

Database Name : Northwind

Server Name : vs-rameshkumar\sqlexpress

Total Number of Tables : 13

Total Number of Views : 16

Total Number of Stored Procedures : 14

Tables

1 . Orders , Created on :12/13/2004 4:11:10 PM

Column Name	Primary Key	Ordinal Position	Default Value	Null Allowed	Data Type	DPre cision	Check Clause	Description
OrderID	PK_Orders	1		NO	int	10		
CustomerID		2		YES	nchar	5		
EmployeeID		3		YES	int	10		
OrderDate		4		YES	datetime	3		
RequiredDate		5		YES	datetime	3		
ShippedDate		6		YES	datetime	3		
ShipVia		7		YES	int	10		
Freight		8	(0)	YES	money	19		
ShipName		9		YES	nvarchar	40		
ShipAddress		10		YES	nvarchar	60		
ShipCity		11		YES	nvarchar	15		
ShipRegion		12		YES	nvarchar	15		
ShipPostalCode		13		YES	nvarchar	10		
ShipCountry		14		YES	nvarchar	15		

Indexes

1 .PK_Orders

Name	PK_Orders
Table Name	Orders
Index Type	Clustered
Unique	False
Index FillFactor	0
Primary Key	True
Ignore Duplicate Row	False

Columns

Column Name	Sort Order
OrderID	Ascending

2 .CustomerID

Name	CustomerID
Table Name	Orders
Index Type	Non-Clustered
Unique	True
Index FillFactor	0
Primary Key	False
Ignore Duplicate Row	False

Columns

Column Name	Sort Order
CustomerID	Ascending

3 .CustomersOrders

Name	CustomersOrders
Table Name	Orders

Index Type	Non-Clustered
Unique	True
Index FillFactor	0
Primary Key	False
Ignore Duplicate Row	False

Columns

Column Name	Sort Order
CustomerID	Ascending

4 .EmployeeID

Name	EmployeeID
Table Name	Orders
Index Type	Non-Clustered
Unique	True
Index FillFactor	0
Primary Key	False
Ignore Duplicate Row	False

Columns

Column Name	Sort Order
EmployeeID	Ascending

5 .EmployeesOrders

Name	EmployeesOrders
Table Name	Orders
Index Type	Non-Clustered
Unique	True
Index FillFactor	0
Primary Key	False
Ignore Duplicate Row	False

Columns

Column Name	Sort Order
EmployeeID	Ascending

6 .OrderDate

Name	OrderDate
Table Name	Orders
Index Type	Non-Clustered
Unique	True
Index FillFactor	0
Primary Key	False
Ignore Duplicate Row	False

Columns

Column Name	Sort Order
OrderDate	Ascending

7 .ShippedDate

Name	ShippedDate
------	-------------

Table Name	Orders
Index Type	Non-Clustered
Unique	True
Index FillFactor	0
Primary Key	False
Ignore Duplicate Row	False

Columns

Column Name	Sort Order
ShippedDate	Ascending

8 .ShippersOrders

Name	ShippersOrders
Table Name	Orders
Index Type	Non-Clustered
Unique	True
Index FillFactor	0
Primary Key	False
Ignore Duplicate Row	False

Columns

Column Name	Sort Order
ShipVia	Ascending

9 .ShipPostalCode

Name	ShipPostalCode
Table Name	Orders
Index Type	Non-Clustered
Unique	True
Index FillFactor	0
Primary Key	False
Ignore Duplicate Row	False

Columns

Column Name	Sort Order
ShipPostalCode	Ascending

Foreign Keys

Name	Primary Key Table	Foreign Key Table	Foreign Key Column	Update Rule	Delete Rule	Primary Key Column
FK_Orders_Customers	Customers	Orders	CustomerID	NO ACTION	NO ACTION	
FK_Orders_Employees	Employees	Orders	EmployeeID	NO ACTION	NO ACTION	
FK_Orders_Shippers	Shippers	Orders	ShipVia	NO ACTION	NO ACTION	

Privileges

Grantee	Privilege	Grantable
dbo	DELETE	YES
dbo	INSERT	YES
dbo	REFERENCES	YES

dbo	SELECT	YES
dbo	UPDATE	YES

2 . Products , Created on :12/13/2004 4:11:10 PM

Column Name	Primary Key	Ordinal Position	Default Value	Null Allowed	Data Type	DPre cision	Check Clause	Description
ProductID	PK_Products	1		NO	int	10		
ProductName		2		NO	nvarchar	40		
SupplierID		3		YES	int	10		
CategoryID		4		YES	int	10		
QuantityPerUnit		5		YES	nvarchar	20		
UnitPrice		6	(0)	YES	money	19		
UnitsInStock		7	(0)	YES	smallint	5		
UnitsOnOrder		8	(0)	YES	smallint	5		
ReorderLevel		9	(0)	YES	smallint	5		
Discontinued		10	(0)	NO	bit			

Indexes

1 .PK_Products

Name	PK_Products
Table Name	Products
Index Type	Clustered
Unique	False
Index FillFactor	0
Primary Key	True
Ignore Duplicate Row	False

Columns

Column Name	Sort Order
ProductID	Ascending

2 .CategoriesProducts

Name	CategoriesProducts
Table Name	Products
Index Type	Non-Clustered
Unique	True
Index FillFactor	0
Primary Key	False
Ignore Duplicate Row	False

Columns

Column Name	Sort Order
CategoryID	Ascending

3 .CategoryID

Name	CategoryID
Table Name	Products
Index Type	Non-Clustered
Unique	True
Index FillFactor	0
Primary Key	False
Ignore Duplicate Row	False

Columns

Column Name	Sort Order
CategoryID	Ascending

4 .ProductName

Name	ProductName
Table Name	Products
Index Type	Non-Clustered
Unique	True
Index FillFactor	0
Primary Key	False
Ignore Duplicate Row	False

Columns

Column Name	Sort Order
ProductName	Ascending

5 .SupplierID

Name	SupplierID
Table Name	Products
Index Type	Non-Clustered
Unique	True
Index FillFactor	0
Primary Key	False
Ignore Duplicate Row	False

Columns

Column Name	Sort Order
SupplierID	Ascending

6 .SuppliersProducts

Name	SuppliersProducts
Table Name	Products
Index Type	Non-Clustered
Unique	True
Index FillFactor	0
Primary Key	False
Ignore Duplicate Row	False

Columns

Column Name	Sort Order
SupplierID	Ascending

Foreign Keys

Name	Primary Key Table	Foreign Key Table	Foreign Key Column	Update Rule	Delete Rule	Primary Key Column
FK_Products_Categories	Categories	Products	CategoryID	NO ACTION	NO ACTION	
FK_Products_Suppliers	Suppliers	Products	SupplierID	NO ACTION	NO ACTION	

Privileges

Grantee	Privilege	Grantable
dbo	DELETE	YES
dbo	INSERT	YES
dbo	REFERENCES	YES
dbo	SELECT	YES
dbo	UPDATE	YES

3 . Order Details , Created on :12/13/2004 4:11:10 PM

Column Name	Primary Key	Ordinal Position	Default Value	Null Allowed	Data Type	DPre cision	Check Clause	Description
OrderID	PK_Order_Details	1		NO	int	10		
ProductID	PK_Order_Details	2		NO	int	10		
UnitPrice		3	(0)	NO	money	19		
Quantity		4	(1)	NO	smallint	5		
Discount		5	(0)	NO	real	24		

Indexes

1 .PK_Order_Details

Name	PK_Order_Details
Table Name	Order Details
Index Type	Clustered
Unique	False
Index FillFactor	0
Primary Key	True
Ignore Duplicate Row	False

Columns

Column Name	Sort Order
OrderID	Ascending
ProductID	Ascending

2 .OrderID

Name	OrderID
Table Name	Order Details
Index Type	Non-Clustered
Unique	True
Index FillFactor	0
Primary Key	False
Ignore Duplicate Row	False

Columns

Column Name	Sort Order
OrderID	Ascending

3 .OrdersOrder_Details

Name	OrdersOrder_Details
Table Name	Order Details
Index Type	Non-Clustered
Unique	True
Index FillFactor	0
Primary Key	False
Ignore Duplicate Row	False

Columns

Column Name	Sort Order

OrderID	Ascending
---------	-----------

4 .ProductID

Name	ProductID
Table Name	Order Details
Index Type	Non-Clustered
Unique	True
Index FillFactor	0
Primary Key	False
Ignore Duplicate Row	False

Columns

Column Name	Sort Order
ProductID	Ascending

5 .ProductsOrder_Details

Name	ProductsOrder_Details
Table Name	Order Details
Index Type	Non-Clustered
Unique	True
Index FillFactor	0
Primary Key	False
Ignore Duplicate Row	False

Columns

Column Name	Sort Order
ProductID	Ascending

Foreign Keys

Name	Primary Key Table	Foreign Key Table	Foreign Key Column	Update Rule	Delete Rule	Primary Key Column
FK_Order_Details_Orders	Orders	Order Details	OrderID	NO ACTION	NO ACTION	
FK_Order_Details_Products	Products	Order Details	ProductID	NO ACTION	NO ACTION	

Privileges

Grantee	Privilege	Grantable
dbo	DELETE	YES
dbo	INSERT	YES
dbo	REFERENCES	YES
dbo	SELECT	YES
dbo	UPDATE	YES

4 . CustomerCustomerDemo , Created on :12/13/2004 4:11:17 PM

Column Name	Primary Key	Ordinal Position	Default Value	Null Allowed	Data Type	DPre cision	Check Clause	Description
CustomerID	PK_CustomerCustomerDemo	1		NO	nchar	5		
CustomerTypeID	PK_CustomerCustomerDemo	2		NO	nchar	10		

Indexes

1 .PK_CustomerCustomerDemo

Name	PK_CustomerCustomerDemo
Table Name	CustomerCustomerDemo
Index Type	Non-Clustered
Unique	False
Index FillFactor	0
Primary Key	True
Ignore Duplicate Row	False

Columns

Column Name	Sort Order
CustomerID	Ascending
CustomerTypeID	Ascending

Foreign Keys

Name	Primary Key Table	Foreign Key Table	Foreign Key Column	Update Rule	Delete Rule	Primary Key Column
FK_CustomerCustomerDemo	CustomerDemographics	CustomerCustomerDemo	CustomerTypeID	NO ACTION	NO ACTION	
FK_CustomerCustomerDemo_Customers	Customers	CustomerCustomerDemo	CustomerID	NO ACTION	NO ACTION	

Privileges

Grantee	Privilege	Grantable
dbo	DELETE	YES
dbo	INSERT	YES
dbo	REFERENCES	YES
dbo	SELECT	YES
dbo	UPDATE	YES

5 . CustomerDemographics , Created on :12/13/2004 4:11:17 PM

Column Name	Primary Key	Ordinal Position	Default Value	Null Allowed	Data Type	DPre cision	Check Clause	Description
CustomerTypeID	PK_CustomerDemographics	1		NO	nchar	10		
CustomerDesc		2		YES	ntext	107374 1823		

Indexes

1 .PK_CustomerDemographics

Name	PK_CustomerDemographics
Table Name	CustomerDemographics
Index Type	Non-Clustered
Unique	False
Index FillFactor	0
Primary Key	True
Ignore Duplicate Row	False

Columns

Column Name	Sort Order
CustomerTypeID	Ascending

Privileges

Grantee	Privilege	Grantable
dbo	DELETE	YES
dbo	INSERT	YES
dbo	REFERENCES	YES
dbo	SELECT	YES
dbo	UPDATE	YES

6 . Region , Created on :12/13/2004 4:11:17 PM

Column Name	Primary Key	Ordinal Position	Default Value	Null Allowed	Data Type	DPre cision	Check Clause	Description
RegionID	PK_Region	1		NO	int	10		
RegionDescription		2		NO	nchar	50		

Indexes

1 .PK_Region

Name	PK_Region
Table Name	Region
Index Type	Non-Clustered
Unique	False
Index FillFactor	0
Primary Key	True
Ignore Duplicate Row	False

Columns

Column Name	Sort Order
RegionID	Ascending

Privileges

Grantee	Privilege	Grantable
dbo	DELETE	YES
dbo	INSERT	YES
dbo	REFERENCES	YES
dbo	SELECT	YES
dbo	UPDATE	YES

7 . Territories , Created on :12/13/2004 4:11:17 PM

Column Name	Primary Key	Ordinal Position	Default Value	Null Allowed	Data Type	DPre cision	Check Clause	Description
TerritoryID	PK_Territories	1		NO	nvarchar	20		
TerritoryDescription		2		NO	nchar	50		
RegionID		3		NO	int	10		

Indexes

1 .PK_Territories

Name	PK_Territories
Table Name	Territories
Index Type	Non-Clustered
Unique	False
Index FillFactor	0
Primary Key	True
Ignore Duplicate Row	False

Columns

Column Name	Sort Order
TerritoryID	Ascending

2 ._WA_Sys_RegionID_35BCFE0A

Name	._WA_Sys_RegionID_35BCFE0A
Table Name	Territories
Index Type	Non-Clustered
Unique	True
Index FillFactor	0
Primary Key	False
Ignore Duplicate Row	False

Foreign Keys

Name	Primary Key Table	Foreign Key Table	Foreign Key Column	Update Rule	Delete Rule	Primary Key Column
FK_Territories_Region	Region	Territories	RegionID	NO ACTION	NO ACTION	

Privileges

Grantee	Privilege	Grantable
dbo	DELETE	YES
dbo	INSERT	YES
dbo	REFERENCES	YES
dbo	SELECT	YES
dbo	UPDATE	YES

8 . EmployeeTerritories , Created on :12/13/2004 4:11:17 PM

Column Name	Primary Key	Ordinal Position	Default Value	Null Allowed	Data Type	DPre cision	Check Clause	Description
EmployeeID	PK_EmployeeTerritories	1		NO	int	10		
TerritoryID	PK_EmployeeTerritories	2		NO	nvarchar	20		

Indexes

1 .PK_EmployeeTerritories

Name	PK_EmployeeTerritories
Table Name	EmployeeTerritories
Index Type	Non-Clustered
Unique	False
Index FillFactor	0
Primary Key	True
Ignore Duplicate Row	False

Columns

Column Name	Sort Order
EmployeeID	Ascending
TerritoryID	Ascending

2 ._WA_Sys_TerritoryID_36B12243

Name	._WA_Sys_TerritoryID_36B12243
Table Name	EmployeeTerritories
Index Type	Non-Clustered
Unique	True
Index FillFactor	0
Primary Key	False
Ignore Duplicate Row	False

Foreign Keys

Name	Primary Key Table	Foreign Key Table	Foreign Key Column	Update Rule	Delete Rule	Primary Key Column
FK_EmployeeTerritories_Employees	Employees	EmployeeTerritories	EmployeeID	NO ACTION	NO ACTION	
FK_EmployeeTerritories_Territories	Territories	EmployeeTerritories	TerritoryID	NO ACTION	NO ACTION	

Privileges

Grantee	Privilege	Grantable
dbo	DELETE	YES
dbo	INSERT	YES
dbo	REFERENCES	YES
dbo	SELECT	YES
dbo	UPDATE	YES

9 . Employees , Created on :12/13/2004 4:11:09 PM

Column Name	Primary Key	Ordinal Position	Default Value	Null Allowed	Data Type	DPre cision	Check Clause	Description
EmployeeID	PK_Employees	1		NO	int	10		
LastName		2		NO	nvarchar	20		
FirstName		3		NO	nvarchar	10		
Title		4		YES	nvarchar	30		
TitleOfCourtesy		5		YES	nvarchar	25		
BirthDate		6		YES	datetime	3		
HireDate		7		YES	datetime	3		
Address		8		YES	nvarchar	60		
City		9		YES	nvarchar	15		
Region		10		YES	nvarchar	15		
PostalCode		11		YES	nvarchar	10		
Country		12		YES	nvarchar	15		
HomePhone		13		YES	nvarchar	24		
Extension		14		YES	nvarchar	4		
Photo		15		YES	image	2147483647		
Notes		16		YES	ntext	1073741823		
ReportsTo		17		YES	int	10		
PhotoPath		18		YES	nvarchar	255		

Indexes

1 .PK_Employees

Name	PK_Employees
Table Name	Employees
Index Type	Clustered
Unique	False
Index FillFactor	0
Primary Key	True
Ignore Duplicate Row	False

Columns

Column Name	Sort Order
EmployeeID	Ascending

2 .LastName

Name	LastName
Table Name	Employees
Index Type	Non-Clustered
Unique	True
Index FillFactor	0
Primary Key	False
Ignore Duplicate Row	False

Columns

Column Name	Sort Order
LastName	Ascending

3 .PostalCode

Name	PostalCode
Table Name	Employees
Index Type	Non-Clustered
Unique	True
Index FillFactor	0
Primary Key	False
Ignore Duplicate Row	False

Columns

Column Name	Sort Order
PostalCode	Ascending

Foreign Keys

Name	Primary Key Table	Foreign Key Table	Foreign Key Column	Update Rule	Delete Rule	Primary Key Column
FK_Employees_Employees	Employees	Employees	ReportsTo	NO ACTION	NO ACTION	

Privileges

Grantee	Privilege	Grantable
dbo	DELETE	YES
dbo	INSERT	YES
dbo	REFERENCES	YES
dbo	SELECT	YES
dbo	UPDATE	YES

10 . Categories , Created on :12/13/2004 4:11:09 PM

Column Name	Primary Key	Ordinal Position	Default Value	Null Allowed	Data Type	DPre cision	Check Clause	Description
CategoryID	PK_Categories	1		NO	int	10		
CategoryName		2		NO	nvarchar	15		
Description		3		YES	ntext	107374 1823		
Picture		4		YES	image	214748 3647		

Indexes

1 .PK_Categories

Name	PK_Categories
Table Name	Categories
Index Type	Clustered
Unique	False
Index FillFactor	0
Primary Key	True
Ignore Duplicate Row	False

Columns

Column Name	Sort Order
CategoryID	Ascending

2 .CategoryName

Name	CategoryName
Table Name	Categories
Index Type	Non-Clustered
Unique	True
Index FillFactor	0
Primary Key	False
Ignore Duplicate Row	False

Columns

Column Name	Sort Order
CategoryName	Ascending

Privileges

Grantee	Privilege	Grantable
dbo	DELETE	YES
dbo	INSERT	YES
dbo	REFERENCES	YES
dbo	SELECT	YES
dbo	UPDATE	YES

11 . Customers , Created on :12/13/2004 4:11:10 PM

Column Name	Primary Key	Ordinal Position	Default Value	Null Allowed	Data Type	DPre cision	Check Clause	Description
CustomerID	PK_Customers	1		NO	nchar	5		
CompanyName		2		NO	nvarchar	40		
ContactName		3		YES	nvarchar	30		
ContactTitle		4		YES	nvarchar	30		
Address		5		YES	nvarchar	60		
City		6		YES	nvarchar	15		
Region		7		YES	nvarchar	15		
PostalCode		8		YES	nvarchar	10		
Country		9		YES	nvarchar	15		
Phone		10		YES	nvarchar	24		
Fax		11		YES	nvarchar	24		

Indexes

1 .PK_Customers

Name	PK_Customers
Table Name	Customers
Index Type	Clustered
Unique	False
Index FillFactor	0
Primary Key	True
Ignore Duplicate Row	False

Columns

Column Name	Sort Order
CustomerID	Ascending

2 .City

Name	City
Table Name	Customers
Index Type	Non-Clustered
Unique	True
Index FillFactor	0
Primary Key	False
Ignore Duplicate Row	False

Columns

Column Name	Sort Order
City	Ascending

3 .CompanyName

Name	CompanyName
Table Name	Customers
Index Type	Non-Clustered
Unique	True
Index FillFactor	0
Primary Key	False

Ignore Duplicate Row	False
----------------------	-------

Columns

Column Name	Sort Order
CompanyName	Ascending

4 .PostalCode

Name	PostalCode
Table Name	Customers
Index Type	Non-Clustered
Unique	True
Index FillFactor	0
Primary Key	False
Ignore Duplicate Row	False

Columns

Column Name	Sort Order
PostalCode	Ascending

5 .Region

Name	Region
Table Name	Customers
Index Type	Non-Clustered
Unique	True
Index FillFactor	0
Primary Key	False
Ignore Duplicate Row	False

Columns

Column Name	Sort Order
Region	Ascending

Privileges

Grantee	Privilege	Grantable
dbo	DELETE	YES
dbo	INSERT	YES
dbo	REFERENCES	YES
dbo	SELECT	YES
dbo	UPDATE	YES

12 . Shippers , Created on :12/13/2004 4:11:10 PM

Column Name	Primary Key	Ordinal Position	Default Value	Null Allowed	Data Type	DPre cision	Check Clause	Description
ShipperID	PK_Shippers	1		NO	int	10		
CompanyName		2		NO	nvarchar	40		
Phone		3		YES	nvarchar	24		

Indexes

1 .PK_Shippers

Name	PK_Shippers
Table Name	Shippers
Index Type	Clustered
Unique	False
Index FillFactor	0
Primary Key	True
Ignore Duplicate Row	False

Columns

Column Name	Sort Order
ShipperID	Ascending

Privileges

Grantee	Privilege	Grantable
dbo	DELETE	YES
dbo	INSERT	YES
dbo	REFERENCES	YES
dbo	SELECT	YES
dbo	UPDATE	YES

13 . Suppliers , Created on :12/13/2004 4:11:10 PM

Column Name	Primary Key	Ordinal Position	Default Value	Null Allowed	Data Type	DPre cision	Check Clause	Description
SupplierID	PK_Suppliers	1		NO	int	10		
CompanyName		2		NO	nvarchar	40		
ContactName		3		YES	nvarchar	30		
ContactTitle		4		YES	nvarchar	30		
Address		5		YES	nvarchar	60		
City		6		YES	nvarchar	15		
Region		7		YES	nvarchar	15		
PostalCode		8		YES	nvarchar	10		
Country		9		YES	nvarchar	15		
Phone		10		YES	nvarchar	24		
Fax		11		YES	nvarchar	24		
HomePage		12		YES	ntext	107374 1823		

Indexes

1 .PK_Suppliers

Name	PK_Suppliers
Table Name	Suppliers
Index Type	Clustered
Unique	False
Index FillFactor	0
Primary Key	True
Ignore Duplicate Row	False

Columns

Column Name	Sort Order
SupplierID	Ascending

2 .CompanyName

Name	CompanyName
Table Name	Suppliers
Index Type	Non-Clustered
Unique	True
Index FillFactor	0
Primary Key	False
Ignore Duplicate Row	False

Columns

Column Name	Sort Order
CompanyName	Ascending

3 .PostalCode

Name	PostalCode
Table Name	Suppliers
Index Type	Non-Clustered
Unique	True
Index FillFactor	0

Primary Key	False
Ignore Duplicate Row	False

Columns

Column Name	Sort Order
PostalCode	Ascending

Privileges

Grantee	Privilege	Grantable
dbo	DELETE	YES
dbo	INSERT	YES
dbo	REFERENCES	YES
dbo	SELECT	YES
dbo	UPDATE	YES

VIEWS

1. Customer and Suppliers by City

```
create view "Customer and Suppliers by City" AS
SELECT City, CompanyName, ContactName, 'Customers' AS Relationship
FROM Customers
UNION SELECT City, CompanyName, ContactName, 'Suppliers'
FROM Suppliers
--ORDER BY City, CompanyName
```

Dependencies

Table Name	Column Name
Customers	CompanyName
Customers	ContactName
Customers	City
Suppliers	CompanyName
Suppliers	ContactName
Suppliers	City

2. Alphabetical list of products

```
create view "Alphabetical list of products" AS
SELECT Products.*, Categories.CategoryName
FROM Categories INNER JOIN Products ON Categories.CategoryID = Products.CategoryID
WHERE (((Products.Discontinued)=0))
```

Dependencies

Table Name	Column Name
Products	ProductID
Products	ProductName
Products	SupplierID
Products	CategoryID
Products	QuantityPerUnit
Products	UnitPrice
Products	UnitsInStock
Products	UnitsOnOrder
Products	ReorderLevel
Products	Discontinued
Categories	CategoryID
Categories	CategoryName

3. Current Product List

```
create view "Current Product List" AS
SELECT Product_List.ProductID, Product_List.ProductName
FROM Products AS Product_List
WHERE (((Product_List.Discontinued)=0))
--ORDER BY Product_List.ProductName
```

Dependencies

Table Name	Column Name
Products	ProductID
Products	ProductName
Products	Discontinued

4. Orders Qry

create view "Orders Qry" AS

```
SELECT Orders.OrderID, Orders.CustomerID, Orders.EmployeeID, Orders.OrderDate, Orders.RequiredDate,
Orders.ShippedDate, Orders.ShipVia, Orders.Freight, Orders.ShipName, Orders.ShipAddress, Orders.ShipCity,
Orders.ShipRegion, Orders.ShipPostalCode, Orders.ShipCountry,
Customers.CompanyName, Customers.Address, Customers.City, Customers.Region, Customers.PostalCode, Customers.Country
FROM Customers INNER JOIN Orders ON Customers.CustomerID = Orders.CustomerID
```

Dependencies

Table Name	Column Name
Orders	OrderID
Orders	CustomerID
Orders	EmployeeID
Orders	OrderDate
Orders	RequiredDate
Orders	ShippedDate
Orders	ShipVia
Orders	Freight
Orders	ShipName
Orders	ShipAddress
Orders	ShipCity
Orders	ShipRegion
Orders	ShipPostalCode
Orders	ShipCountry
Customers	CustomerID
Customers	CompanyName
Customers	Address
Customers	City
Customers	Region
Customers	PostalCode
Customers	Country

5. Products Above Average Price

```
create view "Products Above Average Price" AS
SELECT Products.ProductName, Products.UnitPrice
FROM Products
WHERE Products.UnitPrice>(SELECT AVG(UnitPrice) From Products)
--ORDER BY Products.UnitPrice DESC
```

Dependencies

Table Name	Column Name
Products	ProductName
Products	UnitPrice

6. Products by Category

create view "Products by Category" AS

```
SELECT Categories.CategoryName, Products.ProductName, Products.QuantityPerUnit, Products.UnitsInStock, Products.Discontinued
FROM Categories INNER JOIN Products ON Categories.CategoryID = Products.CategoryID
WHERE Products.Discontinued <> 1
--ORDER BY Categories.CategoryName, Products.ProductName
```

Dependencies

Table Name	Column Name
Products	ProductName
Products	CategoryID
Products	QuantityPerUnit
Products	UnitsInStock
Products	Discontinued
Categories	CategoryID
Categories	CategoryName

7. Quarterly Orders

create view "Quarterly Orders" AS

```
SELECT DISTINCT Customers.CustomerID, Customers.CompanyName, Customers.City, Customers.Country  
FROM Customers RIGHT JOIN Orders ON Customers.CustomerID = Orders.CustomerID  
WHERE Orders.OrderDate BETWEEN '19970101' And '19971231'
```

Dependencies

Table Name	Column Name
Orders	CustomerID
Orders	OrderDate
Customers	CustomerID
Customers	CompanyName
Customers	City
Customers	Country

8. Invoices

create view Invoices AS

```
SELECT Orders.ShipName, Orders.ShipAddress, Orders.ShipCity, Orders.ShipRegion, Orders.ShipPostalCode,
Orders.ShipCountry, Orders.CustomerID, Customers.CompanyName AS CustomerName, Customers.Address, Customers.City,
Customers.Region, Customers.PostalCode, Customers.Country,
(FirstName + ' ' + LastName) AS Salesperson,
Orders.OrderID, Orders.OrderDate, Orders.RequiredDate, Orders.ShippedDate, Shippers.CompanyName As ShipperName,
"Order Details".ProductID, Products.ProductName, "Order Details".UnitPrice, "Order Details".Quantity,
"Order Details".Discount,
(CONVERT(money,("Order Details".UnitPrice*Quantity*(1-Discout)/100))*100) AS ExtendedPrice, Orders.Freight
FROM Shippers INNER JOIN
(Products INNER JOIN
(
(Employees INNER JOIN
(Customers INNER JOIN Orders ON Customers.CustomerID = Orders.CustomerID)
ON Employees.EmployeeID = Orders.EmployeeID)
INNER JOIN "Order Details" ON Orders.OrderID = "Order Details".OrderID)
ON Products.ProductID = "Order Details".ProductID)
ON Shippers.ShipperID = Orders.ShipVia
```

Dependencies

Table Name	Column Name
Orders	OrderID
Orders	CustomerID
Orders	EmployeeID
Orders	OrderDate
Orders	RequiredDate
Orders	ShippedDate
Orders	ShipVia
Orders	Freight
Orders	ShipName
Orders	ShipAddress
Orders	ShipCity
Orders	ShipRegion
Orders	ShipPostalCode
Orders	ShipCountry
Products	ProductID
Products	ProductName
Order Details	OrderID
Order Details	ProductID
Order Details	UnitPrice
Order Details	Quantity
Order Details	Discount
Employees	EmployeeID
Employees	LastName
Employees	FirstName
Customers	CustomerID

Customers	CompanyName
Customers	Address
Customers	City
Customers	Region
Customers	PostalCode
Customers	Country
Shippers	ShipperID
Shippers	CompanyName

9. Order Details Extended

```
create view "Order Details Extended" AS
SELECT "Order Details".OrderID, "Order Details".ProductID, Products.ProductName,
"Order Details".UnitPrice, "Order Details".Quantity, "Order Details".Discount,
(CONVERT(money,("Order Details".UnitPrice*Quantity*(1-Discout)/100))*100) AS ExtendedPrice
FROM Products INNER JOIN "Order Details" ON Products.ProductID = "Order Details".ProductID
--ORDER BY "Order Details".OrderID
```

Dependencies

Table Name	Column Name
Products	ProductID
Products	ProductName
Order Details	OrderID
Order Details	ProductID
Order Details	UnitPrice
Order Details	Quantity
Order Details	Discount

10. Order Subtotals

create view "Order Subtotals" AS

```
SELECT "Order Details".OrderID, Sum(CONVERT(money,("Order Details".UnitPrice*Quantity*(1-Discout)/100))*100) AS Subtotal  
FROM "Order Details"  
GROUP BY "Order Details".OrderID
```

Dependencies

Table Name	Column Name
Order Details	OrderID
Order Details	UnitPrice
Order Details	Quantity
Order Details	Discount

11. Product Sales for 1997

```

create view "Product Sales for 1997" AS
SELECT Categories.CategoryName, Products.ProductName,
Sum(CONVERT(money,("Order Details".UnitPrice*Quantity*(1-Discout)/100))*100) AS ProductSales
FROM (Categories INNER JOIN Products ON Categories.CategoryID = Products.CategoryID)
INNER JOIN (Orders
INNER JOIN "Order Details" ON Orders.OrderID = "Order Details".OrderID)
ON Products.ProductID = "Order Details".ProductID
WHERE (((Orders.ShippedDate) Between '19970101' And '19971231'))
GROUP BY Categories.CategoryName, Products.ProductName

```

Dependencies

Table Name	Column Name
Orders	OrderID
Orders	ShippedDate
Products	ProductID
Products	ProductName
Products	CategoryID
Order Details	OrderID
Order Details	ProductID
Order Details	UnitPrice
Order Details	Quantity
Order Details	Discount
Categories	CategoryID
Categories	CategoryName

12. Category Sales for 1997

create view "Category Sales for 1997" AS

```
SELECT "Product Sales for 1997".CategoryName, Sum("Product Sales for 1997".ProductSales) AS CategorySales
```

```
FROM "Product Sales for 1997"
```

```
GROUP BY "Product Sales for 1997".CategoryName
```

Dependencies

Table Name	Column Name
Product Sales for 1997	CategoryName
Product Sales for 1997	ProductSales

13. Sales by Category

```

create view "Sales by Category" AS
SELECT Categories.CategoryID, Categories.CategoryName, Products.ProductName,
Sum("Order Details Extended".ExtendedPrice) AS ProductSales
FROM Categories INNER JOIN
(Products INNER JOIN
(Orders INNER JOIN "Order Details Extended" ON Orders.OrderID = "Order Details Extended".OrderID)
ON Products.ProductID = "Order Details Extended".ProductID)
ON Categories.CategoryID = Products.CategoryID
WHERE Orders.OrderDate BETWEEN '19970101' And '19971231'
GROUP BY Categories.CategoryID, Categories.CategoryName, Products.ProductName
--ORDER BY Products.ProductName

```

Dependencies

Table Name	Column Name
Orders	OrderID
Orders	OrderDate
Products	ProductID
Products	ProductName
Products	CategoryID
Order Details Extended	OrderID
Order Details Extended	ProductID
Order Details Extended	ExtendedPrice
Categories	CategoryID
Categories	CategoryName

14. Sales Totals by Amount

create view "Sales Totals by Amount" AS

```
SELECT "Order Subtotals".Subtotal AS SaleAmount, Orders.OrderID, Customers.CompanyName, Orders.ShippedDate
FROM Customers INNER JOIN
  (Orders INNER JOIN "Order Subtotals" ON Orders.OrderID = "Order Subtotals".OrderID)
ON Customers.CustomerID = Orders.CustomerID
WHERE ("Order Subtotals".Subtotal >2500) AND (Orders.ShippedDate BETWEEN '19970101' And '19971231')
```

Dependencies

Table Name	Column Name
Orders	OrderID
Orders	CustomerID
Orders	ShippedDate
Order Subtotals	OrderID
Order Subtotals	Subtotal
Customers	CustomerID
Customers	CompanyName

15. Summary of Sales by Quarter

create view "Summary of Sales by Quarter" AS

```
SELECT Orders.ShippedDate, Orders.OrderID, "Order Subtotals".Subtotal  
FROM Orders INNER JOIN "Order Subtotals" ON Orders.OrderID = "Order Subtotals".OrderID  
WHERE Orders.ShippedDate IS NOT NULL  
--ORDER BY Orders.ShippedDate
```

Dependencies

Table Name	Column Name
Orders	OrderID
Orders	ShippedDate
Order Subtotals	OrderID
Order Subtotals	Subtotal

16. Summary of Sales by Year

```
create view "Summary of Sales by Year" AS
SELECT Orders.ShippedDate, Orders.OrderID, "Order Subtotals".Subtotal
FROM Orders INNER JOIN "Order Subtotals" ON Orders.OrderID = "Order Subtotals".OrderID
WHERE Orders.ShippedDate IS NOT NULL
--ORDER BY Orders.ShippedDate
```

Dependencies

Table Name	Column Name
Orders	OrderID
Orders	ShippedDate
Order Subtotals	OrderID
Order Subtotals	Subtotal

Stored Procedures

1. Ten Most Expensive Products

```
create procedure "Ten Most Expensive Products" AS
SET ROWCOUNT 10
SELECT Products.ProductName AS TenMostExpensiveProducts, Products.UnitPrice
FROM Products
ORDER BY Products.UnitPrice DESC
```

Dependencies

Name	Type	Updated	Selected	Column
dbo.Products	user table			
dbo.Products	user table			

2. Employee Sales by Country

create procedure "Employee Sales by Country"

@Beginning_Date DateTime, @Ending_Date DateTime AS

SELECT Employees.Country, Employees.LastName, Employees.FirstName, Orders.ShippedDate, Orders.OrderID, "Order Subtotals".Subtotal AS SaleAmount

FROM Employees INNER JOIN

(Orders INNER JOIN "Order Subtotals" ON Orders.OrderID = "Order Subtotals".OrderID)

ON Employees.EmployeeID = Orders.EmployeeID

WHERE Orders.ShippedDate Between @Beginning_Date And @Ending_Date

Dependencies

Name	Type	Updated	Selected	Column
dbo.Orders	user table			
dbo.Orders	user table			
dbo.Orders	user table			
dbo.Employees	user table			
dbo.Employees	user table			
dbo.Employees	user table			
dbo.Employees	user table			
dbo.Order Subtotals	view			
dbo.Order Subtotals	view			

3. Sales by Year

create procedure "Sales by Year"

@Beginning_Date DateTime, @Ending_Date DateTime AS

SELECT Orders.ShippedDate, Orders.OrderID, "Order Subtotals".Subtotal, DATENAME(yy,ShippedDate) AS Year

FROM Orders INNER JOIN "Order Subtotals" ON Orders.OrderID = "Order Subtotals".OrderID

WHERE Orders.ShippedDate Between @Beginning_Date And @Ending_Date

Dependencies

Name	Type	Updated	Selected	Column
dbo.Orders	user table			
dbo.Orders	user table			
dbo.Order Subtotals	view			
dbo.Order Subtotals	view			

4. CustOrdersDetail

```

CREATE PROCEDURE CustOrdersDetail @OrderID int
AS
SELECT ProductName,
       UnitPrice=ROUND(Od.UnitPrice, 2),
       Quantity,
       Discount=CONVERT(int, Discount * 100),
       ExtendedPrice=ROUND(CONVERT(money, Quantity * (1 - Discount) * Od.UnitPrice), 2)
FROM Products P, [Order Details] Od
WHERE Od.ProductID = P.ProductID and Od.OrderID = @OrderID

```

Dependencies

Name	Type	Updated	Selected	Column
dbo.Products	user table			
dbo.Products	user table			
dbo.Order Details	user table			
dbo.Order Details	user table			
dbo.Order Details	user table			
dbo.Order Details	user table			
dbo.Order Details	user table			

5. CustOrdersOrders

```
CREATE PROCEDURE CustOrdersOrders @CustomerID nchar(5)
AS
SELECT OrderID,
OrderDate,
RequiredDate,
ShippedDate
FROM Orders
WHERE CustomerID = @CustomerID
ORDER BY OrderID
```

Dependencies

Name	Type	Updated	Selected	Column
dbo.Orders	user table			
dbo.Orders	user table			
dbo.Orders	user table			
dbo.Orders	user table			
dbo.Orders	user table			

6. CustOrderHist

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

Dependencies

Name	Type	Updated	Selected	Column
dbo.Orders	user table			
dbo.Orders	user table			
dbo.Products	user table			
dbo.Products	user table			
dbo.Order Details	user table			
dbo.Order Details	user table			
dbo.Order Details	user table			
dbo.Customers	user table			

7. SalesByCategory

```
CREATE PROCEDURE SalesByCategory
```

```
    @CategoryName nvarchar(15), @OrdYear nvarchar(4) = '1998'
```

```
AS
```

```
IF @OrdYear != '1996' AND @OrdYear != '1997' AND @OrdYear != '1998'
```

```
BEGIN
```

```
    SELECT @OrdYear = '1998'
```

```
END
```

```
SELECT ProductName,
```

```
    TotalPurchase=ROUND(SUM(CONVERT(decimal(14,2), OD.Quantity * (1-OD.Discount) * OD.UnitPrice)), 0)
```

```
FROM [Order Details] OD, Orders O, Products P, Categories C
```

```
WHERE OD.OrderID = O.OrderID
```

```
    AND OD.ProductID = P.ProductID
```

```
    AND P.CategoryID = C.CategoryID
```

```
    AND C.CategoryName = @CategoryName
```

```
    AND SUBSTRING(CONVERT(nvarchar(22), O.OrderDate, 111), 1, 4) = @OrdYear
```

```
GROUP BY ProductName
```

```
ORDER BY ProductName
```

Dependencies

Name	Type	Updated	Selected	Column
dbo.Orders	user table			
dbo.Orders	user table			
dbo.Products	user table			
dbo.Products	user table			
dbo.Products	user table			
dbo.Order Details	user table			
dbo.Order Details	user table			
dbo.Order Details	user table			
dbo.Order Details	user table			
dbo.Order Details	user table			
dbo.Order Details	user table			
dbo.Categories	user table			
dbo.Categories	user table			

8. sp_upgraddiagrams

```

CREATE PROCEDURE dbo.sp_upgraddiagrams
AS
BEGIN
IF OBJECT_ID(N'dbo.sysdiagrams') IS NOT NULL
return 0;

CREATE TABLE dbo.sysdiagrams
(
name sysname NOT NULL,
principal_id int NOT NULL, -- we may change it to varbinary(85)
diagram_id int PRIMARY KEY IDENTITY,
version int,

definition varbinary(max)
CONSTRAINT UK_principal_name UNIQUE
(
principal_id,
name
)
);

/* Add this if we need to have some form of extended properties for diagrams */
/*
IF OBJECT_ID(N'dbo.sysdiagram_properties') IS NULL
BEGIN
CREATE TABLE dbo.sysdiagram_properties
(
diagram_id int,
name sysname,
value varbinary(max) NOT NULL
)
END
*/

IF OBJECT_ID(N'dbo.dproperties') IS NOT NULL
begin
insert into dbo.sysdiagrams
(
[name],
[principal_id],
[version],
[definition]
)
select
convert(sysname, dgnm.[uvalue]),
DATABASE_PRINCIPAL_ID(N'dbo'), -- will change to the sid of sa
0, -- zero for old format, dgdef.[version],
dgdef.[lvalue]
from dbo.[dproperties] dgnm
inner join dbo.[dproperties] dggd on dggd.[property] = 'DtgSchemaGUID' and dggd.[objectid] = dgnm.[objectid]
inner join dbo.[dproperties] dgdef on dgdef.[property] = 'DtgSchemaDATA' and dgdef.[objectid] = dgnm.[objectid]

where dgnm.[property] = 'DtgSchemaNAME' and dggd.[uvalue] like N'_EA3E6268-D998-11CE-9454-00AA00A3F36E_'
return 2;
end
return 1;
END

```


9. sp_helpdiagrams

```

CREATE PROCEDURE dbo.sp_helpdiagrams
(
  @diagramname sysname = NULL,
  @owner_id int = NULL
)
WITH EXECUTE AS N'dbo'
AS
BEGIN
  DECLARE @user sysname
  DECLARE @dboLogin bit
  EXECUTE AS CALLER;
  SET @user = USER_NAME();
  SET @dboLogin = CONVERT(bit,IS_MEMBER('db_owner'));
  REVERT;
  SELECT
    [Database] = DB_NAME(),
    [Name] = name,
    [ID] = diagram_id,
    [Owner] = USER_NAME(principal_id),
    [OwnerID] = principal_id
  FROM
    sysdiagrams
  WHERE
    (@dboLogin = 1 OR USER_NAME(principal_id) = @user) AND
    (@diagramname IS NULL OR name = @diagramname) AND
    (@owner_id IS NULL OR principal_id = @owner_id)
  ORDER BY
    4, 5, 1
END

```

Dependencies

Name	Type	Updated	Selected	Column
dbo.sysdiagrams	user table			
dbo.sysdiagrams	user table			
dbo.sysdiagrams	user table			

10. sp_helpdiagramdefinition

```

CREATE PROCEDURE dbo.sp_helpdiagramdefinition
(
    @diagramname sysname,
    @owner_id int = null
)
WITH EXECUTE AS N'dbo'
AS
BEGIN
    set nocount on

    declare @theid int
    declare @IsDbo int
    declare @DiagId int
    declare @UIDFound int

    if(@diagramname is null)
    begin
        RAISERROR (N'E_INVALIDARG', 16, 1);
        return -1
    end

    execute as caller;
    select @theid = DATABASE_PRINCIPAL_ID();
    select @IsDbo = IS_MEMBER(N'db_owner');
    if(@owner_id is null)
        select @owner_id = @theid;
    revert;

    select @DiagId = diagram_id, @UIDFound = principal_id from dbo.sysdiagrams where principal_id = @owner_id and name = @diagramname;
    if(@DiagId IS NULL or (@IsDbo = 0 and @UIDFound <> @theid ))
    begin
        RAISERROR ('Diagram does not exist or you do not have permission.', 16, 1);
        return -3
    end

    select version, definition FROM dbo.sysdiagrams where diagram_id = @DiagId ;
    return 0
END

```

Dependencies

Name	Type	Updated	Selected	Column
dbo.sysdiagrams	user table			
dbo.sysdiagrams	user table			
dbo.sysdiagrams	user table			
dbo.sysdiagrams	user table			
dbo.sysdiagrams	user table			

11. sp_creatediagram

```

CREATE PROCEDURE dbo.sp_creatediagram
(
    @diagramname sysname,
    @owner_id int = null,
    @version int,
    @definition varbinary(max)
)
WITH EXECUTE AS 'dbo'
AS
BEGIN
    set nocount on

    declare @theid int
    declare @retval int
    declare @IsDbo int
    declare @userName sysname
    if(@version is null or @diagramname is null)
    begin
        RAISERROR (N'E_INVALIDARG', 16, 1);
        return -1
    end

    execute as caller;
    select @theid = DATABASE_PRINCIPAL_ID();
    select @IsDbo = IS_MEMBER(N'db_owner');
    revert;

    if @owner_id is null
    begin
        select @owner_id = @theid;
    end
    else
    begin
        if @theid <> @owner_id
        begin
            if @IsDbo = 0
            begin
                RAISERROR (N'E_INVALIDARG', 16, 1);
                return -1
            end
            select @theid = @owner_id
        end
    end
    -- next 2 line only for test, will be removed after define name unique
    if EXISTS(select diagram_id from dbo.sysdiagrams where principal_id = @theid and name = @diagramname)
    begin
        RAISERROR ('The name is already used.', 16, 1);
        return -2
    end

    insert into dbo.sysdiagrams(name, principal_id , version, definition)
        VALUES(@diagramname, @theid, @version, @definition) ;

    select @retval = @@IDENTITY
    return @retval
END

```

Dependencies

Name	Type	Updated	Selected	Column
dbo.sysdiagrams	user table			
dbo.sysdiagrams	user table			
dbo.sysdiagrams	user table			
dbo.sysdiagrams	user table			
dbo.sysdiagrams	user table			

12. sp_renamediagram

```

CREATE PROCEDURE dbo.sp_renamediagram
(
    @diagramname sysname,
    @owner_id int = null,
    @new_diagramname sysname
)
WITH EXECUTE AS 'dbo'
AS
BEGIN
    set nocount on
    declare @theid int
    declare @IsDbo int

    declare @UIDFound int
    declare @DiagId int
    declare @DiagIdTarg int
    declare @u_name sysname
    if((@diagramname is null) or (@new_diagramname is null))
    begin
        RAISERROR ('Invalid value', 16, 1);
        return -1
    end

    EXECUTE AS CALLER;
    select @theid = DATABASE_PRINCIPAL_ID();
    select @IsDbo = IS_MEMBER(N'db_owner');
    if(@owner_id is null)
        select @owner_id = @theid;
    REVERT;

    select @u_name = USER_NAME(@owner_id)

    select @DiagId = diagram_id, @UIDFound = principal_id from dbo.sysdiagrams where principal_id = @owner_id and name = @diagramname
    if(@DiagId IS NULL or (@IsDbo = 0 and @UIDFound <> @theid))
    begin
        RAISERROR ('Diagram does not exist or you do not have permission.', 16, 1)
        return -3
    end

    -- if((@u_name is not null) and (@new_diagramname = @diagramname)) -- nothing will change
    -- return 0;

    if(@u_name is null)
        select @DiagIdTarg = diagram_id from dbo.sysdiagrams where principal_id = @theid and name = @new_diagramname
    else
        select @DiagIdTarg = diagram_id from dbo.sysdiagrams where principal_id = @owner_id and name = @new_diagramname

    if((@DiagIdTarg is not null) and @DiagId <> @DiagIdTarg)
    begin
        RAISERROR ('The name is already used.', 16, 1);
        return -2
    end

    if(@u_name is null)
        update dbo.sysdiagrams set [name] = @new_diagramname, principal_id = @theid where diagram_id = @DiagId
    else
        update dbo.sysdiagrams set [name] = @new_diagramname where diagram_id = @DiagId

```

return 0
END

Dependencies

Name	Type	Updated	Selected	Column
dbo.sysdiagrams	user table			
dbo.sysdiagrams	user table			
dbo.sysdiagrams	user table			

13. sp_alterdiagram

```

CREATE PROCEDURE dbo.sp_alterdiagram
(
    @diagramname sysname,
    @owner_id int = null,
    @version int,
    @definition varbinary(max)
)
WITH EXECUTE AS 'dbo'
AS
BEGIN
    set nocount on

    declare @theid int
    declare @retval int
    declare @IsDbo int

    declare @UIDFound int
    declare @DiagId int
    declare @ShouldChangeUID int

    if(@diagramname is null)
    begin
        RAISERROR ('Invalid ARG', 16, 1)
        return -1
    end

    execute as caller;
    select @theid = DATABASE_PRINCIPAL_ID();
    select @IsDbo = IS_MEMBER(N'db_owner');
    if(@owner_id is null)
        select @owner_id = @theid;
    revert;

    select @ShouldChangeUID = 0
    select @DiagId = diagram_id, @UIDFound = principal_id from dbo.sysdiagrams where principal_id = @owner_id and name = @diagramname

    if(@DiagId IS NULL or (@IsDbo = 0 and @theid <> @UIDFound))
    begin
        RAISERROR ('Diagram does not exist or you do not have permission.', 16, 1);
        return -3
    end

    if(@IsDbo <> 0)
    begin
        if(@UIDFound is null or USER_NAME(@UIDFound) is null) -- invalid principal_id
        begin
            select @ShouldChangeUID = 1 ;
        end
    end

    -- update dds data
    update dbo.sysdiagrams set definition = @definition where diagram_id = @DiagId ;

    -- change owner
    if(@ShouldChangeUID = 1)
        update dbo.sysdiagrams set principal_id = @theid where diagram_id = @DiagId ;

    -- update dds version

```



```
if(@version is not null)
  update dbo.sysdiagrams set version = @version where diagram_id = @DiagId ;

return 0
END
```

Dependencies

Name	Type	Updated	Selected	Column
dbo.sysdiagrams	user table			
dbo.sysdiagrams	user table			
dbo.sysdiagrams	user table			
dbo.sysdiagrams	user table			
dbo.sysdiagrams	user table			

14. sp_dropdiagram

```

CREATE PROCEDURE dbo.sp_dropdiagram
(
    @diagramname sysname,
    @owner_id int = null
)
WITH EXECUTE AS 'dbo'
AS
BEGIN
    set nocount on
    declare @theid int
    declare @IsDbo int

    declare @UIDFound int
    declare @DiagId int

    if(@diagramname is null)
    begin
        RAISERROR ('Invalid value', 16, 1);
        return -1
    end

    EXECUTE AS CALLER;
    select @theid = DATABASE_PRINCIPAL_ID();
    select @IsDbo = IS_MEMBER(N'db_owner');
    if(@owner_id is null)
        select @owner_id = @theid;
    REVERT;

    select @DiagId = diagram_id, @UIDFound = principal_id from dbo.sysdiagrams where principal_id = @owner_id and name = @diagramname
    if(@DiagId IS NULL or (@IsDbo = 0 and @UIDFound <> @theid))
    begin
        RAISERROR ('Diagram does not exist or you do not have permission.', 16, 1)
        return -3
    end

    delete from dbo.sysdiagrams where diagram_id = @DiagId;

    return 0;
END

```

Dependencies

Name	Type	Updated	Selected	Column
dbo.sysdiagrams	user table			
dbo.sysdiagrams	user table			
dbo.sysdiagrams	user table			
dbo.sysdiagrams	user table			