

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-

```
select owner,object_type, status,count(*)
  from dba_objects
 -- where status = 'INVALID'
 group by owner, object_type, status
 order by owner, object_type, status;
```

Listing 1: SQL zur Code-Vorlage

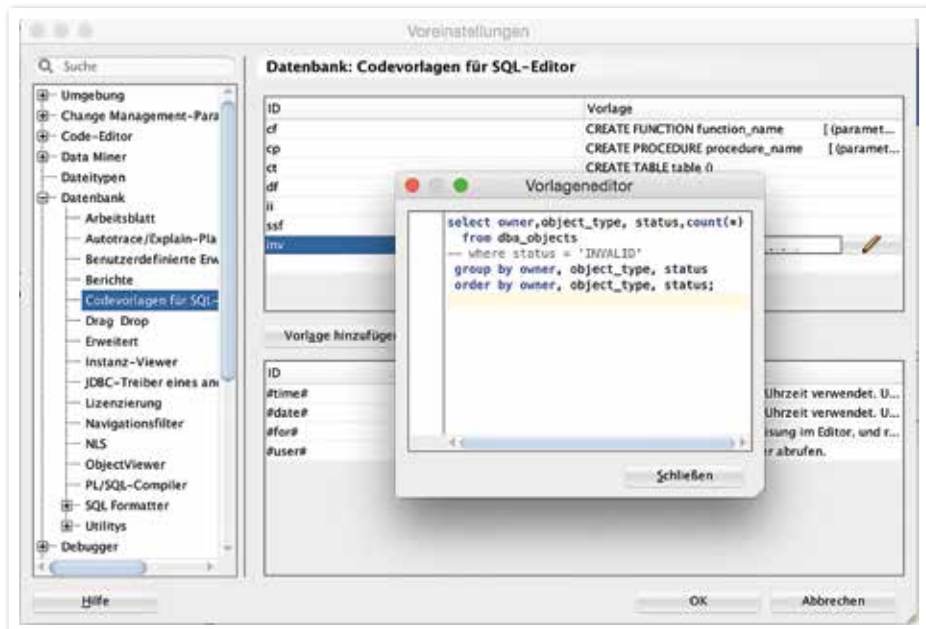


Abbildung 1: Code-Vorlagen eingeben

Funktionen noch nicht enthalten. Dieser Artikel basiert auf der Version 4.1.0.19.

Aufgrund der Erweiterbarkeit von Reports und des Connect Tree können aber auch gerade ältere Versionen des SQL Developer mithilfe einiger XML-Dateien so angepasst werden, dass man als DBA nicht auf gewohnte Funktionen verzichten muss und damit auch in fremden Umgebungen sein persönliches Werkzeug zur Verfügung hat, um die Datenbank zu analysieren und zu administrieren. Eine Sammlung der Berichte des Autors steht unter „<https://dl.dropboxusercontent.com/u/71217794/ele-oracle.xml>“. Diese lassen sich nach einem Rechtsklick auf „Benutzerdefinierte Berichte“ und „Bericht öffnen“ in den SQL Developer laden.

Erster Aufruf

Dieser Artikel geht davon aus, dass bereits erste Erfahrungen mit dem SQL Developer vorliegen und daher die Grundeinstellungen für Verbindungen zu Datenbanken vertraut sind. Dennoch sollte man sich, falls nicht schon erfolgt, beim nächsten Aufruf des SQL Developer die Zeit

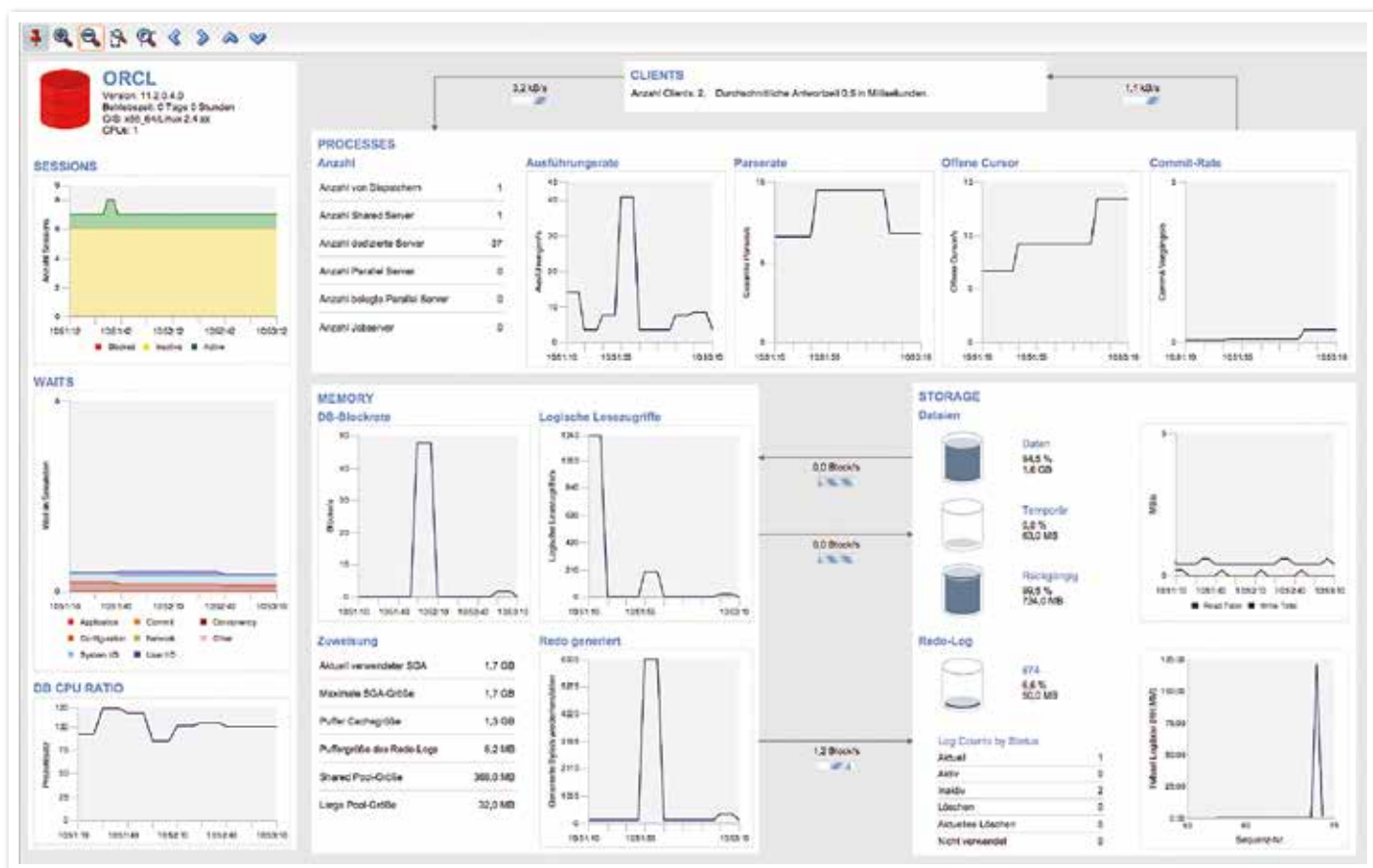


Abbildung 2: Instanz-Übersicht

nehmen und einige grundlegende Einstellungen vornehmen. Unter „Extras“ und „Voreinstellungen“ stehen im Punkt „Datenbank“ einige Einstellungen, die man grundsätzlich vornehmen sollte:

- Unter „Arbeitsblatt“ den Punkt „Raster im Schachbrettmuster oder Streifenmuster“ anwählen
- Unter „NLS“ die gewünschten Formate für die Anzeige von Datum und Uhrzeit einstellen

Ein weiterer Tipp: In den Einstellungen unter „Datenbank“ den Punkt „Codevorlagen für SQL-Editor“ wählen, dort „Vorlage“ hinzufügen und als „ID“ „inv“ eingeben sowie unter „Vorlage“ folgendes SQL-Statement eintippen (siehe Listing 1). Wenn man in Zukunft in einem SQL-Arbeitsblatt „inv“ angibt, wird einem automatisch das komplette SQL-Statement angeboten (siehe Abbildung 1).

Der Dialog „Einstellungen“ bietet noch viele weitere Möglichkeiten für persönliche Anpassungen. Individuelle Einstellungen werden bei der Installation einer neueren Version des SQL Developer automatisch übernommen. Zu den Standard-Funktionen gehören die fest eingebauten Berichte und die separate DBA-Ansicht. Je nach Version des SQL Developer ist der Inhalt der DBA-Ansicht eher rudimentär. Die Funktionen für die Benutzerverwaltung und die Administration des Job-Scheduler rechtfertigen jedoch den Einsatz des Tools allemal.

DBA-Ansicht

Bevor man die Ansicht „DBA“ nutzen kann, muss zunächst eine Verbindung hinzugefügt werden. Dafür dient das grüne Plus-Zeichen oben unter „DBA“. Die ausgewählte Verbindung sollte als „User system“ oder „sys“ definiert sein. Unter „DB Instanz“ steht eine grafische Übersicht über die Vorgänge in der Datenbank, die

ohne weitere Installationen auf der Ziel-Datenbank auskommt (siehe Abbildung 2).

Das ist bei Weitem nicht alles, was die DBA-Ansicht zu bieten hat. Wer öfter mit Data Pump arbeitet, sollte sich den Menüpunkt „Data Pump“ ansehen. Dort stehen Informationen über Export- und Import-Jobs. Ein Rechtsklick auf einen dieser Jobs startet einen Assistenten für Data Pump (siehe Abbildung 3). Er generiert einen PL/SQL-Block für die Steuerung von Data Pump. So erfährt man, dass „expdp“ und „impdp“ auch nur Frontends für „dbms_datapump“ sind.

Backups mit RMAN lassen sich in dem entsprechenden Menüpunkt ebenfalls steuern und anzeigen. Wer die Enterprise Edition nutzt und das Diagnostic und Tuning Pack lizenziert hat, kann unter dem Punkt „Performance“ alle mit diesem Pack verbundenen Funktionen interaktiv steuern und aufrufen.

Ein Tipp: Die Anzeige von AWR- und ASH-Reports ist im SQL Developer für den Autor

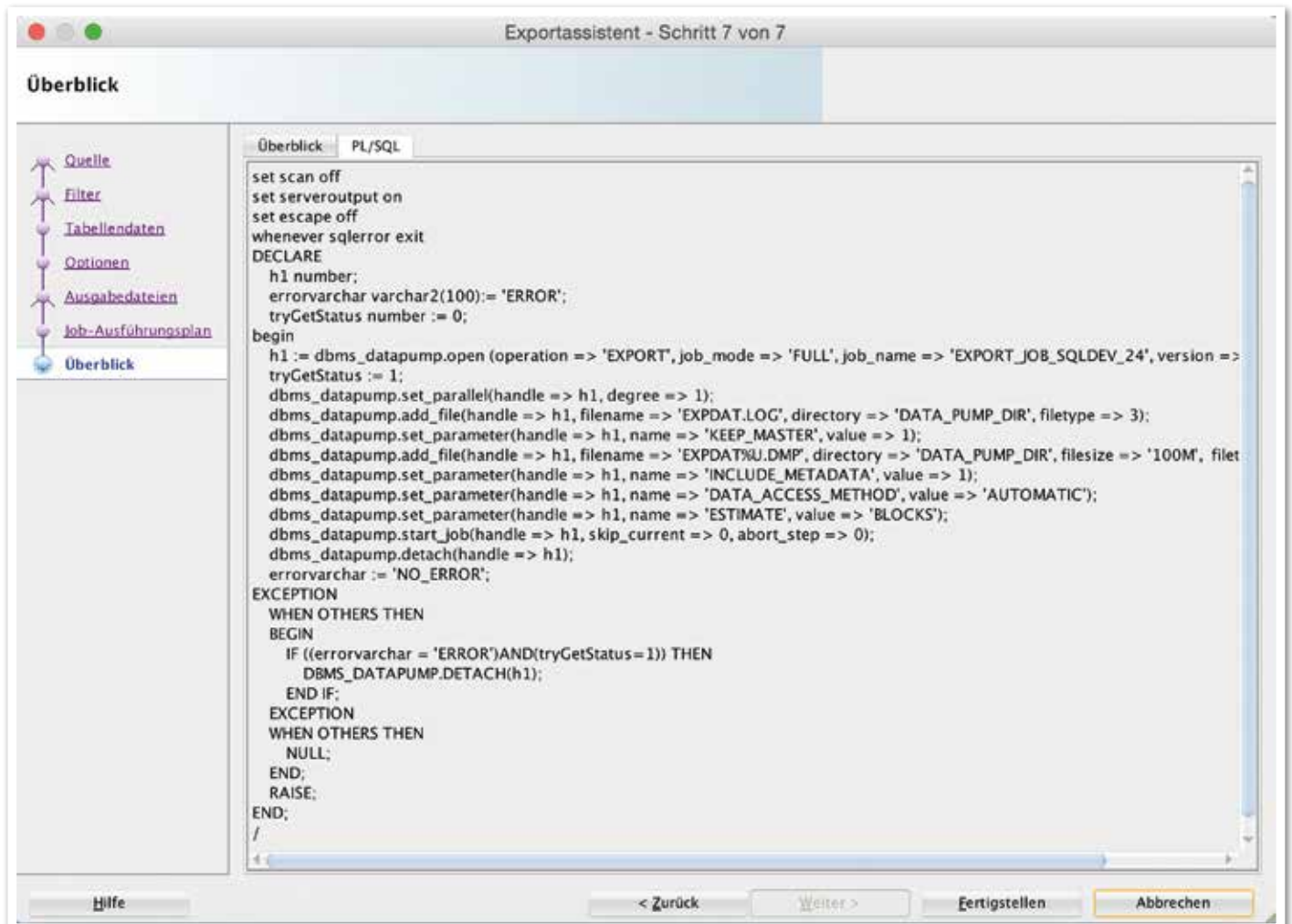


Abbildung 3: Data-Pump-Assistent

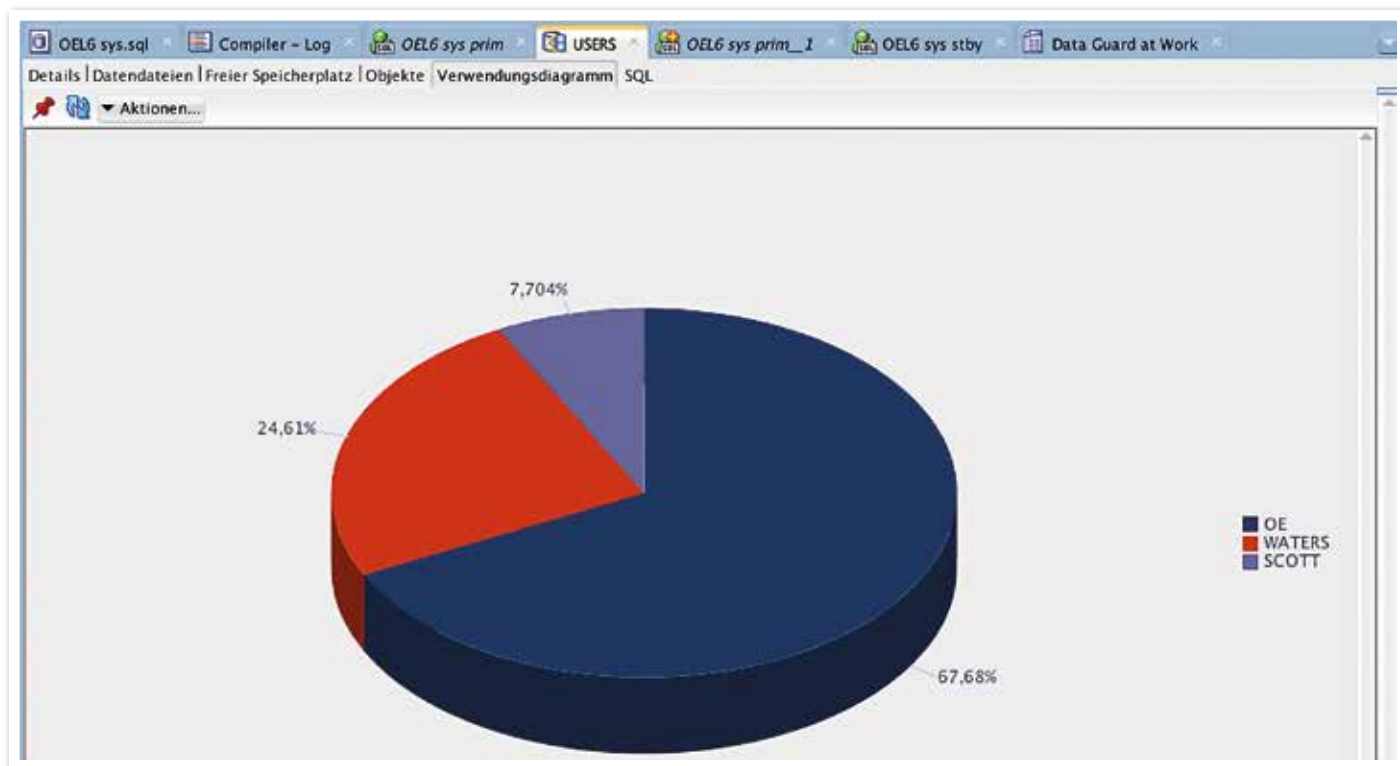


Abbildung 4: Tablespace-Übersicht

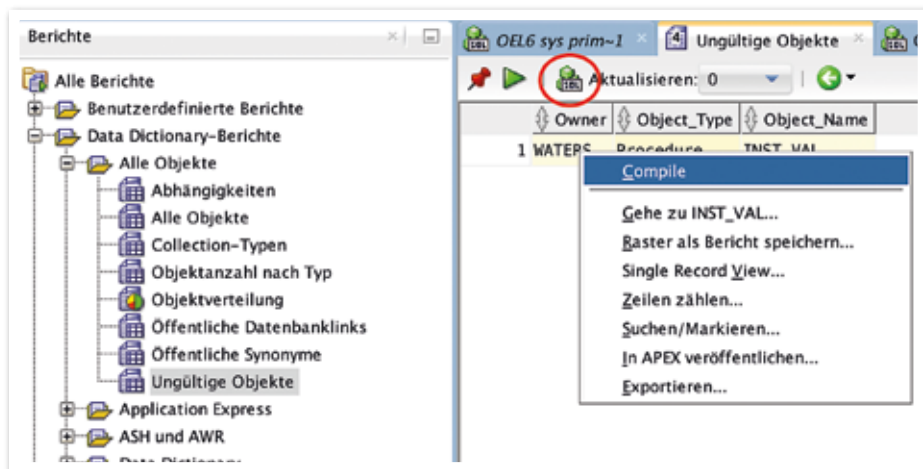


Abbildung 5: Bericht „Ungültige Objekte“

etwas träge. Wenn man oben im Kopf der Seite auf das ganz linke Symbol klickt, wird dieser Report im Standard-Browser aufgerufen.

Als letzter Punkt in der DBA-Ansicht sei noch der Punkt „Speicherung“ genannt. Hier lassen sich unter „Tablespaces“ Daten zu den in der Datenbank vorhandenen Tablespaces anzeigen und administrieren (siehe Abbildung 4).

Standard-Berichte

Die Standard-Berichte im SQL Developer zeichnen sich dadurch aus, dass sie alle Datenbank-Versionen ab 10 unterstützen und

automatisch neue oder nicht vorhandene Spalten in Tabellen berücksichtigen. Falls diese Berichte nicht zu sehen sind, kann man sie über den Menüpunkt „Ansicht“ anzeigen lassen. Unter „Data Dictionary Berichte“ stehen weitere Berichte über die Datenbank. Stöbern lohnt sich hier in jedem Fall, da auch diese Berichte mit jeder neuen Version verbessert werden und neue hinzukommen.

Als Beispiel für die Möglichkeiten der Reports sei hier der Bericht „Ungültige Objekte“ unter „Data Dictionary Berichte“ und „Alle Objekte“ genannt. Nach einem Rechtsklick im Ergebnisfenster klappt ein Menü auf, in

dem ungültige Prozeduren oder Views direkt aus dem Bericht heraus kompiliert werden können. Als weiteren Tipp achtet man in auf das rot eingekreiste Symbol oben in der Titelleiste des Berichts (siehe Abbildung 5). Wer wissen möchte, auf welchem SQL-Statement der aktuelle Bericht basiert, klickt auf dieses Symbol. In einem neuen Fenster wird dann das zugehörige Statement angezeigt.

Benutzerdefinierte Berichte

Anfangs wurde erwähnt, dass der SQL Developer erweitert werden kann. Die einfachste Variante ist die Definition eigener Berichte. Ein Beispiel zeigt die wesentlichen Abläufe beim Erstellen von eigenen Berichten. Als Ausgangspunkt dient ein Statement zum Anzeigen der Statistiken in einer Datenbank (siehe Listing 2 und Abbildung 6).

Nach dem Ausführen des Statements in einem Arbeitsblatt stehen im Ergebnisfenster, hier „Raster“ genannt, die entsprechenden Daten. Ein Rechtsklick in diesem Raster zeigt als Menüpunkt „Raster als Bericht speichern“ an. Im folgenden Dialog gibt man einen Titel für den Bericht ein (siehe Abbildung 7).

Man kann einen benutzerdefinierten Bericht auch erstellen, wenn man unter „Benutzerdefinierte Berichte“ nach einem Rechtsklick „Neuer Bericht“ auswählt und die Daten im folgenden Dialog eingibt. Dieser Bericht hat allerdings einen Nachteil, denn man sieht

hier nur eine Zusammenfassung der Statistiken, Details über das oder die Objekte und deren Statistiken werden nicht angezeigt.

Jetzt kommt ein Feature dieser Berichte ins Spiel, denn man kann beliebig viele Detail-Berichte zu einem Bericht zusammenfassen. Dazu wählt man den Bericht und nach einem Rechtsklick „Bericht bearbeiten“ aus. Jetzt klickt man auf „Untergeordnete Berichte“, dann auf „Untergeordnete hinzufügen“ und gibt einen Namen ein, etwa „Detail“. Jetzt kann man den Menüpunkt „Untergeordnete Berichte“ erweitern und sieht als Unterpunkt den eben eingegebenen Namen „Detail“. Dort findet man den Unterpunkt „SQL-Abfrage“.

Man öffnet diesen, gibt das Statement aus *Listing 3* ein und klickt auf „Anwenden“. Der Master-Bericht ist damit fertig (siehe *Abbildung 8*).

Die Verbindung zwischen dem Master- und dem Detail-Bericht wird über Binde-Variablen hergestellt. Man kann jede Spalte des Master-Statements im Detail-Statement referenzieren, indem man dem Spaltennamen einen Doppelpunkt voransetzt. Falls der Report dennoch nicht funktioniert, sollte man darauf achten, dass keines der SQL-Statements mit einem Semikolon beendet ist (siehe *Abbildung 9*).

Auf eine weitere Besonderheit sei hier noch hingewiesen. DBAs betreuen Daten-

banken mit unterschiedlichen Versionen. In den Reports kann man diesem Umstand Rechnung tragen, indem man innerhalb des Berichts Varianten der SQL-Statements anlegt und diese mit Minimal- und Maximalversion eingrenzt (siehe *Abbildung 10*).

Berichte mit Aktionen lassen sich mit der aktuellen Version des SQL Developer leider noch nicht direkt definieren. Dafür ist die XML-Datei, in der der Report definiert ist, zu bearbeiten. Eine detaillierte Beschreibung, wie man einen Bericht mit Aktionen definiert, steht in den Folien des Autors zu seinem Vortrag „SQL Developer 4 DBAs“ (siehe „<http://www.slideshare.net/ErnstLeber/sql-dev4-dba-2>“).

Wer bereits seine erstellten Berichte auf einem anderen Rechner nutzt oder diese weitergeben möchte, kann sie in einer XML-Datei speichern. Nach einem Rechtsklick auf den Bericht wählt man den Menüpunkt „Bericht speichern“ und legt den Bericht in einer Datei ab (siehe *Abbildung 11*). Wenn mehrere Berichte in Ordnern gesammelt sind, kann man mit der gleichen Methode einen kompletten Ordner in einer XML-Datei speichern. Geladen werden diese Berichte mit einem Rechtsklick auf „Benutzerdefinierte Berichte“ und „Bericht öffnen“.

Fazit

Der SQL Developer wird mit jeder neuen Version erweitert und verbessert. Dieser Artikel gibt eine kurze Übersicht über die Funktionen. Für seine Arbeit ist der SQL Developer für den Autor mittlerweile unverzichtbar geworden. Zum Abschluss noch ein Tipp: Wer einen Data Modeler benötigt, schaut unter „Datei“ und „Data Modeler“ nach.



Ernst Leber
ernst.leber@mt-ag.com

```
select owner, trunc(last_analyzed) last_analyzed, STALE_STATS, count(*) cnt
from dba_tab_statistics
group by owner, trunc(last_analyzed), STALE_STATS
order by owner, trunc(last_analyzed), STALE_STATS;
```

Listing 2: Anzeige der Datenbank-Statistiken

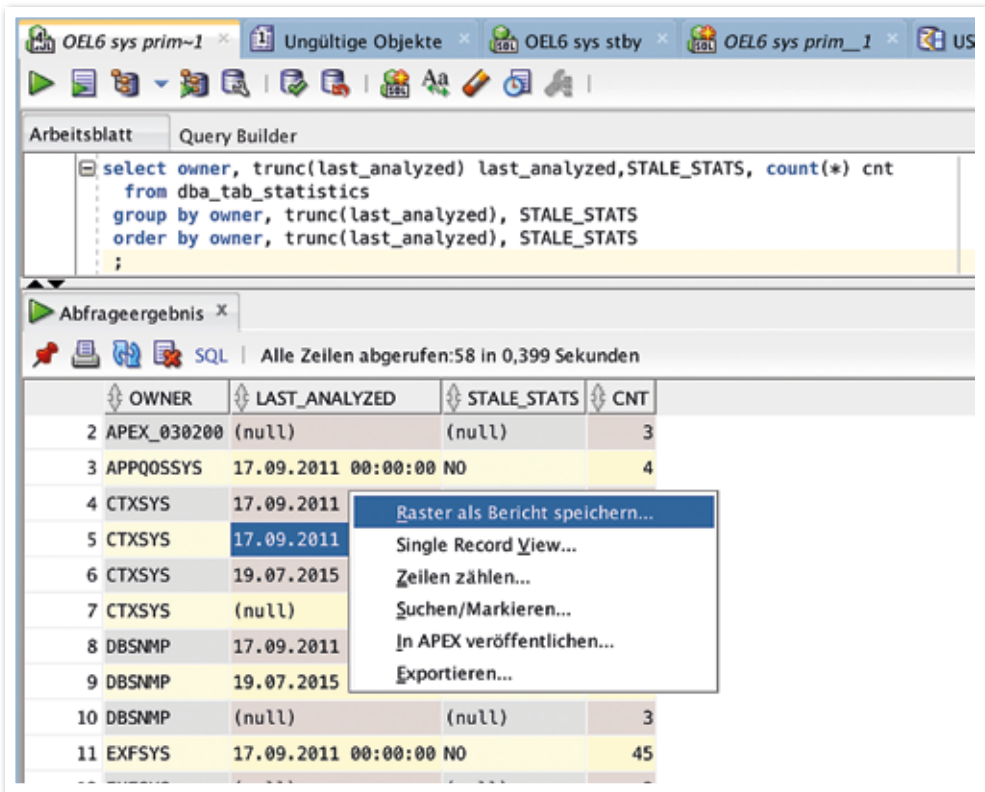


Abbildung 6: Anzeige Statistiken

```
select *
from dba_tab_statistics
where owner = :OWNER
and trunc(last_analyzed)||'x' = :LAST_ANALYZED||'x'
and stale_stats||'x' = :STALE_STATS||'x'
order by owner, table_name
```

Listing 3: Anzeige „Details der Datenbank-Statistiken“

Hinweis: Die Abbildungen 7 bis 11 finden sie online unter: http://www.doag.org/go/news/201505/Leber_Abb