

# Oracle Database 12c: kleine kostenlose und hilfreiche Features

Markus Flechtner, Trivadis GmbH



Wenn man über die neuen Features der Oracle Datenbank 12c spricht, stehen meist die großen Features wie Container-Datenbank-Architektur oder In-Memory-Option im Vordergrund. Oracle 12c bietet aber viele kleine hilfreiche und kostenlose Features, die keine zusätzliche Datenbank-Option erfordern. Allein diese Features, die den Datenbank-Alltag oftmals sehr erleichtern, sind Argumente für ein Upgrade auf die aktuelle Version.

Dieser Artikel zeigt eine subjektive Auswahl derjenigen neuen Features, die schon in der Standard Edition der Datenbank zur Verfügung stehen. Außerdem ist dafür keine besondere Infrastruktur wie ein Cluster mit der Oracle Grid Infrastructure oder ein Engineered System notwendig. Diese Kriterien reduzieren allerdings die Auswahl recht deutlich.

## Mehrere Indizes auf die gleiche Spaltenliste

In 12c ist es möglich, mehrere Indizes auf die gleiche Spaltenliste einer Tabelle anzulegen. Dies kann hilfreich sein, wenn man beispielsweise die Charakteristik eines Index von „Non Unique“ nach „Unique“ ändern und die Dauer dafür reduzieren möchte. Auch wenn man testen möchte, wie sich die Applikation mit dem anderen Index verhält, ist dieses Feature hilfreich. Dabei darf immer nur einer dieser Indizes sichtbar („VISIBLE“) sein (siehe Listing 1).

Da jeder Index bei einem DML-Befehl aktualisiert werden muss, verteuert eine derartige Konstruktion mit unsichtbaren Indizes die DML-Operationen. Daher sollte der Parallelbetrieb mehrerer dieser Indizes nur eine Übergangslösung sein. Der Wechsel von einem Index geschieht schnell und einfach, indem die Sichtbarkeit der Indizes geändert wird (siehe Listing 2).

Anschließend steht der neue Index zur Verfügung. Auf dem gleichen Weg kann auch zurück zum alten Index gewechselt

werden. Bei 11g war für derartige Änderungen noch die Befehlsfolge „DROP INDEX“ und „CREATE INDEX“ erforderlich und dadurch ein Wechsel zwischen Index-Varianten entsprechend aufwändig.

## Obere Grenze für die Process Global Areas

Bis 11g gibt es für die Begrenzung der PGAs nur den Parameter „PGA\_AGGREGATE\_TARGET“. Dieser ist aber nur ein Zielwert, der überschritten werden kann. Derartige Überschreitungen können dann auf dem Datenbank-Server zu Swapping führen. Mit 12c wurde der Parameter „PGA\_AGGREGATE\_LIMIT“ eingeführt. Damit ist eine feste Obergrenze für die kumulierten PGA-Größen der Datenbank-Sitzungen definiert. Wird diese erreicht, erhält die Sitzung, mit der das Limit überschritten würde, die Fehlermeldung „ORA-04036: PGA memory used by the instance exceeds PGA\_AGGREGATE\_LIMIT“. Wenn der Parameter nicht im „spfile“ beziehungsweise in der „init.ora“-

Datei gesetzt ist, nimmt Oracle den kleinsten der folgenden Werte:

- 2 GB
- 2 \* „PGA\_AGGREGATE\_TARGET“
- 3 MB \* Wert des Parameter-Prozesses

Da der Parameter in einer Standard-Datenbank nicht gesetzt ist, kommt automatisch als Default die obige Formel zum Tragen. Zum Ausschalten der Funktionalität muss der Parameter daher explizit auf „0“ gesetzt sein.

## Verbesserungen bei Data Pump

Auch der mit 10g eingeführte Data Pump hat mit 12c viele kleinere Detailverbesserungen erfahren. Ein sehr hilfreiches Feature ist die Möglichkeit, beim Import die Menge der generierten Redo-Log-Informationen zu reduzieren. Dies kann den Import deutlich beschleunigen. Dazu dient der Parameter „TRANSFORM=DISABLE\_ARCHIVE\_LOGGING:Y“. Wenn dieser gesetzt

```
SQL> CREATE INDEX i ON t(a);
SQL> CREATE UNIQUE INDEX ui ON t(a) INVISIBLE;
```

Listing 1

```
SQL> ALTER INDEX i INVISIBLE;
SQL> ALTER INDEX ui VISIBLE;
```

Listing 2

ist, wird das „LOGGING“-Attribut der zu importierenden Objekte während des Imports vorübergehend deaktiviert. Wie bei allen „NOLOGGING“-Operationen gilt, dass auch in diesem Fall nach Ende des Imports ein Backup der Datenbank erfolgen soll. Wenn man die Datenbank im „FORCE LOGGING“-Modus betreibt, wird der Parameter ignoriert.

Bis jetzt war es schwierig, bei Data Pump festzustellen, wie lange die einzelnen Operationen dauern. Der neue Parameter „LOGTIME“ sorgt dafür, dass bei jeder Zeile in der Ausgabe von Data Pump der Zeitstempel vorangestellt wird (siehe Listing 3). Die möglichen Optionen sind „NONE“ (keine Zeitstempel), „STATUS“ (nur Bildschirmausgabe), „LOGFILE“ (nur Protokolldatei) und „ALL“. Eine ähnliche Funktion bietet der seit 11g verfügbare Parameter „METRICS“, der für alle Operationen die benötigte Zeit protokolliert (siehe Listing 4). Die Kombination beider Parameter ist natürlich möglich.

Oftmals sollen Daten, die mithilfe einer View aggregiert wurden, in eine andere Datenbank transferiert werden. Bislang musste man dazu mit einem „Create Table As Select“ eine temporäre Tabelle anlegen, diese exportieren und anschließend dropfen. Der neue Parameter „VIEWS\_AS\_TABLES“ erleichtert derartige Exports, denn Data Pump übernimmt das Anlegen und Löschen der temporären Tabelle, beispielsweise mit „expdp [...] views\_as\_tables=<owner>.<view\_name>“.

Beim Import muss ebenfalls „VIEWS\_AS\_TABLES“ angegeben werden. Data Pump erkennt dadurch, dass das zu importierende Objekt eine View war. Im Zielsystem wird die View natürlich als Tabelle abgelegt. Dabei ist es möglich, die Tabelle über den Parameter „REMAP\_TABLE“ umzubenennen. „VIEWS\_AS\_TABLES“ funktioniert auch beim Import über einen Datenbank-Link.

```
[..]
09-JUL-15 16:20:02.320: .. exported "SCOTT"."A"      72.63 MB 1835008 rows
09-JUL-15 16:20:02.380: .. exported "SCOTT"."B"      8.125 MB  73978 rows
[..]
```

Listing 3

```
[..]
.. exported "SCOTT"."A" 72.63 MB 1835008 rows in 3 seconds
.. exported "SCOTT"."B" 8.125 MB  73978 rows in 1 seconds
[..]
```

Listing 4

```
set serveroutput on
DECLARE
out_sql CLOB;
in_sql  CLOB;
BEGIN
in_sql := ,select * from scott.v_emp_dept where deptno=10`;
dbms_utility.expand_sql_text (in_sql,out_sql);
dbms_output.put_line(out_sql);
END;
/

SELECT "A1"."EMPNO" "EMPNO","A1"."ENAME" "ENAME","A1"."JOB"
"JOB","A1"."MGR" "MGR","A1"."HIREDATE" "HIREDATE","A1"."SAL"
"SAL","A1"."COMM" "COMM","A1"."DEPTNO" "DEPTNO","A1"."DNAME"
"DNAME","A1"."LOC" "LOC" FROM
(SELECT "A3"."EMPNO" "EMPNO","A3"."ENAME" "ENAME","A3"."JOB"
"JOB","A3"."MGR" "MGR","A3"."HIREDATE" "HIREDATE","A3"."SAL"
"SAL","A3"."COMM" "COMM","A3"."DEPTNO" "DEPTNO","A2"."DNAME"
"DNAME","A2"."LOC" "LOC" FROM SCOTT."EMP" "A3",SCOTT."DEPT" "A2" WHERE
"A2"."DEPTNO"="A3"."DEPTNO")
"A1" WHERE "A1"."DEPTNO"=10
```

Listing 5

## DBMS\_UTILITY.EXPAND\_SQL\_TEXT

Eine View holt sich ihre Daten aus einer View, die ihrerseits wieder auf einer anderen View beruht. Ab einer gewissen Verschachtelungstiefe ist es mühselig herauszufinden, auf welche Tabellen die Zugriffe denn nun gehen. 12c bringt dafür die neue Prozedur „EXPAND\_SQL\_TEXT“ im Package „DBMS\_UTILITY“. Die Prozedur erhält als Input einen SQL-Text und erweitert ihn so, dass View-Zugriffe durch Zugriffe auf die darunterliegenden Tabellen ersetzt werden. Auch wenn das Ergebnis etwas unübersichtlich ist, kann es Analysen von verschachtelten Views doch sehr erleichtern (siehe Listing 5).

## Row-Limiting-Clause / Top-N-Queries

Bis 11g war es aufwändig, die ersten „N“ Datensätze einer sortierten Abfrage anzeigen zu lassen. Inline-Queries oder die „WITH“-Klausel waren erforderlich. In 12c vereinfacht die „Row Limiting“-Klausel solche Abfragen (siehe Listing 6).

Entscheidend ist die letzte Zeile: „OFFSET“ legt fest, wie viele Datensätze am Anfang der sortierten Abfrage nicht ausgegeben werden sollen. „FETCH NEXT“ gibt an, wie viele Datensätze angezeigt werden sollen. Damit eignet sich diese Klausel sehr gut, um in einer Bildschirm-Anwendung durch die Ergebnismenge zu scrollen. Ergänzend gibt es noch die Variante „FETCH FIRST n PERCENT ROWS ONLY“, mit der ein prozentualer Anteil der Daten angezeigt werden kann. Auch „Ergebnisgleichheit auf dem letzten Platz“ kann berücksichtigt werden. Ein „FETCH FIRST 3 ROWS WITH TIES;“ gibt nicht die ersten drei Datensätze aus, sondern die Datensätze, die unter den ersten drei gemäß dem festgelegten Sortierkriterium sind.

## SQL-Befehle im RMAN

Der Recovery Manager und SQL wachsen weiter zusammen. Am RMAN-Prompt können jetzt auch beliebige SQL-Befehle ausgeführt werden, etwa „RMAN> select \* from scott.emp;“. Die SQL-Anweisung ist somit nicht mehr erforderlich. Auch „DESCRIBE“-Befehle können direkt im RMAN ausgeführt werden.

## SQL\*Loader Express Mode

Der SQL\*Loader ist ein bewährtes, sehr mächtiges Werkzeug zum Einlesen von ASCII-Dateien in die Datenbank. Über die Jahre hat Oracle einen sehr mächtigen Parser bereitgestellt, mit dem der SQL\*Loader

auch komplexe Dateien in die Datenbank einlesen kann. In der Regel wird der SQL\*Loader jedoch meist für einen Fall gebraucht, das Einlesen einer CSV-Datei in die Datenbank. Genau dieses Szenario hat Oracle mit dem „SQL\*Loader Express Mode“ deutlich vereinfacht. Zum Einlesen einer CSV-Datei reicht im einfachsten Fall der Befehl „sqlldr username TABLE=<tabelle>“. Dabei greifen zahlreiche Voreinstellungen:

- Die einzulesende Datei heißt „<tabelle>.dat“
- Die Reihenfolge der Spalten in der CSV-Datei entspricht der Spaltenreihenfolge der Tabelle
- Die Werte sind Komma-separiert und es gibt einen Datensatz pro Zeile

Wenn SQL\*Loader im Express Mode genutzt wird, erfolgt das Einlesen der Daten über eine externe Tabelle, den „SQL\*Loader in der Datenbank“. Wenngleich der Express-Mode so konzipiert ist, dass er ohne weitere Parameter auskommt, so können doch einige wenige Einstellungen über Parameter angepasst werden. Einzelheiten dazu gibt es im Kapitel 13 des „Oracle 12c Database Utilities Guide“.

Neben den dargestellten neuen Funktionen gibt es auch weitere hilfreiche Features, die die engen Kriterien (Verfügbarkeit in der Standard Edition) leider nicht erfüllen, die wir aber trotzdem nicht verschweigen wollen.

```
SQL> SELECT ename,sal
2 FROM scott.emp
3 ORDER BY sal
4 OFFSET 5 ROWS FETCH NEXT 5 ROWS ONLY;
```

Listing 6

### Online Datafile Move (Enterprise Edition)

Wie alle Online-Operationen ist auch das Verschieben von Datenbank-Dateien im laufenden Betrieb der Enterprise Edition vorbehalten: „ALTER DATABASE MOVE DATAFILE '<alter\_Name>' TO '<neuer\_Name>'“. Diesen Befehl haben viele DBAs in den älteren Oracle-Versionen schmerzlich vermisst. Eben mal schnell einen Tippfehler im Dateinamen korrigieren oder eine Datei in ein größeres Dateisystem verschieben. Früher waren solche Aktionen mit einem aufwändigen Offline-Setzen der Datei verbunden – jetzt ist es ein einfacher Befehl. Dabei ist zu beachten, dass Oracle auch beim einfachen Umbenennen einer Datei intern eine Kopie der Datei macht, daher muss ausreichend Platz im Dateisystem vorhanden sein.

### READ-Privileg (12.1.0.2)

In Oracle 12.1.0.2, derzeit nur in der Enterprise Edition verfügbar, ist ein neues Recht hinzugekommen, das „READ“-Privileg. Es entspricht dem „SELECT“-Recht mit dem kleinen, aber feinen Unterschied, dass ein „SELECT FOR UPDATE“ nicht möglich ist.

Dadurch kann ein Benutzer, der nur das „READ“-Recht hat, keine Datensätze sperren – es ist also ein wirkliches reines Lese-Recht.

### Fazit

Die Datenbank 12c ist nicht nur Multi-tenant-Option, In-Memory-Option oder Information Lifecycle Management. Es gibt viele kleine hilfreiche Features, die das Upgrade auf die aktuelle Datenbank-Version lohnenswert machen.



Markus Flechtner  
markus.flechtner@trivadis.com

## SQL Developer für DBAs

Ernst Leber, MT AG

Die größte Stärke des SQL Developer liegt darin, dass er kostenlos ist und bei der Installation der Oracle-Datenbank mit installiert wird. Damit ist das Tool auf fast jeder Oracle-Datenbank in einer mehr oder weniger aktuellen Version verfügbar. Das macht den SQL Developer ähnlich wie den Texteditor „vi“ interessant, da er quasi universell zur Verfügung steht. Dieser Artikel zeigt die Erfahrungen des Autors mit dem SQL Developer aus DBA-Sicht.

Im Rahmen seiner Aufgaben als DBA nutzt der Autor hauptsächlich die DBA-Ansicht und die Berichtsfunktionen des SQL Developer. Der Schwerpunkt liegt dabei eindeutig auf den benutzerdefinierten Berichten und der Möglichkeit, diese auf

andere Systeme zu übertragen. Der Artikel basiert auf dem Vortrag „SQL Developer 4 DBAs“, den der Autor am 16. Juni 2015 auf der DOAG 2015 Datenbank in Düsseldorf gehalten hat (siehe „<http://www.slideshare.net/ErnstLeber/sql-dev4-dba-2>“).

### Allgemeines

Der SQL Developer enthält Funktionen für die Datenbank-Administration, die mit jedem neuen Release erweitert und verbessert werden. Je nach Version des SQL Developer sind einige der hier vorgestellten Standard-