Oracle® Banking Treasury Management

Database Practices

Release 14.7.0.0.0
F71204-01
November 2022

Oracle Banking Treasury Management Database Practices, Release 14.7.0.0.0

F71204-01

Contents

Preface

This guide helps the user to define standard database initialization parameters that have been derived after performing the required benchmark tests (Performance Load tests).

The images used in the documentation are of illustration purpose and need to be used only for reference.

Audience

This guide is intended for anyone responsible for installing Oracle Banking Application.

Acronyms and Abbreviations

The acronyms and abbreviations are listed in this below table:

Table 1 Acronyms and Abbreviations

Abbreviations or Acronyms	Definition
DV	Derivatives
ETD	Exchange Traded Derivatives
FX	Foreign Exchange
MM	Money Market
OBTR	Oracle Banking Treasury Management
ODT	Open Development Tool
OT	Over the Counter Options
SE	Securities
SR	Securities Repo

Related Resources

For more information, see these Oracle Banking Treasury Management resources:

- *Oracle Banking Treasury Management Release Notes*
- *Oracle Banking Treasury Management Installer Index*

- *Oracle Banking Treasury Management Installer Prerequisite*

1

Database Initialization Parameters

Oracle standard database initialization parameters have been derived after performing the required benchmark tests (Performance Load tests). We recommend installing Oracle Banking Treasury Management in the Pluggable database and a few of these parameters need to be set at the PDB level.

> **Note:**
>
> Since some of the initialization parameters values are specific to customer volume, parameters should be derived using the Oracle Banking Treasury Management-Disk-Layouts-initparams-19c (the excel is in macros) excel sheet baselined along with this document.

Following are the Parameters with the details and their relevance to Oracle Banking Treasury Management:

ALLOW_LEVEL_WITHOUT_CONNECT_BY

Recommended Value: TRUE

The ALLOW_LEVEL_WITHOUT_CONNECT_BY parameter is set to avoid following error:

```
After Upgrading To Oracle 10g, Getting ORA-01788 When Running A
Query That Includes The LEVEL Pseudo Column [ID 455953.1]
```

CURSOR_SHARING

The cursor sharing parameter determines the kind of SQL statements can the same cursors share.

Table 1-1 Cursor Sharing

Property	Description
Parameter Type	String
Default Value	EXACT
Recommended Value	Force

Oracle Banking Treasury Management relevance

Some of the Oracle Banking Treasury Management SQL statements are generated dynamically. So they contain literal values in the where clause conditions. It results in many identical statements with separate parse trees in Oracle's library cache, which can slow performance and cause latch problems.

By setting, the cursor_sharing to the FORCE database, it converts literals to bind variables before parsing the statement.

DB_CACHE_ADVICE

The DB_CACHE_ADVICE parameter enables or disables the statistics gathered for predicting behavior with different cache sizes through the V$DB_CACHE_ADVICE performance view.

Table 1-2 DB_CACHE_ADVICE

Property	Description
Parameter Type	String
Syntax	DB_CACHE_ADVICE = { ON | READY | OFF }
Default Value	If STATISTICS_LEVEL is set to TYPICAL / ALL, then **ON**. If STATISTICS_LEVEL is set to BASIC, then **OFF**.
Recommended Value	**OFF** (Should be **ON** while Performance Monitoring)

Oracle Banking Treasury Management Relevance

Turning ON advisory will have an extra overhead.

> **Note:**
>
> It should be **ON**, only during performance monitoring.

FAST_START_MTTR_TARGET

The FAST_START_MTTR_TARGET parameter enables you to specify the number of seconds the database takes to perform crash recovery of a single instance. When specified, FAST_START_MTTR_TARGET is overridden by LOG_CHECKPOINT_INTERVAL.

Table 1-3 FAST_START_MTTR_TARGET

Property	Description
Parameter Type	Integer
Default Value	0
Range of Values	0 to 3600 seconds
Recommended Value	300

Oracle Banking Treasury Management Relevance

If FAST_START_MTTR_TARGET is not set, to 300 then run time performance for write/redo generation intensive workloads will not be optimized. It will reduce checkpoint writes from DBWR processes, making more room for LGWR IO. To optimize run time performance for write/redo generation intensive workloads, increase the FAST_START_MTTR_TARGET initialization parameter to 300.

JOB_QUEUE_PROCESSES

The JOB_QUEUE_PROCESSES parameter specifies the maximum number of processes that can be created for the execution of jobs. It specifies the number of job queue processes per instance (J000, J999).

Table 1-4 JOB_QUEUE_PROCESSES

Property	Description
Parameter Type	Integer
Default Value	4000
Range of Values	0 to 1000
Recommended Value	Refer Oracle Banking Treasury Management-Disk-Layouts-initparams-19c.xlsx(reference to be given)

Oracle Banking Treasury Management Relevance

The JOB_QUEUE_PROCESSES parameter has to be set with respect to the maximum number of scheduler jobs. To arrive at the right value, refer to Oracle Banking Treasury Management-Disk-Layouts-initparams-19c.xlsx excel (reference to be given).

LOG_BUFFER

Recommended Value: Refer Oracle Banking Treasury Management-Disk-Layouts-initparams-19c.xlsx (reference to be given)

Oracle Banking Treasury Management Relevance

The default log buffer size is too small as Oracle Banking Treasury Management performs heavy DML during batch processing.

MEMORY TARGET/MEMORY_MAX_TARGET

Recommended Value: Refer Oracle Banking Treasury Management-Disk-Layouts-initparams-19c.xlsx (reference to be given)

For Linux systems, make sure that the value of operating system `/dev/shm` mount is set to the appropriate value to accommodate memory_Target.

NLS_DATE_FORMAT

The NLS_DATE_FORMAT parameter specifies the default date format to use with the TO_CHAR and TO_DATE functions.

Table 1-5 NLS_DATE_FORMAT

Property	Description
Parameter Type	String
Syntax	NLS_DATE_FORMAT = "format"
Default Value	Derived from NLS_TERRITORY
Recommended Value	DD-MON-RRRR

Oracle Banking Treasury Management Relevance

Oracle Banking Treasury Management standard date format.

OPEN_CURSORS

The OPEN_CURSORS parameter specifies the maximum number of open cursors (handles to private SQL areas) a session can have at once. You can use this parameter to prevent a session from opening an excessive number of cursors.

Table 1-6 JOB_QUEUE_PROCESSES

Property	Description
Parameter Type	Integer
Default Value	50
Modifiable	ALTER SYSTEM
Range of Values	1 to 4294967295 (4 GB -1)
Recommended Value	5000

Oracle Banking Treasury Management Relevance

The value of OPEN_CURSORS is required to prevent the Oracle Banking Treasury Management application from running out of open cursors. `ORA-01000: maximum open cursors exceeded`

OPTIMIZER_DYNAMIC_SAMPLING

The OPTIMIZER_DYNAMIC_SAMPLING parameter controls the level of dynamic sampling performed by the optimizer.

Table 1-7 OPTIMIZER_DYNAMIC_SAMPLING

Property	Description
Parameter Type	Integer
Default Value	• If OPTIMIZER_FEATURES_ENABLE is set to 10.0.0 or higher, then 2. • If OPTIMIZER_FEATURES_ENABLE is set to 9.2.0, then 1. • If OPTIMIZER_FEATURES_ENABLE is set to 9.0.1 or lower, then 0.
Recommended Values	1
Range of Values	0 to 10

Oracle Banking Treasury Management Relevance

Dynamic Sampling is a method of gathering additional statistics during optimization by recursively sampling statements. When the dynamic sampling is enabled, queries are recursively generated by Oracle to test various selectivity based upon the real values to improve their accuracy. This can result in the production of better explain plans.

Value 1 Sample all tables that have not been analyzed that meet certain criteria.

OPTIMIZER_INDEX_CACHING

The OPTIMIZER_INDEX_CACHING parameter allows to adjust the behavior of cost-based optimization to favor nested loops joins and IN-list iterators.

Table 1-8 OPTIMIZER_DYNAMIC_SAMPLING

Property	Description
Parameter Type	Integer
Default Value	0
Recommended Values	90
Range of Values	0 to 100

Oracle Banking Treasury Management Relevance

The cost of executing an index using IN-list iterators or of executing nested loops join when an index is used to access the inner table depends on the caching of that index in the buffer cache. Oracle Banking Treasury Management favors nested loop joins by setting optimizer_index_caching to 90.

OPTIMIZER_INDEX_COST_ADJ

The OPTIMIZER_INDEX_COST_ADJ parameter allows to tune optimizer behavior for access path selection to be more or less index friendly - that is, to make the optimizer more or less prone to selecting an index access path over a full table scan.

Table 1-9 OPTIMIZER_DYNAMIC_SAMPLING

Property	Description
Parameter Type	Integer
Default Value	100
Recommended Values	50
Range of Values	1 to 10000

Oracle Banking Treasury Management Relevance

Oracle Banking Treasury Management favors index read over full table scan as it is very useful when optimizer favors to give a lower cost to index scans over full-table scans.

PARALLEL_MAX_SERVERS

The PARALLEL_MAX_SERVERS parameter specifies the maximum number of parallel execution processes and parallel recovery processes in an instance. As demand increases, Oracle Database increases the number of processes from the number created at instance startup up to this value.

Table 1-10 PARALLEL_MAX_SERVERS

Property	Description
Parameter Type	Integer
Default Value	Derived from the values of CPU_COUNT, PARALLEL_THREADS_PER_CPU, and PGA_AGGREGATE_TARGET
Recommended Values	Refer Oracle Banking Treasury Management-Disk-Layouts-initparams-19c.xlsx (need to give the reference)
Range of Values	0 to 3600
Real Application Clusters	Multiple instances can have different values

Oracle Banking Treasury Management Relevance

To arrive right value, refer to Oracle Banking Treasury Management-Disk-Layouts-initparams-19c.xlsx excel. (need to give the reference)

PGA_AGGREGATE_LIMIT

Recommended Value: 0

Oracle Banking Treasury Management Relevance

Setting this parameter limits the pga consumed by the instance, hence might cause failure to a few of the running processes.

PLSQL_CODE_TYPE

The PLSQL_CODE_TYPE parameter specifies the compilation mode of the PL/SQL units.

Table 1-11 PLSQL_CODE_TYPE

Property	Description
Parameter Type	String
Default Value	INTERPRETED
Recommended Values	NATIVE
Range of Values	INTERPRETED, NATIVE

Oracle Banking Treasury Management Relevance

The PL/SQL interpreter overhead will be minimal when set to NATIVE.

PROCESSES

The PROCESSES parameter specifies the maximum number of operating system user processes that can simultaneously connect to Oracle. Its value should allow for all background processes such as locks, job queue processes, and parallel execution processes.

Table 1-12 PROCESSES

Property	Description
Parameter Type	Integer
Default Value	100
Range of Values	6 to operating system dependent
Recommended Values	Refer to Oracle Banking Treasury Management-Disk-Layouts-initparams-19c.xlsx

Oracle Banking Treasury Management Relevance

This parameter can set be set with respect to maximum no of sessions connected to DB.

REMOTE_DEPENDENCIES_MODE

The REMOTE_DEPENDENCIES_MODE parameter specifies how Oracle should handle dependencies upon remote PL/SQL stored procedures.

Table 1-13 REMOTE_DEPENDENCIES_MODE

Property	Description
Parameter Type	String
Syntax	REMOTE_DEPENDENCIES_MODE = { TIMESTAMP | SIGNATURE }
Default Value	TIMESTAMP
Recommended Values	SIGNATURE

Oracle Banking Treasury Management Relevance

Oracle allows the procedure to execute as long as the signatures are considered safe. This setting allows client PL/SQL applications to run without recompilation.

SESSION_CACHED_CURSORS

The SESSION_CACHED_CURSORS parameter specifies the number of session cursors to cache. Repeated parse calls of the same SQL statement cause the session cursor for that statement to be moved into the session cursor cache. Subsequent parse calls will find the cursor in the cache and do not need to reopen the cursor. Oracle uses a least recently used algorithm to remove entries in the session cursor cache to make room for new entries when needed.

Table 1-14 SESSION_CACHED_CURSORS

Property	Description
Parameter Type	Integer
Default Value	100
Recommended Values	400
Range of Values	0 to operating system-dependent

Oracle Banking Treasury Management Relevance

The SESSION_CACHED_CURSORS parameter helps to cache the cursor thus avoid parsing of the cursor which is heavy CPU intensive, particularly in batch.

SKIP_UNUSABLE_INDEXES

The SKIP_UNUSABLE_INDEXES parameter enables or disables the use and reporting of tables with unusable indexes or index partitions.

Table 1-15 SKIP_UNUSABLE_INDEXES

Property	Description
Parameter Type	Boolean
Default Value	True
Recommended Values	FALSE

Table 1-15 (Cont.) SKIP_UNUSABLE_INDEXES

Property	Description
Range of Values	true / false

Oracle Banking Treasury Management Relevance

TRUE enables error reporting of indexes marked UNUSABLE. This setting does not allow inserts, deletes, and updates on tables with unusable indexes or index partitions. It is set to false because the Oracle Banking Treasury Management application should throw an error if any of the indexes become UNUSABLE.

UNDO_RETENTION

This UNDO_RETENTION parameter specifies (in seconds) the low threshold value of undo retention. For AUTOEXTEND undo tablespaces, the system retains for at least the time specified in this parameter and automatically tunes the undo retention period to satisfy the undo requirements of the queries. For fixed-size undo tablespaces, the system automatically tunes for the maximum possible undo retention period, based on undo tablespace size and usage history, and ignores UNDO_RETENTION unless retention guarantee is enabled.

The UNDO_RETENTION parameter can only be honored if the current undo tablespace has enough space. If an active transaction requires to undo space, and the undo tablespace does not have available space, then the system starts reusing unexpired undo space. This action can potentially cause some queries to fail with a "snapshot too old" message.

Table 1-16 UNDO_RETENTION

Property	Description
Parameter Type	Integer
Default Value	900
Range of Values	0 to 231 – 1
Recommended Values	1800

Oracle Banking Treasury Management Relevance

Increased value along with automatic undo management helps to avoid `snapshot too old error`.

2
Database Patches Required

Patch 24423416 needs to be applied for the respective platform and below events need to be set in the parameter file

event='10946 trace name context level 8454144','10934 trace name context level 2048'

This relates to a known issue where PLSQL compilation takes time. Setting this event with the patch applied will help to improve the performance of PLSQL compilation performance and storage optimization.

3
Redo Log Files

The default number of redo log files groups and its size is inadequate to run FLEXCUBE. Use the following recommendations:

- 6 redo log groups
- Redo log file size
  - 1 GB each for the DB size up to 1 TB
  - 2 GB each for DB size more than 1 TB

4
PLSQL Optimizer Level

The plsql_optimize_level value for all the PLSQL units should be the same, which would be the value set in the plsql_optimize_level init parameter.

Following SQL provides the PLSQL optimizer level for schema PLSQL units:

```
Select PLSQL_OPTIMIZE_LEVEL,type,count(*) "Count" from
user_plsql_object_settings group by PLSQL_OPTIMIZE_LEVEL,type;
```

PLSQL_OPTIMIZE_LEVEL for all the objects should be the same as the value set in plsql_optimize_level init parameter. If there is a difference, then the objects should be recompiled. It can be done using dbms_utility.compile_schema procedure.

> **Note:**
>
> The **dbms_utility.compile_schema** procedure invalidates and recompiles all the plsql units.

exec dbms_utility.compile_schema('FCCBM2')

Here, **FCCBM2** refers to the schema.

5

Statistics Collection for Schema (Recommended Method)

This topic explains in the statistics collection for schema.

Oracle 19c provides a default scheduled job to collect statistics for the entire database and is default scheduled to run every night. Given that the batch as well runs in the night. It is critical that the statistics gathering is not run during the batch.

It is recommended to use the default database scheduled job that is shipped with Oracle Database to collect statistics for Schema.

Statistics collection recommendation is specific to the schema. Other available statistics like System statistics, Fixed table statistics, dictionary statistics, etc., are not part of this recommendation and are required to be executed on a need basis.

> **Note:**
>
> This document assumes that there is no other tool or program is scheduled to collect statistics for the Database.

Customize Default Statistics Collection Schedule

The Default Scheduler is to be customized for the following:

- Ensure that the default statistics gathering program is configured and Running.

  ```
  SELECT STATUS
  FROM DBA_AUTOTASK_CLIENT
  WHERE CLIENT_NAME='auto optimizer stats collection';
  ```

  Should return - ENABLED

- Ensure that the default statistics gathering program is configured to run only on weekends.
  /* Start of Script – Script to be executed as SYS*/

  ```
  BEGIN
  DBMS_AUTO_TASK_ADMIN.ENABLE
  (CLIENT_NAME => 'auto optimizer stats collection',
  OPERATION => NULL,
  WINDOW_NAME => 'SATURDAY_WINDOW');
  DBMS_AUTO_TASK_ADMIN.ENABLE
  (CLIENT_NAME => 'auto optimizer stats collection',
  OPERATION => NULL,
  WINDOW_NAME => SUNDAY_WINDOW');
  ```

```
END;
/
```

/* End of Script */

- The default schedule is daily. So, disable the daily schedules for optimizer statistics.
  /* Start of Script – Script to be executed as SYS*/

```
BEGIN
DBMS_AUTO_TASK_ADMIN.DISABLE
(CLIENT_NAME => 'auto optimizer stats collection',
OPERATION => NULL,
WINDOW_NAME => 'MONDAY_WINDOW');
DBMS_AUTO_TASK_ADMIN.DISABLE
(CLIENT_NAME =>'auto optimizer stats collection',
OPERATION => NULL,
WINDOW_NAME => 'TUESDAY_WINDOW');
DBMS_AUTO_TASK_ADMIN.DISABLE
(CLIENT_NAME =>'auto optimizer stats collection',
OPERATION => NULL,
WINDOW_NAME => 'WEDNESDAY_WINDOW');
DBMS_AUTO_TASK_ADMIN.DISABLE
(CLIENT_NAME => 'auto optimizer stats collection',
OPERATION => NULL,
WINDOW_NAME => 'THURSDAY_WINDOW');
DBMS_AUTO_TASK_ADMIN.DISABLE
(CLIENT_NAME => 'auto optimizer stats collection',
OPERATION => NULL,
WINDOW_NAME => 'FRIDAY_WINDOW');
END;
/
```

/* End of Script */

Verify the setup using the following SQL:

```
SELECT WINDOW_NAME,OPTIMIZER_STATS FROM
DBA_AUTOTASK_WINDOW_CLIENTS;
```

MONDAY_WINDOW	DISABLED
TUESDAY_WINDOW	DISABLED
WEDNESDAY_WINDOW	DISABLED
THURSDAY_WINDOW	DISABLED
FRIDAY_WINDOW	DISABLED
SATURDAY_WINDOW	ENABLED
SUNDAY_WINDOW	ENABLED

Customize Statistics Gathering

The default statistics gathering is designed to be generic. It is recommended to customize the default statistics gathering to suit online and batch.

Following are the areas that would need customization for the Oracle Banking Treasury Management:

Statistics Histograms

Configure the default statistics gathered without Histograms.

/* Start of Script – Script to be executed as SYS*/

```
BEGIN
DBMS_STATS.SET_PARAM ('METHOD_OPT','FOR ALL COLUMNS SIZE 1');
END;
/
```

/*End of Script */

Verify the setup using the following SQL:

```
SELECT DBMS_STATS.GET_PARAM ('METHOD_OPT') FROM DUAL;
```

Value should return as:

```
FOR ALL COLUMNS SIZE 1
```

6

Database Storage Recommendations

This topic explains the Oracle Banking Treasury Management database storage recommendations.

Oracle database 10g release 2 onwards, Automatic Storage Management (ASM) is the recommended storage option for the Oracle Banking Treasury Management database. ASM is an integrated cluster-aware volume manager and a file system designed and optimized for managing Oracle database files. ASM is the realization of the Oracle Stripe and Mirror Everything (SAME) storage management methodology researched and established as best practices for the Oracle database environment over many years.

> **Note:**
>
> For configuring ASM refer Automatic storage management best practice document provided by Oracle for your database version.

Key benefits of ASM

- I/O is spread evenly across all available disk drives to prevent hot spots and maximize performance.
- ASM eliminates the need for over provisioning and maximizes storage resource utilization facilitating database consolidation.
- Inherent large file support.
- Performs an automatic online redistribution after the incremental addition or removal of the storage capacity.
- Maintains redundant copies of data to provide high availability or leverage 3rd party RAID functionality.
- Supports Oracle Database 19c as well as Oracle Real Application Clusters (RAC).
- Capable of leveraging 3rd party multipathing technologies.
- For simple and easier migration to ASM, an Oracle Database 19c database can contain ASM and non- ASM files. Any new files can be created as ASM files whilst existing files can also be migrated to ASM.
- RMAN commands enable non- ASM managed files to be relocated to an ASM disk group.
- Oracle Database 19c Enterprise Manager is used to manage ASM disk and file management activities.

7

Database Backup Recommendations

Backup Policy is a very important ingredient of any High Availability system. Oracle recommends RMAN utility for database backup.

RMAN is an acronym for Recovery Manager, which is an Oracle utility that will backup, restore, and recover Oracle data files. RMAN is an Oracle-provided utility for efficiently performing Backup and Recovery. RMAN is available as a part of the standard Installation and, no separate installation is required.

Recovery Manager is a client/server application that uses database server sessions to perform backup and recovery. It stores metadata about its operations in the control file of the target database and, optionally, in a recovery catalog schema in an Oracle database.

You can invoke RMAN as a command-line executable from the operating system prompt or use some RMAN features through the Enterprise Manager GUI.

RMAN Vs Conventional Backup

- During a conventional hot backup, the amount of Redo generated during the backup would be more due to the fact that the redo logs during the hot backup store the entire block images rather than the change vectors.
- RMAN doesn't place the tablespace in a backup mode and hence the amount of Redo generated during the RMAN backup is considerably low.
- RMAN can identify block corruption during backup operations and RMAN supports Block recovery.
- RMAN automatically detects new data files and will backup them. Also, RMAN supports the incremental backup method.
- RMAN backs up only the blocks that have been used at least once. Unused blocks are never backed up. Unused block here refers to the blocks where the block header is zeroed.
- RMAN enables us to test the backup without actually restoring the backup.
- RMAN can verify physical and logical structures of the database without actually performing backup.
- Usage of Shared Pool and Large Pool for RMAN.
- RMAN uses DBMS_RCVMAN and DBMS_BACKUP_RESTORE packages for backup and recovery. These packages would be loaded in the shared pool for backup and restore operation. RMAN uses the PGA for backup and restore operation.
- RMAN Requires LARGE_POOL only if TAPE_IO_SLAVES and DBWR_IO_SLAVES are defined.
- **Sizing Large Pool** - LARGE_POOL = (Number of Channels) * (16 MB + Tape Buffer)

Benefits of Using RMAN

- RMAN is an intelligent tool that comes at no extra cost. It is available free with the Oracle Database.

- RMAN introduced in Oracle 8 and has become simpler with newer versions and easier than user-managed backups.
- Provides proper security for backups.
- You can be 100% sure your database has been backed up.
- The control file and spfile of the database can be configured to be automatically backed up by RMAN.
- It contains the detail of the backups taken etc. in its central repository Facility for testing the validity of backups also commands like crosscheck to check the status of the backup.
- Faster backups and restores compared to backups without RMAN.
- RMAN is the only native backup tool which supports incremental backups.
- Oracle 19c has got further optimized incremental backup which has resulted in improvement of performance during backup and recovery time.
- Parallel operations (Multiple Channels for Backup and Restore) are supported.
- Better querying facility for knowing different details of backup.
- No extra redo is generated when the backup is performed, compared to conventional online backup.
- Maintains repository of backup metadata.
- Remembers backup set location.
- Knows what needs to back up.
- Knows what is required for recovery.
- Knows what backup is redundant.
- RMAN can back up the Database to Disk or directly to Tape. It is recommended that RMAN backup is performed to disk and then copied to tape.

Backup Strategy Recommendation

RMAN will not backup the below files so it is advised to take the copy of the below files on a regular basis (weekly/any change/addition to the file).

- Tnsnames.ora
- Listener.ora
- Password file
- Init.ora

The Best practice is to take create the pfile once the spfile is updated.

Below is the recommended strategy.

8
Appendix

This topic explains the script used to check or remove the histograms on the schema.

Script to Check Histograms on Schema

The following script would have to be executed in the schema:

Syntax

```
select distinct table_name
from
(select table_name from user_tab_columns where histogram!='NONE')


Should return No Records
```

Script to Remove Histograms on Schema

The following script would have to be executed in the schema if there are any rows:

```
declare
cursor cur_tables is
select distinct table_name from
(select table_name from user_tab_columns where histogram!='NONE' );
begin for rec_tables in cur_tables loop
dbms_stats.gather_table_stats(ownname=>USER,tabname=>rec_ta
bles.table_name,METHOD_OPT=>'FOR
ALL COLUMNS SIZE 1',CASCADE=>TRUE,DEGREE=>2,ESTIMATE_PERCENT=>NULL);
end loop;
end;
```