

## Table of Contents

VISTA-DEV .....	1
Database .....	1
Northwind .....	1
Northwind .....	34
StoredProcedure .....	34
CustOrderHist .....	34
CustOrdersDetail .....	35
CustOrdersOrders .....	36
Employee Sales by Country .....	37
Sales by Year .....	38
SalesByCategory .....	39
Ten Most Expensive Products .....	40
Table .....	5
Categories .....	5
CustomerCustomerDemo .....	6
CustomerDemographics .....	7
Customers .....	8
Employees .....	9
EmployeeTerritories .....	10
Order Details .....	11
Orders .....	12
Products .....	13
Region .....	14
Shippers .....	15
Suppliers .....	16
Territories .....	17
User .....	119
dbo .....	119
guest .....	120
INFORMATION_SCHEMA .....	121
sys .....	122
UserDefinedFunction .....	41
dm_db_index_operational_stats .....	41
dm_db_index_physical_stats .....	42
dm_db_missing_index_columns .....	43
dm_exec_cached_plan_dependent_objects .....	44
dm_exec_cursors .....	45
dm_exec_plan_attributes .....	46
dm_exec_query_plan .....	47
dm_exec_sql_text .....	48
dm_io_virtual_file_stats .....	49
fn_builtin_permissions .....	50
fn_cColvEntries_80 .....	51
fn_check_object_signatures .....	52
fn_dblog .....	53
fn_dump_dblog .....	56
fn_EnumCurrentPrincipals .....	61
fn_fIsColTracked .....	62
fn_get_sql .....	63
fn_GetCurrentPrincipal .....	64
fn_GetRowsetIdFromRowDump .....	65
fn_helpcollations .....	66
fn_helpdatatypemap .....	67
fn_IsBitSetInBitmask .....	70
fn_isrolemember .....	71
fn_listextendedproperty .....	73
fn_MapSchemaType .....	74
fn_MSdayasnumber .....	80
fn_MSgeneration_downloadonly .....	81
fn_MSget_dynamic_filter_login .....	82
fn_MSorbitmaps .....	84
fn_MSrepl_map_resolver_clsId .....	85
fn_MStestbit .....	86
fn_MSvector_downloadonly .....	87

fn_my_permissions	88
fn_numberOf1InBinaryAfterLoc	90
fn_numberOf1InVarBinary	91
fn_repladjustcolumnmap	92
fn_repldecryptver4	94
fn_replformatdatetime	96
fn_replgetcolidfrombitmap	97
fn_replgetpasedddcmd	98
fn_replreplacesinglequote	99
fn_replreplacesinglequoteplusprotectstring	100
fn_repluniqueid	101
fn_RowDumpCracker	102
fn_serversharedrives	105
fn_sqlvarbasetostr	106
fn_trace_geteventinfo	108
fn_trace_getfilterinfo	109
fn_trace_getinfo	110
fn_trace_gettable	111
fn_translate_permissions	112
fn_varbintohexstr	113
fn_varbintohexsubstring	114
fn_virtualfilestats	116
fn_virtualservernodes	117
fn_yukonsecuritymodelrequired	118
View	18
Alphabetical list of products	18
Category Sales for 1997	19
Current Product List	20
Customer and Suppliers by City	21
Invoices	22
Order Details Extended	23
Order Subtotals	24
Orders Qty	25
Product Sales for 1997	26
Products Above Average Price	27
Products by Category	28
Quarterly Orders	29
Sales by Category	30
Sales Totals by Amount	31
Summary of Sales by Quarter	32
Summary of Sales by Year	33

[Northwind]

## [Northwind]

[Database]

### General

Property	Value
Active Connections	1
Auto Create Statistics Enabled	True
Auto Update Statistics Enabled	True
Case Sensitive	False
Collation	Latin1_General_CI_AS
Compatibility Level	Version90
Create Date	27/07/2007 21:11:03
Database Guid	b3f903a2-0b28-49a0-a519-3c8801138ca7
Database Snapshot Base Name	
Data Space Usage	1328
Dbo Login	True
Default File Group	PRIMARY
Default Full Text Catalog	
Default Schema	dbo
I D	6
Index Space Usage	1496
Is Accessible	True
Is Database Snapshot	False
Is Database Snapshot Base	False
Is Db Access Admin	True
Is Db Backup Operator	True
Is Db Datareader	True
Is Db Datawriter	True
Is Db Ddl Admin	True
Is Db Deny Datareader	False
Is Db Deny Datawriter	False
Is Db Owner	True
Is Db Security Admin	True
Is Full Text Enabled	True
Is Mail Host	False
Is Mirroring Enabled	False
Is System Object	False
Is Updateable	True
Last Backup Date	01/01/0001 00:00:00
Last Log Backup Date	01/01/0001 00:00:00
Log Reuse Wait Status	Nothing
Mirroring Failover Log Sequence Number	
Mirroring I D	
Mirroring Partner	
Mirroring Role	
Mirroring Role Sequence	
Mirroring Safety Level	None
Mirroring Safety Sequence	
Mirroring Status	None
Mirroring Witness	
Mirroring Witness Status	None
Owner	Vista-Dev\starprint
Primary File Path	D:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\DATA
Recovery Fork Guid	0405bbfa-200b-4c49-9280-1f764c211a00
Replication Options	0
Service Broker Guid	2e176c5c-51b4-4dd4-9966-ba23f6a9ce74
Size	4.1875
Space Available	232
Status	Normal
User Name	dbo
Version	611

### Miscellaneous

Property	Value
----------	-------

[Northwind]

Property	Value
Active Connections	1
Case Sensitive	False
Database Guid	b3f903a2-0b28-49a0-a519-3c8801138ca7
Database Snapshot Base Name	
Data Space Usage	1328
Dbo Login	True
Default File Group	PRIMARY
Default Full Text Catalog	
Default Schema	dbo
Index Space Usage	1496
Is Accessible	True
Is Database Snapshot	False
Is Database Snapshot Base	False
Is Db Access Admin	True
Is Db Backup Operator	True
Is Db Datareader	True
Is Db Datawriter	True
Is Db Ddl Admin	True
Is Db Deny Datareader	False
Is Db Deny Datawriter	False
Is Db Owner	True
Is Db Security Admin	True
Is Full Text Enabled	True
Is Mail Host	False
Is Mirroring Enabled	False
Is System Object	False
Is Updateable	True
Log Reuse Wait Status	Nothing
Primary File Path	D:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\DATA
Recovery Fork Guid	0405bbfa-200b-4c49-9280-1f764c211a00
Replication Options	0
Service Broker Guid	2e176c5c-51b4-4dd4-9966-ba23f6a9ce74
User Name	dbo
Version	611

**Files**

Name	File Type	Filegroup	Initial Size	Autogrowth
Northwind	Data	PRIMARY	3264	1024,KB
Northwind_log	Log		1024	10,Percent

**File Groups**

Name	Files	Read Only	Default
PRIMARY	1	False	True

**Options**

Property	Value
Active Connections	1
Ansi Null Default	False
Ansi Nulls Enabled	False
Ansi Padding Enabled	False
Ansi Warnings Enabled	False
Arithmetic Abort Enabled	False
Auto Close	False
Auto Create Statistics	True
Auto Create Statistics Enabled	True
Auto Shrink	False
Auto Update Statistics	True
Auto Update Statistics Async	False
Auto Update Statistics Enabled	True
Case Sensitive	False
Close Cursors On Commit Enabled	False

[Northwind]

Property	Value
Collation	Latin1_General_CI_AS
Compatibility Level	Version90
Concatenate Null Yields Null	False
Create Date	27/07/2007 21:11:03
Database Guid	b3f903a2-0b28-49a0-a519-3c8801138ca7
Database Ownership Chaining	False
Database Snapshot Base Name	
Data Space Usage	1328
Date Correlation Optimization	False
Dbo Login	True
Default File Group	PRIMARY
Default Full Text Catalog	
Default Schema	dbo
I D	6
Index Space Usage	1496
Is Accessible	True
Is Database Snapshot	False
Is Database Snapshot Base	False
Is Db Access Admin	True
Is Db Backup Operator	True
Is Db Datareader	True
Is Db Datawriter	True
Is Db Ddl Admin	True
Is Db Deny Datareader	False
Is Db Deny Datawriter	False
Is Db Owner	True
Is Db Security Admin	True
Is Full Text Enabled	True
Is Mail Host	False
Is Mirroring Enabled	False
Is System Object	False
Is Updateable	True
Last Backup Date	01/01/0001 00:00:00
Last Log Backup Date	01/01/0001 00:00:00
Log Reuse Wait Status	Nothing
Mirroring Failover Log Sequence Number	
Mirroring I D	
Mirroring Partner	
Mirroring Role	
Mirroring Role Sequence	
Mirroring Safety Level	None
Mirroring Safety Sequence	
Mirroring Status	None
Mirroring Witness	
Mirroring Witness Status	None
Numeric Round Abort Enabled	False
Owner	Vista-Dev\starprint
Page Verify	Checksum
Primary File Path	D:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\DATA
Quoted Identifiers Enabled	False
Read Only	False
Recovery Fork Guid	0405bbfa-200b-4c49-9280-1f764c211a00
Recovery Model	Simple
Recursive Triggers Enabled	False
Replication Options	0
Service Broker Guid	2e176c5c-51b4-4dd4-9966-ba23f6a9ce74
Size	4.1875
Space Available	232
Status	Normal
Trustworthy	False
User Name	dbo
Version	611

**Explicit Permissions:**

dbo [User]

Permission	Grantor	Grant	With Grant	Deny
Connect	dbo	X		

**Mirroring**

Property	Value
Mirroring Failover Log Sequence Number	
Mirroring I D	
Mirroring Partner	
Mirroring Role	
Mirroring Role Sequence	
Mirroring Safety Level	None
Mirroring Safety Sequence	
Mirroring Status	None
Mirroring Witness	
Mirroring Witness Status	None

**[dbo].[Categories]**

'Table' in Database 'Northwind'

CHK=Check CLU=Clustered CO=Computed, FI=Full Text Indexed, FK=Foreign Key, ID=Identity, NR=Not For Replication, NU=Nullable, PK=Primary Key, PR=Persisted UNQ=Unique

**Columns**

ID	Name	Data Type	C	F	F	I	N	N	P	P	Default
			O	I	K	D	R	U	K	R	
1	CategoryID	int				X			X		
2	CategoryName	nvarchar(15)									
3	Description	ntext(16)						X			
4	Picture	image						X			

**Primary Keys**

Name	Columns	Clustered	Unique
PK_Categories	CategoryID	X	

**Foreign Keys**

None

**Triggers**

None

**Indices**

Name	Columns	Clustered	Unique
CategoryName	CategoryName		
PK_Categories	CategoryID	X	

**[dbo].[CustomerCustomerDemo]**

'Table' in Database 'Northwind'

**Columns**

ID	Name	Data Type	C O	F I	F K	I D	N R	N U	P K	P R	Default
1	CustomerID	nchar(5)			X				X		
2	CustomerTypeID	nchar(10)			X				X		

**Primary Keys**

Name	Columns	Clustered	Unique
PK_CustomerCustomerDemo	CustomerID CustomerTypeID		

**Foreign Keys**

Name	Columns	Check	Enabled
<a href="#">FK_CustomerCustomerDemo</a>		X	X
<b>Primary/Unique Key Base Table</b>			
CustomerCustomerDemo	CustomerTypeID		
<b>Foreign Key Base Table</b>			
CustomerDemographics	CustomerTypeID		
<a href="#">FK_CustomerCustomerDemo_CustomerCustomerDemo</a>		X	X
<b>Primary/Unique Key Base Table</b>			
CustomerCustomerDemo	CustomerID CustomerTypeID		
<b>Foreign Key Base Table</b>			
CustomerCustomerDemo	CustomerID CustomerTypeID		
<a href="#">FK_CustomerCustomerDemo_Customers</a>		X	X
<b>Primary/Unique Key Base Table</b>			
CustomerCustomerDemo	CustomerID		
<b>Foreign Key Base Table</b>			
Customers	CustomerID		

**Triggers**

None

**Indices**

Name	Columns	Clustered	Unique
PK_CustomerCustomerDemo	CustomerID CustomerTypeID		

**[dbo].[CustomerDemographics]**

'Table' in Database 'Northwind'

**Columns**

ID	Name	DataType	C	F	F	I	N	N	P	P	Default
			O	I	K	D	R	U	K	R	
1	CustomerTypeID	nchar(10)							X		
2	CustomerDesc	ntext(16)						X			

**Primary Keys**

Name	Columns	Clustered	Unique
PK_CustomerDemographics	CustomerTypeID		

**Foreign Keys**

None

**Triggers**

None

**Indices**

Name	Columns	Clustered	Unique
PK_CustomerDemographics	CustomerTypeID		

**[dbo].[Customers]**

'Table' in Database 'Northwind'

**Columns**

ID	Name	DataType	C O	F I	F K	I D	N R	N U	P K	P R	Default
1	CustomerID	nchar(5)							X		
2	CompanyName	nvarchar(40)									
3	ContactName	nvarchar(30)						X			
4	ContactTitle	nvarchar(30)						X			
5	Address	nvarchar(60)						X			
6	City	nvarchar(15)						X			
7	Region	nvarchar(15)						X			
8	PostalCode	nvarchar(10)						X			
9	Country	nvarchar(15)						X			
10	Phone	nvarchar(24)						X			
11	Fax	nvarchar(24)						X			

**Primary Keys**

Name	Columns	Clustered	Unique
PK_Customers	CustomerID	X	

**Foreign Keys**

None

**Triggers**

None

**Indices**

Name	Columns	Clustered	Unique
City	City		
CompanyName	CompanyName		
PK_Customers	CustomerID	X	
PostalCode	PostalCode		
Region	Region		

**[dbo].[Employees]**

'Table' in Database 'Northwind'

**Columns**

ID	Name	DataType	C	F	F	I	N	N	P	P	Default
			O	I	K	D	R	U	K	R	
1	EmployeeID	int				X			X		
2	LastName	nvarchar(20)									
3	FirstName	nvarchar(10)									
4	Title	nvarchar(30)						X			
5	TitleOfCourtesy	nvarchar(25)						X			
6	BirthDate	datetime						X			
7	HireDate	datetime						X			
8	Address	nvarchar(60)						X			
9	City	nvarchar(15)						X			
10	Region	nvarchar(15)						X			
11	PostalCode	nvarchar(10)						X			
12	Country	nvarchar(15)						X			
13	HomePhone	nvarchar(24)						X			
14	Extension	nvarchar(4)						X			
15	Photo	image						X			
16	Notes	ntext(16)						X			
17	ReportsTo	int			X			X			
18	PhotoPath	nvarchar(255)						X			

**Primary Keys**

Name	Columns	Clustered	Unique
PK_Employees	EmployeeID	X	

**Foreign Keys**

Name	Columns	Check	Enabled
<a href="#">FK_Employees_Employees</a>			X
<b>Primary/Unique Key Base Table</b>			
Employees	ReportsTo		
<b>Foreign Key Base Table</b>			
Employees	EmployeeID		

**Triggers**

None

**Indices**

Name	Columns	Clustered	Unique
LastName	LastName		
PK_Employees	EmployeeID	X	
PostalCode	PostalCode		

**[dbo].[EmployeeTerritories]**

'Table' in Database 'Northwind'

**Columns**

ID	Name	Data Type	C	F	F	I	N	N	P	P	Default
			O	I	K	D	R	U	K	R	
1	EmployeeID	int			X				X		
2	TerritoryID	nvarchar(20)			X				X		

**Primary Keys**

Name	Columns	Clustered	Unique
PK_EmployeeTerritories	EmployeeID TerritoryID		

**Foreign Keys**

Name	Columns	Check	Enabled
<a href="#">FK_EmployeeTerritories_Employees</a>		X	X
<b>Primary/Unique Key Base Table</b>			
EmployeeTerritories	EmployeeID		
<b>Foreign Key Base Table</b>			
Employees	EmployeeID		
<a href="#">FK_EmployeeTerritories_Territories</a>		X	X
<b>Primary/Unique Key Base Table</b>			
EmployeeTerritories	TerritoryID		
<b>Foreign Key Base Table</b>			
Territories	TerritoryID		

**Triggers**

None

**Indices**

Name	Columns	Clustered	Unique
PK_EmployeeTerritories	EmployeeID TerritoryID		

**[dbo].[Order Details]**

'Table' in Database 'Northwind'

**Columns**

ID	Name	Data Type	C	F	F	I	N	N	P	P	Default
			O	I	K	D	R	U	K	R	
1	OrderID	int			X				X		
2	ProductID	int			X				X		
3	UnitPrice	money									((0))
4	Quantity	smallint									((1))
5	Discount	real									((0))

**Primary Keys**

Name	Columns	Clustered	Unique
PK_Order_Details	OrderID ProductID	X	

**Foreign Keys**

Name	Columns	Check	Enabled
<a href="#">FK_Order_Details_Orders</a>			X
<b>Primary/Unique Key Base Table</b>			
Order Details	OrderID		
<b>Foreign Key Base Table</b>			
Orders	OrderID		
<a href="#">FK_Order_Details_Products</a>			X
<b>Primary/Unique Key Base Table</b>			
Order Details	ProductID		
<b>Foreign Key Base Table</b>			
Products	ProductID		

**Triggers**

None

**Indices**

Name	Columns	Clustered	Unique
OrderID	OrderID		
OrdersOrder_Details	OrderID		
PK_Order_Details	OrderID	X	
	ProductID		
ProductID	ProductID		
ProductsOrder_Details	ProductID		

**[dbo].[Orders]**

'Table' in Database 'Northwind'

**Columns**

ID	Name	DataType	C	F	F	I	N	N	P	P	Default
			O	I	K	D	R	U	K	R	
1	OrderID	int				X			X		
2	CustomerID	nchar(5)			X			X			
3	EmployeeID	int			X			X			
4	OrderDate	datetime						X			
5	RequiredDate	datetime						X			
6	ShippedDate	datetime						X			
7	ShipVia	int			X			X			
8	Freight	money						X		((0))	
9	ShipName	nvarchar(40)						X			
10	ShipAddress	nvarchar(60)						X			
11	ShipCity	nvarchar(15)						X			
12	ShipRegion	nvarchar(15)						X			
13	ShipPostalCode	nvarchar(10)						X			
14	ShipCountry	nvarchar(15)						X			

**Primary Keys**

Name	Columns	Clustered	Unique
PK_Orders	OrderID	X	

**Foreign Keys**

Name	Columns	Check	Enabled
<u>FK_Orders_Customers</u>			X
<b>Primary/Unique Key Base Table</b>			
Orders	CustomerID		
<b>Foreign Key Base Table</b>			
Customers	CustomerID		
<u>FK_Orders_Employees</u>			X
<b>Primary/Unique Key Base Table</b>			
Orders	EmployeeID		
<b>Foreign Key Base Table</b>			
Employees	EmployeeID		
<u>FK_Orders_Shippers</u>			X
<b>Primary/Unique Key Base Table</b>			
Orders	ShipVia		
<b>Foreign Key Base Table</b>			
Shippers	ShipperID		

**Triggers**

None

**Indices**

Name	Columns	Clustered	Unique
CustomerID	CustomerID		
CustomersOrders	CustomerID		
EmployeeID	EmployeeID		
EmployeesOrders	EmployeeID		
OrderDate	OrderDate		
PK_Orders	OrderID	X	
ShippedDate	ShippedDate		
ShippersOrders	ShipVia		
ShipPostalCode	ShipPostalCode		

**[dbo].[Products]**

'Table' in Database 'Northwind'

**Columns**

ID	Name	DataType	C O	F I	F K	I D	N R	N U	P K	P R	Default
1	ProductID	int				X			X		
2	ProductName	nvarchar(40)									
3	SupplierID	int			X			X			
4	CategoryID	int			X			X			
5	QuantityPerUnit	nvarchar(20)						X			
6	UnitPrice	money						X			((0))
7	UnitsInStock	smallint						X			((0))
8	UnitsOnOrder	smallint						X			((0))
9	ReorderLevel	smallint						X			((0))
10	Discontinued	bit									((0))

**Primary Keys**

Name	Columns	Clustered	Unique
PK_Products	ProductID	X	

**Foreign Keys**

Name	Columns	Check	Enabled
<u>FK_Products_Categories</u>			X
<b>Primary/Unique Key Base Table</b>			
Products	CategoryID		
<b>Foreign Key Base Table</b>			
Categories	CategoryID		
<u>FK_Products_Suppliers</u>			X
<b>Primary/Unique Key Base Table</b>			
Products	SupplierID		
<b>Foreign Key Base Table</b>			
Suppliers	SupplierID		

**Triggers**

None

**Indices**

Name	Columns	Clustered	Unique
CategoriesProducts	CategoryID		
CategoryID	CategoryID		
PK_Products	ProductID	X	
ProductName	ProductName		
SupplierID	SupplierID		
SuppliersProducts	SupplierID		

**[dbo].[Region]**

'Table' in Database 'Northwind'

**Columns**

ID	Name	DataType	C	F	F	I	N	N	P	P	Default
			O	I	K	D	R	U	K	R	
1	RegionID	int							X		
2	RegionDescription	nchar(50)									

**Primary Keys**

Name	Columns	Clustered	Unique
PK_Region	RegionID		

**Foreign Keys**

None

**Triggers**

None

**Indices**

Name	Columns	Clustered	Unique
PK_Region	RegionID		

**[dbo].[Shippers]**

'Table' in Database 'Northwind'

**Columns**

ID	Name	DataType	C	F	F	I	N	N	P	P	Default
			O	I	K	D	R	U	K	R	
1	ShipperID	int				X			X		
2	CompanyName	nvarchar(40)									
3	Phone	nvarchar(24)						X			

**Primary Keys**

Name	Columns	Clustered	Unique
PK_Shippers	ShipperID	X	

**Foreign Keys**

None

**Triggers**

None

**Indices**

Name	Columns	Clustered	Unique
PK_Shippers	ShipperID	X	

**[dbo].[Suppliers]**

'Table' in Database 'Northwind'

**Columns**

ID	Name	Data Type	C O	F I	F K	I D	N R	N U	P K	P R	Default
1	SupplierID	int				X			X		
2	CompanyName	nvarchar(40)									
3	ContactName	nvarchar(30)						X			
4	ContactTitle	nvarchar(30)						X			
5	Address	nvarchar(60)						X			
6	City	nvarchar(15)						X			
7	Region	nvarchar(15)						X			
8	PostalCode	nvarchar(10)						X			
9	Country	nvarchar(15)						X			
10	Phone	nvarchar(24)						X			
11	Fax	nvarchar(24)						X			
12	HomePage	ntext(16)						X			

**Primary Keys**

Name	Columns	Clustered	Unique
PK_Suppliers	SupplierID	X	

**Foreign Keys**

None

**Triggers**

None

**Indices**

Name	Columns	Clustered	Unique
CompanyName	CompanyName		
PK_Suppliers	SupplierID	X	
PostalCode	PostalCode		

**[dbo].[Territories]**

'Table' in Database 'Northwind'

**Columns**

ID	Name	Data Type	C	F	F	I	N	N	P	P	Default
			O	I	K	D	R	U	K	R	
1	TerritoryID	nvarchar(20)							X		
2	TerritoryDescription	nchar(50)									
3	RegionID	int			X						

**Primary Keys**

Name	Columns	Clustered	Unique
PK_Territories	TerritoryID		

**Foreign Keys**

Name	Columns	Check	Enabled
<a href="#">FK_Territories_Region</a>		X	X
<b>Primary/Unique Key Base Table</b>			
Territories	RegionID		
<b>Foreign Key Base Table</b>			
Region	RegionID		

**Triggers**

None

**Indices**

Name	Columns	Clustered	Unique
PK_Territories	TerritoryID		

**[dbo].[Alphabetical list of products]**

'View' in Database 'Northwind'

CHK=Check CLU=Clustered CO=Computed, FI=Full Text Indexed, FK=Foreign Key, ID=Identity, NR=Not For Replication, NU=Nullable, PK=Primary Key, PR=Persisted UNQ=Unique

**Columns**

ID	Name	Data Type	C	F	F	I	N	N	P	P
			O	I	K	D	R	U	K	R
1	ProductID	int								
2	ProductName	nvarchar(40)								
3	SupplierID	int						X		
4	CategoryID	int						X		
5	QuantityPerUnit	nvarchar(20)						X		
6	UnitPrice	money						X		
7	UnitsInStock	smallint						X		
8	UnitsOnOrder	smallint						X		
9	ReorderLevel	smallint						X		
10	Discontinued	bit								
11	CategoryName	nvarchar(15)								

**Script:**

00001

00002 **create view** "Alphabetical list of products" **AS**

00003

00004 **SELECT** Products.\*, Categories.CategoryName00005 **FROM** Categories **INNER JOIN** Products **ON** Categories.CategoryID = Products.CategoryID00006 **WHERE** (((Products.Discontinued)=0))**Triggers**

None

**Indices**

None

**[dbo].[Category Sales for 1997]**

'View' in Database 'Northwind'

**Columns**

ID	Name	DataType	C	F	F	I	N	N	P	P
			O	I	K	D	R	U	K	R
1	CategoryName	nvarchar(15)								
2	CategorySales	money						X		

**Script:**

00001

00002 **create view** "Category Sales for 1997" **AS**

00003

00004 **SELECT** "Product Sales for 1997".CategoryName, **Sum**("Product Sales for 1997".ProductSales) **AS** CategorySales00005 **FROM** "Product Sales for 1997"00006 **GROUP BY** "Product Sales for 1997".CategoryName**Triggers**

None

**Indices**

None

**[dbo].[Current Product List]**

'View' in Database 'Northwind'

**Columns**

ID	Name	DataType	C	F	F	I	N	N	P	P
			O	I	K	D	R	U	K	R
1	ProductID	int				X				
2	ProductName	nvarchar(40)								

**Script:**

```
00001
00002 create view "Current Product List" AS
00003
00004 SELECT Product_List.ProductID, Product_List.ProductName
00005 FROM Products AS Product_List
00006 WHERE (((Product_List.Discontinued)=0))
00007 --ORDER BY Product_List.ProductName
```

**Triggers**

None

**Indices**

None

**[dbo].[Customer and Suppliers by City]**

'View' in Database 'Northwind'

**Columns**

ID	Name	DataType	C	F	F	I	N	N	P	P
			O	I	K	D	R	U	K	R
1	City	nvarchar(15)						X		
2	CompanyName	nvarchar(40)								
3	ContactName	nvarchar(30)						X		
4	Relationship	varchar(9)								

**Script:**

00001

00002 **create view** "Customer and Suppliers by City" **AS**

00003

00004 **SELECT** City, CompanyName, ContactName, 'Customers' **AS** Relationship00005 **FROM** Customers00006 **UNION SELECT** City, CompanyName, ContactName, 'Suppliers'00007 **FROM** Suppliers00008 **--ORDER BY** City, CompanyName**Triggers**

None

**Indices**

None

**[dbo].[Invoices]**

'View' in Database 'Northwind'

**Columns**

ID	Name	DataType	C	F	F	I	N	N	P	P
			O	I	K	D	R	U	K	R
1	ShipName	nvarchar(40)						X		
2	ShipAddress	nvarchar(60)						X		
3	ShipCity	nvarchar(15)						X		
4	ShipRegion	nvarchar(15)						X		
5	ShipPostalCode	nvarchar(10)						X		
6	ShipCountry	nvarchar(15)						X		
7	CustomerID	nchar(5)						X		
8	CustomerName	nvarchar(40)								
9	Address	nvarchar(60)						X		
10	City	nvarchar(15)						X		
11	Region	nvarchar(15)						X		
12	PostalCode	nvarchar(10)						X		
13	Country	nvarchar(15)						X		
14	Salesperson	nvarchar(31)								
15	OrderID	int								
16	OrderDate	datetime						X		
17	RequiredDate	datetime						X		
18	ShippedDate	datetime						X		
19	ShipperName	nvarchar(40)								
20	ProductID	int								
21	ProductName	nvarchar(40)								
22	UnitPrice	money								
23	Quantity	smallint								
24	Discount	real								
25	ExtendedPrice	money						X		
26	Freight	money						X		

**Script:**

```

00001
00002 create view Invoices AS
00003
00004 SELECT Orders.ShipName, Orders.ShipAddress, Orders.ShipCity, Orders.ShipRegion, Orders.ShipPostalCode,
00005 Orders.ShipCountry, Orders.CustomerID, Customers.CompanyName AS CustomerName, Customers.Address,
00006 » Customers.City,
00007 Customers.Region, Customers.PostalCode, Customers.Country,
00008 (FirstName + ' ' + LastName) AS Salesperson,
00009 Orders.OrderID, Orders.OrderDate, Orders.RequiredDate, Orders.ShippedDate, Shippers.CompanyName As
00010 » ShipperName,
00011 "Order Details".ProductID, Products.ProductName, "Order Details".UnitPrice, "Order Details".Quantity,
00012 "Order Details".Discount,
00013 (CONVERT(money,("Order Details".UnitPrice*Quantity*(1-Discount)/100))*100) AS ExtendedPrice, Orders.Freight
00014 » AS Freight
00015 FROM Shippers INNER JOIN
00016 (Products INNER JOIN
00017 (
00018 (Employees INNER JOIN
00019 (Customers INNER JOIN Orders ON Customers.CustomerID =
00020 » Orders.CustomerID)
00021 ON Employees.EmployeeID = Orders.EmployeeID)
00022 INNER JOIN "Order Details" ON Orders.OrderID = "Order Details".OrderID)
00023 ON Products.ProductID = "Order Details".ProductID)
00024 ON Shippers.ShipperID = Orders.ShipVia

```

**Triggers**

None

**Indices**

None

**[dbo].[Order Details Extended]**

'View' in Database 'Northwind'

**Columns**

ID	Name	DataType	C	F	F	I	N	N	P	P
			O	I	K	D	R	U	K	R
1	OrderID	int								
2	ProductID	int								
3	ProductName	nvarchar(40)								
4	UnitPrice	money								
5	Quantity	smallint								
6	Discount	real								
7	ExtendedPrice	money							X	

**Script:**

00001

00002 **create view** "Order Details Extended" **AS**

00003

00004 **SELECT** "Order Details".OrderID, "Order Details".ProductID, Products.ProductName,

00005 "Order Details".UnitPrice, "Order Details".Quantity, "Order Details".Discount,

00006 (CONVERT(money,("Order Details".UnitPrice\*Quantity\*(1-Discount)/100))\*100) **AS** ExtendedPrice00007 **FROM** Products **INNER JOIN** "Order Details" **ON** Products.ProductID = "Order Details".ProductID00008 **--ORDER BY** "Order Details".OrderID**Triggers**

None

**Indices**

None

**[dbo].[Order Subtotals]**

'View' in Database 'Northwind'

**Columns**

ID	Name	DataType	C	F	F	I	N	N	P	P
			O	I	K	D	R	U	K	R
1	OrderID	int								
2	Subtotal	money						X		

**Script:**

00001

00002 **create view** "Order Subtotals" **AS**

00003

00004 **SELECT** "Order Details".OrderID, **Sum**(**CONVERT**(**money**,("Order Details".UnitPrice\*Quantity\*(**1-Discount**)/**100**))\***1**  
» **00) AS Subtotal**00005 **FROM** "Order Details"00006 **GROUP BY** "Order Details".OrderID**Triggers**

None

**Indices**

None

**[dbo].[Orders Qry]**

'View' in Database 'Northwind'

**Columns**

ID	Name	DataType	C	F	F	I	N	N	P	P
			O	I	K	D	R	U	K	R
1	OrderID	int								
2	CustomerID	nchar(5)						X		
3	EmployeeID	int						X		
4	OrderDate	datetime						X		
5	RequiredDate	datetime						X		
6	ShippedDate	datetime						X		
7	ShipVia	int						X		
8	Freight	money						X		
9	ShipName	nvarchar(40)						X		
10	ShipAddress	nvarchar(60)						X		
11	ShipCity	nvarchar(15)						X		
12	ShipRegion	nvarchar(15)						X		
13	ShipPostalCode	nvarchar(10)						X		
14	ShipCountry	nvarchar(15)						X		
15	CompanyName	nvarchar(40)								
16	Address	nvarchar(60)						X		
17	City	nvarchar(15)						X		
18	Region	nvarchar(15)						X		
19	PostalCode	nvarchar(10)						X		
20	Country	nvarchar(15)						X		

**Script:**

```

00001
00002 create view "Orders Qry" AS
00003
00004 SELECT Orders.OrderID, Orders.CustomerID, Orders.EmployeeID, Orders.OrderDate, Orders.RequiredDate,
00005 Orders.ShippedDate, Orders.ShipVia, Orders.Freight, Orders.ShipName, Orders.ShipAddress, Orders.ShipCity,
00006 Orders.ShipRegion, Orders.ShipPostalCode, Orders.ShipCountry,
00007 Customers.CompanyName, Customers.Address, Customers.City, Customers.Region, Customers.PostalCode,
    » Customers.Country
00008 FROM Customers INNER JOIN Orders ON Customers.CustomerID = Orders.CustomerID

```

**Triggers**

None

**Indices**

None

**[dbo].[Product Sales for 1997]**

'View' in Database 'Northwind'

**Columns**

ID	Name	DataType	C	F	F	I	N	N	P	P
			O	I	K	D	R	U	K	R
1	CategoryName	nvarchar(15)								
2	ProductName	nvarchar(40)								
3	ProductSales	money						X		

**Script:**

00001

00002 **create view** "Product Sales for 1997" **AS**

00003

00004 **SELECT** Categories.CategoryName, Products.ProductName,00005 **Sum**(CONVERT(money,("Order Details".UnitPrice\*Quantity\*(1-Discount)/100))\*100) **AS ProductSales**00006 **FROM** (Categories **INNER JOIN** Products **ON** Categories.CategoryID = Products.CategoryID)00007 **INNER JOIN** (Orders00008 **INNER JOIN** "Order Details" **ON** Orders.OrderID = "Order Details".OrderID)00009 **ON** Products.ProductID = "Order Details".ProductID00010 **WHERE** (((Orders.ShippedDate) **Between** '19970101' **And** '19971231'))00011 **GROUP BY** Categories.CategoryName, Products.ProductName**Triggers**

None

**Indices**

None

**[dbo].[Products Above Average Price]**

'View' in Database 'Northwind'

**Columns**

ID	Name	DataType	C	F	F	I	N	N	P	P
			O	I	K	D	R	U	K	R
1	ProductName	nvarchar(40)								
2	UnitPrice	money						X		

**Script:**

00001

00002 **create view** "Products Above Average Price" **AS**

00003

00004 **SELECT** Products.ProductName, Products.UnitPrice00005 **FROM** Products00006 **WHERE** Products.UnitPrice > (SELECT AVG(UnitPrice) From Products)00007 **--ORDER BY Products.UnitPrice DESC****Triggers**

None

**Indices**

None

**[dbo].[Products by Category]**

'View' in Database 'Northwind'

**Columns**

ID	Name	DataType	C	F	F	I	N	N	P	P
			O	I	K	D	R	U	K	R
1	CategoryName	nvarchar(15)								
2	ProductName	nvarchar(40)								
3	QuantityPerUnit	nvarchar(20)						X		
4	UnitsInStock	smallint						X		
5	Discontinued	bit								

**Script:**

00001

00002 **create view** "Products by Category" **AS**

00003

00004 **SELECT** Categories.CategoryName, Products.ProductName, Products.QuantityPerUnit, Products.UnitsInStock,  
» Products.Discontinued00005 **FROM** Categories **INNER JOIN** Products **ON** Categories.CategoryID = Products.CategoryID00006 **WHERE** Products.Discontinued <> 100007 **--ORDER BY** Categories.CategoryName, Products.ProductName**Triggers**

None

**Indices**

None

**[dbo].[Quarterly Orders]**

'View' in Database 'Northwind'

**Columns**

ID	Name	DataType	C	F	F	I	N	N	P	P
			O	I	K	D	R	U	K	R
1	CustomerID	nchar(5)						X		
2	CompanyName	nvarchar(40)						X		
3	City	nvarchar(15)						X		
4	Country	nvarchar(15)						X		

**Script:**

00001

00002 create view "Quarterly Orders" AS

00003

00004 SELECT DISTINCT Customers.CustomerID, Customers.CompanyName, Customers.City, Customers.Country

00005 FROM Customers RIGHT JOIN Orders ON Customers.CustomerID = Orders.CustomerID

00006 WHERE Orders.OrderDate BETWEEN '19970101' And '19971231'

**Triggers**

None

**Indices**

None

**[dbo].[Sales by Category]**

'View' in Database 'Northwind'

**Columns**

ID	Name	DataType	C	F	F	I	N	N	P	P
			O	I	K	D	R	U	K	R
1	CategoryID	int								
2	CategoryName	nvarchar(15)								
3	ProductName	nvarchar(40)								
4	ProductSales	money						X		

**Script:**

```

00001
00002 create view "Sales by Category" AS
00003
00004 SELECT Categories.CategoryID, Categories.CategoryName, Products.ProductName,
00005     Sum("Order Details Extended".ExtendedPrice) AS ProductSales
00006 FROM     Categories INNER JOIN
00007         (Products INNER JOIN
00008             (Orders INNER JOIN "Order Details Extended" ON Orders.OrderID = "Order Details Extended
00009             »     d".OrderID)
00010             ON Products.ProductID = "Order Details Extended".ProductID)
00011     ON Categories.CategoryID = Products.CategoryID
00012 WHERE Orders.OrderDate BETWEEN '19970101' And '19971231'
00013 GROUP BY Categories.CategoryID, Categories.CategoryName, Products.ProductName

```

**Triggers**

None

**Indices**

None

**[dbo].[Sales Totals by Amount]**

'View' in Database 'Northwind'

**Columns**

ID	Name	DataType	C	F	F	I	N	N	P	P
			O	I	K	D	R	U	K	R
1	SaleAmount	money						X		
2	OrderID	int								
3	CompanyName	nvarchar(40)								
4	ShippedDate	datetime						X		

**Script:**

00001

00002 **create view** "Sales Totals by Amount" **AS**

00003

00004 **SELECT** "Order Subtotals".Subtotal **AS** SaleAmount, Orders.OrderID, Customers.CompanyName,

» Orders.ShippedDate

00005 **FROM** Customers **INNER JOIN**00006 (Orders **INNER JOIN** "Order Subtotals" **ON** Orders.OrderID = "Order Subtotals".OrderID)00007 **ON** Customers.CustomerID = Orders.CustomerID00008 **WHERE** ("Order Subtotals".Subtotal >2500) **AND** (Orders.ShippedDate **BETWEEN** '19970101' **And** '19971231')**Triggers**

None

**Indices**

None

**[dbo].[Summary of Sales by Quarter]**

'View' in Database 'Northwind'

**Columns**

ID	Name	DataType	C	F	F	I	N	N	P	P
			O	I	K	D	R	U	K	R
1	ShippedDate	datetime						X		
2	OrderID	int								
3	Subtotal	money						X		

**Script:**

00001

00002 **create view** "Summary of Sales by Quarter" **AS**

00003

00004 **SELECT** Orders.ShippedDate, Orders.OrderID, "Order Subtotals".Subtotal00005 **FROM** Orders **INNER JOIN** "Order Subtotals" **ON** Orders.OrderID = "Order Subtotals".OrderID00006 **WHERE** Orders.ShippedDate **IS NOT NULL**00007 **--ORDER BY** Orders.ShippedDate**Triggers**

None

**Indices**

None

**[dbo].[Summary of Sales by Year]**

'View' in Database 'Northwind'

**Columns**

ID	Name	DataType	C	F	F	I	N	N	P	P
			O	I	K	D	R	U	K	R
1	ShippedDate	datetime						X		
2	OrderID	int								
3	Subtotal	money						X		

**Script:**

00001

00002 **create view** "Summary of Sales by Year" **AS**

00003

00004 **SELECT** Orders.ShippedDate, Orders.OrderID, "Order Subtotals".Subtotal00005 **FROM** Orders **INNER JOIN** "Order Subtotals" **ON** Orders.OrderID = "Order Subtotals".OrderID00006 **WHERE** Orders.ShippedDate **IS NOT NULL**00007 **--ORDER BY** Orders.ShippedDate**Triggers**

None

**Indices**

None

**[dbo].[CustOrderHist]**

'StoredProcedure' in Database 'Northwind'

**Script:**

```
00001 CREATE PROCEDURE CustOrderHist @CustomerID nchar(5)
00002 AS
00003
00004 SELECT ProductName, Total=SUM(Quantity)
00005 FROM Products P, [Order Details] OD, Orders O, Customers C
00006 WHERE C.CustomerID = @CustomerID
00007 AND C.CustomerID = O.CustomerID AND O.OrderID = OD.OrderID AND OD.ProductID = P.ProductID
00008 GROUP BY ProductName
```

**Properties**

Property	Value
Assembly Name	
Class Name	
Create Date	27/07/2007 21:11:11
Date Last Modified	27/07/2007 21:11:11
Execution Context	Caller
For Replication	False
Implementation Type	TransactSql
Method Name	
Recompile	False
Startup	False

**Explicit Permissions:**

None

**[dbo].[CustOrdersDetail]**

'StoredProcedure' in Database 'Northwind'

**Script:**

```
00001
00002 CREATE PROCEDURE CustOrdersDetail @OrderID int
00003 AS
00004
00005 SELECT ProductName,
00006    UnitPrice=ROUND(Od.UnitPrice, 2),
00007    Quantity,
00008    Discount=CONVERT(int, Discount * 100),
00009    ExtendedPrice=ROUND(CONVERT(money, Quantity * (1 - Discount) * Od.UnitPrice), 2)
00010 FROM Products P, [Order Details] Od
00011 WHERE Od.ProductID = P.ProductID and Od.OrderID = @OrderID
```

**Properties**

Property	Value
Assembly Name	
Class Name	
Create Date	27/07/2007 21:11:11
Date Last Modified	27/07/2007 21:11:11
Execution Context	Caller
For Replication	False
Implementation Type	TransactSql
Method Name	
Recompile	False
Startup	False

**Explicit Permissions:**

None

**[dbo].[CustOrdersOrders]**

'StoredProcedure' in Database 'Northwind'

**Script:**

```
00001
00002 CREATE PROCEDURE CustOrdersOrders @CustomerID nchar(5)
00003 AS
00004
00005 SELECT OrderID,
00006    OrderDate,
00007    RequiredDate,
00008    ShippedDate
00009 FROM Orders
00010 WHERE CustomerID = @CustomerID
00011 ORDER BY OrderID
```

**Properties**

Property	Value
Assembly Name	
Class Name	
Create Date	27/07/2007 21:11:11
Date Last Modified	27/07/2007 21:11:11
Execution Context	Caller
For Replication	False
Implementation Type	TransactSql
Method Name	
Recompile	False
Startup	False

**Explicit Permissions:**

None

**[dbo].[Employee Sales by Country]**

'StoredProcedure' in Database 'Northwind'

**Script:**

```
00001
00002 create procedure "Employee Sales by Country"
00003 @Beginning_Date DateTime, @Ending_Date DateTime AS
00004
00005 SELECT Employees.Country, Employees.LastName, Employees.FirstName, Orders.ShippedDate, Orders.OrderID,
    » "Order Subtotals".Subtotal AS SaleAmount
00006 FROM Employees INNER JOIN
00007 (Orders INNER JOIN "Order Subtotals" ON Orders.OrderID = "Order Subtotals".OrderID)
00008 ON Employees.EmployeeID = Orders.EmployeeID
00009 WHERE Orders.ShippedDate Between @Beginning_Date And @Ending_Date
```

**Properties**

Property	Value
Assembly Name	
Class Name	
Create Date	27/07/2007 21:11:05
Date Last Modified	27/07/2007 21:11:05
Execution Context	Caller
For Replication	False
Implementation Type	TransactSql
Method Name	
Recompile	False
Startup	False

**Explicit Permissions:**

None

**[dbo].[Sales by Year]**

'StoredProcedure' in Database 'Northwind'

**Script:**

```
00001
00002 create procedure "Sales by Year"
00003   @Beginning_Date DateTime, @Ending_Date DateTime AS
00004
00005 SELECT Orders.ShippedDate, Orders.OrderID, "Order Subtotals".Subtotal, DATENAME(yy,ShippedDate) AS Year
00006 FROM Orders INNER JOIN "Order Subtotals" ON Orders.OrderID = "Order Subtotals".OrderID
00007 WHERE Orders.ShippedDate Between @Beginning_Date And @Ending_Date
```

**Properties**

Property	Value
Assembly Name	
Class Name	
Create Date	27/07/2007 21:11:05
Date Last Modified	27/07/2007 21:11:05
Execution Context	Caller
For Replication	False
Implementation Type	TransactSql
Method Name	
Recompile	False
Startup	False

**Explicit Permissions:**

None

**[dbo].[SalesByCategory]**

'StoredProcedure' in Database 'Northwind'

**Script:**

```

00001 CREATE PROCEDURE SalesByCategory
00002   @CategoryName nvarchar(15), @OrdYear nvarchar(4) = '1998'
00003 AS
00004
00005 IF @OrdYear != '1996' AND @OrdYear != '1997' AND @OrdYear != '1998'
00006 BEGIN
00007   SELECT @OrdYear = '1998'
00008 END
00009
00010 SELECT ProductName,
00011   TotalPurchase=ROUND(SUM(CONVERT(decimal(14,2), OD.Quantity * (1-OD.Discount) * OD.UnitPrice)), 0)
00012 FROM [Order Details] OD, Orders O, Products P, Categories C
00013 WHERE OD.OrderID = O.OrderID
00014   AND OD.ProductID = P.ProductID
00015   AND P.CategoryID = C.CategoryID
00016   AND C.CategoryName = @CategoryName
00017   AND SUBSTRING(CONVERT(nvarchar(22), O.OrderDate, 111), 1, 4) = @OrdYear
00018 GROUP BY ProductName
00019 ORDER BY ProductName

```

**Properties**

Property	Value
Assembly Name	
Class Name	
Create Date	27/07/2007 21:11:11
Date Last Modified	27/07/2007 21:11:11
Execution Context	Caller
For Replication	False
Implementation Type	TransactSql
Method Name	
Recompile	False
Startup	False

**Explicit Permissions:**

None

**[dbo].[Ten Most Expensive Products]**

'StoredProcedure' in Database 'Northwind'

**Script:**

```
00001
00002 create procedure "Ten Most Expensive Products" AS
00003
00004 SET ROWCOUNT 10
00005 SELECT Products.ProductName AS TenMostExpensiveProducts, Products.UnitPrice
00006 FROM Products
00007 ORDER BY Products.UnitPrice DESC
```

**Properties**

Property	Value
Assembly Name	
Class Name	
Create Date	27/07/2007 21:11:05
Date Last Modified	27/07/2007 21:11:05
Execution Context	Caller
For Replication	False
Implementation Type	TransactSql
Method Name	
Recompile	False
Startup	False

**Explicit Permissions:**

None

**[sys].[dm\_db\_index\_operational\_stats]**

'UserDefinedFunction' in Database 'Northwind'

**Properties**

Property	Value
Ansi Nulls Status	True
Assembly Name	
Body Start Index	188
Class Name	
Create Date	14/10/2005 01:40:15
Data Type	
Data Type Schema	
Date Last Modified	14/10/2005 01:40:15
Function Type	Inline
I D	-500638940
Implementation Type	TransactSql
Is Deterministic	False
Is Encrypted	False
Is Schema Bound	False
Is System Object	True
Method Name	
Returns Null On Null Input	
System Type	
Table Variable Name	

**Script:**

```

00001 create function sys.dm_db_index_operational_stats
00002 (
00003     @Databaseld          SMALLINT          = 0,
00004     @TableId             INT                = 0,
00005     @IndexId             INT                = -1,
00006     @PartitionNumber INT                = 0
00007 )
00008 returns table
00009 as
00010     return select * from OpenRowset
00011     (
00012         VirtualIndexStats,
00013         @Databaseld,
00014         @TableId,
00015         @IndexId,
00016         @PartitionNumber)
00017

```

**Parameters**

Name	Type	Default
@Databaseld	smallint	no default
@TableId	int	no default
@IndexId	int	no default
@PartitionNumber	int	no default

**[sys].[dm\_db\_index\_physical\_stats]**

'UserDefinedFunction' in Database 'Northwind'

**Properties**

Property	Value
Ansi Nulls Status	True
Assembly Name	
Body Start Index	225
Class Name	
Create Date	14/10/2005 01:40:16
Data Type	
Data Type Schema	
Date Last Modified	14/10/2005 01:40:16
Function Type	Inline
I D	-333557461
Implementation Type	TransactSql
Is Deterministic	False
Is Encrypted	False
Is Schema Bound	False
Is System Object	True
Method Name	
Returns Null On Null Input	
System Type	
Table Variable Name	

**Script:**

```

00001 create function sys.dm_db_index_physical_stats
00002 (
00003     @Databaseld          SMALLINT          = 0,
00004     @ObjectId             INT                = 0,
00005     @IndexId              INT                = -1,
00006     @PartitionNumber INT    = 0,
00007     @Mode                  nvarchar(20)     = NULL
00008 )
00009 returns table
00010 as
00011     return select * from OpenRowset
00012     (
00013         IndexAnalysis,
00014         @Databaseld,
00015         @ObjectId,
00016         @IndexId,
00017         @PartitionNumber,
00018         @Mode
00019     )

```

**Parameters**

Name	Type	Default
@Databaseld	smallint	no default
@ObjectId	int	no default
@IndexId	int	no default
@PartitionNumber	int	no default
@Mode	nvarchar(20)	no default

**[sys].[dm\_db\_missing\_index\_columns]**

'UserDefinedFunction' in Database 'Northwind'

**Properties**

Property	Value
Ansi Nulls Status	True
Assembly Name	
Body Start Index	83
Class Name	
Create Date	14/10/2005 01:40:11
Data Type	
Data Type Schema	
Date Last Modified	14/10/2005 01:40:11
Function Type	Inline
I D	-963369315
Implementation Type	TransactSql
Is Deterministic	False
Is Encrypted	False
Is Schema Bound	False
Is System Object	True
Method Name	
Returns Null On Null Input	
System Type	
Table Variable Name	

**Script:**

```

00001 CREATE FUNCTION sys.dm_db_missing_index_columns (@handle int)
00002 RETURNS TABLE
00003 AS
00004     RETURN SELECT * FROM OPENROWSET(TABLE MISSING_IDX_COLUMNS, @handle)

```

**Parameters**

Name	Type	Default
@handle	int	no default

**[sys].[dm\_exec\_cached\_plan\_dependent\_objects]**

'UserDefinedFunction' in Database 'Northwind'

**Properties**

Property	Value
Ansi Nulls Status	True
Assembly Name	
Body Start Index	206
Class Name	
Create Date	14/10/2005 01:40:47
Data Type	
Data Type Schema	
Date Last Modified	14/10/2005 01:40:47
Function Type	Table
I D	-1050437832
Implementation Type	TransactSql
Is Deterministic	False
Is Encrypted	False
Is Schema Bound	False
Is System Object	True
Method Name	
Returns Null On Null Input	
System Type	
Table Variable Name	@tab

**Script:**

```

00001
00002 create function sys.dm_exec_cached_plan_dependent_objects
00003 (
00004     @planhandle varbinary(64)
00005 )
00006 returns @tab table(
00007     usecounts Int,
00008     memory_object_address varbinary(8),
00009     cacheobjtype nvarchar(50))
00010 as
00011 begin
00012     insert @tab
00013     select * from OpenRowset(TABLE SYSDMEXECCACHEDPLANDEPENDENTOBJECTS, @planhandle)
00014
00015     return
00016 end -- dm_exec_cached_plan_dependent_objects

```

**Parameters**

Name	Type	Default
@planhandle	varbinary(64)	no default

**[sys].[dm\_exec\_cursors]**

'UserDefinedFunction' in Database 'Northwind'

**Properties**

Property	Value
Ansi Nulls Status	True
Assembly Name	
Body Start Index	69
Class Name	
Create Date	14/10/2005 01:40:48
Data Type	
Data Type Schema	
Date Last Modified	14/10/2005 01:40:48
Function Type	Inline
I D	-838405656
Implementation Type	TransactSql
Is Deterministic	False
Is Encrypted	False
Is Schema Bound	False
Is System Object	True
Method Name	
Returns Null On Null Input	
System Type	
Table Variable Name	

**Script:**

```

00001 CREATE FUNCTION sys.dm_exec_cursors (@spid int)
00002 RETURNS table
00003 AS
00004     RETURN SELECT *
00005     FROM OpenRowSet(TABLE DM_EXEC_CURSORS, @spid)

```

**Parameters**

Name	Type	Default
@spid	int	no default

**[sys].[dm\_exec\_plan\_attributes]**

'UserDefinedFunction' in Database 'Northwind'

**Properties**

Property	Value
Ansi Nulls Status	True
Assembly Name	
Body Start Index	88
Class Name	
Create Date	14/10/2005 01:40:46
Data Type	
Data Type Schema	
Date Last Modified	14/10/2005 01:40:46
Function Type	Inline
I D	-69842288
Implementation Type	TransactSql
Is Deterministic	False
Is Encrypted	False
Is Schema Bound	False
Is System Object	True
Method Name	
Returns Null On Null Input	
System Type	
Table Variable Name	

**Script:**

```

00001 CREATE FUNCTION sys.dm_exec_plan_attributes(@handle varbinary(64))
00002 RETURNS TABLE
00003 AS
00004     RETURN SELECT * FROM OPENROWSET(TABLE FNGETPLANATTRIBS, @handle)

```

**Parameters**

Name	Type	Default
@handle	varbinary(64)	no default

**[sys].[dm\_exec\_query\_plan]**

'UserDefinedFunction' in Database 'Northwind'

**Properties**

Property	Value
Ansi Nulls Status	True
Assembly Name	
Body Start Index	83
Class Name	
Create Date	14/10/2005 01:40:45
Data Type	
Data Type Schema	
Date Last Modified	14/10/2005 01:40:45
Function Type	Inline
I D	-430121581
Implementation Type	TransactSql
Is Deterministic	False
Is Encrypted	False
Is Schema Bound	False
Is System Object	True
Method Name	
Returns Null On Null Input	
System Type	
Table Variable Name	

**Script:**

```

00001 CREATE FUNCTION sys.dm_exec_query_plan(@handle varbinary(64))
00002 RETURNS TABLE
00003 AS
00004     RETURN SELECT * FROM OPENROWSET(TABLE FNGETQUERYPLAN, @handle)

```

**Parameters**

Name	Type	Default
@handle	varbinary(64)	no default

**[sys].[dm\_exec\_sql\_text]**

'UserDefinedFunction' in Database 'Northwind'

**Properties**

Property	Value
Ansi Nulls Status	True
Assembly Name	
Body Start Index	81
Class Name	
Create Date	14/10/2005 01:40:45
Data Type	
Data Type Schema	
Date Last Modified	14/10/2005 01:40:45
Function Type	Inline
I D	-245973352
Implementation Type	TransactSql
Is Deterministic	False
Is Encrypted	False
Is Schema Bound	False
Is System Object	True
Method Name	
Returns Null On Null Input	
System Type	
Table Variable Name	

**Script:**

```

00001 CREATE FUNCTION sys.dm_exec_sql_text(@handle varbinary(64))
00002 RETURNS TABLE
00003 AS
00004     RETURN SELECT * FROM OPENROWSET(TABLE FNGETSQL, @handle)

```

**Parameters**

Name	Type	Default
@handle	varbinary(64)	no default

**[sys].[dm\_io\_virtual\_file\_stats]**

'UserDefinedFunction' in Database 'Northwind'

**Properties**

Property	Value
Ansi Nulls Status	True
Assembly Name	
Body Start Index	117
Class Name	
Create Date	14/10/2005 01:40:20
Data Type	
Data Type Schema	
Date Last Modified	14/10/2005 01:40:20
Function Type	Inline
I D	-806439612
Implementation Type	TransactSql
Is Deterministic	False
Is Encrypted	False
Is Schema Bound	False
Is System Object	True
Method Name	
Returns Null On Null Input	
System Type	
Table Variable Name	

**Script:**

```

00001 CREATE FUNCTION sys.dm_io_virtual_file_stats
00002 (
00003     @Databaseld Int = -1,
00004     @Fileld Int = -1
00005 )
00006 RETURNS TABLE
00007 as
00008     RETURN SELECT * FROM OpenRowset(VirtualFileStats, @Databaseld, @Fileld)

```

**Parameters**

Name	Type	Default
@Databaseld	int	no default
@Fileld	int	no default

**[sys].[fn\_builtin\_permissions]**

'UserDefinedFunction' in Database 'Northwind'

**Properties**

Property	Value
Ansi Nulls Status	True
Assembly Name	
Body Start Index	105
Class Name	
Create Date	14/10/2005 01:39:34
Data Type	
Data Type Schema	
Date Last Modified	14/10/2005 01:39:34
Function Type	Inline
I D	-413116802
Implementation Type	TransactSql
Is Deterministic	False
Is Encrypted	False
Is Schema Bound	False
Is System Object	True
Method Name	
Returns Null On Null Input	
System Type	
Table Variable Name	

**Script:**

```

00001 CREATE FUNCTION sys.fn_builtin_permissions
00002     ( @level nvarchar(60) )
00003 RETURNS table
00004 AS
00005 RETURN (
00006     SELECT * FROM OpenRowset(BuiltinPermissions, @level)
00007 )

```

**Parameters**

Name	Type	Default
@level	nvarchar(60)	no default

**[sys].[fn\_cColvEntries\_80]**

'UserDefinedFunction' in Database 'Northwind'

**Properties**

Property	Value
Ansi Nulls Status	False
Assembly Name	
Body Start Index	106
Class Name	
Create Date	14/10/2005 01:43:08
Data Type	int
Data Type Schema	sys
Date Last Modified	14/10/2005 01:43:08
Function Type	Scalar
I D	-61545096
Implementation Type	TransactSql
Is Deterministic	False
Is Encrypted	False
Is Schema Bound	False
Is System Object	True
Method Name	
Returns Null On Null Input	
System Type	int
Table Variable Name	

**Script:**

```

00001 create function sys.fn_cColvEntries_80
00002   (@pubid uniqueidentifier, @artnick int)
00003   returns int
00004 as
00005 begin
00006   declare @cCols int
00007   declare @cMissing int
00008   declare @cExcluded int
00009   declare @sync_objid int
00010
00011   select @sync_objid= sync_objid, @cMissing= missing_col_count, @cExcluded= excluded_col_count from
    »   dbo.sysmergearticles where nickname=@artnick and pubid=@pubid
00012   select @cCols= count(*) from sys.columns where object_id = @sync_objid
00013   return @cCols + @cMissing + @cExcluded
00014 end

```

**Parameters**

Name	Type	Default
@pubid	uniqueidentifier	no default
@artnick	int	no default

**[sys].[fn\_check\_object\_signatures]**

'UserDefinedFunction' in Database 'Northwind'

**Properties**

Property	Value
Ansi Nulls Status	True
Assembly Name	
Body Start Index	112
Class Name	
Create Date	14/10/2005 01:39:49
Data Type	
Data Type Schema	
Date Last Modified	14/10/2005 01:39:49
Function Type	Inline
I D	-234465429
Implementation Type	TransactSql
Is Deterministic	False
Is Encrypted	False
Is Schema Bound	False
Is System Object	True
Method Name	
Returns Null On Null Input	
System Type	
Table Variable Name	

**Script:**

```

00001 CREATE FUNCTION sys.fn_check_object_signatures(@class SYSNAME, @thumbprint VARBINARY(20))
00002 RETURNS TABLE
00003 AS
00004     RETURN
00005         SELECT o.type_desc                type,      -- OBJECT
00006                o.object_id              entity_id,
00007                CASE WHEN is_objectsigned('OBJECT', o.object_id, @class, @thumbprint) IS NULL
00008                THEN 0 ELSE 1 END         is_signed,
00009                CASE WHEN is_objectsigned('OBJECT', o.object_id, @class, @thumbprint) = 1 THEN 1
00010                ELSE 0 END               is_signature_valid
00011         FROM sys.objects o
00012         LEFT JOIN sys.crypt_properties p ON o.object_id = p.major_id
00013         WHERE sys.fn_is_signable_object(o.type) = 1
00014         GROUP BY o.object_id, o.type_desc
00015     UNION
00016     SELECT 'ASSEMBLY'                    class,     -- ASSEMBLY
00017            s.assembly_id                  entity_id,
00018            CASE WHEN is_objectsigned('ASSEMBLY', s.assembly_id, @class, @thumbprint) IS NULL
00019            THEN 0 ELSE 1 END              is_signed,
00020            CASE WHEN is_objectsigned('ASSEMBLY', s.assembly_id, @class, @thumbprint) = 1 THEN
00021            » 1                            is_signature_valid
00021            ELSE 0 END
00022         FROM sys.assemblies s
00023         LEFT JOIN sys.crypt_properties p ON s.assembly_id = p.major_id
00024         GROUP BY s.assembly_id

```

**Parameters**

Name	Type	Default
@class	sysname	no default
@thumbprint	varbinary(20)	no default

**[sys].[fn\_dblog]**

'UserDefinedFunction' in Database 'Northwind'

**Properties**

Property	Value
Ansi Nulls Status	True
Assembly Name	
Body Start Index	127
Class Name	
Create Date	14/10/2005 01:40:41
Data Type	
Data Type Schema	
Date Last Modified	14/10/2005 01:40:41
Function Type	Inline
I D	-857107446
Implementation Type	TransactSql
Is Deterministic	False
Is Encrypted	False
Is Schema Bound	False
Is System Object	True
Method Name	
Returns Null On Null Input	
System Type	
Table Variable Name	

**Script:**

```

00001 create function sys.fn_dblog
00002 (
00003     @start    nvarchar (25) = NULL,
00004     @end      nvarchar (25) = NULL
00005 )
00006
00007 returns table
00008 as
00009     return select
00010         [Current LSN],
00011         [Operation],
00012         [Context],
00013         [Transaction ID],
00014         [Tag Bits],
00015         [Log Record Fixed Length],
00016         [Log Record Length],
00017         [Previous LSN],
00018         [Flag Bits],
00019         [AllocUnitId],
00020         [AllocUnitName],
00021         [Page ID],
00022         [Slot ID],
00023         [Previous Page LSN],
00024         [PartitionId],
00025         [RowFlags],
00026         [Num Elements],
00027         [Offset in Row],
00028         [Checkpoint Begin],
00029         [CHKPT Begin DB Version],
00030         [Max XDESID],
00031         [Num Transactions],
00032         [Checkpoint End],
00033         [CHKPT End DB Version],
00034         [Minimum LSN],
00035         [Dirty Pages],
00036         [Oldest Replicated Begin LSN],
00037         [Next Replicated End LSN],
00038         [Last Distributed Backup End LSN],
00039         [Last Distributed End LSN],
00040         [Server UID],
00041         [UID],

```

```

00042      [SPID],
00043      [Beginlog Status],
00044      [Begin Time],
00045      [Transaction Name],
00046      [Transaction SID],
00047      [End Time],
00048      [Transaction Begin],
00049      [Replicated Records],
00050      [Oldest Active LSN],
00051      [Server Name],
00052      [Database Name],
00053      [Mark Name],
00054      [Master XDESID],
00055      [Master DBID],
00056      [Prelog Begin LSN],
00057      [Prepare Time],
00058      [Virtual Clock],
00059      [Previous Savepoint],
00060      [Savepoint Name],
00061      [Rowbits First Bit],
00062      [Rowbits Bit Count],
00063      [Rowbits Bit Value],
00064      [Number of Locks],
00065      [Lock Information],
00066      [LSN before writes],
00067      [Pages Written],
00068      [Data Pages Delta],
00069      [Reserved Pages Delta],
00070      [Used Pages Delta],
00071      [Data Rows Delta],
00072      [Command Type],
00073      [Publication ID],
00074      [Article ID],
00075      [Partial Status],
00076      [Command],
00077      [Byte Offset],
00078      [New Value],
00079      [Old Value],
00080      [New Split Page],
00081      [Rows Deleted],
00082      [Bytes Freed],
00083      [CI Table Id],
00084      [CI Index Id],
00085      [FileGroup ID],
00086      [Meta Status],
00087      [File Status],
00088      [File ID],
00089      [Physical Name],
00090      [Logical Name],
00091      [Format LSN],
00092      [RowsetId],
00093      [TextPtr],
00094      [Column Offset],
00095      [Flags],
00096      [Text Size],
00097      [Offset],
00098      [Old Size],
00099      [New Size],
00100      [Description],
00101      [RowLog Contents 0],
00102      [RowLog Contents 1],
00103      [RowLog Contents 2],
00104      [RowLog Contents 3],
00105      [RowLog Contents 4],
00106      [Log Record]
00107      from OpenRowset (DBLog, @start, @end, NULL, 1,
00108      NULL,NULL,NULL,NULL,NULL,NULL,NULL,
00109      NULL,NULL,NULL,NULL,NULL,NULL,NULL,
00110      NULL,NULL,NULL,NULL,NULL,NULL,NULL,

```

```
00111 NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,
00112 NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,
00113 NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,
00114 NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,
00115 NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL)
```

**Parameters**

Name	Type	Default
@start	nvarchar(25)	no default
@end	nvarchar(25)	no default

**[sys].[fn\_dump\_dblog]**

'UserDefinedFunction' in Database 'Northwind'

**Properties**

Property	Value
Ansi Nulls Status	True
Assembly Name	
Body Start Index	2553
Class Name	
Create Date	14/10/2005 01:40:41
Data Type	
Data Type Schema	
Date Last Modified	14/10/2005 01:40:41
Function Type	Inline
I D	-1024577103
Implementation Type	TransactSql
Is Deterministic	False
Is Encrypted	False
Is Schema Bound	False
Is System Object	True
Method Name	
Returns Null On Null Input	
System Type	
Table Variable Name	

**Script:**

```

00001 create function sys.fn_dump_dblog
00002 (
00003     @start    nvarchar (25) = NULL,
00004     @end      nvarchar (25) = NULL,
00005     @devtype  nvarchar (260) = NULL, -- NULL(DISK) | DISK | TAPE | VIRTUAL_DEVICE
00006     @seqnum   Int          = 1,
00007     @fname1   nvarchar (260) = NULL,
00008     @fname2   nvarchar (260) = NULL,
00009     @fname3   nvarchar (260) = NULL,
00010     @fname4   nvarchar (260) = NULL,
00011     @fname5   nvarchar (260) = NULL,
00012     @fname6   nvarchar (260) = NULL,
00013     @fname7   nvarchar (260) = NULL,
00014     @fname8   nvarchar (260) = NULL,
00015     @fname9   nvarchar (260) = NULL,
00016     @fname10  nvarchar (260) = NULL,
00017     @fname11  nvarchar (260) = NULL,
00018     @fname12  nvarchar (260) = NULL,
00019     @fname13  nvarchar (260) = NULL,
00020     @fname14  nvarchar (260) = NULL,
00021     @fname15  nvarchar (260) = NULL,
00022     @fname16  nvarchar (260) = NULL,
00023     @fname17  nvarchar (260) = NULL,
00024     @fname18  nvarchar (260) = NULL,
00025     @fname19  nvarchar (260) = NULL,
00026     @fname20  nvarchar (260) = NULL,
00027     @fname21  nvarchar (260) = NULL,
00028     @fname22  nvarchar (260) = NULL,
00029     @fname23  nvarchar (260) = NULL,
00030     @fname24  nvarchar (260) = NULL,
00031     @fname25  nvarchar (260) = NULL,
00032     @fname26  nvarchar (260) = NULL,
00033     @fname27  nvarchar (260) = NULL,
00034     @fname28  nvarchar (260) = NULL,
00035     @fname29  nvarchar (260) = NULL,
00036     @fname30  nvarchar (260) = NULL,
00037     @fname31  nvarchar (260) = NULL,
00038     @fname32  nvarchar (260) = NULL,
00039     @fname33  nvarchar (260) = NULL,
00040     @fname34  nvarchar (260) = NULL,
00041     @fname35  nvarchar (260) = NULL,

```

```
00042      @fname36 nvarchar (260) = NULL,
00043      @fname37 nvarchar (260) = NULL,
00044      @fname38 nvarchar (260) = NULL,
00045      @fname39 nvarchar (260) = NULL,
00046      @fname40 nvarchar (260) = NULL,
00047      @fname41 nvarchar (260) = NULL,
00048      @fname42 nvarchar (260) = NULL,
00049      @fname43 nvarchar (260) = NULL,
00050      @fname44 nvarchar (260) = NULL,
00051      @fname45 nvarchar (260) = NULL,
00052      @fname46 nvarchar (260) = NULL,
00053      @fname47 nvarchar (260) = NULL,
00054      @fname48 nvarchar (260) = NULL,
00055      @fname49 nvarchar (260) = NULL,
00056      @fname50 nvarchar (260) = NULL,
00057      @fname51 nvarchar (260) = NULL,
00058      @fname52 nvarchar (260) = NULL,
00059      @fname53 nvarchar (260) = NULL,
00060      @fname54 nvarchar (260) = NULL,
00061      @fname55 nvarchar (260) = NULL,
00062      @fname56 nvarchar (260) = NULL,
00063      @fname57 nvarchar (260) = NULL,
00064      @fname58 nvarchar (260) = NULL,
00065      @fname59 nvarchar (260) = NULL,
00066      @fname60 nvarchar (260) = NULL,
00067      @fname61 nvarchar (260) = NULL,
00068      @fname62 nvarchar (260) = NULL,
00069      @fname63 nvarchar (260) = NULL,
00070      @fname64 nvarchar (260) = NULL
00071  )
00072
00073 returns table
00074 as
00075     return select
00076         [Current LSN],
00077         [Operation],
00078         [Context],
00079         [Transaction ID],
00080         [Tag Bits],
00081         [Log Record Fixed Length],
00082         [Log Record Length],
00083         [Previous LSN],
00084         [Flag Bits],
00085         [AllocUnitId],
00086         [AllocUnitName],
00087         [Page ID],
00088         [Slot ID],
00089         [Previous Page LSN],
00090         [PartitionId],
00091         [RowFlags],
00092         [Num Elements],
00093         [Offset in Row],
00094         [Checkpoint Begin],
00095         [CHKPT Begin DB Version],
00096         [Max XDESID],
00097         [Num Transactions],
00098         [Checkpoint End],
00099         [CHKPT End DB Version],
00100         [Minimum LSN],
00101         [Dirty Pages],
00102         [Oldest Replicated Begin LSN],
00103         [Next Replicated End LSN],
00104         [Last Distributed Backup End LSN],
00105         [Last Distributed End LSN],
00106         [Server UID],
00107         [UID],
00108         [SPID],
00109         [Beginlog Status],
00110         [Begin Time],
```

```

00111      [Transaction Name],
00112      [Transaction SID],
00113      [End Time],
00114      [Transaction Begin],
00115      [Replicated Records],
00116      [Oldest Active LSN],
00117      [Server Name],
00118      [Database Name],
00119      [Mark Name],
00120      [Master XDESID],
00121      [Master DBID],
00122      [Prelog Begin LSN],
00123      [Prepare Time],
00124      [Virtual Clock],
00125      [Previous Savepoint],
00126      [Savepoint Name],
00127      [Rowbits First Bit],
00128      [Rowbits Bit Count],
00129      [Rowbits Bit Value],
00130      [Number of Locks],
00131      [Lock Information],
00132      [LSN before writes],
00133      [Pages Written],
00134      [Data Pages Delta],
00135      [Reserved Pages Delta],
00136      [Used Pages Delta],
00137      [Data Rows Delta],
00138      [Command Type],
00139      [Publication ID],
00140      [Article ID],
00141      [Partial Status],
00142      [Command],
00143      [Byte Offset],
00144      [New Value],
00145      [Old Value],
00146      [New Split Page],
00147      [Rows Deleted],
00148      [Bytes Freed],
00149      [CI Table Id],
00150      [CI Index Id],
00151      [FileGroup ID],
00152      [Meta Status],
00153      [File Status],
00154      [File ID],
00155      [Physical Name],
00156      [Logical Name],
00157      [Format LSN],
00158      [RowsetId],
00159      [TextPtr],
00160      [Column Offset],
00161      [Flags],
00162      [Text Size],
00163      [Offset],
00164      [Old Size],
00165      [New Size],
00166      [Description],
00167      [RowLog Contents 0],
00168      [RowLog Contents 1],
00169      [RowLog Contents 2],
00170      [RowLog Contents 3],
00171      [RowLog Contents 4],
00172      [Log Record]
00173      from OpenRowset (DBLog, @start, @end, @devtype, @seqnum,
00174      @fname1,
00175      @fname2,
00176      @fname3,
00177      @fname4,
00178      @fname5,
00179      @fname6,

```

```

00180      @fname7,
00181      @fname8,
00182      @fname9,
00183      @fname10,
00184      @fname11,
00185      @fname12,
00186      @fname13,
00187      @fname14,
00188      @fname15,
00189      @fname16,
00190      @fname17,
00191      @fname18,
00192      @fname19,
00193      @fname20,
00194      @fname21,
00195      @fname22,
00196      @fname23,
00197      @fname24,
00198      @fname25,
00199      @fname26,
00200      @fname27,
00201      @fname28,
00202      @fname29,
00203      @fname30,
00204      @fname31,
00205      @fname32,
00206      @fname33,
00207      @fname34,
00208      @fname35,
00209      @fname36,
00210      @fname37,
00211      @fname38,
00212      @fname39,
00213      @fname40,
00214      @fname41,
00215      @fname42,
00216      @fname43,
00217      @fname44,
00218      @fname45,
00219      @fname46,
00220      @fname47,
00221      @fname48,
00222      @fname49,
00223      @fname50,
00224      @fname51,
00225      @fname52,
00226      @fname53,
00227      @fname54,
00228      @fname55,
00229      @fname56,
00230      @fname57,
00231      @fname58,
00232      @fname59,
00233      @fname60,
00234      @fname61,
00235      @fname62,
00236      @fname63,
00237      @fname64)

```

**Parameters**

Name	Type	Default
@start	nvarchar(25)	no default
@end	nvarchar(25)	no default
@devtype	nvarchar(260)	no default
@seqnum	int	no default
@fname1	nvarchar(260)	no default
@fname2	nvarchar(260)	no default
@fname3	nvarchar(260)	no default

Name	Type	Default
@fname4	nvarchar(260)	no default
@fname5	nvarchar(260)	no default
@fname6	nvarchar(260)	no default
@fname7	nvarchar(260)	no default
@fname8	nvarchar(260)	no default
@fname9	nvarchar(260)	no default
@fname10	nvarchar(260)	no default
@fname11	nvarchar(260)	no default
@fname12	nvarchar(260)	no default
@fname13	nvarchar(260)	no default
@fname14	nvarchar(260)	no default
@fname15	nvarchar(260)	no default
@fname16	nvarchar(260)	no default
@fname17	nvarchar(260)	no default
@fname18	nvarchar(260)	no default
@fname19	nvarchar(260)	no default
@fname20	nvarchar(260)	no default
@fname21	nvarchar(260)	no default
@fname22	nvarchar(260)	no default
@fname23	nvarchar(260)	no default
@fname24	nvarchar(260)	no default
@fname25	nvarchar(260)	no default
@fname26	nvarchar(260)	no default
@fname27	nvarchar(260)	no default
@fname28	nvarchar(260)	no default
@fname29	nvarchar(260)	no default
@fname30	nvarchar(260)	no default
@fname31	nvarchar(260)	no default
@fname32	nvarchar(260)	no default
@fname33	nvarchar(260)	no default
@fname34	nvarchar(260)	no default
@fname35	nvarchar(260)	no default
@fname36	nvarchar(260)	no default
@fname37	nvarchar(260)	no default
@fname38	nvarchar(260)	no default
@fname39	nvarchar(260)	no default
@fname40	nvarchar(260)	no default
@fname41	nvarchar(260)	no default
@fname42	nvarchar(260)	no default
@fname43	nvarchar(260)	no default
@fname44	nvarchar(260)	no default
@fname45	nvarchar(260)	no default
@fname46	nvarchar(260)	no default
@fname47	nvarchar(260)	no default
@fname48	nvarchar(260)	no default
@fname49	nvarchar(260)	no default
@fname50	nvarchar(260)	no default
@fname51	nvarchar(260)	no default
@fname52	nvarchar(260)	no default
@fname53	nvarchar(260)	no default
@fname54	nvarchar(260)	no default
@fname55	nvarchar(260)	no default
@fname56	nvarchar(260)	no default
@fname57	nvarchar(260)	no default
@fname58	nvarchar(260)	no default
@fname59	nvarchar(260)	no default
@fname60	nvarchar(260)	no default
@fname61	nvarchar(260)	no default
@fname62	nvarchar(260)	no default
@fname63	nvarchar(260)	no default
@fname64	nvarchar(260)	no default

**[sys].[fn\_EnumCurrentPrincipals]**

'UserDefinedFunction' in Database 'Northwind'

**Properties**

Property	Value
Ansi Nulls Status	True
Assembly Name	
Body Start Index	206
Class Name	
Create Date	14/10/2005 01:43:02
Data Type	
Data Type Schema	
Date Last Modified	14/10/2005 01:43:02
Function Type	Table
I D	-485928087
Implementation Type	TransactSql
Is Deterministic	False
Is Encrypted	False
Is Schema Bound	False
Is System Object	True
Method Name	
Returns Null On Null Input	
System Type	
Table Variable Name	@mirrorinstances

**Script:**

```

00001 create function sys.fn_EnumCurrentPrincipals()
00002 RETURNS @mirrorinstances TABLE
00003 (
00004     db_name                sysname,
00005     current_principal      sysname,
00006     mirroring_role         int NULL,
00007     mirroring_state        int NULL
00008 )
00009 as
00010 begin
00011 insert into @mirrorinstances
00012 select databases.name as db_name,
00013     sys.fn_GetCurrentPrincipal(databases.name) as current_principal,
00014     db_mirroring.mirroring_role as mirroring_role,
00015     db_mirroring.mirroring_state as mirroring_state
00016 from sys.database_mirroring db_mirroring, sys.databases databases where
00017     db_mirroring.database_id = databases.database_id
00018     and (databases.is_published = 1 or databases.is_merge_published = 1)
00019     and db_mirroring.mirroring_role is NOT NULL
00020 return
00021 end

```

**Parameters**

Name	Type	Default
------	------	---------

**[sys].[fn\_flsColTracked]**

'UserDefinedFunction' in Database 'Northwind'

**Properties**

Property	Value
Ansi Nulls Status	False
Assembly Name	
Body Start Index	79
Class Name	
Create Date	14/10/2005 01:43:09
Data Type	int
Data Type Schema	sys
Date Last Modified	14/10/2005 01:43:09
Function Type	Scalar
I D	-242919846
Implementation Type	TransactSql
Is Deterministic	False
Is Encrypted	False
Is Schema Bound	False
Is System Object	True
Method Name	
Returns Null On Null Input	
System Type	int
Table Variable Name	

**Script:**

```

00001 create function sys.fn_flsColTracked
00002   (@artnick int)
00003   returns int
00004 as
00005 begin
00006   declare @flsColTracked int
00007   set @flsColTracked=(select top 1 column_tracking from dbo.sysmergearticles where nickname=@artnick)
00008   return @flsColTracked
00009 end

```

**Parameters**

Name	Type	Default
@artnick	int	no default

**[sys].[fn\_get\_sql]**

'UserDefinedFunction' in Database 'Northwind'

**Properties**

Property	Value
Ansi Nulls Status	True
Assembly Name	
Body Start Index	190
Class Name	
Create Date	14/10/2005 01:40:54
Data Type	
Data Type Schema	
Date Last Modified	14/10/2005 01:40:54
Function Type	Inline
I D	-432442948
Implementation Type	TransactSql
Is Deterministic	False
Is Encrypted	False
Is Schema Bound	False
Is System Object	True
Method Name	
Returns Null On Null Input	
System Type	
Table Variable Name	

**Script:**

00001

**Parameters**

Name	Type	Default
@handle	varbinary(64)	no default

**[sys].[fn\_GetCurrentPrincipal]**

'UserDefinedFunction' in Database 'Northwind'

**Properties**

Property	Value
Ansi Nulls Status	True
Assembly Name	
Body Start Index	86
Class Name	
Create Date	14/10/2005 01:43:02
Data Type	sysname
Data Type Schema	sys
Date Last Modified	14/10/2005 01:43:02
Function Type	Scalar
I D	-986367695
Implementation Type	TransactSql
Is Deterministic	False
Is Encrypted	False
Is Schema Bound	False
Is System Object	True
Method Name	
Returns Null On Null Input	
System Type	nvarchar
Table Variable Name	

**Script:**

```

00001 create function sys.fn_GetCurrentPrincipal
00002 (@db_name sysname)
00003 RETURNS sysname
00004 as
00005 begin
00006     return
00007     (
00008         select case when (db_mirroring.mirroring_role = 1)
00009             then @@servername
00010             else
00011                 case when db_mirroring.mirroring_partner_instance is NULL
00012                     then @@servername
00013                     else db_mirroring.mirroring_partner_instance
00014                 end
00015         end
00016     from sys.database_mirroring db_mirroring, sys.databases databases where
00017     db_mirroring.database_id = databases.database_id
00018     and (databases.is_published = 1 or databases.is_merge_published = 1)
00019     and databases.name = @db_name
00020 )
00021 end

```

**Parameters**

Name	Type	Default
@db_name	sysname	no default

**[sys].[fn\_GetRowsetIdFromRowDump]**

'UserDefinedFunction' in Database 'Northwind'

**Properties**

Property	Value
Ansi Nulls Status	True
Assembly Name	
Body Start Index	414
Class Name	
Create Date	14/10/2005 01:40:42
Data Type	bigint
Data Type Schema	sys
Date Last Modified	14/10/2005 01:40:42
Function Type	Scalar
I D	-633331936
Implementation Type	TransactSql
Is Deterministic	False
Is Encrypted	False
Is Schema Bound	False
Is System Object	True
Method Name	
Returns Null On Null Input	
System Type	bigint
Table Variable Name	

**Script:**

```

00001 -----
00002 -- Name: sys.fn_GetRowsetIdFromRowDump
00003 --
00004 -- Description:
00005 -- Cracks the output of %%rowdump%% virtual column and returns the rowset id
00006 --
00007 -- Notes:
00008 -----
00009 CREATE FUNCTION sys.fn_GetRowsetIdFromRowDump (@rowdump AS varbinary (8000))
00010 RETURNS BIGINT
00011 AS
00012 BEGIN
00013     declare @offset INT
00014     declare @rowsetId BIGINT
00015
00016     set @offset = 1
00017
00018     -- Rowset id is first 8 bytes
00019     --
00020     select @rowsetId = convert (bigint, convert (binary (8), reverse (substring (@rowdump, @offset, 8))))
00021     RETURN @rowsetId
00022 END

```

**Parameters**

Name	Type	Default
@rowdump	varbinary(8000)	no default

**[sys].[fn\_helpcollations]**

'UserDefinedFunction' in Database 'Northwind'

**Properties**

Property	Value
Ansi Nulls Status	True
Assembly Name	
Body Start Index	69
Class Name	
Create Date	14/10/2005 01:40:55
Data Type	
Data Type Schema	
Date Last Modified	14/10/2005 01:40:55
Function Type	Inline
I D	-852794453
Implementation Type	TransactSql
Is Deterministic	False
Is Encrypted	False
Is Schema Bound	False
Is System Object	True
Method Name	
Returns Null On Null Input	
System Type	
Table Variable Name	

**Script:**

```

00001 create function sys.fn_helpcollations
00002 (
00003 )
00004
00005 returns table
00006 as
00007 return select * from OpenRowset(collations)

```

**Parameters**

Name	Type	Default
------	------	---------

**[sys].[fn\_helpdatatypemap]**

'UserDefinedFunction' in Database 'Northwind'

**Properties**

Property	Value
Ansi Nulls Status	False
Assembly Name	
Body Start Index	1441
Class Name	
Create Date	14/10/2005 01:45:02
Data Type	
Data Type Schema	
Date Last Modified	14/10/2005 01:45:02
Function Type	Table
I D	-705212384
Implementation Type	TransactSql
Is Deterministic	False
Is Encrypted	False
Is Schema Bound	False
Is System Object	True
Method Name	
Returns Null On Null Input	
System Type	
Table Variable Name	@retDataMap

**Script:**

```

00001
00002 --
00003 -- Name:
00004 --     fn_helpdatatypemap
00005 --
00006 -- Description:
00007 --     Retrieve data type map as inline table
00008 --
00009 -- Returns:
00010 --     0 if successful
00011 --     1 if failed
00012 --
00013 -- Security:
00014 --     public
00015 --
00016 -- Notes:
00017 --     Produces the full data type map based on input
00018 --     parameters. Includes filtered version based
00019 --     on source and destination dbms, and defaults.
00020 --
00021
00022 CREATE FUNCTION sys.fn_helpdatatypemap
00023 (
00024     @source_dbms          sysname = '%',
00025     @source_version       sysname = '%',
00026     @source_type          sysname = '%',
00027     @destination_dbms    sysname = '%',
00028     @destination_version sysname = '%',
00029     @destination_type     sysname = '%',
00030     @defaults_only       bit          = 0
00031 )
00032 RETURNS @retDataMap TABLE
00033 (
00034     mapping_id            int,
00035     source_dbms           sysname collate database_default,
00036     source_version        sysname NULL,
00037     source_type           sysname collate database_default,
00038     source_length_min     bigint,
00039     source_length_max     bigint,
00040     source_precision_min  bigint,
00041     source_precision_max  bigint,

```

```

00042 source_scale_min int,
00043 source_scale_max int,
00044 source_nullable bit,
00045 source_createparams int,
00046 destination_dbms sysname collate database_default,
00047 destination_version sysname collate database_default NULL,
00048 destination_type sysname collate database_default,
00049 destination_length bigint,
00050 destination_precision bigint,
00051 destination_scale int,
00052 destination_nullable bit,
00053 destination_createparams int,
00054 dataloss bit,
00055 is_default bit
00056 )
00057 AS
00058 BEGIN
00059 DECLARE @filter nvarchar(4000)
00060
00061 -- Prepare dbms for case insensitive searches
00062 SET @source_dbms = UPPER(@source_dbms)
00063 SET @destination_dbms = UPPER(@destination_dbms)
00064
00065 INSERT @retDataMap
00066 SELECT dm.datatype_mapping_id,
00067 src.dbms,
00068 src.version,
00069 srcdt.type,
00070 map.src_len_min,
00071 map.src_len_max,
00072 map.src_prec_min,
00073 map.src_prec_max,
00074 map.src_scale_min,
00075 map.src_scale_max,
00076 map.src_nullable,
00077 srcdt.createparams,
00078 dest.dbms,
00079 dest.version,
00080 destdt.type,
00081 dm.dest_length,
00082 CASE
00083 WHEN dm.dest_precision > 0 and dm.dest_scale > dm.dest_precision THEN
00084 » dm.dest_scale
00085 ELSE dm.dest_precision
00086 END,
00087 dm.dest_scale,
00088 dm.dest_nullable,
00089 dm.dest_createparams,
00090 dm.dataloss,
00091 case
00092 when map.default_datatype_mapping_id = dm.datatype_mapping_id then 1
00093 else 0
00094 end as [is_default]
00095 FROM msdb.dbo.MSdbms src,
00096 msdb.dbo.MSdbms dest,
00097 msdb.dbo.MSdbms_datatype srcdt,
00098 msdb.dbo.MSdbms_datatype destdt,
00099 msdb.dbo.MSdbms_map map,
00100 msdb.dbo.MSdbms_datatype_mapping dm
00101 WHERE src.dbms_id = map.src_dbms_id
00102 AND dest.dbms_id = map.dest_dbms_id
00103 AND srcdt.datatype_id = map.src_datatype_id
00104 AND map.map_id = dm.map_id
00105 AND dm.dest_datatype_id = destdt.datatype_id
00106 AND (@source_dbms = '%' OR src.dbms = @source_dbms)
00107 AND sys.fn_IHcompareversion(src.version, @source_version) = 1
00108 AND (@destination_dbms = '%' OR dest.dbms = @destination_dbms)
00109 AND sys.fn_IHcompareversion(dest.version, @destination_version) = 1
00110 AND (@source_type = N%' OR srcdt.type = @source_type)

```

```

00110    AND (@destination_type = N%' OR destdt.type = @destination_type)
00111    AND (@defaults_only = 0 OR map.default_datatype_mapping_id = dm.datatype_mapping_id)
00112    ORDER BY src.dbms,
00113           src.version,
00114           dest.dbms,
00115           dest.version,
00116           srcdt.type,
00117           [is_default] desc,
00118           destdt.type
00119
00120    RETURN
00121 END

```

#### Parameters

Name	Type	Default
@source_dbms	sysname	no default
@source_version	sysname	no default
@source_type	sysname	no default
@destination_dbms	sysname	no default
@destination_version	sysname	no default
@destination_type	sysname	no default
@defaults_only	bit	no default

**[sys].[fn\_IsBitSetInBitmask]**

'UserDefinedFunction' in Database 'Northwind'

**Properties**

Property	Value
Ansi Nulls Status	True
Assembly Name	
Body Start Index	104
Class Name	
Create Date	14/10/2005 01:43:25
Data Type	int
Data Type Schema	sys
Date Last Modified	14/10/2005 01:43:25
Function Type	Scalar
I D	-92987834
Implementation Type	TransactSql
Is Deterministic	False
Is Encrypted	False
Is Schema Bound	False
Is System Object	True
Method Name	
Returns Null On Null Input	
System Type	int
Table Variable Name	

**Script:**

```

00001
00002 create function sys.fn_IsBitSetInBitmask
00003 (@bitmask varbinary(500), @colid int)
00004 returns int
00005 as
00006 begin
00007 declare @word smallint
00008 declare @bit smallint
00009 declare @mask binary(2)
00010 declare @mval int
00011 declare @oldword binary(2)
00012
00013 if @colid < 1 return 0
00014
00015 SELECT @word = 1 + FLOOR((@colid -1

```

**Parameters**

Name	Type	Default
@bitmask	varbinary(500)	no default
@colid	int	no default

**[sys].[fn\_isrolemember]**

'UserDefinedFunction' in Database 'Northwind'

**Properties**

Property	Value
Ansi Nulls Status	True
Assembly Name	
Body Start Index	278
Class Name	
Create Date	14/10/2005 01:43:29
Data Type	int
Data Type Schema	sys
Date Last Modified	14/10/2005 01:43:29
Function Type	Scalar
I D	-977642094
Implementation Type	TransactSql
Is Deterministic	False
Is Encrypted	False
Is Schema Bound	False
Is System Object	True
Method Name	
Returns Null On Null Input	
System Type	int
Table Variable Name	

**Script:**

```

00001 create function sys.fn_isrolemember
00002 (
00003     @mode int -- 0 = generic pal role check in DB, 1 = DB owner role check, 2 = specific tran PAL role check
00004     ,@login sysname -- @login we need to check for
00005     ,@tranpubid int -- optional - needed for mode = 2
00006 )
00007 returns int
00008 as
00009 begin
00010     declare @fret int
00011
00012     if (@mode not in (0,1,2))
00013         select @mode = 0
00014     if (@login is null)
00015         select @login = suser_sname()
00016     select @fret = 0
00017
00018     if (@mode = 0)
00019     begin
00020         --
00021         -- generic PAL check in the current db
00022         --
00023         if exists (select * from sys.database_role_members
00024             where role_principal_id in (select principal_id
00025                 from sys.database_principals
00026                 where name like N'MSReplPAL_' + cast(db_id() as nvarchar(10)) + N'_%'
00027                 or name like N'MSMerge_%')
00028             and member_principal_id = (select dp.principal_id
00029                 from sys.database_principals as dp join master.dbo.syslogins as s
00030                 on dp.sid = s.sid
00031                 and s.name = @login))
00032             begin
00033                 --
00034                 -- has access
00035                 --
00036                 select @fret = 1
00037             end
00038     end
00039     else if (@mode = 1)
00040     begin
00041         --

```

```

00042  -- db owner check
00043  --
00044  if exists (select * from sys.database_role_members
00045            where role_principal_id in (select principal_id
00046                                       from sys.database_principals
00047                                       where lower(name) = 'db_owner')
00048            and member_principal_id = (select dp.principal_id
00049                                       from sys.database_principals as dp join master.dbo.syslogins as s
00050                                       on dp.sid = s.sid
00051                                       and s.name = @login))
00052  begin
00053  --
00054  -- has access
00055  --
00056  select @fret = 1
00057  end
00058  end
00059  else if (@mode = 2)
00060  begin
00061  --
00062  -- PAL role membership check for specific transactional publication
00063  --
00064  if exists (select * from sys.database_role_members
00065            where role_principal_id = (select principal_id
00066                                       from sys.database_principals
00067                                       where name = N'MSRepIPAL_' + cast(db_id() as nvarchar(10)) + N'_' + cast(
00068  » @tranpubid as nvarchar(10)) )
00069            and member_principal_id = (select dp.principal_id
00070                                       from sys.database_principals as dp join master.dbo.syslogins as s
00071                                       on dp.sid = s.sid
00072                                       and s.name = @login))
00073  begin
00074  --
00075  -- has access
00076  --
00077  select @fret = 1
00078  end
00079  end
00080  -- all done
00081  --
00082  return @fret
00083  end

```

#### Parameters

Name	Type	Default
@mode	int	no default
@login	sysname	no default
@tranpubid	int	no default

**[sys].[fn\_listextendedproperty]**

'UserDefinedFunction' in Database 'Northwind'

**Properties**

Property	Value
Ansi Nulls Status	True
Assembly Name	
Body Start Index	405
Class Name	
Create Date	14/10/2005 01:39:41
Data Type	
Data Type Schema	
Date Last Modified	14/10/2005 01:39:41
Function Type	Table
I D	-572068538
Implementation Type	TransactSql
Is Deterministic	False
Is Encrypted	False
Is Schema Bound	False
Is System Object	True
Method Name	
Returns Null On Null Input	
System Type	
Table Variable Name	@tab

**Script:**

```

00001 create function sys.fn_listextendedproperty
00002   (@name sysname           = NULL,
00003   @level0type      varchar(128) = NULL,
00004   @level0name      sysname      = NULL,
00005   @level1type      varchar(128) = NULL,
00006   @level1name      sysname      = NULL,
00007   @level2type      varchar(128) = NULL,
00008   @level2name      sysname      = NULL
00009 )
00010 returns @tab table(objtype varchar(128) null,
00011   objname sysname null,
00012   name sysname not null,
00013   value sql_variant null)
00014 as
00015 begin
00016
00017 declare @class int, -- convert to tinyint,
00018   @major int,
00019   @minminor int,
00020   @maxminor int,
00021   @basetype varchar(128)
00022

```

**Parameters**

Name	Type	Default
@name	sysname	no default
@level0type	varchar(128)	no default
@level0name	sysname	no default
@level1type	varchar(128)	no default
@level1name	sysname	no default
@level2type	varchar(128)	no default
@level2name	sysname	no default

**[sys].[fn\_MapSchemaType]**

'UserDefinedFunction' in Database 'Northwind'

**Properties**

Property	Value
Ansi Nulls Status	False
Assembly Name	
Body Start Index	159
Class Name	
Create Date	14/10/2005 01:49:54
Data Type	sysname
Data Type Schema	sys
Date Last Modified	14/10/2005 01:49:54
Function Type	Scalar
I D	-465170922
Implementation Type	TransactSql
Is Deterministic	False
Is Encrypted	False
Is Schema Bound	False
Is System Object	True
Method Name	
Returns Null On Null Input	
System Type	nvarchar
Table Variable Name	

**Script:**

```

00001 -- This function takes returns the type of the schema change
00002 create function sys.fn_MapSchemaType (@schematype int, @schemasubtype int)
00003 returns sysname
00004 as
00005 begin
00006     declare @friendly_schematype sysname
00007
00008     declare @SCHEMA_TYPE_SQLCOMMAND                int
00009     declare @SCHEMA_TYPE_SCHEMASCRIPT              int
00010     declare @SCHEMA_TYPE_NATIVEBCP                 int
00011     declare @SCHEMA_TYPE_CHARACTERBCP              int
00012     declare @SCHEMA_TYPE_LASTRECGEN                int
00013     declare @SCHEMA_TYPE_LASTSENTGEN               int
00014     declare @SCHEMA_TYPE_DIRECTORY                 int
00015     declare @SCHEMA_TYPE_PRIORITYCHANGE            int
00016     declare @SCHEMA_TYPE_RETENTIONCHANGE           int
00017     declare @SCHEMA_TYPE_TRIGGERSCRIPT             int
00018     declare @SCHEMA_TYPE_ALTERTABLE                int
00019     declare @SCHEMA_TYPE_REINITALL                  int
00020     declare @SCHEMA_TYPE_NONSQLALTERTABLE           int
00021     declare @SCHEMA_TYPE_REINITWITHUPLOAD           int
00022     declare @SCHEMA_TYPE_DRISCRIPIT                int
00023     declare @SCHEMA_TYPE_METADATACLEANUP           int
00024     declare @SCHEMA_TYPE_UPDATELASTSENTGEN         int
00025     declare @SCHEMA_TYPE_BACKWARD_COMPAT_LEVEL     int
00026     declare @SCHEMA_TYPE_VALIDATE_SUBSCRIBERINFO   int
00027     declare @SCHEMA_TYPE_WELLPARTITIONED           int
00028     declare @SCHEMA_TYPE_ARTICLERESOLVER           int
00029     declare @SCHEMA_TYPE_ARTICLEPROCESSINGORDER    int
00030     declare @SCHEMA_TYPE_PUBLISHED_IN_TRANPUB      int
00031     declare @SCHEMA_TYPE_COMPENSATE_FOR_ERRORS     int
00032     declare @SCHEMA_TYPE_ALTSNAPSHOTFOLDER         int
00033     declare @SCHEMA_TYPE_UPLOADOPTIONS             int
00034     declare @SCHEMA_TYPE_DELETETRACKING            int
00035     declare @SCHEMA_TYPE_DROPARTICLE               int
00036     declare @SCHEMA_TYPE_RETENTIONUNITCHANGE       int
00037     declare @SCHEMA_TYPE_USER_SCHEMA               int
00038     declare @SCHEMA_TYPE_PRESNAPSHOTSCRIPT         int
00039     declare @SCHEMA_TYPE_POSTSNAPSHOTSCRIPT        int
00040     declare @SCHEMA_TYPE_ONDEMANDUSERSCRIPT        int
00041     declare @SCHEMA_TYPE_MERGEPROCEDURESCRIPT      int

```

```

00042 declare @SCHEMA_TYPE_SNAPSHOTHEADERBEGIN int
00043 declare @SCHEMA_TYPE_SNAPSHOTHEADEREND int
00044 declare @SCHEMA_TYPE_SNAPSHOTTRAILER int
00045 declare @SCHEMA_TYPE_FTPADDRESS int
00046 declare @SCHEMA_TYPE_FTPPORT int
00047 declare @SCHEMA_TYPE_FTPSUBDIR int
00048 declare @SCHEMA_TYPE_SNAPSHOTCOMPRESSED int
00049 declare @SCHEMA_TYPE_FTPLOGIN int
00050 declare @SCHEMA_TYPE_FTPPASSWORD int
00051 declare @SCHEMA_TYPE_SYSPRESCRIPT int
00052 declare @SCHEMA_TYPE_PROCSHEMA int
00053 declare @SCHEMA_TYPE_VIEWSCHEMA int
00054 declare @SCHEMA_TYPE_UDFSHEMA int
00055 declare @SCHEMA_TYPE_VIEWINDEXES int
00056 declare @SCHEMA_TYPE_EXTENDEDPROPERTIES int
00057 declare @SCHEMA_TYPE_VALIDATE int
00058 declare @SCHEMA_TYPE_PRESNAPSHOTSQLCOMMAND int
00059 declare @SCHEMA_TYPE_DYNSNAPVALIDATION int
00060 declare @SCHEMA_TYPE_SYSTABLE_NATIVE_BCP_90 int
00061 declare @SCHEMA_TYPE_SYSTABLE_CHAR_BCP_90 int
00062 declare @SCHEMA_TYPE_SYSTABLE_NATIVE_BCP_90_FORGLOBALONLY int
00063 declare @SCHEMA_TYPE_SYSTABLE_CHAR_BCP_90_FORGLOBALONLY int
00064 declare @SCHEMA_TYPE_SYSTABLE_NATIVE_BCP_90_FORLIGHTWEIGHT int
00065 declare @SCHEMA_TYPE_SYSTABLE_CHAR_BCP_90_FORLIGHTWEIGHT int
00066 declare @SCHEMA_TYPE_USER_DEFINED_DATA_TYPE int
00067 declare @SCHEMA_TYPE_CLR_USER_DEFINED_DATA_TYPE int
00068 declare @SCHEMA_TYPE_ASSEMBLY int
00069 declare @SCHEMA_TYPE_PARTITIONSCHEME int
00070 declare @SCHEMA_TYPE_PARTITIONFUNCTION int
00071 declare @SCHEMA_TYPE_XMLSCHEMANAMESPACE int
00072 declare @SCHEMA_TYPE_FULLTEXTCATALOG int
00073 declare @SCHEMA_TYPE_SYNONYM int
00074 declare @SCHEMA_TYPE_DYNAMICBCPBIT int
00075 declare @SCHEMA_TYPE_DYNAMICNATIVEBCP int
00076 declare @SCHEMA_TYPE_DYNAMICCHARBCP int
00077 declare @SCHEMA_TYPE_DYNAMIC_SYSTABLE_NATIVE_BCP_90 int
00078 declare @SCHEMA_TYPE_DYNAMIC_SYSTABLE_CHAR_BCP_90 int
00079 declare @SCHEMA_TYPE_DYNAMIC_SYSTABLE_NATIVE_BCP_90_FORGLOBALONLY int
00080 declare @SCHEMA_TYPE_DYNAMIC_SYSTABLE_CHAR_BCP_90_FORGLOBALONLY int
00081 declare @SCHEMA_TYPE_DYNAMIC_SYSTABLE_NATIVE_BCP_90_FORLIGHTWEIGHT int
00082 declare @SCHEMA_TYPE_DYNAMIC_SYSTABLE_CHAR_BCP_90_FORLIGHTWEIGHT int
00083 declare @SCHEMA_TYPE_DDL_ACTIONS int
00084 declare @SCHEMA_TYPE_DDL_RELATED_COLUMN_FIXUP int
00085 declare @SCHEMA_TYPE_INCREMENTALSNAPSHOTCONTROL int
00086 declare @SCHEMA_TYPE_INCREMENTALALTSNAPSHOTFOLDER int
00087 declare @SCHEMA_TYPE_INCREMENTALSNAPSHOTHEADERBEGIN int
00088 declare @SCHEMA_TYPE_INCREMENTALSNAPSHOTHEADEREND int
00089 declare @SCHEMA_TYPE_INCREMENTALSNAPSHOTTRAILER int
00090 declare @SCHEMA_TYPE_INCREMENTALFTPADDRESS int
00091 declare @SCHEMA_TYPE_INCREMENTALFTPPORT int
00092 declare @SCHEMA_TYPE_INCREMENTALFTPSUBDIR int
00093 declare @SCHEMA_TYPE_INCREMENTALSNAPSHOTCOMPRESSED int
00094 declare @SCHEMA_TYPE_INCREMENTALFTPLOGIN int
00095 declare @SCHEMA_TYPE_INCREMENTALFTPPASSWORD int
00096 declare @SCHEMA_TYPE_CHANGE_MERGE_PUBLICATION_90RTM int
00097 declare @SCHEMA_TYPE_CHANGE_MERGE_ARTICLE_90RTM int
00098
00099 set @SCHEMA_TYPE_SQLCOMMAND = 1
00100 set @SCHEMA_TYPE_SCHEMASCRIPT = 2
00101 set @SCHEMA_TYPE_NATIVEBCP = 3
00102 set @SCHEMA_TYPE_CHARACTERBCP = 4
00103 set @SCHEMA_TYPE_LASTRECGEN = 5
00104 set @SCHEMA_TYPE_LASTSENTGEN = 6
00105 set @SCHEMA_TYPE_DIRECTORY = 7
00106 set @SCHEMA_TYPE_PRIORITYCHANGE = 8
00107 set @SCHEMA_TYPE_RETENTIONCHANGE = 9
00108 set @SCHEMA_TYPE_TRIGGERSCRIPT = 10
00109 set @SCHEMA_TYPE_ALTERTABLE = 11
00110 set @SCHEMA_TYPE_REINITALL = 12

```

```
00111 set @SCHEMA_TYPE_NONSQLALERTTABLE = 13
00112 set @SCHEMA_TYPE_REINITWITHUPLOAD = 14
00113 set @SCHEMA_TYPE_DRISCRIP = 15
00114 set @SCHEMA_TYPE_METADATACLEANUP = 16
00115 set @SCHEMA_TYPE_UPDATELASTSENTGEN = 17
00116 set @SCHEMA_TYPE_BACKWARD_COMPAT_LEVEL = 18
00117 set @SCHEMA_TYPE_VALIDATE_SUBSCRIBERINFO = 19
00118 set @SCHEMA_TYPE_WELLPARTITIONED = 20
00119 set @SCHEMA_TYPE_ARTICLERESOLVER = 21
00120 set @SCHEMA_TYPE_ARTICLEPROCESSINGORDER = 22
00121 set @SCHEMA_TYPE_PUBLISHED_IN_TRANPUB = 23
00122 set @SCHEMA_TYPE_COMPENSATE_FOR_ERRORS = 24
00123 set @SCHEMA_TYPE_ALTSNAPSHOTFOLDER = 25
00124 set @SCHEMA_TYPE_UPLOADOPTIONS = 26
00125 set @SCHEMA_TYPE_DELETETRACKING = 27
00126 set @SCHEMA_TYPE_DROPARTICLE = 28
00127 set @SCHEMA_TYPE_RETENTIONUNITCHANGE = 29
00128 set @SCHEMA_TYPE_USER_SCHEMA = 31
00129 set @SCHEMA_TYPE_PRESNAPSHOTSCRIPT = 40
00130 set @SCHEMA_TYPE_POSTSNAPSHOTSCRIPT = 45
00131 set @SCHEMA_TYPE_ONDEMANDUSERSCRIPT = 46
00132 set @SCHEMA_TYPE_MERGEPROCEDURESCRIPT = 48
00133 set @SCHEMA_TYPE_SNAPSHOTHEADERBEGIN = 50
00134 set @SCHEMA_TYPE_SNAPSHOTHEADEREND = 51
00135 set @SCHEMA_TYPE_SNAPSHOTTRAILER = 52
00136 set @SCHEMA_TYPE_FTPADDRESS = 53
00137 set @SCHEMA_TYPE_FTPPORT = 54
00138 set @SCHEMA_TYPE_FTPSUBDIR = 55
00139 set @SCHEMA_TYPE_SNAPSHOTCOMPRESSED = 56
00140 set @SCHEMA_TYPE_FTPLOGIN = 57
00141 set @SCHEMA_TYPE_FTPPASSWORD = 58
00142 set @SCHEMA_TYPE_SYSPRESCRIPT = 60
00143 set @SCHEMA_TYPE_PROCSHEMA = 61
00144 set @SCHEMA_TYPE_VIEWSCHEMA = 62
00145 set @SCHEMA_TYPE_UDFSHEMA = 63
00146 set @SCHEMA_TYPE_VIEWINDEXES = 64
00147 set @SCHEMA_TYPE_EXTENDEDPROPERTIES = 65
00148 set @SCHEMA_TYPE_VALIDATE = 66
00149 set @SCHEMA_TYPE_PRESNAPSHOTSQLCOMMAND = 67
00150 set @SCHEMA_TYPE_DYNSNAPVALIDATION = 71
00151 set @SCHEMA_TYPE_SYSTABLE_NATIVE_BCP_90 = 80
00152 set @SCHEMA_TYPE_SYSTABLE_CHAR_BCP_90 = 81
00153 set @SCHEMA_TYPE_SYSTABLE_NATIVE_BCP_90_FORGLOBALONLY = 82
00154 set @SCHEMA_TYPE_SYSTABLE_CHAR_BCP_90_FORGLOBALONLY = 83
00155 set @SCHEMA_TYPE_SYSTABLE_NATIVE_BCP_90_FORLIGHTWEIGHT = 84
00156 set @SCHEMA_TYPE_SYSTABLE_CHAR_BCP_90_FORLIGHTWEIGHT = 85
00157 set @SCHEMA_TYPE_USER_DEFINED_DATA_TYPE = 88
00158 set @SCHEMA_TYPE_CLR_USER_DEFINED_DATA_TYPE = 89
00159 set @SCHEMA_TYPE_ASSEMBLY = 90
00160 set @SCHEMA_TYPE_PARTITIONSCHEME = 91
00161 set @SCHEMA_TYPE_PARTITIONFUNCTION = 92
00162 set @SCHEMA_TYPE_XMLSCHEMANAMESPACE = 93
00163 set @SCHEMA_TYPE_FULLTEXTCATALOG = 96
00164 set @SCHEMA_TYPE_SYNONYM = 100
00165 set @SCHEMA_TYPE_DYNAMICBCPBIT = 128
00166 set @SCHEMA_TYPE_DYNAMICNATIVEBCP = 131
00167 set @SCHEMA_TYPE_DYNAMICCHARBCP = 132
00168 set @SCHEMA_TYPE_DYNAMIC_SYSTABLE_NATIVE_BCP_90 = 208
00169 set @SCHEMA_TYPE_DYNAMIC_SYSTABLE_CHAR_BCP_90 = 209
00170 set @SCHEMA_TYPE_DYNAMIC_SYSTABLE_NATIVE_BCP_90_FORGLOBALONLY = 210
00171 set @SCHEMA_TYPE_DYNAMIC_SYSTABLE_CHAR_BCP_90_FORGLOBALONLY = 211
00172 set @SCHEMA_TYPE_DYNAMIC_SYSTABLE_NATIVE_BCP_90_FORLIGHTWEIGHT = 212
00173 set @SCHEMA_TYPE_DYNAMIC_SYSTABLE_CHAR_BCP_90_FORLIGHTWEIGHT = 213
00174 set @SCHEMA_TYPE_DDL_ACTIONS = 300
00175 set @SCHEMA_TYPE_DDL_RELATED_COLUMN_FIXUP = 301
00176 set @SCHEMA_TYPE_CHANGE_MERGE_PUBLICATION_90RTM = 302
00177 set @SCHEMA_TYPE_CHANGE_MERGE_ARTICLE_90RTM = 303
00178 set @SCHEMA_TYPE_INCREMENTALSNAPSHOTCONTROL = 1024
00179 set @SCHEMA_TYPE_INCREMENTALALTSNAPSHOTFOLDER = 1049
```

```

00180 set @SCHEMA_TYPE_INCREMENTALSNAPSHOTHEADERBEGIN= 1074
00181 set @SCHEMA_TYPE_INCREMENTALSNAPSHOTHEADEREND= 1075
00182 set @SCHEMA_TYPE_INCREMENTALSNAPSHOTTRAILER= 1076
00183 set @SCHEMA_TYPE_INCREMENTALFTPADDRESS= 1077
00184 set @SCHEMA_TYPE_INCREMENTALFTPPORT= 1078
00185 set @SCHEMA_TYPE_INCREMENTALFTPSUBDIR= 1079
00186 set @SCHEMA_TYPE_INCREMENTALSNAPSHOTCOMPRESSED= 1080
00187 set @SCHEMA_TYPE_INCREMENTALFTPLOGIN= 1081
00188 set @SCHEMA_TYPE_INCREMENTALFTPPASSWORD= 1082
00189
00190
00191 --enum of constants
00192 declare @schemasubtype_addcolumn int
00193         ,@schemasubtype_dropcolumn int
00194         ,@schemasubtype_altercolumn int
00195         ,@schemasubtype_disabletrigger int
00196         ,@schemasubtype_enabletrigger int
00197         ,@schemasubtype_disabletriggerall int
00198         ,@schemasubtype_enabletriggerall int
00199         ,@schemasubtype_dropconstraint int
00200         ,@schemasubtype_enableconstraint int
00201         ,@schemasubtype_disableconstraint int
00202         ,@schemasubtype_enableconstraintall int
00203         ,@schemasubtype_disableconstraintall int
00204         ,@schemasubtype_addfk int
00205         ,@schemasubtype_addun int
00206         ,@schemasubtype_addchk int
00207         ,@schemasubtype_adddf int
00208         ,@schemasubtype_addpk int
00209
00210 select @schemasubtype_addcolumn = 1
00211         ,@schemasubtype_dropcolumn = 2
00212         ,@schemasubtype_altercolumn = 3
00213         ,@schemasubtype_addpk = 4
00214         » MARYKEY'
00215         ,@schemasubtype_addun = 5
00216         » QUE'
00217         ,@schemasubtype_addfk = 6
00218         ,@schemasubtype_dropconstraint = 7
00219         ,@schemasubtype_adddf = 8
00220         ,@schemasubtype_addchk = 9
00221         » K'
00222         ,@schemasubtype_disabletrigger = 10
00223         ,@schemasubtype_enabletrigger = 11
00224         ,@schemasubtype_disabletriggerall = 12
00225         ,@schemasubtype_enabletriggerall = 13
00226         ,@schemasubtype_enableconstraint = 14
00227         ,@schemasubtype_disableconstraint = 15
00228         ,@schemasubtype_enableconstraintall = 16
00229         ,@schemasubtype_disableconstraintall = 17
00230
00231 select @friendly_schematype = case @schematype
00232 when @SCHEMA_TYPE_SQLCOMMAND then 'SCHEMA_TYPE_SQLCOMMAND'
00233 when @SCHEMA_TYPE_SCHEMASCRIPT then 'SCHEMA_TYPE_SCHEMASCRIPT'
00234 when @SCHEMA_TYPE_NATIVEBCP then 'SCHEMA_TYPE_NATIVEBCP'
00235 when @SCHEMA_TYPE_CHARACTERBCP then 'SCHEMA_TYPE_CHARACTERBCP'
00236 when @SCHEMA_TYPE_LASTRECGEN then 'SCHEMA_TYPE_LASTRECGEN'
00237 when @SCHEMA_TYPE_LASTSENTGEN then 'SCHEMA_TYPE_LASTSENTGEN'
00238 when @SCHEMA_TYPE_DIRECTORY then 'SCHEMA_TYPE_DIRECTORY'
00239 when @SCHEMA_TYPE_PRIORITYCHANGE then 'SCHEMA_TYPE_PRIORITYCHANGE'
00240 when @SCHEMA_TYPE_RETENTIONCHANGE then 'SCHEMA_TYPE_RETENTIONCHANGE'
00241 when @SCHEMA_TYPE_TRIGGERSCRIPT then 'SCHEMA_TYPE_TRIGGERSCRIPT'
00242 when @SCHEMA_TYPE_ALPERTABLE then 'SCHEMA_TYPE_ALPERTABLE'
00243 when @SCHEMA_TYPE_REINITALL then 'SCHEMA_TYPE_REINITALL'
00244 when @SCHEMA_TYPE_NONSQLALPERTABLE then 'SCHEMA_TYPE_NONSQLALPERTABLE'
00245 when @SCHEMA_TYPE_REINITWITHUPLOAD then 'SCHEMA_TYPE_REINITWITHUPLOAD'
00246 when @SCHEMA_TYPE_DRISCRIP then 'SCHEMA_TYPE_DRISCRIP'
00247 when @SCHEMA_TYPE_METADATACLEANUP then 'SCHEMA_TYPE_METADATACLEANUP'
00248 when @SCHEMA_TYPE_UPDATELASTSENTGEN then 'SCHEMA_TYPE_UPDATELASTSENTGEN'

```

```
00246 when @SCHEMA_TYPE_BACKWARD_COMPAT_LEVEL then 'SCHEMA_TYPE_BACKWARD_COMPAT_LEV
» EL'
00247 when @SCHEMA_TYPE_VALIDATE_SUBSCRIBERINFO then 'SCHEMA_TYPE_VALIDATE_SUBSCRIBERINF
» O'
00248 when @SCHEMA_TYPE_WELLPARTITIONED then 'SCHEMA_TYPE_WELLPARTITIONED'
00249 when @SCHEMA_TYPE_ARTICLERESOLVER then 'SCHEMA_TYPE_ARTICLERESOLVER'
00250 when @SCHEMA_TYPE_ARTICLEPROCESSINGORDER then 'SCHEMA_TYPE_ARTICLEPROCESSINGORD
» ER'
00251 when @SCHEMA_TYPE_PUBLISHED_IN_TRANPUB then 'SCHEMA_TYPE_PUBLISHED_IN_TRANPUB'
00252 when @SCHEMA_TYPE_COMPENSATE_FOR_ERRORS then 'SCHEMA_TYPE_COMPENSATE_FOR_ERRO
» RS'
00253 when @SCHEMA_TYPE_ALTSNAPSHOTFOLDER then 'SCHEMA_TYPE_ALTSNAPSHOTFOLDER'
00254 when @SCHEMA_TYPE_UPLOADOPTIONS then 'SCHEMA_TYPE_UPLOADOPTIONS'
00255 when @SCHEMA_TYPE_DELETETRACKING then 'SCHEMA_TYPE_DELETETRACKING'
00256 when @SCHEMA_TYPE_DROPARTICLE then 'SCHEMA_TYPE_DROPARTICLE'
00257 when @SCHEMA_TYPE_RETENTIONUNITCHANGE then 'SCHEMA_TYPE_RETENTIONUNITCHANGE'
00258 when @SCHEMA_TYPE_USER_SCHEMA then 'SCHEMA_TYPE_USER_SCHEMA'
00259 when @SCHEMA_TYPE_PRESNAPSHOTSCRIPT then 'SCHEMA_TYPE_PRESNAPSHOTSCRIPT'
00260 when @SCHEMA_TYPE_POSTSNAPSHOTSCRIPT then 'SCHEMA_TYPE_POSTSNAPSHOTSCRIPT'
00261 when @SCHEMA_TYPE_ONDEMANDUSERSSCRIPT then 'SCHEMA_TYPE_ONDEMANDUSERSSCRIPT'
00262 when @SCHEMA_TYPE_MERGEPROCEDURESCRIPT then 'SCHEMA_TYPE_MERGEPROCEDURESCRIPT'
00263 when @SCHEMA_TYPE_SNAPSHOTHEADERBEGIN then 'SCHEMA_TYPE_SNAPSHOTHEADERBEGIN'
00264 when @SCHEMA_TYPE_SNAPSHOTHEADEREND then 'SCHEMA_TYPE_SNAPSHOTHEADEREND'
00265 when @SCHEMA_TYPE_SNAPSHOTTRAILER then 'SCHEMA_TYPE_SNAPSHOTTRAILER'
00266 when @SCHEMA_TYPE_FTPADDRESS then 'SCHEMA_TYPE_FTPADDRESS'
00267 when @SCHEMA_TYPE_FTPPORT then 'SCHEMA_TYPE_FTPPORT'
00268 when @SCHEMA_TYPE_FTPSUBDIR then 'SCHEMA_TYPE_FTPSUBDIR'
00269 when @SCHEMA_TYPE_SNAPSHOTCOMPRESSED then 'SCHEMA_TYPE_SNAPSHOTCOMPRESSED'
00270 when @SCHEMA_TYPE_FTPLOGIN then 'SCHEMA_TYPE_FTPLOGIN'
00271 when @SCHEMA_TYPE_FTPPASSWORD then 'SCHEMA_TYPE_FTPPASSWORD'
00272 when @SCHEMA_TYPE_SYSPRESCRIPT then 'SCHEMA_TYPE_SYSPRESCRIPT'
00273 when @SCHEMA_TYPE_PROCSHEMA then 'SCHEMA_TYPE_PROCSHEMA'
00274 when @SCHEMA_TYPE_VIEWSCHEMA then 'SCHEMA_TYPE_VIEWSCHEMA'
00275 when @SCHEMA_TYPE_UDFSHEMA then 'SCHEMA_TYPE_UDFSHEMA'
00276 when @SCHEMA_TYPE_VIEWINDEXES then 'SCHEMA_TYPE_VIEWINDEXES'
00277 when @SCHEMA_TYPE_EXTENDEDPROPERTIES then 'SCHEMA_TYPE_EXTENDEDPROPERTIES'
00278 when @SCHEMA_TYPE_VALIDATE then 'SCHEMA_TYPE_VALIDATE'
00279 when @SCHEMA_TYPE_PRESNAPSHOTSQLCOMMAND then 'SCHEMA_TYPE_PRESNAPSHOTSQLCOMM
» AND'
00280 when @SCHEMA_TYPE_DYNSNAPVALIDATION then 'SCHEMA_TYPE_DYNSNAPVALIDATION'
00281 when @SCHEMA_TYPE_SYSTABLE_NATIVE_BCP_90 then 'SCHEMA_TYPE_SYSTABLE_NATIVE_BCP_90'
00282 when @SCHEMA_TYPE_SYSTABLE_CHAR_BCP_90 then 'SCHEMA_TYPE_SYSTABLE_CHAR_BCP_90'
00283 when @SCHEMA_TYPE_SYSTABLE_NATIVE_BCP_90_FORGLOBALONLY then 'SCHEMA_TYPE_SYSTABL
» E_NATIVE_BCP_90_FORGLOBALONLY'
00284 when @SCHEMA_TYPE_SYSTABLE_CHAR_BCP_90_FORGLOBALONLY then 'SCHEMA_TYPE_SYSTABLE_
» CHAR_BCP_90_FORGLOBALONLY'
00285 when @SCHEMA_TYPE_SYSTABLE_NATIVE_BCP_90_FORLIGHTWEIGHT then 'SCHEMA_TYPE_SYSTABL
» E_NATIVE_BCP_90_FORLIGHTWEIGHT'
00286 when @SCHEMA_TYPE_SYSTABLE_CHAR_BCP_90_FORLIGHTWEIGHT then 'SCHEMA_TYPE_SYSTABLE
» _CHAR_BCP_90_FORLIGHTWEIGHT'
00287 when @SCHEMA_TYPE_USER_DEFINED_DATA_TYPE then 'SCHEMA_TYPE_USER_DEFINED_DATA_TYP
» E'
00288 when @SCHEMA_TYPE_CLR_USER_DEFINED_DATA_TYPE then 'SCHEMA_TYPE_CLR_USER_DEFINED_
» DATA_TYPE'
00289 when @SCHEMA_TYPE_ASSEMBLY then 'SCHEMA_TYPE_ASSEMBLY'
00290 when @SCHEMA_TYPE_PARTITIONSHEMA then 'SCHEMA_TYPE_PARTITIONSHEMA'
00291 when @SCHEMA_TYPE_PARTITIONFUNCTION then 'SCHEMA_TYPE_PARTITIONFUNCTION'
00292 when @SCHEMA_TYPE_XMLSCHEMANAMESPACE then 'SCHEMA_TYPE_XMLSCHEMANAMESPACE'
00293 when @SCHEMA_TYPE_FULLTEXTCATALOG then 'SCHEMA_TYPE_FULLTEXTCATALOG'
00294 when @SCHEMA_TYPE_SYNONYM then 'SCHEMA_TYPE_SYNONYM'
00295 when @SCHEMA_TYPE_DYNAMICBCPBIT then 'SCHEMA_TYPE_DYNAMICBCPBIT'
00296 when @SCHEMA_TYPE_DYNAMICNATIVEBCP then 'SCHEMA_TYPE_DYNAMICNATIVEBCP'
00297 when @SCHEMA_TYPE_DYNAMICCHARBCP then 'SCHEMA_TYPE_DYNAMICCHARBCP'
00298 when @SCHEMA_TYPE_DYNAMIC_SYSTABLE_NATIVE_BCP_90 then 'SCHEMA_TYPE_DYNAMIC_SYSTAB
» LE_NATIVE_BCP_90'
00299 when @SCHEMA_TYPE_DYNAMIC_SYSTABLE_CHAR_BCP_90 then 'SCHEMA_TYPE_DYNAMIC_SYSTABL
» E_CHAR_BCP_90'
00300 when @SCHEMA_TYPE_DYNAMIC_SYSTABLE_NATIVE_BCP_90_FORGLOBALONLY then 'SCHEMA_TYPE
» _DYNAMIC_SYSTABLE_NATIVE_BCP_90_FORGLOBALONLY'
```

```

00301 when @SCHEMA_TYPE_DYNAMIC_SYSTABLE_CHAR_BCP_90_FORGLOBALONLY then 'SCHEMA_TYPE_
» DYNAMIC_SYSTABLE_CHAR_BCP_90_FORGLOBALONLY'
00302 when @SCHEMA_TYPE_DYNAMIC_SYSTABLE_NATIVE_BCP_90_FORLIGHTWEIGHT then 'SCHEMA_TYPE
» _DYNAMIC_SYSTABLE_NATIVE_BCP_90_FORLIGHTWEIGHT'
00303 when @SCHEMA_TYPE_DYNAMIC_SYSTABLE_CHAR_BCP_90_FORLIGHTWEIGHT then 'SCHEMA_TYPE_
» DYNAMIC_SYSTABLE_CHAR_BCP_90_FORLIGHTWEIGHT'
00304 when @SCHEMA_TYPE_DDL_ACTIONS then
00305     case @schemasubtype
00306         when @schemasubtype_addcolumn then 'SCHEMA_TYPE_DDL_ACTIONS_ADDCOLUMN'
00307         when @schemasubtype_dropcolumn then 'SCHEMA_TYPE_DDL_ACTIONS_DROPCOLUMN'
00308         when @schemasubtype_altercolumn then 'SCHEMA_TYPE_DDL_ACTIONS_ALTERCOLUMN'
00309         when @schemasubtype_addpk then 'SCHEMA_TYPE_DDL_ACTIONS_ADDPRIMARYKEY'
00310         when @schemasubtype_addun then 'SCHEMA_TYPE_DDL_ACTIONS_ADDUNIQUE'
00311         when @schemasubtype_addfk then 'SCHEMA_TYPE_DDL_ACTIONS_ADDREFERENCE'
00312         when @schemasubtype_dropconstraint then 'SCHEMA_TYPE_DDL_ACTIONS_DROPCONSTRAINT'
00313         when @schemasubtype_adddf then 'SCHEMA_TYPE_DDL_ACTIONS_ADDDEFAULT'
00314         when @schemasubtype_addchk then 'SCHEMA_TYPE_DDL_ACTIONS_ADDCHECK'
00315         when @schemasubtype_disabletrigger then 'SCHEMA_TYPE_DDL_ACTIONS_DISABLETRIGGER'
00316         when @schemasubtype_enabletrigger then 'SCHEMA_TYPE_DDL_ACTIONS_ENABLETRIGGER'
00317         when @schemasubtype_disabletriggerall then 'SCHEMA_TYPE_DDL_ACTIONS_DISABLETRIGGER'
00318         when @schemasubtype_enabletriggerall then 'SCHEMA_TYPE_DDL_ACTIONS_ENABLETRIGGER'
00319         when @schemasubtype_enableconstraint then 'SCHEMA_TYPE_DDL_ACTIONS_ENABLECONSTRAINT'
00320         when @schemasubtype_disableconstraint then 'SCHEMA_TYPE_DDL_ACTIONS_DISABLECONSTRAINT'
00321         when @schemasubtype_enableconstraintall then 'SCHEMA_TYPE_DDL_ACTIONS_ENABLECONSTRAIN
» T'
00322         when @schemasubtype_disableconstraintall then 'SCHEMA_TYPE_DDL_ACTIONS_DISABLECONSTRAI
» NT'
00323     else 'SCHEMA_TYPE_DDL_ACTIONS'
00324     end
00325 when @SCHEMA_TYPE_DDL_RELATED_COLUMN_FIXUP then 'SCHEMA_TYPE_DDL_RELATED_COLUMN
» _FIXUP'
00326 when @SCHEMA_TYPE_CHANGE_MERGE_PUBLICATION_90RTM then 'SCHEMA_TYPE_CHANGE_MERG
» E_PUBLICATION_90RTM'
00327 when @SCHEMA_TYPE_CHANGE_MERGE_ARTICLE_90RTM then 'SCHEMA_TYPE_CHANGE_MERGE_AR
» TICLE_90RTM'
00328 when @SCHEMA_TYPE_INCREMENTALSNAPSHOTCONTROL then 'SCHEMA_TYPE_INCREMENTALSNAP
» SHOTCONTROL'
00329 when @SCHEMA_TYPE_INCREMENTALALTSNAPSHOTFOLDER then 'SCHEMA_TYPE_INCREMENTALALT
» SNAPSHOTFOLDER'
00330 when @SCHEMA_TYPE_INCREMENTALSNAPSHOTHEADERBEGIN then 'SCHEMA_TYPE_INCREMENTALS
» NAPSHOTHEADERBEGIN'
00331 when @SCHEMA_TYPE_INCREMENTALSNAPSHOTHEADEREND then 'SCHEMA_TYPE_INCREMENTALSN
» APSHOTHEADEREND'
00332 when @SCHEMA_TYPE_INCREMENTALSNAPSHOTTRAILER then 'SCHEMA_TYPE_INCREMENTALSNAPS
» HOTTRAILER'
00333 when @SCHEMA_TYPE_INCREMENTALFTPADDRESS then 'SCHEMA_TYPE_INCREMENTALFTPADDRESS
» '
00334 when @SCHEMA_TYPE_INCREMENTALFTPSPORT then 'SCHEMA_TYPE_INCREMENTALFTPSPORT'
00335 when @SCHEMA_TYPE_INCREMENTALFTPSUBDIR then 'SCHEMA_TYPE_INCREMENTALFTPSUBDIR'
00336 when @SCHEMA_TYPE_INCREMENTALSNAPSHOTCOMPRESSED then 'SCHEMA_TYPE_INCREMENTALS
» NAPSHOTCOMPRESSED'
00337 when @SCHEMA_TYPE_INCREMENTALFTPLOGIN then 'SCHEMA_TYPE_INCREMENTALFTPLOGIN'
00338 when @SCHEMA_TYPE_INCREMENTALFTPPASSWORD then 'SCHEMA_TYPE_INCREMENTALFTPPASSW
» ORD'
00339 else 'SCHEMA_TYPE_INVALID'
00340 end
00341 return @friendly_schematype
00342 end

```

**Parameters**

Name	Type	Default
@schematype	int	no default
@schemasubtype	int	no default

**[sys].[fn\_MSdayasnumber]**

'UserDefinedFunction' in Database 'Northwind'

**Properties**

Property	Value
Ansi Nulls Status	True
Assembly Name	
Body Start Index	72
Class Name	
Create Date	14/10/2005 01:50:40
Data Type	int
Data Type Schema	sys
Date Last Modified	14/10/2005 01:50:40
Function Type	Scalar
I D	-929297756
Implementation Type	TransactSql
Is Deterministic	False
Is Encrypted	False
Is Schema Bound	False
Is System Object	True
Method Name	
Returns Null On Null Input	
System Type	int
Table Variable Name	

**Script:**

```

00001 create function sys.fn_MSdayasnumber (@day datetime)
00002     returns int
00003 as
00004 begin
00005     return (year(@day) - 2000) * 366 + datepart(dayofyear,@day)
00006 end

```

**Parameters**

Name	Type	Default
@day	datetime	no default

**[sys].[fn\_MSgeneration\_downloadonly]**

'UserDefinedFunction' in Database 'Northwind'

**Properties**

Property	Value
Ansi Nulls Status	False
Assembly Name	
Body Start Index	486
Class Name	
Create Date	14/10/2005 01:50:26
Data Type	bigint
Data Type Schema	sys
Date Last Modified	14/10/2005 01:50:26
Function Type	Scalar
I D	-649029873
Implementation Type	TransactSql
Is Deterministic	False
Is Encrypted	False
Is Schema Bound	False
Is System Object	True
Method Name	
Returns Null On Null Input	
System Type	bigint
Table Variable Name	

**Script:**

```

00001 -- Warning: Only use this function when enumerating deletes, but not changes.
00002 -- If it was used for changes, the metadata type would falsely look like contents deferred.
00003 -- If the download only subscriber has the row, it there also looks like contents deferred.
00004 -- Consequently, CReplRowChange::Reconcile does nothing with that row.
00005 create function sys.fn_MSgeneration_downloadonly(
00006     @generation bigint,
00007     @tablenick int)
00008     returns bigint
00009 as
00010 begin
00011     declare @gen bigint
00012
00013     if 1 = sys.fn_MSarticle_has_downloadonly_property(@tablenick)
00014         set @gen= null
00015     else
00016         set @gen= coalesce(@generation, 0)
00017
00018     return @gen
00019 end

```

**Parameters**

Name	Type	Default
@generation	bigint	no default
@tablenick	int	no default

**[sys].[fn\_MSget\_dynamic\_filter\_login]**

'UserDefinedFunction' in Database 'Northwind'

**Properties**

Property	Value
Ansi Nulls Status	False
Assembly Name	
Body Start Index	414
Class Name	
Create Date	14/10/2005 01:50:22
Data Type	sysname
Data Type Schema	sys
Date Last Modified	14/10/2005 01:50:22
Function Type	Scalar
I D	-928826610
Implementation Type	TransactSql
Is Deterministic	False
Is Encrypted	False
Is Schema Bound	False
Is System Object	True
Method Name	
Returns Null On Null Input	
System Type	nvarchar
Table Variable Name	

**Script:**

```

00001 -- This function returns the dynamic filter login (suser_sname) corresponding to the given
00002 -- value of partitionid. If the publication uses both hostname and suser_sname for filtering
00003 -- it uses the current value of hostname set on the connection to get the correct partition id.
00004 create function sys.fn_MSget_dynamic_filter_login
00005 (
00006     @publication_number int,
00007     @partition_id int
00008 )
00009 returns sysname
00010 as
00011 begin
00012     declare @pubid uniqueidentifier
00013     declare @retcode int
00014     declare @uses_host_name bit
00015     declare @suser_sname sysname
00016     declare @host_name sysname
00017     declare @select_command nvarchar(400)
00018
00019     if (is_member('db_owner') <> 1)
00020     begin
00021         return NULL
00022     end
00023
00024     select @pubid = pubid from dbo.sysmergepublications where publication_number = @publication_number
00025     if @pubid is NULL
00026         return NULL
00027
00028     select @uses_host_name = sys.fn_MSpublication_uses_host_name(@pubid)
00029     if @@error <> 0
00030         return NULL
00031
00032     if @uses_host_name = 1
00033         select @host_name = host_name()
00034     else
00035         select @host_name = NULL
00036
00037     -- the following is in dynamic sql because the hostname column may not exist when all publications
00038     -- use suser_sname for filtering
00039     if @uses_host_name = 1
00040         select @suser_sname = sys.fn_MSget_dynamic_filter_login_with_hostname(@publication_number,
    »         @partition_id)

```

```
00041 else
00042     select @suser_sname = SUSER_SNAME_FN from dbo.MSmerge_partition_groups
00043     where publication_number = @publication_number
00044     and partition_id = @partition_id
00045
00046 return @suser_sname
00047 end
```

**Parameters**

Name	Type	Default
@publication_number	int	no default
@partition_id	int	no default

**[sys].[fn\_MSorbitmaps]**

'UserDefinedFunction' in Database 'Northwind'

**Properties**

Property	Value
Ansi Nulls Status	True
Assembly Name	
Body Start Index	108
Class Name	
Create Date	14/10/2005 01:50:39
Data Type	varbinary
Data Type Schema	sys
Date Last Modified	14/10/2005 01:50:39
Function Type	Scalar
I D	-701331644
Implementation Type	TransactSql
Is Deterministic	False
Is Encrypted	False
Is Schema Bound	False
Is System Object	True
Method Name	
Returns Null On Null Input	
System Type	varbinary
Table Variable Name	

**Script:**

```

00001 create function sys.fn_MSorbitmaps (@bm1 varbinary(128), @bm2 varbinary(128))
00002     returns varbinary(128)
00003 as
00004 begin
00005
00006     declare @bm3 varbinary(128)
00007     exec sys.xp_ORbitmap @bm1, @bm2, @bm3 output
00008     return @bm3
00009
00010 end

```

**Parameters**

Name	Type	Default
@bm1	varbinary(128)	no default
@bm2	varbinary(128)	no default

**[sys].[fn\_MSrepl\_map\_resolver\_clsId]**

'UserDefinedFunction' in Database 'Northwind'

**Properties**

Property	Value
Ansi Nulls Status	False
Assembly Name	
Body Start Index	392
Class Name	
Create Date	14/10/2005 01:48:48
Data Type	nvarchar
Data Type Schema	sys
Date Last Modified	14/10/2005 01:48:48
Function Type	Scalar
I D	-676201881
Implementation Type	TransactSql
Is Deterministic	False
Is Encrypted	False
Is Schema Bound	False
Is System Object	True
Method Name	
Returns Null On Null Input	
System Type	nvarchar
Table Variable Name	

**Script:**

```

00001
00002 create function sys.fn_MSrepl_map_resolver_clsId (
00003   @compatibility_level int,

```

**Parameters**

Name	Type	Default
@compatibility_level	int	no default
@article_resolver	nvarchar(255)	no default
@resolver_clsId	nvarchar(60)	no default

**[sys].[fn\_MStestbit]**

'UserDefinedFunction' in Database 'Northwind'

**Properties**

Property	Value
Ansi Nulls Status	True
Assembly Name	
Body Start Index	94
Class Name	
Create Date	14/10/2005 01:50:39
Data Type	bit
Data Type Schema	sys
Date Last Modified	14/10/2005 01:50:39
Function Type	Scalar
I D	-831527161
Implementation Type	TransactSql
Is Deterministic	False
Is Encrypted	False
Is Schema Bound	False
Is System Object	True
Method Name	
Returns Null On Null Input	
System Type	bit
Table Variable Name	

**Script:**

```

00001 create function sys.fn_MStestbit (@bitmap varbinary(128), @colidx smallint)
00002 returns bit
00003 as
00004 begin
00005     -- size of bitmap is 1024 bits
00006     if @colidx < 1 or @colidx > 1024 return 0
00007
00008     declare @byte binary(1)
00009     declare @mask tinyint
00010     declare @idxinbyte tinyint
00011     declare @shift tinyint
00012
00013     set @byte= substring(@bitmap, (@colidx-1

```

**Parameters**

Name	Type	Default
@bitmap	varbinary(128)	no default
@colidx	smallint	no default

**[sys].[fn\_MSvector\_downloadonly]**

'UserDefinedFunction' in Database 'Northwind'

**Properties**

Property	Value
Ansi Nulls Status	False
Assembly Name	
Body Start Index	158
Class Name	
Create Date	14/10/2005 01:50:26
Data Type	varbinary
Data Type Schema	sys
Date Last Modified	14/10/2005 01:50:26
Function Type	Scalar
I D	-257717349
Implementation Type	TransactSql
Is Deterministic	False
Is Encrypted	False
Is Schema Bound	False
Is System Object	True
Method Name	
Returns Null On Null Input	
System Type	varbinary
Table Variable Name	

**Script:**

```

00001 create function sys.fn_MSvector_downloadonly(
00002     @vector varbinary(2953),
00003     @tablenick int)
00004     returns varbinary(311)
00005 as
00006 begin
00007     if 1 = sys.fn_MSarticle_has_downloadonly_property(@tablenick)
00008         set @vector= null
00009
00010     return @vector
00011 end

```

**Parameters**

Name	Type	Default
@vector	varbinary(2953)	no default
@tablenick	int	no default

**[sys].[fn\_my\_permissions]**

'UserDefinedFunction' in Database 'Northwind'

**Properties**

Property	Value
Ansi Nulls Status	True
Assembly Name	
Body Start Index	212
Class Name	
Create Date	14/10/2005 01:39:48
Data Type	
Data Type Schema	
Date Last Modified	14/10/2005 01:39:48
Function Type	Table
I D	-462384620
Implementation Type	TransactSql
Is Deterministic	False
Is Encrypted	False
Is Schema Bound	False
Is System Object	True
Method Name	
Returns Null On Null Input	
System Type	
Table Variable Name	@myperms

**Script:**

```

00001 CREATE FUNCTION sys.fn_my_permissions(@entity sysname, @class nvarchar(60))
00002 RETURNS @myperms table (
00003     entity_name sysname           NULL,
00004     subentity_name sysname       NULL,
00005     permission_name nvarchar(60))
00006 AS
00007 BEGIN
00008     IF (@class = 'database')
00009         SET @entity = NULL
00010     IF (@class IS NULL or len(@class) = 0)
00011         SET @class = 'server'
00012
00013     DECLARE @pseudoclass sysname
00014     IF (@class = 'server')
00015         SET @pseudoclass = NULL
00016     ELSE
00017         SET @pseudoclass = @class
00018
00019     DECLARE @entityname sysname
00020     SET @entityname = CASE @class
00021         WHEN 'server' THEN 'server'
00022         WHEN 'database' THEN 'database'
00023         ELSE @entity
00024     END
00025     IF (@class = 'server')
00026         SET @pseudoclass = NULL
00027     ELSE
00028         SET @pseudoclass = @class
00029
00030     -- Get Entity Level Permissions
00031     INSERT INTO @myperms
00032     SELECT @entityname, '', fbp.permission_name
00033     FROM sys.fn_built_in_permissions(@class) AS fbp
00034     WHERE has_perms_by_name(@entity, @pseudoclass, fbp.permission_name) <> 0
00035
00036     -- Get Sub Entity Level Permissions
00037     IF (@class = 'object')
00038         BEGIN
00039             DECLARE @objId int
00040             DECLARE @thisPerm sysname
00041

```

```
00042      SELECT @objId = object_id(@entity)
00043
00044      INSERT INTO @myperms
00045          SELECT @entity, name, 'SELECT'
00046          FROM sys.columns where object_id = @objId AND
00047              has_perms_by_name(@entity, 'object', 'select', name, 'column') <> 0
00048
00049      INSERT INTO @myperms
00050          SELECT @entity, name, 'UPDATE'
00051          FROM sys.columns where object_id = @objId AND
00052              has_perms_by_name(@entity, 'object', 'update', name, 'column') <> 0
00053
00054      INSERT INTO @myperms
00055          SELECT @entity, name, 'REFERENCES'
00056          FROM sys.columns where object_id = @objId AND
00057              has_perms_by_name(@entity, 'object', 'references', name, 'column') <> 0
00058      end
00059      RETURN
00060      END
```

**Parameters**

Name	Type	Default
@entity	sysname	no default
@class	nvarchar(60)	no default

**[sys].[fn\_numberOf1InBinaryAfterLoc]**

'UserDefinedFunction' in Database 'Northwind'

**Properties**

Property	Value
Ansi Nulls Status	False
Assembly Name	
Body Start Index	92
Class Name	
Create Date	14/10/2005 01:50:28
Data Type	int
Data Type Schema	sys
Date Last Modified	14/10/2005 01:50:28
Function Type	Scalar
I D	-725621673
Implementation Type	TransactSql
Is Deterministic	False
Is Encrypted	False
Is Schema Bound	False
Is System Object	True
Method Name	
Returns Null On Null Input	
System Type	int
Table Variable Name	

**Script:**

```

00001 create function sys.fn_numberOf1InBinaryAfterLoc (@byte binary, @loc int)
00002 returns int
00003 as
00004 begin
00005     declare @index int, @mark int, @counter int
00006     set @counter=0
00007     set @index=0
00008     if @loc not between 1 and 8
00009         return 0
00010     while @index<@loc
00011     begin
00012         set @mark=power(2, @index)
00013         if @mark&@byte <>0
00014             begin
00015                 select @counter = @counter+1
00016             end
00017         set @index=@index+1
00018     end
00019     return @counter
00020 end

```

**Parameters**

Name	Type	Default
@byte	binary	no default
@loc	int	no default

**[sys].[fn\_numberOf1InVarBinary]**

'UserDefinedFunction' in Database 'Northwind'

**Properties**

Property	Value
Ansi Nulls Status	False
Assembly Name	
Body Start Index	83
Class Name	
Create Date	14/10/2005 01:50:28
Data Type	int
Data Type Schema	sys
Date Last Modified	14/10/2005 01:50:28
Function Type	Scalar
I D	-510634
Implementation Type	TransactSql
Is Deterministic	False
Is Encrypted	False
Is Schema Bound	False
Is System Object	True
Method Name	
Returns Null On Null Input	
System Type	int
Table Variable Name	

**Script:**

```

00001 create function sys.fn_numberOf1InVarBinary (@bm varbinary(128))
00002 returns int
00003 as
00004 begin
00005     declare @len int
00006     declare @count int
00007     declare @subbyte binary
00008     set @count=0
00009     set @len=DATALENGTH (@bm)
00010     while (@len>0)
00011     begin
00012         set @subbyte=SUBSTRING(@bm, @len, 1)
00013         set @count=@count+sys.fn_numberOf1InBinaryAfterLoq(@subbyte, 8)
00014         set @len=@len-1
00015     end
00016     return @count
00017
00018 end

```

**Parameters**

Name	Type	Default
@bm	varbinary(128)	no default

**[sys].[fn\_repladjustcolumnmap]**

'UserDefinedFunction' in Database 'Northwind'

**Properties**

Property	Value
Ansi Nulls Status	False
Assembly Name	
Body Start Index	149
Class Name	
Create Date	14/10/2005 01:43:15
Data Type	varbinary
Data Type Schema	sys
Date Last Modified	14/10/2005 01:43:15
Function Type	Scalar
I D	-661149850
Implementation Type	TransactSql
Is Deterministic	False
Is Encrypted	False
Is Schema Bound	False
Is System Object	True
Method Name	
Returns Null On Null Input	
System Type	varbinary
Table Variable Name	

**Script:**

```

00001 create function sys.fn_repladjustcolumnmap
00002 (
00003     @objid int
00004     ,@total_col int
00005     ,@inmap varbinary(4000)
00006 )
00007 returns varbinary(4000)
00008 as
00009 begin
00010     declare @colmap table (relativeorder int identity(1,1), colid int)
00011     declare
00012         @outmap varbinary(4000)
00013         ,@relpos int
00014         ,@colid int
00015         ,@bytepos int
00016         ,@bitpos int
00017         ,@num_bytes int
00018
00019     --
00020     -- initialize
00021     --
00022     select @outmap = 0
00023     insert into @colmap (colid)
00024         select column_id from sys.columns where object_id = @objid order by column_id
00025     if (@@error != 0)
00026     begin
00027         return cast(NULL as varbinary(4000))
00028     end
00029
00030     --
00031     -- for each column in the column map
00032     --
00033     declare #colmap_cursor cursor local fast_forward for
00034         select relativeorder, colid from @colmap
00035         order by relativeorder
00036     for read only
00037
00038     open #colmap_cursor
00039     fetch #colmap_cursor into @relpos, @colid
00040     while (@@fetch_status = 0)
00041     begin

```

```
00042  --
00043  -- select the absolute column position
00044  --
00045  select @bytepos = 1 + ((@colid-1)
```

**Parameters**

Name	Type	Default
@objid	int	no default
@total_col	int	no default
@inmap	varbinary(4000)	no default

**[sys].[fn\_repldecryptver4]**

'UserDefinedFunction' in Database 'Northwind'

**Properties**

Property	Value
Ansi Nulls Status	True
Assembly Name	
Body Start Index	731
Class Name	
Create Date	14/10/2005 01:44:40
Data Type	nvarchar
Data Type Schema	sys
Date Last Modified	14/10/2005 01:44:40
Function Type	Scalar
I D	-882476690
Implementation Type	TransactSql
Is Deterministic	False
Is Encrypted	False
Is Schema Bound	False
Is System Object	True
Method Name	
Returns Null On Null Input	
System Type	nvarchar
Table Variable Name	

**Script:**

```

00001 --
00002 -- Name:
00003 -- fn_repldecryptver4
00004 --
00005 -- Description:
00006 -- This function accepts an encrypted password as input
00007 -- and decrypts the value using the server provided crypto
00008 -- based on the master key created in the local database.
00009 -- If it detects that the logged on user is not DBO or that
00010 -- the password has not been encrypted using the server dpapi
00011 -- then the original password input is returned.
00012 --
00013 -- Parameters:
00014 -- See the procedure definition.
00015 --
00016 -- Returns:
00017 -- 0 - On Success
00018 -- 1 - On Failure
00019 --
00020 -- Result:
00021 -- None
00022 --
00023 -- Security:
00024 -- DBO (func is public because snapshot agent calls it)
00025 --
00026 CREATE FUNCTION sys.fn_repldecryptver4
00027 (
00028     @password nvarchar(524)
00029 )
00030 RETURNS nvarchar(524)
00031 AS
00032 BEGIN
00033     -- if we are dbo and the encryption on the password
00034     -- is version 4 then we will decrypt using the cert
00035     IF IS_MEMBER('db_owner') = 1
00036         AND sys.fn_replcryptversion(@password) = 4
00037         AND EXISTS(SELECT *
00038             FROM sys.symmetric_keys
00039             WHERE name = '##MS_DatabaseMasterKey##')
00040         AND EXISTS(SELECT *
00041             FROM sys.symmetric_keys

```

```
00042 WHERE name = 'SQLSERVER_REPLICATION')
00043 BEGIN
00044 SELECT @password = CONVERT(nvarchar(524), DECRYPTBYKEYAUTOCERT(CERT_ID('SQLSERVER_REP
» LICATION'), NULL, CONVERT(varbinary(4000), RIGHT(@password, LEN(@password) - 4))))
00045 END
00046
00047 RETURN @password
00048 END
```

**Parameters**

Name	Type	Default
@password	nvarchar(524)	no default

**[sys].[fn\_replformatdatetime]**

'UserDefinedFunction' in Database 'Northwind'

**Properties**

Property	Value
Ansi Nulls Status	False
Assembly Name	
Body Start Index	535
Class Name	
Create Date	14/10/2005 01:43:09
Data Type	nvarchar
Data Type Schema	sys
Date Last Modified	14/10/2005 01:43:09
Function Type	Scalar
I D	-154846366
Implementation Type	TransactSql
Is Deterministic	False
Is Encrypted	False
Is Schema Bound	False
Is System Object	True
Method Name	
Returns Null On Null Input	
System Type	nvarchar
Table Variable Name	

**Script:**

```

00001 --
00002 -- Name: fn_replformatdatetime
00003 --
00004 -- Description: This function encapsulates the definitive method for
00005 --           formatting a datetime field into a string in SQL Replication.
00006 --
00007 -- Parameter: @datetime datetime (mandatory)
00008 --
00009 -- Returns: nvarchar(50) (leaves room for modification in the future)
00010 --
00011 -- Security: Execute permission of this function is granted to public
00012 --           (public interface object)
00013 --
00014 create function sys.fn_replformatdatetime (
00015     @datetime datetime
00016 ) returns nvarchar(50)
00017 as
00018 begin
00019     declare @datetime_string nvarchar(50)
00020     select @datetime_string = convert(nvarchar(12), @datetime, 112) +
00021         substring(convert(nvarchar(24), @datetime, 121), 11, 13)
00022     return @datetime_string
00023 end

```

**Parameters**

Name	Type	Default
@datetime	datetime	no default

**[sys].[fn\_replgetcolidfrombitmap]**

'UserDefinedFunction' in Database 'Northwind'

**Properties**

Property	Value
Ansi Nulls Status	False
Assembly Name	
Body Start Index	107
Class Name	
Create Date	14/10/2005 01:43:14
Data Type	
Data Type Schema	
Date Last Modified	14/10/2005 01:43:14
Function Type	Table
I D	-94841804
Implementation Type	TransactSql
Is Deterministic	False
Is Encrypted	False
Is Schema Bound	False
Is System Object	True
Method Name	
Returns Null On Null Input	
System Type	
Table Variable Name	@colid

**Script:**

```

00001 create function sys.fn_replgetcolidfrombitmap (@columns binary(32))
00002 returns @colid table (colid int)
00003 as
00004 begin
00005     declare @current_word binary(2), @reverse_idx int, @bit_idx int, @mask int
00006
00007     if @columns is NULL
00008         return
00009     select @reverse_idx = 15
00010     while(@reverse_idx >= 0)
00011     begin
00012         select @current_word = convert( binary(2), substring( convert( nvarchar(16), @columns ), 16-@reverse_idx, 1
00013         »    ) )
00014         select @bit_idx = 15
00015         while(@bit_idx >= 0)
00016         begin
00017             select @mask = power(2, @bit_idx)
00018             if (@current_word & @mask) = @mask
00019                 insert @colid select @reverse_idx * 16 + @bit_idx + 1
00020             select @bit_idx = @bit_idx - 1
00021         end
00022         select @reverse_idx = @reverse_idx - 1
00023     end
00024 end

```

**Parameters**

Name	Type	Default
@columns	binary	no default

**[sys].[fn\_replgetparsedddlcmd]**

'UserDefinedFunction' in Database 'Northwind'

**Properties**

Property	Value
Ansi Nulls Status	False
Assembly Name	
Body Start Index	1504
Class Name	
Create Date	14/10/2005 01:45:30
Data Type	nvarchar
Data Type Schema	sys
Date Last Modified	14/10/2005 01:45:30
Function Type	Scalar
I D	-834653472
Implementation Type	TransactSql
Is Deterministic	False
Is Encrypted	False
Is Schema Bound	False
Is System Object	True
Method Name	
Returns Null On Null Input	
System Type	nvarchar
Table Variable Name	

**Script:**

```

00001 --
00002 -- Name: fn_replgetparsedddlcmd
00003 --
00004 -- Description: This helper function strips out the first part
00005 -- of DDL cmd, up to the point right after object name.
00006 --
00007 --
00008 -- Parameters:
00009 -- @ddlcmd nvarchar(max)
00010 -- ,@FirstToken sysname
00011 -- ,@objectType sysname -- complete form: e.g. procedure

```

**Parameters**

Name	Type	Default
@ddlcmd	nvarchar(-1)	no default
@FirstToken	sysname	no default
@objectType	sysname	no default
@dbname	sysname	no default
@owner	sysname	no default
@objname	sysname	no default
@targetobject	nvarchar(512)	no default

**[sys].[fn\_replreplacesinglequote]**

'UserDefinedFunction' in Database 'Northwind'

**Properties**

Property	Value
Ansi Nulls Status	False
Assembly Name	
Body Start Index	105
Class Name	
Create Date	14/10/2005 01:45:58
Data Type	nvarchar
Data Type Schema	sys
Date Last Modified	14/10/2005 01:45:58
Function Type	Scalar
I D	-718610251
Implementation Type	TransactSql
Is Deterministic	False
Is Encrypted	False
Is Schema Bound	False
Is System Object	True
Method Name	
Returns Null On Null Input	
System Type	nvarchar
Table Variable Name	

**Script:**

```

00001 create function sys.fn_replreplacesinglequote (
00002   @pstrin nvarchar(max) )
00003 returns nvarchar(max)
00004 as
00005 begin
00006   return REPLACE(@pstrin, N"", N"")
00007 end

```

**Parameters**

Name	Type	Default
@pstrin	nvarchar(-1)	no default

**[sys].[fn\_replreplacesinglequoteplusprotectstring]**

'UserDefinedFunction' in Database 'Northwind'

**Properties**

Property	Value
Ansi Nulls Status	False
Assembly Name	
Body Start Index	124
Class Name	
Create Date	14/10/2005 01:45:58
Data Type	nvarchar
Data Type Schema	sys
Date Last Modified	14/10/2005 01:45:58
Function Type	Scalar
I D	-1031773240
Implementation Type	TransactSql
Is Deterministic	False
Is Encrypted	False
Is Schema Bound	False
Is System Object	True
Method Name	
Returns Null On Null Input	
System Type	nvarchar
Table Variable Name	

**Script:**

```

00001 create function sys.fn_replreplacesinglequoteplusprotectstring (
00002   @pstrin nvarchar(4000) )
00003 returns nvarchar(4000)
00004 as
00005 begin
00006   declare @pstrout nvarchar(4000)
00007
00008   if (@pstrin is not null)
00009     begin
00010       select @pstrout = REPLACE(@pstrin, N'"', N'""')
00011       select @pstrout = N'"" + @pstrout + N'""'
00012     end
00013   else
00014     select @pstrout = N'NULL'
00015
00016   return @pstrout
00017 end

```

**Parameters**

Name	Type	Default
@pstrin	nvarchar(4000)	no default

**[sys].[fn\_repluniquename]**

'UserDefinedFunction' in Database 'Northwind'

**Properties**

Property	Value
Ansi Nulls Status	False
Assembly Name	
Body Start Index	261
Class Name	
Create Date	14/10/2005 01:43:18
Data Type	nvarchar
Data Type Schema	sys
Date Last Modified	14/10/2005 01:43:18
Function Type	Scalar
I D	-408691375
Implementation Type	TransactSql
Is Deterministic	False
Is Encrypted	False
Is Schema Bound	False
Is System Object	True
Method Name	
Returns Null On Null Input	
System Type	nvarchar
Table Variable Name	

**Script:**

```

00001 create function sys.fn_repluniquename (
00002 -- cannot use newid() in a UDF
00003     @guid uniqueidentifier = NULL,
00004     @prefix1 sysname = NULL,
00005     @prefix2 sysname = NULL,
00006     @prefix3 sysname = NULL,
00007     @prefix4 sysname = NULL
00008 ) returns nvarchar(100)
00009 as
00010 begin
00011     declare @name nvarchar(100)
00012     select @name = N''
00013
00014     -- Max len is (16 + 1) * 4 + 32 = 100
00015     if @prefix1 is not null
00016         select @name = @name + CONVERT(nvarchar(16), @prefix1) + N'-'
00017     if @prefix2 is not null
00018         select @name = @name + CONVERT(nvarchar(16), @prefix2) + N'-'
00019     if @prefix3 is not null
00020         select @name = @name + CONVERT(nvarchar(16), @prefix3) + N'-'
00021     if @prefix4 is not null
00022         select @name = @name + CONVERT(nvarchar(16), @prefix4) + N'-'
00023
00024     select @name = @name + replace(isnull(convert(nvarchar(36), @guid), N''), N'-', N'')
00025     return @name
00026 end

```

**Parameters**

Name	Type	Default
@guid	uniqueidentifier	no default
@prefix1	sysname	no default
@prefix2	sysname	no default
@prefix3	sysname	no default
@prefix4	sysname	no default

**[sys].[fn\_RowDumpCracker]**

'UserDefinedFunction' in Database 'Northwind'

**Properties**

Property	Value
Ansi Nulls Status	True
Assembly Name	
Body Start Index	1033
Class Name	
Create Date	14/10/2005 01:40:42
Data Type	
Data Type Schema	
Date Last Modified	14/10/2005 01:40:42
Function Type	Table
I D	-898718929
Implementation Type	TransactSql
Is Deterministic	False
Is Encrypted	False
Is Schema Bound	False
Is System Object	True
Method Name	
Returns Null On Null Input	
System Type	
Table Variable Name	@varColumns

**Script:**

```

00001 -----
00002 -- Name: sys.fn_RowDumpCracker
00003 --
00004 -- Description:
00005 -- Cracks the output of %%rowdump%% virtual column
00006 --
00007 -- Notes:
00008 -- 1. If inrowLength is 0 it implies the column is null or 0-length
00009 -- 2. If information about a column is not output then it must be a trailing
00010 -- null or 0-length variable length column.
00011 -- 3. Filters out columns that have been dropped.
00012 -- 4. Filters out uniquifiers.
00013 -- 5. Does not report clustering key columns in non-clustered indexes,
00014 -----
00015 CREATE FUNCTION sys.fn_RowDumpCracker (@rowdump AS varbinary (8000))
00016 RETURNS @varColumns TABLE
00017 (
00018 partition_id BIGINT NOT NULL, -- partition id to use for join against system views
00019 colName sysname NOT NULL, -- name of column
00020 isInrow BIT NOT NULL, -- is the column's value inlined or off-row.
00021 inrowLength INT NOT NULL -- what is the length of in-row content of the column
00022 )
00023 AS
00024 BEGIN
00025 declare @partition_id BIGINT
00026 declare @nvars INT
00027 declare @offset INT
00028 declare @i INT
00029 declare @colInfo INT
00030 declare @isInrow BIT
00031 declare @inrowLength INT
00032 declare @colOffset INT
00033 declare @colName sysname
00034 declare @partition_column_id INT
00035
00036 -- Initialize iterator offset to start from start of byte-bag.
00037 --
00038 set @offset = 1
00039
00040 -- Rowset id is first 8 bytes

```

```

00041 --
00042 select @partition_id = convert (bigint, convert (binary (8), reverse (substring (@rowdump, @offset, 8))))
00043 set @offset = @offset + 8
00044
00045 -- Column count is next 4 bytes
00046 --
00047 select @nvars = convert (int, convert (binary (4), reverse (substring (@rowdump, @offset, 4))))
00048 set @offset = @offset + 4
00049
00050 -- Extract information about each column
00051 --
00052 set @i = 0
00053 set @colOffset = -1
00054 while (@i < @nvars)
00055 BEGIN
00056     SET @colInfo = convert (int, convert (binary (2), reverse (substring (@rowdump, @offset, 2))))
00057     set @offset = @offset + 2
00058
00059     -- Most significant bit indicates whether column is in-row or off-row
00060     -- Remaining bits give length of inrow content
00061     --
00062     if ((@colInfo & 0x8000) <> 0)
00063     BEGIN
00064         set @isInrow = 0
00065         set @inrowLength = @colInfo - 0x8000
00066     END
00067     else
00068     BEGIN
00069         set @isInrow = 1
00070         set @inrowLength = @colInfo
00071     END
00072
00073     -- Output this column's information as long as it isnt a uniquifier,
00074     -- isnt dropped and has a name
00075     --
00076     select @partition_column_id = partition_column_id
00077     from sys.system_internals_partition_columns se
00078     where se.partition_id = @partition_id and
00079           se.leaf_offset = @colOffset and
00080           se.is_dropped = 0 and -- not dropped
00081           se.is_uniqueifier = 0 -- not uniqueifier
00082
00083     if (@partition_column_id is not null)
00084     begin
00085         select @colName = name
00086         from sys.columns cols join
00087             (select ic.column_id, ic.object_id
00088              from sys.columns ic join sys.system_internals_partitions sip
00089                on (ic.object_id = sip.object_id)
00090              where sip.partition_id = @partition_id and
00091                   ic.column_id = @partition_column_id) colidSrc
00092         on (colidSrc.object_id = cols.object_id and colidSrc.column_id = cols.column_id)
00093
00094         if (@colName is not null)
00095         begin
00096             insert @varColumns values (@partition_id, @colName, @isInrow, @inrowLength)
00097         end
00098     end
00099
00100     -- Move to next column
00101     --
00102     set @colOffset = @colOffset - 1
00103     set @i = @i + 1
00104 END
00105 RETURN
00106 END

```

**Parameters**

Name	Type	Default
@rowdump	varbinary(8000)	no default

**[sys].[fn\_serversharedrives]**

'UserDefinedFunction' in Database 'Northwind'

**Properties**

Property	Value
Ansi Nulls Status	True
Assembly Name	
Body Start Index	112
Class Name	
Create Date	14/10/2005 01:40:04
Data Type	
Data Type Schema	
Date Last Modified	14/10/2005 01:40:04
Function Type	Table
I D	-292273640
Implementation Type	TransactSql
Is Deterministic	False
Is Encrypted	False
Is Schema Bound	False
Is System Object	True
Method Name	
Returns Null On Null Input	
System Type	
Table Variable Name	@tab

**Script:**

```

00001 create function sys.fn_serversharedrives
00002 (
00003 )
00004
00005 returns @tab table(DriveName nchar(1) NOT NULL)
00006 as
00007 begin
00008     insert @tab
00009     select * from OpenRowset(serversharedrives)
00010
00011     return
00012 end -- fn_serversharedrives

```

**Parameters**

Name	Type	Default
------	------	---------

**[sys].[fn\_sqlvarbasetostr]**

'UserDefinedFunction' in Database 'Northwind'

**Properties**

Property	Value
Ansi Nulls Status	False
Assembly Name	
Body Start Index	96
Class Name	
Create Date	14/10/2005 01:45:59
Data Type	nvarchar
Data Type Schema	sys
Date Last Modified	14/10/2005 01:45:59
Function Type	Scalar
I D	-1051588973
Implementation Type	TransactSql
Is Deterministic	False
Is Encrypted	False
Is Schema Bound	False
Is System Object	True
Method Name	
Returns Null On Null Input	
System Type	nvarchar
Table Variable Name	

**Script:**

```

00001 create function sys.fn_sqlvarbasetostr (
00002     @ssvar sql_variant
00003 )
00004 returns nvarchar(max)
00005 as
00006 begin
00007     declare @pstrout nvarchar(max)
00008             ,@basetype sysname
00009
00010     select @basetype = lower(cast(SQL_VARIANT_PROPERTY ( @ssvar, 'BaseType' ) as sysname) collate
00011         » database_default)
00012     if (@ssvar IS NOT NULL and @basetype IS NOT NULL)
00013         begin
00014             if (@basetype = N'varchar')
00015                 select @pstrout = N'' + REPLACE(CAST(@ssvar as nvarchar(max)), '', '') + N''
00016             else if (@basetype in (N'nvarchar', N'xml'))
00017                 select @pstrout = N'' + REPLACE(CAST(@ssvar as nvarchar(max)), '', '') + N''
00018             else if (@basetype = N'char')
00019                 select @pstrout = N'' + REPLACE(RTRIM(CAST(@ssvar as nvarchar(max))), '', '') + N''
00020             else if (@basetype = N'nchar')
00021                 select @pstrout = N'' + REPLACE(RTRIM(CAST(@ssvar as nvarchar(max))), '', '') + N''
00022             else if (@basetype in (N'binary', N'varbinary'))
00023                 select @pstrout = sys.fn_varbintohexsubstring(1, CAST(@ssvar as varbinary(max)), 1, 0)
00024             else if (lower(@basetype collate SQL_Latin1_General_CP1_CS_AS) in ('bit', 'bigint', 'int', 'smallint', 'tinyint', 'decimal', 'numeric'))
00025                 select @pstrout = CAST(@ssvar as nvarchar(40))
00026             else if (lower(@basetype collate SQL_Latin1_General_CP1_CS_AS) in ('float', 'real'))
00027                 select @pstrout = CONVERT(nvarchar(60), @ssvar, 2)
00028             else if (@basetype in (N'money', N'smallmoney'))
00029                 select @pstrout = CONVERT(nvarchar(40), @ssvar, 2)
00030             else if (@basetype = N'uniqueidentifier')
00031                 select @pstrout = N'' + CAST(@ssvar as nvarchar(40)) + N''
00032             else if (@basetype in (N'datetime', N'smalldatetime'))
00033                 select @pstrout = N'' + CONVERT(nvarchar(40), @ssvar, 112) + N' ' + CONVERT(nvarchar(40), @ssvar,
00034         » 114) + N''
00035             else
00036                 select @pstrout = N'Invalid Datatype' + @basetype + N'(' + CAST(@ssvar as nvarchar(max)) + N')'
00037     end
00038 -- All done
00039 return @pstrout
00040 end

```

**Parameters**

Name	Type	Default
@ssvar	sql_variant	no default

**[sys].[fn\_trace\_geteventinfo]**

'UserDefinedFunction' in Database 'Northwind'

**Properties**

Property	Value
Ansi Nulls Status	True
Assembly Name	
Body Start Index	135
Class Name	
Create Date	14/10/2005 01:41:12
Data Type	
Data Type Schema	
Date Last Modified	14/10/2005 01:41:12
Function Type	Table
I D	-81646787
Implementation Type	TransactSql
Is Deterministic	False
Is Encrypted	False
Is Schema Bound	False
Is System Object	True
Method Name	
Returns Null On Null Input	
System Type	
Table Variable Name	@tab

**Script:**

```

00001 create function sys.fn_trace_geteventinfo
00002   (@handle int
00003   )
00004
00005 returns @tab table(eventid int NOT NULL,
00006   columnid int NOT NULL)
00007 as
00008 begin
00009   insert @tab
00010   select * from OpenRowset(TraceEventInfo, @handle)
00011
00012   return
00013 end -- fn_trace_geteventinfo

```

**Parameters**

Name	Type	Default
@handle	int	no default

**[sys].[fn\_trace\_getfilterinfo]**

'UserDefinedFunction' in Database 'Northwind'

**Properties**

Property	Value
Ansi Nulls Status	True
Assembly Name	
Body Start Index	206
Class Name	
Create Date	14/10/2005 01:41:12
Data Type	
Data Type Schema	
Date Last Modified	14/10/2005 01:41:12
Function Type	Table
I D	-99011299
Implementation Type	TransactSql
Is Deterministic	False
Is Encrypted	False
Is Schema Bound	False
Is System Object	True
Method Name	
Returns Null On Null Input	
System Type	
Table Variable Name	@tab

**Script:**

```

00001 create function sys.fn_trace_getfilterinfo
00002   (@handle int = 0
00003   )
00004
00005 returns @tab table(columnid int NOT NULL,
00006   logical_operator int NOT NULL,
00007   comparison_operator int NOT NULL,
00008   value sql_variant)
00009 as
00010 begin
00011   insert @tab
00012   select * from OpenRowset(TraceFilterInfo, @handle)
00013
00014   return
00015 end -- fn_trace_getfilterinfo

```

**Parameters**

Name	Type	Default
@handle	int	no default

**[sys].[fn\_trace\_getinfo]**

'UserDefinedFunction' in Database 'Northwind'

**Properties**

Property	Value
Ansi Nulls Status	True
Assembly Name	
Body Start Index	153
Class Name	
Create Date	14/10/2005 01:41:11
Data Type	
Data Type Schema	
Date Last Modified	14/10/2005 01:41:11
Function Type	Table
I D	-905021835
Implementation Type	TransactSql
Is Deterministic	False
Is Encrypted	False
Is Schema Bound	False
Is System Object	True
Method Name	
Returns Null On Null Input	
System Type	
Table Variable Name	@tab

**Script:**

```

00001 create function sys.fn_trace_getinfo
00002   (@handle int = 0
00003   )
00004 returns @tab table(traceid int NOT NULL,
00005   property int NOT NULL,
00006   value sql_variant)
00007 as
00008 begin
00009   insert @tab
00010   select * from OpenRowset(TraceInfo, @handle)
00011
00012   return
00013 end -- fn_trace_getinfo

```

**Parameters**

Name	Type	Default
@handle	int	no default

**[sys].[fn\_trace\_gettable]**

'UserDefinedFunction' in Database 'Northwind'

**Properties**

Property	Value
Ansi Nulls Status	True
Assembly Name	
Body Start Index	114
Class Name	
Create Date	14/10/2005 01:41:11
Data Type	
Data Type Schema	
Date Last Modified	14/10/2005 01:41:11
Function Type	Inline
I D	-718311503
Implementation Type	TransactSql
Is Deterministic	False
Is Encrypted	False
Is Schema Bound	False
Is System Object	True
Method Name	
Returns Null On Null Input	
System Type	
Table Variable Name	

**Script:**

```

00001 create function sys.fn_trace_gettable
00002   (@filename nvarchar(4000),
00003   @numfiles int = -1)
00004
00005 returns table as
00006 return select * from OpenRowset(TABLE TRCTABLE, @filename, @numfiles)

```

**Parameters**

Name	Type	Default
@filename	nvarchar(4000)	no default
@numfiles	int	no default

**[sys].[fn\_translate\_permissions]**

'UserDefinedFunction' in Database 'Northwind'

**Properties**

Property	Value
Ansi Nulls Status	True
Assembly Name	
Body Start Index	127
Class Name	
Create Date	14/10/2005 01:39:35
Data Type	
Data Type Schema	
Date Last Modified	14/10/2005 01:39:35
Function Type	Inline
I D	-795025293
Implementation Type	TransactSql
Is Deterministic	False
Is Encrypted	False
Is Schema Bound	False
Is System Object	True
Method Name	
Returns Null On Null Input	
System Type	
Table Variable Name	

**Script:**

```

00001 CREATE FUNCTION sys.fn_translate_permissions
00002     ( @level nvarchar(60), @perms varbinary(8))
00003 RETURNS table
00004 AS
00005 RETURN (
00006     SELECT * FROM OpenRowset(TranslatePermissions, @level, @perms)
00007 )

```

**Parameters**

Name	Type	Default
@level	nvarchar(60)	no default
@perms	varbinary(8)	no default

**[sys].[fn\_varbintohexstr]**

'UserDefinedFunction' in Database 'Northwind'

**Properties**

Property	Value
Ansi Nulls Status	False
Assembly Name	
Body Start Index	102
Class Name	
Create Date	14/10/2005 01:45:59
Data Type	nvarchar
Data Type Schema	sys
Date Last Modified	14/10/2005 01:45:59
Function Type	Scalar
I D	-565035985
Implementation Type	TransactSql
Is Deterministic	False
Is Encrypted	False
Is Schema Bound	False
Is System Object	True
Method Name	
Returns Null On Null Input	
System Type	nvarchar
Table Variable Name	

**Script:**

```

00001 create function sys.fn_varbintohexstr
00002 (
00003     @pbinin varbinary(max)
00004 )
00005 returns nvarchar(max)
00006 as
00007 begin
00008     return sys.fn_varbintohexsubstring(1, @pbinin, 1, 0)
00009 end

```

**Parameters**

Name	Type	Default
@pbinin	varbinary(-1)	no default

**[sys].[fn\_varbintohexsubstring]**

'UserDefinedFunction' in Database 'Northwind'

**Properties**

Property	Value
Ansi Nulls Status	False
Assembly Name	
Body Start Index	299
Class Name	
Create Date	14/10/2005 01:45:59
Data Type	nvarchar
Data Type Schema	sys
Date Last Modified	14/10/2005 01:45:59
Function Type	Scalar
I D	-403477184
Implementation Type	TransactSql
Is Deterministic	False
Is Encrypted	False
Is Schema Bound	False
Is System Object	True
Method Name	
Returns Null On Null Input	
System Type	nvarchar
Table Variable Name	

**Script:**

```

00001 create function sys.fn_varbintohexsubstring (
00002     @fsetprefix bit = 1           -- append '0x' to the output
00003     ,@pbinin varbinary(max) -- input binary stream
00004     ,@startoffset int = 1         -- starting offset
00005     ,@cbytesin int = 0           -- length of input to consider, 0 means total length
00006 )
00007 returns nvarchar(max)
00008 as
00009 begin
00010     declare @pstrout nvarchar(max)
00011             ,@i int
00012             ,@firstnibble int
00013             ,@secondnibble int
00014             ,@tempint int
00015             ,@hexstring char(16)
00016
00017     --
00018     -- initialize and validate
00019     --
00020     if (@pbinin IS NOT NULL)
00021     begin
00022         select @i = 0
00023                ,@cbytesin = case when (@cbytesin > 0) then @cbytesin else DATALENGTH(
00024     » @pbinin) end
00025                ,@pstrout = case when (@fsetprefix = 1) then N'0x' else N'' end
00026                ,@hexstring = '0123456789abcdef'
00027
00028         if ( ((@cbytesin * 2) + 2 > 4000) or ((@cbytesin * 2) + 2 < 1) )
00029             return NULL
00030
00031         if ( ( @startoffset > DATALENGTH(@pbinin) ) or ( @startoffset < 1 ) )
00032             return NULL
00033
00034         --
00035         -- adjust the length to process based on start offset and
00036         -- total length
00037         --
00038         if ((DATALENGTH(@pbinin) - @startoffset + 1) < @cbytesin)
00039             select @cbytesin = DATALENGTH(@pbinin) - @startoffset + 1
00040         --

```

```
00041      -- do for each byte
00042      --
00043      while (@i < @cbytesin)
00044      begin
00045          --
00046          -- Each byte has two nibbles
00047          -- which we convert to character
00048          --
00049          select @tempint = cast(substring(@pbinin, @i + @startoffset, 1) as int)
00050          select @firstnibble = @tempint
```

**Parameters**

Name	Type	Default
@fsetprefix	bit	no default
@pbinin	varbinary(-1)	no default
@startoffset	int	no default
@cbytesin	int	no default

**[sys].[fn\_virtualfilestats]**

'UserDefinedFunction' in Database 'Northwind'

**Properties**

Property	Value
Ansi Nulls Status	True
Assembly Name	
Body Start Index	469
Class Name	
Create Date	14/10/2005 01:40:34
Data Type	
Data Type Schema	
Date Last Modified	14/10/2005 01:40:34
Function Type	Table
I D	-337551382
Implementation Type	TransactSql
Is Deterministic	False
Is Encrypted	False
Is Schema Bound	False
Is System Object	True
Method Name	
Returns Null On Null Input	
System Type	
Table Variable Name	@tab

**Script:**

```

00001 create function sys.fn_virtualfilestats
00002 (
00003     @Databaseld Int = -1,
00004     @Fileld Int = -1
00005 )
00006 returns @tab table(Dbld SmallInt NOT NULL, Fileld SmallInt NOT NULL, TimeStamp Int NOT NULL,
00007     NumberReads BigInt NOT NULL, BytesRead BigInt NOT NULL, IoStallReadMS BigInt NOT NULL,
00008     NumberWrites BigInt NOT NULL, BytesWritten BigInt NOT NULL, IoStallWriteMS BigInt NOT NULL,
00009     IoStallMS BigInt NOT NULL, BytesOnDisk BigInt NOT NULL, FileHandle VarBinary(8) NOT NULL )
00010 as
00011 begin
00012     insert @tab
00013     select * from OpenRowset(VirtualFileStats, @Databaseld, @Fileld)
00014
00015     return
00016 end -- fn_virtualfilestats

```

**Parameters**

Name	Type	Default
@Databaseld	int	no default
@Fileld	int	no default

**[sys].[fn\_virtualservernodes]**

'UserDefinedFunction' in Database 'Northwind'

**Properties**

Property	Value
Ansi Nulls Status	True
Assembly Name	
Body Start Index	110
Class Name	
Create Date	14/10/2005 01:40:05
Data Type	
Data Type Schema	
Date Last Modified	14/10/2005 01:40:05
Function Type	Table
I D	-947370682
Implementation Type	TransactSql
Is Deterministic	False
Is Encrypted	False
Is Schema Bound	False
Is System Object	True
Method Name	
Returns Null On Null Input	
System Type	
Table Variable Name	@tab

**Script:**

```

00001 create function sys.fn_virtualservernodes
00002 (
00003 )
00004
00005 returns @tab table(NodeName sysname NOT NULL)
00006 as
00007 begin
00008     insert @tab
00009     select * from OpenRowset(virtualservernodes)
00010
00011     return
00012 end -- fn_virtualservernodes

```

**Parameters**

Name	Type	Default
------	------	---------

**[sys].[fn\_yukonsecuritymodelrequired]**

'UserDefinedFunction' in Database 'Northwind'

**Properties**

Property	Value
Ansi Nulls Status	False
Assembly Name	
Body Start Index	97
Class Name	
Create Date	14/10/2005 01:45:06
Data Type	bit
Data Type Schema	sys
Date Last Modified	14/10/2005 01:45:06
Function Type	Scalar
I D	-650457639
Implementation Type	TransactSql
Is Deterministic	False
Is Encrypted	False
Is Schema Bound	False
Is System Object	True
Method Name	
Returns Null On Null Input	
System Type	bit
Table Variable Name	

**Script:**

```

00001 CREATE FUNCTION sys.fn_yukonsecuritymodelrequired
00002 (
00003     @username sysname
00004 )
00005 RETURNS bit
00006 AS
00007 BEGIN
00008     -- if the user is sysadmin then we can allow fall back to 8.0 security
00009     return case when (@username IS NULL)
00010         then case when (IS_SRVROLEMEMBER(N'sysadmin') = 1) then 0 else 1 end
00011         else case when (IS_SRVROLEMEMBER(N'sysadmin', @username) = 1) then 0 else 1 end
00012         end
00013 END

```

**Parameters**

Name	Type	Default
@username	sysname	no default

**[dbo]**

'User' in Database 'Northwind'

**Database role membership**

db\_owner

**Schemas owned by this User**

dbo

**Properties**

Property	Value
Create Date	08/04/2003 09:10:42
Date Last Modified	27/07/2007 21:11:03
Default Schema	dbo
Has D B Access	True
Is System Object	True
Login	Vista-Dev\starprint
Login Type	WindowsUser
User Type	SqlLogin

**[guest]**

'User' in Database 'Northwind'

**Database role membership**

None

**Schemas owned by this User**

guest

**Properties**

Property	Value
Create Date	08/04/2003 09:10:42
Date Last Modified	08/04/2003 09:10:42
Default Schema	guest
Has D B Access	False
Is System Object	True
Login	
Login Type	SqlLogin
User Type	NoLogin

**[INFORMATION\_SCHEMA]**

'User' in Database 'Northwind'

**Database role membership**

None

**Schemas owned by this User**

INFORMATION\_SCHEMA

**Properties**

Property	Value
Create Date	14/10/2005 01:36:18
Date Last Modified	14/10/2005 01:36:18
Default Schema	
Has D B Access	False
Is System Object	True
Login	
Login Type	SqlLogin
User Type	NoLogin

**[sys]**

'User' in Database 'Northwind'

**Database role membership**

None

**Schemas owned by this User**

sys

**Properties**

Property	Value
Create Date	14/10/2005 01:36:18
Date Last Modified	14/10/2005 01:36:18
Default Schema	
Has D B Access	False
Is System Object	True
Login	
Login Type	SqlLogin
User Type	NoLogin