Database Security and Auditing: Protecting Data Integrity and Accessibility

Chapter 4
Profiles, Password Policies, Privileges, and Roles

Objectives

- Define and use a *profile*
- Design and implement *password policies*
- Implement password policies in Oracle and SQL Server
- Grant and revoke user *privileges*
- Create, assign, and revoke user *roles*
- List best practices for securing a network environment

Defining and Using Profiles

- Profile:
  - Describes limitation of database resources
  - Defines database users behavior
  - Prevents users from wasting resources

Creating Profiles in Oracle

- Define two elements of security:
  - Restriction on resources
  - Implementation of password policies
- CREATE PROFILE statement
- To view all created profiles, query the data dictionary view DBA_PROFILES
- Resource Manager tool: creates different CPU usage policies

Create a Profile

```
SQL> CREATE PROFILE CH04_PROF
     2  LIMIT
     3          SESSION_PER_USER                default
     4          CPU_PER_SESSION                 default
     5          CPU_PER_CALL                    1000
     6          CONNECT_TIME                    120
     7          IDLE_TIME                       15
     8          LOGICAL_READS_PER_SESSION       default
     9          LOGICAL_READS_PER_CALL          default
     10         COMPOSITE_LIMIT                 default
     11         PRIVATE_SGA                     default
     12  /
```

View a Profile

```
SQL> SELECT * FROM DBA_PROFILES
2       WHERE PROFILE = 'CH04_PROF'
3       /
```

Creating Profiles in Oracle (continued)

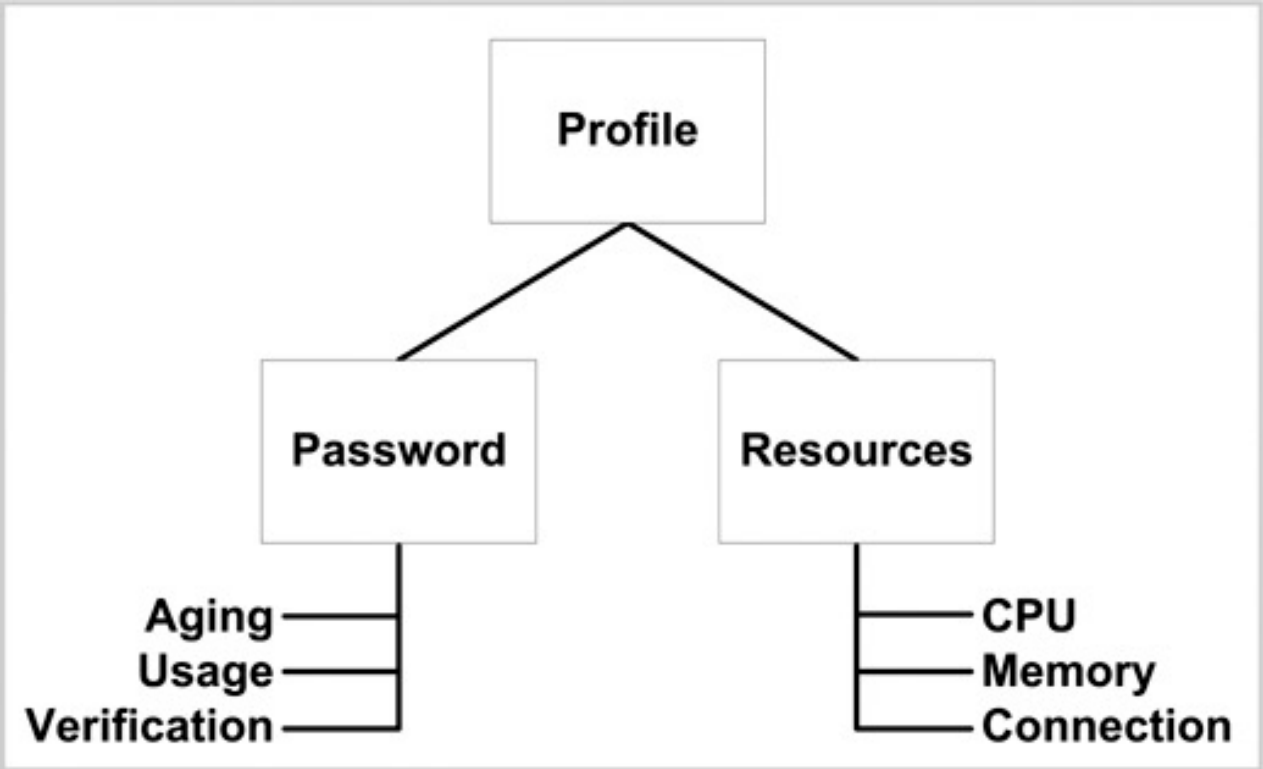

FIGURE 4-1 Profile feature in Oracle

Creating Profiles in Oracle (continued)

- ALTER PROFILE: *modifies a limit for a profile*
- ALTER USER: *assigns a profile to a user*
- Oracle Enterprise Manager Security Tool: view all details about *users* and *profiles* in a GUI

Alter/Select Profiles

```
//modify a limit for a profile
SQL> ALTER PROFILE CH04_PROF
2       LIMIT  IDLE_TIME        30
3       /

//assign a profile to a user
SQL> ALTER USER <username>  PROFILE CH04_PROF
2       /
//return username and profile whose username begins with character 's'
SQL> SELECT USERNAME, PROFILE
2       FROM DBA_USERS
3       WHERE USERNAME LIKE  'S%'
4       /
//return all rows whose profile is default
SQL> SELECT * FROM DBA_PROFILES
2       WHERE  PROFILE = 'DEFAULT'
3       /
```

Creating Profiles in Oracle (continued)

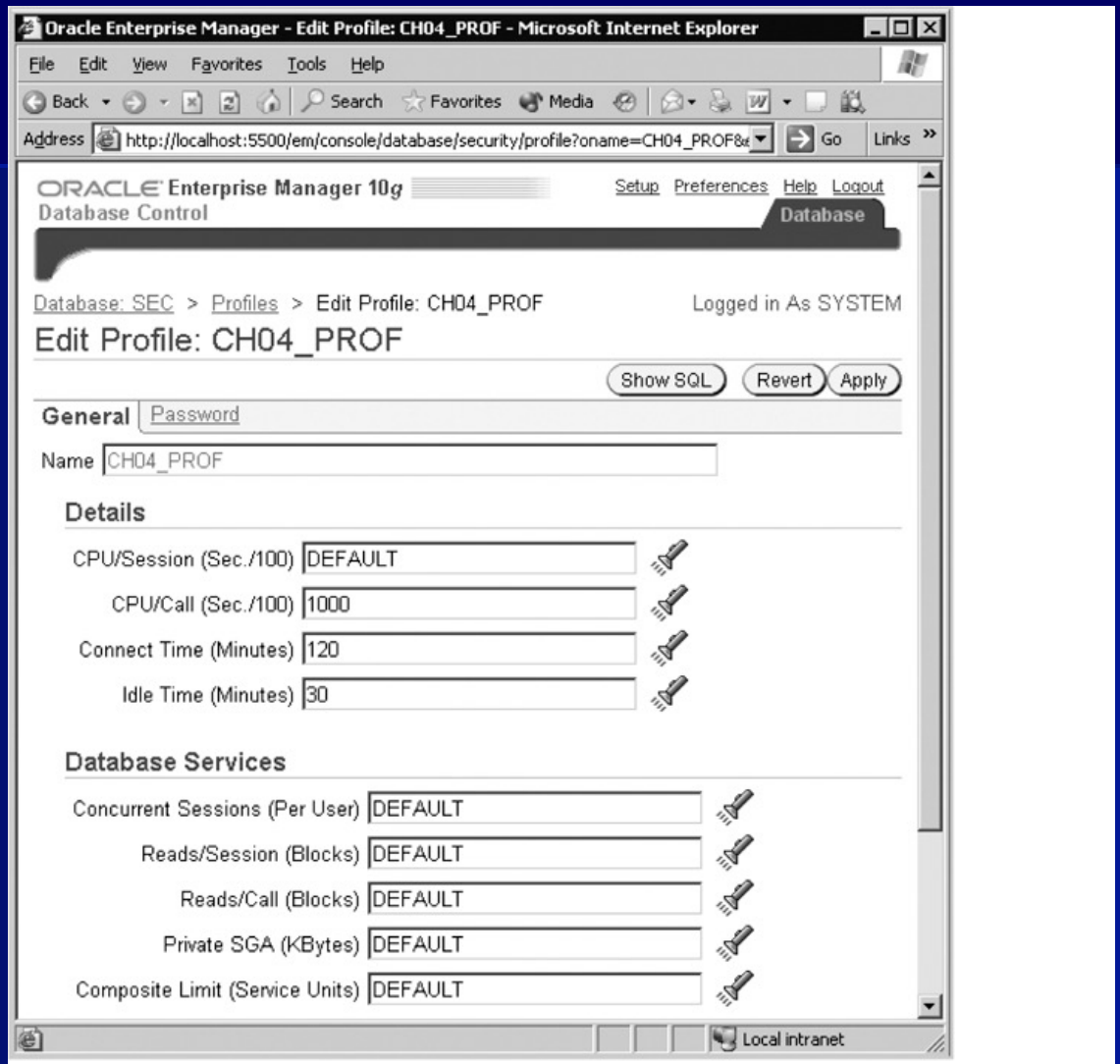

FIGURE 4-2 Oracle Enterprise Manager showing profile CH04_PROF through the Security Manager tool

Designing and Implementing Password Policies

- Password is the key to open a user account; strong passwords are harder to break
- User authentication depends on passwords
- Hacker violations begin with breaking a password
- Companies spend on:
  - Training
  - Education

What Is a Password Policy?

- Set of guidelines:
  - Enhances the robustness of a password
  - Reduces the likelihood of password breaking
- Deals with:
  - Complexity
  - Change frequency
  - Reuse

Importance of Password Policies

- First line of defense
- Most companies invest considerable resources to strengthen authentication by adopting technological measures that protect their assets
- Forces employees to abide by the guidelines set by the company and raises employee awareness of password protection
- Helps ensure that a company does not fail audits

Designing Password Policies

- Complexity: set of guidelines for creating passwords
- Aging: how long a password can be used
- Usage: how many times a password can be used
- Storage: storing a password in an encrypted manner

Implementing Password Policies

- Oracle using profiles:
  - CREATE PROFILE
  - Oracle Enterprise Manager
  - PASSWORD_VERIFY_FUNCTION

Create Password Profile

```
CREATE PROFILE PASSWORD_PROFILE
LIMIT
{    {   FAILED_LOGIN_ATTEMPTS
     |   PASSWORD_LIFE_TIME
     |   PASSWORD_REUSE_TIME
     |   PASSWORD_REUSE_MAX
     |    PASSWORD_LOCK_TIME
     |    PASSWORD_GRACE_TIME
     }
     {expr | UNLIMITED | DEFAULT }
| PASSWORD_VERIFY_FUNCTION
     {   function | NULL | DEFAULT }
}
```

Create a Password Profile

```
SQL>CREATE PROFILE
    ACME_PASSWORD_PROFILE
2   LIMIT
3   FAILED_LOGIN_ATTEMPTS      1
4   PASSWORD_LIFE_TIME         15
5   PASSWORD_REUSE_TIME        DEFAULT
6   PASSWORD_REUSE_MAX         1
7   /
Profile created
```

Create verify_function for Password Complexity

CREATE OR REPLACE FUNCTION
verify_function(username varchar2, password varchar2, old_password verchar2) RETURN Bollean;

Page 110 example: password complexity requires the password to be 10 characters and cannot all be digits.

Implementing Password Policies (continued)

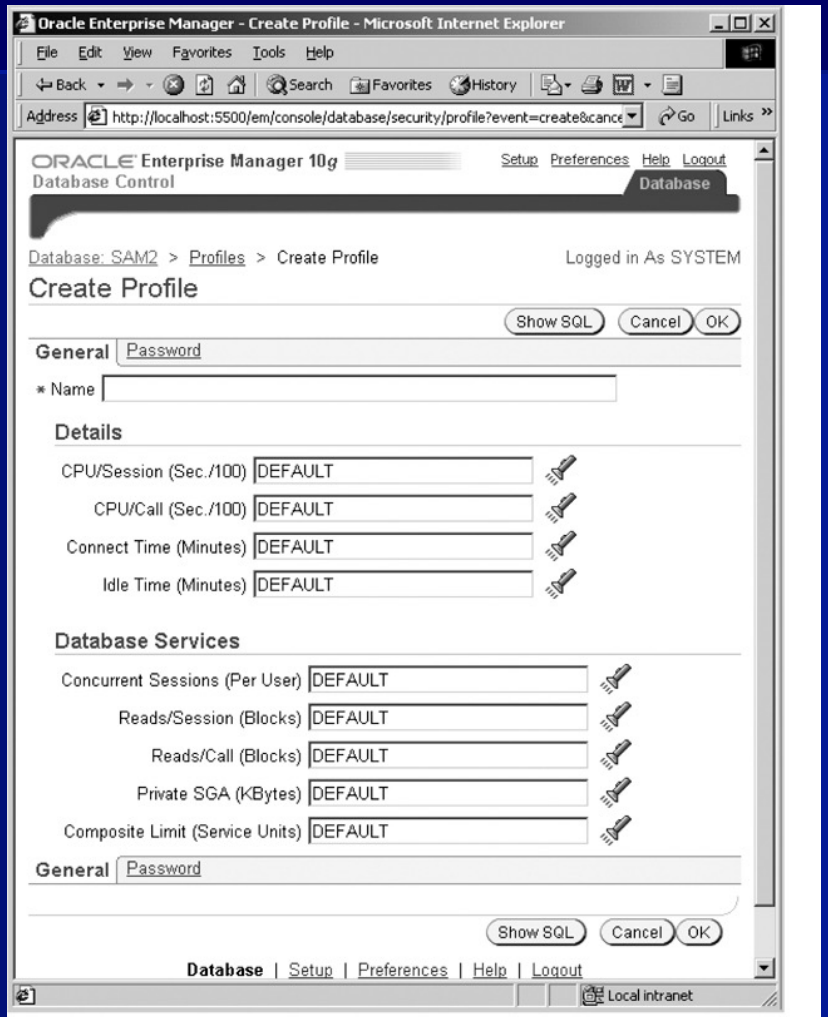

FIGURE 4-3 Profile creation using the Oracle Enterprise Manager Security tool

Implementing Password Policies (continued)

- Microsoft SQL Server:
  - Integrated server system
  - Windows authentication mode
- NTLM:
  - Used to authenticate local user, not domain user
  - Challenge/response methodology
  - Challenge is eight bytes of random data
  - Response is a 24-byte DES-encrypted hash

Implementing Password Policies (continued)

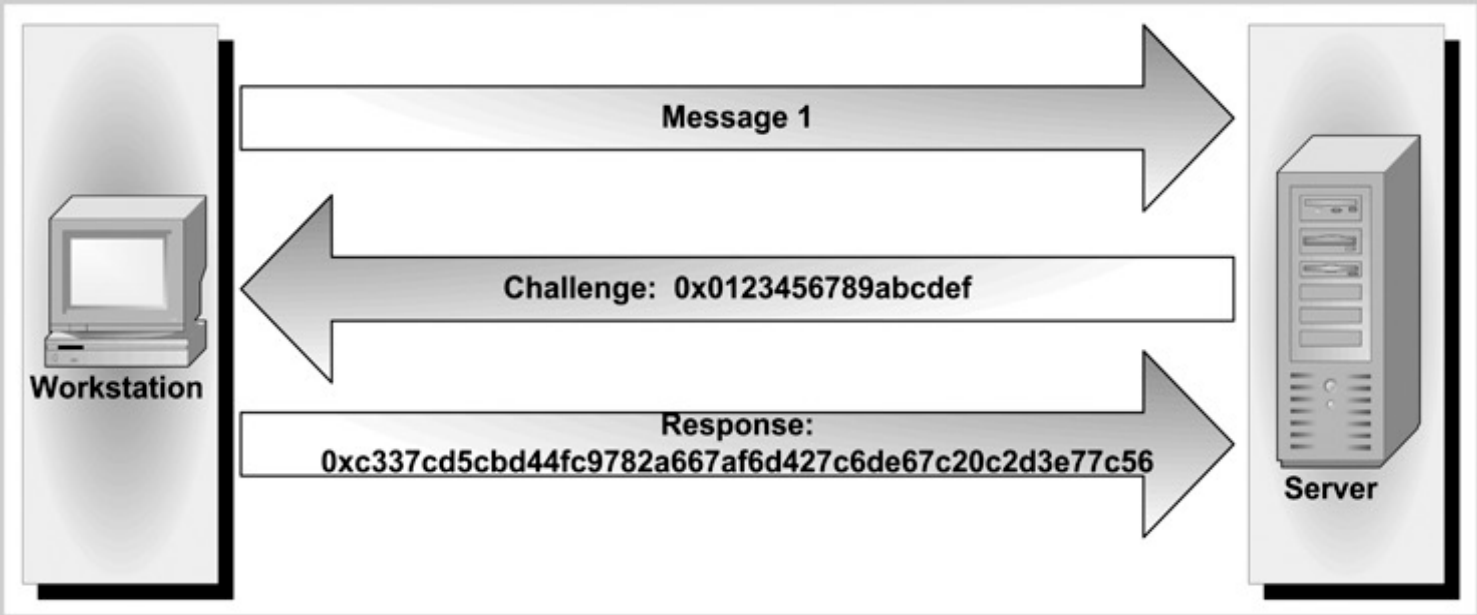

FIGURE 4-4 NTLM authentication

Implementing Password Policies (continued)

- Kerberos:
  - A key known by client and server encrypts handshake data
  - Requires a Key Distribution Center (KDC)
  - Tickets
  - Time must be synchronized networkwide

Implementing Password Policies (continued)

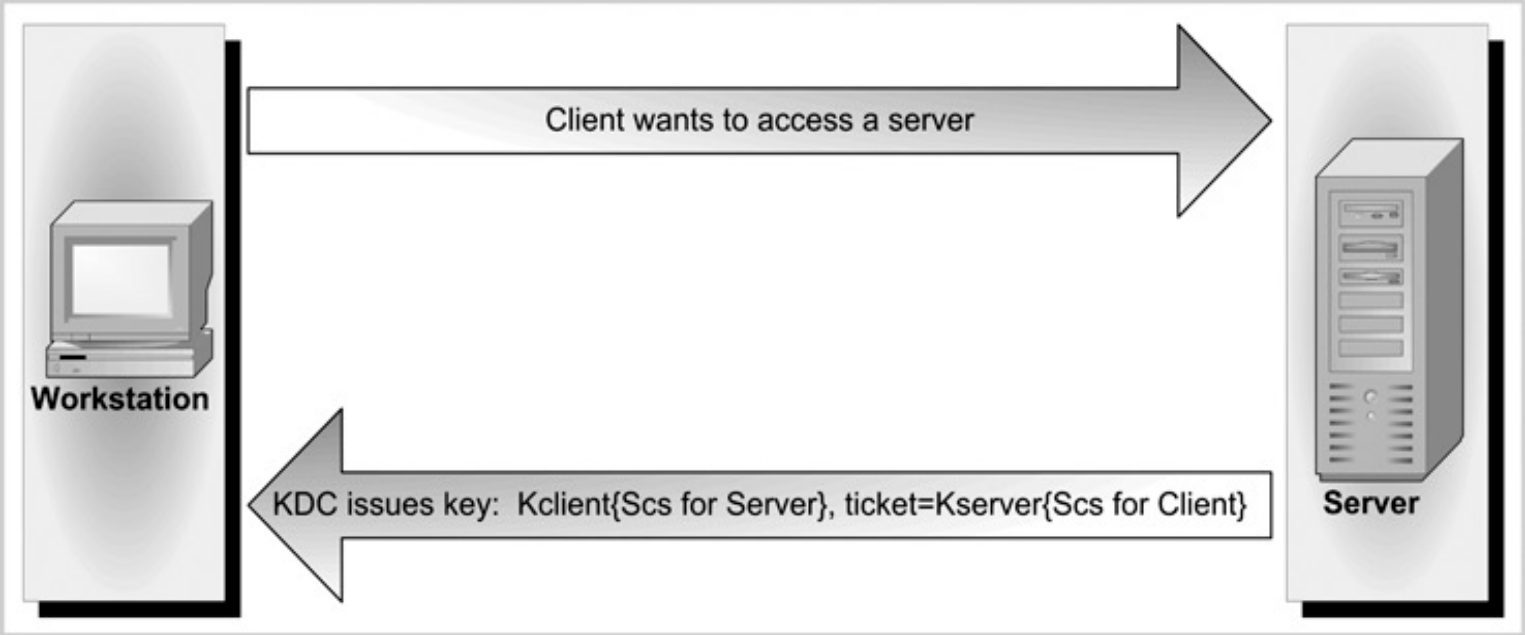

FIGURE 4-5 KDC generates a key and issues a session ticket to the client

Implementing Password Policies (continued)

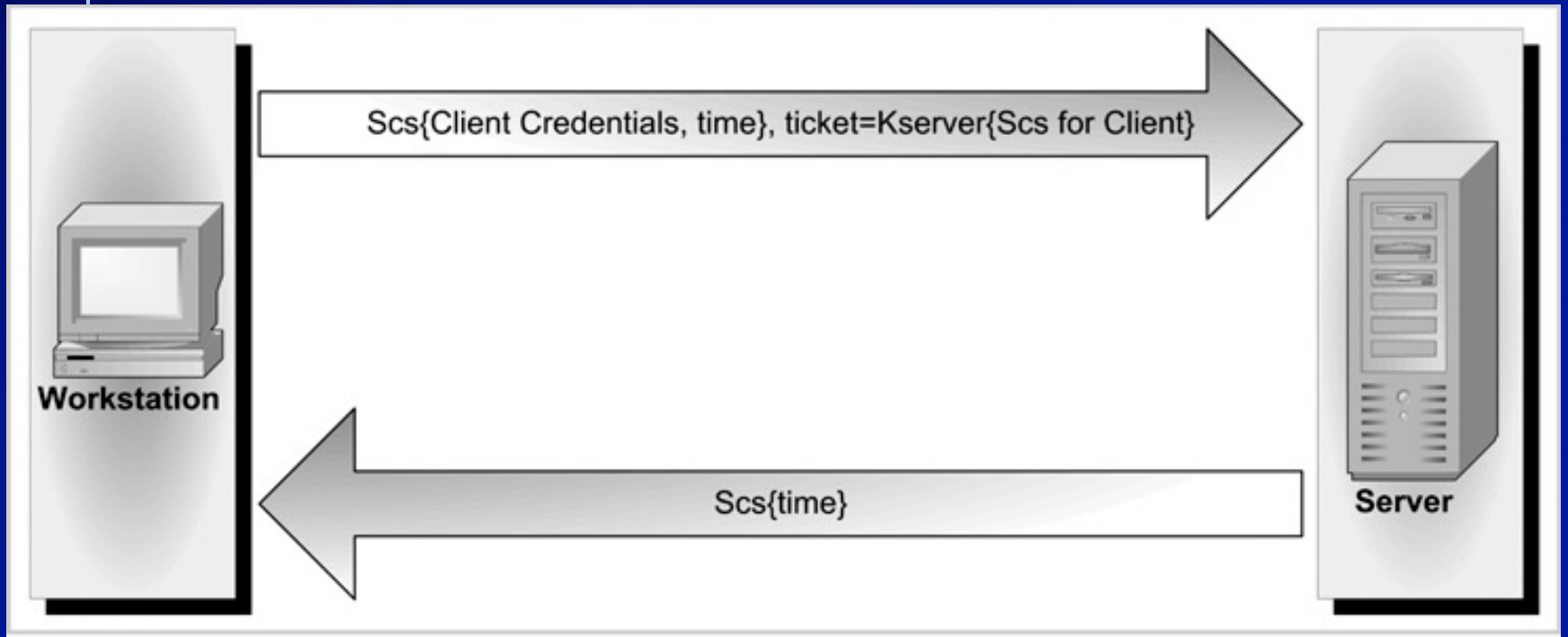

FIGURE 4-6 Client sends authentication proof to the server

Lab -- Setting Password Policies

- Local and domain policies are identical
- Start→ all programs → administrative tools → Local Security Policy → Account Policies → Password Policy
- Account lockout policy
  - Account lockout duration
  - Account lockout threshold
  - Reset account lockout counter after

Password Policy Selection

Policy	Description
Enforce password history	Indicates that when users change passwords, the new password must be different from the last n passwords
Maximum password age	Indicates how many days must pass before a new password expires and must be changed
Minimum password length	indicates that a user's password must be at least n characters in length
Password must meet complexity requirements	Indicates whether or not a password must meet a predetermined level of complexity, e.g., it must use mixed case (capital and noncapital letters) and must contain one or more letters, numbers, and symbols
Store passwords using reversible encryption	Indicates whether or not to store the password as a hash that can be decrypted.

Granting and Revoking User Privileges

- Permit or deny access to data or to perform database operations
- In Oracle:
  - System privileges:
    - Granted only by a database administrator
    - Granted by a user with administration privileges
  - Object privileges:
    - Granted to a user by the schema owner
    - Granted by a user with GRANT privileges

Granting and Revoking User Privileges (continued)

- In Oracle (continued):
  - Grant a privilege using the Data Control Language (DCL) GRANT statement
  - Revoke a privilege using the DCL REVOKE statement:
    - ADMIN option
    - GRANT option
  - Oracle Enterprise Manager Security

Granting and Revoking User Privileges (continued)

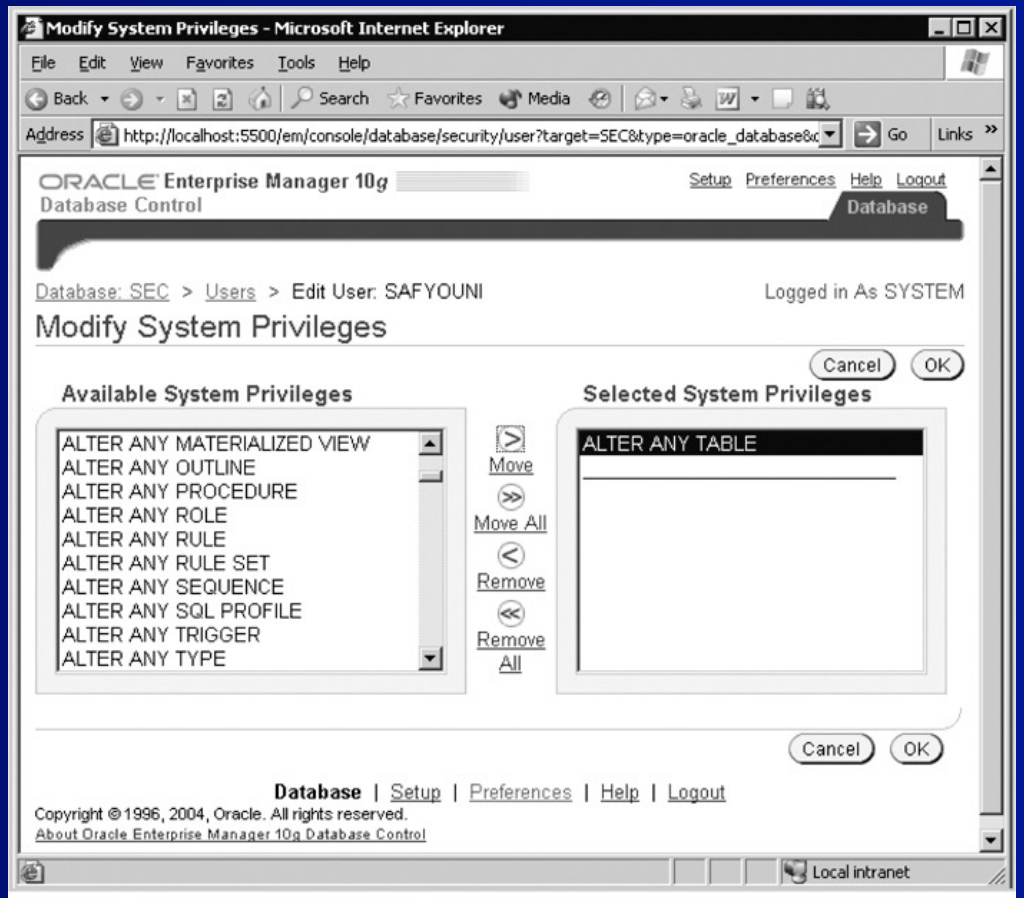

FIGURE 4-13 Granting a system privilege to a user

Granting and Revoking User Privileges (continued)

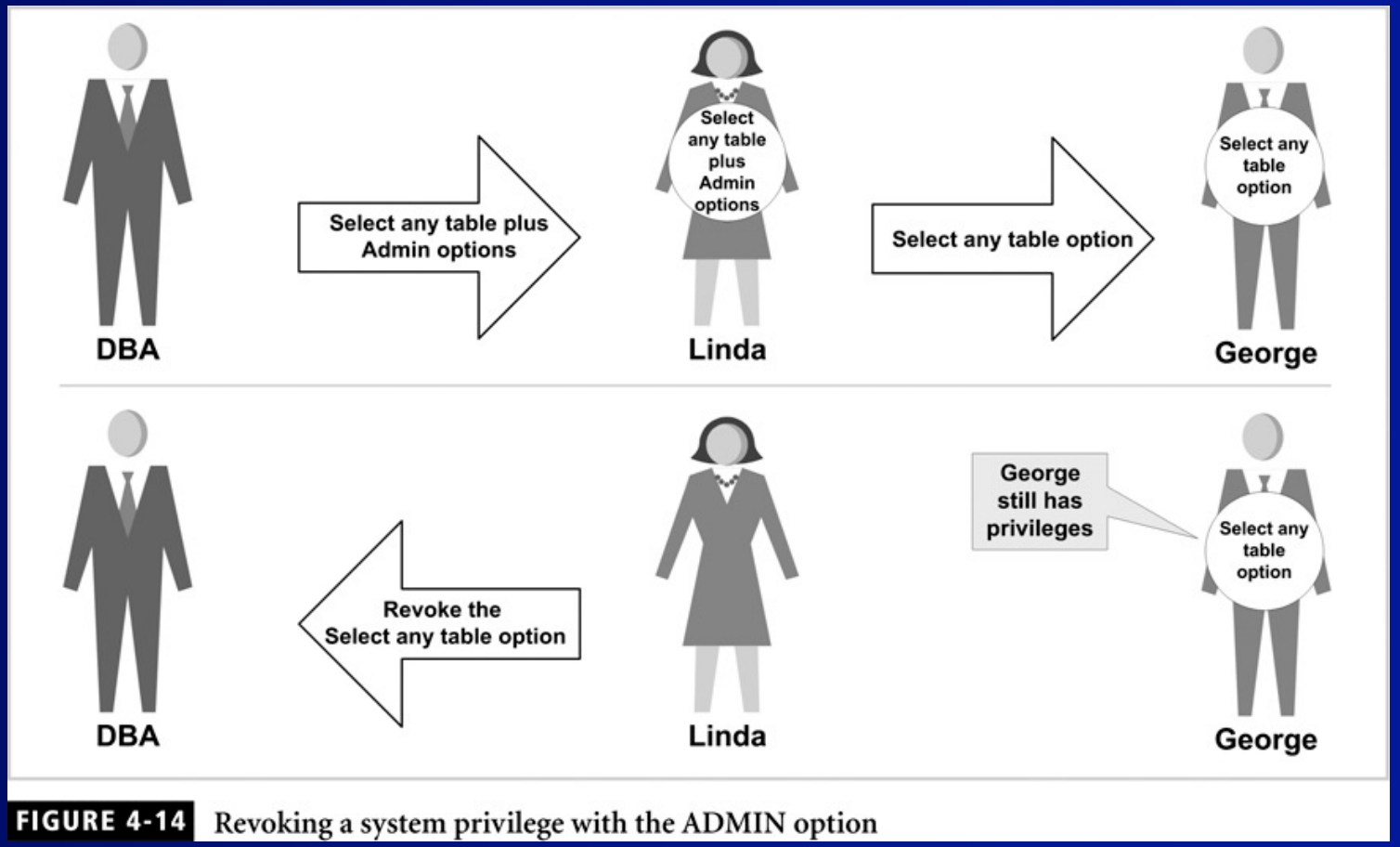

FIGURE 4-14 Revoking a system privilege with the ADMIN option

Granting and Revoking User Privileges (continued)

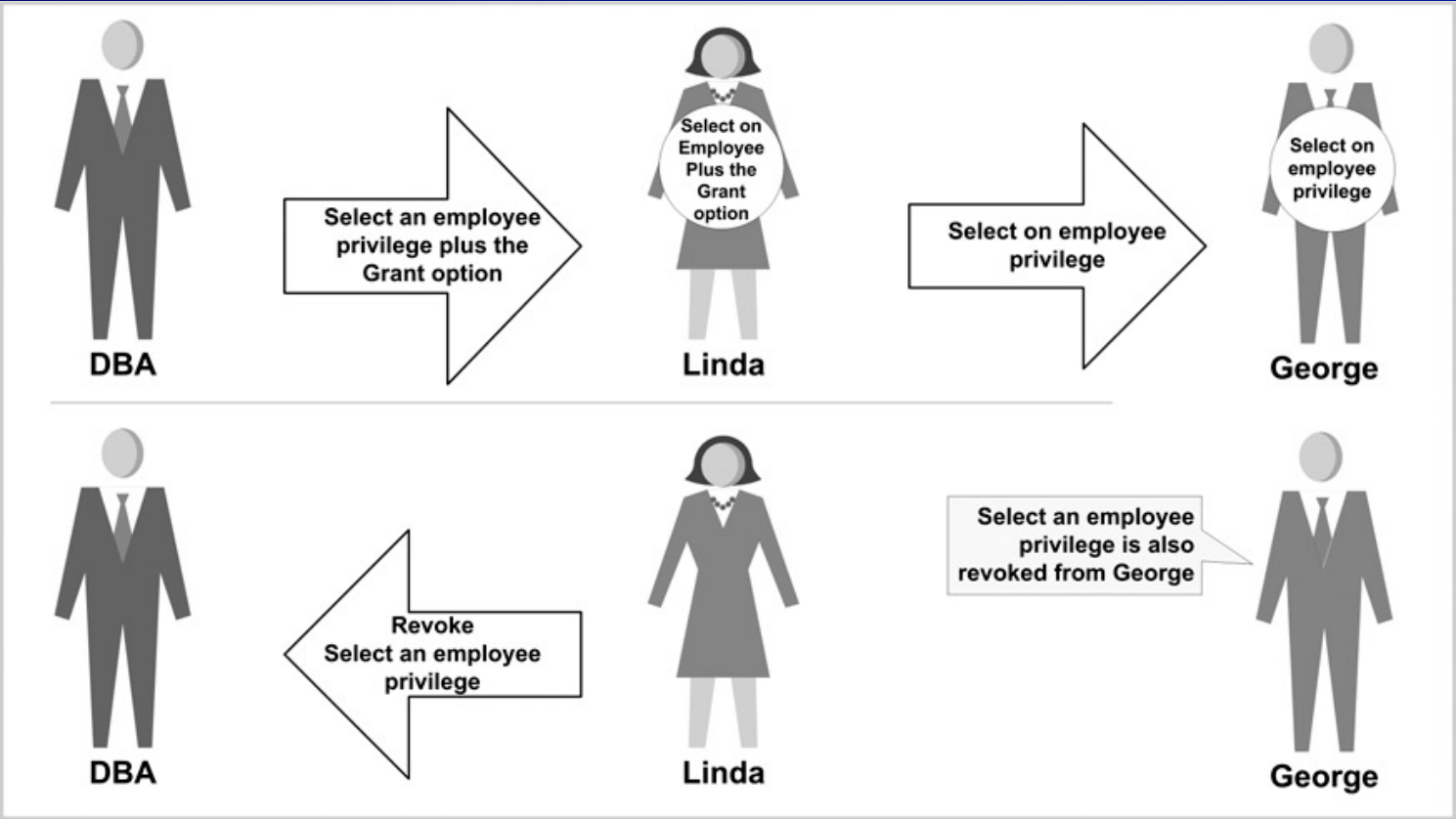

FIGURE 4-16 Revoking an object privilege with the GRANT option

Creating, Assigning, and Revoking User Roles

- Role:
  - Used to organize and administer privileges
  - It is like a user, except it cannot own object
  - Can be assigned privileges
  - Can be assigned to users

Creating, Assigning, and Revoking User Roles (continued)

- In Oracle:
  - Create a role using CREATE ROLE statement
  - Assign a role using GRANT statement
  - Oracle Enterprise Manager Roles tool
  - Revoke a role using REVOKE statement
  - Drop a role using DROP statement

Create and Assign Role

```
SQL> CREATE ROLE DEV_ROLE;
        GRANT CREATE SESSION TO DEV_ROLE
        GRANT DEV_ROLE TO YANG
```

Best Practices

- Develop a secure environment:
  - Never store passwords for an application in plaintext
  - Change passwords frequently
  - Use passwords at least eight characters long
  - Pick a password that you can remember
  - Use roles to control and administer privileges
  - Report compromise or loss of a password
  - Report any violation of company guidelines

Best Practices (continued)

- Develop a secure environment (continued):
  - Never give your password to anyone
  - Never share your password with anyone
  - Never give your password over the phone.
  - Never type your password in an e-mail
  - Make sure your password is complex enough
  - Use Windows integrated security mode

Best Practices (continued)

- When configuring policies:
  - Require complex passwords with special characters in the first seven bytes
  - Require a password length of at least eight
  - Set an account lockout threshold
  - Do not allow passwords to automatically reset
  - Expire end-user passwords
  - Do not expire application-user passwords
  - Enforce a password history

Summary

- Profiles define database users behavior
- In Oracle:
  - DBA_PROFILE view
  - ALTER USER
- Password policy:
  - Enhances password robustness
  - Reduces likelihood of password breaking

Summary (continued)

- In SQL Server:
  - NTLM
  - Kerberos
- In Oracle:
  - System privileges
  - Object privileges
- In SQL Server:
  - System or server, database, table and column privileges

Summary (continued)

- GRANT and REVOKE
- Role is used to:
  - Organize and administer privileges in an easy manner
  - Role is like a user but cannot own objects
  - Role can be assigned privileges
  - GRANT and REVOKE
- Best practices for developing a secure environment

Quick Quiz

- A _____ is a security concept that describes the limitation of database resources that are granted database users.
  a. role
  b. privilege
  c. profile
  d. password
- In Oracle, to view all profiles created in the database, query the data dictionary view, _____.
  a. DB_PROFILES
  b. DBA_PROFILES
  c. SYS_PROFILES
  d. DBMS_PROFILES
- A(n) _____ is a set of guidelines that enhances the robustness of a password and reduces the likelihood of its being broken.

Quick Quiz

- A _____ is a method to permit or deny access to data or to perform a database operation.
  - role
  - privilege
  - password policy
  - profile
- In Oracle you can grant a privilege by using the data control language (DCL) _____ statement.
- A _____ is a concept used to organize and administer privileges in an easy manner.
  - role
  - privilege
  - password policy
  - profile

Granting and Revoking User Privileges (continued)

- In SQL Server (continued):
  - Database privileges:
    - Fixed database roles
    - Statement permissions
  - Grant permission using the GRANT statement
  - Revoke permission using the REVOKE statement
  - Enterprise Manager
  - Deny permission using the DENY statement

Granting and Revoking User Privileges (continued)

Table 4-4 SQL Server statement permissions

Statement	Permits the User to
CREATE TABLE	Create tables in the database
CREATE VIEW	Create views in the database
CREATE PROCEDURE	Create stored procedures in the database
CREATE FUNCTION	Create functions in the database
CREATE DEFAULT	Create defaults in the database
CREATE RULE	Create rules in the database
BACKUP DATABASE	Back up the database
BACKUP LOG	Back up the database transaction log(s)

Granting and Revoking User Privileges (continued)

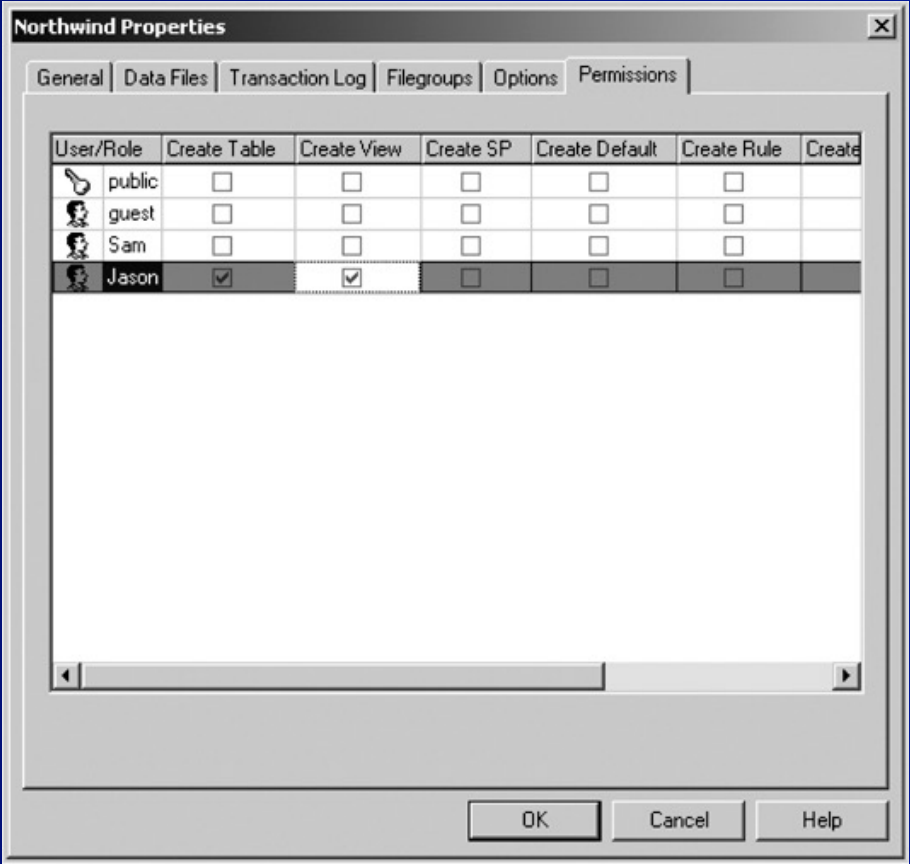

FIGURE 4-17 Granting user permissions

Granting and Revoking User Privileges (continued)

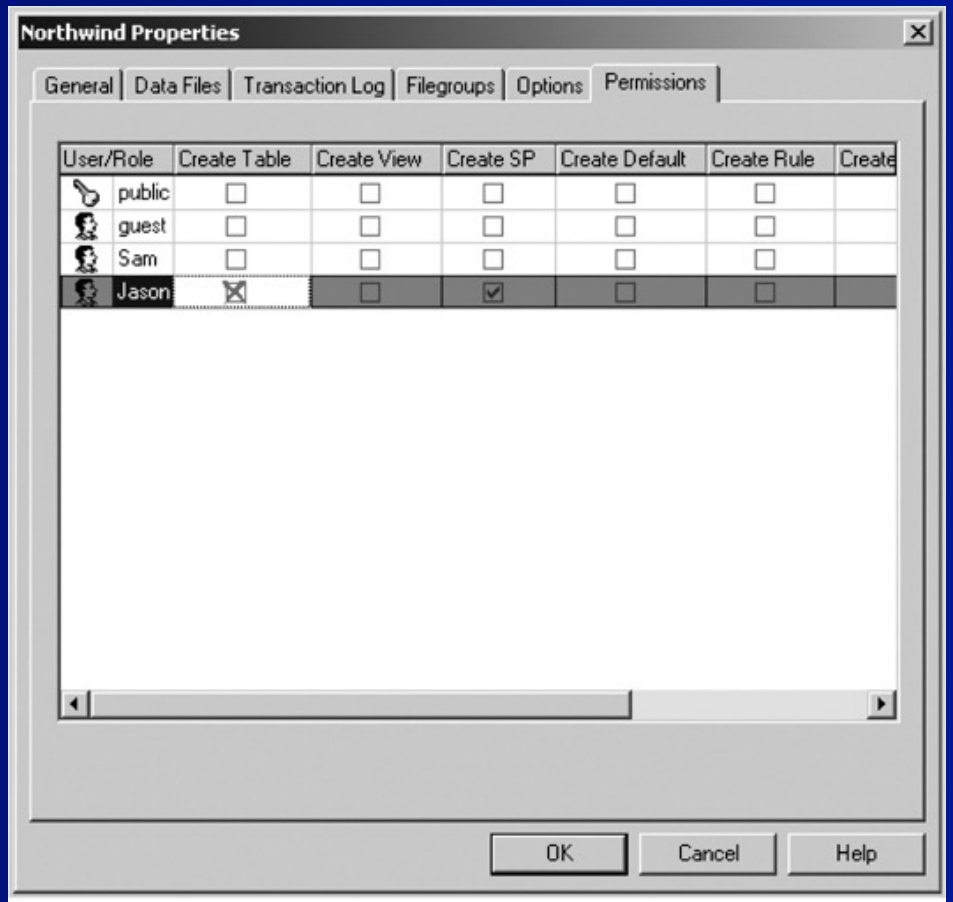

FIGURE 4-19 Revoking database permissions with SQL Server

Granting and Revoking User Privileges (continued)

- In SQL Server:
  - Table and database objects privileges:
    - GRANT, REVOKE, and DENY
    - EXECUTE permission
    - Enterprise Manager (3 methods)
  - Column privileges:
    - GRANT, REVOKE, and DENY
    - Enterprise Manager (2 methods)

Granting and Revoking User Privileges (continued)

- In SQL Server (4 levels); system/server privileges:
  - Sysadmin
  - Serveradmin
  - Setupadmin
  - Securityadmin
  - Processadmin
  - Dbcreator
  - Diskadmin
  - Bulkadmin
  - Page 129-130

Creating, Assigning, and Revoking User Roles (continued)

- In SQL Server; user-defined roles:
  - Standard and application
  - Create roles using SP_ADDROLE system-stored procedure
  - Add members to a role using SP_ADDROLEMEMBER stored procedure
  - Drop members from a role using SP_DROPROLEMEMBER stored procedure

Creating, Assigning, and Revoking User Roles (continued)

- In SQL Server (continued):
  - User-defined roles (continued):
    - Drop roles using SP_DROPROLE stored procedure
    - Use Enterprise Manager
  - Fixed server roles:
    - Cannot be modified or created
    - Add member to a role using SP_ADDSRVROLEMEMBER stored procedure

Lab – Manage User-Defined Roles

exec sp_addrole ‘sales’
exec sp_addrolemember ‘sales’, ‘jason’
exec sp_droprolemember ‘sales’, ‘jason’
exec sp_droprole ‘sales’

Enterprise Manager → roles → properties

Creating, Assigning, and Revoking User Roles (continued)

Table 4-5 Description of fixed server roles

SQL Server Fixed Role	Role Description
sysadmin	A super role that allows assigned user to perform any task within SQL Server
serveradmin	Allows assigned user to modify SQL Server configuration.
setupadmin	Allows assigned user to perform specific SQL Server setup, such as linking servers and execution of system stored procedures
securityadmin	Allows assigned user to administer SQL Server logins
processadmin	Allows assigned user to administer SQL Server instance processes
dbcreator	Allows assigned user to create and modify SQL Server database
diskadmin	Allows assigned user to administer database data files
bulkadmin	Allows assigned user to perform BULK INSERT statement

* Information presented in this table is adapted from Microsoft SQL Server 2000 documentation

Creating, Assigning, and Revoking User Roles (continued)

- In SQL Server (continued):
  - Fixed server roles (continued):
    - Drop members from a role using SP_DROPSRVROLEMEMBER stored procedure
    - Use Enterprise Manager
  - Fixed database roles:
    - Cannot be modified
    - Give access to database administrative tasks
    - Add members to a role using SP_ADDROLEMEMBER stored procedure

Creating, Assigning, and Revoking User Roles (continued)

Table 4-6 SQL Server built-in roles

Fixed Database Role	Description
db_owner	Has all permissions in the database
db_accessadmin	Can add or remove user IDs
db_securityadmin	Can manage all permissions, object ownerships, roles, and role memberships
db_ddladmin	Can issue all DDL, but cannot issue GRANT, REVOKE, or DENY statements
db_backupoperator	Can issue DBCC, CHECKPOINT, and BACKUP statements
db_datareader	Can select all data from any user table in the database
db_datawriter	Can modify any data in any user table in the database
db_denydatareader	Cannot select any data from any user table in the database
db_denydatawriter	Cannot modify any data in any user table in the database

* From Microsoft SQL Server Books Online

Creating, Assigning, and Revoking User Roles (continued)

- In SQL Server (continued):
  - Fixed database roles (continued):
    - Drop members from a role using SP_DROPROLEMEMBER stored procedure
    - Use Enterprise Manager
  - Public database role:
    - Cannot be dropped
    - Users automatically belong to this role
    - Users cannot be dropped

Lab -- Manage Fixed Server/Database Roles

Fixed server roles

exec sp_addsrvrolemember 'mydomain\jason', 'sysadmin'

exec sp_addsrvrolemember 'sam', 'securityadmin'

Fixed database roles

exec sp_addrolemember 'db_securityadmin', 'jason'

exec sp_droprolemember 'db_securityadmin', 'jason'

Enterprise Manager → roles → properties