## EXAM OBJECTIVES: AUTOMATICALLY SWITCH A SESSION BACK TO THE ORIGINAL CONSUMER GROUP AT THE END OF THE TOP CALL

AN OVERVIEW OF DATABASE MANAGER

By using the Database Resource Manager, the database administrator (DBA) has more control over certain resource utilization than is normally possible through operating system resource management alone. With the Database Resource Manager, the DBA can guarantee groups of users a minimum amount of processing resources regardless of the load on the system and the number of users. The Database Resource Manager has been around since for a while now, and has made considerable enhancements with each release of Oracle.
As a very general level working with the database resource manager involves the following steps:

1. Create resource plans – a named means to allocate resources.
2. Create resource consumer groups – a group of users with similar resource needs.
3. Create resource plan directives – means of associating groups and resources to plans.
4. Assign users to consumer groups
5. Specify plan to be used by an instance

CHANGES IN ORACLE 10g

In the CREATE_PLAN_DIRECTIVE procedure that is part of the DBMS_RESOURCE_MANAGER package a number of new arguments have been introduced.
**SWITCH_BACK_ AT_CALL_END** parameter - reverts a session back to the original consumer group when the top call is completed.
**SWITCH_GROUP** parameter – indicates the consumer group to switch a user's session to on exceeding the value of the SWITCH_TIME directive value.
**SWITCH_TIME** parameter – the amount of time the user session remains in the original group. Upon exceeding the value specified for SWITCH_TIME the user it switched to the consumer group specified by the SWITCH_GROUP directive value.
Consider the example displayed below:

```
DBMS_RESOURCE_MANAGER.CREATE_PLAN_DIRECTIVE
(
 plan                        => 'Day_Plan' ,
 Group_or_subplan            => 'DSS_GROUP',
 Comment                     => 'Switch back example',
 Switch_group                => 'LONGRUN_GROUP',
```

```
 Switch_time                    => 600,
 Cpu_p1                         => 100,
 Cpu_p2                         => 0,
 Switch_back_at_call_end        => true
);

DBMS_RESOURCE_MANAGER.CREATE_PLAN_DIRECTIVE
(
Plan                            => 'Day_Plan',
Group_or_subplan                => 'LONGRUN_GROUP',
Comment                         => 'Switch Back Example',
Cpu_p1                          => 0,
Cpu_p2                          => 100,
 Switch_back_at_call_end        => true
);
```

Imagine a user with a username USERA logging into the database. Let us say that this user has been assigned an original consumer group as the DSS_GROUP. We will discuss shortly the topic of 'consumer group mappings' that tell how a user can be assigned to multiple consumer groups and which one is the original group.

When the user USERA issues a long running query that exceeds the value indicated by the SWITCH_TIME parameter, which in the example is 600, he will be automatically moved to the LONGRUN_GROUP consumer group. Since the SWITCH_BACK_AT_CALL_END has been set to TRUE, when the query completes the resource manager switches USERA back to the DSS_GROUP consumer group. Hence by this example we can see that the SWITCH_BACK_AT_CALL_END directive provides a way to reinitialize the resource usage state at a session after each call.

## EXAM OBJECTIVE: SET IDLE TIME-OUTS FOR CONSUMER GROUPS

In the CREATE_PLAN_DIRECTIVE there is another parameter can be used to limit the maximum time that a session can be idle. It is the MAX_IDLE_TIME directive.

The **MAX_IDLE_TIME** directive is a directive whose value is set in seconds indicating the maximum time that a session can be idle. When the session exceeds this limit, the PMON background process forcibly kills the sessions and cleans up its state.

Another directive, new in Oracle 10g is the **MAX_IDLE_BLOCKER_TIME.** This resource directive also takes a value in seconds and indicates the number of seconds to allow a session to be idle while blocking another session.

Consider the example given below:

```
DBMS_RESOURCE_MANAGER.CREATE_PLAN_DIRECTIVE
(
Plan                    => 'DAY_PLAN',
Group_or_subplan        =>'DSS_GROUP',
Comment                 => 'Limit idle time',
Max_idle_time           => 400,
Max_idle_blocker_time   => 100
);
```

In this example any user session that is idle for more than 400 seconds will be forcibly killed by the PMON background process.
Any user session that is idle for more than 100 seconds and blocking other sessions will be forcibly killed by the PMON background process.

## EXAM OBJECTIVE: CREATE MAPPINGS FOR THE AUTOMATIC ASSIGNMENT OF SESSIONS TO CONSUMER GROUPS

This is a new feature introduced in Oracle 10g. A consumer group mapping is a set of mapping from a session attribute's value to a consumer group. A session attribute could be to mention a few, the ORACLE_USER (indicates username that user specifies when connecting to the database), CLIENT_OS_USER (indicates the username that a user specifies when connecting to the operating system), SERVICE_NAME (indicates the name of the service the user is connecting to) or CLIENT_MACHINE (the host name of the machine).
A consumer group mapping can be created executing the **SET_CONSUMER_GROUP_MAPPING** procedure of the DBMS_RESOURCE_MANAGER package.

It takes the following arguments:

```
DBMS_RESOURCE_MANAGER.SET_CONSUMER_GROUP_MAPPING (
  attribute          IN VARCHAR2,
  value              IN VARCHAR2,
  consumer_group     IN VARCHAR2 DEFAULT NULL);
```

| | |
|---|---|
| attribute | The mapping attribute to add/modify. |
| value | The attribute value to match. |
| consumer_group | The name of the mapped consumer group, or NULL to delete a mapping. |

The mapping attribute mentioned below can be one of the following constants :

| | |
|---|---|
| client_machine | CONSTANT VARCHAR2(30) := 'CLIENT_MACHINE'; |
| client_os_user | CONSTANT VARCHAR2(30) := 'CLIENT_OS_USER'; |
| client_program | CONSTANT VARCHAR2(30) := 'CLIENT_PROGRAM'; |
| module_name | CONSTANT VARCHAR2(30) := 'MODULE_NAME'; |
| module_name_action | CONSTANT VARCHAR2(30) := 'MODULE_NAME_ACTION'; |
| oracle_user | CONSTANT VARCHAR2(30) := 'ORACLE_USER' |
| service_module | CONSTANT VARCHAR2(30) := 'SERVICE_MODULE'; |
| service_module_action | CONSTANT VARCHAR2(30) := 'SERVICE_MODULE_ACTION'; |
| service_name | CONSTANT VARCHAR2(30) := 'SERVICE_NAME'; |

Consider the example given below:

1) DBMS_RESOURCE_MANAGER.SET_CONSUMER_GROUP_MAPPING (DBMS_RESOURCE_MANAGER.ORACLE_USER, 'USERA', 'DSS_GROUP');

2) DBMS_RESOURCE_MANAGER.SET_CONSUMER_GROUP_MAPPING ( DBMS_RESOURCE_MANAGER.ORACLE_USER, 'USERZ','OLTP_GROUP');

3) DBMS_RESOURCE_MANAGER.SET_CONSUMER_GROUP_MAPPING ( DBMS_RESOURCE_MANAGER.CLIENT_OS_USER, 'OSUSERT', 'HR_GROUP');

In the first example the any user whose oracle username is USERA will be automatically assigned to the DSS_GROUP consumer group.

In the second example, any user whose oracle username is USERZ will be automatically assigned to the OLTP_GROUP consumer group.

In the third example, if the client's OS username is OSUSERT, then resource manager assigns the session to the HR_GROUP consumer group.

It is possible for a session to map to different consumer groups for different attributes. Resource Manager provides away to stipulate a priority level for each attribute to resolve such ambiguities.

Consider a situation where the user logs in to the client machine as the user OSUSERT and then connects to the database at the user USERZ. By default the attribute priorities are set so that the Oracle username takes precedence over the Client OS username. Hence, in this case the resource manager assigns the user to the OLTP_GROUP consumer group.

## MODIFYING ATTRIBUTE MAPPING PRIORITIES

In the example above, we discussed the default attribute priorities. However you can assign unique values to attributes indicating their priority. You can assign each attribute a unique value between 1 and 8, where 1 has a higher priority over the value 8.
You can use the SET_CONSUMER_GROUP_MAPPING_PRI procedure of the DBMS_RESOURCE_MANAGER package to do so.

The general syntax is:

```
DBMS_RESOURCE_MANAGER.SET_CONSUMER_GROUP_MAPPING_PRI
(
  explicit                IN NUMBER,
  oracle_user             IN NUMBER,
  service_name            IN NUMBER,
  client_os_user          IN NUMBER,
  client_program          IN NUMBER,
  client_machine          IN NUMBER,
  module_name             IN NUMBER,
  module_name_action      IN NUMBER,
  service_module          IN NUMBER,
  service_module_action   IN NUMBER
);
```

Given below is an example of the default values used for consumer group mapping.

```
DBMS_RESOURCE_MANAGER.SET_CONSUMER_GROUP_MAPPING_PRI
(explicit                 => 1,
Oracle_user               => 5,
Service_name              => 4,
Client_os_user            => 7,
Client_program            => 6,
Client_machine            => 8,
Module_name               => 3,
Module_name_action        => 2
```

);

You can notice in the example above that the ORACLE_USER attribute (5) has a higher priority over the CLIENT_OS_USER attribute (7).

If you modify it so the values are **interchanged** then in the example we were discussing the user logging in to the client machine as OSUSERT and to the database as USERZ will be assigned by resource manager to the HR_GROUP consumer group.