

References :

1. Hassan A. Afyouni, “Database Security and Auditing”, Third Edition, Cengage Learning, 2009
2. Charu C. Aggarwal, Philip S Yu, “Privacy Preserving Data Mining”: Models and Algorithms, Kluwer Academic Publishers, 2008
3. Ron Ben Natan, ”Implementing Database Security and Auditing”, Elsevier Digital Press, 2005.

##### 15CS338E – DATABASE SECURITY AND PRIVACY

UNIT II : ADMINISTRATION OF USERS & PROFILES, PASSWORD POLICIES,PRIVILEGES AND ROLES

✔ Administration of Users

* + Introduction
  + Authentication
  + Creating Users

✔ SQL Server

* + User Removing
  + Modifying Users
  + Default Users

✔ Remote Users

✔ Database Links

✔ Linked Servers

✔ Remote Servers

✔ Practices for administrators and Managers- Best Practices

✔ Profiles, Password Policies, Privileges and Roles

* + Introduction
  + Defining and Using Profiles
  + Designing and Implementing Password Policies

✔ Granting and Revoking User Privileges

✔ Creating, Assigning and Revoking User Roles-Best Practices

✔ Introduction

* + - Authentication and Authorization are essential services for every OS
    - Another service is Administration of Users
    - Administrators use this functionality
      * Creating users
      * Set Password Policies
      * Grant privileges

✔ At every type of organization, many security violations are caused by negligence and ignorance and in particular by failing to consider documentation

✔ Documentation is a main part of administration process

✔ There top three excuses for failing to incorporate documentation

* Lack of Time
* Belief that the administration process is already in documented in the system
* Reluctance to complicate a process that is simple

✔ Everything is documented for two reasons

* To provide a paper trail to retrace exactly what happened when breach of security occurs
* To ensure administration consistency

Documentation in Administration context includes the following

✔ Administration Policies

* Documentation includes all policies for handling new and terminated employees, managers, system and database administrator, database managers, operation managers, and human resources.
* A detailed document should describe guidelines for every task that is required for all common administrative situations.

✔ Security Procedures

* This is an outline of a step-by-step process for performing administrative task according to company policies.

✔ Procedures implementation scripts and programs

* This is documentation of any script or program used to perform an administrative task.
* This includes user’s manual and operational manual

Documentation in Administration context includes the following …

✔ Predefined roles description

* + This provides the full description of all predefined roles, outlining all tasks for which the role is responsible and the role’s relationship to other roles

✔ Administration staff and management

* + This is usually a detailed description of each administration staff and management position.
  + This document includes an organizational chart.

Documentation of User Administration …

Many companies develop procedures and forms used to perform any security-related process. The following figure presents a sample process of creating a database user account that you can customize per your business requirements and company policies.

DBA Completes all the paper work and documentation for new employees

Access Identification

Document Completion



DBA provides list of access operations that are necessary for employees to perform their jobs

DBA completes the database user account application form DBA obtains department Manger’s approval on the application

Operational Approval

Department Approval

Account application Completion

DBA obtains operational Manger’s approval on the application

DBA or Operator creates the account Account holder verifies access

Test Access

Implement Access

# Creating users

#### ✔ Creating users is one of the main tasks you will perform as a database operator or DBA

✔ In most organization , this process is standardized , well documented, and surely managed

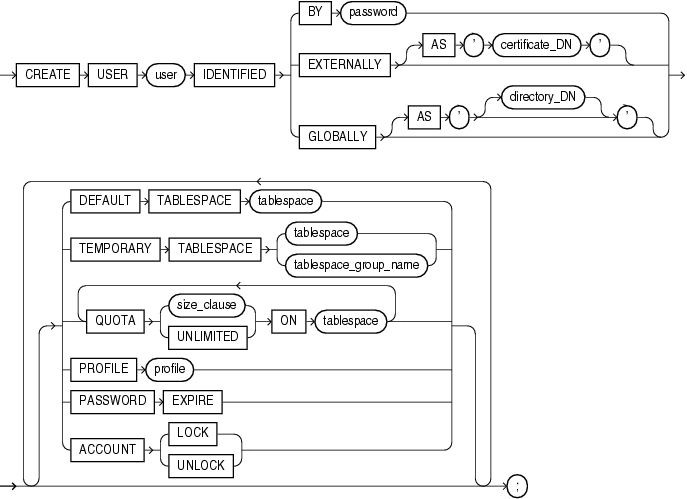
#### ✔ The DBA had written a script to create a user for every developer working on the project

✔ This script granted privileges to read and write data to the database scheme

✔ Regardless of the database you use , creating the user is generally an easy task once a policy is documented and followed

# Creating users …

Creating an ORACLE 10g User



# Creating users …

***user***

✔ Specify the name of the user to be created. This name can contain only characters from your database character set and must follow the rules described in the section "Schema Object Naming Rules". Oracle recommends that the user name contain at least one single-byte character regardless of whether the database character set also contains multibyte characters.

**IDENTIFIED Clause**

✔ The IDENTIFIED clause lets you indicate how Oracle Database authenticates the user.

**BY password**

✔ The BY *password* clause lets you creates a **local user** and indicates that the user must specify *password* to log on to the database. Passwords are case sensitive. Any subsequent CONNECT string used to connect this user to the database must specify the password using the same case (upper, lower, or mixed) that is used in this CREATE USER statement or a subsequent ALTER USER statement. Passwords can contain any single-byte, multibyte, or special characters, or any combination of these, from your database character set

**EXTERNALLY Clause**

✔ Specify EXTERNALLY to create an **external user.** Such a user must be authenticated by an external service, such as an operating system or a third-party service. In this case, Oracle Database relies on authentication by the operating system or third-party service to ensure that a specific external user has access to a specific database user.



AS '*certificate\_DN*'

✔ This clause is required for and used for SSL-authenticated external users only. The *certificate\_DN* is the distinguished name in the user's PKI certificate in the user's wallet.

GLOBALLY Clause

✔ The GLOBALLY clause lets you create a global user. Such a user must be authorized by the enterprise directory service (Oracle Internet Directory).

DEFAULT TABLESPACE Clause

✔ Specify the default tablespace for objects that the user creates. If you omit this clause, then the user's objects are stored in the database default tablespace. If no default tablespace has been specified for the database, then the user's objects are stored in the SYSTEM tablespace.

✔ Restriction on Default Tablespaces You cannot specify a locally managed temporary tablespace, including an undo tablespace, or a dictionary-managed temporary tablespace, as a user's default tablespace.

TEMPORARY TABLESPACE Clause

✔ Specify the tablespace or tablespace group for the user's temporary segments. If you omit this clause, then the user's temporary segments are stored in the database default temporary tablespace or, if none has been specified, in the SYSTEM tablespace.

✔ Specify *tablespace* to indicate the user's temporary tablespace.

✔ Specify *tablespace\_group\_name* to indicate that the user can save temporary segments in any tablespace in the tablespace group specified by *tablespace\_group\_name*.

✔ Restrictions on Temporary Tablespace

* This clause is subject to the following restrictions:
* The tablespace must be a temporary tablespace and must have a standard block size.
* The tablespace cannot be an undo tablespace or a tablespace with automatic segment-space management.



✔ QUOTA Clause

* Use the QUOTA clause to specify the maximum amount of space the user can allocate in the tablespace.
* A CREATE USER statement can have multiple QUOTA clauses for multiple tablespaces.
* UNLIMITED lets the user allocate space in the tablespace without bound.
* Restriction on the QUOTA Clause You cannot specify this clause for a temporary tablespace.

✔ PASSWORD EXPIRE Clause

* Specify PASSWORD EXPIRE if you want the user's password to expire. This setting forces the user or the DBA to change the password before the user can log in to the database.

✔ ACCOUNT Clause

* Specify ACCOUNT LOCK to lock the user's account and disable access.

Specify ACCOUNT UNLOCK to unlock the user's account and enable access to the account.

#### ✔ The following create user statement implements the creation of user called bmnantha

SQL> CREATE USER bmnantha IDENTIFIED BY bmnantha23

1. DEFAULT TABLESPACE users
2. TEMPORARY TABLESPACE temp
3. QUOTA 25M ON users
4. PROFILE default
5. PASSWORD EXPIRE
6. ACCOUNT UNLOCK 8 /

User created

#### ✔ Once the user is created you can modify a user account with an ALTER USER statement using clause listed in the previous example

✔ DBA\_USERS describes all users of the database.

Column

USER NAME

Datatype VARCHAR2(30)

NULL NOT NULL

Name of the user

Description

USER\_ID

NUMBER

NOT NULL

ID number of the user

PASSWORD VARCHAR2(30)

This column is deprecated in favor of

the AUTHENTICATION\_TYPE column

ACCOUNT\_ STATUS

VARCHAR2(32)

NOT NULL

Account status:

✔ OPEN

✔ EXPIRED

✔ EXPIRED(GRACE)

✔ LOCKED(TIMED)

✔ LOCKED

✔ EXPIRED & LOCKED(TIMED)

✔ EXPIRED(GRACE) & LOCKED(TIMED)

✔ EXPIRED & LOCKED

✔ EXPIRED(GRACE) & LOCKED

|  |  |  |  |
| --- | --- | --- | --- |
| Column | Datatype | NULL | Description |
| LOCK\_DATE | DATE |  | Date the account was locked if account status was LOCKED |
| EXPIRY\_DAT E | DATE |  | Date of expiration of the account |
| DEFAULT\_ TABLESPACE | VARCHAR2(30) | NOT NULL | Default tablespace for data |
| TEMPORARY  \_TABLESPAC E | VARCHAR2(30) | NOT NULL | Name of the default tablespace for temporary tables or the name of a tablespace group |
| CREATED | DATE | NOT NULL | User creation date |
| PROFILE | VARCHAR2(30) | NOT NULL | User resource profile name |
| INITIAL\_RSR C\_CONSUME R\_GROUP | VARCHAR2(30) |  | Initial resource consumer group for the user |

|  |  |  |  |
| --- | --- | --- | --- |
| Column | Datatype | NULL | Description |
| EXTERNAL\_ NAME | VARCHAR2(4000  ) |  | User external name |
| PASSWORD\_ VERSIONS | VARCHAR2(8) |  | Database version in which the password was created or changed |
| EDITIONS\_ ENABLED | VARCHAR2(1) |  | Indicates whether editions have been enabled for the corresponding user (Y) or not (N) |
| AUTHENTICATI ON\_TYPE | VARCHAR2(8) |  | Indicates the authentication mechanism for the user:  ✔ EXTERNAL - CREATE  USER *user1* IDENTIFIED EXTERNALLY;  ✔ GLOBAL - CREATE  USER *user2* IDENTIFIED GLOBALLY;  ✔ PASSWORD - CREATE  USER *user3* IDENTIFIED BY *user3*; |

✔ To create a login id in SQL server can be member of SYSTEMADMIN OR SECURITYADMIN

✔ There are two types of login IDs:

* Windows Integrated (Trusted) Logins
  + User can associate a Microsoft Windows account or group with either the server in which SQL Server is installed or the domain in which the server is a member
* SQL Server Login



Creating Windows integrated Logins

✔ From the command Line

To create a new login associated with a Window account (Windows Integrated) , in the Query Analyser tool use the SP\_GRANTLOGIN system Procedure .

✔ The syntax is as follows:

sp\_grantlogin [@login =] ‘login’

✔ The login syntax is the fully qualified name of the Windows user account in the form of *machine\_name\user\_name* for local Windows users.

✔ *domain\username* for Windows domain accounts.

✔ Windows integrated login can also be associated can also be associated with windows groups on either the local server or domain

For example,

✔ If you have a local windows account named ‘bmnantha’ on the SQL Server itself where the server name is myserver, you enter the following

exec sp\_grantlogin ‘myserver\bmnantha’

✔ For windows domain account named ‘manish’ in the mydomain, you are entering the following

exec sp\_grantlogin ‘mydomain\manish’

✔ To associate local windows group called SQL\_DBA , you are entering

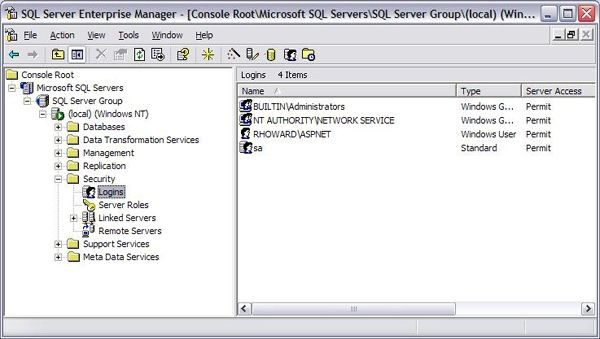
exec sp\_grantlogin ‘myserver\sql\_dba

✔ NOTE : A login must be between 1 to 128 characters in length and cannot contain any spaces.

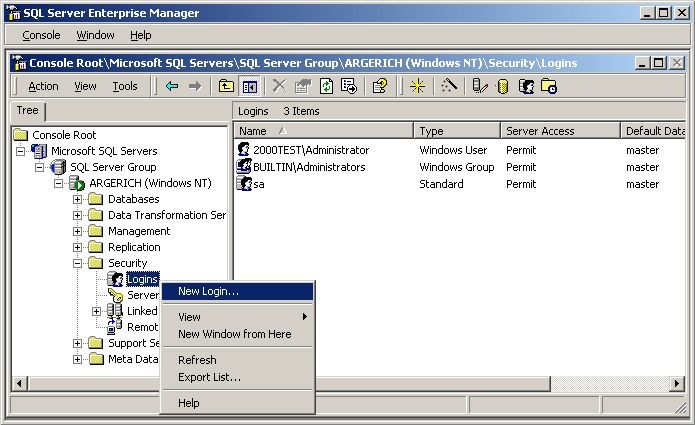
**Creating a SQL Server User from Enterprise Manager**

To create a new login associated with a Windows account (Windows Integrated) in Enterprise Manager, take the following steps

1. Open Enterprise Manager



1. Expand the server group in which your server is functioning
2. Expand the server you want to create the login for
3. Expand the security container
4. Click Logins
5. On the menu bar , click action , then click new login

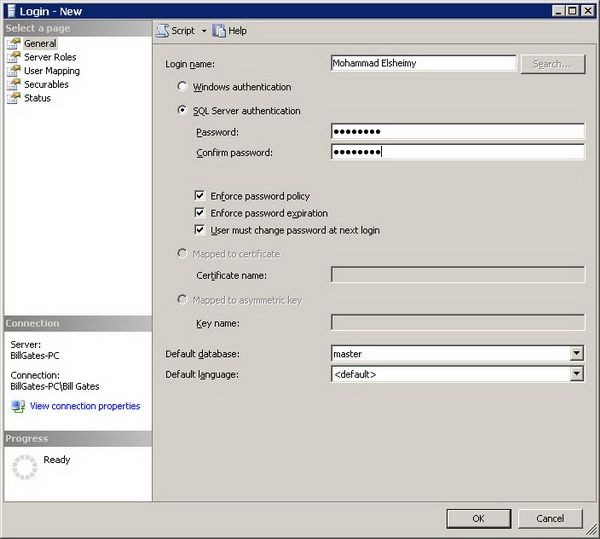


1. Type the name of user
2. Depending on the type of Windows account you are creating , select either the local server name or the domain name from the domain drop-down list. Enterprise Manager automatically fills in

the machine or domain name in front of the username

1. Select the default database for the login from the Database drop-down list.
2. Select the default language for the login from the language drop-down list.



1. Click OK

✔ The second type of login is a SQL Server Login, sometimes called a SQL Server active login.

✔ This login associated with a windows account, instead , it is a security account created within SQL Server itself.

✔ Creating SQL Server Logins from command line

* + To create a SQL Server login from the Query analyzer , you use the SP\_ADDLOGIN system stored procedure.
  + The syntax is as follows :

@loginame – choose for the login

sp\_addlogin [@loginame = ] ‘login’ [ , [ @passwrd = ] ‘password’ ]

[ , [ @dbdef=] ‘database’]

[ , [ @deflanguage = ] ‘language’] [ , [ @sid =] sid]

[ , [ @encryptopt =] ‘encryption\_opotion’]

@dbdef – Name of the default database for the user, The default is NULL @deflanguage – The default language for the user.

The default is the current default language of the SQL Server Instance @sid – Security Identification Number (SID).

The default is NULL, if it is NULL SQL Server automatically generates SID for the login

@encryptopt – Specifies weather or not to encrypt the password in the database

For example

✔ To create a SQL Server login named ‘bmnantha’ with password ‘manish’ you issue the following command

exec sp\_addlogin ‘bmnantha’ , ‘manish’

✔ To specify a default database of Northwind for bmnantha, enter the following

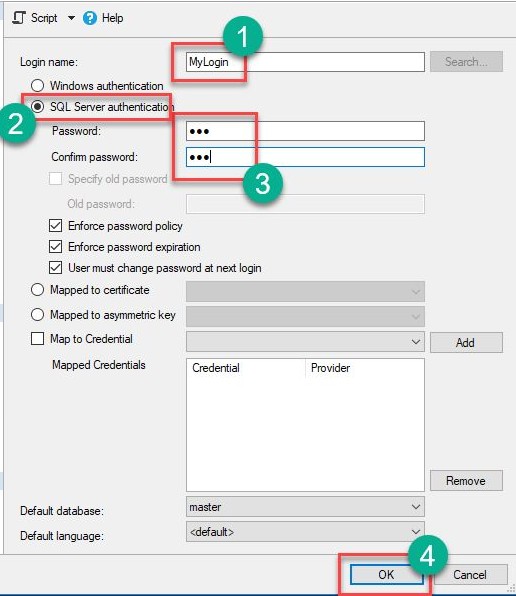
exec sp\_addlogin ‘bmnantha’, ‘manish’, ‘Northwind’

From Enterprise Manager

To create a new SQL Server login in Enterprise Manager , follow these steps

1. Open Enterprise Manager
2. Expand the server group your is in
3. Expand the server you want to create the login for.
4. Expand the Security container
5. Click Logins
6. On the menu bar , Click Action, then click New Login
7. Type the name of the user, in this case , bmnantha
8. Click the SQL Server Authentication option button
9. Provide a password for the user in the password textbox. The password is marked as you type
10. Click OK

The following figure gives the Server login properties – new login screen (Latest Version)



✔ Removing an ORACLE User

SQL > DROP USER SCOTT;

User Dropped

✔ If the user does not have any objects , the command is successfully executed. If the user own any objects CASECADE option should be used

SQL> DROP USER SCOTT CASCADE;

User Dropped

✔ SQL Server: Removing Windows Integrated Logins

From the command Line : Use the SP\_DENYLOGIN system procedures sp\_denylogin [ @loginame = ] ‘login’

✔ The following statement drop the login account bmnantha.

exec sp\_denylogin ‘myserver\bmnantha’

✔ From the Enterprise Manager

To drop the login in Enterprise Manager simply highlight the desired login and choose delete from the action menu



The existing user account can be changed such as password, database, tablespace, quota, password profile, account by the DBA

✔ Modifying an ORACLE User

SQL > ALTER USER SCOTT IDENTIFIED BY LION;

User Altered

✔ SQL Server : Modifying Windows Integrated Login Attributes

✔ From the Command Line

The default database for the user initially set to master, to change the database SP\_DEFAULTDB system stored procedure is used.

sp\_default [ @loginame = ] ‘login’ , [ @defdb =] ‘database’

✔ To change the default database to the login mydomain\bmnantha , issue the following statement

exec sp\_defaultdb ‘mydomain \bmnantha’ ,’Northwind’

✔ ORACLE default users, will be created at the time of ORACLE software installation

* SYS (Super user will all DBA rights , can’t be changed)
* SYSTEM (With Minimal DBA rights
* SCOTT (User without DBA rights)

✔ SQL server default users, will be created at the time of SQL Server software installation

* SA ( System Administrator , It is equivalent to SYS in Oracle and can’t be changed)
* BUILT-IN\Administrators ( Associated with the local administrators’ group on the Windows server)

✔ All the DB user accounts are created and stored in the DB regardless of whether they are connected locally or remotely.

✔ When a user logs on to the DB through the machine where the DB is located , called as Local user.

✔ When a user logs on to the DB through the machine where the DB is not located , called as remote user.

✔ ORACLE10g , remote users can be authenticated by the OS provided the REMOTE\_OS\_AUTHENT initialization parameter is set to TRUE. If the parameter is set to FALSE , user can’t login from remote.

✔ SQL Server does not support this type of remote user authentication.

✔ It is a connection from one DB to another DB

✔ The linked DBs can be like

* Both be ORACLE10g
* Both be SQL Server
* Mix of ORACLE10g and SQL Server

✔ A DB link enables a user to perform Data Manipulation Language (DML) or any other valid SQL statements on a DB.

✔ The following figure gives the architecture of DB Link

DB1

DB2

DB LINK

✔ In Oracle 10g ,DB Links can be created in two ways as

1. Public – Which makes the database links accessible by every user in DB 2.Private – Which gives the ownership of the database to a user

The DB is not accessible by any other user unless the user has been access by the owner



Authentication Methods

✔ Authentication methods for connecting ORACLE10g DB using DB link mechanism.

✔ There are three types of authentication methods when creating a DB link.

✔ Authentication Method 1: CURRENT USER

* This authentication method orders ORACLE10g to use the current user credentials for authentication to the DB to which the user is trying to link.

SQL > CONNECT SYSTEM@DB1

Enter password: \*\*\*\*\*\* Connected

SQL > CREATE PUBLIC DATABASE LINK DB2

1. CONNECT TO CURRENT\_USER
2. USING ‘DB2’ 4 /

Database link created

✔ Authentication Method 2: FIXED USER

This authentication method orders ORACLE10g to use the user password provided in this clause for authentication to the DB to which the user is trying to link.

SQL > CREATE PUBLIC DATABASE LINK DB2

1. CONNECT TO SCOTT IDENTIFIED BY TIGER
2. USING ‘DB2’ 4 /

Database link created

✔ Authentication Method 3: CONNECT USER

This authentication method orders ORACLE10g to use credentials of the connected user who has an existing account in the database to which the user is trying to link.

SQL > CREATE PUBLIC DATABASE LINK DB2

2 USING ‘DB2’ 3 /

Database link created



✔ Linked serves allow you to connect to almost any object Linking Embedding Database (OLEDB) or Open Database Connectivity .

✔ Microsoft SQL Server 2000 also uses the concept of linked serves.

✔ OLEDB is a Microsoft component that allows Windows applications to connect and access different database systems.

✔ ODBC is a Microsoft protocol used for connecting Windows applications to different DB systems

✔ The following figure represents the Linked server architecture using SQL Server

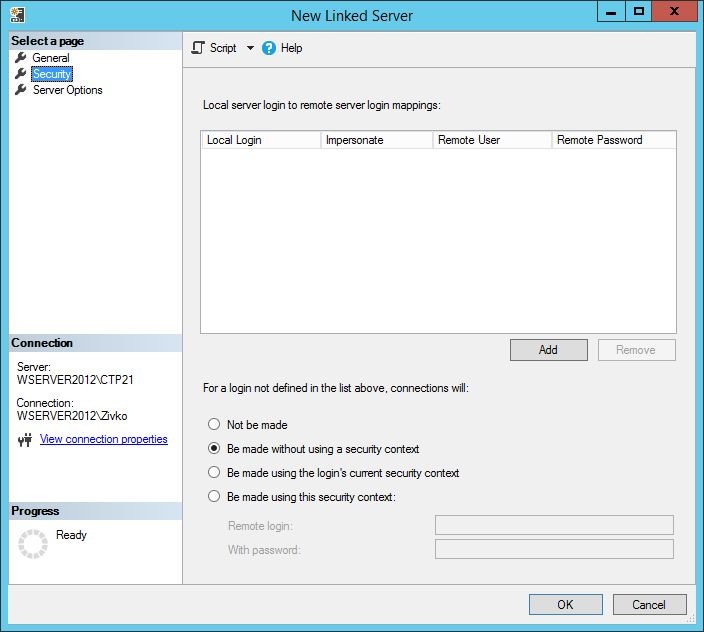
Server bmnantha Server manish



Linked Server

**Linked Server …**

Creating a new linked server with SQL Server



✔ Along the same line as Linked Servers , you can communicate with another SQL server by creating remote server

✔ Instead of using OLEDB , communications occurs across a Remote Procedure Call (RPC)

✔ The DBA job is never ending and very challenging

✔ DBA is constantly performing other administrative tasks such as backup, recovery and performance tuning.

✔ To make wise decisions DBA have the sizable responsibility of keeping up with database practices, database technology and database security issues.

✔ These are the best practices for administrating users, privileges , and roles.

* Follow you company ‘s procedures and policies to create , remove or modify database users.
* Always change the default password and never write it, or save it in a file that neither encrypted nor safe.
* Never share the user accounts with anyone , especially DBA accounts.
* Always document and create logs for changes to removals of database user accounts.

✔ These are the best practices for administrating users, privileges , and roles…

* + Never remove an account even if it is out dated, Instead disable or revoke connections privileges of the account.
  + Give access permission to users only as required and use different logins and passwords for different applications.
  + Educate users, developers and administrators on user administration best practices as well as the company policies and procedures.
  + Keep abreast (up-to date) of database and security technology. Should be aware of all new vulnerabilities that may increase database security risks.
  + Constantly review and modify the procedures as necessary to be in line up with the company’s policies and procedures. Keep procedures up to date with the dynamic nature of database and security technology

Introduction

✔ The key to the house is the password

✔ Put the scenario into the context of computer passwords.

✔ For home security , in addition to changing the key , you might install an alarm, , motion detector, camera, etc.,

✔ A company’s user accounts should have equal protection.

✔ The company needs to protect its assets and enforce stringent (strict, precise, and exacting) guidelines to protect the keys to computer accounts.

✔ This key is the password

* A profile is a security concept that describes the limitation of database resources that are granted database uses.
* A profile is a way of defining database user behaviour to prevent users from wasting resources such as memory and CPU consumption
* For this reason some DBMSs have implemented the profile concept.
* Not every DBMS offers profile concept.
* ORACLE does and Microsoft SQL Server 2000 doesn’t.

✔ Creating Profiles in ORACLE

✔ A profile in ORACLE helps define two elements of Security

✔ Restrictions on Resources

✔ Implementation of password policy

✔ The following figure shows the two aspects of a profile in ORACLE

S

RESOURCE

PASSWORD

PROFILE



Aging Usage

Verification

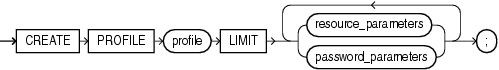
CPU

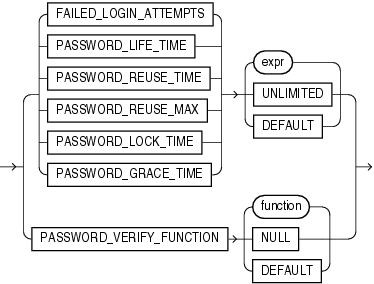
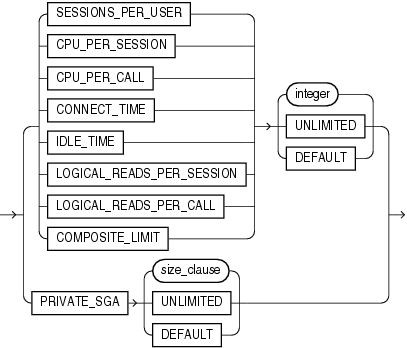
Memory

Connections

ORACLE allows you to create a profiles using the CREATE PROFILE

statement. The full syntax of the statement follows

Create profile

Resource parameters Password parameters



Resource Limits

Password Limits

CREATE PROFILE Profile\_name LIMIT

SESSIONS\_PER\_USER number CPU\_PER\_SESSION hunderth of seconds CPU\_PER\_CALL hunderth of seconds CONNECT\_TIME UNLIMITED minutes

IDLE\_TIME minutes

LOGICAL\_READS\_PER\_SESSION DEFAULT db\_blocks LOGICAL\_READS\_PER\_CALL DEFAULT db blocks COMPOSITE\_LIMIT DEFAULT number

PRIVATE\_SGA bytes FAILED\_LOGIN\_ATTEMPTS number PASSWORD\_LIFE\_TIME days

PASSWORD\_REUSE\_TIME number PASSWORD\_REUSE\_MAX number PASSWORD\_LOCK\_TIME days PASSWORD\_GRACE\_TIME days

PASSWORD\_VERIFY\_FUNCTION function\_name;

✔ In this syntax:

* + First, specify the name of the profile that you want to create.
  + Second, specify the LIMIT on either database resources or password

✔ Resource Parameters

* + SESSIONS\_PER\_USER – specify the number of concurrent sessions that a user can have when connecting to the Oracle database.
  + CPU\_PER\_SESSION – specify the CPU time limit for a user session, represented in hundredth of seconds.
  + CPU\_PER\_CALL – specify the CPU time limit for a call such as a parse, execute, or fetch, expressed in hundredths of seconds.
  + CONNECT\_TIME – specify the total elapsed time limit for a user session, expressed in minutes.
  + IDLE\_TIME – specify the number of minutes allowed periods of continuous inactive time during a user session. Note that the long-running queries and other operations will not subject to this limit.
  + LOGICAL\_READS\_PER\_SESSION – specify the allowed number of data blocks read in a user session, including blocks read from both memory and disk.
  + LOGICAL\_READS\_PER\_CALL – specify the allowed number of data blocks read for a call to process a SQL statement.
  + PRIVATE\_SGA – specify the amount of private memory space that a session can allocate in the shared pool of the system global area (SGA).
  + COMPOSITE\_LIMIT – specify the total resource cost for a session, expressed in service units. The total service units are calculated as a weighted sum of

of CPU\_PER\_SESSION CONNECT\_TIME, LOGICAL\_READS\_PER\_SESSION, and PRIVATE\_SGA.

✔ Password\_parameters

* You use the following clauses to set the limits for password parameters:
* FAILED\_LOGIN\_ATTEMPTS – Specify the number of consecutive failed login attempts before the user is locked. The default is 10 times.
* PASSWORD\_LIFE\_TIME – specify the number of days that a user can use the same password for authentication. The default value is 180 days.
* PASSWORD\_REUSE\_TIME – specify the number of days before a user can reuse a password.
* PASSWORD\_REUSE\_MAX – specify the number of password changes required before the current password can be reused. Note that you must set values for

both PASSWORD\_REUSE\_TIME and PASSWORD\_REUSE\_MAX parameters make these parameters take effect.

* PASSWORD\_LOCK\_TIME – specify the number of days that Oracle will lock an account after a specified number of a consecutive failed login. The default is 1 day if you omit this clause.
* PASSWORD\_GRACE\_TIME – specify the number of days after the grace period starts during which a warning is issued and login is allowed. The default is 7 days when you omit this clause.

✔ Note that to create a new profile, your user needs to have the CREATE PROFILE system privilege.

**Setting Profile Resource Limits: Example** The following statement creates the profile app\_user:

SQL> CREATE PROFILE app\_user 2 LIMIT

1. SESSIONS\_PER\_USER UNLIMITED
2. CPU\_PER\_SESSION UNLIMITED
3. CPU\_PER\_CALL 3000
4. CONNECT\_TIME 45
5. IDLE\_TIME 15
6. LOGICAL\_READS\_PER\_SESSION DEFAULT
7. LOGICAL\_READS\_PER\_CALL 1000
8. PRIVATE\_SGA 15K
9. COMPOSITE\_LIMIT 5000000;

12 / Profile created



✔ To view all profiles created in the database , query the data dictionary view, DBA\_PROFILES

SQL> select \* from dba\_profiles where profile = 'DEFAULT';

PROFILE RESOURCE\_NAME RESOURCE\_TYPE LIMIT

DEFAULT COMPOSITE\_LIMIT KERNEL UNLIMITED DEFAULT SESSIONS\_PER\_USER KERNEL UNLIMITED DEFAULT CPU\_PER\_SESSION

KERNEL UNLIMITED DEFAULT CPU\_PER\_CALL KERNEL UNLIMITED DEFAULT LOGICAL\_READS\_PER\_SESSION KERNEL

UNLIMITED DEFAULT LOGICAL\_READS\_PER\_CALL KERNEL UNLIMITED DEFAULT IDLE\_TIME KERNEL UNLIMITED DEFAULT CONNECT\_TIME KERNEL UNLIMITED DEFAULT PRIVATE\_SGA

KERNEL UNLIMITED DEFAULT FAILED\_LOGIN\_ATTEMPTS PASSWORD UNLIMITED DEFAULT PASSWORD\_LIFE\_TIME PASSWORD UNLIMITED DEFAULT PASSWORD\_REUSE\_TIME PASSWORD UNLIMITED DEFAULT PASSWORD\_REUSE\_MAX PASSWORD UNLIMITED DEFAULT PASSWORD\_VERIFY\_FUNCTION PASSWORD NULL

DEFAULT PASSWORD\_LOCK\_TIME PASSWORD UNLIMITED DEFAULT PASSWORD\_GRACE\_TIME PASSWORD UNLIMITED

16 rows selected.



✔ To Modify a limit for profile , you use ALTER PROFILE as follows

SQL> ALTER PROFILE APP\_USER

2 LIMIT IDLE\_TIME 30; Profile altered

✔ To assign a profile , use ALTER USER as follows

SQL> ALTER USER BMNANTHA PROFILE APP\_USER

2 / User altered

✔ In SQL Server 2000 or 2005 profiles of similar objects are not available

✔ Password is key to opening the user account.

✔ The stronger the password, the longer it takes a hacker to break it.

✔ Many hackers security violations begin with breaking password.

✔ If you joining any financial company the orientation program on security administration including password selection, password storage, and the company’s policies on password.

✔ Password policy is a set of guidelines that enhances the robustness of the password and reduces the likelihood of its being broken

✔ Importance of Password Policies

* The frontline defence of your account is your password.
* If your password is weak, the hacker can break in, destroy your data, and violate your sense of security .
* For this specific reason, most of the companies invest considerable resources to strengthen authentication by adopting technological measures that protect their assets.

Designing password policies

✔ Most companies use a standard set of guidelines for their password policies

✔ These guidelines can comprise one or more of the following

✔ Password Complexity – A set of guidelines used when selecting

password, for example minimum 8 characters, 1 special character, 1 Capital letter, etc.,

The purpose of password complexity is to decrease the chances of a hacker guessing or breaking a password.

✔ Password Aging – Indication of how long the password can be used before it expires

✔ Password usage – Indication of how many times the same password can be used

✔ Password storage – A method of storing a password in an encrypted manner

✔ Implementing Password Policies

✔ How to implement password policy depends on whether or not DBMS provides functions that support password security

✔ ORACLE has invested heavily in providing mechanism to enforce security , including implementation of password policies.

✔ Whereas a Microsoft SQL Server depends on the OS to implement password policies.



✔ Password Policies in ORACLE

CREATE PROFILE PASSWORD \_POLICY LIMIT

{ {

|PASSWORD\_LIFE\_TIME 365

|PASSWORD\_GRACE\_TIME 10

|PASSWORD\_REUSE\_TIME UNLIMITED

|PASSWORD\_REUSE\_MAX 0

|FAILED\_LOGIN\_ATTEMPTS 3

|PASSWORD\_LOCK\_TIME UNLIMITED;

}

{ expr | UNLIMITED | DEFAULT }

|PASSWORD\_VERYFY\_FUNCTION

{function | NULL | DEFAULT }

}



✔ Oracle password security profile parameters

✔ Here are the password security parameters:

* **failed\_login\_attempts** - This is the number of failed login attempts before locking the Oracle user account. The default in 11g is 10 failed attempts.
* **password\_grace\_time** - This is the grace period after the password\_life\_time limit is exceeded.
* **password\_life\_time** - This is how long an existing password is valid. The default in 11g forces a password change every 180 days.
* **password\_lock\_time** - This is the number of days that must pass after an account is locked before it is unlocked. It specifies how long to lock the

account after the failed login attempts is met. The default in 11g is one day.

* **password\_reuse\_max** - This is the number of times that you may reuse a password and

is intended to prevent repeating password cycles (north, south, east,

west).

* **password\_reuse\_time** - This parameter specifies a time limit before a previous

password can be re-entered. To allow unlimited use of previously used passwords, set *password\_reuse\_time* to UNLIMITED.

* **password\_verify\_function** - This allows you to specify the name of a custom password verification function.

✔ Profile creation using ORACLE Enterprise Manager Security Tools



Password Policies in SQL Server

✔ Microsoft SQL Server 2000 as a stand-alone product, does not provide for password policy enforcement when logging on a SQL Server

✔ Microsoft architecture follows a model known as an Integrated Server System.

✔ In this method all the server applications and the resources they provide are tightly integrated with the Windows server system and its security architecture.

✔ Password policy enforcement in a SQL Server environment handled by implementing SQL server in Windows authentication mode and applying polices within the Windows Server System

✔ There are two authentication protocols supported by Windows

* NTLM (Network LAN Manager)
* Kerberos 5

## Designing and Implementing password policies …

#### NTLM

✔ NTLM authenticates using a challenge / response methodology

✔ When the user attempt to access a resource , the server hosting the resource “challenges” , user to prove his / her identity.

✔ User then issue a “response” to that challenge

✔ If the response is correct then the user is authenticated to the server.

✔ The server goes through an authorization process for the requested resource.

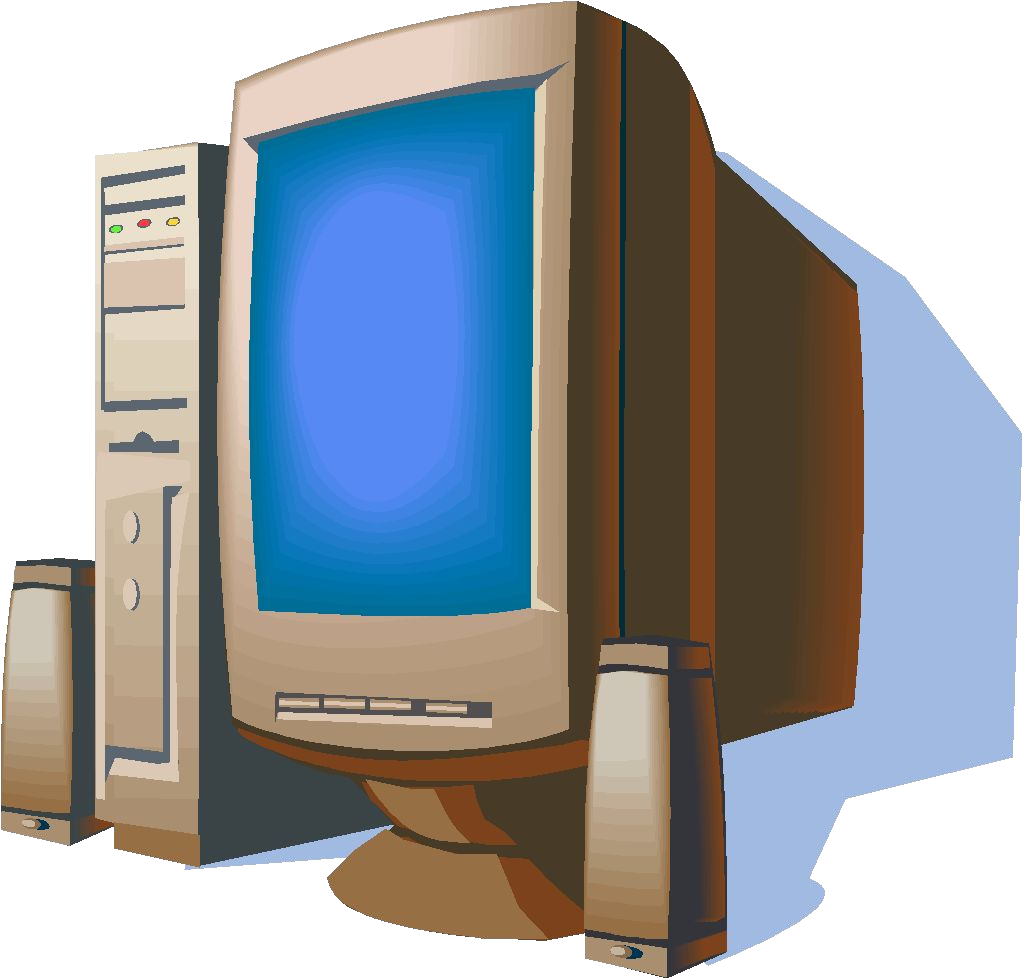
**Designing and Implementing password policies …**

✔ Authentication process consists of three messages

✔ Message 1 : Sent from the client to the server and is the initial request for authentication

✔ Message 2 : Sent from the server to client, contains challenge ( Eight bytes of Random Data)

✔ Message 3 : Sent from client to server , contains response to the challenge



Workstation

Server

Message 1

Message 2

Message 3

✔ The response is a 24-byte DES encrypted hash of the 8 byte challenge that can be decrypted only by a set of DES keys created using the user’s password.

✔ The benefit to NTLM is that password are verified without ever actually sending the password across the Web

**Designing and Implementing password policies …**

Kerberos

✔ Kerberos authentication differs from NTLM in many ways.

✔ Instead of using password encrypt / decrypt challenge / response messages, a secret key, known only to the server and client and also unique to the session, used to encrypt the handshake data.

✔ This allows not only for the server to validate the authenticity of client , but for the client to validate the authenticity of the server.

✔ This is an important difference and is one the reason Kerberos is more secure than NTLM

✔ Kerberos authentication requires a trusted third resource known as Key Distribution Center (KDC).

✔ The KDC generates the secret key for each session established.

✔ The new session ticket , containing the new key, has a time-out value associated with it.

✔ Once the secret key is obtained from the KDC

* + The client encrypts its request for a resource with the secret key.
  + The server decrypts the message using the same key, decrypts just on time stamp on the message and send back to client.
  + This tells the server and the client has the same key for the session which is established.



The following figures explain the authentication process in Kerberos



Workstation

Server

Clients wants to access a Server

KDC issues key : Kclient {Scs for Server} , ticket = Kserver {Scs for Client}

KDC generates a key and issues a session ticket to the client



Workstation

Server

Scs { Client Credentials , time}, ticket = Kserver { Scs for Client }

Scs { time }

Client sends authentication proof to the server

✔ Privilege is a method to permit or deny access to data or to perform database operations (Data Manipulation)

✔ Privileges in ORACLE

* + System Privileges – Privileges granted only by DBA or users who have been granted the administration option.
  + Object Privileges – Privileges granted to an ORACLE user by the scheme owner of a database object or a user who has been granted

the GRANT option.

✔ System Privileges :

privileges in

There are more than 100

system

some privileges

important frequently used

ORACLE , these are

* CREATE USER
* CREATE SESSION
* CREATE ROLE
* CREATE PROCEDURE
* CREATE TRIGGER
* CREATE TABLESPACE
* CREATE TYPE
* CREATE DATABASE LINK
* CREATE TABLE
* CREATE VIEW
* CREATE SEQUENCE
* DROP VIEW
* DROP USER
* DRO P TABLE

✔ Object Privileges: All DML are come

into

object privileges

* INSERT
* UPDATE
* DELETE
* SELECT
* INDEX
* REFERENCES



SQL GRANT Command

SQL GRANT is a command used to provide access or privileges on the database objects to the users.

✔ The Syntax for the GRANT command is:

GRANT privilege\_name ON object\_name TO {user\_name |PUBLIC |role\_name} [WITH GRANT OPTION];

✔ *privilege\_name* is the access right or privilege granted to the user. Some of the access rights are ALL, EXECUTE, and SELECT.

✔ *object\_name* is the name of an database object like TABLE, VIEW, STORED PROC and SEQUENCE.

✔ *user\_name* is the name of the user to whom an access right is being granted.

✔ *PUBLIC* is used to grant access rights to all users.

✔ *ROLES* are a set of privileges grouped together.

✔ *WITH GRANT OPTION* - allows a user to grant access rights to other users.

Eaxmple :

SQL > Grant select on emp to bmnantha; Grant succeeded

The schema owner of emp object gave select privilege to user bmnantha

SQL REVOKE Command:

The REVOKE command removes user access rights or privileges to the database objects.

✔ The Syntax for the REVOKE command is:

REVOKE privilege\_name ON object\_name FROM {user\_name |PUBLIC |role\_name}

✔ Example :

SQL > Revoke select on emp from bmnantha; Revoke succeeded

The schema owner of emp object get back the select privilege to user bmnantha

Privileges in SQL Server

✔ SQL Server has four levels of permissions

* System or Server level
* Database level
* Table (Object) level
* Column level

✔ Note : It is important to note that having server or database level permission doesn’t mean you have access to subordinate objects.



Privileges in SQL Server

#### Server Privileges

✔ Sysadmin – Can perform any function within the system

✔ Serveradmin – Can perform certain server-level functions.

✔ Setupadmin – Can manage linked servers and startup procedures

✔ Securityadmin – Can manage logons, change passwords

✔ Processadmin – Can manage processes running

✔ Dbcreator – Create, Alter and Drop Databases

✔ Diskadmin – Can manage the disk files for the server and database

✔ Bulkadmin – Can insert bulk insert operations

### Privileges in SQL Server

Database Privileges – Fixed Database Roles

✔ db\_owner – Have complete access to the database

✔ db\_accessadmin – Can add or remove users

✔ db\_securityadmin – Can change all permissions, object ownership, roles and role membership

✔ db\_ddladmin – Can execute all DDL statements

✔ db\_backupoperator – Can execute DBCC statements ( DBCC is a SQL Server tool used for DB performance)

✔ db\_datareader – Can issue SELECT and READTEXT statements

✔ db\_datawriter – Can issue INSERT, UPDATE, DELETE and UPDATENEXT statements

✔ db\_denydatareader – Explicitly denied SELECT and READTEXT statements

✔ db\_denydatawriter – Explicitly denied INSERT, UPDATE, DELETE and UPDATENEXT statements

### Privileges in SQL Server

Database Privileges – Statement permissions

✔ CREATE TABLE

✔ CREATE VIEW

✔ CREATE PROCEDURE

✔ CREATE FUNCTION

✔ CREATE DEFAULT

✔ CREATE ROLE

✔ BACKUP DATABASE

✔ BACKUP LOG

#### Privileges in SQL Server

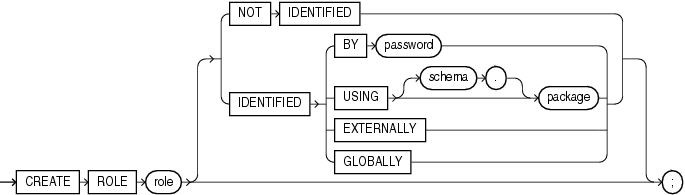
Table and Database Objects privileges and Column level privileges

✔ Same as ORACLE Grant and Revoke command.

✔ Refer Slide numbers : 68 and 69

## Creating , Assigning and Revoking User Roles

Creating role with ORACLE



✔ **NOT IDENTIFIED Clause -** Specify NOT IDENTIFIED to indicate that this role is authorized by the database and that no password is required to

enable the role.

✔ **IDENTIFIED Clause -** Use the IDENTIFIED clause to indicate that a user must be authorized by the specified method before the role is enabled with

the SET ROLE statement.

Creating role with ORACLE – Example

✔ The following statement creates the role dw\_manager:

CREATE ROLE dw\_manager;

* Users who are subsequently granted the dw\_manager role will inherit all of the privileges that have been granted to this role.

✔ You can add a layer of security to roles by specifying a password, as in the following example:

CREATE ROLE dw\_manager IDENTIFIED BY warehouse;

* Users who are subsequently granted the dw\_manager role must specify the password warehouse to enable the role with the SET ROLE statement.

✔ The following statement creates global role warehouse\_user:

CREATE ROLE warehouse\_user IDENTIFIED

✔ The followGiLngOsBtaAteLmLeYnt;creates the same role as an external role:

CREATE ROLE warehouse\_user IDENTIFIED EXTERNALLY;

Assigning Role to User in ORACLE - Example

✔ To assign privileges to role issue the following statement

SQL > GRANT CREATE SESSION TO dw\_manager;

Grant succeeded

✔ To assign a role to a user (Ex: bm\_nantha) issue the following statement

SQL > GRANT dw\_manager to bm\_nantha;

Grant succeeded

Create Roles with SQL Server

✔ To create a new database role using Query Analyzer , execute the SP\_ADDROLE system stored procedure

sp\_addrole [ @rolename = ] ‘role’ [ , [ @ownername = ] ‘owner’ ]

@rolename – The name of the new role

@ownername – The owner of new role , default is dbo

✔ To add the role of “sales” to the database Northwind

use northwind

exec sp\_addrole ‘sales’

✔ To add the user bm\_nantha to the role sales

exec sp\_addrolemember ‘sales’ , ‘bm\_nantha’

**Dropping a Role in ORACLE**

✔ **Example :** To drop the role dw\_manager, issue the following statement

DROP ROLE dw\_manager;

###### Dropping a Role in SQL Server

✔ **Example :** To drop the user ‘bm\_nantha’ from the role sales, issue the following statement

use northwind

exec sp\_droprolemember ‘sales’ , ‘jason’

**Creating , Assigning and Revoking User Roles Best Practices**

✔ Never store passwords in plain text, make sure it is encrypted

✔ Change passwords frequently

✔ Make sure the passwords are complex

✔ Pick password that you can remember

✔ Use roles to control administer privileges

✔ Should report the compromise or loss of password security

✔ Should report to security any violation of company guidelines like roles, profiles, privileges, passwords, etc.,

✔ Never give / share the password

✔ Never give the password over the phone

✔ Never type your password in an e-mail

✔ Use Windows integrated security mode for securing SQL Server

✔ Use Kerberos

✔ When Configuring Policies:

Require complex passwords , Set an account lockout threshold Do not allow passwords to automatically reset , Expire end-user passwords , Enforce password history