CREATE PROFILE

Purpose

Use the CREATE PROFILE statement to create a **profile**, which is a set of limits on database resources. If you assign the profile to a user, then that user cannot exceed these limits.

> See Also: *Oracle Database Security Guide* for a detailed description and explanation of how to use password management and protection

Prerequisites

To create a profile, you must have the CREATE PROFILE system privilege.

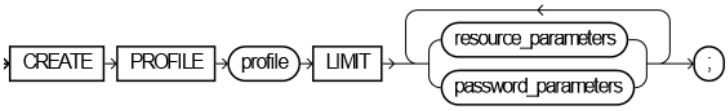
To specify resource limits for a user, you must:

- Enable resource limits dynamically with the ALTER SYSTEM statement or with the initialization parameter RESOURCE_LIMIT. This parameter does not apply to password resources. Password resources are always enabled.
- Create a profile that defines the limits using the CREATE PROFILE statement
- Assign the profile to the user using the CREATE USER or ALTER USER statement

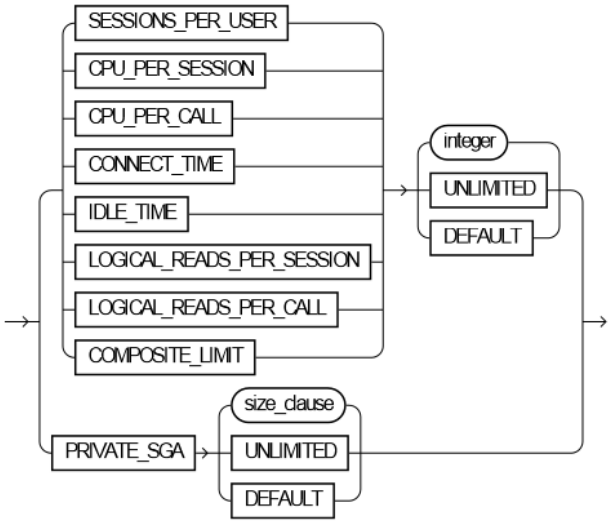
> See Also:
>
> - ALTER SYSTEM on page 11-60 for information on enabling resource limits dynamically
> - *Oracle Database Reference* for information on the RESOURCE_LIMIT parameter
> - CREATE USER on page 17-26 and ALTER USER on page 13-18 for information on profiles

Syntax

***create_profile*::=**

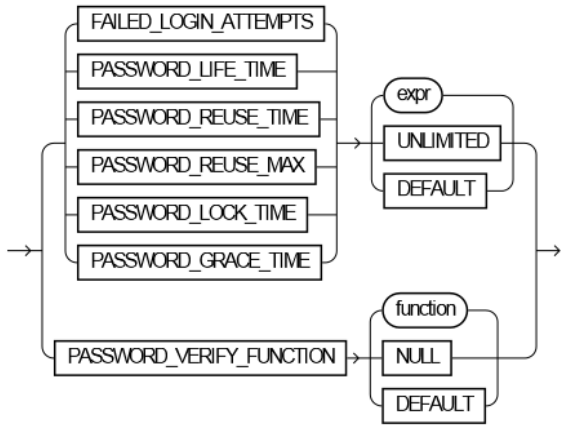


***resource_parameters*::=**



(*size_clause*::= on page 8-45

***password_parameters*::=**



Semantics

profile

Specify the name of the profile to be created. Use profiles to limit the database resources available to a user for a single call or a single session.

Oracle Database enforces resource limits in the following ways:

- If a user exceeds the `CONNECT_TIME` or `IDLE_TIME` session resource limit, then the database rolls back the current transaction and ends the session. When the user process next issues a call, the database returns an error.
- If a user attempts to perform an operation that exceeds the limit for other session resources, then the database aborts the operation, rolls back the current statement, and immediately returns an error. The user can then commit or roll back the current transaction, and must then end the session.

- If a user attempts to perform an operation that exceeds the limit for a single call, then the database aborts the operation, rolls back the current statement, and returns an error, leaving the current transaction intact.

Notes:

- You can use fractions of days for all parameters that limit time, with days as units. For example, 1 hour is 1/ 24 and 1 minute is 1/ 1440.
- You can specify resource limits for users regardless of whether the resource limits are enabled. However, Oracle Database does not enforce the limits until you enable them.

See Also: "Creating a Profile: Example" on page 15-58

UNLIMITED

When specified with a resource parameter, UNLIMITED indicates that a user assigned this profile can use an unlimited amount of this resource. When specified with a password parameter, UNLIMITED indicates that no limit has been set for the parameter.

DEFAULT

Specify DEFAULT if you want to omit a limit for this resource in this profile. A user assigned this profile is subject to the limit for this resource specified in the DEFAULT profile. The DEFAULT profile initially defines unlimited resources. You can change those limits with the ALTER PROFILE statement.

Any user who is not explicitly assigned a profile is subject to the limits defined in the DEFAULT profile. Also, if the profile that is explicitly assigned to a user omits limits for some resources or specifies DEFAULT for some limits, then the user is subject to the limits on those resources defined by the DEFAULT profile.

resource_parameters

SESSIONS_PER_USER Specify the number of concurrent sessions to which you want to limit the user.

CPU_PER_SESSION Specify the CPU time limit for a session, expressed in hundredth of seconds.

CPU_PER_CALL Specify the CPU time limit for a call (a parse, execute, or fetch), expressed in hundredths of seconds.

CONNECT_TIME Specify the total elapsed time limit for a session, expressed in minutes.

IDLE_TIME Specify the permitted 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 Specify the permitted number of data blocks read in a session, including blocks read from memory and disk.

LOGICAL_READS_PER_CALL Specify the permitted number of data blocks read for a call to process a SQL statement (a parse, execute, or fetch).

PRIVATE_SGA Specify the amount of private space a session can allocate in the shared pool of the system global area (SGA). Please refer to *size_clause* on page 8-45 for information on that clause.

Note: This limit applies only if you are using shared server architecture. The private space for a session in the SGA includes private SQL and PL/ SQL areas, but not shared SQL and PL/ SQL areas.

COMPOSITE_LIMIT Specify the total resource cost for a session, expressed in **service units**. Oracle Database calculates the total service units as a weighted sum of CPU_PER_SESSION, CONNECT_TIME, LOGICAL_READS_PER_SESSION, and PRIVATE_SGA.

See Also:

- ALTER RESOURCE COST on page 11-35 for information on how to specify the weight for each session resource
- "Setting Profile Resource Limits: Example" on page 15-58

password_parameters

Use the following clauses to set password parameters. Parameters that set lengths of time are interpreted in number of days. For testing purposes you can specify minutes (*n*/ 1440) or even seconds (*n*/ 86400).

FAILED_LOGIN_ATTEMPTS Specify the number of failed attempts to log in to the user account before the account is locked.

PASSWORD_LIFE_TIME Specify the number of days the same password can be used for authentication. If you also set a value for PASSWORD_GRACE_TIME, the password expires if it is not changed within the grace period, and further connections are rejected. If you do not set a value for PASSWORD_GRACE_TIME, its default of UNLIMITED will cause the database to issue a warning but let the user continue to connect indefinitely.

PASSWORD_REUSE_TIME and PASSWORD_REUSE_MAX These two parameters must be set in conjunction with each other. PASSWORD_REUSE_TIME specifies the number of days before which a password cannot be reused. PASSWORD_REUSE_MAX specifies the number of password changes required before the current password can be reused. For these parameter to have any effect, you must specify an integer for both of them.

- If you specify an integer for both of these parameters, then the user cannot reuse a password until the password has been changed the password the number of times specified for PASSWORD_REUSE_MAX during the number of days specified for PASSWORD_REUSE_TIME.

  For example, if you specify PASSWORD_REUSE_TIME to 30 and PASSWORD_REUSE_MAX to 10, then the user can reuse the password after 30 days if the password has already been changed 10 times.

- If you specify an integer for either of these parameters and specify UNLIMITED for the other, then the user can never reuse a password.
- If you specify DEFAULT for either parameter, then Oracle Database uses the value defined in the DEFAULT profile. By default, all parameters are set to UNLIMITED in the DEFAULT profile. If you have not changed the default setting of UNLIMITED in the DEFAULT profile, then the database treats the value for that parameter as UNLIMITED.
- If you set both of these parameters to UNLIMITED, then the database ignores both of them.

PASSWORD_LOCK_TIME Specify the number of days an account will be locked after the specified number of consecutive failed login attempts.

PASSWORD_GRACE_TIME Specify the number of days after the grace period begins during which a warning is issued and login is allowed. If the password is not changed during the grace period, the password expires.

PASSWORD_VERIFY_FUNCTION The PASSWORD_VERIFY_FUNCTION clause lets a PL/ SQL password complexity verification script be passed as an argument to the CREATE PROFILE statement. Oracle Database provides a default script, but you can create your own routine or use third-party software instead.

- For *function*, specify the name of the password complexity verification routine.
- Specify NULL to indicate that no password verification is performed.

If you specify *expr* for any of the password parameters, the expression can be of any form except scalar subquery expression.

See Also: "Setting Profile Password Limits: Example" on page 15-59

Examples

Creating a Profile: Example The following statement creates the profile new_profile:

```
CREATE PROFILE new_profile
  LIMIT PASSWORD_REUSE_MAX 10
        PASSWORD_REUSE_TIME 30;
```

Setting Profile Resource Limits: Example The following statement creates the profile app_user:

```
CREATE PROFILE app_user LIMIT
   SESSIONS_PER_USER          UNLIMITED
   CPU_PER_SESSION            UNLIMITED
   CPU_PER_CALL               3000
   CONNECT_TIME               45
   LOGICAL_READS_PER_SESSION  DEFAULT
   LOGICAL_READS_PER_CALL     1000
   PRIVATE_SGA                15K
   COMPOSITE_LIMIT            5000000;
```

If you assign the app_user profile to a user, the user is subject to the following limits in subsequent sessions:

- The user can have any number of concurrent sessions.
- In a single session, the user can consume an unlimited amount of CPU time.

- A single call made by the user cannot consume more than 30 seconds of CPU time.
- A single session cannot last for more than 45 minutes.
- In a single session, the number of data blocks read from memory and disk is subject to the limit specified in the DEFAULT profile.
- A single call made by the user cannot read more than 1000 data blocks from memory and disk.
- A single session cannot allocate more than 15 kilobytes of memory in the SGA.
- In a single session, the total resource cost cannot exceed 5 million service units. The formula for calculating the total resource cost is specified by the ALTER RESOURCE COST statement.
- Since the app_user profile omits a limit for IDLE_TIME and for password limits, the user is subject to the limits on these resources specified in the DEFAULT profile.

Setting Profile Password Limits: Example The following statement creates the app_user2 profile with password limits values set:

```
CREATE PROFILE app_user2 LIMIT
   FAILED_LOGIN_ATTEMPTS 5
   PASSWORD_LIFE_TIME 60
   PASSWORD_REUSE_TIME 60
   PASSWORD_REUSE_MAX 5
   PASSWORD_VERIFY_FUNCTION verify_function
   PASSWORD_LOCK_TIME 1/24
   PASSWORD_GRACE_TIME 10;
```

This example uses the default Oracle Database password verification function, verify_function. Please refer to *Oracle Database Security Guide* for information on using this verification function provided or designing your own verification function.