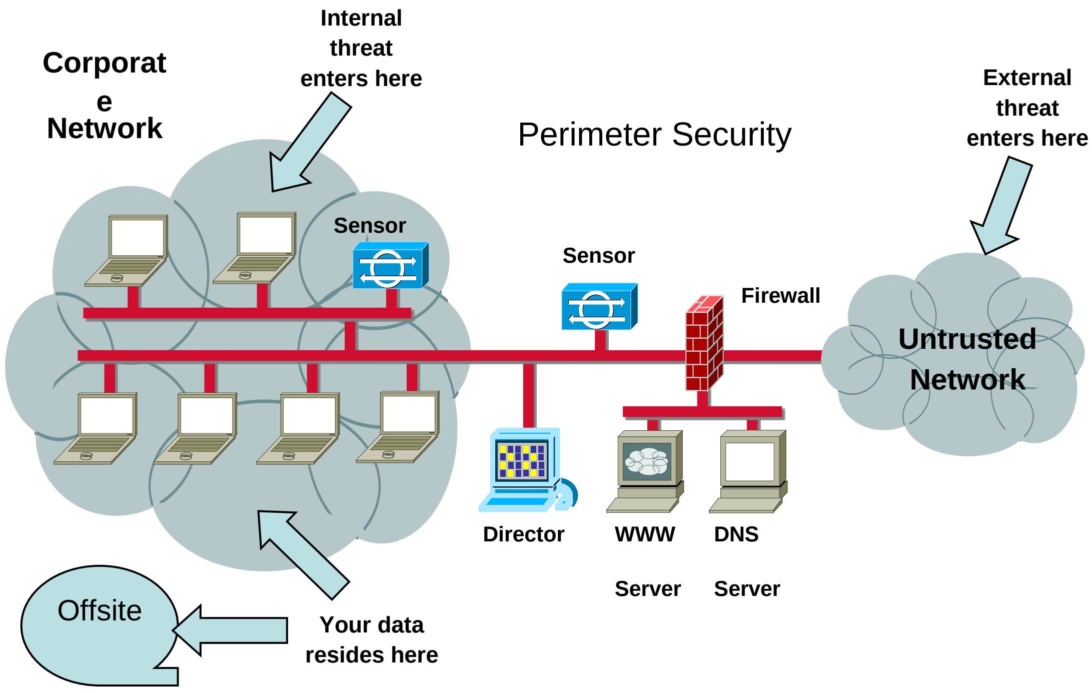
Database Administration - Security

Database security is the last line of defense in an information system. Most organizations do a reasonable job placing defensive layers at the network level by installing physical security at the data center, access controls on the operating system, firewalls and intrusion detection on the network. Unfortunately when these defensive measures are breached, as indicated by the figure below, the database can be compromised.



Notice in the figure above that if the threat compromises the firewalls and intrusion detection systems the last line of defense is the database. In order to strengthen the database security we need to use some of the capabilities that come with most modern database management systems. In this lab exercise we will be exploring some of the access control security features of Oracle 11g.

Learning Outcomes

<u>**Management of information technology**</u> **-** By completely this lab, you will be able to integrate technical and solution development concepts with the principles of IT governance. ***After completing this lab, you will be able to:***

- Propose, implement and maintain database security mechanisms

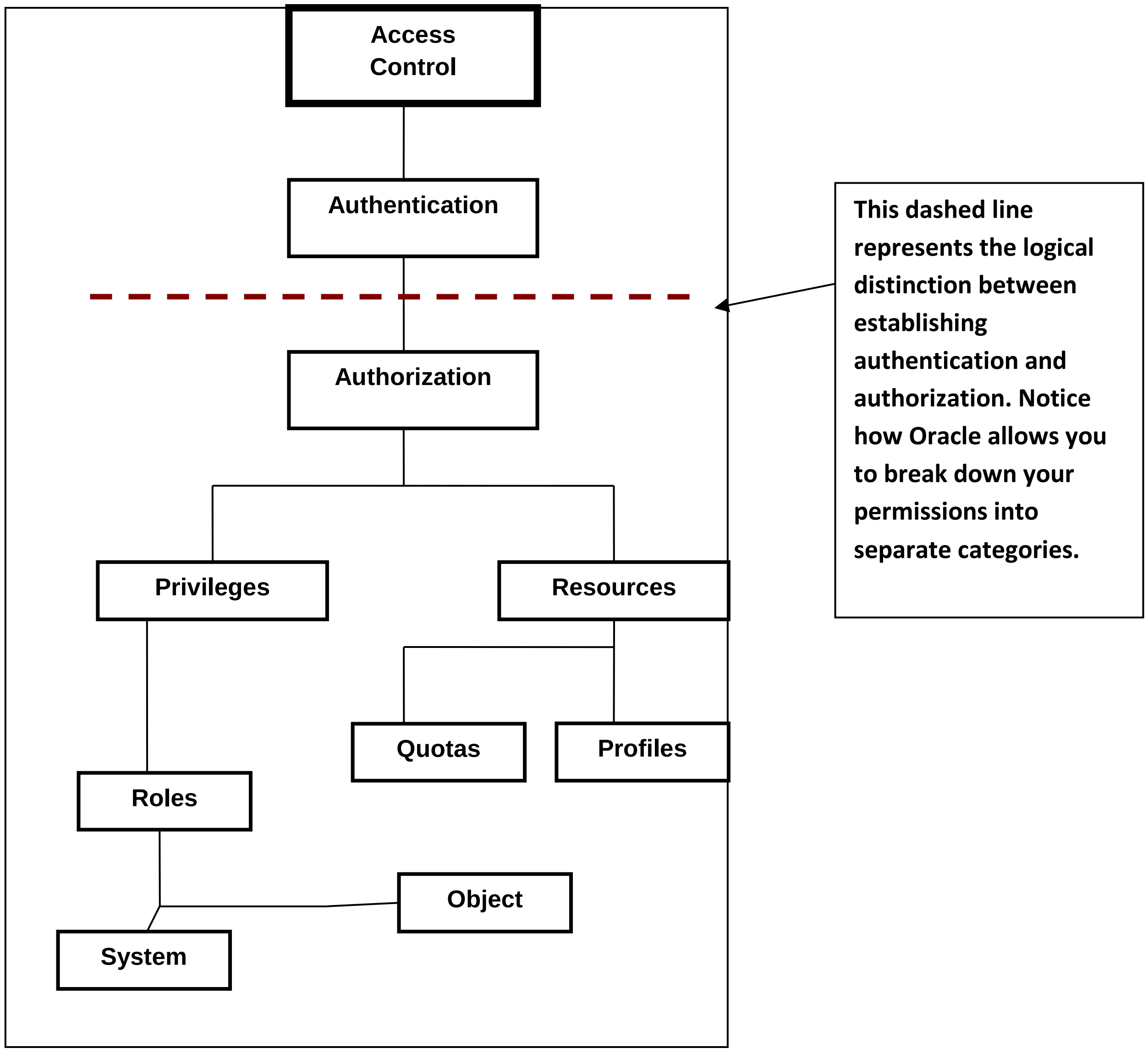
Specifically, after this topic you will be able to:

- Identify database security risks
- Discuss the role of the DBA in establishing and maintaining database security
- Understanding when and how to use Security Manager
- Create database Profiles
- Create database Roles

Access Control

Access controls are the security mechanisms that allow authorized or trusted users the capability to access the database and DBMS facilities they need to do their job. Access controls also prevent unauthorized users from accessing sensitive data or using powerful DBMS facilities, such as altering or dropping database objects. One use of access controls is known as "authentication."

The second access control mechanism, established after authentication, is known as "authorization." Authorization is the process of establishing the functions or services that a trusted user can perform in order to do their job. You may also hear authorization referred to as "permissions" or "authorities." The figure below graphically represents the Oracle access control hierarchy. Notice that the further down in the hierarchy you go the more granular the security becomes.


Access Control
Authentication
Authorization
Privileges
Resources
Quotas
Profiles
Roles
Object
System
This dashed line represents the logical distinction between establishing authentication and authorization. Notice how Oracle allows you to break down your permissions into separate categories.

There are two basic ways to establish the above security mechanisms: using Oracle Enterprise Manager Console (EM) or by using SQL commands.

0. Getting Started

IMPORTANT!!!!!! Before you begin, make sure you have the **EMPLOYEES** schema loaded up and you've connected to SQL Developer as the **EMPLOYEES** user. You can find instructions for doing this in the same place you found this lab. **See: Employees Schema document.**

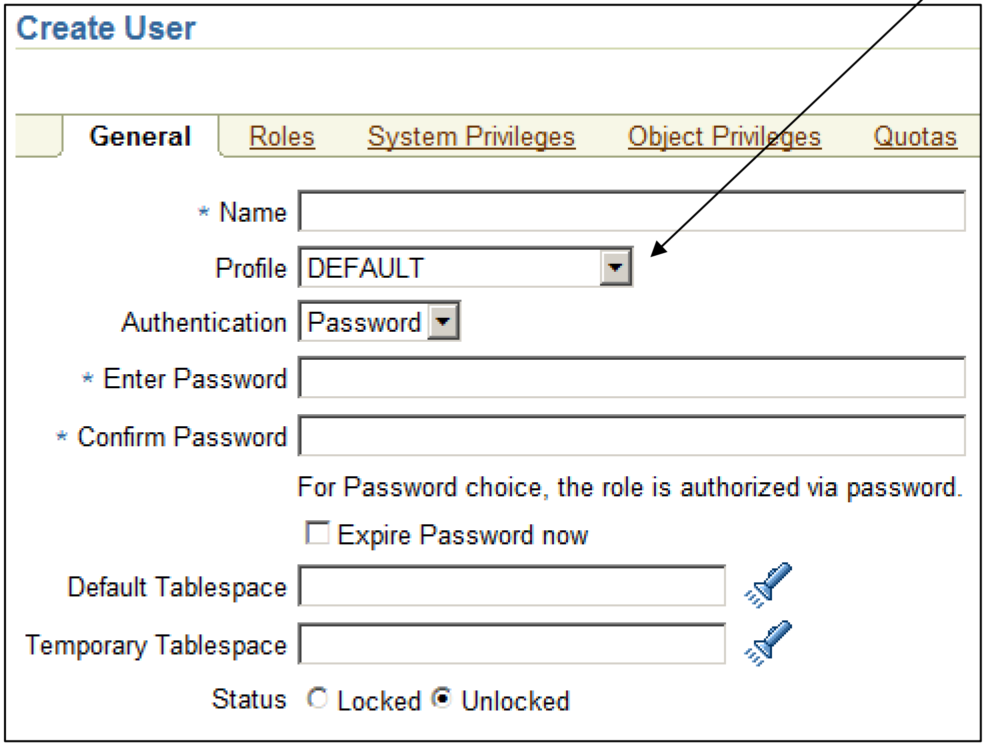
1. Establishing Authentication - Users

As stated earlier, authentication is the mechanism used to create a user account that allows a "trusted" user access to the database. This only provides verification that the user is who they claim they are and nothing more. As a general rule, once authenticated, the user still doesn't have permission to do anything. All you have done is created a user account with a password.

To create a user account using SQL you use the CREATE USER command. The format looks like this:

```
CREATE USER username IDENTIFIED BY password
      DEFAULT TABLESPACE tablespacename
      TEMPORARY TABLESPACE tablesapacename
      QUOTA nK|M|UNLIMITED ON TABLESPACE tablespacename
      PROFILE profilename
      ACCOUNT UNLOCK;
```

Notice that the SQL to create a user has the same parameters as the create user dialog box located in OEM Security.

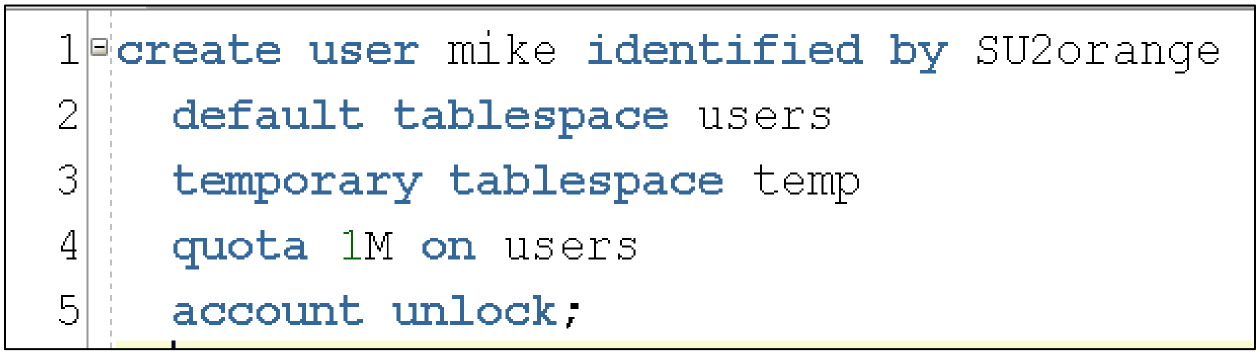


* Name	
Profile	DEFAULT
Authentication	Password
* Enter Password	
* Confirm Password	
	For Password choice, the role is authorized via password.
	☐ Expire Password now
Default Tablespace	
Temporary Tablespace	
Status	Locked [✓ Unlocked]

Caution!!!!!

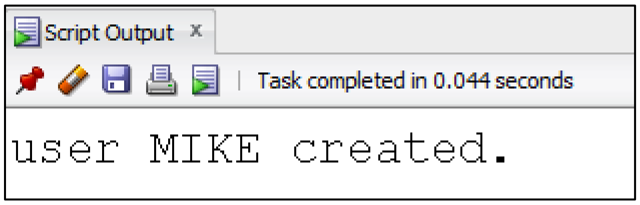
The only parameters required to create a user are the user name and the password. The rest of the parameters are optional. If a parameter is not specified, Oracle will establish a default value. This can be unwise. For example if the user or temporary tablespaces are not specified Oracle will use SYSTEM for the default tablespace. The SYSTEM tablespace is reserved for Oracle data dictionary objects. So you do not what to place user database objects in the same location as SYSTEM objects, as this can compromise security as well as adversely affect performance.

To create a user using the SQL CREATE USER command you would enter it in in SQL Developer just like any other SQL command. **Try This:**



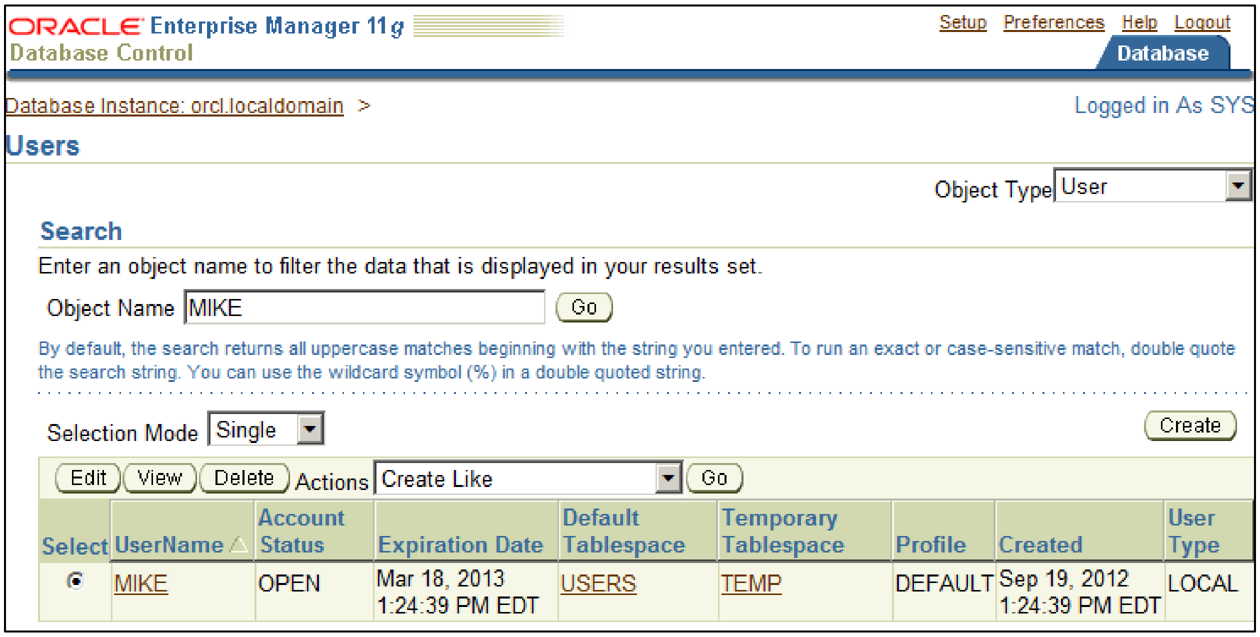
```
create user mike identified by SU2orange
  default tablespace users
  temporary tablespace temp
  quota 1M on users
  account unlock;
```

Upon executing this command you will see the following:



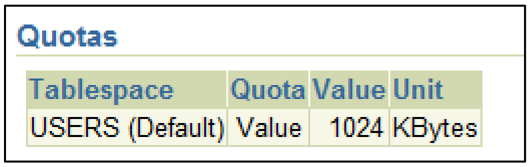
```
user MIKE created.
```

If you go into the OEM you will find this new user account that looks like this:


ORACLE Enterprise Manager 11g
Database Control
Setup
Preferences
Help
Logout
Database
Database Instance: orcl.localdomain >
Logged in As SYS
Users
Object Type User
Search
Enter an object name to filter the data that is displayed in your results set.
Object Name MIKE
Go
By default, the search returns all uppercase matches beginning with the string you entered. To run an exact or case-sensitive match, double quote the search string. You can use the wildcard symbol (%) in a double quoted string.
Selection Mode Single
Create
Edit
View
Delete
Actions Create Like
Go

Select	UserName	Account Status	Expiration Date	Default Tablespace	Temporary Tablespace	Profile	Created	User Type
●	MIKE	OPEN	Mar 18, 2013 1:24:39 PM EDT	USERS	TEMP	DEFAULT	Sep 19, 2012 1:24:39 PM EDT	LOCAL

Notice that no Roles, System or Object privileges were established but the quotas were:



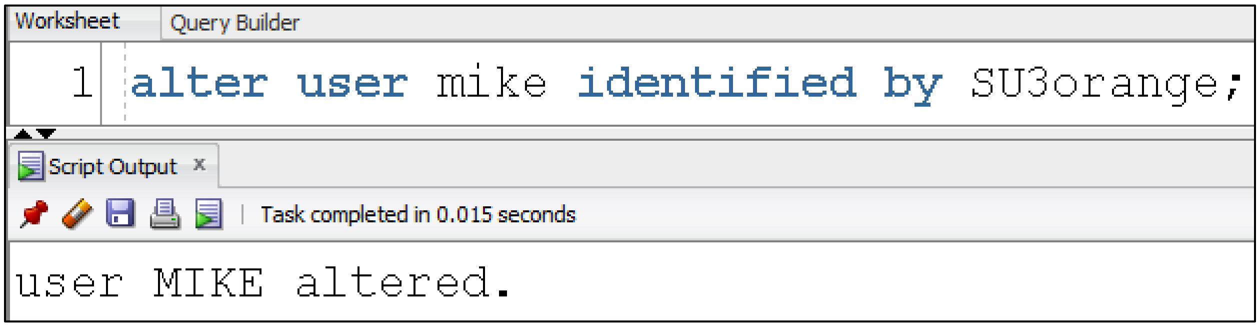
Quotas

Tablespace	Quota	Value	Unit
USERS (Default)	Value	1024	KBytes

You can change a user account by using the ALTER statement.

```
ALTER USER username IDENTIFIED BY password
      DEFAULT TABLESPACE tablespacename
      TEMPORARY TABLESPACE tablesapacename
      QUOTA nK|M|UNLIMITED ON TABLESPACE tablespacename
      PROFILE profilename
      PASSWORD EXPIRE
      ACCOUNT LOCK|UNLOCK;
```

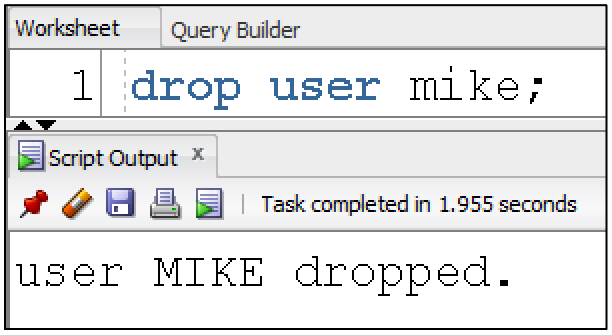
Here is an example where I changed the password from SU2orange to SU3orange. Go ahead and **try it.**



```
alter user mike identified by SU3orange;
```

```
user MIKE altered.
```

To remove a user account you use the DROP command. To delete the user account **mike** you would use the following command. **Try It.**



```
drop user mike;
```

```
user MIKE dropped.
```

Some thoughts about setting up new users

When setting up new user accounts in Oracle the default privileges granted to the newly created account will vary depending on the method you use to create the account i.e. SQL or OEM. Using the SQL method the CREATE USER command does not provide any system or object privileges. So essentially the user has an account but really can't do anything. Using OEM the user account gets the CONNECT role by default in 11g only provides the CREATE SESSON system privilege.

2. Establishing Authorization

After you have created a user account you can now start assigning privileges or permissions so that the new user can do their job effectively but at the same time you can assure that the database environment is not accidentally or intentionally compromised.

System and Object Privileges

Privileges come in two varieties: **System** and **Object**.

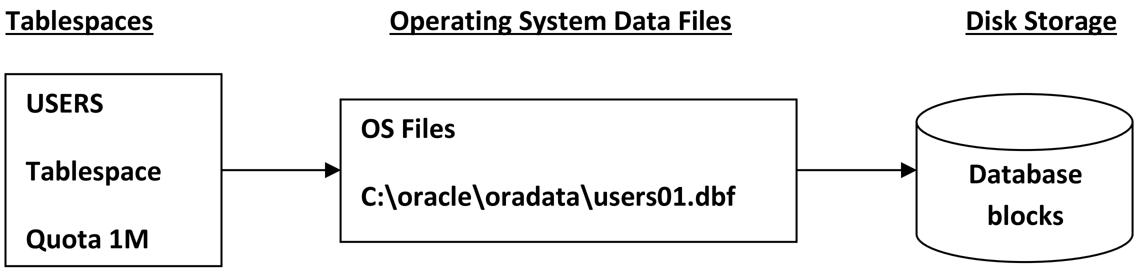
- **System privileges** allow the database user to make structural or operational changes to the database. These might include the capability to create objects like tables, views, procedures or alter these and other database objects. System privileges also include the ability to connect to a database instance or start and stop a database instance.
- **Object Privileges** are less powerful but they need to be managed as well. Object privileges provide the user the capability to delete, insert, select or update the contents contained by a database object. These privileges can be established to a very granular … right down to a table column level.

Privileges can be very time consuming and tedious to assign to individual user accounts especially in a very large organization. For a list of all System and Object privileges in Oracle 11g, visit: http://docs.oracle.com/cd/B28359_01/server.111/b28286/statements_9013.htm#i2077938

Quotas

When a new user account is established it doesn't have the authority or a location within the database to store objects. For example, if you create a user account with the capability to create database objects like: tables, views, procedures, functions or triggers the user account needs a special kind of privilege called a Quota in order to have enough disk storage to save these objects.

Quotas are really disk space allocations made through logical containers call tablespaces. They are usually made in increments or kilobytes or megabytes. The figure below illustrates the relationship between a tablespace quota, the associated operating system file and the physical disk drive that make up the database.



Roles

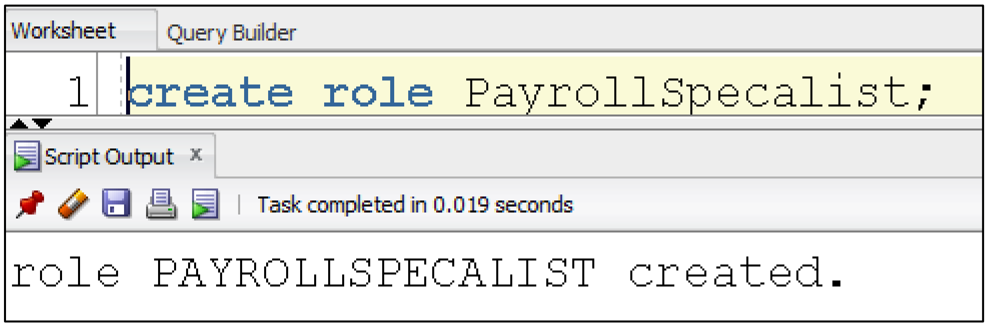
A best practice in database security is to establish user permissions through the use of Roles. A role is a collection of system and/or object privileges. The advantage of roles is that a role can be assigned to a user account giving all of the system and object privileges to the user at one time, versus the alternative of trying to assign individual system and object privileges to each user. This use of the role makes security maintenance significantly less work and less error prone. In addition it provides for consistent implementation of your database security policies.

A practice in most database organizations is to create roles that map system and object privileges to specific job duties. This way all employees in a work unit with similar job duties have the same permissions needed to do their job. A few examples of roles might include:

- **InventoryManager** - you might assign this class of employee the capability to create a session, select all of the data from the Product, Inventory and SalesOrder tables
- **BenefitsSpecialist** - here you might give this class of employee the capability to create a session, select all data from the EmployeeBenefit and Employee tables but not be able to access the employee salary
- **PayrollSpecialist** - this group of employee needs to create a session and is responsible for working with all of the Employee data

You can create roles using both SQL and OEM. With SQL you need to create the role first then you can grant it specific privileges.

Here is an example of creating and granting privileges to a role using SQL. **Try It:**

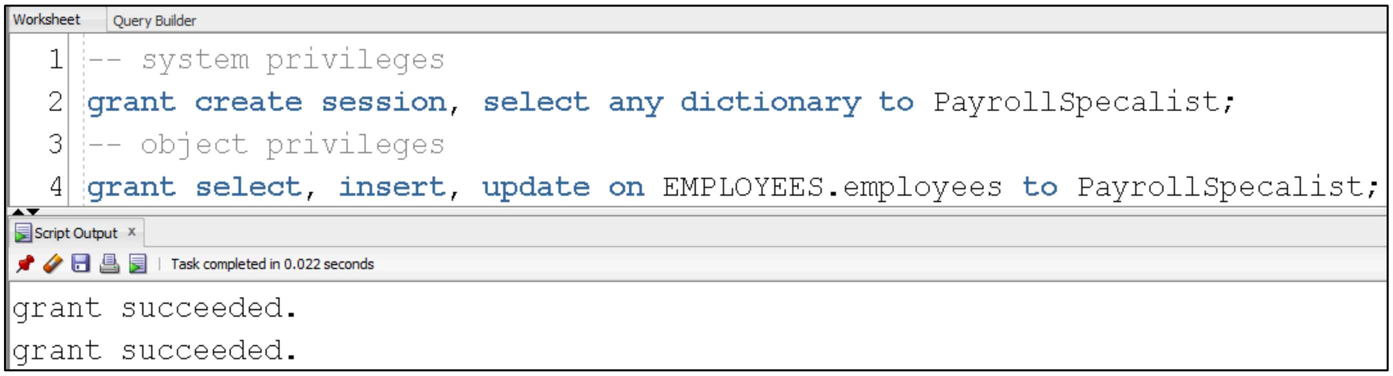


```
create role PayrollSpecalist;
```

Task completed in 0.019 seconds

```
role PAYROLLSPECALIST created.
```

Once you have created the role you can now provide system and object privileges to it by using the grant command. **Try This:**

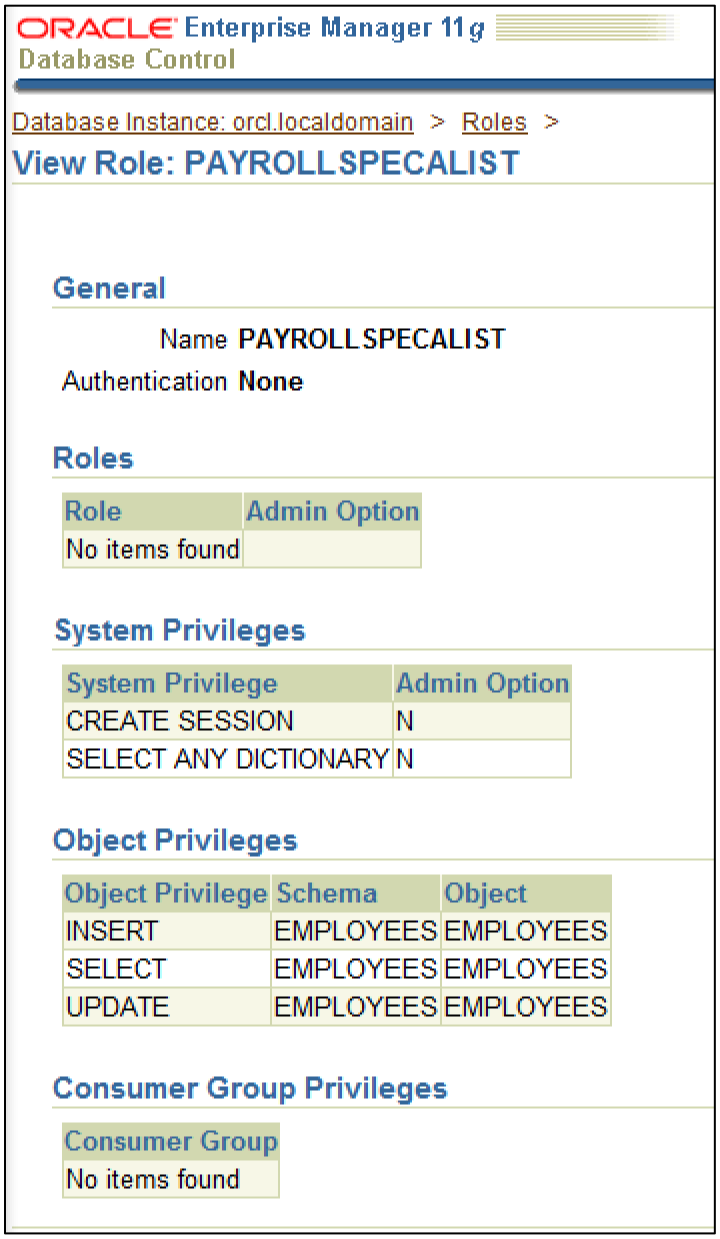


```
-- system privileges
grant create session, select any dictionary to PayrollSpecalist;
-- object privileges
grant select, insert, update on EMPLOYEES.employees to PayrollSpecalist;
```

Task completed in 0.022 seconds

```
grant succeeded.
grant succeeded.
```

After these commands the ROLE in OEM looks like this. Here are the system privileges:



ORACLE Enterprise Manager 11g
Database Control
Database Instance: orcl.localdomain > Roles >

View Role: PAYROLLSPECALIST

General

Name PAYROLLSPECALIST
Authentication None

Roles

Role	Admin Option
No items found	

System Privileges

System Privilege	Admin Option
CREATE SESSION	N
SELECT ANY DICTIONARY	N

Object Privileges

Object Privilege	Schema	Object
INSERT	EMPLOYEES	EMPLOYEES
SELECT	EMPLOYEES	EMPLOYEES
UPDATE	EMPLOYEES	EMPLOYEES

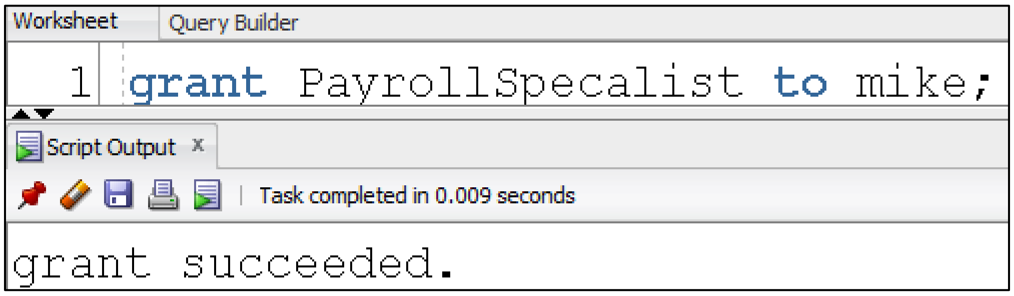
Consumer Group Privileges

Consumer Group
No items found

To assign the Role to a user account you use the GRANT command.

NOTE: Before you can try this example, first **re-create** the **mike** user from part 1.

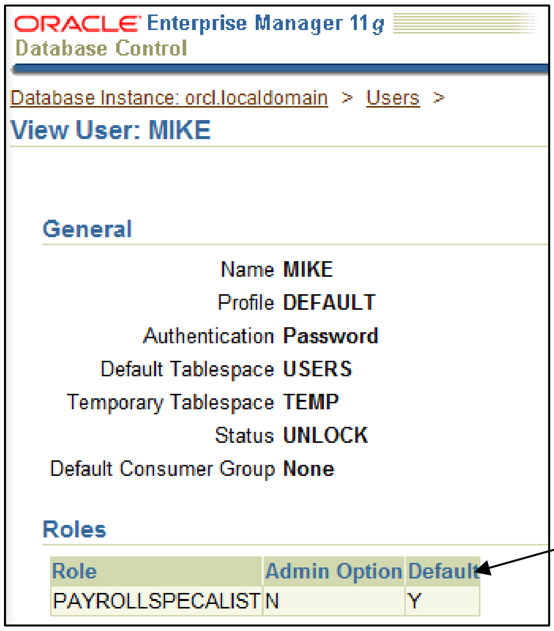
After you've created the user, **try this**:



```
grant PayrollSpecalist to mike;
```

```
grant succeeded.
```

Let's take a look at the **mike** account inside **OEM:** Here you see that the account has the **PayrollSpecialist** role with all of its privileges assigned:


ORACLE Enterprise Manager 11g
Database Control
Database Instance: orcl.localdomain > Users >

View User: MIKE

General

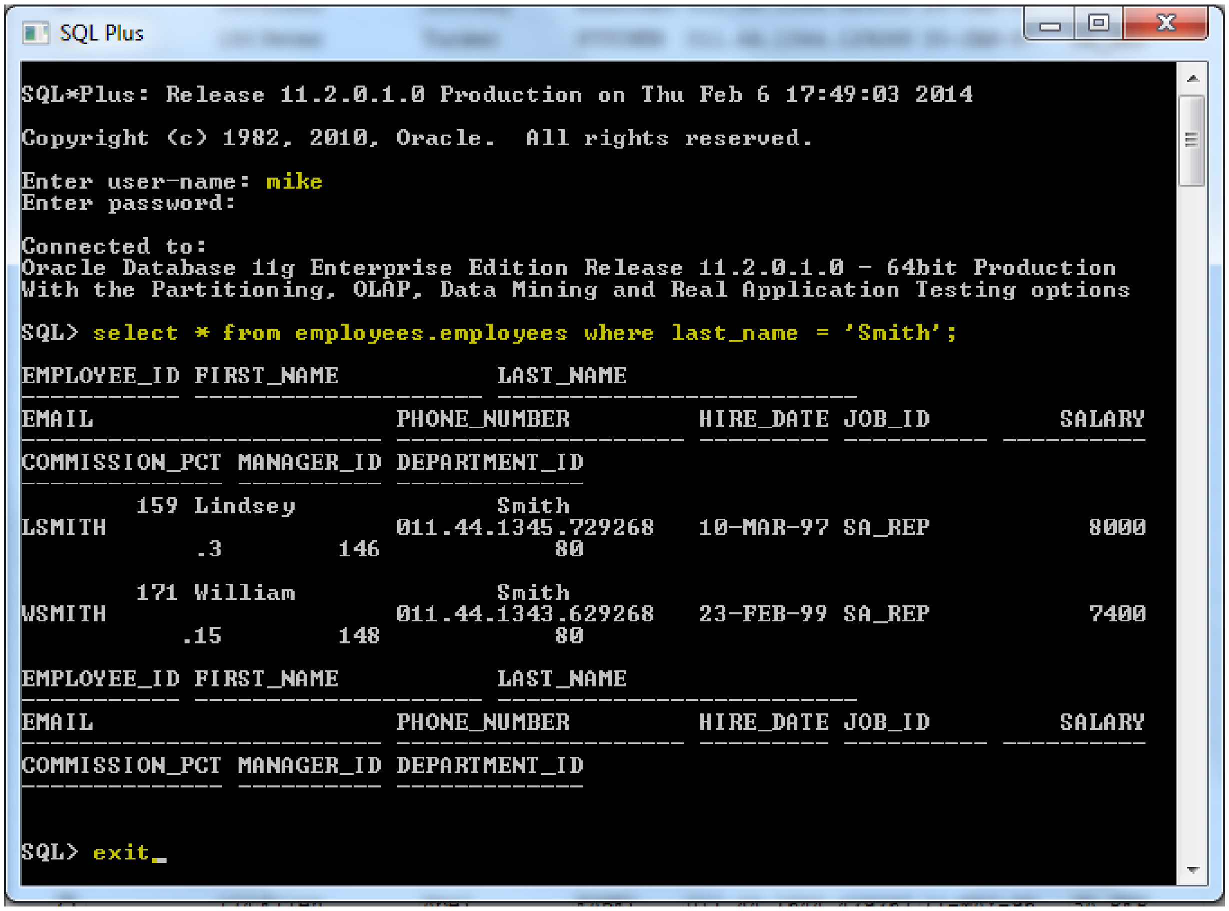
Name	MIKE
Profile	DEFAULT
Authentication	Password
Default Tablespace	USERS
Temporary Tablespace	TEMP
Status	UNLOCK
Default Consumer Group	None

Roles

Role	Admin Option	Default
PAYROLLSPECIALIST	N	Y

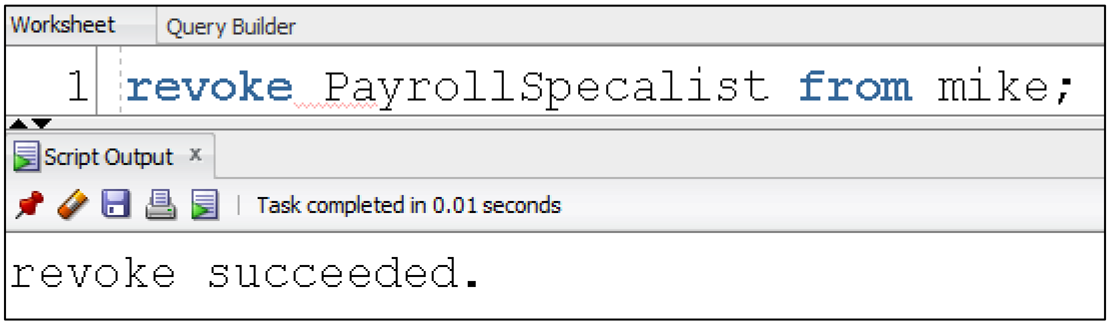
Here you can see that the role has been successfully assigned to the account.

Next, let's take the **MIKE** user for a spin and verify the user can SELECT from the EMPLOYEES.EMPLOYEES table. Note: we'll need to qualify the ***schema*** in this case because the user MIKE has a default table space of USERS. Here's a screenshot of what you should try using SQL Plus. (You can use SQL developer if you like).



```
SQL*Plus: Release 11.2.0.1.0 Production on Thu Feb 6 17:49:03 2014

Copyright (c) 1982, 2010, Oracle.  All rights reserved.

Enter user-name: mike
Enter password:

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> select * from employees.employees where last_name = 'Smith';

EMPLOYEE_ID FIRST_NAME           LAST_NAME
----------- -------------------- -------------------------
EMAIL                     PHONE_NUMBER         HIRE_DATE JOB_ID         SALARY
------------------------- -------------------- --------- ---------- ----------
COMMISSION_PCT MANAGER_ID DEPARTMENT_ID
-------------- ---------- -------------
        159 Lindsey              Smith
LSMITH                    011.44.1345.729268   10-MAR-97 SA_REP           8000
            .3        146            80

        171 William              Smith
WSMITH                    011.44.1343.629268   23-FEB-99 SA_REP           7400
           .15        148            80

EMPLOYEE_ID FIRST_NAME           LAST_NAME
----------- -------------------- -------------------------
EMAIL                     PHONE_NUMBER         HIRE_DATE JOB_ID         SALARY
------------------------- -------------------- --------- ---------- ----------
COMMISSION_PCT MANAGER_ID DEPARTMENT_ID
-------------- ---------- -------------


SQL> exit_
```

And finally, To remove a Role or Privilege from a user account or a role you use the REVOKE command. **Try This:**



```
revoke PayrollSpecalist from mike;
```

```
revoke succeeded.
```

On Your Own: Database Security Lab Exercise

Based on your recommendation, your boss, the manager of database administration has made the decision to move all database applications and databases to Oracle 11g. She has asked you to develop the security procedures since you are the only DBA familiar with Oracle security. You decide to set up the database security objects with the following rules:

1. Make an **CST4714User** Role
    a. This role can connect to the database
    b. This role can select, insert, update, and delete data in the CST4714 tablespace.

2. Make an **CST4714Developer** Role
    a. This role can connect to the database
    b. This role can create, alter, and drop tables, sequences and views for the CST4714 Schema.
    c. This role can select, insert, update, and delete data in the CST4714 tablespace.

3. Make an **CST4714User** user
    a. This user's default tablespace is CST4714
    b. This user has a 5 MB quota on CST4714 tablespace.
    c. This user is a member of the CST4714User role

4. Make a **CST4714Dev** user
    a. This user's default tablespace is CST4714
    b. This user has unlimited quota on CST4714 tablespace.
    c. This user is a member of the CST4714Developer role

5. Make a **CST4714DBA** User
    a. This user can perform all system dba functions including database startup and shutdown
    b. This use can create and maintain tablespaces, roles, profiles, users and schemas.
    c. This user can use OEM
    d. This user has no other privileges. They cannot create and maintain any other database objects, such as tables, views, procedures, etc. They can also not add, read or change data from the tables.

Turn in:

- A word document with code and screenshots.
- Create the above objects using SQL Developer (**do not** create these users using OEM). Attach the **well documented** SQL syntax for all objects.
- For your new roles and users attach screen shots that prove these objects were set up correctly. Using screen shots from SQL Developer show that your CREATE SQL created the object correctly. Launch OEM to prove a user has the proper role, or a role has the proper privileges.
- Indicate which portions of your SQL satisfy the above requirements by using the above number and the letter (e.g. 3.a., 5.d, etc…)