

# Oracle REVOKE Statement Explained

**Summary:** In this lab, you will learn how to use the Oracle `REVOKE` statement to revoke system and object privileges from a specific user.

## Oracle `REVOKE` statement

The Oracle `REVOKE` statement revokes system and object privileges from a user. Here is the basic syntax of the Oracle `REVOKE` statement:

```
REVOKE {system_privilege | object_privilege } FROM user;
```

In this syntax:

- First, specify the system or object privileges that you want to revoke from the user.
- Second, specify the user from which you want to revoke the privileges.

In order to revoke a system privilege from a user, you must have been granted the system privilege with the `ADMIN OPTION`.

To revoke an object privilege from a user, you must previously granted the object privilege to the user or you must have the `GRANT ANY OBJECT PRIVILEGE` system privilege.

On top of this, you can use the `REVOKE` statement to revoke only privileges that were granted directly with a `GRANT` statement. In other words, you cannot use the `REVOKE` statement to revoke privileges that were granted through the operating system or roles.

To revoke all system privileges from a user, you can use the following statement:

```
REVOKE ALL PRIVILEGES FROM user;
```

## Oracle `REVOKE` statement example

First, create a usernames `bob` and grant him the `CREATE SESSION` system privilege so that he can log in to the Oracle Database:

```
CREATE USER bob IDENTIFIED BY abcd1234;  
GRANT CREATE SESSION TO bob;
```

Second, grant the `CREATE TABLE` system privilege to `bob`:

```
GRANT CREATE TABLE TO bob;
```

Third, grant the `SELECT`, `INSERT`, `UPDATE` and `DELETE` object privileges to `bob` on `sys.customers` table:

```
CREATE TABLE customers(name varchar(50));  
  
INSERT INTO customers(name) VALUES('a');  
INSERT INTO customers(name) VALUES('b');  
INSERT INTO customers(name) VALUES('c');  
INSERT INTO customers(name) VALUES('d');  
INSERT INTO customers(name) VALUES('e');
```

```
GRANT SELECT, INSERT, UPDATE, DELETE ON sys.customers
TO bob;
```

Now, `bob` can create a new table in his own schema and manipulate data in the `sys.customers` table.

Fourth, log in to the Oracle Database as `bob` and execute the following statements:

```
CREATE TABLE t1(id int);

SELECT
    name
FROM
    sys.customers
ORDER BY
    name
FETCH FIRST 5 ROWS ONLY;
```

Both queries are executed successfully because the user `bob` has sufficient privileges.

Fifth, revoke the object privileges from `bob` :

```
REVOKE SELECT, INSERT, UPDATE, DELETE ON sys.customers
FROM bob;
```

Sixth, go to the `bob` 's session and select data from the `sys.customers` table:

```
SELECT
    name
FROM
    sys.customers
ORDER BY
    name
FETCH FIRST 5 ROWS ONLY;
```

Oracle issued the following error:

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

This is correct because `bob` is no longer has the `SELECT` object privilege on the `sys.customers` table.

Seventh, revoke the `CREATE TABLE` system privilege from `bob` :

```
REVOKE CREATE TABLE FROM bob;
```

Eighth, go to `bob`'s session and attempt to `create a new table` :

```
CREATE TABLE t2(id INT);
```

Oracle issued the following error, which is what we expected.

```
ORA-01031: insufficient privileges
```

If you don't want `bob` to log in, you can revoke the `CREATE SESSION` system privilege as shown in the following statement:

```
REVOKE CREATE SESSION FROM bob;
```

Next time, `bob` won't be able to log in to the Oracle Database anymore.

In this lab, you have learned how to use the Oracle `REVOKE` statement to revoke system and object privileges from a user.