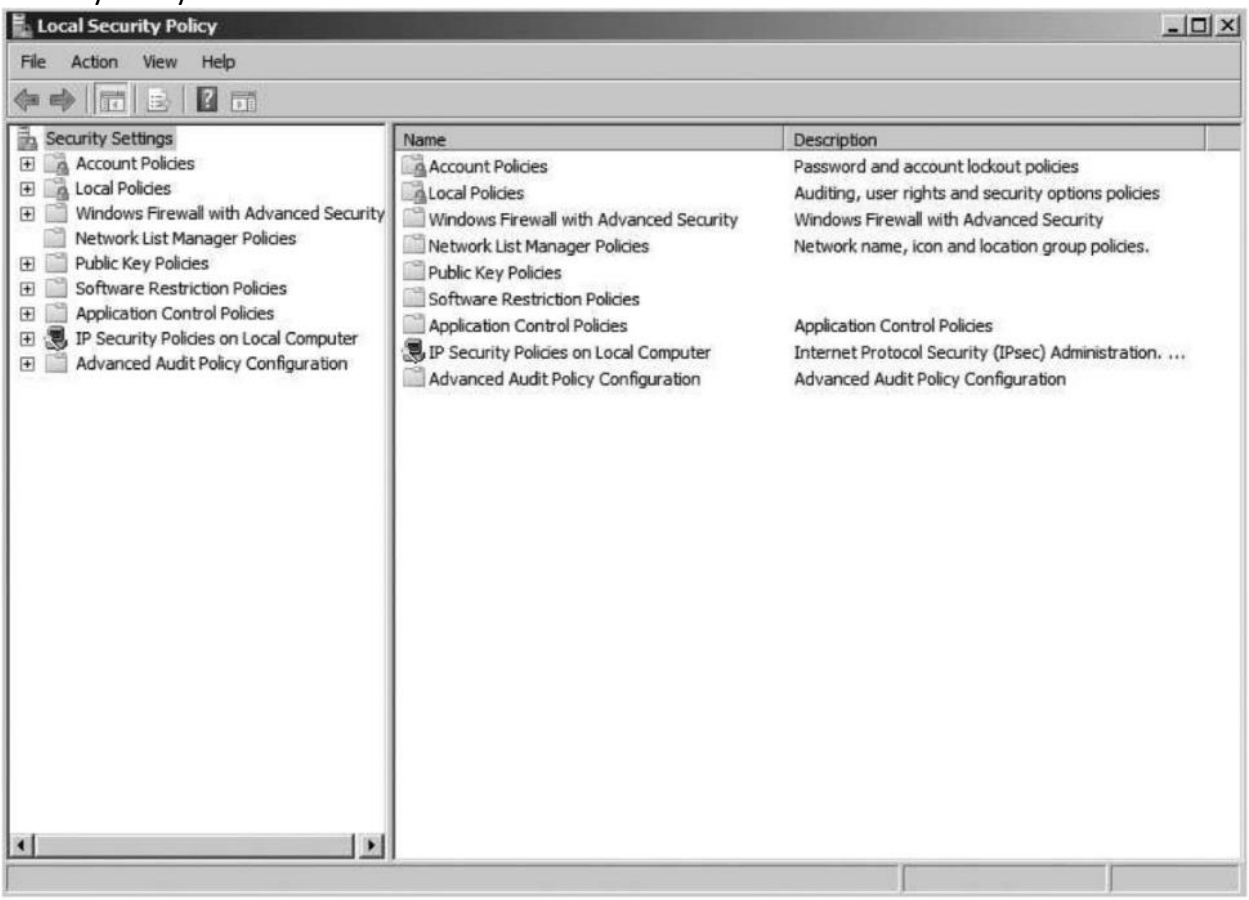
Managing SQL Server Logins – Part 1

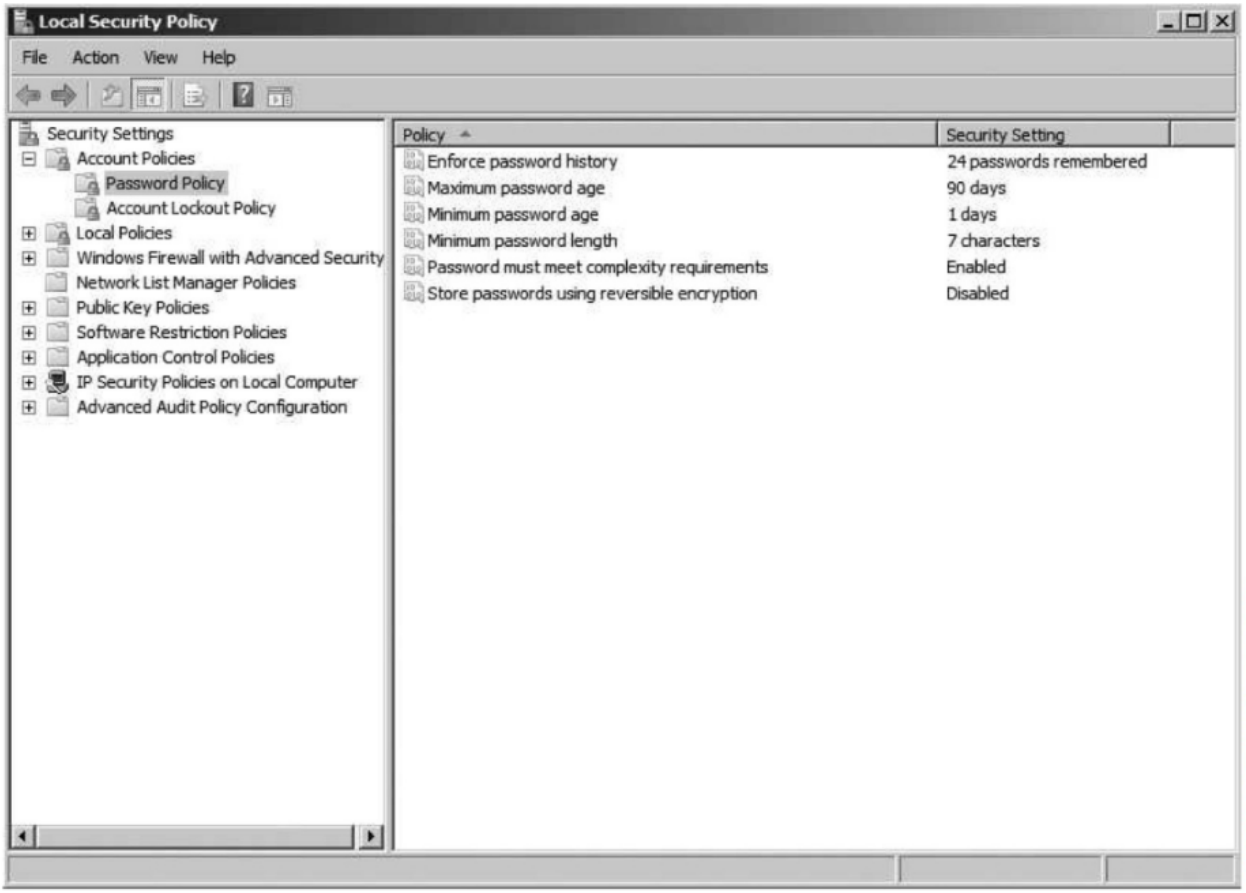
you will change the different password policies for a Windows account and a SQL Server login. You will create new Windows accounts and groups on the local operating system of your computer, and you will map these to SQL Server logins. Using SQL Server Management Studio, you will also discover how to enable mixed mode authentication and set up a SQL Server login that uses SQL Server authentication.

1. Click the Start button and click Run. In the Run dialog box, type secpol.msc in the Open text box, and then click OK to launch the Microsoft Management Console (MMC) Local Security Policy window.



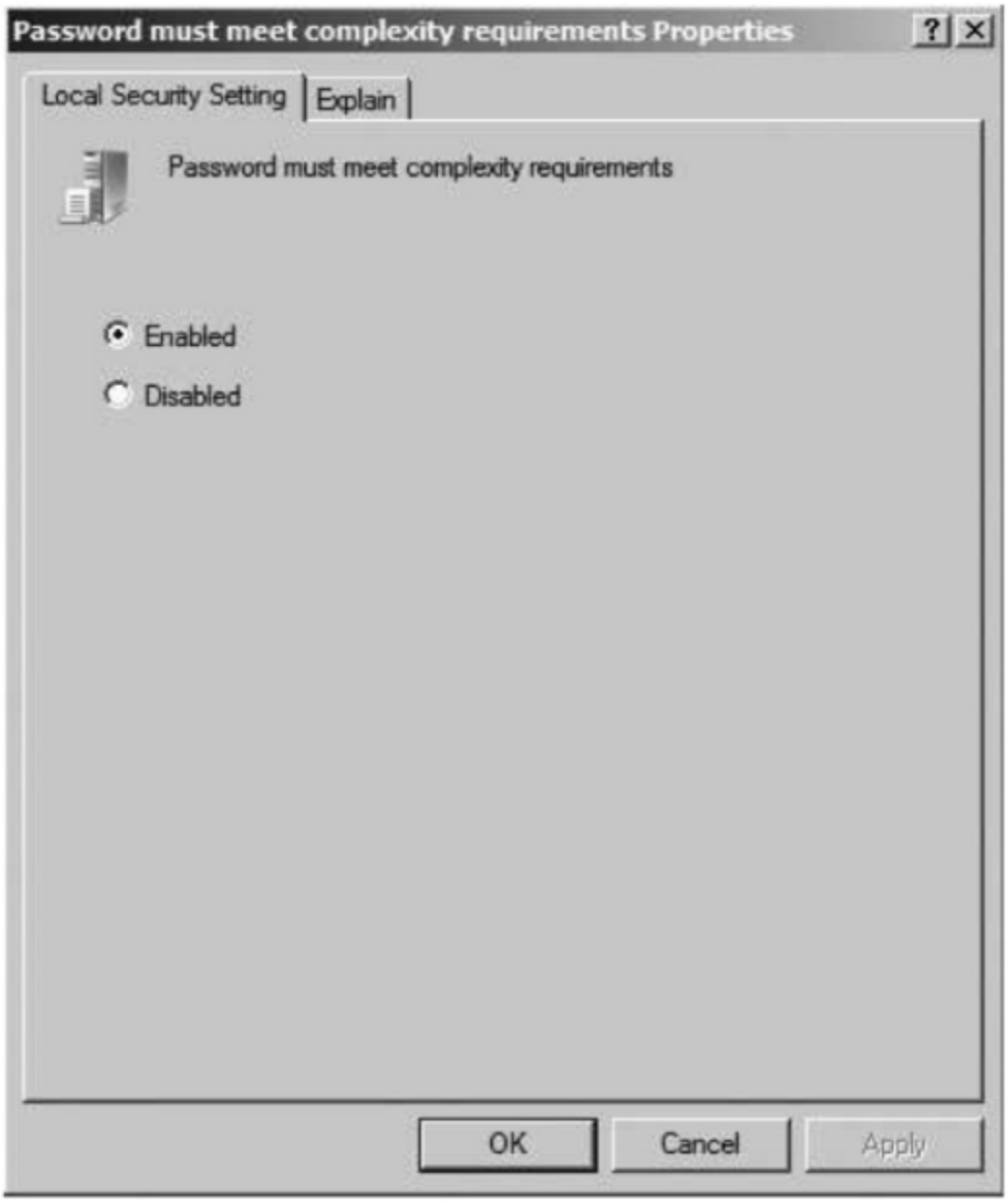
2. In the left navigation pane of the Local Security Policy window, click the + symbol to expand the Account Policies folder. Then, click the Password Policy folder to view the current password policies and associated security settings in the right pane. These policies control the settings for Local Users that are configured on the server. The same

policies can also be applied for domain users on a domain controller.



3. In the right pane, right-click Password must meet complexity requirements, and then click Properties. On the Local Security Setting tab of the Password must meet com-

plexity requirements Properties window, click Enabled.



4. Click the Explain tab, and read the description of the policy. Click OK to save the settings and exit the window. Close the MMC Local Security Policy window.
5. Click the Start button, point to Administrative Tools, and then click Server Manager to launch Server Manager. In the left navigation pane, click the + symbol to expand

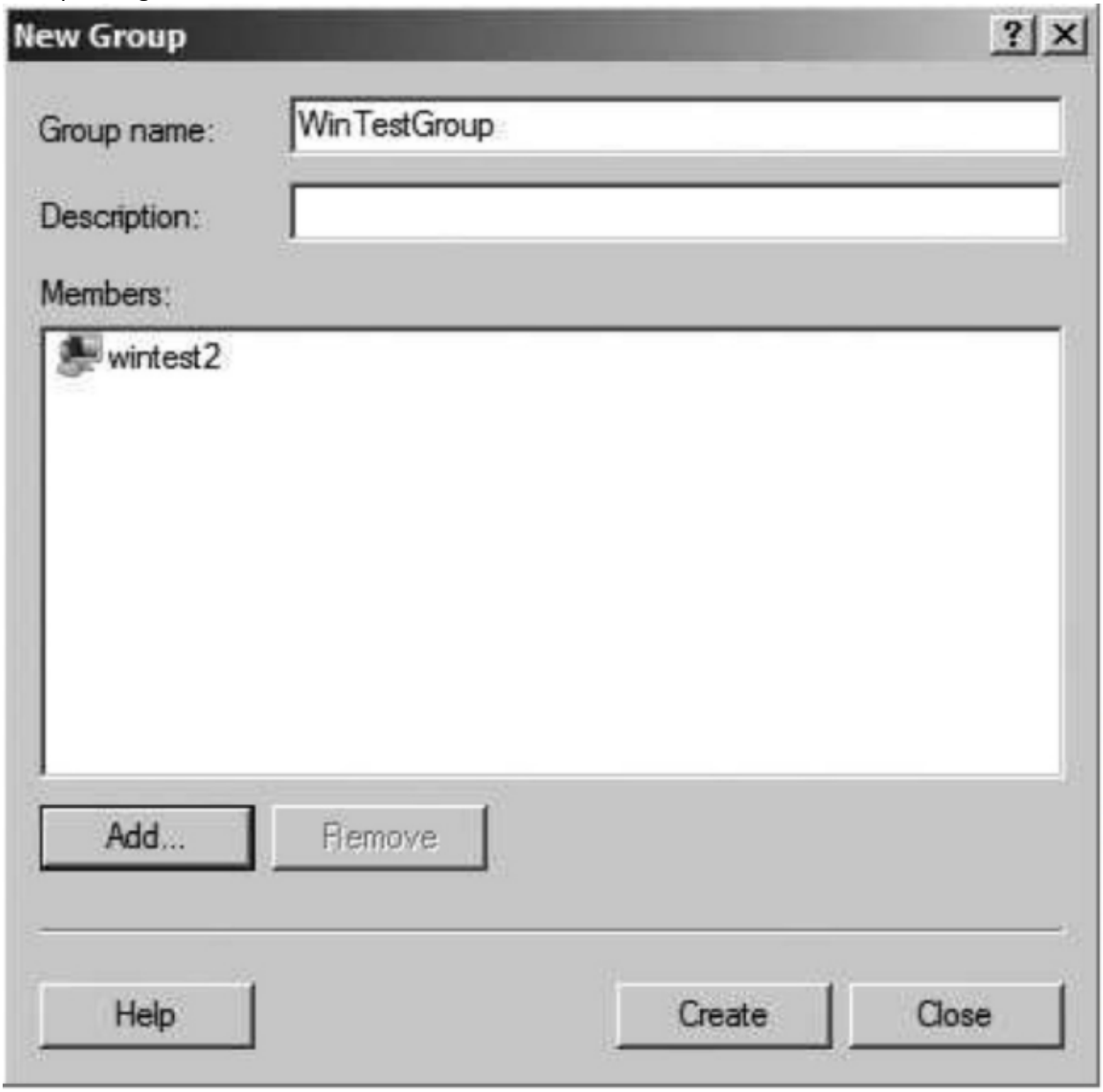
Configuration, expand Local Users and Groups, and then click Users.

6. On the menu bar, click Action, and then click New User. In the New User dialog box, type wintest1 in the User name text box, and then type Windows Test One in the Full name text box. Type Compl3$xity in the Password and Confirm password text boxes. Uncheck the User must change password at next logon check box because you will only

be using this account for testing.

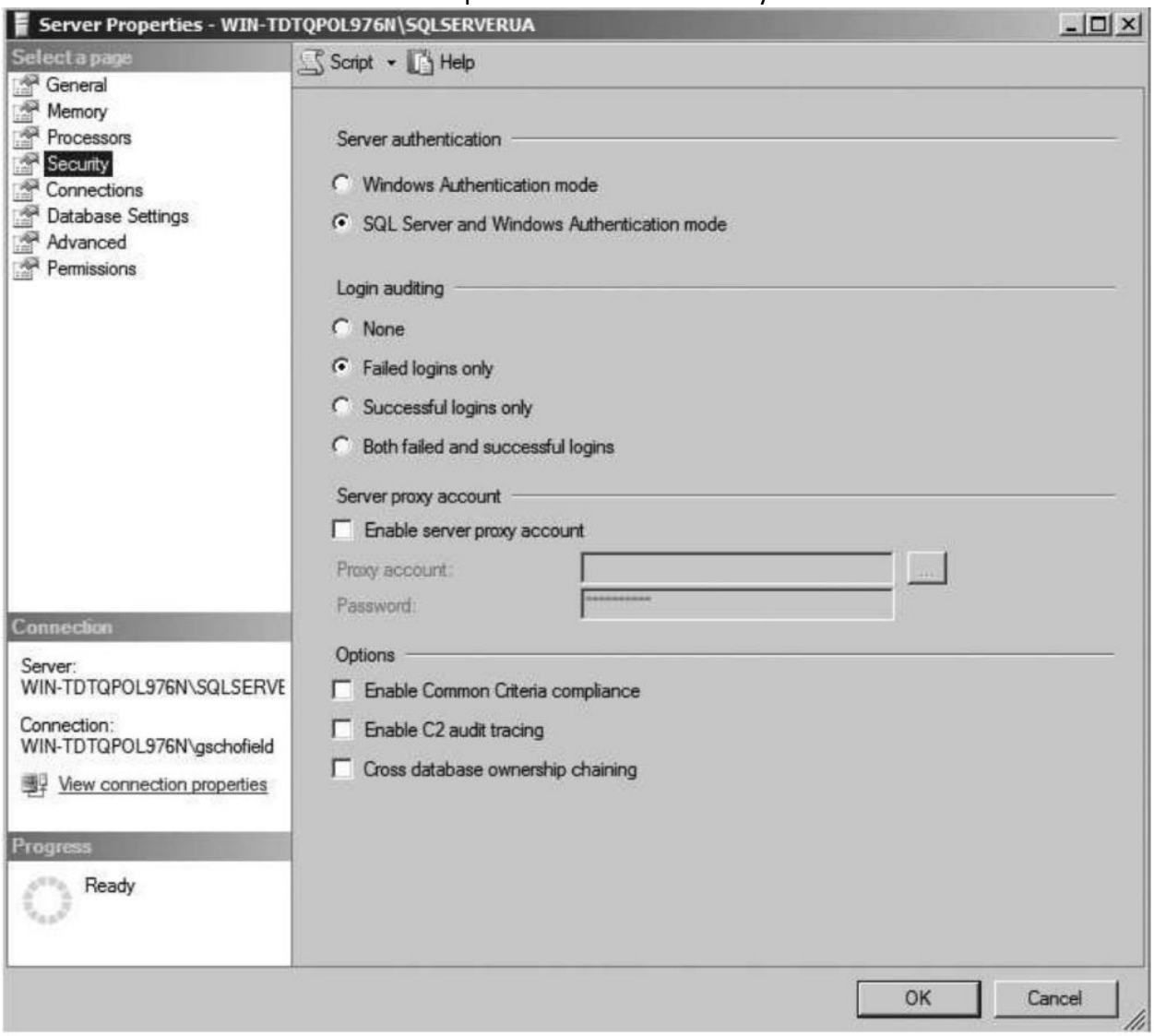
7. Click Create to create the new Windows user. A blank New User dialog box opens. Following the same steps and settings as for the wintest1 user, create a second test user called wintest2. Set the full name to Windows Test Two, and use Comple3$xity for the password. Click Close after creating the second user.
8. In the left navigation of Server Manager, click Groups below Local Users and Groups. On the menu bar, click Action, and then click New Group. In the New Group dialog box, type WinTestGroup in the Group name text box, and then click the Add button. Type wintest2 in the Select Users dialog box, and then click Check names. The name should resolve automatically to <Computer Name>\wintest2. Click OK to return to the New

Group dialog box.



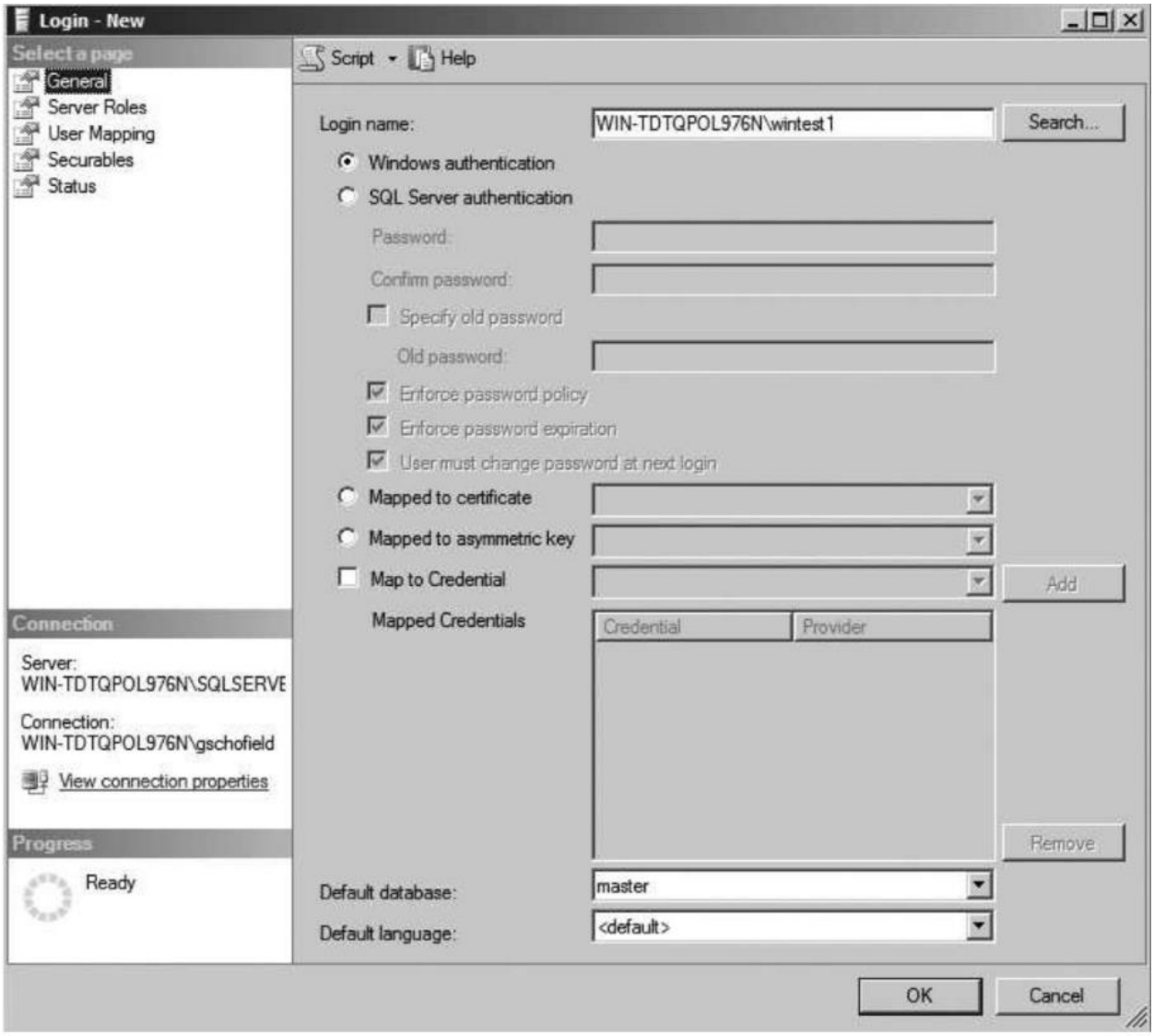
9. In the New Group dialog box, click Create to create the new WinTestGroup group with member wintest2. Click Close to exit the New Group dialog box, and then close Server Manager.
10. Click the Start button, point to All Programs, click Microsoft SQL Server 2012, and then click SQL Server Management Studio. In the Connect to Server dialog box, select Database Engine as the server type, type LOCALHOST\SQLSERVER in the Server name text box, and then select Windows Authentication from the Authentication list box. Click Connect.
11. In SQL Server Management Studio, right-click SSVR in the left navigation pane of Object Explorer, and then click Properties. Click Security in the left navigation menu of the Server Properties window. Review the Server authentication settings, which you set in

Windows Authentication mode. Click the “SQL Server and Windows Authentication mode” Click OK to close the Server Properties window. Restart your Windows Server.



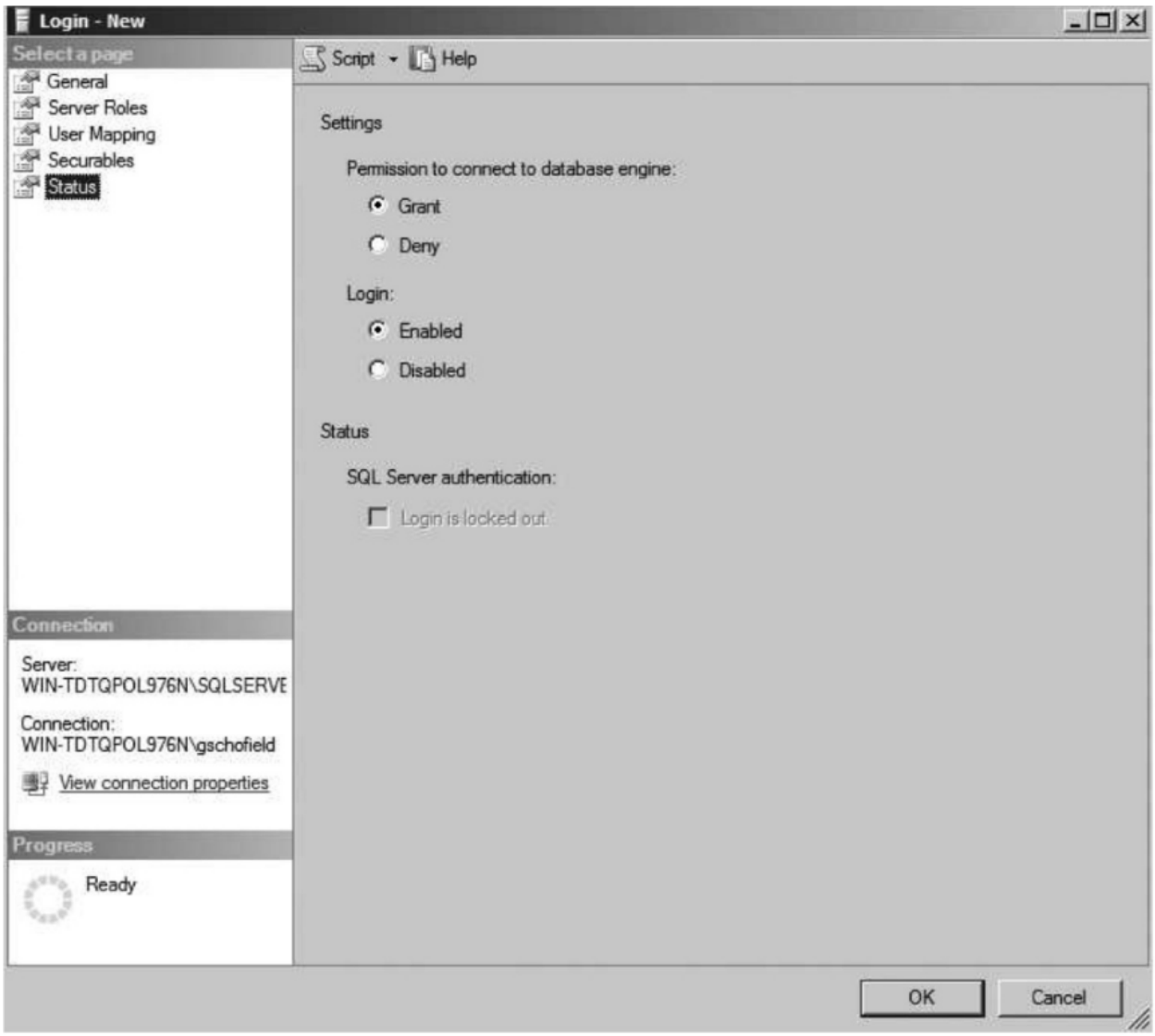
12. Start SQL Server Management Studio, click the + symbol to expand the SSVR instance in the Object Explorer left navigation pane. Click the + symbol to expand the Security folder, and then expand the Logins folder. Right-click the Logins folder, and click New Login.

13. On the General page of the Login – New window, keep the default Windows authentication option checked, and click the Search button to the right of the Login name text box. In the Select User or Group dialog box, type wintest1 in the Enter the object name to select text box, and then click Check Names to resolve this name. Click OK.



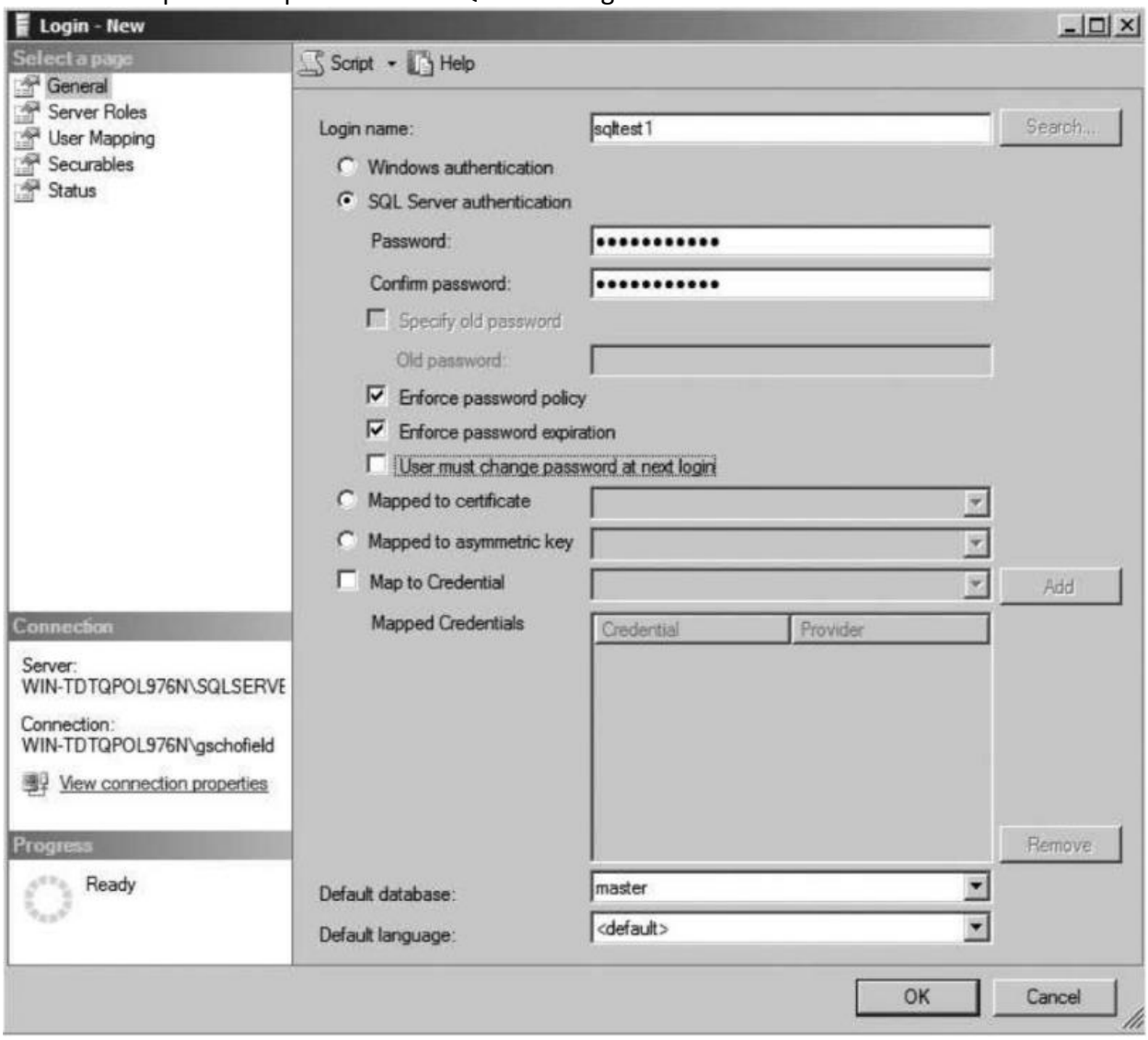
14. In the left navigation pane of the Login – New window, click Server Roles. This page allows you to assign server-level roles. Check that only the public role is checked for the wintest1 login. In the left navigation pane, click Securables. This page allows you to view the individual server-level permissions of the login.
15. In the left navigation pane, click Status. Check that Permission to connect to the database engine is set to Grant and the Login status is Enabled. Click OK to create a login

for the Windows account wintest1.

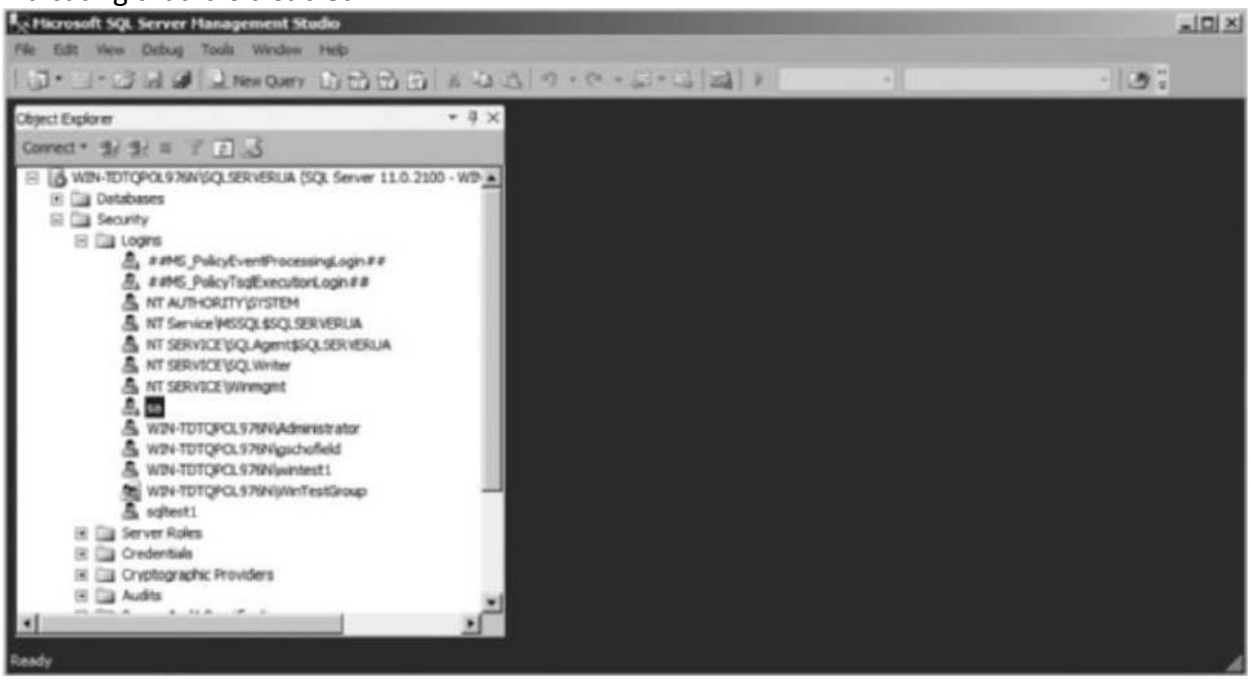


16. Right-click the Logins folder in SQL Server Management Studio Object Explorer, and then click New Login. On the General page of the Login – New window, keep the default Windows authentication option checked, and click the Search button to the right of the Login name text box.
17. In the Select User or Group dialog box, click the Object Types button, and then check Groups in the Object types dialog box. Click OK to return to the Select User or Group dialog box. Type WinTestGroup in the Enter the object name to select text box, and then click Check Names to resolve this name. Click OK to create a login for the Windows group WinTestGroup. Any Windows account that is a member of the WinTestGroup Windows group will now be able to log in to the SQL Server instance.
18. Right-click the Logins folder in SQL Server Management Studio Object Explorer and then click New Login. On the General page of the Login – New window, type sqltest1 in the Login name text box. Click the SQL Server authentication option button to select it, and then type Comple3$xity in the Password and Confirm password text boxes. Uncheck the

User must change password at next login check box as you will only be using this user ID for testing. Keep the other two password policies enabled. Note the difference between the available password policies for a SQL Server login versus Windows.



19. Click OK to create the SQL Server login sqltest1. You should now see the new logins that you created appear in the Logins folder of Object Explorer. Note that the icon for the group login that you created in previous step, WinTestGroup, is different from the user logins. Notice also that the sa login has a small red downward pointing arrow on its icon, indicating that it is disabled.



20. Right-click the user sa, and then click Properties. On the General tab of the Login Properties – sa window, you can add a new password. Click the Status page in the left navigation menu. Notice that the login is disabled. To enable a login, you would need to click the Login Enabled option button. Due to the security concerns associated with using the sa default system Administrator account, you will leave it disabled. Click Cancel to exit the window without saving any changes.

Implementing a New Database

You have been asked to implement a database that will track products, product categories, suppliers, and current product inventory for a cookware supply store. You have been given a database specification that includes table requirements and sample data for the test environment. The design calls for four tables to be created: Inventory, Product, ProductCategory, and Supplier. Tables 5-5 through 5-10 contain the column names, data types, constraints, and keys that must be implemented for each table. Tables 5-11 through 5-14 contain the sample data that you will need to insert into each table for testing.

Table 5-5 Column definitions for the Inventory table

Key	Name	Data type	Allow nulls	Default
PK	InventoryID	int	N	
FK	ProductID	int	N	
	UnitsOnHand	int	N	
	ModifiedOn	datetime	N	(getdate())

Table 5-6 Foreign key definitions for the Inventory table

Name	Columns
FK_Inventory_Product	ProductID->dbo.Product.ProductID

Table 5-7 Column definitions for the Product table

Key	Name	Data type	Allow nulls	Default
PK	ProductID	int	N	
FK	ProductCategoryID	int	N	
FK	SupplierID	int	N	
	Description	nvarchar(50)	N	
	Color	nvarchar(20)	N	
	UnitPrice	money	N	
	ModifiedOn	datetime	N	(getdate())

Table 5-8 Foreign key definitions for the Product table

Name	Columns
FK_Product_ProductCategory	ProductCategoryID->dbo.ProductCategory.ProductCategoryID
FK_Product_Supplier	SupplierID->dbo.Supplier.SupplierID

Table 5-9 Column definitions for the ProductCategory table

Key	Name	Data type	Allow nulls	Default
PK	ProductCategoryID	int	N	
	Description	nvarchar(50)	N	
	ModifiedOn	datetime	N	(getdate())

Table 5-10 Column definitions for the Supplier table

Key	Name	Data type	Allow nulls	Default
PK	SupplierID	int	N	
	SupplierName	nvarchar(50)	N	
	City	nvarchar(50)	N	
	Country	nvarchar(50)	N	
	ContactPhone	nvarchar(20)	N	
	ModifiedOn	datetime	N	(getdate())

Table 5-11 Sample data for the Inventory table

InventoryID	ProductID	UnitsOnHand	ModifiedOn
1	1	40	7/3/12 8:56
2	2	24	7/3/12 8:57
3	4	5	7/3/12 8:57
4	5	2	7/3/12 8:57
5	6	7	7/3/12 8:57
6	7	16	7/3/12 8:57
7	8	12	7/3/12 8:57
8	10	27	7/3/12 8:58

Table 5-12 Sample data for the Product table

ProductID	ProductCategoryID	SupplierID	Description	Color	UnitPrice	ModifiedOn
1	2	3	Stainless steel flatware	Silver	30.99	7/3/12 8:49
2	2	4	Serving spoons	Nickel	18.5	7/3/12 8:50
3	5	1	Chef's knife	Silver	25	7/3/12 8:50
4	5	1	Cutting board	Black	15.77	7/3/12 8:50
5	5	2	Sharpening steel	Carbon	12.24	7/3/12 8:51
6	3	2	Napkin set	Red	9.31	7/3/12 8:52
7	3	2	Table cloth	Various	21.89	7/3/12 8:52
8	1	3	Large frying pan	Black	13.5	7/3/12 8:53
9	1	3	Small frying pan	Black	7.25	7/3/12 8:53
10	1	3	Nonstick saucepan	Silver	16	7/3/12 8:54

Table 5-13 Sample data for the ProductCategory table

ProductCategoryID	Description	ModifiedOn
1	Cookware	7/3/12 8:37
2	Cutlery	7/3/12 8:38
3	Linens	7/3/12 8:38
4	Tableware	7/3/12 8:38
5	Cooks Tools	7/3/12 8:38

Table 5-14 Sample data for the Supplier table

SupplierID	SupplierName	City	Country	ContactPhone	ModifiedOn
1	Sanzone	Oakland	USA	(510) 555-7200	7/3/12 8:44
2	Itex	Frankfurt	Germany	+49 (0) 8731-9140	7/3/12 8:45
3	Newnix	New York	USA	(212) 555-8100	7/3/12 8:46
4	Zenice	Chicago	USA	(707) 555-1400	7/3/12 8:47
5	Waredom	Paris	France	+33 (0) 123-4496	7/3/12 8:47

1. Using the CREATE DATABASE command, create a new test database on your SQLSERVER instance that uses the default schema. The database should be named ProductInformation.

2. Create and execute a SQL query to create the four individual tables Product, ProductCategory, Supplier, and Inventory with the null and default constraints.
3. Using the ALTER TABLE syntax, add a primary key and a foreign key constraint to each table.
4. Use the INSERT statement to add the sample rows to the tables. Use the column default for the ModifiedOn column.
5. Retrieve all Product Descriptions that have no associated inventory. Save this screen shot to upload in Etudes.
6. Create a SELECT query that lists the Product Descriptions and the total inventory on hand (expressed as UnitPrice * UnitsOnHand). Use a column alias to assign any calculated columns a meaningful name in the result set. Save this screen shot to upload in Etudes.
7. Create a SELECT statement that returns the top two products with the most inventory units on hand. Save this screen shot to upload in Etudes.
8. Create a SELECT statement that lists the total inventory on hand (expressed as UnitPrice * UnitsOnHand) for each ProductCategory. Only return records where the total inventory on hand for a product category is greater than $150. Order the output by total inventory on hand in descending order. Use a column alias to assign any calculated columns a meaningful name in the result set. Save this screen shot to upload in Etudes.

Managing SQL Server Logins – Part 2

1. Create a new Windows user account called WinHoa1.
2. Create a new SQL Server login that uses Windows authentication and is mapped to the Windows user WinHoa1. Ensure that the SQL Server login is enabled and granted permission to connect to the default SQL Server instance.
3. Map the new SQL Server login to a new database user on the ProductInformation database. Grant the database user the db_datareader role.
4. Test the new SQL Server login by logging on to your computer using the new Windows user account, WinHoa1 that you created in Step 1, and use SQL Server Management Studio to connect to the default SQL Server instance using Windows integrated authentication. In Object Explorer, confirm that you are able to browse the folders within the ProductInformation database but not the AdventureWorks database. Save this screen shot to upload in Etudes.
5. You will now test that the database user has read access to all tables in the ProductInformation database. In SQL Server Management Studio logged on as the WinHoa1 user, open a new Query Editor window. Write a SELECT statement to query all rows in the Product and Inventory tables. Save this screen shot to upload in Etudes.
6. You will now test that the database user does not have INSERT permission to add records to the tables in the ProductInformation database. Write a SELECT statement to INSERT a new record in the ProductCategory table. Document this step by taking a

screen shot of the Query Editor window after attempting to execute the query. Save this screen shot to upload in Etudes.

7. You will now test that the database user does not have permission to delete records from the tables in the ProductInformation database. Write a DELETE statement to delete a record from the Supplier table. Save this screen shot to upload in Etudes.
8. Close SQL Server Management Studio and log off of your computer. Log back on using your usual administrator account and disable the WinHoa1 SQL Server login and the associated WinHoa1 Windows account.