),
    [RejectedDate] [datetime] NOT NULL,
    [ContactID] [int] NULL,
    CONSTRAINT [FK_JobCandidateHistory_Contact_ContactID]
    FOREIGN KEY(ContactID) REFERENCES [Person].[Contact]
(ContactID),
    CONSTRAINT [CKJobCandi dateHistoryRating]
    CHECK ([Rating]>=0 AND [Rating]<=10)
) ON [PRIMARY]
```

```

2 - JobCandidate
HumanResources.dJobCandidate.
JobCandidateHistory
JobCandidate. , -
JobCandidateID Resume.
RejectedDate
GETDATEQ. Rating ,
ContactID NULL.

```

```
3 —
:
USE AdventureWorks
GO
DELETE FROM HumanResources.JobCandidate
```

JobCandidateID =

(SELECT MIN(JobCandidateID) FROM  
HumanResources.JobCandidate)

JobCandidateHistory. :

SELECT \* FROM [HumanResources].[JobCandidateHistory]

JobCandidateHistory, :

TRUNCATE TABLE [HumanResources].[JobCandidateHistory]

4 - :

1 OrderDetailNotDiscontinued Sales.SalesOr-  
derDetail. -

, -

Production.Product. -

, DiscontinuedDate ,

NULL. -

RAIS ERROR .

2 .

,

Tpurrepa . Produc-  
tion.Product ,

, ,

Product. :

USE AdventureWorks

GO

SELECT ProductID, Name FROM Production.Product

WHERE DiscontinuedDate IS NOT NULL

3 . , , :

UPDATE Production.Product

SET DiscontinuedDate = GETDATE()

WHERE ProductID = 680

4 .

, :



RT Sales.SalesOrderDetail

(SalesOrderID, OrderQty, ProductID, SpecialOfferID, UnitPrice,  
UnitPriceDiscount)

VALUES (43660, 5, 680, 1, 1431, 0)



1.	.....	3
2.	.....	5
3.	.....	8
4.	.....	10
5.	.....	14
6.	.....	16
7.	.....	18
8.	.....	19
9.	UDF.....	22
10.	.....	26
11.	.....	28

. . .

09.10.08. 60x84 1/16. .  
. «Times». . . 2,0.  
80 . *Qi*

« »

197376, .- , . . , 5

€

€

€