# Oracle CREATE ROLE Statement

**Summary**: In this lab, you will learn how to use the Oracle `CREATE ROLE` statement to create roles in the Oracle Database.

## Oracle `CREATE ROLE` statement

A role is a group of privileges. Instead of granting individual privileges to users, you can group related privileges into a role and grant this role to users. Roles help manage privileges more efficiently.

To create a new role, you use the `CREATE ROLE` statement. The basic syntax of the `CREATE ROLE` statement is as follows:

```
CREATE ROLE role_name
[IDENTIFIED BY password]
[NOT IDENTIFIED]
```

In this syntax:

- First, specify the name of the role that you want to create.
- Second, use `IDENTIFIED BY password` option to create a local role and indicate that the user, who was granted the role, must provide the `password` to the database when enabling the role.
- Third, use `NOT IDENTIFIED` to indicate that the role is authorized by the database and that the user, who was granted this role, doesn't need a password to enable the role.

After a role is created, it is empty. To grant privileges to a role, you use the `GRANT` statement:

```
GRANT {system_privileges | object_privileges} TO role_name;
```

In addition, you can use the `GRANT` statement to grant privileges of a role to another role:

```
GRANT role_name TO another_role_name;
```

## Oracle CREATE ROLE statement examples

Let's take some examples of using the `CREATE ROLE` statement.

### 1) Using Oracle CREATE ROLE without a password example

First, create a new role named `mdm` (master data management) in the sample database as sys user:

```
CREATE ROLES mdm;
```

Second, grant object privileges on `customers`, `t2` tables to the `mdm` role:

```
GRANT SELECT, INSERT, UPDATE, DELETE
ON customers
TO mdm;

GRANT SELECT, INSERT, UPDATE, DELETE
ON t2
TO mdm;
```

**Note:** You can create `customers` and `t2` tables if they don't exist already before running above commands.

Third, create a new user named `alice` and grant the `CREATE SESSION` privilege to `alice`:

```
CREATE USER alice IDENTIFIED BY abcd1234;

GRANT CREATE SESSION TO alice;
```

Fourth, log in to the database as `alice` using sql developer.

and attempt to `query data` from the `sys.customers` table:

```
SELECT * FROM sys.customers;
```

Oracle issued the following error:

```
ORA-00942: table or view does not exist
```

Go back to the first session and grant `alice` the `mdm` role:

```
GRANT mdm TO alice;
```

Go to the Alice's session and enable the role using the `[SET ROLE]` statement:

```
SET ROLE mdm;
```

To query all roles of the current user, you use the following query:

```
SELECT * FROM session_roles;
```

Here is the role of `alice`:

```
ROLE

MDM
```

Now, `alice` can manipulate data in the master data tables such as `customers` and `t2`. Now, attempt to `query data` from the `sys.customers` table:

```
SELECT * FROM sys.customers;
```

### 2) Using Oracle `CREATE ROLE` to create a role with `IDENTIFIED BY password` example

First, create a new role named `order_entry` with the password `xyz123`:

```
CREATE ROLE order_entry IDENTIFIED BY xyz123;
```

Next, grant object privileges of the `orders` and `order_items` tables to the `order_entry` role:

```
CREATE TABLE orders(order_name varchar(50));

INSERT INTO orders(order_name) VALUES('testOrder');
INSERT INTO orders(order_name) VALUES('grocery');
```

```
INSERT INTO orders(order_name) VALUES('hardware');

CREATE TABLE order_items(item_name varchar(50));

GRANT SELECT, INSERT, UPDATE, DELETE
ON orders
TO order_entry;

GRANT SELECT, INSERT, UPDATE, DELETE
ON order_items
TO order_entry;
```

Then, grant the `order_entry` role to the user `alice`:

```
GRANT order_entry TO alice;
```

After that, log in as `alice` and enable the `order_entry` role by using the `[SET ROLE]` statement:

```
SET ROLE
    order_entry IDENTIFIED BY xyz123,
    mdm;
```

Finally, use the following statement to get the current roles of `alice`:

```
SELECT * FROM session_roles;
```

Here are the current roles of `alice`:

```
ROLE

MDM
ORDER_ENTRY
```

In this lab, you have learned how to use the Oracle `CREATE ROLE` statement to create roles in the database.