

« - « »»

« »

«Microsoft SQL Server 2008»

220201.68 « »

-
2012 .

681.5

Microsoft SQL Server 2008:

220201.68 «

», / . . . , - , , 2012. 31 .

:

. . , _____ . .

:

. . () . . , ;
. . () . . , ;

Microsoft SQL Server 2008.

220201.68 «

Microsoft SQL Server 2008

»

©

2012 .

Microsoft SQL Server.	1.	Microsoft SQL Server	4
Microsoft SQL Server.	2.	Transact-SQL.....	6
Microsoft SQL Server.	3.	9
Microsoft SQL Server.	4.	12
Microsoft SQL Server.	5.	Microsoft SQL Server	13
Microsoft SQL Server.	6.	17
Microsoft SQL Server.	7.	19
Microsoft SQL Server.	8.	21
Microsoft SQL Server.	9.	23
Microsoft SQL Server.	10.	24
Microsoft SQL Server.	11.	27

Microsoft SQL Server.

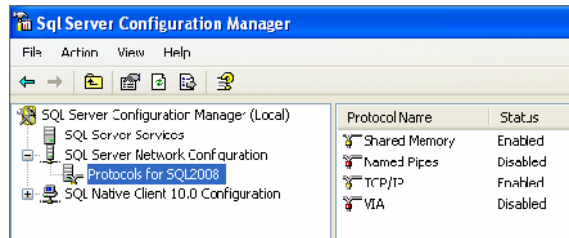
1. Microsoft SQL Server

Microsoft SQL Server 2008

:
 ,
 1 «
 , 2 « Microsoft SQL Server 2008»,
 MS SQL Server 2008,
 :
 .
 ,
 «
 ...».

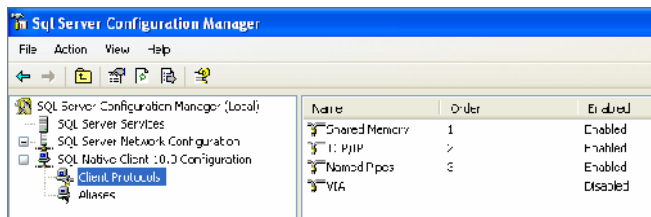
1. SQL Server Configuration.

1. SQL Server Configuration Manager
2. Windows XP Services,
 – Administrative Tools.
3. MS SQL Server 2008 Services
 MS SQL Server (SQL Server Configuration Manager).
4. Services,
 MS SQL Server (. 1.1). ?



. 1.1.

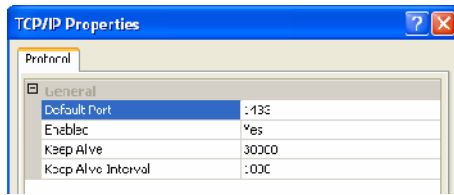
5. SQL Server Configuration Manager
 MS SQL Server (. 1.2).



. 1.2.

2.

SQL Server Configuration Manager – Client Protocols
SQL Native Client Configuration
 MS SQL Server, TCP- (. 1.3).



. 1.3.

TCP/IP

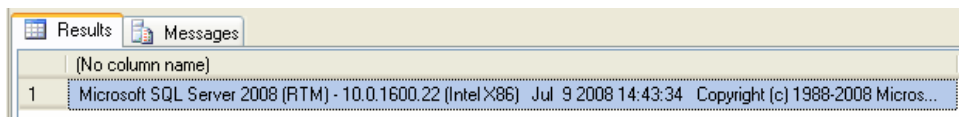
1. Server. MS SQL
2. Aliases (New alias...) SQL Native Client Configuration – Alias Name MyServ, TCP/IP, Server .
3. ?

3. SQL

1. – SQL Server Management Studio MyServ, SQL Server Configuration Manager. <Browse for more...>

2. Server: MyServ «sa», MS SQL
3. New Query SQL Server Management Studio.

4.) Results ()
SELECT @@version
 MS SQL Server () (. 1.4).



. 1.4.

MS SQL Server

- (F5), Query – Execute
- Query – Parse (Ctrl+F5).

MS SQL Server

File – New – Database Engine Query.

5. **File – Save.**
Object Explorer
 Databases *Object Explorer*).

4.

MS SQL Server.

MSSQLServer

SQL Server Management Studio,

1.

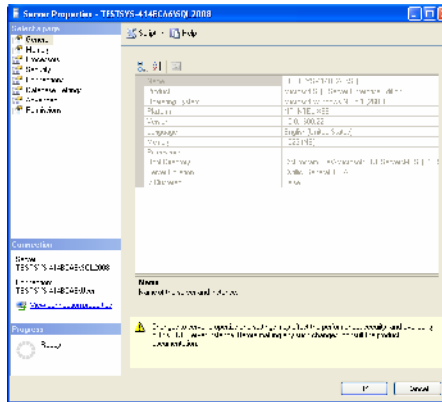
SQL Server Management Studio

Properties.

Object Explorer

2.

MyServ (. . . 1.5).



. 1.5.

MS SQL Server 2008

General

Memory

SQL Server:

Security

Connections

Advanced

Database Settings

3.

4.

MS SQL Server,

Microsoft SQL Server.

2.

Transact-SQL

:

Transact-SQL.

3 «

Transact-SQL»,

SQL,

SQL.

«

SQL-

».

MS

1. MyServ Management Studio.

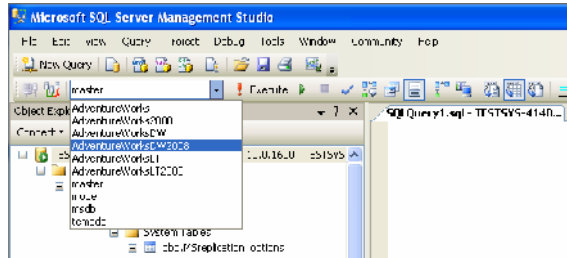
1. SQL Server Management Studio – Microsoft SQL Server 2008.

2. : SQL Server Authentication. User name: sa, Password: Connect.

2. Product.

1. AdventureWorks2008.

– USE AdventureWorks2008;
– AdventureWorks2008 (. . 2.1).



. 2.1.

2. Permissions object_id Product.

3. Permissions 32- Permissions

– ;
– object_id – ;
– : object_id –

Permissions.

. 2.1.

Permissions

	()		Object_id	Object_id
0	1	CREATE DATABASE	SELECT ALL	SELECT
1	2	CREATE TABLE	UPDATE ALL	UPDATE
2	4	CREATE PROCEDURE	REFERENCES ALL	REFERENCES
3	8	CREATE VIEW	INSERT	
4	16	CREATE RULE	DELECT	
5	32	CREATE DEFAULT	EXECUTE	
6	64	BACKUP DATABASE		
7	128	BACKUP LOG		
8	256			
12	4096		SELECT ANY	
13	8192		UPDATE ANY	
14	16384		REFERENCES ANY	

1. Permissions (. 2.2):
SELECT Permissions (object_id('production.product'))

Results		Messages	
(No column name)			
1	1881108511		

. 2.2. **Permissions**

2. `IF Permissions (object_id('production.product'))&1=1`
`SELECT 'Да'`
`ELSE`
`SELECT 'Нет'`

Results		Messages	
(No column name)			
1	Есть право выборки		

. 2.3. **Permissions**

3. `SELECT` `PRINT,` `SELECT.`
`IF Permissions (object_id('production.product'))&1=1`
`PRINT 'Да'`
`ELSE`
`PRINT 'Нет'`

Results, Messages (. 2.4).

Messages	
Есть право выборки	

. 2.4. **PRINT**

3. `PRINT`

1. `HOST_NAME().`
 2. `DataLength.`
 3. `IF (DATALENGTH(HOST_NAME()) / 2 > 9)`
`SELECT LEFT(HOST_Name(), 6) + '...'`
`ELSE`
`SELECT HOST_Name()`
`DATALENGTH`
`nvarchar,`

4. `HOST_NAME()`

4. `DECLARE @num int, @cnt int` (. . 2.5):
`SET @num = 19`
`IF (@num = 0) SET @cnt = 1`
`ELSE BEGIN`
`SET @cnt = 0`


```

WHILE (@num <> 0) BEGIN
    SET @cnt = @cnt + 1
    SET @num = @num / 10
END
END
SELECT @cnt AS '

```

Results	
Количество цифр	
1	2

. 2.5.

1. WHILE, 4: ,
2. , 'CardNumber', CreditCard
AdventureWorks2008. « », – «
».
3. (1 453,6).
:
4. 3.3 (/) – 1 496 .
5. User400-01\SQL2008 User400-01\User with dbo permissions.
6. MS SQL
Server.
7. ,
8. .
9. 33 , .
10. – , ()
11. , : – , ,

Microsoft SQL Server.

3.

: SELECT

4 « »,

SQL – SELECT

SQL- ,

« ».

1. MyServ

Management Studio.

1. SQL Server Management Studio – – Microsoft SQL
Server 2008.
2. : SQL Server Authentication. User name: sa,
Password: Connect.

2. **CountryRegion AdventureWorks2008.**

1. `CountryRegionCode, [Name], ModifiedDate`
`Person.CountryRegion.`
Person,
AS.
AdventureWorks2008
SELECT CountryRegionCode AS , [Name] AS ,
ModifiedDate AS ,
FROM Person.CountryRegion

. 3.1.

CountryRegionCode	Name	ModifiedDate
1	AdventureWorks2008	2008-06-30 00:00:00
2	AdventureWorks2008	2008-06-30 00:00:00
3	AdventureWorks2008	2008-06-30 00:00:00
4	AdventureWorks2008	2008-06-30 00:00:00
5	AdventureWorks2008	2008-06-30 00:00:00
6	AdventureWorks2008	2008-06-30 00:00:00
7	AdventureWorks2008	2008-06-30 00:00:00
8	AdventureWorks2008	2008-06-30 00:00:00
9	AdventureWorks2008	2008-06-30 00:00:00
10	AdventureWorks2008	2008-06-30 00:00:00
11	AdventureWorks2008	2008-06-30 00:00:00

. 3.1.

3.

1. `AS,`
`: LastName FirstName.`
Substring.
ORDER
BY.
SELECT LastName+' '+Substring(FirstName,1,1)+'' as [
], BusinessEntityID
FROM Person.Person
ORDER BY [
]

. 3.2.

LastName	FirstName	BusinessEntityID
1	AdventureWorks2008	1
2	AdventureWorks2008	2
3	AdventureWorks2008	3
4	AdventureWorks2008	4
5	AdventureWorks2008	5
6	AdventureWorks2008	6
7	AdventureWorks2008	7
8	AdventureWorks2008	8
9	AdventureWorks2008	9
10	AdventureWorks2008	10
11	AdventureWorks2008	11

. 3.2.

ASC (
– DESC.
4.

\$20,

\$12

1. **WHERE.**
 2.

SELECT [Name], ListPrice From Production.Product
WHERE (ListPrice>=12) and (ListPrice<=20)

ORDER BY 2
 3. BETWEEN, :
 SELECT [Name], ListPrice From Production.Product
 WHERE ListPrice Between 12 and 20
 ORDER BY ListPrice

. 3.3.

	Name	ListPrice
1	Taillights - Battery-Powered	13.99
2	Minipump	19.99

. 3.3. \$12 \$20

USA. 5. LIKE,
 1. :
 2. %, 0

3. HumanResources.vEmployee :
 Select LastName, JobTitle From HumanResources.vEmployee
 Where CountryRegionName Not Like '%United States%'

. 3.4.

	LastName	JobTitle
1	Pak	Sales Representative
2	Saraiva	Sales Representative
3	Tsofilias	Sales Representative
4	Valdez	Sales Representative
5	Vargas	Sales Representative
6	Varkey Chudukatil	Sales Representative

. 3.4.

6. IN. OR.
 1. :
 2. IN :
 Select LastName, JobTitle From HumanResources.vEmployee
 WHERE CountryRegionName IN ('United States', 'Canada')

. 3.5.

	LastName	JobTitle
1	Alles	Part. Sales Manager
2	Abercrombie	Production Technician - WC60
3	Abelcius	Quality Assurance Manager
4	Ackerman	Shipping and Receiving Supervisor
5	Adams	Production Technician - WC60
6	Ajenstat	Database Administrator
7	Alberis	European Sales Manager
8	Alderson	Production Technician - WC45
9	Alexander	Quality Assurance Technician
10	Altman	Facilities Manager
11	Ancerson	Production Technician - WC60

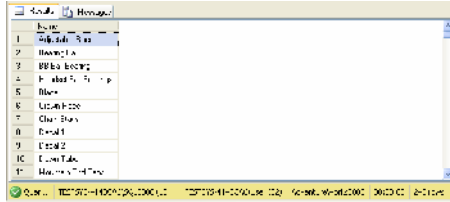
. 3.5.

7.

1. IS NULL.

2. *Product:*
SELECT [Name] FROM Production.Product
WHERE Color IS NULL

. 3.6.



. 3.6.

1. (ShipMethod),

2. (Employee)

EmployeeID, (dd.mm.yyyy)

3. :

4. : CONVERT (, [,]).

5. Production.Location, 'Paint'

6. vIndividualCustomer, , . . .

Microsoft SQL Server.

4.

SQL, 3 « Transact-SQL»,

5 « »,

SQL-

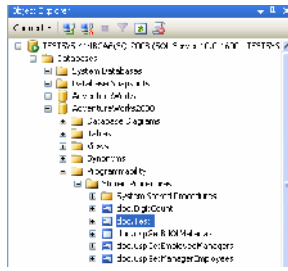
1. « ».

1. HOST_NAME().

2. :

```
CREATE PROCEDURE Test AS
IF (DATALENGTH(HOST_NAME()) / 2 > 9)
SELECT LEFT(HOST_Name(), 6) + '...'
ELSE
SELECT HOST_Name()
```

3. (. . 4.1).



. 4.1.

4. TEST , :

2. ,

OUTPUT.

1. DigitCount :

```
CREATE PROCEDURE DigitCount
    @num int,
    @cnt int OUTPUT
AS
IF (@num = 0) SET @cnt = 1
ELSE BEGIN
    SET @cnt = 0
    WHILE (@num <> 0) BEGIN
        SET @cnt = @cnt + 1
        SET @num = @num / 10
    END
END
END
```

2. DigitCount , :

```
DECLARE @num int = 1249
DECLARE @cnt int
EXEC DigitCount @num, @cnt OUTPUT
SELECT @cnt
```

EXEC

SQL , . , . . .

1.

2. : , ,

: SET

DATEFORMAT .

mdy, dmy, ymd, myd dym (m, d y- ,).

3. 16

4. , ,

: @,

5.

Microsoft SQL Server.

5.

Microsoft SQL

Server

:
MS SQL Server

4. sa, . . .

5.

sp_addsrvrolemember 'TempUser', 'securityadmin'

4.

AdventureWorks2008

db_owner.

1. sp_helprole

2. sp_helprolemember 'db_owner'

db_owner

(. . . 5.3).

	DbRole	MemberName	MemberSID
1	db_owner	dbo	0x010100000000000512000000

. 5.3.

db_owner

5.

TempUser.

1. sp_adduser 'TempUser', 'MyFirstUser'

2. sp_helpuser

3. db_datareader:

sp_addrolemember 'db_datareader', 'MyFirstUser'

6.

Andy:

Product

Name ListPrice

1. TestUser

AdventureWorks2008

Orders

GRANT select, update on AdventureWorks2008.Production.WorkOrder to TestUser

2. Andy

Name ListPrice Product AdventureWorks2008:

GRANT select on AdventureWorks2008.Production.Product (Name, ListPrice) to Andy

7.

Management Studio.

1.

Logins Security

2.

Login...

Logins,

New

-

General:

MS SQL Server -

-

Server Roles:

-

User Mapping:

User

Properties,

- Delete.

3.

Server Roles

Security

4. ,

5. Security/Users User
 Name, Login Name – New
User..., *User name*, *Login Name*

– **Delete.** **Properties,**

6. Roles.
 Properties.

7. Users ()
 Roles ().
 Securables.

() () – : (),

8.

TempUser.

1. :
 sp_droprolemember 'db_datareader', 'MyFirstUser' :

2. :
 sp_dropuser 'MyFirstUser'

3. :
 sp_dropsvrolemember 'TempUser', 'securityadmin'

4. :
 sp_droplogin 'TempUser'

1. ,
 dbcreator.

2. , sa.
 3. ,

4. sp_password.
 5. AdventureWorks2008,

MS SQL Server,

6. , manager AdventureWorks2008 ,

7. manager ,
 AdventureWorks2008.

Production.WorkOrder:

SELECT * FROM Production.WorkOrder.
 8. manager AdventureWorks2008,
 ?

- 9.
10. SQL Server Management Studio
SQL,
- AdventureWorks2008, diskadmin.
11. diskadmin
12. AdventureWorks2008
13. AdventureWorks2008
- Production.Product ?
14. AdventureWorks2008 SELECT * from Production.Product.
managers. Production.Product AdventureWorks2008.
managers ?
AdventureWorks2008 ?

Microsoft SQL Server.

6.

: SQL-

7 «

MS SQL Server»,
MS SQL Server 2008,

SQL-

«

1.

Sales c

1. CREATE DATABASE:

```
CREATE DATABASE Sales
ON
( NAME = Sales_dat,
  FILENAME = 'e:\saledat.mdf',
  SIZE = 10,
  MAXSIZE = 50,
  FILEGROWTH = 5 )
LOG ON
( NAME = Sales_log,
  FILENAME = 'e:\salelog.ldf',
  SIZE = 5MB,
  MAXSIZE = 25MB,
  FILEGROWTH = 5MB )
```

2.

(. 6.1):

sp_helpdb Sales

. 6.1.

name	db_size	owner	dbid	created	status
1 Sales	15.00 MB	TESTSYS-414BCA6\User	12	Sep 30 2009	Status=ONLINE, Updateability=READ_WRITE

name	fileid	filename	filegroup	size	maxsize	growth	usage
1 Sales_dat	1	e:\saledat.mdf	PRIMARY	10240 KB	51200 KB	5120 KB	data only
2 Sales_log	2	e:\salelog.ldf	NULL	5120 KB	25600 KB	5120 KB	log only

Query... TESTSYS-414BCA6\SQL2008 (10... TESTSYS-414BCA6\User (52) AdventureWorks2008 00:00:00 3 rows

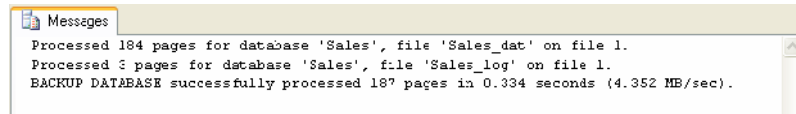
. 6.1.

2.

Sales.

1. sp_addumpdevice 'disk', 'backupdisk', 'e:\sales.bak'
2. BACKUP DATABASE Sales TO backupdisk

. 6.2.



. 6.2.

BACKUP

sp_helpdb

1.

1. ,
2. -2 . 5 , 10 c 20% .
3. D:\MSSQL\DATA.

2.

1. ,
2. ,
3. ,
4. .

3.

1. ,
2. 2
3. .

5.

1. ,
2. AdventureWorks2008. AdventureWorks2008 , 25%

6.

1. ,
2. ,

- 3.
- 7.
1. AdventureWorks2008
- PRIMARY.
2. AdventureWorks2008.
- 3.
- 8.
1. AdventureWorks2008.
- : D:\MSSQL\BACKUP\AdventureWorks_backup.bak.
2. AdventureWorks2008.
- 9.
1. AdventureWorks2008.
2. AdventureWorks_backup.bak.
- 3.

Microsoft SQL Server. 7.

SQL – SELECT, JOIN

4 « » ,

« ».

1.

Product AdventureWorks2008

Product ProductSubcategory.

SELECT

Product ProductSubcategory

Name.

SELECT Product.Name, ProductSubcategory.Name, Product.ListPrice
 FROM Production.Product INNER JOIN Production.ProductSubcategory
 ON Product.ProductSubcategoryID = ProductSubcategory.ProductSubcategoryID

. 7.1.

	Name	Name	ListPrice
1	H_road Frame - Black, E8	Road Frames	1437.50
2	H_road Frame - Red, E8	Road Frames	1137.50
3	Spor-100 Helmet, Red	Helmets	34.99
4	Spor-100 Helmet, Black	Helmets	34.99
5	Mountain Bike Socks, M	Socks	9.00
6	Mountain Bike Socks, L	Socks	9.00
7	Spor-100 Helmet, Blue	Helmets	34.99
8	AWT-1 Ingr Cap	Caps	8.99
9	Long-Sleeve Logo Jersey, S	Jerseys	43.99
10	Long-Sleeve Logo Jersey, M	Jerseys	43.99
11	Long-Sleeve Logo Jersey, L	Jerseys	43.99

. 7.1.

2.

Microsoft SQL Server.

8.

GROUP BY SELECT
 SELECT
 SQL-
 AVG.
 ListPrice,
 (. 8.1):
 SELECT AVG(ListPrice) FROM Production.Product

Results	
[No column name]	
1	438.6662

. 8.1.

CustomerID,
 Select CustomerID From Sales.SalesOrderHeader
 Where OrderDate=(SELECT Max(OrderDate) From Sales. SalesOrderHeader)
 . 8.2.

Results	
CustomerID	
1	11078
2	11927
3	28789
4	11794
5	14680
6	19585
7	27686
8	20601
9	26564
10	16170
11	16927

. 8.2. ID

Person,
 Person
 Customer.
 CustomerID,
 SELECT FirstName + ' ' + LastName AS 'CustomerName'
 FROM Person.Person
 INNER JOIN Sales.Customer ON Customer.PersonID = Person.BusinessEntityID
 WHERE CustomerID in
 (SELECT CustomerID FROM Sales.SalesOrderHeader
 WHERE OrderDate=(SELECT Max(OrderDate) FROM Sales. SalesOrderHeader))
 . 8.3.

	CustomerName
1	Alisha Raji
2	Chloe Harris
3	Kristi Fernandez
4	Tamara Lu
5	Devin Phillips
6	Rebekah Gonzalez
7	Caleb Lal
8	Marvin Munoz
9	Jacqueline Murphy
10	Chloe Martin
11	Christina Morris

8.3.

3.

300

1.

2.

3.

```
SELECT SalesOrderID, CustomerID
FROM Sales.SalesOrderHeader oh
WHERE 300<(SELECT sum(OrderQty)
FROM Sales.SalesOrderDetail od WHERE od.SalesOrderID=oh.SalesOrderID)
```

4.

1.

SalesOrderDetail,

Product

2.

SalesOrderDetail

3.

```
SELECT Name, OrderQty, SalesOrderID
FROM Production.Product p INNER JOIN Sales.SalesOrderDetail od1
ON od1.ProductID=p.ProductID
WHERE OrderQty = (SELECT Max(OrderQty)
FROM Sales.SalesOrderDetail od2
WHERE od1.ProductID=od2.ProductID)
```

5.

1.

SalesOrderID

ProductID.

```
SELECT SalesOrderID, count(ProductID) AS [ ]
FROM Sales.SalesOrderDetail
GROUP BY SalesOrderID
```

GROUP BY

2.

```

SELECT SalesOrderID,
count(ProductID) AS [
],
sum(OrderQty) as [
]
FROM Sales.SalesOrderDetail
GROUP BY SalesOrderID

```

WHERE,

1. AdventureWorks2008.
- 2.
- 3.
- 4.
- 5.
6. :
7. , 50 .

Microsoft SQL Server.

9.

```

SELECT,
4 « »,
5 « »,
SQL-
»
1.
AdventureWorks2008, : ID
( )

```

1. CREATE VIEW InfoEmployees ([], [], []) AS
SELECT BusinessEntityID, JobTitle + '(' + LoginID + ')', CONVERT (char(10), BirthDate, 104)
FROM HumanResources.Employee
2. (. 9.1):
SELECT * FROM InfoEmployees

Идентификатор	Фамилия	Дата рождения
1	Chief Executive Officer(AdventureWorks2008)	02.02.1952
2	Vice President of Engineering(AdventureWorks2008)	01.01.1901
3	Engineering Manager(AdventureWorks2008)	13.12.1964
4	Senior Tool Designer(AdventureWorks2008)	23.01.1985
5	Design Engineer(AdventureWorks2008)	28.10.1942
6	Design Engineer(AdventureWorks2008)	11.04.1944
7	Research and Development Manager(AdventureWorks2008)	27.02.1977
8	Research and Development Engineer(AdventureWorks2008)	06.07.1976
9	Research and Development Engineer(AdventureWorks2008)	21.02.1986
10	Research and Development Manager(AdventureWorks2008)	01.01.1944
11	Senior Tool Designer(AdventureWorks2008)	10.02.1902

9.1.

InfoEmployees

Employee,

2.

1. Name Vendor.

2. « » WHERE.

3. CREATE VIEW OnlineVendors AS
 SELECT [Name] FROM Purchasing.Vendor
 WHERE PurchasingWebServiceURL IS NOT NULL

1. Customer, Person, SalesOrderHeader SalesOrderDetail.

2. SalesOrderID. GROUP BY SELECT.

3. UnitPrice * OrderQty * (1 - UnitPriceDiscount).

4. CREATE VIEW InfoOrders AS
 SELECT FirstName + ' ' + LastName as [],
 SalesOrderHeader.SalesOrderID as [],
 Convert (money, SUM(UnitPrice*OrderQty*(1-UnitPriceDiscount)),0) AS []
 FROM (Person.Person INNER JOIN Sales.Customer
 ON Person.BusinessEntityID = Customer.PersonID
 INNER JOIN Sales.SalesOrderHeader
 ON Customer.CustomerID=SalesOrderHeader.CustomerID)
 INNER JOIN Sales.SalesOrderDetail
 ON SalesOrderHeader.SalesOrderID=SalesOrderDetail.SalesOrderID
 GROUP BY SalesOrderHeader.SalesOrderID, FirstName + ' ' + LastName

1. 5 ()
 InfoOrders.
 AdventureWorks2008,
 convert (char(7),OrderDate,102),
 yyyy.mm.dd.
 7 yyyy.mm.
 3. InfoSalesPerson,
 AdventureWorks2008:

Microsoft SQL Server. 10.

SQL-

10 « » 11 « »

3

«

».

Microsoft

- 1.
- 2.
- 3.
- 4.
- 5.
- 6.

(- - -),

(- - -).

1900

'unknown';

00-00-00,

'no'.

1900

- 4 (): NULL.
- :
- :
- , .
- , ,
- , - .
- 1. , ,
- 2. : MS SQL Server, ISO SQL:2008.
- : PRIMARY KEY.
- FOREIGN KEY. (NULL)
- UNIQUE. NULL.
- NULL.
- DEFAULT.
- CHECK.

sp_helpconstraint

```

CREATE TABLE Students
(StudentID int IDENTITY(1,1)
CONSTRAINT prkStudentID PRIMARY KEY,
LastName char(20) NULL,
FirstName char(20) NULL,
BirthDate datetime NULL
CONSTRAINT bdChek CHECK (BirthDate < GetDate() ) )

```

: sp_help

MS SQL Server

SQL Management Studio.

information_schema.

- table_constraints -
- check_constraints - CHECK.
- referential_constraints -

1.
2. PRIMARY KEY
3. CHECK.

Microsoft SQL Server.

11.

SQL

4 « », SELECT,
9 «
»,

SQL-

2

1. Lect

1. : Brdate, FirstName, LastName.

2.

3.

INSERT INTO Lect (Brdate, FirstName, LastName)
VALUES (Convert(datetime,'01.01.1970',104), '...', '...')

INSERT INTO Lect DEFAULT VALUES

2.

Lect

Employee

AdventureWorks2008:

1. 4 Lect: FirstName, LastName, Brdate, HireDate.

2.

SELECT:

INSERT INTO Lect (FirstName, LastName, Brdate, HireDate)
SELECT FirstName, LastName, BirthDate, HireDate
FROM AdventureWorks2008.HumanResources.Employee
INNER JOIN AdventureWorks2008.Person.Contact
ON Contact.ContactID = Employee.ContactID

SELECT

INSERT

3. Students

```
1. C: students.txt
   1 Ivan      Petrov 1.1.1990
   2 Petr      Ivanov 12.10.1992
   3 Sergey    Kazakov5.3.1991
```

TAB,

```
2. BULK INSERT Students FROM 'c:\students.txt'
```

StudentID

students.txt,

KEEPIDENTITY.

4.

King.

```
1. LIKE.
2. 'King' 35-35-35:
   UPDATE Lect
   SET Phone = '35-35-35'
   WHERE FirstName like 'King'
```

5.

King.

```
1. Course
   : SELECT LectID From Lect Where (FirstName like
   'King').
```

```
2.
3. UPDATE Course SET LectID =
   (SELECT LectID From Lect Where (FirstName like 'King'))
   WHERE CourseName like '%[Ii]nform%'
```

6.

Ivanov.

```
1. WHERE
   DELETE.
```

```
2. DELETE Students
   WHERE LastName = 'Ivanov'
```

TABLE

: TRUNCATE

DELETE, . .

```
1. Lect Class
2. Students
3.
4. 1990
```