36 DBMS_AUTO_INDEX

The DBMS_AUTO_INDEX package provides the interface for managing auto indexes in an Oracle database.

This chapter contains the following topics:

- DBMS_AUTO_INDEX Overview
- Summary of DBMS_AUTO_INDEX Subprograms

DBMS_AUTO_INDEX Overview

The DBMS_AUTO_INDEX package is the interface for configuring auto indexes and generating reports of auto indexing operations in an Oracle database.

Summary of DBMS_AUTO_INDEX Subprograms

This table lists the DBMS_AUTO_INDEX package subprograms and briefly describes them.

Table 36-1 DBMS_AUTO_INDEX Package Subprograms

Procedure	Description
CONFIGURE Procedure	Configures settings related to automatic indexing.
DROP_AUTO_INDEXES Procedure	This procedure can be used to manually drop the automatically created indexes that overrides the retention parameter setting.
DROP_SECONDARY_INDEXES Procedure	Deletes all the indexes, except the ones used for constraints, from a schema or a table.
REPORT_ACTIVITY Function	Returns a report of the automatic indexing operations executed during a specific period in a database.
REPORT_LAST_ACTIVITY Function	Returns a report of the last automatic indexing operation executed in a database.
RECOMMEND Function	Run auto index on demand. This function analyzes all statements within a workload, creates invisible auto indexes and evaluates the performance of the statements in the workload with and without the candidate indexes.

CONFIGURE Procedure

This procedure configures settings related to automatic indexing.

Syntax

```
DBMS_AUTO_INDEX.CONFIGURE (
   parameter_name         IN VARCHAR2,
parameter_value        IN VARCHAR2,
allow                  IN BOOLEAN  DEFAULT TRUE);
```

Parameters

Table 36-2   CONFIGURE Procedure Parameters

Parameter	Description
`parameter_name`	Automatic indexing configuration setting. It can have one of the following values: • `AUTO_INDEX_MODE`: Modes of operation of auto indexes. It can have one of the following values: – `IMPLEMENT`: In this mode, new auto indexes are created as *visible* indexes and any existing *invisible* auto indexes are also set to *visible* indexes. In this mode, auto indexes are available to be used in SQL statements. – `REPORT ONLY`: In this mode, new auto indexes are created as *invisible* indexes and are not available to be used in SQL statements. – `OFF`: Setting the mode to `OFF` prevents new auto indexes from being considered and created. However, it does not disable existing auto indexes. • `AUTO_INDEX_SCHEMA`: Schemas to include or exclude from using auto indexes. Its value is case-sensitive and can include wildcards. Its behavior is controlled by the `allow` parameter. The automatic indexing process manages two schema lists – the *inclusion list* and the *exclusion list*. The inclusion list contains the schemas that can use auto indexes. The exclusion list contains the schemas that cannot use auto indexes. Initially, both these lists are empty and all the schemas in the database can use auto indexes when automatic indexing is enabled for a database. If the inclusion list contains at least one schema, then only the schemas listed in the inclusion list can use auto indexes. If the inclusion list is empty and the exclusion list contains at least one schema, then all the schemas can use auto indexes, except the schemas listed in the exclusion list. If both the lists (the inclusion list and the exclusion list) contain at least one schema, then all the schemas can use auto indexes, except the schemas listed in the exclusion list. • `AUTO_INDEX_RETENTION_FOR_AUTO`: Number of days for which the unused auto indexes are retained in the database, after which they are deleted. Default value is 373 days. • `AUTO_INDEX_RETENTION_FOR_MANUAL`: Number of days for which the unused manually created indexes (non-auto indexes) are retained in the database, after which they are

Table 36-2 (Cont.) CONFIGURE Procedure Parameters

Parameter	Description
	deleted. When it is set to `NULL`, the manually created indexes are not deleted by the automatic indexing process. Default value is `NULL`. • `AUTO_INDEX_REPORT_RETENTION`: Number of days for which automatic indexing logs are retained in the database before they are deleted. As automatic indexing report is generated based on these logs, automatic indexing report cannot be generated for a period beyond the value specified for `AUTO_INDEX_REPORT_RETENTION`. Default value is 373 days. • `AUTO_INDEX_DEFAULT_TABLESPACE`: Tablespace to use to store auto indexes. Default is `NULL`, which means the default permanent tablespace specified during the database creation is used to store auto indexes. Note that you cannot specify an Oracle-owned tablespace (such as `SYSAUX`) as the default tablespace. • `AUTO_INDEX_SPACE_BUDGET`: Percentage of tablespace size to use for auto indexes. This configuration setting can be used only when the default tablespace specified during the database creation is used for storing auto indexes. • `AUTO_INDEX_COMPRESSION`: Values to enable and disable advanced index compression for auto indexes. The supported values are: – `ON`: to enable advanced index compression for auto indexes – `OFF`: to disable advanced index compression for auto indexes The default value is `OFF`. • `AUTO_INDEX_TABLE`: You can use the `AUTO_INDEX_TABLE` configuration setting to specify tables that can use auto indexes. When you enable automatic indexing for a schema, all the tables in that schema can use auto indexes. However, if there is a conflict between the schema level and table level setting, the table level setting takes precedence. The parameter value is `<schema_name>.<table_name>`. You can then specify `TRUE` or `FALSE` to enable or disable auto indexes on the table respectively. To remove all tables from inclusion/exclusion list run the statement: `EXEC DBMS_AUTO_INDEX.CONFIGURE('AUTO_INDEX_TABLE', NULL);` • `AUTO_INDEX_INCLUDE_DML_COST`, default is `ON`. Because indexes must be maintained

Table 36-2 (Cont.) CONFIGURE Procedure Parameters

Parameter	Description
	when data is changed, this increase the overhead of DML operations, such as insert, update, and delete. When this option is enabled, automatic indexes factors in this overhead in deciding whether a new index is beneficial or not, particularly for tables with significant write activity.
`parameter_value`	Value for the configuration setting specified in `parameter_name`. When it is set to `NULL`, the configuration setting is assigned the default value.
`allow`	This parameter is applicable only for the `AUTO_INDEX_SCHEMA` configuration setting and it can have one of the following values: • `TRUE`: Add the specified schema to the inclusion list. • `FALSE`: Add the specified schema to the exclusion list. • `NULL`: Remove the specified schema from the list to which it is currently added. Refer to the description of the `AUTO_INDEX_SCHEMA` configuration setting for more information about the inclusion list and the exclusion list.

Examples

These examples are based on the assumption that the inclusion list and the exclusion list are initially empty.

The following example adds the `SH` and `HR` schemas to the exclusion list, so that only the `SH` and `HR` schemas cannot use auto indexes.

```
begin
    dbms_auto_index.configure(
         parameter_name  => 'AUTO_INDEX_SCHEMA',
         parameter_value => 'SH',
         allow           => FALSE);

    dbms_auto_index.configure(
         parameter_name  => 'AUTO_INDEX_SCHEMA',
         parameter_value => 'HR',
         allow           => FALSE);
end;
```

The following example removes the `HR` schema from the exclusion list, so that it can also use auto indexes. Now, only the `SH` schema cannot use auto indexes, because it is the only schema added to the exclusion list.

```
begin
    dbms_auto_index.configure(
         parameter_name  => 'AUTO_INDEX_SCHEMA',
         parameter_value => 'HR',
         allow           => NULL);
end;
```

The following example removes all the schemas from the exclusion list, so that all the schemas can use auto indexes.

```
begin
    dbms_auto_index.configure(
         parameter_name  => 'AUTO_INDEX_SCHEMA',
         parameter_value => NULL,
         allow           => TRUE);
end;
```

The following example adds the HR schema to the inclusion list, so that only the HR schema can use auto indexes.

```
begin
    dbms_auto_index.configure(
         parameter_name  => 'AUTO_INDEX_SCHEMA',
         parameter_value => 'HR',
         allow           => TRUE);
end;
```

The following example sets the retention period for auto indexes to 90 days.

```
begin
    dbms_auto_index.configure(
         parameter_name  => 'AUTO_INDEX_RETENTION_FOR_AUTO',
         parameter_value => '90');
end;
```

The following example sets the retention period for auto indexes to the default value of 373 days.

```
begin
    dbms_auto_index.configure(
         parameter_name  => 'AUTO_INDEX_RETENTION_FOR_AUTO',
         parameter_value => NULL);
end;
```

The following example enables a table:

```
EXEC DBMS_AUTO_INDEX.CONFIGURE('AUTO_INDEX_TABLE', 'SH.SALES', TRUE);
```

To remove all tables from inclusion/exclusion list:

```
EXEC DBMS_AUTO_INDEX.CONFIGURE('AUTO_INDEX_TABLE', NULL);
```

DROP_AUTO_INDEXES Procedure

This procedure can be used to manually drop the automatically created indexes that overrides the retention parameter setting.

Syntax

```
DBMS_AUTO_INDEX.DROP_AUTO_INDEXES (
   owner           IN  VARCHAR2 DEFAULT NULL,
   index_name      IN  VARCHAR2 DEFAULT NULL,
   allow_recreate  IN  BOOLEAN DEFAULT FALSE);
```

Parameters

Table 36-3 DROP_AUTO_INDEXES Procedure Parameters

Parameter	Description
owner	The name of the index owner.
index_name	The name of the index.
allow_recreate	Set this parameter to allow or disallow automatic creation of the dropped index again.

Examples

Drop a single index and allow recreate:

```
exec dbms_auto_index.drop_auto_indexes('SH','"SYS_AI_612ud3j5ngf0c"',TRUE);
```

Drop all indexes owned by SH and allow recreate:

```
exec dbms_auto_index.drop_auto_indexes('SH',NULL,TRUE);
```

Drop all indexes owned by HR, disallowing recreate and then change the recreation status back to allow:

```
exec dbms_auto_index.drop_auto_indexes('HR',NULL);
exec dbms_auto_index.drop_auto_indexes('HR', NULL, TRUE);
```

Usage Notes

- If the values of the parameters owner and index_name are explicitly set to NULL, all auto indexes which the user has privileges on will be dropped, except the ones used for constraints.
- If owner is explicitly specified and index_name is set to NULL, all auto indexes within the given schema will be dropped, except the ones used for constraints. The dropped indexes are not recreated automatically by the system by default. Set allow_recreate parameter to TRUE to change this behavior.
- This procedure updates the allow_recreate status associated with the dropped indexes from FALSE to TRUE and vice-versa.

DROP_SECONDARY_INDEXES Procedure

This procedure deletes all the indexes, except the ones used for constraints, from a schema or a table.

Syntax

```
DBMS_AUTO_INDEX.DROP_SECONDARY_INDEXES (
   ownname     IN  VARCHAR2 DEFAULT NULL,
   tabname     IN  VARCHAR2 DEFAULT NULL);
```

Parameters

Table 36-4 DROP_SECONDARY_INDEXES Procedure Parameters

Parameter	Description
`ownname`	(Optional) Name of the schema from which all the indexes need to be deleted. **Note:** The indexes used for constraints are not deleted.
`tabname`	(Optional) Name of the table from which all the indexes need to be deleted. **Note:** The indexes used for constraints are not deleted.

Examples

The following example deletes all the indexes, except the ones used for constraints, from the `SH` schema.

```
begin
    dbms_auto_index.drop_secondary_indexes('SH');
end;
```

The following example deletes all the indexes, except the ones used for constraints, from the `EMP` table in the `HR` schema.

```
begin
    dbms_auto_index.drop_secondary_indexes('HR', 'EMP');
end;
```

The following example deletes all the indexes, except the ones used for constraints, for which the user has the delete privileges from all the schemas in a database.

```
begin
    dbms_auto_index.drop_secondary_indexes;
end;
```

REPORT_ACTIVITY Function

This function returns a report of the automatic indexing operations executed during a specific period in a database.

Syntax

```
DBMS_AUTO_INDEX.REPORT_ACTIVITY (
   activity_start  IN  TIMESTAMP WITH TIME ZONE DEFAULT SYSTIMESTAMP - 1,
   activity_end    IN  TIMESTAMP WITH TIME ZONE DEFAULT SYSTIMESTAMP,
   type            IN  VARCHAR2 DEFAULT 'TEXT',
   section         IN  VARCHAR2 DEFAULT 'ALL',
   level           IN  VARCHAR2 DEFAULT 'TYPICAL')
RETURN CLOB;
```

Parameters

Table 36-5 REPORT_ACTIVITY Function Parameters

Parameter	Description
activity_start	Time starting from which the executed automatic indexing operations are considered for the report. If NULL is specified, the last executed automatic indexing operation is considered for the report. If no value is specified for this parameter, then the current time minus one day (24 hours) is considered at the start time.
activity_end	Time till which the executed automatic indexing operations are considered for the report. If no value is specified, then the current time is considered as the end time.
type	Format of the report. It can have one of the following values: • TEXT • HTML • XML The default value is TEXT.
section	Sections to include in the report. It can have a combination of the following values: • SUMMARY: Include only the summary details section in the report. • INDEX_DETAILS: Include only the auto index details section in the report. • VERIFICATION_DETAILS: Include only the auto index verification details section in the report. • ERRORS: Include only the error details section in the report. • ALL: Include all the sections (summary details, auto index details, auto index verification details, and error details) in the report. This is the default value. A combination of these values can be specified using the + or - operators as shown in the following examples: • SUMMARY +INDEX_DETAILS +ERRORS: Include summary details, auto index details, and error details sections in the report. • ALL -ERRORS: Include all the sections in the report, except the error details section.

Table 36-5 (Cont.) REPORT_ACTIVITY Function Parameters

Parameter	Description
`level`	Level of automatic indexing information to include in the report. It can have one of the following values: • `BASIC`: Include basic automatic indexing information in the report. • `TYPICAL`: Include typical automatic indexing information in the report. This is the default value. • `ALL`: Include all the automatic indexing information in the report.

Return Value

A report of the automatic indexing operations executed during the specified period in a database.

Examples

The following example generates a typical report of the automatic indexing operations executed in the last 24 hours. The report is generated in the text format and contains all the sections (summary details, auto index details, auto index verification details, and error details).

```
declare
  report clob := null;
begin
  report := dbms_auto_index.report_activity();
end;
```

REPORT_LAST_ACTIVITY Function

This function returns a report of the last automatic indexing operation executed in a database.

Syntax

```
DBMS_AUTO_INDEX.REPORT_LAST_ACTIVITY (
   type            IN  VARCHAR2 DEFAULT 'TEXT',
   section         IN  VARCHAR2 DEFAULT 'ALL',
   level           IN  VARCHAR2 DEFAULT 'TYPICAL')
RETURN CLOB;
```

Parameters

Table 36-6 REPORT_LAST_ACTIVITY Function Parameters

Parameter	Description
`type`	Format of the report. It can have one of the following values: • `TEXT` • `HTML` • `XML` The default value is `TEXT`.
`section`	Sections to include in the report. It can have a combination of the following values: • `SUMMARY`: Include only the summary details section in the report. • `INDEX_DETAILS`: Include only the auto index details section in the report. • `VERIFICATION_DETAILS`: Include only the auto index verification details section in the report. • `ERRORS`: Include only the error details section in the report. • `ALL`: Include all the sections (summary details, auto index details, auto index verification details, and error details) in the report. This is the default value. A combination of these values can be specified using the `+` or `-` operators as shown in the following examples: • `SUMMARY +INDEX_DETAILS +ERRORS`: Include summary details, auto index details, and error details sections in the report. • `ALL -ERRORS`: Include all the sections in the report, except the error details section.
`level`	Level of automatic indexing information to include in the report. It can have one of the following values: • `BASIC`: Include basic automatic indexing information in the report. • `TYPICAL`: Include typical automatic indexing information in the report. This is the default value. • `ALL`: Include all the automatic indexing information in the report.

Return Value

A report of the last automatic indexing operation executed in a database.

Examples

The following example generates a typical report of the last automatic indexing operation executed in a database. The report is generated in the text format and contains all the sections (summary details, auto index details, auto index verification details, and error details).

```
declare
  report clob := null;
begin
  report := dbms_auto_index.report_last_activity();
end;
```

RECOMMEND Function

Run auto indexing on demand.

This function analyzes all statements within a workload, creates invisible auto indexes and evaluates the performance of the statements in the workload both with and without the candidate indexes. If called in `IMPLEMENT` mode, this function marks as visible any indexes that improve performance. If called in `REPORT ONLY` mode, all indexes created in this task are dropped at the end of the task. In that case you can run `DBMS_AUTO_INDEX.REPORT_ACTIVITY()` to view the results.

Syntax

```
DBMS_AUTO_INDEX.RECOMMEND (
   WORKLOAD_START_TIME IN TIMESTAMP DEFAULT NULL,
   WORKLOAD_END_TIME   IN TIMESTAMP DEFAULT NULL,
   AUTO_INDEX_MODE     IN VARCHAR2 )
RETURN VARCHAR2;
```

Parameters

Table 36-7 RECOMMEND Function Parameters

Parameter	Description
`WORKLOAD_START_TIME`	The starting point for analysis within the ASTS (Automatic SQL tuning set) workload. The default NULL means all statements from the start of the workload to `WORKLOAD_END_TIME` are analyzed.
`WORKLOAD_END_TIME`	The end point for analysis within the ASTS workload. The default NULL means that all statements from `WORKLOAD_START_TIME` to the end of the workload are analyzed.
`AUTO_INDEX_MODE`	• `REPORT ONLY` Analysis is performed, but there is no change. Created indexes are dropped. See `DBMS_AUTO_INDEX.REPORT_ACTIVITY()` to view verification details. • `IMPLEMENT` Marks performance-improving indexes as visible. The default is `REPORT ONLY`.

Return Value

The execution name of the task is returned.

Example 36-1 Using the RECOMMEND Function

Execute the function and then call `DBMS_AUTO_INDEX.REPORT_LAST_ACTIVITY` to return the report.

```
var tname varchar2(100)
EXEC :tname := dbms_auto_index.recommend()
SELECT :tname task_name FROM dual;
```

Get the report:

```
set linesize 250
set trims on
set pagesize 1000
set long 10000000
column report format a120
spool report.txt
SELECT dbms_auto_index.report_last_activity('text','all','all') report FROM 
dual;
spool off
```