Managing Roles

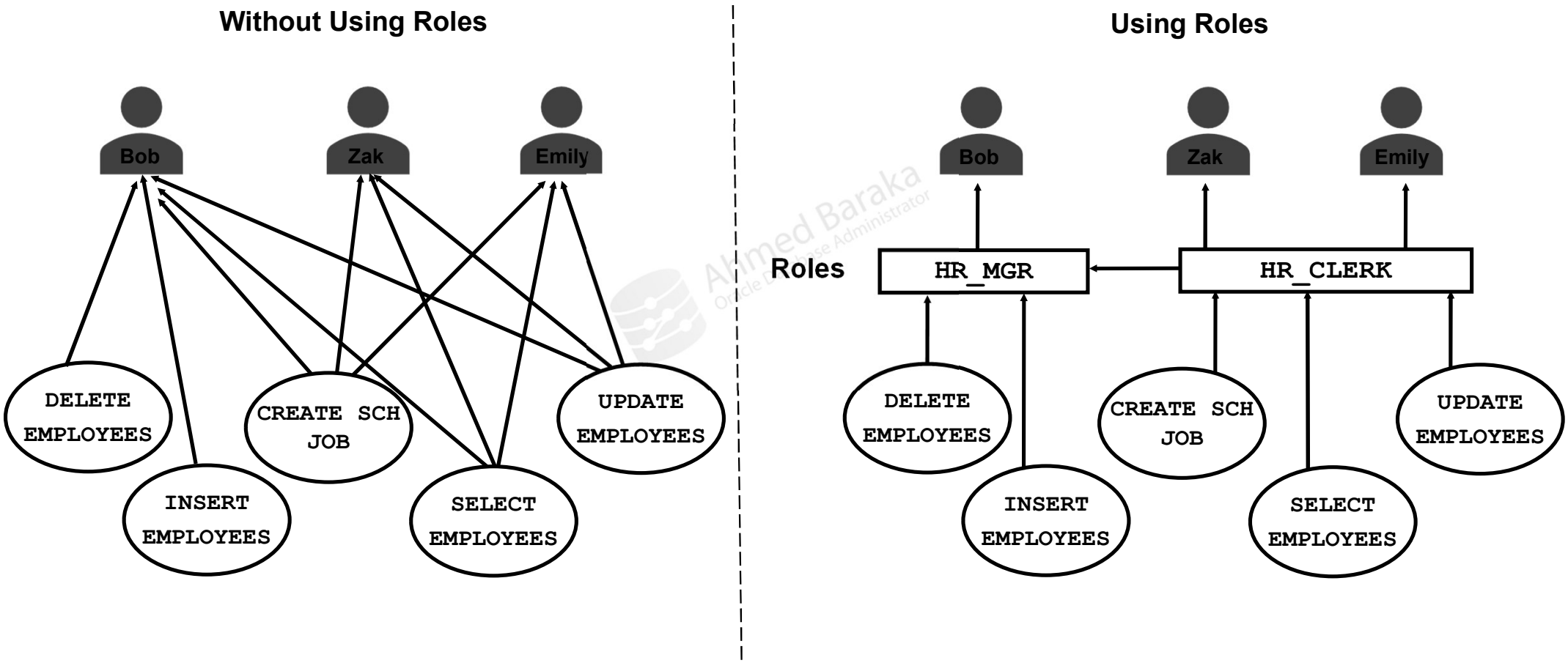
By Ahmed Baraka

Objectives

In this lecture, you will learn how to perform the following:

- Understand the purpose of using roles
- List predefined roles
- Create and use roles
- Use password-protected roles
- Understand using **`PUBLIC`** role
- Drop roles
- Obtain information about the granted privileges and roles
- Understand the guideline of using roles and privileges

Assigning Privileges to Roles and Assigning Roles to Users



About Roles

- A group of privileges that can be granted to a user or another role
- Benefits:
  - Easier privilege management
  - Dynamic privilege management
  - Selective availability of privileges
- Within a database, each role name must be unique and different from all user names
- They are not contained in any schema
- Multiple roles can be granted to the same user

Known Predefined Roles

Role	Description
CONNECT	CREATE SESSION
DBA	Most system privileges; several other roles. Do not grant to non-administrators.
RESOURCE	CREATE CLUSTER, CREATE INDEXTYPE, CREATE OPERATOR, CREATE PROCEDURE, CREATE SEQUENCE, CREATE TABLE, CREATE TRIGGER, CREATE TYPE
SCHEDULER_ADMIN	CREATE ANY JOB, CREATE EXTERNAL JOB, CREATE JOB, EXECUTE ANY CLASS, EXECUTE ANY PROGRAM, MANAGE SCHEDULER
SELECT_ CATALOG_ROLE	No system privileges; HS_ADMIN_ROLE and over 1,700 object privileges on the data dictionary

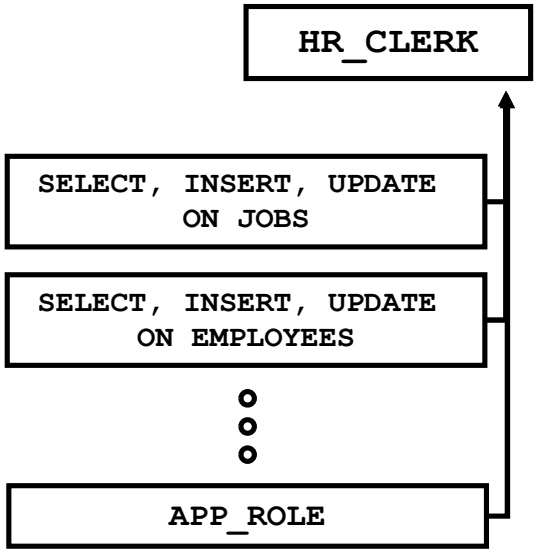
```
SELECT ROLE, ORACLE_MAINTAINED
FROM DBA_ROLES WHERE ORACLE_MAINTAINED='Y';
```

Creating Roles

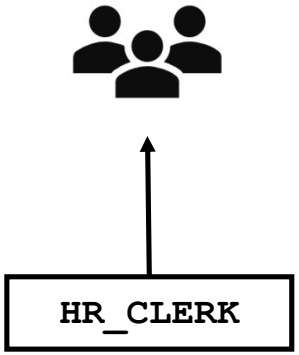
1 Create a role

non-authenticated or authenticated

2 Grant privileges to the role



3 Grant the role to the required users



Role Authorization Types

- Non-authorized
- Authorized:
  - by using the database (password)
  - by using an application
  - by using an external source
  - by using the operating system
  - by using a network client
  - a Global role authorized by an Enterprise Directory Service

Using Roles

1. Create a role:
   - Non-authorized:

```
CREATE ROLE <role name>;
```

   - Authenticate by a password:

```
CREATE ROLE <role name> IDENTIFIED BY <password>;
```

2. Grant privileges or roles to the role (multiple privileges allowed):

```
GRANT <privilege|role> TO <role name>;
```

3. Grant the role to the target users:

```
GRANT <role> TO <user name>;
```

Managing User Default Roles

- By default, any new role granted to a user is added to the user default roles
- A default role is automatically enabled for a user when the user creates a session.
- We cannot set default roles for a user in the **CREATE USER** statement
- To change the default roles for the user:

```
ALTER USER scott DEFAULT ROLE clerk_mgr, connect;
```

```
ALTER USER scott DEFAULT ROLE ALL EXCEPT clerk_mgr;
```

  - The roles must already be granted to the user

Protecting a Role with a Password

- Useful to make sure specific roles are enabled only from the application
- To use it:
  - Set a password for a role:

```
CREATE ROLE payroll IDENTIFIED BY ABcd##1234;
```

  - Exempt the role from the user default roles:

```
GRANT payroll TO scott;
ALTER USER scott DEFAULT ROLE ALL EXCEPT payroll;
```

- To enable a password protected role, from SQL or PL/SQL:

```
SET ROLE payroll IDENTIFIED BY ABcd##1234;
BEGIN
 DBMS_SESSION.SET_ROLE('payroll'||' identified by ABcd##1234');
END;
```

About the `PUBLIC` Role

- Is a special role that every database user account automatically has
- Is used when granting privileges to **all** the database users
- It does not appear in the **`DBA_ROLES`** and **`SESSION_ROLES`**
- Avoid granting privileges to **`PUBLIC`** unless you are certain it is safe to do so

Dropping a Role

- You must have the **DROP ANY ROLE** system privilege or have been granted the role with the **ADMIN** option.

```
DROP ROLE <role name>;
```

- Drop a user does not drop roles created by the user
- **PUBLIC** role cannot be dropped

User and Role Privilege Dictionary Views

View	Description
DBA_ROLES	describes all roles in the database (does not have `OWNER` column)
DBA_ROLE_PRIVS	describes the roles granted to all users and roles in the database
ROLE_ROLE_PRIVS	retrieves information about roles granted to the roles accessible to current user
ROLE_SYS_PRIVS	retrieves system privileges granted to the roles accessible to current user
ROLE_TAB_PRIVS	retrieves object privileges granted to the roles accessible to current user
DBA_SYS_PRIVS	system privileges granted to users and roles
USER_TAB_PRIVS	describes the object grants for which the current user is the object owner, grantor, or grantee
DBA_TAB_PRIVS	describes all object grants in the database
USER_TAB_PRIVS_MADE	retrieves all the object grants made by the current user
USER_TAB_PRIVS_RECD	retrieves all the object grants for which the current user is the grantee
SESSION_ROLES	retrieves the roles that are currently enabled by the current session

Guideline of Using Roles and Privileges

- Keep using roles, not direct grant
- Make the authority structure as simple as possible
- Grant the least privileges needed for users to do their jobs
- Make it documented
- Frequently have them reviewed by the system owners
- Split DBA role from developer role
- Proper auditing policy must be in place

Summary

In this lecture, you should have learnt how to perform the following:

- Understand the purpose of using roles
- List predefined roles
- Create and use roles
- Use password-protected roles
- Understand using **`PUBLIC`** role
- Drop roles
- Obtain information about the granted privileges and roles
- Understand the guideline of using roles and privileges