



## Select \* from anywhere

Alligator Software präsentiert neues Tool zur Migration von Daten zwischen unterschiedlichen Datenbanken während der Produktion.

10\_001\_03

Select \* from anywhere

Mit AlligatorDBC präsentiert Alligator Company Software GmbH jetzt ein Tool zur direkten Interaktion zwischen unterschiedlichen Datenbanksystemen. Innerhalb einer relationalen Datenbank ermöglicht es den unmittelbaren Zugriff auf Daten einer anderen relationalen Datenbank. In der Praxis können damit z.B. Daten aus einer MySQL Tabelle direkt in Oracle gelesen oder auch manipuliert werden. So ergeben sich gerade bei der Umstellung einer relationalen Datenbasis mit AlligatorDBC entscheidende Vorteile: Die Migration der Daten geschieht permanent, so dass sich der bisher unvermeidliche Produktionsstopp entscheidend verkürzt oder sogar ganz erübrigt. Und das spart nicht nur Zeit und damit Nerven, sondern hilft auch, finanzielle Ressourcen zu schonen.

### Analyse der gängigen Praxis als Ausgangspunkt der Entwicklung

Die herkömmliche Vorgehensweise zur Migration in einem Produktionssystem mit mehreren relationalen Datenbanken erfordert dagegen einen deutlich höheren Aufwand. Etwa, wenn Daten aus einer MySQL Datenbank in eine Oracle Datenbank als Zielsystem übernommen werden sollen. Anders als bei gleichgearteten Datenbanksystemen gibt es hier keine Möglichkeit, über so genannte Links eine direkte Verknüpfung aufzubauen. Man kann sich also nur behelfen, indem man die zu migrierenden Daten aus dem Altsystem in Dateien ausgelagert, um diese in einem zweiten Schritt in das Zielsystem einzulagern. Auch ein Programm – welche Programmiersprache auch immer – müsste die Daten auslesen und diese in das Zielsystem schreiben. Und genau hier kann es bei diesem Verfahren tückisch werden: Die Ursprungsdaten dürfen sich nach der Auslagerung nicht mehr ändern. Das Produktionssystem muss also vorher gestoppt werden. Die Migration beginnt und damit oft auch der Wettlauf gegen die Zeit. Denn jetzt kann das Migrationsteam eigentlich nur noch hoffen, dass die Prozeduren laufen und der Ausfallplan ausreicht, um eventuelle Fehler während der Migrationsphase aufzufangen.

### AlligatorDBC für deutlich größere Unabhängigkeit und Flexibilität

Mit AlligatorDBC lautet der Migrationsbefehl für dieselbe Prozedur:

#### **CREATE scott.oracletab AS SELECT \* FROM prodmysqltab**

Dieses SQL Statement legt eine Tabelle mit Namen "scott.oracletab" mit der Struktur der Tabelle prod.mysqltab aus der MySQL Datenbank an. Der Inhalt der Tabelle wird mit kopiert. Fertig. Die Übernahme der "Altdaten" behindert nicht die Produktion und ist daher nicht weiterer Bestandteil der Migration. Die gewonnene Zeit kann für Ausfallprozeduren oder Population der Daten verwendet werden. Voraussetzung für die permanente Datenübernahme ist die Möglichkeit, auf dem Altsystem "Trigger" anzulegen. Diese Trigger sorgen dafür, dass bei jeder Manipulation der Altdaten die neuen Tabellen auf dem Zielsystem abgeglichen werden.

#### **Maximaler Komfort in Verbindung mit AlligatorSQL**

Alligator Software bietet AlligatorDBC nicht allein als separates Tool an. Seine Stärken wird es künftig innerhalb der einfachen Navigationsstrukturen der AlligatorSQL Enterprise Edition entfalten. Schon in Basisversion AlligatorSQL 1.15 lassen sich über einen direkten Zugriff bis zu sechs relationale Datenbanken mit einer einzigen Oberfläche übersichtlich verwalten. Spezielle Features wie PL/SQL-Debugger und PL/SQL-Profiler für Oracle Datenbanken sind im Alligator SQL 1.15 bereits implementiert.

Für eine 30 Tage Trial-Version der AlligatorSQL Enterprise Edition inklusive AlligatorDBC klicken Sie bitte hier:

<http://www.alligatorsql.com/download/alligator.zip>

Alligator Company Software GmbH bietet auch die Möglichkeit, AlligatorDBC separat zu testen. Für den kostenlosen Download einschließlich einer ausführlichen technischen Anleitung zum Gebrauch klicken Sie bitte hier:

<http://www.alligatorsql.com/download/adbc.zip>

## Die Verwendung von AlligatorDBC als eigenständiges Tool

Mit AlligatorDBC haben Sie innerhalb einer relationalen Datenbank den direkten Zugriff auf Daten einer anderen relationalen Datenbank. Damit können in der Praxis z.B. Daten aus einer MySQL Tabelle direkt in Oracle gelesen oder auch manipuliert werden. Im Folgenden erhalten Sie dazu eine technische Beschreibung der Vorgehensweise. Diese Beschreibung ist für Sie jedoch nur dann relevant, falls Sie AlligatorDBC separat verwenden möchten. Wenn Sie AlligatorDBC im Rahmen der AlligatorSQL Enterprise Edition verwenden, dann übernimmt die Enterprise Edition alle folgende Schritte für Sie.

### Die Technik

Um nun Tabellen aus einer fremden relationalen Datenbank zu selektieren oder zu manipulieren, muß die jeweilige Datenbank (in unserem Beispiel Oracle) um einige Funktionen erweitert werden. Dazu bedienen wir uns der Programmiersprache „C“. Wir erstellen eine Kernelroutine, um auf eine MySQL Datenbank zuzugreifen. An dieser Stelle wollen wir nicht näher auf den Sourcecode eingehen. Laden Sie sich bitte die komplette Kernelroutine MySQL unter der Adresse [www.alligatorsql.com](http://www.alligatorsql.com) und erstellen Sie eine dynamische oder shared Bibliothek. Für die Betriebssystem Windows 98/NT/2000/XP und Linux sind bereits Bibliotheken vorhanden. Falls Ihre Oracle Datenbank unter UNIX einem Derivat oder einem anderen Betriebssystem läuft, benutzen Sie bitte das mitgelieferte Makefile um die entsprechende ADBC Bibliothek zu generieren. Die so erstellte DLL oder SO Datei muß nun in ein für die Oracle Datenbank bekanntes Verzeichnis kopiert werden. Angenommen Ihr Oracle Homeverzeichnis ist D:\ORACLE\ORA81, dann kopieren Sie bitte die Bibliothek nach D:\ORACLE\ORA81\EXTLIB. Nun müssen wir der Datenbank mitteilen, daß Ihre Funktionalität erweitert worden ist. Dazu benötigen wir ein „Libraryobjekt“. Mit dem DDL Befehl

```
CREATE OR REPLACE LIBRARY SYS.MYDLL 'd:\oracle\ora81\extlib\mydll.dll'
```

erstellen Sie eine Verbindung zu Ihrer DLL. Analog verfahren Sie mit einer shared library. Als letzten Schritt installieren Sie bitte das Oraclepackage ALLIGATOR\_DBC. ADBC steht für Alligator Database Connectivity. In diesem Package sind alle Funktionsaufrufe enthalten, die sie für den Zugriff auf eine MySQL Datenbank benötigen. Das Packagescript (sys\_adbc.sql) befindet sich in der ZIP Datei. Führen Sie dieses Script bitte als Benutzer „SYS“ in SQLPLUS aus. Da speziell für Oracle in speziellen Parameterdateien der externe Zugriff auf Prozeduren geschaltet werden muß, kontrollieren Sie bitte Ihre Dateien TNSNAMES.ORA und LISTENER.ORA und vergleichen Sie diese mit unseren Beispieldateien aus der mysql.zip Datei. So, das wäre es. Testen Sie nun, ob Ihr ALLIGATOR\_DBC Paket installiert und bereit ist. Starten Sie dazu eine Prozedur mit Namen ADBC.ISACTIVE. Diese Prozedur liefert den Wert TRUE zurück, wenn das Paket „startklar“ ist.

### „Joinen“ von Tabellen aus verschiedenen relationalen Datenbanken

Um nun eine MySQL Tabellen direkt mit einer Oracltabelen zu vereinen, bedarf es einiger Hilfsobjekte, die manuell oder aber direkt mit AlligatorSQL angelegt werden können. Zunächst benötigt man einen eigenen Oracltypen, der genau die Tabellenstruktur der MySQL Tabelle widerspiegelt. Stellen wir uns vor, wir wollen eine Tabelle mit Namen „MyTestTab“ und der Struktur „Vorname VARCHAR(30)“ und „Name VARCHAR(30)“ mit einer Oracltabelle verknüpfen. Die Tabelle „MyTestTab“ liegt in der Datenbank „test“ auf einem MySQL Server. (Siehe scott\_adbc.sql)

Legen Sie nun einen Oracletypen und die „Nested Table“ mit genau derselben Struktur an, wie die beschriebene „MyTestTab“ Tabelle. (Siehe scott\_adbc.sql)

Wir benötigen ein Oracle Paket, welches die Verbindung zwischen dem Select und ALLIGATOR\_DBC herstellt. An dieser Stelle möchte ich nicht auf die einzelnen Schritte eingehen. Schauen Sie sich bitte das Script ADBC\_MYTESTTAB an und starten Sie das Script. Der vorher angelegte Objekttyp und das Paket sollten sich im gleichen Schema befinden. Wir benutzen für unser Beispiel das allseits beliebte SCOTT Schema. Die eigentliche Aufgabe des Paketes besteht darin, die Daten aus der MySQL Tabelle zu lesen und als Typen zurück zugeben.

Als letzten Schritte legen wir eine View über die im Package befindliche Funktion „SelectData“ an (Siehe „MyTestTab\_View“). Mit dieser View kann nun wie mit jeder anderen View oder Tabelle gearbeitet werden. Die einzige Einschränkung sind Manipulationen wie INSERT, UPDATE oder DELETE. Diese können aber über die direkte Funktion aus dem ALLIGATOR\_DBC Paket durchgeführt werden.

Damit diese Schritte nicht immer wieder manuell durchgeführt werden müssen, enthält das Produkt AlligatorSQL einen entsprechenden Wizard um alle beschriebenen Objekte automatisch anzulegen. Damit wird das Veröffentlichen von MySQL Tabellen zu Oracle zum Kinderspiel. (Siehe AlligatorSQL Hilfe ADBC)

Die volle Unterstützung aller Datenbanken wird ab der Version 1.2 enthalten sein.

Windows ist ein eingetragenes Warenzeichen der Firma Microsoft.  
Oracle ist ein eingetragenes Warenzeichen der Firma Oracle.  
AlligatorSQL, AlligatorDBC und ADBC sind eingetragene Warenzeichen der Firma Alligator Company Software GmbH.