193
DBMS_SQLQ

The `DBMS_SQLQ` package provides the interface for configuring quarantine thresholds for execution plans of SQL statements. If any of the Resource Manager thresholds is equal to or less than the quarantine threshold specified in a SQL statement's quarantine configuration, then the SQL statement is not allowed to run, if it uses the execution plan specified in its quarantine configuration.

This chapter contains the following topics:

- DBMS_SQLQ Overview
- Summary of DBMS_SQLQ Subprograms

DBMS_SQLQ Overview

The `DBMS_SQLQ` package provides the interface for configuring quarantine thresholds for execution plans of SQL statements. If any of the Resource Manager thresholds is equal to or less than the quarantine threshold specified in a SQL statement's quarantine configuration, then the SQL statement is not allowed to run, if it uses the execution plan specified in its quarantine configuration.

You can use the `DBMS_SQLQ` package subprograms to:

- create quarantine configurations for execution plans of SQL statements and specify quarantine thresholds for their resource consumption
- query quarantine thresholds specified in quarantine configurations
- delete quarantine configurations
- transfer quarantine configurations from one database to another

Summary of DBMS_SQLQ Subprograms

This table lists the `DBMS_SQLQ` subprograms and briefly describes them.

Table 193-1 DBMS_SQLQ Package Subprograms

Procedure	Description
ALTER_QUARANTINE Procedure	Specifies a quarantine threshold in a quarantine configuration for execution plans of a SQL statement
CREATE_QUARANTINE_BY_SQL_ID Function	Creates a quarantine configuration for execution plans of a SQL statement using SQL ID
CREATE_QUARANTINE_BY_SQL_TEXT Function	Creates a quarantine configuration for execution plans of a SQL statement using SQL text
CREATE_STGTAB_QUARANTINE Procedure	Creates a staging table to store quarantine configurations
DROP_QUARANTINE Procedure	Deletes a quarantine configuration

Table 193-1 (Cont.) DBMS_SQLQ Package Subprograms

Procedure	Description
GET_PARAM_VALUE_QUARANTINE Function	Returns the value for a quarantine threshold specified in a quarantine configuration
PACK_STGTAB_QUARANTINE Function	Adds one or more quarantine configurations to a staging table
UNPACK_STGTAB_QUARANTINE Function	Creates quarantine configurations in a database from a staging table

ALTER_QUARANTINE Procedure

This procedure specifies a quarantine threshold for a resource in a quarantine configuration for execution plans of a SQL statement.

Syntax

```
DBMS_SQLQ.ALTER_QUARANTINE (
   quarantine_name   IN VARCHAR2,
   parameter_name    IN VARCHAR2,
   parameter_value   IN VARCHAR2);
```

Parameters

Table 193-2 ALTER_QUARANTINE Procedure Parameters

Parameter	Description
quarantine_name	Name of the quarantine configuration.
parameter_name	Name of the resource for which quarantine threshold needs to be specified. You can specify any one of the following values: • CPU_TIME: CPU time • ELAPSED_TIME: Elapsed time • IO_MEGABYTES: I/O in megabytes • IO_REQUESTS: Number of physical I/O requests • IO_LOGICAL: Number of logical I/O requests • ENABLED: Flag to enable or disable the quarantine configuration. Specify YES to enable it and NO to disable it. The default value is YES. • AUTOPURGE: Flag to enable or disable automatic purging of the quarantine configuration. If it is set to YES, the quarantine configuration is automatically purged after 53 weeks, if not used. If it is set to NO, the quarantine configuration is never purged. The default value is YES.
parameter_value	Quarantine threshold for the resource specified in parameter_name.

Examples

In the following example, the quarantine threshold specified for CPU time is 5 seconds and elapsed time is 10 seconds for the quarantine configuration `SQL_QUARANTINE_3z0mwuq3aqsm8cfe7a0e4`.

```
BEGIN

  DBMS_SQLQ.ALTER_QUARANTINE(
   QUARANTINE_NAME  =>  'SQL_QUARANTINE_3z0mwuq3aqsm8cfe7a0e4',
   PARAMETER_NAME   =>  'CPU_TIME',
   PARAMETER_VALUE  =>  '5');

  DBMS_SQLQ.ALTER_QUARANTINE(
   QUARANTINE_NAME  =>  'SQL_QUARANTINE_3z0mwuq3aqsm8cfe7a0e4',
   PARAMETER_NAME   =>  'ELAPSED_TIME',
   PARAMETER_VALUE  =>  '10');

END;
/
```

When the SQL statement is executed using the execution plan specified in the quarantine configuration, and if the Resource Manager threshold for CPU time is 5 seconds or less, or elapsed time is 10 seconds or less, then the SQL statement is not allowed to run.

CREATE_QUARANTINE_BY_SQL_ID Function

This function creates a quarantine configuration for execution plans of a SQL statement based on SQL ID.

Syntax

```
DBMS_SQLQ.CREATE_QUARANTINE_BY_SQL_ID (
   sql_id           IN VARCHAR2,
   plan_hash_value  IN NUMBER DEFAULT NULL)
RETURN VARCHAR2;
```

Parameters

Table 193-3 CREATE_QUARANTINE_BY_SQL_ID Function Parameters

Parameter	Description
`sql_id`	SQL ID of the SQL statement.
`plan_hash_value`	Hash value of the execution plan of the SQL statement. Default value is `NULL`. When it is `NULL`, the quarantine configuration applies to all the execution plans of the SQL statement.

Return Value

Name of the quarantine configuration.

Examples

The following example creates a quarantine configuration for the SQL statement having the SQL ID of `8vu7s907prbgr`. The quarantine configuration applies to all the execution plans of the SQL statement.

```
DECLARE
  quarantine_config VARCHAR2(30);
BEGIN
  quarantine_config := DBMS_SQLQ.CREATE_QUARANTINE_BY_SQL_ID(SQL_ID => '8vu7s907prbgr');
END;
/
```

The following example creates a quarantine configuration for the execution plan having the hash value of `3488063716` for the SQL statement having the SQL ID of `8vu7s907prbgr`.

```
DECLARE
  quarantine_config VARCHAR2(30);
BEGIN
  quarantine_config := DBMS_SQLQ.CREATE_QUARANTINE_BY_SQL_ID(SQL_ID => '8vu7s907prbgr',
PLAN_HASH_VALUE => '3488063716');
END;
/
```

CREATE_QUARANTINE_BY_SQL_TEXT Function

This function creates a quarantine configuration for execution plans of a SQL statement based on SQL text.

Syntax

```
DBMS_SQLQ.CREATE_QUARANTINE_BY_SQL_TEXT (
   sql_text         IN CLOB,
   plan_hash_value  IN NUMBER DEFAULT NULL)
RETURN VARCHAR2;
```

Parameters

Table 193-4 CREATE_QUARANTINE_BY_SQL_TEXT Function Parameters

Parameter	Description
`sql_text`	SQL statement.
`plan_hash_value`	Hash value of the execution plan of the SQL statement. Default value is `NULL`. When it is `NULL`, the quarantine configuration applies to all the execution plans of the SQL statement.

Return Value

Name of the quarantine configuration.

Examples

The following example creates a quarantine configuration that applies to all the execution plans of the SQL statement `'select count(*) from emp'`.

```
DECLARE
  quarantine_config VARCHAR2(30);
BEGIN
  quarantine_config := DBMS_SQLQ.CREATE_QUARANTINE_BY_SQL_TEXT(SQL_TEXT =>
to_clob('select count(*) from emp'));
END;
/
```

The following example creates a quarantine configuration for the execution plan having the hash value of `3488063716` for the SQL statement having the SQL text of `'select count(*) from emp'`.

```
DECLARE
  quarantine_config VARCHAR2(30);
BEGIN
  quarantine_config := DBMS_SQLQ.CREATE_QUARANTINE_BY_SQL_TEXT(SQL_TEXT =>
to_clob('select count(*) from emp'), PLAN_HASH_VALUE => '3488063716');
END;
/
```

CREATE_STGTAB_QUARANTINE Procedure

This procedure creates a staging table to store the quarantine configurations, so that the staging table can be exported from the current database and imported into another database, thus enabling the quarantine configurations to be used across databases.

Syntax

```
DBMS_SQLQ.CREATE_STGTAB_QUARANTINE (
   staging_table_name   IN VARCHAR2,
   staging_table_owner  IN VARCHAR2 DEFAULT NULL,
   tablespace_name      IN VARCHAR2 DEFAULT NULL);
```

Parameters

Table 193-5 CREATE_STGTAB_QUARANTINE Procedure Parameters

Parameter	Description
`staging_table_name`	Name of the staging table.
`staging_table_owner`	Name of the schema owner of the staging table. Default value is `NULL`, which means the database user executing this procedure is set as the staging table owner.
`tablespace_name`	Name of the tablespace in which the staging table needs to be created. Default value is `NULL`, which means the staging table is created in the default tablespace of the database.

Examples

The following example creates the staging table `TBL_STG_QUARANTINE` in the default tablespace of the database and sets its table owner to the database user executing this procedure.

```
BEGIN
  DBMS_SQLQ.CREATE_STGTAB_QUARANTINE(STAGING_TABLE_NAME => 'TBL_STG_QUARANTINE');
END;
/
```

DROP_QUARANTINE Procedure

This procedure deletes a quarantine configuration.

Syntax

```
DBMS_SQLQ.DROP_QUARANTINE(quarantine_name IN VARCHAR2);
```

Parameters

Table 193-6 DROP_QUARANTINE Procedure Parameters

Parameter	Description
`quarantine_name`	Name of the quarantine configuration to delete.

Examples

The following example deletes the quarantine configuration having the name `SQL_QUARANTINE_3z0mwuq3aqsm8cfe7a0e4`.

```
BEGIN
  DBMS_SQLQ.DROP_QUARANTINE('SQL_QUARANTINE_3z0mwuq3aqsm8cfe7a0e4');
END;
/
```

GET_PARAM_VALUE_QUARANTINE Function

This function returns the quarantine threshold for a resource specified in a quarantine configuration.

Syntax

```
DBMS_SQLQ.GET_PARAM_VALUE_QUARANTINE (
   quarantine_name    IN VARCHAR2,
   parameter_name     IN VARCHAR2)
RETURN VARCHAR2;
```

Parameters

Table 193-7 GET_PARAM_VALUE_QUARANTINE Function Parameters

Parameter	Description
quarantine_name	Name of the quarantine configuration.
parameter_name	Resource for which the quarantine threshold needs to be retrieved.

Return Value

Returns the quarantine threshold for a resource specified in a quarantine configuration.

Examples

The following example returns the quarantine threshold for CPU time specified in the quarantine configuration having the name SQL_QUARANTINE_3z0mwuq3aqsm8cfe7a0e4.

```
DECLARE
  quarantine_config_setting_value VARCHAR2(30);
BEGIN
  quarantine_config_setting_value := DBMS_SQLQ.GET_PARAM_VALUE_QUARANTINE(
                                       QUARANTINE_NAME  =>
'SQL_QUARANTINE_3z0mwuq3aqsm8cfe7a0e4',
                                       PARAMETER_NAME   =>  'CPU_TIME');
END;
/
```

PACK_STGTAB_QUARANTINE Function

This function adds one or more quarantine configurations into a staging table.

Syntax

```
DBMS_SQLQ.PACK_STGTAB_QUARANTINE (
   staging_table_name    IN VARCHAR2,
   staging_table_owner   IN VARCHAR2 DEFAULT NULL,
   name                  IN VARCHAR2 DEFAULT '%',
   sql_text              IN VARCHAR2 DEFAULT '%',
   enabled               IN VARCHAR2 DEFAULT NULL)
RETURN NUMBER;
```

Parameters

Table 193-8 PACK_STGTAB_QUARANTINE Function Parameters

Parameter	Description
staging_table_name	Name of the staging table in which the quarantine configurations need to be added.
staging_table_owner	Name of the schema owner of the staging table. Default value is NULL, which means the database user executing this procedure is set as the staging table owner.

Table 193-8 (Cont.) PACK_STGTAB_QUARANTINE Function Parameters

Parameter	Description
name	Name of the quarantine configuration. Its value is case-sensitive and it accepts wildcard characters.
sql_text	SQL statement text. Its value is case-sensitive and it accepts wildcard characters.
enabled	Flag indicating whether the quarantine configuration should be enabled or disabled. If it is set to YES, then the quarantine configuration is enabled, else it is disabled. Default value is NULL, which means the quarantine configuration is disabled by default.

Return Value

Number of quarantine configurations added to the staging table.

Examples

The following example adds all the quarantine configurations having the names starting with SQL_QUARANTINE_ into the staging table TBL_STG_QUARANTINE.

```
DECLARE
  quarantine_configs NUMBER;
BEGIN
  quarantine_configs := DBMS_SQLQ.PACK_STGTAB_QUARANTINE(
                          STAGING_TABLE_NAME => 'TBL_STG_QUARANTINE',
                          NAME => 'SQL_QUARANTINE_%');
END;
/
```

UNPACK_STGTAB_QUARANTINE Function

This function creates quarantine configurations in a database from a staging table.

Syntax

```
DBMS_SQLQ.UNPACK_STGTAB_QUARANTINE (
   staging_table_name    IN VARCHAR2,
   staging_table_owner   IN VARCHAR2 DEFAULT NULL,
   name                  IN VARCHAR2 DEFAULT '%',
   sql_text              IN VARCHAR2 DEFAULT '%',
   enabled               IN VARCHAR2 DEFAULT NULL)
RETURN NUMBER;
```

Parameters

Table 193-9 UNPACK_STGTAB_QUARANTINE Function Parameters

Parameter	Description
staging_table_name	Name of the staging table from which the quarantine configurations need to be created in the database.
staging_table_owner	Name of the schema owner of the staging table. Default value is NULL, which means the database user executing this procedure is set as the staging table owner.
name	Name of the quarantine configuration. Its value is case-sensitive and it accepts wildcard characters.
sql_text	SQL statement text. Its value is case-sensitive and it accepts wildcard characters.
enabled	Flag indicating whether the quarantine configuration should be enabled or disabled. If its value is YES, then the quarantine configuration is enabled, else it is disabled. Default value is NULL, which means the quarantine configuration is disabled by default.

Return Value

Number of quarantine configurations created in the database from the staging table.

Examples

The following example creates the quarantine configurations in the database from all the quarantine configurations stored in the staging table TBL_STG_QUARANTINE.

```
DECLARE
  quarantine_configs NUMBER;
BEGIN
  quarantine_configs := DBMS_SQLQ.UNPACK_STGTAB_QUARANTINE(
                                    STAGING_TABLE_NAME => 'TBL_STG_QUARANTINE');
END;
/
```