

SQLdetective - Ein günstiges Tool unter die Lupe genommen

Einleitung

Das Standardtool, um auf Oracle zuzugreifen ist SQL*Plus. SQL*Plus hat zwar den Vorteil, dass es auf allen Systemen läuft, auf denen auch Oracle installiert werden kann. Es hat aber dadurch den Nachteil, dass es nicht sehr benutzerfreundlich ist. Daher gibt es für Windows diverse grafische Tools, um auf Oracle zuzugreifen. SQLdetective kann für die Oracle Versionen 7.3 bis 10.1 verwendet werden und läuft auf einem Windows Betriebssystem (XP, 2000, NT;ME,9x). SQLdetective ist ein Tool, das bewusst sehr günstig angeboten wird. Doch wie gut und wie zuverlässig ist so ein Tool, das nur wenig kostet?

Dieser Artikel zeigt die Möglichkeiten aber auch ein paar Grenzen des Tools SQLdetective, das vor etwa zwei Monaten die Version 3.0 und ganz neu die Version 3.1 auf den Markt gebracht hat.

Einfache Abfragen mit SQL Kenntnissen

Der „Object Navigator“

Nach Start von SQLdetective und Anmeldung an der Datenbank ist der „Object Navigator“ sichtbar. Wie bei grafischen Datenbankwerkzeugen üblich, können so alle sichtbaren Objekte der Datenbank in einem Objektbaum durchstöbert werden. Der wohl interessanteste Ast im Objektbaum ist „My Schema“, das alle Objekte des angemeldeten Schemas zeigt. Die wichtigsten Objekttypen wie Tabellen und Views sind in der Objektliste zuerst aufgeführt, was die Arbeit klar erleichtert. Im Fenster rechts vom Objektbaum sind die Details sichtbar. Wählt man eine der Tabellen aus, so sieht man rechts die Daten der Tabelle.

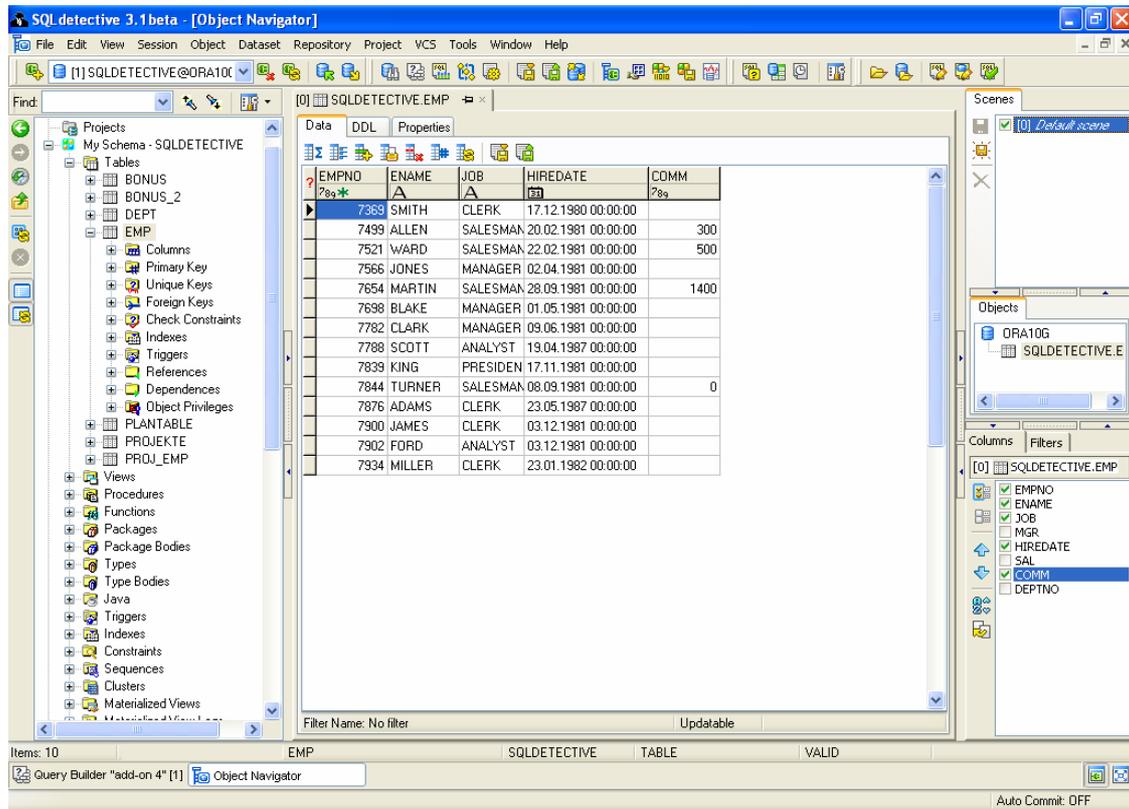


Abbildung 1: „Object Navigator“ mit der Tabelle emp.

Nun stellt sich die Frage, wie langsam das Tool bei grossen Tabellen wird. Da nur die ersten Records gelesen werden müssen, sind auch grosse Tabellen schnell sichtbar. Erst beim Scrollen kommt das Warten, da SQLdetective dann alle Daten lesen muss. Welche Attribute und Zeilen einer Tabelle wirklich dargestellt werden sollen, kann unten rechts im „Column Selector“ festgelegt werden. Im Tab Columns können die darzustellenden Attribute und ihre Reihenfolge ausgewählt werden. Im Filter kann relativ einfach eine Einschränkung definiert werden. So können einfache Sichten auf eine Tabelle zusammengestellt werden. Damit diese Sicht nicht verloren geht, kann sie jederzeit als Szene („scene“) abgespeichert werden. Die gespeicherten Szenen können jederzeit im Fenster oben rechts abgerufen werden. Dies ist vor allem interessant, wenn ich immer dieselben Daten wieder analysieren möchte. So wähle ich nacheinander die relevanten Tabellen aus beliebigen Schemas aus, definiere mit dem „Column Selector“ was ich genau sehen möchte und fixiere die Tabelle dann mit dem Pin. Die so aufgebaute Szene mit mehreren Tabellen speichere ich dann und kann sie immer wieder hervorholen. Dies ist für Tests und Datenanalyse recht hilfreich.

Wieder auf oberster Ebene im „Object Navigator“ kann ich auch weitere Objekte der Datenbank anschauen. Bei einigen Punkten kommt jedoch die Fehlermeldung, dass man DBA sein muss um diese Information zu sehen. Das ist nicht ganz schön, aber korrekt und daher kein Problem.

Der SQL-Editor

Oft will ich aber Daten aus mehr als einer Tabelle sehen, also einfach eine normale SQL-Abfrage absetzen. Dazu dient der „SQL Editor“. Im unteren Fenster wird das SQL-Statement oder ein ganzes Skript geschrieben. Dann kann ein einzelnes Statement oder das ganze Skript gestartet werden. Das Resultat der Abfrage ist im oberen Fenster sichtbar. Im oberen Fenster kann man über den Tab angeben, in welcher Form der Output dargestellt werden soll. So kann man die Daten statt als einfache Texttabelle auch als grafische Tabelle sehen. Ist der „Smart Output“ angewählt, so wird sichergestellt, dass automatisch ein passender Output gewählt wird. Bei einem normalen Select wählt der „Smart Output“ den Tab „Data“. Wird ein ganzes Skript abgearbeitet, so erscheint unter diesem Tab nur das Resultat des letzten Statements.

Sollen langsame Statements optimiert werden, so interessiert der Execution Plan. Dieser wird mit der Ikone „Explain Plan“ gestartet und im entsprechenden Tab dann dargestellt. Die Darstellung des Execution Plans ist sehr übersichtlich und hilfreich. Für die Optimierung empfehle ich den „Smart Output“ auszuschalten. Im normalen Output wird auch die Laufzeit angegeben, was für die Optimierung wichtig ist.

Hilfe für Anfänger dank Query Builder

Wird der Query Builder gestartet, erscheint in der Mitte ein weisses Fenster. Nun können Tabellen oder andere Datenbankobjekte für die Query ausgewählt werden.

Sinnvollerweise wird die Option „autolink“ aktiviert. Dann werden die bekannten Relationen zwischen den Tabellen als Links direkt dargestellt. Sind keine Links vorhanden, können diese selber mit der Maus gezogen werden.

Nun kann ausgewählt werden, welche Felder dargestellt werden sollen. Damit ist die Query fertig und kann in den „Test SQL“ übernommen werden. Ähnlich wie im „SQL Editor“ kann die Query ausgeführt werden. Natürlich kann hier die Query auch noch manuell weiter bearbeitet werden.

Leider wird mit dem Link immer ein Outer Join erstellt, sobald eines der Join Attribute leer sein kann. Will man einen Inner Join, so kann dies in der aktuellen Maske unter Property angepasst werden. Somit ist der Query Builder ein Werkzeug, das noch mit Vorsicht zu geniessen ist.

In einem der nächsten releases, also voraussichtlich in ein paar Monaten, sollte der Query Builder für Joins dann auch die ANSI Syntax unterstützen. Das wird den Query Builder klar aufwerten.

Manipulieren von Daten

Wird im „Object Navigator“ eine Tabelle angewählt, so werden die Daten der Tabelle als „Dataset“ dargestellt. In diesem „Dataset“ können nun die Daten bearbeitet werden. Über das Menü oder über die Icons können Datensätze gelöscht oder eingefügt werden. Der ausgewählte Datensatz kann auch geändert werden.

Alle vorgenommenen Änderungen sind im „Dataset“ sofort sichtbar. Wie in Oracle üblich sind die gemachten Änderungen aber nur für die aktuelle Session, sichtbar. Erst wenn die Änderungen mit Commit bestätigt wurden, sind sie für alle sichtbar.

Werden die Änderungen mit Rollback zurückgenommen, so wird das „Dataset“ absichtlich noch im alten Zustand belassen. Bei einem Rollback ist es daher sinnvoll, dass mit „Refresh Data“ die Daten des Datensets aktualisiert werden.

Erstellen von Objekten

Tabellen erstellen

Am einfachsten wählt man im „Object Navigator“ im eigenen Schema den Ast „Tables“ an und drückt die rechte Maustaste. Dann kann „Create“ angewählt werden. Die andere Möglichkeit ist über das Menü „Create“ zu wählen, und dann „Tables“ zu selektieren. Nun erscheint der „New Objects Wizzard“. Hier können nun der Reihe nach alle Angaben zu einer neuen Tabelle eingegeben werden. Wenn alle Angaben vorhanden sind, kann das generierte SQL angeschaut werden oder die Tabelle direkt mit „Apply“ angelegt werden.

Beim Erstellen einer Tabelle werden diverse Oracle-Möglichkeiten unterstützt. So kann recht einfach eine partitionierte Tabelle mit allen Partitionen erstellt werden. Wird nach einem Datumsfeld partitioniert, so kann die Partitionsgrenze einfach als Datum eingetippt werden. Die Umwandlung mit „TO_DATE“ wird von SQLdetective im Code ergänzt. Auch Constraints können direkt mit einem Wizzard erstellt werden. Auch hier ist die Unterstützung gut, so kann beispielsweise bei der referenziellen Integrität der referenzierte Constraint ausgewählt werden.

Views erstellen

Zum Erstellen einer View ist der Wizzard auf den ersten Blick nicht sehr hilfreich, da das Select der View direkt eingetippt werden muss. Es gibt jedoch die Möglichkeit direkt in den „Query Builder“ zu wechseln, dort die Query der View zusammen zu stellen und dann wieder in den „View Wizzard“ zurück zu wechseln. Diese Kombination der Wizzards ist recht bequem. Natürlich kann das Select auch im SQL-Editor erstellt werden und dann in den „View Wizzard“ kopiert werden.

Prozeduren, Funktionen und Packages erstellen

Wird eine Prozedur, Funktion oder ein Package erstellt, so wird direkt ein Codegerüst inklusive Exceptionhandler zur Verfügung gestellt. So ist schon mal die erste Fehlerquelle, die der Syntaxfehler, reduziert. Das grosse Problem bei Prozeduren, Funktionen und Packages ist normalerweise nicht nur die Syntax, sonder das Debuggen, das Finden der Laufzeitfehler.

Hier ist der Debugger von SQLdetective eine grosse Hilfe. Damit der Code aber überhaupt geprüft werden kann, muss er vorher entsprechend kompiliert worden sein. Dazu wählt man im Menü unter <Object> den Punkt <Compile> und dort <Compile Debug>. Als Beispiel, wie der Debugger funktioniert, wird hier mit einer einfachen Funktion gearbeitet, die zwei Zahlen addiert. Damit einfach ein Fehler provoziert werden kann, darf das Resultat nur eine Stelle haben, muss also kleiner 10 sein.

```

CREATE OR REPLACE FUNCTION SQLDETECTIVE.TEST_FUNCTION (a IN NUMBER, b
IN NUMBER)
RETURN NUMBER
--{[AUTHID { CURRENT_USER | DEFINER }] | DETERMINISTIC |
PARALLEL_ENABLE [( PARTITION ARGUMENT BY { ANY | { HASH | RANGE } (
COLUMN [, COLUMN]... ) )]}]}
AS --{AS | IS}
  --[variable datatype;]
  res NUMBER(1);
BEGIN
  res := a;
  res := res + b;
  RETURN res;
EXCEPTION
  WHEN OTHERS THEN
    NULL; --pl/sql;
END TEST_FUNCTION;

```

Beispiel einer einfachen Funktion

Nun kann mit der rechten Maustaste <Debug> gestartet werden. Die Inputparameter können angegeben werden und SQLdetective zeigt die Funktion im Stored Program Editor. Wie bei jedem Debugger können Breaks gesetzt werden oder man kann zeilenweise durch den Code gehen. Was nun interessiert, sind die Werte der einzelnen Variablen. Diese werden dargestellt, sobald man die Maus auf die Variabel im Code bewegt. Das ist sehr bequem.

Die vorliegende Funktion wurde mit den Werten 7 und 8 gestartet. Das Resultat 15 ist klar zu gross, um in der Variable res gespeichert zu werden. Als Resultat wird also die Definition von res auf NUMBER(10) geändert und die Funktion nochmals gestartet. Korrekterweise frag SQLdetective nach, ob ich den Code kompilieren möchte, damit die gemachte Änderung auch berücksichtigt wird.

Recycle Bin

Das Konzept vom Recycle Bin wird von SQLdetective unterstützt. Wird eine Tabelle mit DROP gelöscht, so kann angegeben werden, ob diese mit PURGE definitiv gelöscht werden soll. Default ist ohne PURGE und die Tabelle kommt in den Recycle Bin. Als normale Tabelle ist sie dann nicht mehr sichtbar, dafür im Recycle Bin. Eine Tabelle kann aus dem Recycle Bin mit Flashback wieder zurück geholt werden.

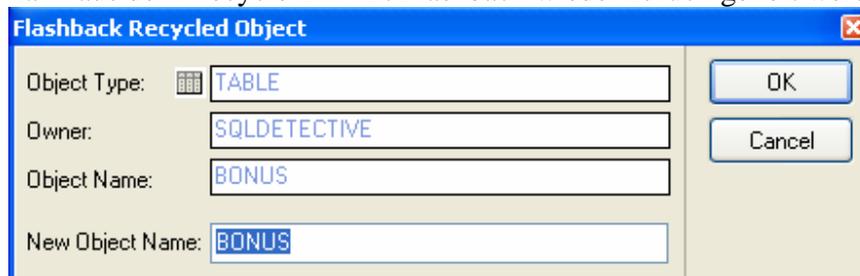


Tabelle aus dem Recycle Bin zurückholen

Zusammenfassung

SQLdetective bietet erstaunlich viel und ist im Grossen und Ganzen in der Bedienung recht intuitiv. Wie bei jedem grafischen Werkzeug zu Oracle, ist Grundwissen über Oracle weiterhin unumgänglich. Die neuste Version 3.1 bietet mit dem Storage Manager auch hilfreiche Infos für einen DBA. So kann SQLdetective sowohl für Entwickler als auch für DBA's nützlich sein, auch wenn gewisse Details nicht ganz perfekt sind. Dass regelmässig neue Versionen mit Erweiterungen herausgegeben werden und auch neuere Features von Oracle unterstützt werden ist ein weiterer Pluspunkt. Als Betatesterin war ich auch erstaunt, wie schnell Verbesserungsvorschläge umgesetzt wurden. SQLdetective ist sicher ein Beweis dafür, dass Qualität wohl einen Preis hat, aber nicht teuer sein muss.

Eine gratis Testlizenz, welche auf 15 Tage nach Installation limitiert ist, kann unter www.sqldetective.com herunter geladen werden.

Andrea Kennel, InfoPunkt Kennel GmbH