

# Oracle GoldenGate

## Die Replikation beginnt mit Initial-Load!

Joachim Jaensch  
ORACLE Deutschland B. V. & Co. KG  
München

### Schlüsselworte:

Oracle, GoldenGate, Replikation, Initial-Load, Erstbefüllung, Datenbank, Tabellen, Quelle, Ziel, Extract, Replicat, Change-Synchronisation, HANDLECOLLISIONS

### Einleitung

Plant man eine Replikation von einer Datenbank zur anderen, so muß man zuerst dafür sorgen, daß der Inhalt beider Datenbanken stimmig ist. Dabei müssen die Objekte der Quell- und Zieldatenbanken aber nicht identisch sein. Der Nutzer muß in Abhängigkeit der Gegebenheiten selbst festlegen, wie Inhalt und Struktur der beteiligten Tabellen aussehen müssen. In fast allen Situationen ist dabei der Aufbau und das Erstbefüllen von Tabellen auf der Zieldatenbank notwendig. Diesen Vorgang bezeichnet man auch als Initial-Load oder Erstbefüllung. Es gibt bei GoldenGate verschiedene Möglichkeiten für den Initial-Load. Der Vortrag gibt einen Überblick darüber, mit welchen Methoden die Erstbefüllung der beteiligten Datenbankobjekte erfolgen kann. Eigentlich wäre der Initial-Load ein recht einfacher Vorgang, wenn man dafür ein Zeitfenster hätte, indem die Quelldatenbank in einen „read-only“ Zustand versetzt werden könnte. Das bleibt aber in der Praxis meist ein nicht erfüllbarer Wunsch. In der Realität laufen ständig Transaktionen, die die Quelldatenbank verändern. Diese Änderungen müssen durch einen Change-Extract erfaßt und nach der Erstbefüllung auf der Zieldatenbank ausgeführt werden. Bei der Betrachtung von Initial-Load hat man es deshalb auf der Quelldatenbank mit Initial-Extract und Change-Extract, auf der Zieldatenbank mit Initial-Replicat und Change-Replicat zu tun. Die Erstbefüllung ist deshalb ein komplexer Vorgang, der in fest definierten Schritten erfolgen muß. Prinzipell ist der Initial-Load für homogene und heterogene Umgebungen gleich. Aus Gründen der Übersichtlichkeit werden im Vortrag alle Replikationsmethoden am Beispiel zweier Oracle Datenbanken gezeigt.

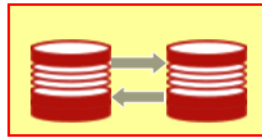
### Replikationsszenarien

Replikation erfolgt immer zwischen einer Quell- und einer Zieldatenbank. Gibt es nur eine Richtung dann spricht man von uni-direktionaler Replikation. Werden Änderungen auch in die entgegengesetzte Richtung repliziert wird das als bi-direktionale Replikation bezeichnet. In diesem Fall sind beide Datenbanken gleichzeitig Quelle und Ziel von Änderungen. In der Praxis bestehen Replikationsumgebungen häufig aus mehr als nur zwei Datenbanken. Die Art der Replikation und die Anzahl der beteiligten Datenbanken führen zu einer Vielzahl von Replikationsszenarien, die bei GoldenGate wie in Abbildung 1 bezeichnet sind:

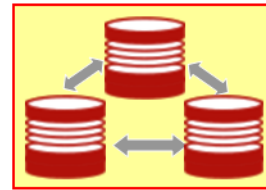
**Unidirectional**  
Reporting Instance



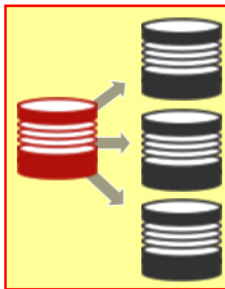
**Bi-directional**  
Instant Failover, "Active"



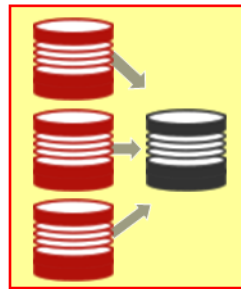
**Peer-to-Peer**  
Load Balancing, HA, DR



**Broadcast**  
Data Distribution



**Consolidation**  
Data Warehouse



**Cascading**  
Data Marts



*Abbildung 1: Replikationsszenarien bei Oracle GoldenGate*

Vergleicht man die GoldenGate Begriffe mit denen anderer Replikationslösungen, dann stellt man fest, daß es unterschiedliche Bezeichnungen für ein und die selbe Replikation gibt. So entspricht die uni-direktionale Replikation einer „Master-Slave“, die bi-direktionale und die Peer-To-Peer Replikation der „Master-Master“ Replikation. Broadcast und Consolidation in Kombination werden bei bei Oracle Streams als Hub & Spoke Replikation bezeichnet.

**Wie funktioniert eine Replikation?**

Eine Replikation ist immer dann notwendig, wenn lokale Daten auch an anderer Stelle benötigt werden. Die Replikation ist das Mittel um Daten von A nach B oder auch in beide Richtungen zu übertragen. Am Markt existieren dafür sehr viele Lösungen unterschiedlicher Software-Hersteller. Dieser Beitrag beschäftigt sich mit der Datenreplikation durch Oracle GoldenGate, im folgenden GoldenGate genannt. GoldenGate ist eine heterogene Replikationslösung, die für sehr viele Datenbanken bzw. Datenhaltungssysteme verfügbar ist. Prinzipell funktionieren homogene und heterogene Replikationen nach dem gleichen Prinzip. Die Eigenheiten unterschiedlicher Datenbanken und die Behandlung von abweichenden Datenformaten in heterogenen Umgebungen werden im Vortrag nicht angesprochen. Am Beginn einer Replikation steht die Planung. Es ergeben sich Fragen wie: Welche Daten sollen repliziert werden? In welchen Tabellen befinden sich diese Daten? Muß zu mehreren Datenbanken repliziert werden? Müssen Daten bi-direktional ausgetauscht werde, oder reicht eine Richtung? Entsprechen die Quelltabellen den Zieltabellen? Ist es nötig, Daten zu filtern oder zu transformieren? Mit der Beantwortung dieser Fragen ergibt sich ein Replikationsszenarium dessen Komplexität durch die gewünschten Funktionalitäten bestimmt wird. Je umfangreicher eine

Replikationsumgebung ist, desto größer wird der Aufwand für Administration und Monitoring. So müssen z.B. bei bi-direktionale Replikation Lösungen für mögliche Konflikte von Anfang an geplant und über Programmlogik realisiert werden. Sinnvoll erscheint eine stufenweise Implementierung bei der die Komplexität schrittweise zunimmt und die nächste Stufe erst realisiert wird, wenn die Vorstufen fehlerfrei funktionieren.

## Das Einrichten einer Replikation

Wie schon zu Beginn erwähnt, kann eine Produktivdatenbank zum Aufsetzen einer Replikation nicht in den „read-only“ Modus versetzt werden. Das wäre zwar sehr vorteilhaft, weil man Veränderungen des Datenbestandes für den Zeitraum des Initial-Load bzw. der Erstbefüllung ausschließen könnte, würde aber die Verfügbarkeit der Datenbank einschränken. Es ist deshalb notwendig parallel zum Initial-Load anfallende Änderungen in der Quelldatenbank zu erfassen und später auf der Zieldatenbank einzupflegen. In Abbildung 2 wird genau das gezeigt. Zum einen haben wir den Vorgang der Erstbefüllung bei dem Quelldaten ausgelesen und auf der Zielseite in die entsprechenden Tabellen gebracht werden. Parallel dazu werden alle Änderungen in den betroffenen Tabellen erfaßt und für später gespeichert. Dazu noch eine Erklärung der einzelnen Begriffe. Der Initial-Load ist ein zweistufiger Prozeß, der aus einem Initial-Extract auf der Quellseite und einem Initial-Replicat auf der Zielseite besteht. In Abbildung 2 ist dieser Ablauf im oberen Teil dargestellt. Ähnlich verhält es sich mit der im unteren Teil gezeigten Change-Synchronisation. Sie besteht aus einem Change-Extract auf der Quell- und einem Change-Replicat auf der Zielseite. Der Change-Extract muß immer vor dem Initial-Extract gestartet werden. Durch diese Reihenfolge können keine Änderungen verloren gehen. GoldenGate Extract gestattet die Benutzung lokaler Trail-Files oder von Remote-Files. Werden Daten lokal abgelegt, müssen sie später noch über Data-Pump auf die Zielseite übertragen werden. Schreibt der Extract sofort in ein Remote-File spart man sich den zusätzlichen Data-Pump, benötigt aber zwingend eine aktive TCP/IP-Verbindung zwischen Quelle und Ziel. Ergebnis dieses Ablaufes ist eine voll funktionierende Replikation zwischen den beteiligten Datenbanken, weil neben dem Initial-Load auch schon der Change-Extract und der Change-Replicat für die spätere kontinuierliche Übertragung der Änderungen mit eingerichtet wurden.

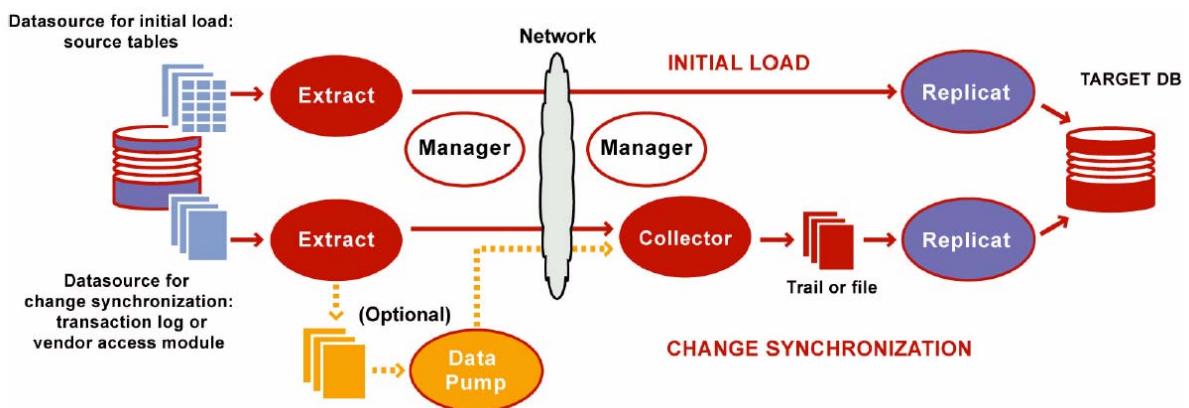


Abbildung 2: GoldenGate Initial-Load mit anschließender Change-Synchronisation

## Initial-Load Methoden bei GoldenGate

GoldenGate ist sehr flexible und bietet dem Anwender eine Reihe unterschiedlicher Möglichkeiten für die Erstbefüllung der Tabellen in der Zieldatenbank:

Lfd. Nummer	Bezeichnung	Bemerkung
1	Database Utility	Oracle: Export / Import
2	Load from File	GoldenGate
3	Database Utility from File	Oracle: SQL*Loader
4	GoldenGate Direct Load	GoldenGate
5	Direct Bulk Load	Oracle: SQL*Loader API

*Abbildung 3: GoldenGate Initial-Load Methoden*

Im Vortrag werden die einzelnen Methoden erläutert und die Vor- und Nachteile angesprochen. Die Problematik der Konsistenz des Initial-Load Vorganges und der durch GoldenGate bereitgestellten Verarbeitungsparameter HANDLECOLLISIONS und NOHANDLECOLLISIONS wird am Beispiel der einzigen konsistenten Initial-Load Methode erklärt.

### Kontaktadresse:

**Joachim Jaensch**  
**ORACLE Deutschland B. V. & Co. KG**  
**Riesstrasse 25**  
**D-80992 München**

**Telefon:** +49 (0) 89-1430 1326  
**Fax:** +49 (0) 89-1430 2160  
**E-Mail:** [joachim.jaensch@oracle.com](mailto:joachim.jaensch@oracle.com)  
**Internet:** [www.oracle.com/de](http://www.oracle.com/de)

Deutsche Oracle Data Integration Community: <http://apex.oracle.com/pls/otn/f?p=43477:1>