

Creating User Profiles

Overview:

Oracle uses Profiles to set resource limits and password parameters that restrict database usage and instance resources for a user. Using profiles is database security best practice as a database administrator can easily implement and enforce agency or business rules for database usage. This document will provide details of how to create, use, alter and delete Profiles within the Oracle database.

Restricted Use:

For security reasons, some permissions have not been provided to students to avoid accidental deletion or modification of other user accounts and profiles.

Profile parameters:

When you create a Profile both resource and password parameters are available to be set. A default profile is assumed if a specific profile is not provided. Table 1 provides the parameters and a brief description.

Table 1. Oracle Profile Parameters

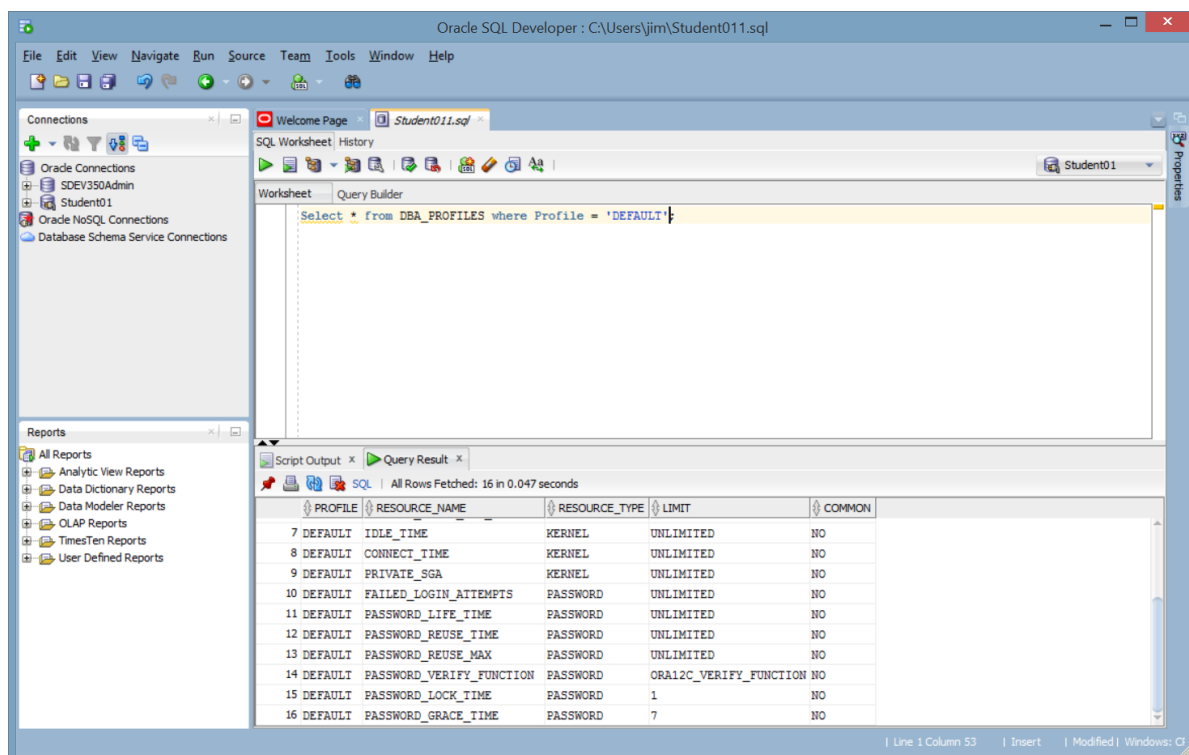
Parameter	Description
SESSIONS_PER_USER	Number of concurrent sessions to which you want to limit the user
CPU_PER_SESSION	CPU time limit for a session, expressed in hundredth of seconds
CPU_PER_CALL	CPU time limit for a call (a parse, execute, or fetch), expressed in hundredths of seconds.
CONNECT_TIME	Total elapsed time limit for a session, expressed in minutes
IDLE_TIME	Allowed periods of continuous inactive time during a session, expressed in minutes. Long-running queries and other operations are not subject to this limit.
LOGICAL_READS_PER_SESSION	Permitted number of data blocks read in a session, including blocks read from memory and disk.
LOGICAL_READS_PER_CALL	Permitted number of data blocks read for a call to process a SQL statement (a parse, execute, or fetch).
PRIVATE_SGA	Amount of private space a session can allocate in the shared pool of the system global area (SGA).
FAILED_LOGIN_ATTEMPTS	Number of failed attempts to log in to the user account before the account is locked.
PASSWORD_LIFE_TIME	Number of days the same password can be used for authentication.
PASSWORD_REUSE_TIME	Number of days before which a password cannot be reused
PASSWORD_REUSE_MAX	Number of password changes required before the current password can be reused
PASSWORD_LOCK_TIME	Number of days an account will be locked after the specified number of consecutive failed login attempts
PASSWORD_GRACE_TIME	Number of days after the grace period begins during which a warning is issued and login is allowed
PASSWORD_VERIFY_FUNCTION	Password complexity

Although the CPU and logical read parameters are interesting, for this course, we will focus on the password-related parameters. The default values for the password-related parameters are:

- FAILED_LOGIN_ATTEMPTS – 10 times
- PASSWORD_LIFE_TIME - UNLIMITED
- PASSWORD_REUSE_TIME - UNLIMITED
- PASSWORD_REUSE_MAX - UNLIMITED
- PASSWORD_LOCK_TIME – 1 day
- PASSWORD_GRACE_TIME – 7 days
- PASSWORD_VERIFY_FUNCTION – null

Recalling the data dictionary work from Lab 2, you can query the DBA_PROFILES view to retrieve the current settings for the DEFAULT profile (See figure 1).

```
Select * from DBA_PROFILES where Profile = 'DEFAULT';
```



The screenshot shows the Oracle SQL Developer interface. The main window displays the query results for the query `Select * from DBA_PROFILES where Profile = 'DEFAULT';`. The results are shown in a table with 5 columns: PROFILE, RESOURCE_NAME, RESOURCE_TYPE, LIMIT, and COMMON. The table contains 16 rows of data for the DEFAULT profile.

PROFILE	RESOURCE_NAME	RESOURCE_TYPE	LIMIT	COMMON
7 DEFAULT	IDLE_TIME	KERNEL	UNLIMITED	NO
8 DEFAULT	CONNECT_TIME	KERNEL	UNLIMITED	NO
9 DEFAULT	PRIVATE_SGA	KERNEL	UNLIMITED	NO
10 DEFAULT	FAILED_LOGIN_ATTEMPTS	PASSWORD	UNLIMITED	NO
11 DEFAULT	PASSWORD_LIFE_TIME	PASSWORD	UNLIMITED	NO
12 DEFAULT	PASSWORD_REUSE_TIME	PASSWORD	UNLIMITED	NO
13 DEFAULT	PASSWORD_REUSE_MAX	PASSWORD	UNLIMITED	NO
14 DEFAULT	PASSWORD_VERIFY_FUNCTION	PASSWORD	ORA12C_VERIFY_FUNCTION	NO
15 DEFAULT	PASSWORD_LOCK_TIME	PASSWORD	1	NO
16 DEFAULT	PASSWORD_GRACE_TIME	PASSWORD	7	NO

Figure 1 Retrieving the DEFAULT DBA_PROFILE Values

You can see the default values are not very secure and most likely conflict with most organizations password policies. Therefore, the creation of Profiles aligned with organizational policies and security best practices is recommended.

Create Profile:

To create a Profile, you must have the Create Profile system privilege.

Figure 2 provides Oracle's diagram showing the overall syntax for creating a Profile.

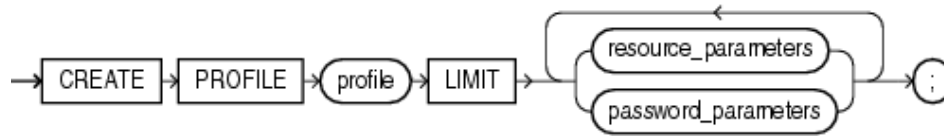


Figure 2 Create Profile Syntax

Although challenging to read at first, the syntax diagram lists the overall syntax structure with a loop where multiple parameters can be entered. For example, the following Create Profile statement creates a profile named `app_users` and limits the `FAILED_LOGIN_ATTEMPTS`, `PASSWORD_LIFE_TIME`, `PASSWORD_REUSE_TIME`, `PASSWORD_REUSE_MAX`, `PASSWORD_VERIFY_FUNCTION`, `PASSWORD_LOCK_TIME`, and `PASSWORD_GRACE_TIME` to the specific values listed.

```
CREATE PROFILE app_users LIMIT
  FAILED_LOGIN_ATTEMPTS 5
  PASSWORD_LIFE_TIME 60
  PASSWORD_REUSE_TIME 60
  PASSWORD_REUSE_MAX 5
  PASSWORD_VERIFY_FUNCTION ORA12C_VERIFY_FUNCTION
  PASSWORD_LOCK_TIME 1/24
  PASSWORD_GRACE_TIME 10;
```

In this example, no more than 5 failed login attempts are permitted. In addition, the password life is no more than 60 days and passwords can't be used more than 5 times. The 1/24 is equivalent to one hour as 1 stands for the hour and there are 24 hours in a day. If you wanted a lock time of 15 minutes, you could divide by 4 and use 1/96.

Password complexity functions check that each password is complex enough to protect against intruders who try to guess user passwords. These functions force users to create strong, secure passwords for database user accounts.

Table 2 lists and describes the available complexity functions.

Table 2 Oracle Password Complexity Functions

Password Complexity Function	Description
verify_function_11G	Originated in Oracle Database Release 11g providing baseline password complexity requirements.
ora12c_verify_function	Provides some requirements based on Department of Defense (DoD) Security Technical Implementation Guides (STIGs).
ora12c_strong_verify_function	Fulfills the DoD Database STIG requirements.

For the verify_function_11G Password Complexity function passwords must adhere to the following requirements:

- Length >=8 and Length <=30 characters.
- Does not contain the double-quotation character (")
- != username AND != backwards(username) AND != username appended with numbers 1-100
- != servername AND != servername appended with numbers 1-100
- ! Simple (oracle, oracle32, user1234 ...l).
- Includes at least 1 numeric and 1 alphabetic character
- Differs from the previous password by at least 3 characters

For the ora12c_verify_function Password Complexity function passwords must adhere to the following requirements:

- Length >=8 AND Length <=30 characters.
- Does not contain the double-quotation character (")
- Includes at least 1 numeric and 1 alphabetic character
- != username AND != backwards(username)
- != servername AND != backwards(servername)
- Does not contain the word oracle
- ! Simple (oracle, oracle32, user1234 ...l).
- Differs from the previous password by at least 3 characters
- Contains at least one special character.

For the ora12c_strong_verify_function Password Complexity function passwords must adhere to the following requirements:

- All of the ora12c_Verify_Functions, AND
- Contain at least 2 upper case characters, 2 lower case characters, 2 numeric characters, and 2 special characters
- Differ from the previous password by at least 4 characters
- Length >=9 AND Length <=30 characters

It is possible to create custom Profiles beyond the available options Oracle provides. Additional execute permissions are required to use this functionality.

As an example create Profile statement consider the following simple Create statement that creates a profile named studentProf with a maximum password reuse of 4 and password reuse time of 30 days.

```
CREATE PROFILE studentProf
  LIMIT PASSWORD_REUSE_MAX 4
  PASSWORD_REUSE_TIME 30;
```

Querying the DBA_Profiles for the new Profile shows these parameters as well as the default assigned values. Note the Profile name is stored in UpperCase so be sure to invoke the correct SQL statement:

```
select * from DBA_PROFILES where Profile = 'STUDENTPROF';
```

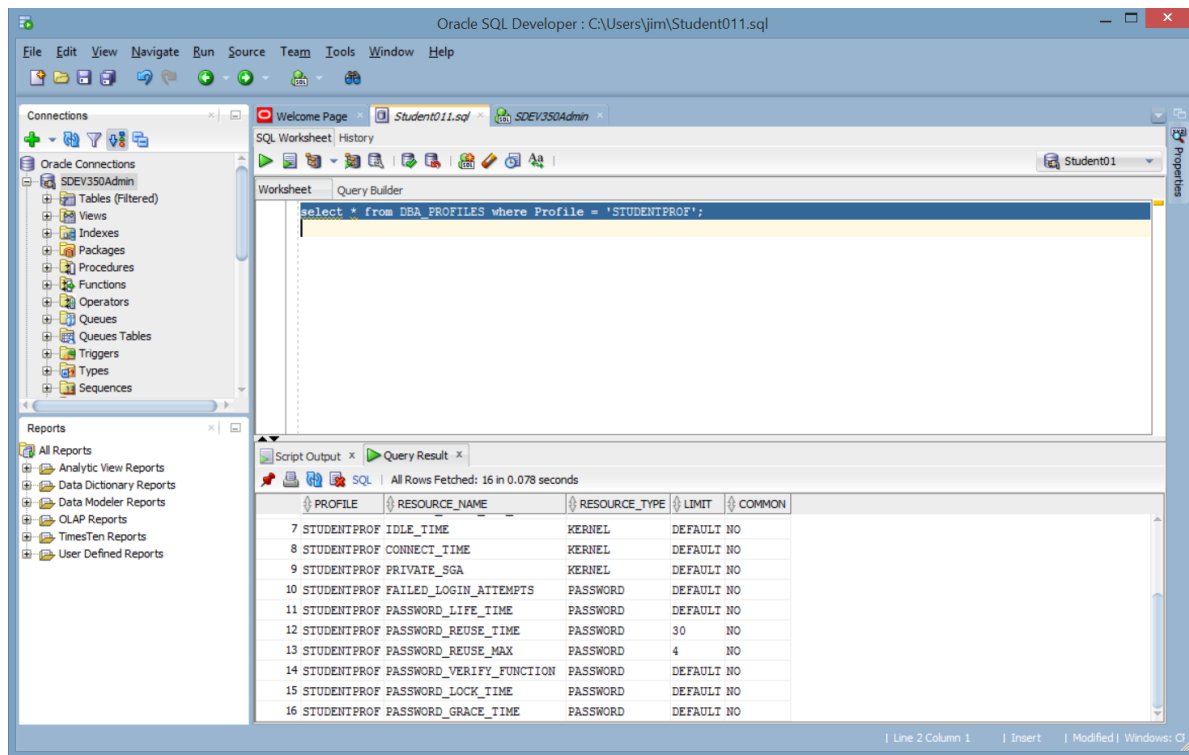


Figure 3 Creating a Profile

Altering a Profile:

The Alter Profile statement can be used to alter an existing profile. You do need the Alter Profile permissions for a successful Alter call.

The following statement Alters the STUDENTPROP to set the failed login attempts 10 and provides a password life time of 180 days.

```
ALTER PROFILE studentProf LIMIT
  FAILED_LOGIN_ATTEMPTS 10
  PASSWORD_LIFE_TIME 180;
```

Running this statement from an account with the ALTER PROFILE privileges results in two additional password parameters to be modified:

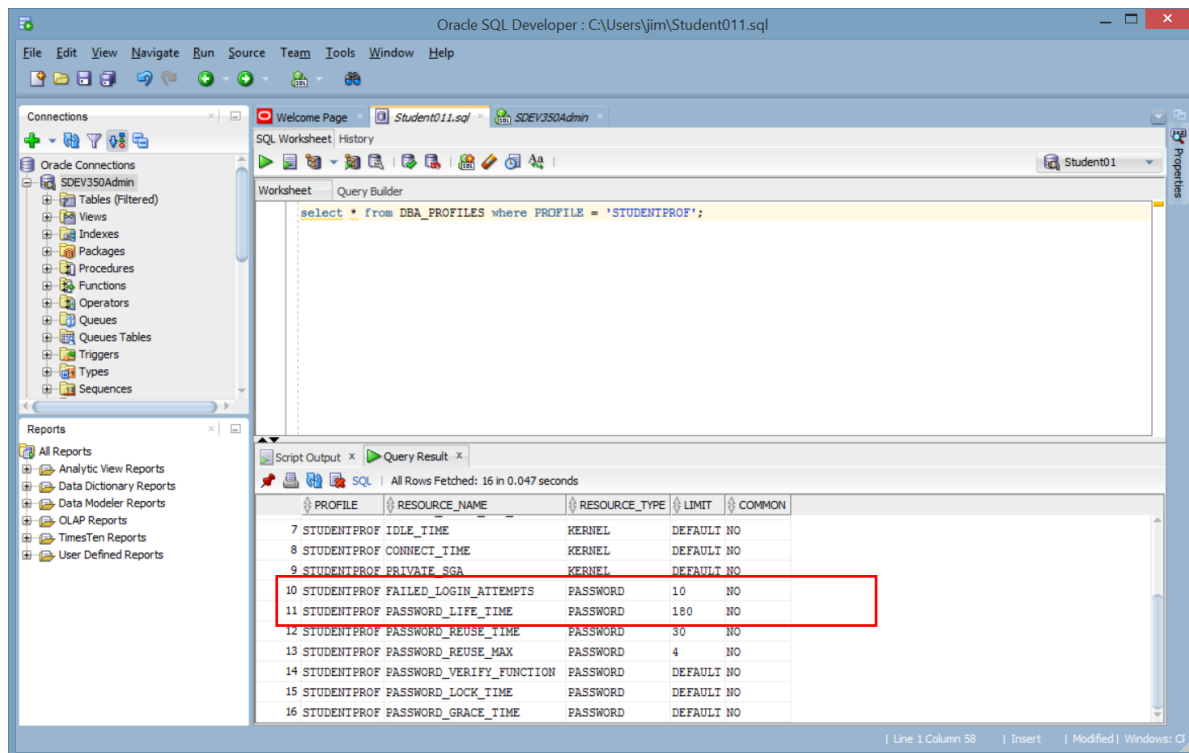


Figure 4 Altering a Profile

Drop a Profile:

The Drop Profile statement can be used to delete an existing profile. You do need the Drop Profile permissions for a successful Drop call. **If a user has already been assigned the profile, the cascade clause must be used**

```
Drop PROFILE studentProf cascade;
```

Successfully dropping the profile will remove it from the DBA_PROFILES view as shown in Figure 5.

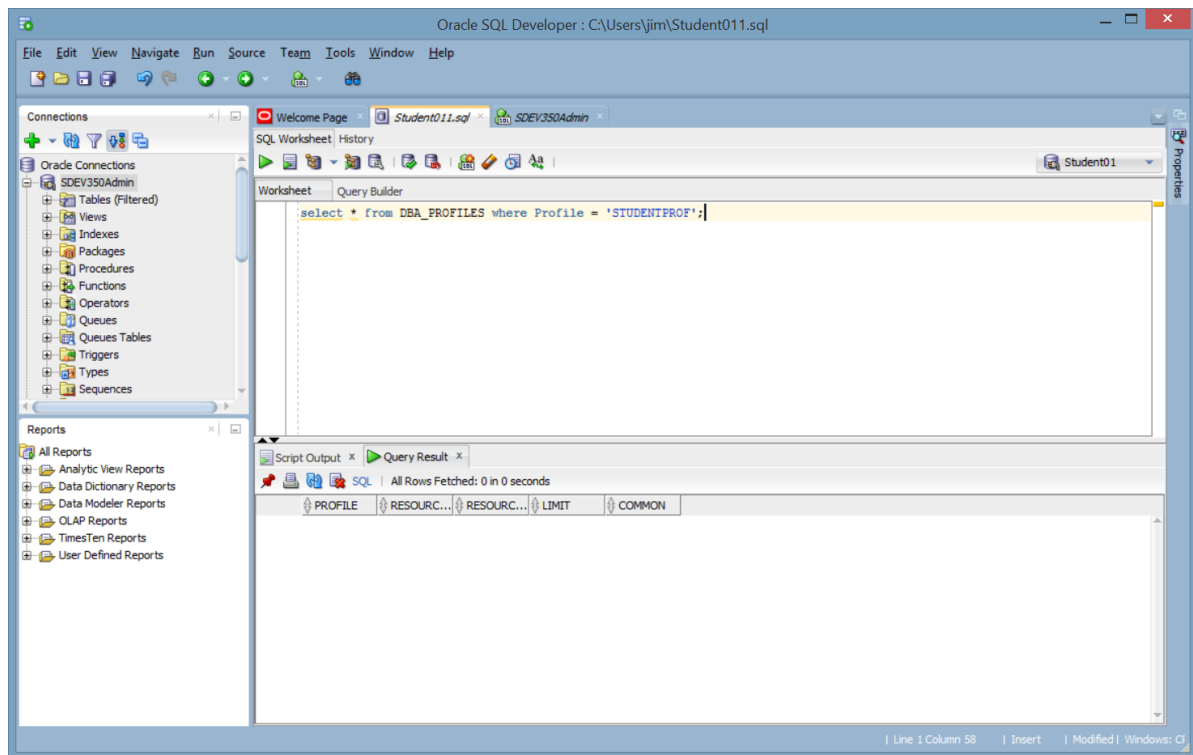


Figure 5 Verifying the Profile has been dropped