
MODULE TITLE: **Monitor and Administer Database**

MODULE CONTENTS:

LO1. Start up a database

1.1 Configuring system for ***database*** start-up

1.2 Monitoring database start-up and operation for irregularities

- **Configuring system for database start-up**

When you install SQL Server, Setup writes a set of default startup options in the Microsoft Windows registry.
If the Database Engine cannot locate the necessary files, SQL Server will not start.
Startup options can be set by using SQL Server Configuration Manager.

Connect to SQL Server When System Administrators Are Locked Out

This topic describes how you can regain access to the SQL Server Database Engine as a system administrator.
A system administrator can lose access to an instance of SQL Server because of one of the following reasons:

- All logins that are members of the sysadmin fixed server role have been removed by mistake.
- All Windows Groups that are members of the sysadmin fixed server role have been removed by mistake.
- The logins that are members of the sysadmin fixed server role are for individuals who have left the company or who are not available.
- The sa account is disabled or no one knows the password.

One way in which you can regain access is to reinstall SQL Server and attach all the databases to the new instance. This solution is time-consuming; and, to recover the logins, it might require restoring the master database from a backup.

- **Monitoring database start-up and operation for irregularities**

Microsoft SQL Server and the Microsoft Windows operating system provide utilities that let you view the current condition of the database and to track performance as conditions change.

This topic describes how to open the Activity Monitor to obtain information about SQL Server processes and how these processes affect the current instance of SQL Server. It also describes how to set the refresh interval of the Activity Monitor.

Activity Monitor runs queries on the monitored instance to obtain information for the Activity Monitor display panes. When the refresh interval is set to less than 10 seconds, the time that is used to run these queries can affect server performance.

Date: ____/____/10

To view the Activity Monitor, a user must have VIEW SERVER STATE permission. To view the Data File I/O section of Activity Monitor, you must have CREATE DATABASE, ALTER ANY DATABASE, or VIEW ANY DEFINITION permission in addition to VIEW SERVER STATE.
To open Activity Monitor in SQL Server Management Studio

- On the SQL Server Management Studio standard toolbar, click Activity Monitor.
- In the Connect to Server dialog box, select the server name and authentication mode, and then click Connect.

To open Activity Monitor in Object Explorer

- In Object Explorer, right-click the instance name, and then select Activity Monitor.

To open Activity Monitor when opening SQL Server Management Studio

- ✓ On the Tools menu, click Options.
- ✓ In the Options dialog box, expand Environment, and then select General.
- ✓ In the At startup box, select Open Object Explorer and Activity Monitor.
- ✓ To activate the changes, close and reopen SQL Server Management Studio.

Instruction Sheet-2	**Learning Guide #2**

This learning guide is developed to provide you the necessary information regarding the following content coverage and topics.

- compiling data dictionary and data structure
- Maintaining data integrity constraints according to ***business requirements***
- Creating and designing indexes and multiple-field keys according to business requirements
- Monitoring the lock options chosen for the database
- Monitoring the data storage space
- Updating data according to ***organizational guidelines***

This guide will also assist the trainee to attain the learning outcome stated in the cover page. Specifically, upon completion of this Learning Guide, the trainee will be able to:

- › compile data dictionary and data structure
- › Maintain data integrity constraints according to ***business requirements***
- › Create and design indexes and multiple-field keys according to business requirements
- › Monitor the lock options chosen for the database
- › Confirm stored recent back-ups of the database and retrieved data
- › Monitor the data storage space
- › Update data according to ***organizational guidelines***

Learning Instructions:

1. Read the specific objectives of this Learning Guide.

Date: ____/____/10

2. Read the information written in the "Information Sheets 3". Try to understand what are being discussed. Ask your teacher for assistance if you have hard time understanding them.
3. Accomplish the "Self-check 3".
4. Ask from your teacher the key to correction (key answers) or you can request your teacher to correct your work. (You are to get the key answer only after you finished answering the Self-check 3).
5. If you earned a satisfactory evaluation proceed to "Information Sheet unit of competency". However, if your rating is not satisfactory, see your teacher for further instructions,
6. Submit your accomplished Self-check. This will form part of your training portfolio.

LO2. <u>Manage database</u>

- **<u>compiling data dictionary and data structure</u>**

A **data dictionary** is a collection of descriptions of the data objects or items in a data model for the benefit of programmers and others who need to refer to them.
I.e.: It is a set of information describing the contents, format, and structure of a database and the relationship between its elements, used to control access to and manipulation of the database.

When developing programs that use the data model, a data dictionary can be consulted to understand where a data item fits in the structure, what values it may contain, and basically what the data item means in real-world terms.

* Most DBMS keep the data dictionary hidden from users to prevent them from accidentally destroying its contents.

* A data dictionary may contains:
  - The definitions of all schema objects in the database.

- How much space has been allocated for, and is currently used by the schema objects
- Default values for columns.
- Integrity constraint information (Constraints that apply to each field, if any)
- Auditing information, such as who has accessed or updated various schema objects
- Privileges and roles each user has been granted (Access Authorization)
- Description of database users, their responsibilities and their access rights.

Data dictionaries do not contain any **actual value** from the database, only bookkeeping information for managing it.

What is an advantage of a Data Dictionary?
When a new user is introduced to the system or a new administrator takes over the system, identifying table structures and types becomes simpler.

➢ **Maintaining data integrity constraints according to business requirements**

Data integrity is a constraint which used to ensure accuracy and consistency of data in a database by validating the data before getting stored in the columns of the table.

Data integrity refers to the overall completeness, accuracy and consistency of data in according to business requirements.

- **Types of integrity constraints**
  - › Entity integrity
  - › Referential integrity
  - › Domain integrity
  - › User defined integrity

- **Entity integrity**

This is concerned with the concept of primary keys. The rule states that every table must have its own primary key and that each has to be unique and not null.

- **Referential Integrity**

This is the concept of foreign keys. The rule states that the foreign key value can be in two states. The first state is that the foreign key value would refer to a primary key value of another table, or it can be null. Being null could simply mean that there are no relationships, or that the relationship is unknown.
Referential integrity is a feature provided by relational DBMS that prevents users from entering inconsistent data.

- **Domain Integrity**

This states that all columns in a relational database are in a defined domain.

The concept of data integrity ensures that all data in a database can be traced and connected to other data. This ensures that everything is recoverable and searchable. Having a single, well defined and well controlled data integrity system increases stability, performance, reusability and maintainability.

- **User Defined Integrity**

User-defined integrity allows you to define specific business rules that do not fall into one of the other integrity categories. All of the integrity categories support user-defined integrity (all column- and table-level constraints in CREATE TABLE, stored procedures, and triggers).
Business rules may dictate/state that when a specific action occurs further actions should be triggered.
For example, deletion of a record automatically writes that record to an audit table.

➢ **Creating and designing indexes and multiple-field keys according to business requirements**

- **What is index?**

An **index** is a separate physical data structure that enables queries to access one or more data rows fast.

A **database index** is a separate physical data structure that improves the speed of data retrieval operations on a database table at the cost of additional writes and the use of more storage space to maintain the extra copy of data. Indexes are used to quickly locate data without having to search every row in a database table every time a database table is accessed. Indexes can be created using one or more columns of a database table, providing the basis for both rapid random lookups and efficient access of ordered records.
Why Use Indexes? Two primary reasons exist for creating indexes in SQL Server:

- To maintain uniqueness of the indexed column(s)
- To provide fast access to the data in tables.

- **Deciding which fields to be index**

The following list gives guidelines in choosing columns to index:

- You should create indexes on columns that are used frequently in WHERE clauses.
- You should create indexes on columns that are used frequently to join tables.
- You should create indexes on columns that are used frequently in ORDER BY clauses.
- You should create indexes on columns that have few of the same values or unique values in the table.
- You should not create indexes on small tables (tables that use only a few blocks) because a full table scan may be faster than an indexed query.
- If possible, choose a primary key that orders the rows in the most appropriate order.

- **Creating indexes**

Indexes can be created to order the values in a column in ascending or descending sequence.

- You can use the CREATE INDEX statement to create indexes.

The general form of CREATE INDEX statement is:

```
CREATE INDEX index_name ON table_name (column1 [ASC | DESC] ,...)
```

Example: Create an index for the EmpID column of the employee table.

- **Delete an index**

Deleting an index means removing one or more relational indexes from the current database.
The DROP INDEX statement is used to delete an index in a table.
Syntax : DROP INDEX index_name ON table_name

Why Use theDROP INDEX Statement?

You may drop an index permanently when it is no longer useful or temporarily. If the index is harming or not helping performance, it could be dropped.

Indexes may slow down the loading of data because they must be maintained during the data load process. For high performance loads, an index could be dropped for the duration of a load and then recreated.

To delete an index by using Object Explorer, you can follow the steps as shown below:

- In Object Explorer, expand the database that contains the table on which you want to delete an index.
- Expand the Tables folder.
- Expand the table that contains the index you want to delete.
- Expand the Indexes folder.
- Right-click the index you want to delete and select Delete.
- In the Delete Object dialog box, verify that the correct index is in the Object to be deleted grid and click OK.

To delete an index using Table Designer

- In Object Explorer, expand the database that contains the table on which you want to delete an index.
- Expand the Tables folder.
- Right-click the table that contains the index you want to delete and click Design.
- On the Table Designer menu, click Indexes/Keys.
- In the Indexes/Keys dialog box, select the index you want to delete.
- Click Delete.
- Click Close.
- On the File menu, select Save table_name.

- **View and edit indexes**

To view all indexes in a database

- In Object Explorer, connect to an instance of the SQL Server Database Engine and then expand that instance.
- Expand Databases, expand the database that contains the table with the specified index, and then expand Tables.
- Expand the table in which the index belongs and then expand Indexes.

To modify an index using wizard

› In Object Explorer, connect to an instance of the SQL Server Database Engine and then expand that instance.
› Expand Databases, expand the database in which the table belongs, and then expand Tables.
› Expand the table in which the index belongs and then expand Indexes.
› Right-click the index that you want to modify and then click Properties.
› In the Index Properties dialog box, make the desired changes. For example, you can add or remove a column from the index key, or change the setting of an index option.

- **Create multiple-field keys**

Relational database designs use a set of columns as the primary key for a table. When this set includes more than one column, it is known as a "composite" or "compound" primary key.

If the values in a single common key field are insufficiently unique to accurately join or relate two tables, you need to use multiple common key fields in combination.

➢ **Monitoring the lock options chosen for the database**

Database locks serve to protect shared resources or objects.

These protected resources could be:

- Tables
- Data Rows
- Data blocks
- Cached Items
- Connections
- Entire Systems

There are also many types of locks such as shared locks, transaction locks, DML locks, and backup-recovery locks.

➢ **Monitoring the data storage space**

This section describes the storage structures of your database, and explains how to monitor and manage the amount of storage that is in use and available for the database and its backups. It contains the following topics:

› About the Database Storage Structures
› Monitoring Storage Space Usage
› Compacting Storage
› Viewing Log Files

A database is the collection of logical and physical structures that together contain all the data and metadata for your applications. The database also contains control structures (such as control files) that it needs for startup and operation.

Date: ____/____/10

- **Updating data according to organizational guidelines**

Updating data

You usually use the following two application pages to update data in a database:

- An update form
- An update action page

You can create an update form that calls an update action page. The update action page should also contain a confirmation message for the end user.
The modification of data that is already in the database is referred to as updating. You can update individual rows, all the rows in a table, or a subset of all rows. Each column can be updated separately without affecting the other columns.

Syntax: UPDATE {table_name| view_name}

SET {column_name= {expression|default | null}}|

WHERE {search condition}

Use the UPDATE statement to change single rows, groups of rows, or all of the rows in a table.

When you update rows, consider the following facts and guidelines:

- Specify the new values with the SET clause
- Verify that the input values have compatible data types with the data types that are defined for the columns
- You can change the data in only one table at a time
- You can set one or more columns or variables to an expression.
- You can specify the search condition in where clause to update the values of selected column.

Instruction Sheet-3	Learning Guide #3

This learning guide is developed to provide you the necessary information regarding the following content coverage and topics.

- Allocating or removing access privileges according to user status
- Monitoring ***network server*** log-in log file for illegal log-in attempts or for security breach
- Managing system resources in the context of database administration

This guide will also assist the trainee to attain the learning outcome stated in the cover page. Specifically, upon completion of this Learning Guide, the trainee will be able to:

- Allocate or removing access privileges according to user status
- Monitor ***network server*** log-in log file for illegal log-in attempts or for security breach

Date: ____/____/10

- Manage system resources in the context of database administration

Learning Instructions:

1. Read the specific objectives of this Learning Guide.
2. Read the information written in the "Information Sheets 3". Try to understand what are being discussed. Ask your teacher for assistance if you have hard time understanding them.
3. Accomplish the "Self-check 3".
4. Ask from your teacher the key to correction (key answers) or you can request your teacher to correct your work. (You are to get the key answer only after you finished answering the Self-check 3).
5. If you earned a satisfactory evaluation proceed to "Information Sheet unit of competency". However, if your rating is not satisfactory, see your teacher for further instructions,
6. Submit your accomplished Self-check. This will form part of your training portfolio.

LO3. Manage database access

3.1 Allocating or removing access privileges according to user status

3.1.1 The Database Administrator's Operating System Account

3.1.2 Administrative User Accounts

3.2 Monitoring ***network server*** log-in log file for illegal log-in attempts or for security breach

3.2.1 Backup operator
3.2.2 Account operator
3.2.3 Server operator
3.2.4 Domain administrator settings

3.3 Managing system resources in the context of database administration

LO3. Manage database access

Date: ____/____/10

- **Allocating or removing access privileges according to user status**

1. **Managing Server and database Security**
   - **Creating Logins**

Most Windows users need SQL Server login account to connect to SQL Server.
This topic shows how to create a SQL Server login account.

To create a SQL Server login that uses Windows Authentication using wizard

- In SQL Server, open Object Explorer and expand the folder of the server instance.
- Expand security folder, Right-click on login folder, and then select New Login.
- Click on the General, and enter the name of a Windows user in the Login name box.
- Select Windows Authentication, and then Click OK.

To create a SQL Server login that uses SQL Server Authentication using wizard

* In SQL Server, open Object Explorer and expand the folder of the server instance.
* Expand security folder, Right-click on login folder, and then select New Login.
* Click on the General, and enter the name of a Windows user in the Login name box.
* Select SQL Server Authentication. However, Windows Authentication is the more secure option.
* Enter a password for the login.
* Select the password policy options that should be applied to the new login, and then Click OK.
  - In general, enforcing password policy is the more secure option.

To create a SQL Server login that uses Windows Authentication using Transact-SQL code

- Open New Query Editor,
- use the following Transact-SQL **syntax**:

  CREATE LOGIN [computer Name\name of Windows User] FROM WINDOWS

  Example: CREATE LOGIN [PC-Name\Admin] from windows

To create a SQL Server login that uses SQL Server Authentication using T-SQL code

- ✓ Open New Query Editor
- ✓ Use the following Transact-SQL **syntax**:

  CREATE LOGIN [login_Name] WITH PASSWORD = 'password'

  Example: CREATE LOGIN student WITH PASSWORD=’abc/123’

- We can drop login by using the DROP LOGIN login_name statement

2. **Creating and Managing database Users**

To create a database user using SQL Server

- In SQL Server, open Object Explorer and expand the Databases folder.
- Expand the database in which to create the new database user.
- Expand Security folder, right click on user folder and select User.
- Select General, and enter a name for the new user in the User name box.

Date: ____/____/10

- In the Login name box, enter the name of a SQL Server login to map to the database user.
- Click OK.

To create a database user using T-SQL code

- ✓ Open New Query Editor
- ✓ Use the following Transact-SQL **syntax**:

CREATE USER <new user name> FOR LOGIN <login name>

Example: CREATE USER **Admin1** for login **Student**

3. Creating Schemas

In SQL Server, schema is an object that conceptually holds definitions for database objects such as tables, views, stored procedures, etc. The main advantage of creating a schema is that you can grant permissions to database objects by using a single CREATE SCHEMA statement.

Syntax: CREATE SCHEMA [schema_name] Authorization [user_name]

Example: create a **student** schema owned by **Admin** as follows:

Create schema **student** Authorization **Admin**

Create table stud (fname varchar(20), Id int, sex char(6))

We can use the DROP SCHEMA schema_name statement to remove schema from the database.

Note: Windows Authentication mode is the default and recommended authentication mode.

Configuring SQL Server Authentication Modes

To select or change the server authentication mode, follow these steps:

* In SQL Server, right-click on a desired SQL Server, select Properties and then Select Security
* Select the desired server authentication mode under Server Authentication and then click OK.
* In Object Explorer, right-click on a desired server and then click Restart.

- Using Windows authentication is a more secure choice.

➢ Managing Server and database Security

1. Creating Roles

Role is a random set of privileges that is granted to users. There are three types of roles in SQL server:

- Fixed server roles
- Fixed database roles
- User defined database roles

We cannot create or change server level roles, but it is possible database level role.
After you create a database level role, configure the database-level permissions of the role by using GRANT, DENY, and REVOKE.
Users can be added to a fixed server level role using sp_addsrvrolemember stored procedure.

Syntax: sp_addsrvrolemember ‘role, ‘user_nane’

Example: sp_addsrvrolemember ‘dbcreater’,’u1’

To add members to a database role, use the sp_addrolemember stored procedure.
Syntax: CREATE ROLE role_name [AUTHORIZATION owner_name]

- Role_name is the name of the role to be created.
- AUTHORIZATION owner_name is the database user or role that is to own the new role. If no user is specified, the role will be owned by the user that executes CREATE ROLE.

Example: CREATE ROLE student

-To add user u1 to be the member of student role, EXECUTE sp_addrolemember ‘student’,’u1’

-To add user u1 to be the member of fixed database role, EXECUTE sp_addrolemember

Example: sp_addrolemember ‘db_accessadmin’,’u1’

- We can drop roles using the Drop role role_name code

- We can remove membership from roles using sp_droprolemember stored procedure

Example; sp_droprolemember db_accessadmin, **‘u1’**

2. **Granting Permissions**

The GRANT statement is used to give privilege to users or roles.

Note: if the permission is given via the [WITH GRANT OPTION], all users in the TO clause can themselves pass on the privilege to other users.

Examples: GRANT SELECT ON student to u1

GRANT SELECT, INSERT, UPDATE (salary) ON employee to u1

GRANT SELECT ON student to u1 WITH GRANT OPTION

GRANT CREATE TABLE TO u1 WITH GRANT OPTION

3. **Revoking Permissions**

Revoke statement is used to withdraw privileges from a user without deleting that user.

Syntax: REVOKE [GRANT OPTION FOR]

[GRANT OPTION FOR]:- Indicates that the ability to grant the specified permission will be revoked.

Examples: REVOKE DELETE ON employee from u1

REVOKE DELETE, INSERT ON employee from u1

REVOKE GRANT OPTION FOR DELETE ON EMPLOYEE FROM U1 CASCAD

- **The Database Administrator's Operating System Account**

Date: ____/____/10

To perform the administrative tasks of a Database, you need specific privileges within the database and possibly in the operating system of the server on which the database runs.
Depending on the operating system on which Database is running, you might need an operating system account or ID to gain access to the operating system. Your operating system account might require operating system privileges or access rights that other database users do not require.

- **Administrative User Accounts**

What is the difference between Database Administrator and System Administrator?

A database administrator is a person responsible for the installation, configuration, upgrade, administration, monitoring and maintenance of databases in an organization.

The role includes the development and design of database strategies, system monitoring and improving database performance and capacity, and planning for future expansion requirements. They may also plan,

Co-ordinate and implement security measures to safeguard the database.

A System Administrator is generally responsible for all parts of the computer network, such as user accounts, computer accounts, domain trusts, email accounts, etc.

The System Administrator is probably specialized in the network server operating systems and user administration; where as a Database Administrator will be highly specialized with the specific database server and client.

A network administrator maintains network infrastructure such as switches and routers, and diagnoses problems with these or with the behavior of network-attached computers.

Authorization, privileges, and roles

Users can successfully execute operations only if they have the authority to perform the specified function.
For example: To create a table, a user must be authorized to create tables; to alter a table.

Authorization

In computing systems, authorization is the process of determining which permissions a person or system is supposed to have. In multi-user computer systems, a system administrator defines which users are allowed access to the system, as well as the privileges of use for which they are eligible (e.g., access to file directories, hours of access, amount of allocated storage space).

Privileges

A **privilege** is a permission to perform an action or a task. Authorized users can create objects, have access to objects they own, and can pass on privileges on their own objects to other users by using the GRANT statement. Privileges may be granted to individual users or roles (groups).
You can apply five different kinds of user privileges. A user may be able to view, delete, insert, or update information in a table or view.
A user who has no privileges to a table is not able to use the table at all.

Role

Date: ____/____/10

A **role** is a group of privileges that can be granted to users as one unit. You can create roles and assign users to certain roles. A single user may have more than one role assigned, and a single role may have more than one user assigned. All roles are granted to users with the GRANT ROLE statement.

- **Monitoring network server log-in log file for illegal log-in attempts or for security breach**

Network monitoring

The term **network monitoring** describes the use of a system that constantly monitors a computer network for slow or failing components and that notifies the network administrator in case of outages.

Network Server Monitoring allows a network administrator to track the health of network servers in real time. Network Server Monitoring can identify servers that are in danger of malfunctioning before a malfunction occurs so that the administrator can proactively repair the server. Network Server Monitoring allows a single administrator to maintain many remote network servers.

- **Backup operator**

A backup operator is a user that can backup and restore the computer regardless of file system security.
By default, users are allowed to backup and restore files for which they have the appropriate file and directory permissions without requiring membership in the Backup Operators group.
The Backup Operators group allows users to backup and restore files regardless of whether they have read or write access to the files.

- **Account operator**

By default, Account Operators have permission to create, modify, and delete accounts for users, groups, and computers in all containers and organizational units of Active Directory except the Built-in container and the Domain Controllers.
Account Operators do not have permission to modify the Administrators and Domain Admins groups, nor do they have permission to modify the accounts for members of those groups.

- **Server operator**

Server Operators is a local group that allows a user to perform general administrator tasks. These tasks include sharing server resources, performing file backup and recovery, etc. As with other operator accounts, Server Operators can also log on to a server locally and shut it down. Server Operators can perform most common server administration tasks.
Members of this group can perform server management tasks such as creating, changing, and deleting shared printers, shared directories, and files. They can also backup and restore files, lock the server console and shutdown the system, but they cannot modify system policies or start and stop services.

- **Domain administrator settings**

The domain administrator creates, Edit and deletes users, manages domains settings and View domains statistics.

The domain administrator account members are allowed administrative privileges for the entire domain.
By default, the group has the local Administrator account on the Domain Controller as its member.
When a computer joins a domain, the Domain Administrator group is added to the Administrators group.
When a server becomes a domain controller, the Enterprise Administrator group also is added to the Administrators group. The Administrators group has built-in capabilities that give its members full control over the system. The group is the default owner of any object that is created by a member of the group.

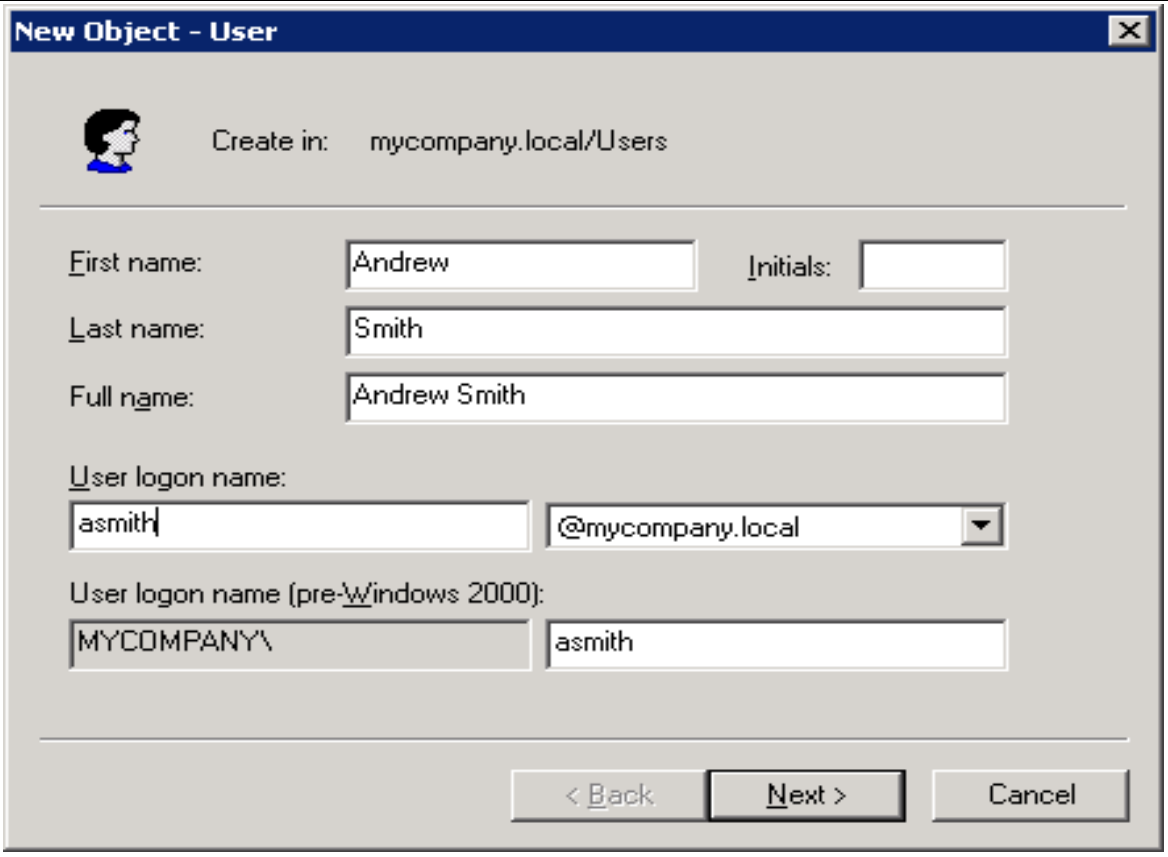
Setting Up Domain Administrator Account

After you have created and configured the Active Directory domain, you should make a domain administrator account. To set up a domain administrator account, you should:

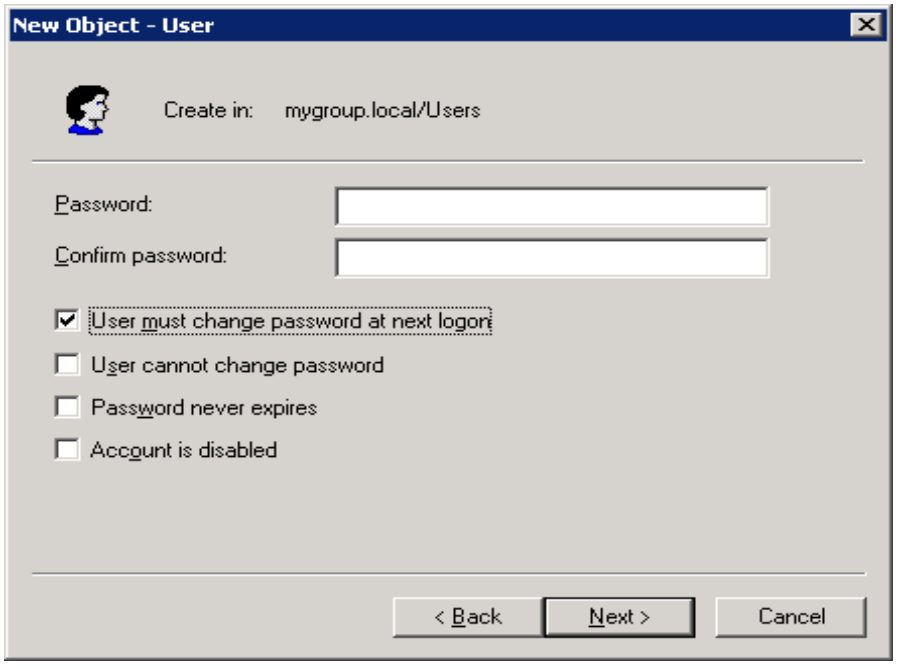
- create a new user on the domain controller;
- Include the newly created user in the Domain Admins group.

First, you should create a new user account on the domain controller. To this effect, complete the following tasks:

1. Log in to the domain controller.
2. Click **Start**, point to **Administrative Tools**, and click **Active Directory Users and Computers**.
3. In the left pane of the **Active Directory Users and Computers** window, expand the contents of the newly created Active Directory domain.
4. Right-click the **Users** folder, point to **New**, and select **User**.
5. In the **New Object - User** window, do the following:
    - Type your first and last names in the **First name** and **Last name** fields, respectively.
    - In the **User logon name** field, type a name that will be used to log on to the Active Directory domain. For example:



6. After providing the necessary information, click **Next**.
7. Specify an arbitrary password for the domain administrator account and click **Next**.

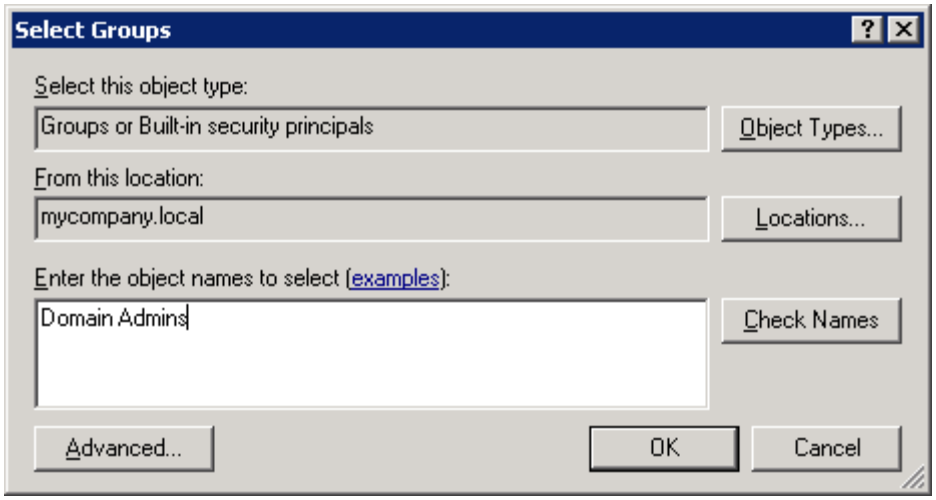


8. The last window allows you to review the parameters provided by on the previous steps. If you wish to modify any parameters, click **Back**; otherwise, click **Finish** to create the domain administrator account.

Date: ____/____/10

Now you should include the newly created account in the `Domain Admins` group, which will allow this account to perform administrative tasks in the domain context. To this effect, do the following:

9. In the **Active Directory Users and Computers** window (**Start** --> **Administrative Tools** --> **Active Directory Users and Computers**), right-click the created user account and select **Properties**.
10. Select the **Member Of** tab and click **Add**.
11. In the **Select Groups** dialog box, type `Domain Admins` and click **OK**.



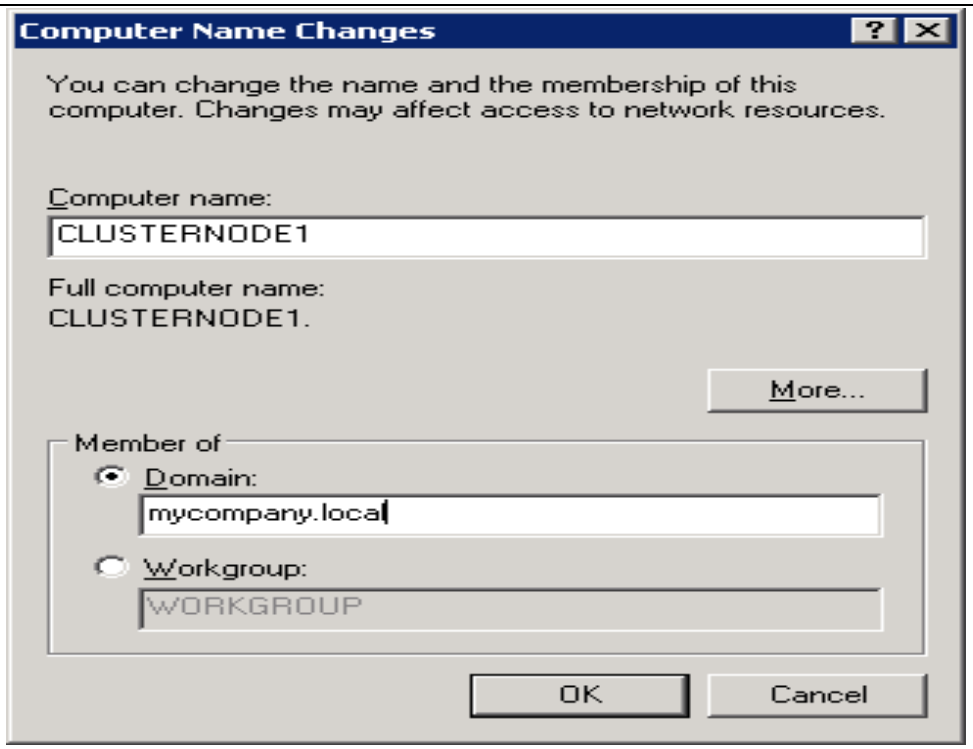
12. Click **OK**.

Adding Nodes to Domain

After you have created the Active Directory domain and the domain administrator account, you should add all the nodes to the domain. This can be done as follows:

1. Log in to the first node you wish to add to the domain, right-click the **My Computer** icon, and click **Properties**.
2. Select the **Computer Name** tab and click **Change**.
3. In the **Computer Name Changes** window, do the following:
    - In the **Computer name** field, specify a server hostname. This name will be used to uniquely identify the given node among other nodes in the cluster. By default, you are offered to use the hostname assigned to the node during the Windows Server 2003 installation. However, we recommend that you change this hostname to something more descriptive (e.g. `CLUSTERNODE1`).
    - Select the **Domain** radio button and type the domain DNS name (you specified this name during the Active Directory domain). In our example the domain DNS name should be set to `mycompany.local`.

After providing the necessary information, your window may look like the following:



When you are ready, click **OK**.

4. In the **Computer Name Changes** window, type the username and password of the domain administrator account and click **OK**.

5. Click **OK** to close the displayed message welcoming you to the domain and then click **OK** once more to close the **Computer Name Changes** window.
6. Restart the node.
7. Perform **Steps 1-6** for all the remaining cluster nodes.

❖ **Managing *system* resources in the context of database administration**

Date: ____/____/10

System resource is a tool used by either **hardware** to alert **software** of a need or by software to control a function of hardware.

Resource management is the dynamic allocation and de-allocation by an operating system of processor cores, memory pages, and various types of bandwidth to computations that compete for those resources. The objective is to allocate resources so as to optimize responsiveness subject to the finite resources available.

Data administration or data resource management is an organizational function working in the areas of information systems and computer science that plans, organizes, describes and controls data resources. Data resources are usually as stored in databases under a database management system.

Date: ____/____/10