

Erweiterte Funktionen des SQL Developers

Autor: André Schulte

Der SQL Developer ist eine grafische Version von SQL*Plus und wird von Datenbank-Entwicklern hauptsächlich für das Schreiben komplexer SQL-Statements verwendet. Der Artikel gibt einen Überblick zu den Funktionalitäten des SQL Developers und beschäftigt sich mit den erweiterten Funktionen: Wie wird mit dem SQL Developer 1.5 eine MySQL-Datenbank zu einer Oracle Datenbank migriert und diese dann mit grafischen Reports ausgewertet.

Der SQL Developer bietet die Möglichkeit, sich mit Oracle Datenbanken zu verbinden und einen SQL- oder PL/SQL-Code zu entwickeln, auszuführen oder zu debuggen. Man kann Ausführungspläne erstellen und Datenbank-Objekte betrachten, erstellen, ändern oder löschen. Eine Anbindung über JDBC mit Datenbanken anderer Hersteller wie MySQL, Microsoft SQL Server, Sybase Adaptive Server oder Microsoft Access ist möglich. Mit dieser Verbindung können Metadaten ausgelesen und SQL-Statements ausgeführt werden. Darüber hinaus gibt es unter anderem folgende wichtige Funktionalitäten:

- Unterstützung einiger SQL*Plus-Kommandos, wie desc[ribe] oder ,@' zum Aufrufen von Skripten
- Komfort-Funktionen im SQL-Worksheet wie Explain plan, Auto Trace, SQL-History, Smart Completion (Einblendung von Tabellen-Namen, Spalten-Namen beim Schreiben eines Statements), Popup-Beschreibungen für Objekte (Rechte Maustaste Describe) sowie Speichern und Einfügen von SQL-Snippets nach Kategorien
- Vergleichen und Kopieren von Schemata
- Erweiterte Suche nach Objekten mit Wildcards. Findet man beispielsweise eine Tabelle oder einen Trigger, kann das Objekt per Doppelklick geöffnet werden.
- Unterstützung für die Konfigurationsmanagement-Tools CVS und Subversion
- Wizard für das Exportieren von Metadaten und Daten in ein SQL-Skript bestehend aus DDL und DML-Statements

- Migrieren von Datenbanken sowie Nutzen von Standard-Reports und Entwickeln eigener Reports – näheres in diesem Artikel

Bei kleineren Firmen führen Entwickler einfache Migrations-Szenarien häufig selbst durch. Quick Migrate Wizard ermöglicht es, schnell ein Ergebnis zu erzielen. Der SQL Developer unterstützt aber auch komplexe Migrationen durch die Erstellung von Skripten für die Entladung von Daten sowie von SQL*Loader-Skripten für das Laden von Daten. Die Vorgehensweise zur Installation des SQL Developers und der Aufbau einer Oracle-Datenbank-Verbindung ist nicht Teil des Artikels – kann allerdings dem SQL Developer User's Guide entnommen werden. Der Oracle SQL Developer ist ein kostenfreies Produkt, das durch den Oracle Support für alle Kunden mit einem Oracle Datenbank-Support-Vertrag unterstützt wird. Die Migrations-Funktionalität in SQL Developer ist eine Weiterentwicklung der Oracle Migration-Workbench. Es werden derzeit folgende Datenbanken unterstützt:

- Microsoft Access
- Microsoft SQL Server
- MySQL
- Sybase Adaptive Server

Sobald der SQL Developer alle unterstützten Plattformen der Oracle Migration-Workbench unterstützt, fällt die Oracle Migration-Workbench aus dem Support, siehe hierzu auch Oracle Migration Tools Statement of Direction vom Mai 2008. Eine Einarbeitung in die Migration mit SQL Developer lohnt sich also. Schritte zur Vorbereitung einer Migration sind:

1. Die Anforderungen an das Migrationsprojekt festlegen
2. Den Aufwand abschätzen
3. Auswirkungen auf die Produktion abschätzen
4. Auswirkungen auf die Applikation abschätzen
5. Die Migration planen

Diese Schritte geben einen groben Überblick über die Komplexität einer Migration. Wer ein konkretes Migrationsprojekt plant, sollte sich im SQL Developer User's Guide über die genauen Tätigkeiten informieren, da dieser Artikel nur die Durchführung einer einfachen Migration ohne Rahmenbedingungen beschreibt. Als Datenbank wird MySQL Version 5.0.22 eingesetzt sowie die Sakila Sample Database Version 0.8 als Schema verwendet. Sie ist ein Verleihshop für Filme, der aus 16 Tabellen inklusive Daten, Indizes und Triggern besteht.

Vorbereitung

Zur Vorbereitung wird der SQL Developer auf den Zugriff einer MySQL-Datenbank mittels JDBC konfiguriert. Dazu muss man zuerst einen Treiber herunterladen. In diesem Fall den MySQL Connector/J 5.1 von der MySQL.com-Webseite. Korrekt installiert, ist im unteren Bereich des Fensters ein MySQL-Tab vorhanden, der die Wahl von Hostname und Port der Datenbank anbietet. Nach einem erfolgreichen Connection Test kann der Dialog geschlossen werden.

Für das Migration Repository zum Zwischenspeichern von Metadaten sowie für das Import-Schema sind zwei neue Accounts in der 11g-Datenbank notwendig:

```
SQL> grant resource, connect,
create view to test identified
by test;
SQL> create user mig_repository
identified by mig_repository;
```

Im SQL Developer erstellt man eine Connection für den Repository Account und für den Import Account (User test). Hier heißen sie repository und DOAG_Q32008. Der repository Account ist unter ‚Migration -> Repository Management -> Create Repository‘ als Repository anzugeben.

Migration

Der Aufruf des Wizards erfolgt im SQL Developer über den Menüpunkt ‚Migration -> Quick Migrate‘. In den Dialog-Schritten werden zuerst die MySQL-Connection, danach die Oracle Connection als Zielschema angegeben. Letzte Connection ist das Migration Repository (siehe Abb. 1). Der nächste Schritt des Assistenten ist ein Test, bei dessen Ausführung eine Privilege-Warnung (Create Role not present) erscheint, die ignoriert werden kann, da nur ein Schema migriert wird. Bei der Migration mehrerer Schemata kann man die fehlenden Privilegien im SQL

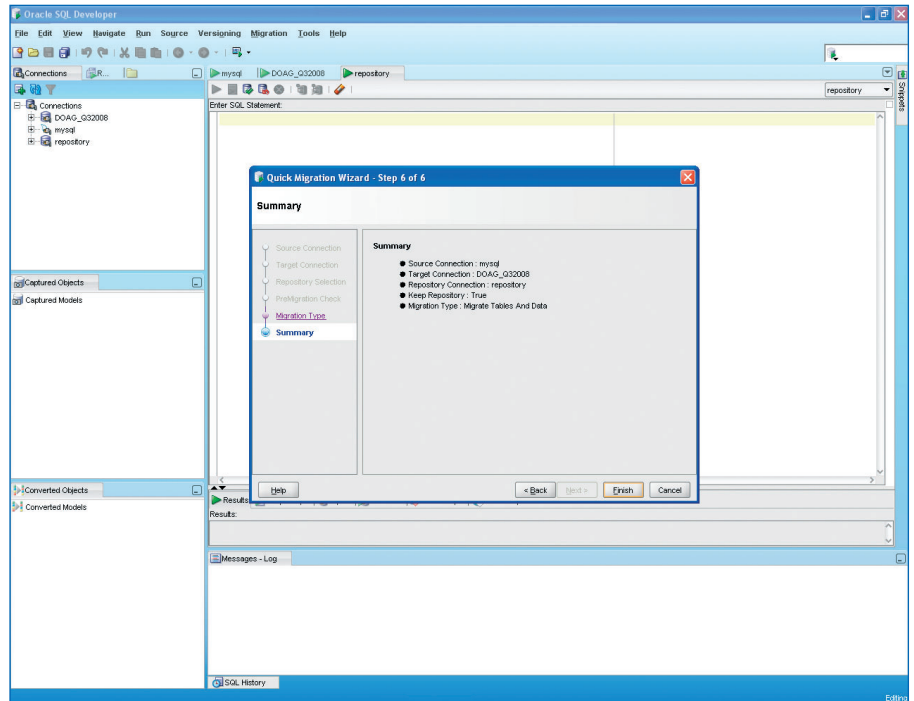


Abbildung 1: Zusammenfassung Quick Migration Wizard

Developer User's Guide nachlesen. Nun geht darum, was migriert werden soll. Zur Auswahl stehen Tabellen, Daten sowie alles Migrieren. Man wählt Tabellen und Daten und kommt zur Zusammenfassung.

Anschließend führt der Wizard folgende Schritte durch:

- **Online Capturing der Source Database**
Der Prozess, Metadaten aus der Quelldatenbank zu lesen, heißt Capturing. Das Capturing kann auch ohne Assistent gestartet werden. Beim offline Capturing wird ein Skript erzeugt, das die Metadaten aus der Quelldatenbank ausliest. Dabei entsteht eine Object-Capture-Properties-Datei (.ocp), die in SQL Developer eingelesen wird.
- **Converted Model erstellen**
Das Converted Model ist die Struktur der Zieldatenbank, die aus dem Captured Model erstellt wird und Tabellen, Indices, Trigger und andere Objekte enthält. Die Prozeduren werden dabei automatisch in Oracle PL/SQL umgewandelt. Nur bei einem Fehler wird der Original-Code übernommen. Es ist dann ein manuelles Nacharbeiten notwendig.
- **Generate und Build**
Bei diesen Schritten wird aus dem Converted Model ein DDL-Skript erstellt, das die Quelldatenbank nachbilden soll. Anschließend wird es ausgeführt (Build).
- **Online Data Move**
Übertragen der Daten von der Quelldatenbank zur Zieldatenbank. Dabei nutzt der SQL Developer die

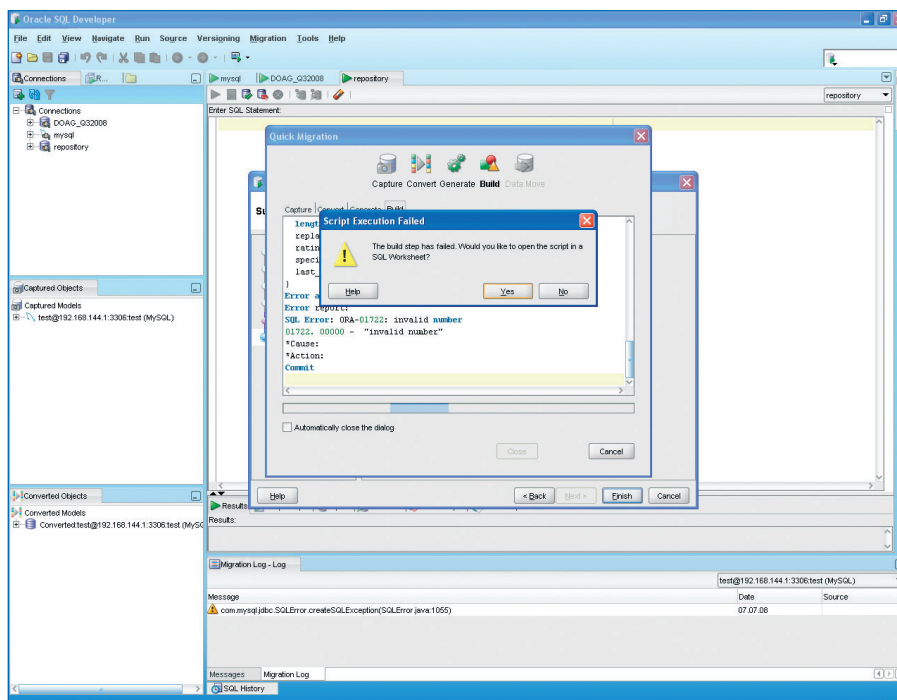
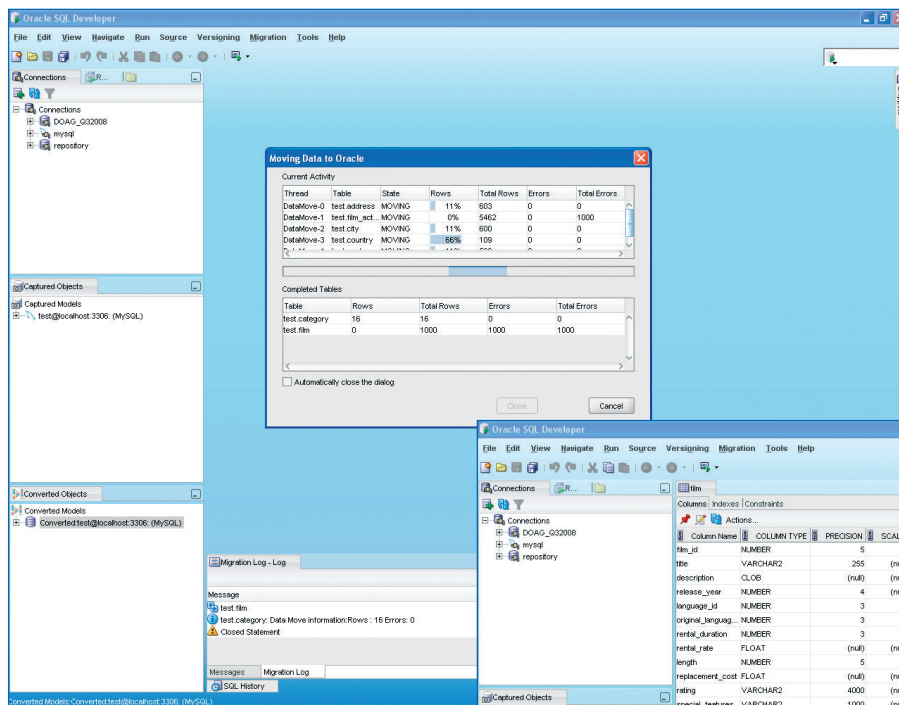


Abbildung 2: Quick Migration Wizard – Fehler beim Build



jects das zuletzt erstellte Converted Model mit der rechten Maustaste aus und wählt Move Data aus. Im Dialog muss noch MySQL als Quelldatenbank und DOAG_Q32008 als Zieldatenbank angegeben werden.

Abbildung 3 zeigt, dass mehrere Threads gleichzeitig für die Übertragung genutzt werden. Zudem ist ersichtlich, dass bei der Tabelle Film ein Fehler aufgetreten ist, da 0 der 1000 Datensätze

Abbildung 3: Übertragung der Daten

angegebenen Connections, um die Daten zu lesen und ohne Zwischenspeicherung in die Oracle Datenbank zu schreiben. Im Gegensatz dazu wird beim Offline Data Move ein Skript zum Entladen der Daten für die Quelldatenbank erstellt sowie ein SQL*Loader Skript zum Laden der Daten in die Oracle Datenbank. Offline Data Move ist für große Datenmengen geeignet.

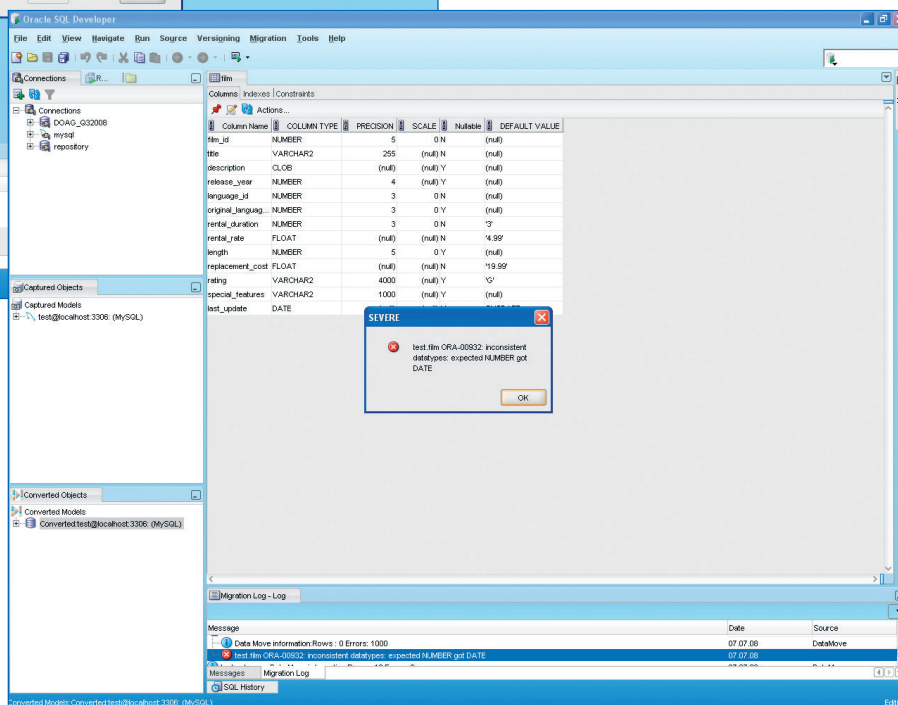
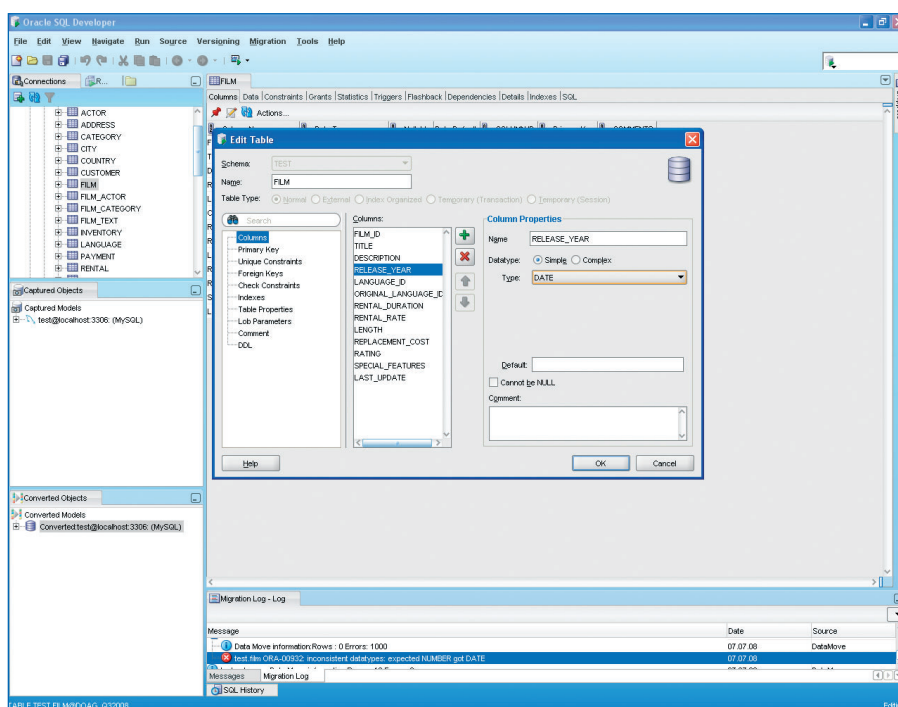


Abbildung 4: Fehlermeldung

Abbildung 5: SQL Developer Edit-Table-Dialog

Im Idealfall ist die Migration nun beendet. Bei der Ausführung des Skripts ist allerdings ein Fehler aufgetreten (siehe Abbildung 2).

Der Versuch, eine Domain-Index mit mehr als einer Spalte aufzubauen hat einen Fehler produziert. Man beendet den Assistenten und öffnet mit dem SQL-Skript das SQL-Worksheet. Dort werden der fehlerhafte Index und die bereits ausgeführten Statements auskommentiert sowie der Rest des Skripts ausgeführt. Dabei treten keine weiteren Fehler auf. Damit wäre die Build Phase beendet. Es müssen aber noch die Daten in das Oracle-Schema übertragen werden. Da sich der Wizard beim Auftreten des Fehlers geschlossen hat, muss das jetzt manuell ausgeführt werden. Dazu wählt man im linken Teil des Fensters unter Converted Ob-



übertragen wurden. Den Fehler findet man bei der Analyse des Migration Logs. Für eine Spalte der Tabelle Film wurde ein Number-Typ erwartet aber ein Date-Typ übertragen (siehe Abbildung 4).

Zur Behebung des Fehlers muss in der Tabelle Film das Feld ausfindig gemacht werden, der Daten-Typ manuell auf Date gesetzt und anschließend nur die Daten für die Tabelle Film neu übertragen werden. Bei Betrachtung der beiden Datenbank-Strukturen wurde klar, dass der RELEASE_YEAR-Daten-Typ der Tabelle Film ein Date-Typ sein muss. Dies kann man mit dem Edit-Table-Dialog ändern, der mit der rechten Maustaste auf der Tabelle Film ‚Edit‘ geöffnet wird:

Nach einer Änderung des Datentyps wird aus dem Captured Model ein neues Converted Model erstellt, und, wie im Bild dargestellt, nur die Tabelle Film ausgewählt. Mit einem Rechtsklick auf das Converted Model wählt man Move Data aus: es werden nur die Daten für die Tabelle Film übertragen.

Auswertung

Nach der Übertragung des Schemas ist dessen Auswertung über die Reporting-Funktionen des SQL Developers möglich. Es gibt Reports, die bereits über Datenbank-Objekte oder -Zustände, wie Tabellen ohne Index auf einem Foreign Key, Auskunft geben. Um einen Überblick über die Reports zu erhalten, werden hier einige vorgestellt:

- *All Objects -> All Objects*
Dieser Bericht kann mit Eingabe einer Bind-Variablen gestartet werden und zeigt entweder alle Objekte eines Schemas oder alle Objekte der Datenbank an
- *All Objects -> Invalid Objects*
Zeigt alle Objekt an, die invalid sind, wie beispielsweise Prozeduren, die nicht kompiliert werden können
- *Database Administration -> Locks by User*
Gibt Informationen über Locks in der Datenbank inklusive Nutzer an
- *Database Administration -> Storage -> Locks-> Free Space*
Auswertung auf Tablespace-Ebene.

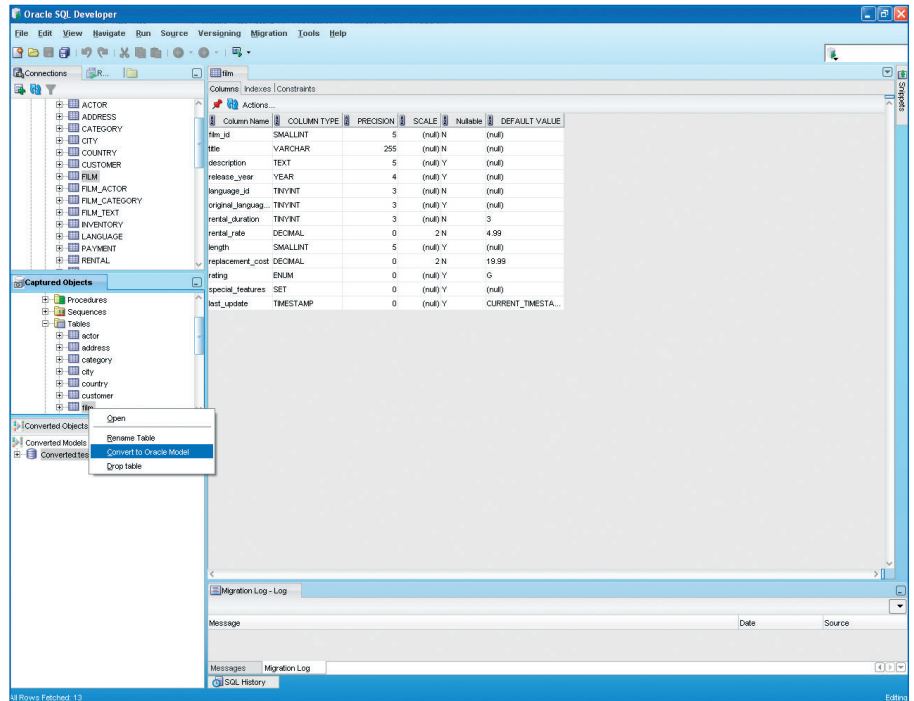


Abbildung 6: Erstellung Converted Model für die Tabelle Film

- *Database Administration -> Top SQL*
Hier gibt es mehrere Reports, sortiert nach Buffer Gets, CPU-Zeit, Disk Reads oder Anzahl der Executions
- *Tables -> Quality Assurance -> Tables with Unindexed Foreign Keys*
Dieser Report zeigt alle Tabellen an, die Foreign Keys ohne einen Index haben
- *Migration Report -> Migration Summary*
Gibt Auskunft über das Captured und Converted Model aber nicht über die Inhalte der übertragenen Tabellen. Der Report muss mit der Connection aufgerufen werden, die als Repository verwendet wird

Es gibt aber keinen Report, der eine Zusammenfassung des Schemas anzeigen kann. Hierfür kann man jedoch einen User-Defined-Report selbst erstellen (siehe Listing 1).

Damit die Auswertung Werte liefert, müssen zuerst die Statistiken berechnet werden. Im SQL Developer klickt man dazu mit der rechten Maustaste auf die Connection DOAG_Q32008 und wählt Gather Schema Statistics. Im folgenden Dialog gibt man für Estimate Percent 5 an.

Dazu sollen zwei Auswertungen gegenüber gestellt werden. Zuerst öffnet man im Report-Tab die Auswertung Data Dictionary Reports -> All Objects -> All Objects. Es wird nach einer Bind-

```
select ut.table_name, ut.num_rows,
(select count(*) from USER_TAB_COLUMNS utc where ut.table_name = utc.table_name) colAnz,
(select count(*) from user_indexes ui where ut.table_name = ui.table_name) ind,
(select count(*) from user_triggers utr where utr.table_name = ut.table_name) trg,
(select SUM (b.bytes) / 1024 / 1024 sizeMB
FROM user_extents b WHERE ut.table_name = b.segment_name
GROUP BY ut.table_name) sizeMB
from user_tables ut order by sizeMB desc
```

Listing 1

The screenshot shows the Oracle SQL Developer interface. On the left, a tree view displays the database schema. The main window shows a list of objects with columns for Object Name, Status, Date Created, and Last DDL. Below this, a table displays data for various tables, including RENTAL, PAYMENT, FILM, and ACTOR.

Table Name	Number of Rows	COLANZ	IND	TRIG	SIZEKB
RENTAL	16042	7	5	1	0,875
PAYMENT	15980	7	4	1	0,8125
FILM	1000	13	5	3	0,3125
FILM_TEXT	1000	3	2	0	0,25
FILM_ACTOR	5455	3	2	0	0,1875
INVENTORY	4591	4	3	1	0,1875
CUSTOMER	599	9	4	1	0,125
ADDRESS	603	6	2	1	0,125
STORE	2	4	3	1	0,0625
STAFF	2	11	4	1	0,0625
LANGUAGE	6	3	1	1	0,0625
FILM_CATEGORY	1000	3	2	0	0,0625
COUNTRY	109	3	1	1	0,0625
CITY	600	4	2	1	0,0625
ACTOR	200	4	2	1	0,0625
CATEGORY	16	3	1	1	0,0625

Abbildung 7: Auswertung mit Text Reports

Variablen gefragt. Hier gibt man das Schema, in welchem sich die Tabellen befinden, an (hier test). Die Auswertung kann mit der Stecknadel oben links (siehe Abb. 7) geöffnet gehalten werden. Danach öffnet man die selbst erstellte Auswertung unter User Defined Reports -> AnalyseDB und stellt die Fenster zur besseren Betrachtung gegenüber:

Abbildung 7 zeigt, dass im linken Report alle im Schema migrierten Objekte dargestellt werden. Im rechten Report wird für jede Tabelle die Anzahl der Datensätze, Spalten, Indices und Trigger sowie der verbrauchte Speicherplatz angezeigt. Damit hat man sich einen guten Überblick über das Schema verschafft, nicht aber über die Daten selbst. Um zu sehen, wie viele Ausleihen pro Monat stattgefunden haben, eignet sich die Chart-Analyse des SQL Developers. Dazu klickt man unter Reports -> User Defined Reports auf Add Report. Um einen Chart-Report zu erstellen, muss das SQL-Statement folgendes Format haben:

```
SELECT
<group>,<series>,<value>
FROM <table(s)>.
```

Für jede Group wird ein eigenes Chart erstellt. Innerhalb des Charts werden die Werte dargestellt und mit dem unter Series angegebenen Namen beschriftet. Da nur ein Chart pro Jahr erstellt werden soll, wird für Group statisch eine 1 ausgewählt und dieses Jahr als Bind-Variable übergeben. In Series werden Jahr, Monat und in Value die

Anzahl der Ausleihen übergeben. Um die benötigten Daten zu selektieren, wird als SQL-Statement im Create-Report-Dialog Folgendes angegeben:

```
select 1, to_char(rental_
date, ,yyyy mm') monat,
count(*) anzahl from rental
where to_char(rental_date,
,yyyy') = :0
group by to_char(rental_
date, ,yyyy mm')
order by monat
```

Oben im Dialog werden als Name, Beschreibung und Tooltip 'Ausleihen pro Monat' angegeben. Als Style wird Chart ausgewählt. Im unteren Teil des Dialogs wechselt man auf den Bind-Tab und betätigt den Add-Bind-Button mit den Werten: Name=0, Prompt='Bitte das Auswertejahr eingeben' sowie Default=2005. Jetzt kann das Aussehen des Charts noch konfiguriert werden, dazu wechselt man auf den Chart Details Tab. Dort gibt man als Chart Type Pie_Muli für das Tortendiagramm an und setzt 3D Graph und Show Legend auf true. Die anderen Einstellungen kann man bei den Default-Werten belassen. Mit Apply werden die Ein-

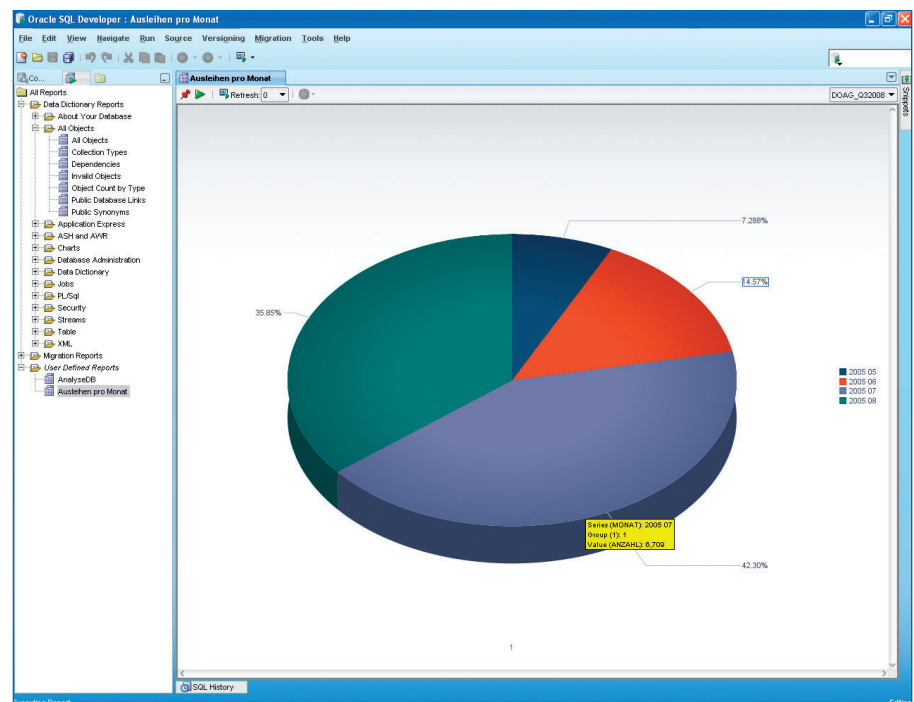


Abbildung 8: SQL Developer Chart-Report

stellungen gespeichert und es wird zur Ausführung des Reports nach der Bind-Variablen gefragt. Man kann die Default Vorgabe 2005 als Jahr übernehmen und den Report erstellen (siehe Abbildung 8).

Wenn man im Bild mit der Maus auf ein Tortenstück zeigt, erscheint ein Tooltip mit dem Zeitraum und dem Wert des Datensatzes. Eine Neuheit im SQL Developer 1.5 ist die Funktionalität „Add reports with gauges“. Es werden zwei Typen unterschieden: Dial (wie eine analoge Tachoanzeige im Auto) und Statusmeter, eine Balkenanzeige. Man muss aber keinen Report erstellen, um eine grafische Auswertung zu erhalten. Es ist auch möglich, im SQL Worksheet eine grafische Auswertung zu erhalten. Die Syntax für eine Gauge-Auswertung ist:

```
SQLDEV:GAUGE:<min>:<max>:
<low>:<high>:' || <value-
column>.
```

Min und Max legt den Wertebereich des Balkens fest. Mit Low und High lassen sich die Farbabschnitte des Hintergrunds einstellen. Bei der Sakila Database wird mit folgendem SQL

```
select cat.name, ,SQLDEV:GAUGE:0:80:40:60:' || count(*) an-
zahl
from film_category fc, category cat
where fc.category_id = cat.category_id
group by cat.name order by count(*) desc;
```

Fazit

Der SQL Developer ist ein kompaktes Tool, das mit vielen Funktionalitäten bei der Entwicklung von SQL- und PL/SQL-Statements hilfreich ist. Neben dieser Kernfunktionalität bietet der SQL Developer eine Reihe weiterer Funktionen von der CVS/Subversion-Anbindung über Schema-Kopieren bis zur Migration und dem Reporting. Auch wenn in diesem Artikel bei der Migration kleinere Nacharbeiten notwendig gewesen sind, ist man trotzdem deutlich schneller fertig, als wenn

ausgewertet, wie viele Filme in den Kategorien sind. Dieses SQL Statement kann im normalen SQL-Worksheet ausgeführt werden:

man ohne Tool-Unterstützung gearbeitet hätte. Die Reporting-Funktionalität bietet mit Standard Reports und User Defined Reports gute Möglichkeiten für textbasierte und grafische Auswertungen. Alle komplexeren Aufgaben von der Migration bis zum Export werden von dem SQL Developer durch unterstützende Wizards begleitet und erleichtern damit die Bedienung.

Kontakt:
 André Schulte
 contact@andschulte.de

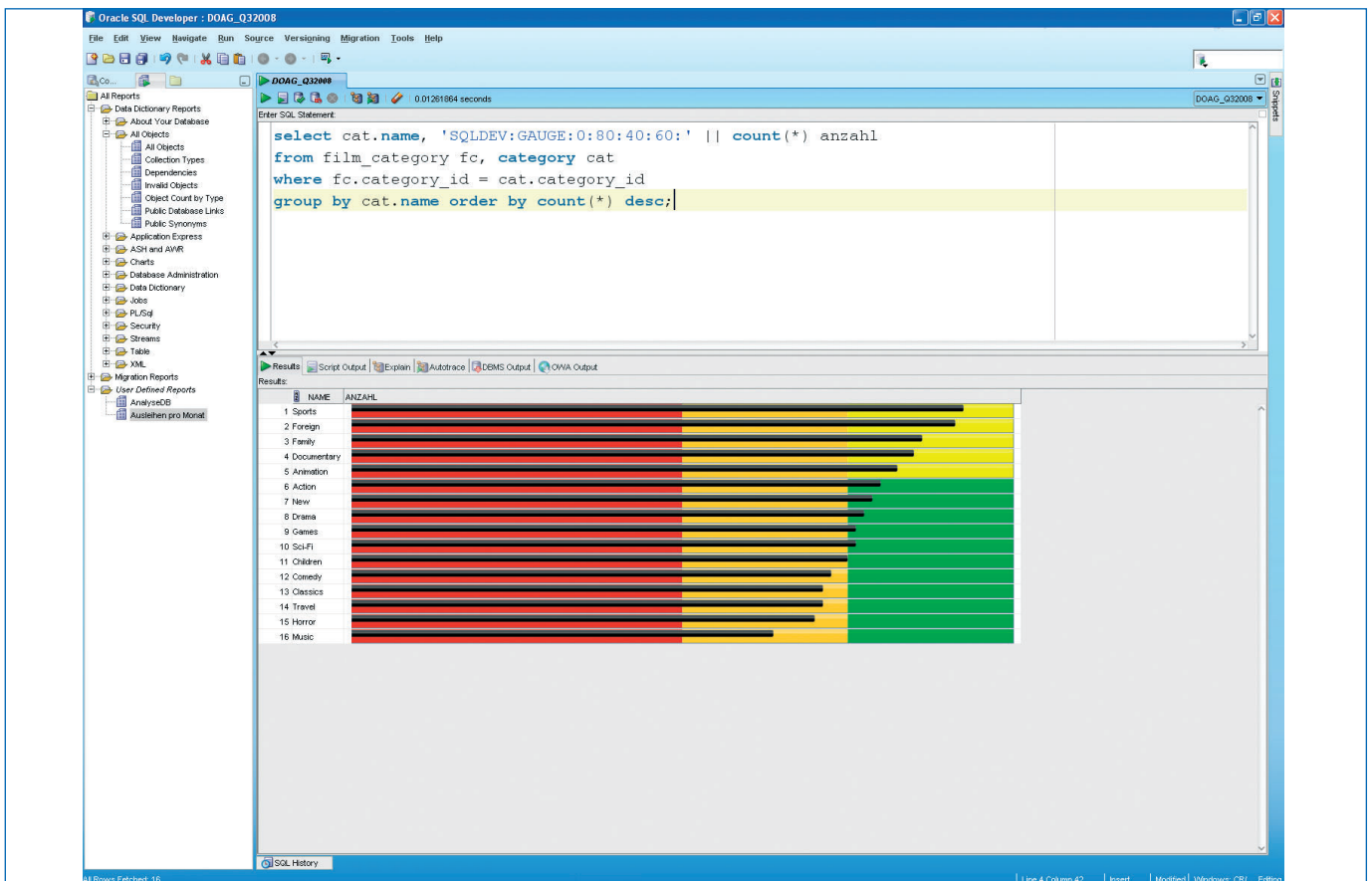


Abbildung 9: SQL Developer Gauges-Report