

Advisors für die Oracle Datenbank 12c

Ulrike Schwinn
Oracle Deutschland B.V. & Co.KG
München

Schlüsselwörter: Automated Maintenance Task, Optimizer Statistics Advisor, 12c, 12.2, Multitenant, fixed objects, cdb_advisor_tasks, Index Monitoring

Einleitung

Was ist eigentlich ein Advisor? Der englische Begriff „advisor“ lässt sich mit dem Ausdruck Berater oder Ratgeber übersetzen. Der Terminus Advisor findet mittlerweile in vielen Bereichen eine Anwendung; so gibt es Personal Advisor, Investment Advisor, Trusted Advisor etc - so auch in der Datenbank oder in den Datenbank-Werkzeugen.

Was zeichnet nun einen Advisor in der Oracle Datenbank aus? Müssen Advisors generell lizenziert werden? Muss man Advisors aktivieren oder sind diese automatisch eingeschaltet? Wie kann das Advisory Framework bei der täglichen Arbeit unterstützen? Was ist neu in Oracle Database 12c? Wie funktionieren die Advisors in einer Multitenant Umgebung? Dieser Beitrag soll dabei helfen, diese und weitere Fragen zu beantworten. Da das Thema sehr weitreichend ist, wird zu Beginn ein Überblick über einige Advisors gegeben, um Gemeinsamkeiten und Unterschiede herauszustellen. Weitere Links und Hinweise auf Veröffentlichungen zum Thema runden den Artikel ab und vervollständigen das Gesamtbild.

Advisors in der Oracle Datenbank

Advisors analysieren die Datenbank in den unterschiedlichsten Bereichen - wie zum Beispiel im Bereich Memory, Speicherverwaltung, Undo-Management oder im Bereich Performance. Advisors müssen in der Regel nicht zusätzlich installiert werden und sind teilweise schon aktiviert. Ausnahmen gibt es aber auch hier - wie zum Beispiel der Advisor für die Database In-Memory Option, das Oracle Database Security Assessment Tool (kurz DBSAT) oder das Health Check Utility ORACHK. Meist sind diese aber leicht zu installieren bzw. nach einem einfachen Download nutzbar.

Standardmässig werden nur Empfehlungen gegeben; dem Datenbankuser bleibt es überlassen diese selbst explizit anzuwenden. Dabei können Skripte generiert werden, die eine gute Hilfestellung bei der Implementierung liefern können. Sie sollten allerdings vor der Ausführung immer genau geprüft und gegebenenfalls angepasst werden. Wichtig zu wissen ist, wo und wie die Advisors und ihre Ratschläge zu finden sind. Im Unterschied zu Alerts sind Advisors Ressource intensiver, da ihre Analyse und Lösungsvorschläge einen größeren Detaillierungsgrad aufweisen.

Advisors sind in der Regel über unterschiedliche Methoden zu nutzen - graphisch über den Enterprise Manager oder SQL Developer, skriptgesteuert über PL/SQL Packages, über Initialisierungsparameter und V\$-Views. Folgende Liste – in alphabetischer Reihenfolge - zeigt eine Auswahl von gängigen Advisors (Stand August 2017):

- **ADDM (Automatic Database Diagnostic Advisor)** diagnostiziert Datenbank Performance Probleme. Normalerweise werden diese Empfehlungen automatisch nach jedem AWR Snapshot generiert.
- Der **Compression Advisor** berechnet die Komprimierungsrate pro Tabelle oder Partition für unterschiedliche Komprimierungsalgorithmen - sogar für EHCC (Exadata Hybrid Columnar Compression). Darüber hinaus kann er den eingestellten Komprimierungstyp diagnostizieren.
- Der **Data Recovery Advisor** analysiert persistente "Failure" (z.B. Daten-Korruptionen) und gibt Ratschläge, wie diese Probleme zu beheben sind.
- Der **Database In-Memory Advisor** analysiert einen Workload und gibt Ratschläge wie die Datenbank für In-Memory konfiguriert werden soll.
- Der **Memory Advisor** analysiert die unterschiedlichen Cache Größen in einer WHAT-IF Analyse. Zu beachten ist, dass die Oracle Datenbank das Memory je nach Wahl der Memory Management Methode selbst einstellt und somit kein manuelles Eingreifen nötig ist.
- Der **MTTR Advisor** analysiert das Checkpoint Verhalten mit dem aktuellen Wert von FAST_START_MTTR_TARGET und vier weiteren abgeleiteten Werten.
- Der **Segment Advisor** findet Segmente, die zu viel Platz verbrauchen. Die Empfehlungen beinhalten die entsprechenden Kommandos zur Freigabe von Speicherplatz.
- Der **SQL Access Advisor** gibt Informationen und Empfehlungen über die Zugriffsstrukturen (fehlende oder überflüssige) - wie Bitmap Indizes, Function Based Indizes, B*tree Indizes, materialisierte Views usw. Zusätzlich empfiehlt er eventuell Partitionierung für nicht partitionierte Tabellen.
- Der **SQL Performance Analyzer (SPA)** Der **SQL Performance Analyzer (SPA)** führt eine detaillierte Statementanalyse eines definierten SQL Workload durch und gibt einen Überblick über die Performance Unterschiede und die Pläne der einzelnen Statements.
- Der **SQL Repair Advisor** analysiert Statements mit kritischen Fehlern und versucht, einen SQL Patch vorzuschlagen, der dem Optimizer einen alternativen Plan vorschlägt.
- Der **SQL Tuning Advisor** führt einen Optimizer-Analyse-Lauf durch und gibt Empfehlungen in folgenden Kategorien: Statistiken, SQL Profiles, Zugriffsstrukturen, Statementstruktur.
- Der **(Optimizer) Statistics Advisor** analysiert wie Statistiken erzeugt wurden, validiert die Qualität von Statistiken und überprüft den Status der Maintenance Tasks (hier: auto stats gathering).
- Der **Undo Advisor** hilft dabei, eine angemessene Größe des UNDO Tablespaces für langlaufende Abfragen und Flashback Queries festzulegen.

Beachten sollte man, dass bei der Verwendung einiger Advisor eine Lizenzierung erforderlich ist. Details dazu finden sich wie immer im [Database Licensing Information User Manual](#).

Voraussetzungen und Monitoring

Eine wichtige Voraussetzung zum Funktionieren einiger Advisors ist das Setzen des Parameters STATISTICS_LEVEL. Der empfohlene Defaultwert ist dabei TYPICAL. Diese Einstellung ist notwendig, um das Sammeln von Advisor Statistiken zu aktivieren und hat keinen Einfluss auf die Lizenzierung von Packs. Die Einstellung BASIC schaltet einige Advisors aus und kann auch nur bei manueller SGA Verwaltung verwendet werden. Folgendes Listing gibt Auskunft über die aktivierten Statistiken.

```
SQL> SELECT statistics_name, activation_level, session_settable,
           session_status, statistics_view_name
FROM v$statistics_level ORDER BY 1;
```

```
STATISTICS_NAME          ACTIVAT SES SESSION_ STATISTICS_VIEW_NAME
```

```

-----
Active Session History          TYPICAL NO  ENABLED  V$ACTIVE_SESSION_HISTORY
Adaptive Thresholds Enabled    TYPICAL NO  ENABLED
Automated Maintenance Tasks    TYPICAL NO  ENABLED
Automatic DBOP Monitoring      TYPICAL YES ENABLED  V$SQL_MONITOR
Bind Data Capture              TYPICAL NO  ENABLED  V$SQL_BIND_CAPTURE
Buffer Cache Advice            TYPICAL NO  ENABLED  V$DB_CACHE_ADVICE
Column Tracking Level          TYPICAL YES ENABLED
Global Cache CPU Statistics     ALL        NO  DISABLED
Global Cache Statistics        TYPICAL NO  ENABLED
Longops Statistics             TYPICAL NO  ENABLED  V$SESSION_LONGOPS
MTRR Advice                    TYPICAL NO  ENABLED  V$MTRR_TARGET_ADVICE
Modification Monitoring        TYPICAL NO  ENABLED
OLAP row load time precision   TYPICAL YES ENABLED
PGA Advice                     TYPICAL NO  ENABLED  V$PGA_TARGET_ADVICE
Plan Execution Sampling        TYPICAL YES ENABLED  V$ACTIVE_SESSION_HISTORY
Plan Execution Statistics       ALL        YES  DISABLED V$SQL_PLAN_STATISTICS
SQL Monitoring                 TYPICAL YES ENABLED  V$SQL_MONITORING
Segment Level Statistics       TYPICAL NO  ENABLED  V$SEGSTAT
Shared Pool Advice            TYPICAL NO  ENABLED  V$SHARED_POOL_ADVICE
Streams Pool Advice           TYPICAL NO  ENABLED  V$STREAMS_POOL_ADVICE
Threshold-based Alerts        TYPICAL NO  ENABLED
Time Model Events             TYPICAL YES ENABLED  V$SESS_TIME_MODEL
Timed OS Statistics           ALL        YES  DISABLED
Timed Statistics              TYPICAL YES ENABLED
Ultrafast Latch Statistics     TYPICAL NO  ENABLED
Undo Advisor, Alerts and Fast  TYPICAL NO  ENABLED  V$UNDOSTAT
Ramp up
V$IOSTAT_* statistics         TYPICAL NO  ENABLED

```

27 rows selected.

Je nach Lizenzierungsvoraussetzung der Advisor muss der Initialisierungsparameter **CONTROL_MANAGEMENT_PACK_ACCESS** die richtige Einstellung haben. So ist für die Verwendung von **SQL Tuning**, **SQL Access** und **Database In-Memory Advisor** die Einstellung **DIAGNOSTIC+TUNING** erforderlich. Für das Funktionieren von **ADDM** ist der Wert **DIAGNOSTIC** hingegen ausreichend.

Um zu sehen, welche Advisor Tasks in der Datenbank ausgeführt werden eignet sich das Abfragen der View **DBA_ADVISOR_TASKS** bzw. **CDB_ADVISOR_TASKS** in einer Multitenant Umgebung. Das folgende Scripting zeigt das Resultat bei der Abfrage mit einem Common User aus **CDB\$ROOT**.

```
SQL> show pdbs
```

```

-----
CON_ID  CON_NAME                                OPEN MODE  RESTRICTED
-----
2  PDB$SEED                                READ ONLY  NO
3  PDB1                                     READ WRITE NO
4  DEMOS                                    MIGRATE    YES

```

```
SQL> select advisor_name, task_name, execution_start, con_id
       from cdb_advisor_tasks where execution_start > sysdate-2
       order by 2 desc;
```

```

ADVISOR_NAME          TASK_NAME                                EXECUTION  CON_ID
-----
SQL Tuning Advisor    SYS_AUTO_SQL_TUNING_TASK                02-AUG-17  1
SPM Evolve Advisor    SYS_AUTO_SPM_EVOLVE_TASK                02-AUG-17  3
Statistics Advisor    AUTO_STATS_ADVISOR_TASK                 02-AUG-17  3
ADDM                  ADDM:1468223924_1_2881                  03-AUG-17  1

```

```

ADDM          ADDM:1468223924_1_2880          03-AUG-17          1
ADDM          ADDM:1468223924_1_2879          03-AUG-17          1
ADDM          ADDM:1468223924_1_2878          03-AUG-17          1
...

```

Die ADDM Tasks und der SQL Tuning Advisor werden offensichtlich nur in der CDB\$ROOT (CON_ID 1) ausgeführt wohingegen der Statistics Advisor und SPM Evolve Advisor in der PDB1 (CON_ID 3) ausgeführt werden.

Eine gute Übersicht über die Tasks – ihre Speicherung und Sichtbarkeit in einer Multitenant Umgebung – erhält man über die Tabelle 40-2 (Manageability Features in a CDB) im Database Administrator's Guide.

Automatische Maintenance Tasks in Oracle Database 12c

Einige Advisors werden bzw. können automatisch in den sogenannten automatischen Maintenance Tasks ausgeführt werden. Die **Automatische Maintenance Tasks** sind dabei Operationen, die in regelmässigen Intervallen in sogenannten Scheduler Windows automatisch in der Datenbank ausgeführt werden. In Oracle Database 12c Release 2 sind dies folgende Operationen.

```
SQL> SELECT client_name, status, attributes FROM dba_autotask_client;
```

```

CLIENT_NAME                                     STATUS
-----
ATTRIBUTES
-----
auto optimizer stats collection                 ENABLED
ON BY DEFAULT, VOLATILE, SAFE TO KILL

auto space advisor                             ENABLED
ON BY DEFAULT, VOLATILE, SAFE TO KILL

sql tuning advisor                             ENABLED
ONCE PER WINDOW, ON BY DEFAULT, VOLATILE, SAFE TO KILL

```

Was steckt nun hinter den Client Namen und was ist neu in Oracle Database 12c?

Der Client Name „**auto space advisor**“ steht für den automatischen Segment Advisor, der Segmente findet, die zu viel Platz verbrauchen. Die Advisor Empfehlungen beinhalten die entsprechenden Kommandos zur Freigabe von Speicherplatz. Die automatische Analyse umfasst im Unterschied zur manuellen Analyse mit dem Segment Advisor nicht alle Segmente, sondern nur diejenigen, die eine hohe Wachstumsrate und eine hohe Aktivität aufweisen sowie in einem Tablespace liegen, der das Critical oder Warning Threshold überschreitet. Dabei werden sogar OLTP Compression Empfehlungen für Tabellen mit gewissen Eigenschaften gegeben.

Mit „**auto optimizer stats collection**“ werden automatisch Statistiken für Objekte gesammelt, die entweder keine Statistiken oder keine aktuellen (auch stale) Statistiken besitzen. Im Unterschied zum manuellen Start mit DBMS_STATS.GATHER_... wird bei der automatischen Ausführung nach Prioritäten vorgegangen: Objekte ohne Statistiken oder mit sehr alten Statistiken werden zuerst behandelt. Dabei werden alle Tabellen in allen Schemas berücksichtigt – **ab 12c** sogar auch die sogenannten "**fixed objects**". Das sind die dynamischen Performance Tabellen und ihre Indizes. Wichtig zu wissen ist, dass dabei die Standardeinstellungen der DBMS_STATS Prozeduren verwendet werden. Möchte man diese verändern, kann man die Prozedur

DBMS_STATS.SET_GLOBAL_PREFS oder DBMS_STATS.SET_DATABASE_PREFS verwenden. Die Einstellungen überprüfen, kann man übrigens mit DBMS_STATS.GET_PREFS.

Neu mit 12.2 ist der **(Optimizer) Statistics Advisor**, der die Qualität der Statistiken analysiert und automatisch im Maintenance Window läuft. Das Ergebnis des Advisors lässt sich beispielsweise mit einem Abfrage in Linemode generieren. Folgende Abfrage generiert den Ergebnisbericht zur letzten Ausführung der **AUTO_STATS_ADVISOR_TASK**.

```
-- gegebenenfalls mit PDB verbinden
-- alter session set container=pdbl;

SET LONG 1000000 LONGCHUNKSIZE 100000 PAGESIZE 0

SELECT DBMS_STATS.REPORT_ADVISOR_TASK(
    task_name      => 'AUTO_STATS_ADVISOR_TASK'
    , execution_name => NULL
    , type          => 'TEXT' - oder HTML, XML
    , section       => 'ALL' ) AS report
FROM DUAL;

GENERAL INFORMATION
-----

Task Name      : AUTO_STATS_ADVISOR_TASK
Execution Name : EXEC_664
Created        : 09-28-16 04:58:02
Last Modified  : 08-02-17 10:03:14
-----

SUMMARY
-----

For execution EXEC_664 of task AUTO_STATS_ADVISOR_TASK, the Statistics Advisor
has 2 finding(s). The findings are related to the following rules:
AVOIDSTALESTATS, UNLOCKNONVOLATILETABLE. Please refer to the finding section
for detailed information.

-----

FINDINGS
-----

Rule Name:      AvoidStaleSt
Rule Description: Avoid objects with stale or no statistics
Finding: There are 1 object(s) with stale statistics.
Schema: ...
```

Das Ganze lässt sich übrigens auch als Skript mit **DBMS_STATS.SCRIPT_ADVISOR_TASK** generieren.

```
SQL> select DBMS_STATS.SCRIPT_ADVISOR_TASK('AUTO_STATS_ADVISOR_TASK') from dual;
-- Script generated for the recommendations from execution EXEC_664
-- in the statistics advisor task AUTO_STATS_ADVISOR_TASK
-- Script version 12.2
...
```

Mehr Informationen dazu findet sich auch in unserem Community Tipp auf <https://blogs.oracle.com/coretec>.

Der **SQL Tuning Advisor** führt eine Optimizer-Analyse-Lauf durch und kann Empfehlungen in folgenden Kategorien geben: Statistiken, SQL Profiles, Zugriffsstrukturen und Statementstruktur. Wie

die anderen beiden Operationen ist der automatische SQL Tuning Advisor im Maintenance Window eingeplant. Bei jedem automatischen Lauf werden SQL Abfragen im System ausgewählt und die entsprechenden Tuning Empfehlungen ausgegeben. Dabei kann der automatische Lauf so konfiguriert werden, dass automatisch die SQL Profile Empfehlungen implementiert werden. Zur Erinnerung: Ein SQL Profile enthält zusätzliche Informationen, um den Optimizer dabei zu unterstützen, einen besseren Ausführungsplan zu generieren.

Das manuelle Starten bzw. Änderungen an den Standardeinstellungen können dabei über das Package DBMS_AUTO_SQLTUNE erfolgen. So ist beispielsweise ein manuelles Anstarten durch die Prozedur DBMS_AUTO_SQLTUNE.EXECUTE_AUTO_TUNING_TASK möglich; über DBMS_AUTO_SQLTUNE.SET_AUTO_TUNING_TASK_PARAMETER können die einzelnen Parameter vorab verändert werden. Das Handbuch Oracle Database PL/SQL Packages and Types Reference gibt einen Überblick über alle möglichen Einstellungen.

Neu mit Oracle Database 12c ist der sogenannte **Evolve Advisor**. Nutzt man das SQL Plan Management (auch SPM), ist der Evolve Advisor automatisch zusammen mit dem SQL Tuning Advisor aktiviert. Wird ein neuer SQL Plan (Ausführungsplan) zur SQL Plan Baseline hinzugefügt, wird damit automatisch überprüft, ob dieser Ausführungsplan besser ist - in 11g musste diese Überprüfung manuell initiiert werden. Einstellen lässt sich dabei die Dauer der Überprüfung und das Akzeptieren des Plans. Die Default-Einstellung erlaubt eine Dauer von 3600 (TIME_LIMIT) Sekunden und akzeptiert den Plan automatisch (ACCEPT_PLANS=TRUE), falls er besser ist.

```
SQL> col PARAMETER_NAME format a25
SQL> col VALUE format a10
SQL> SELECT parameter_name, parameter_value AS "VALUE"
        FROM dba_advisor_parameters
        WHERE ((TASK_NAME = 'SYS_AUTO_SPM_EVOLVE_TASK') AND
              ((PARAMETER_NAME = 'ACCEPT_PLANS') OR
               (PARAMETER_NAME = 'TIME_LIMIT')));
```

PARAMETER_NAME	VALUE
TIME_LIMIT	3600
ACCEPT_PLANS	TRUE

```
-- Änderung der Dauer auf 1200
execute DBMS_SPM.SET_EVOLVE_TASK_PARAMETER(
        task_name => 'SYS_AUTO_SPM_EVOLVE_TASK',
        parameter => 'LOCAL_TIME_LIMIT',
        value      => 1200);
```

Zusätzlich besteht natürlich auch die Möglichkeit Advisors manuell anzustossen. Je nach Advisor Typ stehen dazu unterschiedliche PL/SQL Packages zur Verfügung. Folgende Übersicht gibt einen Überblick über die Verwendungsmöglichkeiten.

Advisor Name	Verwendung
ADDM	Automatisch nach AWR-Snapshot oder DBMS_ADDM
Compression Advisor	DBMS_COMPRESSION
Data Recovery Advisor	RMAN und ADRCI Kommandos
In-Memory Advisor	Download über Doc ID 1965343.1
Memory Advisor	Einstellung für SGA, PGA, Shared Pool etc. (mit DB_CACHE_ADVICE=ON)

MTTR Advisor (Fast Start Fault Revocery)	FAST_START_MTTR_TARGET= <i>wert</i>
Segment Advisor	Automatische Maintenance Task oder DBMS_ADVISOR
SQL Access Advisor	Automatische Maintenance Task oder DBMS_ADVISOR
SQL Performance Analyzer	DBMS_SQLPA
SQL Repair Advisor	DBMS_SQLDIAG
SQL Tuning Advisor	Automatische Maintenance Task oder DBMS_ADVISOR
(Optimizer) Statistics Advisor	Automatische Maintenance Task oder DBMS_STATS
Undo Advisor	DBMS_ADVISOR oder DBMS_UNDO_ADV

Eine Advisor Task wird normalerweise mit einem CREATE Kommando erzeugt und einem EXECUTE Kommando ausgeführt. Die Empfehlungen lassen sich in einer Advisor Data Dictionary View einsehen. SCRIPT Kommandos helfen dann bei der Implementierung der Empfehlungen.

Exemplarisch soll am Beispiel des neusten Advisors dieses Vorgehen demonstriert werden. So wird ein "On-Demand" Lauf des Statistics Advisors mit den Funktionen CREATE_ADVISOR_TASK und EXECUTE_ADVISOR_TASK des Packages DBMS_STATS durchgeführt.

```

DECLARE
v_tname VARCHAR2(128) := 'ADV_TASK_STAT';
v_execname VARCHAR2(128) := NULL;
BEGIN
v_tname := DBMS_STATS.CREATE_ADVISOR_TASK(v_tname);
v_execname := DBMS_STATS.EXECUTE_ADVISOR_TASK(v_tname);
END;
/

```

Das Ergebnis lässt sich dann mit der Funktion REPORT_ADVISOR_TASK ausgeben (siehe auch AUTO_STATS_ADVISOR_TASK).

```

set pagesize 0 long 1000000 LONGCHUNKSIZE 100000
spool manual_report.html
SELECT DBMS_STATS.REPORT_ADVISOR_TASK (
    task_name      => 'ADV_TASK_STAT'
    , execution_name => NULL
    , type          => 'HTML' -- TEXT oder XML
    , section       => 'ALL' ) AS report
FROM DUAL;
spool off

```

Das vollständige Beispiel und weitere Code Beispiele zu anderen Advisors finden sich in unserem Blog (Suchwort: Advisor bzw. Optimizer Statistics Advisor).

Weitere nützliche Werkzeuge

An dieser Stelle soll noch auf zwei weitere Werkzeuge hingewiesen werden, die nicht unbedingt zum Advisory Framework gezählt werden aber trotzdem nützlich sein können. Ein Beispiel stellt der **PL/SQL Hierarchical Profiler** dar – das ultimative Werkzeug um PL/SQL Code (speziell langlaufenden Code) zu analysieren.

Der hierarchical Profiler besteht aus dem Package DBMS_HPROF, mit dem der Start- und Endpunkt beim Sammeln von Profiler Daten definiert wird. Die Daten lassen sich dann entweder aus speziellen Profiler Tabellen auszulesen oder mit dem Linemode Werkzeug *plshprof* in HTML Format aufbereiten. Oracle Handbücher und Tutorials im Internet zeigen die einfache Verwendung. Dabei war es vor 12.2 immer schwierig herauszufinden ob und welche SQL Statements im PL/SQL Code verwendet werden: Nur die verwendete Zeit, die ein SQL Statement verbraucht hat, wured aufgelistet, aber keine Information zum SQL Code selbst mitgeführt. In **12.2** ist es nun einfach und intuitiv möglich SQL Statements in den PL/SQL Programmen aufzuspüren. Führt man die Analyse mit dem Linemode Werkzeug *plshprof* in 12.2 durch, stellt man fest, dass DBMS_HPROF nun auch die entsprechenden SQL ID und den SQL Text mitliefert. Damit ist es einfach möglich den zugehörigen SQL Code zu analysieren.

Immer wieder erhalten wir Anfragen zum Thema Überprüfung der Indexnutzung in der Oracle Datenbank. Meist geht es darum herauszufinden, ob Indizes gänzlich ungenutzt sind oder selten verwendet werden. Seit jeher gibt es verschiedene Methoden des Index Trackings. Jede Methode hat dabei ihre Vor- und Nachteile und ist mit unterschiedlichem Aufwand verbunden. Eine erste Idee um Indizes zu monitoren, bietet sicherlich die manuelle Methode - das Generieren bzw. Selektieren des **Ausführungsplans**. Einfacher geht es mit dem Advisory Framework - dem **SQL Access Advisor**. Im aktuellen Datenbankrelease (12.2) ist das **Indexmonitoring** nicht nur stark vereinfacht worden, sondern liefert darüberhinaus auch detaillierte Ergebnisse. Neue Data Dictionary Strukturen liefern dabei automatisch Informationen über den Gebrauch von Indizes in der Datenbank. Neue Data Dictionary Views geben dabei an, welche Indizes Verwendung finden und auch wie häufig ein Index genutzt wurde. Auf diese Weise lässt sich leicht herausfinden, ob die gewählte Indexstrategie die richtige ist oder ob es Indizes gibt, die wegfallen könnten. Kein Einschalten des Features ist erforderlich und es steht in allen Editionen zur Verfügung. Ne Views

Beispielsweise gibt V\$INDEX_USAGE_INFO allgemeine Informationen zum Indexmonitoring und zeigt an, wann der letzte "Flush" (Leerung) erfolgt ist. Dabei erfolgt ein Flush alle 15 Minuten. Der Wert "1" in der Spalte INDEX_STATS_ENABLED steht dabei für "Enabled" (eingeschaltet). Bei INDEX_STATS_COLLECTION_TYPE gibt der Wert "1" an, dass für die Statistiken ein Sampling bei den Ausführungen verwendet wird. Dies bedeutet, dass nicht bei jeder Ausführung eines Statements, die Indexnutzung aufgezeichnet wird.

```
SQL> select active_elem_count active, alloc_elem_count alloc, flush_count,
        last_flush_time last_flush, con_id from v$index_usage_info;
```

ACTIVE	ALLOC	FLUSH_COUNT	LAST_FLUSH	CON_ID
1	1	5	05.01.2017 12:37	3

Wann ein Index zum letzten Mal und wie häufig dieser verwendet wurde, kann man dann aus der Data Dictionary Tabelle DBA_INDEX_USAGE auslesen. Durch den Wert in der Spalte TOTAL_ROWS_RETURNED lässt sich sogar etwas über die Güte des Index aussagen.

```
SQL> select object_id, name, total_access_count, total_rows_returned, last_used
        from dba_index_usage where owner='HR';
```

OBJECT_ID	NAME	TOTAL_ACCESS_COUNT	TOTAL_ROWS_RETURNED	LAST_USED
88289	EMP_JOB_IX	454	48578	05.01.2017 10:42
88296	LOC_CITY_IX	143	3289	05.01.2017 13:57

Beachten sollte man, dass das Feature aus Performancegründen mit einem Sampling Algorithmus konfiguriert ist. Das heisst, nicht jede Ausführung wird automatisch berücksichtigt. Da die Verwendung sehr einfach ist und keinen zusätzlichen Overhead beinhaltet, sollte man dieses Index Tracking bei jeder Tuning Massnahme einplanen.

Fazit

Das Advisory Framework und die oben genannten Werkzeuge bieten eine vielfältige Auswahl an Anwendungsmöglichkeiten. Nutzt man die Maintenance Tasks sind schon einige Advisors automatisch eingeschaltet. Jeder Datenbankuser mit ausreichenden Privilegien – minimale Voraussetzung ist das ADVISOR Recht – kann das Advisor Framework sehr einfach verwenden. Teilweise stehen dazu auch graphische Werkzeuge zur Verfügung. In jedem Fall lassen sie sich im Linemode über Skripte ansprechen. Vermutet man beispielsweise, dass gewisse Segmente oder Queries vom Tuning Lauf oder der Komprimierung profitieren könnten, sollte man sich die Mühe machen, den entsprechenden Advisor manuell in der graphischen Oberfläche oder über PL/SQL Packages anzustoßen. Auch der Empfehlungen des neuen Statistics Advisor können hilfreich sein.

Übrigens die hier erwähnten Advisors und Werkzeuge stehen ohne Einschränkungen in Multitenant wie auch Non CDB Umgebungen zur Verfügung. Dabei macht es auch keinen Unterschied ob die Datenbank über die Cloud oder On-Premise genutzt wird. Die Beispiel Skripte im Text sind in allen Umgebungen ablaufbar.

Weitere Informationen

- Blog: Deutschsprachiger Datenbank und Cloud Technology Community (<http://blogs.oracle.com/coretec>)
- My Oracle Support: Optimizer Statistics Advisor in 12.2 (Doc ID 2259398.1)
- My Oracle Support: Oracle Database Security Assessment Tool (DBSAT) (Doc ID 2138254.1)
- My Oracle Support: Oracle Database In-Memory Advisor (Doc ID 1965343.1)
- Handbuch: Administrator Guide (Kapitel 19)
- Informationen auf OTN zu Diagnostic und Tuning http://www.oracle.com/technetwork/database/focus-areas/manageability/info/performance_diagnostics/index.html
- SQL *Developer Online Demonstration : SQL Tuning Advisor http://download.oracle.com/otn_hosted_doc/sqldev/tuningadvisor/tuningadvisor_viewlet_swf.html

Kontaktadresse

Ulrike Schwinn
BU Database & Cloud Technologies
ORACLE Deutschland B.V. & Co. KG
Riesstr 25, 80992 München
Telefon: +49 89 1430 1865

E-Mail Ulrike.Schwinn@oracle.com
Blog: <https://blogs.oracle.com/coretec>
Twitter: @uschwinn