

# Mit „Smart Replication“ Daten effizient, robust und flexibel verteilen

Carsten Kaftan, Sanacorp Pharmahandel GmbH

Erfreulicherweise kann der Autor alle Ankündigungen bezüglich Advanced Replication und Streams sowie die Preislisten von GoldenGate mit Gelassenheit studieren – er setzt seit dem Jahr 2008 für Replikationen die Eigenentwicklung „Smart Replication“ ein. Dieser Artikel ist die aktualisierte und überarbeitete Version eines Vortrags auf der DOAG-Jahreskonferenz; er richtet sich an Leser mit Interesse an einer schlichten und allgemein einsetzbaren Lösung für Replikationsaufgaben.

Wegen erheblicher Probleme mit der Flexibilität (Multi-Master-Architektur bei Advanced Replication) und hohen Ressourcen-Bedarfs (CPU-Verbrauch bei Streams) sowie der Tatsache, dass die angepriesenen und beworbenen Eigenschaften und Möglichkeiten zum Teil gar nicht den Bedürfnissen entsprachen, hat das Unternehmen des Autors bereits vor vielen Jahren den Schritt zur Eigenentwicklung „Smart Replication“ unternommen. Herausgekommen ist ein vollständig in PL/SQL geschriebenes Werkzeug, das sich auf die stabile und anpassungsfähige Weitergabe von Datenänderungen konzentriert. Elementare Verfahren erfassen diese und übertragen sie in die gewünschten Ziele. Die üblichen Replikations-Szenarien – sehr wenige Änderungen im großen zeitlichen Abstand, Massendaten-Änderungen, mehrere Quellen kombiniert mit mehreren Senken, Gültigkeitszeitraum-gesteuertes Herausziehen von aktuellen Daten aus Stammtabellen etc. – sind out of the box abgedeckt.

## Prinzip

Eine Smart Replication bezieht sich auf eine Quell-Tabelle beziehungsweise einen Quell-View und gegebenenfalls auf mehrere gleichartige Ziel-Tabellen beziehungsweise Views. Es lassen sich beliebig viele voneinander unabhängige Smart Replications einrichten, die mit einfachen Methoden wie Zwischen- und temporären Tabellen, Triggern und Jobs die Daten aus der Quelle heraus auf die gewünschten

Ziele verteilen – innerhalb einer oder zwischen verschiedenen Datenbanken. *Abbildung 1* zeigt den strukturellen Aufbau einer Smart Replication.

Zunächst müssen die Quelldaten so, wie im Ziel benötigt, bereitgestellt werden; im einfachsten Fall sind Ziel und Quelle Tabellen von gleicher Struktur und mit gleichem Primärschlüssel. In anderen Fällen kann etwa eine passende View auf die Quell-Tabellen eingerichtet werden. Die Konfiguration der Smart Replication legt Quell- und Ziel-Datenobjekte, zugehörige Schlüsselfelder (normalerweise den Primärschlüssel) und Übertragungsparameter fest (Quelle, Ziel und Transfer). Bei Änderung eines Quelldatensatzes wird der zugehörige Schlüssel in die „Schlüsselwerte der Änderungen“-Tabelle eingefügt.

Die Erfassung der Datenänderungen erfolgt am einfachsten über Trigger (Auslöser) auf die Quelltable beziehungsweise auf die Basis-Tabellen zum Quellview; es ist aber auch möglich, die zu übertragenden Schlüssel gezielt in die Smart-Replication-Tabelle einzutragen. Beim Eintrag werden neben dem Schlüssel auch der vorgesehene Übertragungszeitraum und die Priorität festgelegt.

Ein Aufräumjob soll gewährleisten, dass sich keine überflüssigen Einträge ansammeln: So werden bei Bedarf Einträge zusammengefasst, etwa nach Mehrfach-Änderungen eines Datensatzes beziehungsweise Überschneidungen in den vorgesehenen Übertragungsfenstern. Gemäß der jeweiligen Konfiguration werden auch soge-

nannte „Ewigkeits-Einträge“ entfernt, deren Übertragungsfenster sehr weit in der Zukunft liegen. Diese können bei Historisierungen entstehen, falls etwa ein Datensatz mit Gültigkeit bis zum 31.12.9999 („für immer“) geändert wird und die zugehörige Smart Replication so konfiguriert ist, dass sie für „gültig von“- und „gültig bis“-Daten Aktualisierungen vormerkt.

Der Übertragungsjob wird immer bei Änderungen in der „Schlüsselwerte der Änderungen“-Tabelle aktualisiert und plant sich für den Beginn des nächsten Übertragungsfensters sowie danach solange in den vorgegebenen Zeitabständen ein, bis alle anliegenden Änderungen in der konfigurierten Blockgröße abgearbeitet sind. Danach plant sich der Übertragungsjob für den Beginn des nächstfolgenden Übertragungsfensters beziehungsweise, falls keine weiteren Datenänderungen vorliegen, für „nie“ ein. Die von diesem Job aufgerufene Datenübertragung läuft in folgenden Schritten ab:

- Zusammenstellen des nächsten Datensatzblocks nach Priorität und vorgesehendem Übertragungsfenster mit der konfigurierten Anzahl an Einträgen und Übertragen in die Temporary-Table im Ziel
- Löschen der Einträge mit Schlüsseln aus der Temporary-Table aus der Ziel-Tabelle
- Kopieren der Einträge mit Schlüsseln aus der Temporary-Table von der Quell- in die Ziel-Tabelle

### Architektur

Es sind Quell-, Ziel- und gegebenenfalls Administrations-Datenbanken beteiligt, die problemlos auch identisch sein können. Überall muss ein „Admin“-User mit Hilfscode, auf den Quell- und Ziel-Datenbanken außerdem ein „Replic“-User eingerichtet sein. Bei optionaler Verwendung einer Administrations-Datenbank (enthält lediglich ein zusätzliches „Admin“-Package) können zentral zusätzliche nützliche Funktionen zum Einsatz kommen.

Das Verfahren Smart Replication ist in dem Package „DBA\_REPLIC“ codiert. Dort finden sich auch die Prozeduren zur Einrichtung, Administration und Entfernung von Smart Replications. Bei dem Einrichten der ersten Smart Replication werden im „Replic“-Schema einige nötige Hilfsstrukturen (Views, Functions, Jobs etc.) erstellt, die beim Entfernen der letzten Smart Replication auch wieder gelöscht werden. Datenbankübergreifende Funktionen, etwa zur Administration, sind im Package „ADM\_REPLIC“ auf der Administrations-Datenbank.

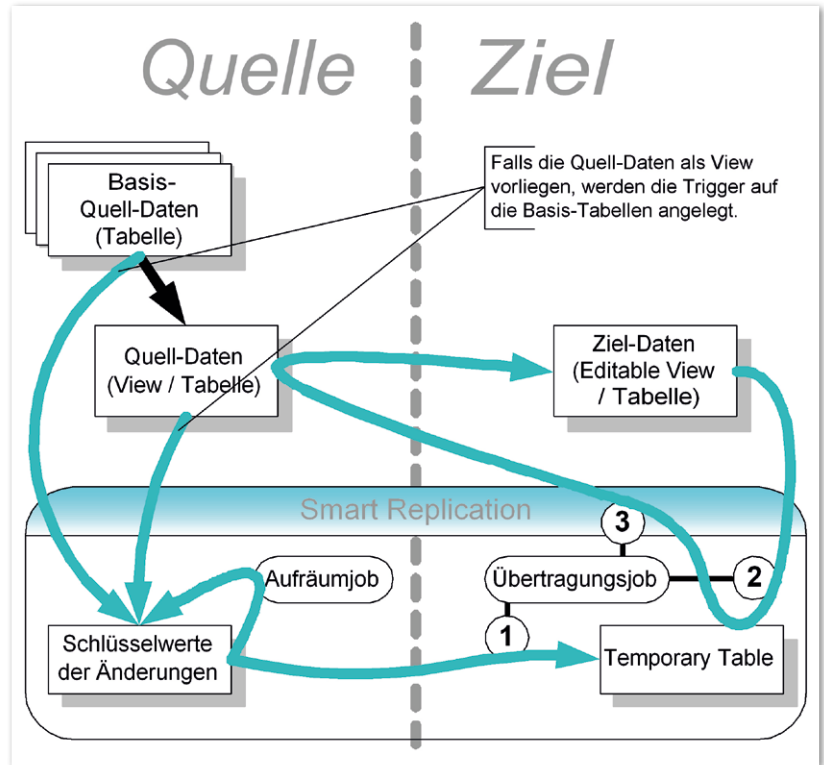


Abbildung 1: Schema-Diagramm der Smart Replication

### Installation

Man geht entsprechend der Dokumentation (siehe „[http://www.doag.org/go/doagnews/smart\\_replication](http://www.doag.org/go/doagnews/smart_replication)“) vor und setzt, unter anderem damit die Compiler-Direktiven korrekt arbeiten, in den Skripten „SR\_install\_...sql“ die „Install“-Konstanten passend (siehe „<http://www.doag.org/go/doagnews/sr>“ und Tabelle 1).

Die Diagnose-Funktion prüft, ob alles grundsätzlich funktioniert, beispielsweise bei Verwendung nur einer Datenbank mit dem Aufruf „clobResult := ADM\_REPLIC.Diagnose(«SCHEMA», „“, «SCHEMA», „“, „“);“. Die Diagnose erstellt vorübergehend einige Test-Tabellen und Views mit mehreren Replikationen, es werden verschiedene Aufrufe getestet und eventuelle Probleme im Ergebnis-Report hervorgehoben.

### Einrichtung einer Smart Replication

Beispielhaft wird jetzt die Einrichtung einer Smart Replication skizziert; Quell- und Ziel-Tabelle seien in zwei verschiedenen Datenbanken vorhanden, gleich geformt und benannt und mit Primärschlüsseln versehen; außerdem existieren die nötigen Datenbank-Links:

««««« EDIT «««««	Weist auf Code-Stellen hin, die gegebenenfalls bearbeitet werden sollten
INSTALL_MINIMAL	Auf „true“ setzen – das „Admin“-Framework ist hier eingeschränkt
INSTALL_USER_ADMIN	Name des Administrations-Users
INSTALL_USER_REPLIC	Name des Replikations-Users
INSTALL_GRANTS_ROLE	Am einfachsten „public“; sonst eine kommagetrennte Liste derjenigen Benutzer, die Replikationen benutzen werden
INSTALL_DIRECTORY_LOG	Directory für Logdateien (sollte häufiger aufgeräumt werden; die Logdateien können groß werden)

Tabelle 1

- *Einrichtung der Senke (Aufruf in der Zieldatenbank)*  
ADMIN.DBA\_REPLIC.SmartTargetAdd(' &&SCHEMA','&&TABELLE');
- *Einrichtung der Quelle (dieser und die weiteren Aufrufe in der Quelldatenbank)*  
ADMIN.DBA\_REPLIC.SmartSourceAdd(' &&SCHEMA','&&TABELLE');
- *Einrichtung des Transfers*  
ADMIN.DBA\_REPLIC.SmartTransferAdd(' &&SCHEMA','&&TABELLE', '&&QUELLEDATENBANK','&&ZIELDATENBANK');
- *Erstellung und Ausführung des Skripts für die Auslöser-Trigger*  
sResult := ADMIN.DBA\_REPLIC.SmartScriptTrigger(' &&SCHEMA','&&TABELLE');
- *execute immediate sResult;*

Das Erstellen des Skripts legt den Trigger noch nicht an, dazu muss das generierte Skript in einem weiteren Schritt ausgeführt werden. Dies gibt gegebenenfalls die Möglichkeit zur Kontrolle und Anpassung. Das Anlegen dieser Trigger ist der entscheidende Punkt bei der Einrichtung; in diesem Schritt werden sozusagen die Datenstrukturen mit der Replikation verbunden und ab diesem Moment die Änderungen repliziert.

Die Initialisierung der Senken bei bereits vorhandenen Daten kann auf viele verschiedene Weisen durchgeführt werden, ein passendes Skript lässt sich beispielsweise auf der Administrations-Datenbank generieren (siehe Listing 1). Das

```
select TEXT_CONTENT from table(ADMIN.ADM_REPLIC.
SmartScript('&&ZIELDATENBANK', '&&SCHEMA','&&TABELLE'));
```

Listing 1

```
select TEXT_CONTENT from table(ADMIN.DBA_REPLIC.SmartReport('%'));
commit;
```

Listing 2

```
select      SR_OWNER, SR_OBJECT,
            SR_OVERDUE, SR_CURRENT,
            trunc(case when SR_SCHEDULED_OVERDUE-SR_SCHEDULED_FIRST >= .9/24
                    and SR_SCHEDULED_FIRST < SR_DATEREF
                    then 100*(SR_DATEREF-SR_SCHEDULED_FIRST)/
                        (SR_SCHEDULED_OVERDUE-SR_SCHEDULED_FIRST)
                    else NULL end) PROZENT,
            case when nvl(SR_JOB_NEXTDATE,to_date('9999','yyyy')) >
                    greatest(SR_SCHEDULED_FIRST, SR_DATEREF + 5/1440)
            then 'JOB_NEXTDATE > SCHEDULED_FIRST'
            else 'ok'
            end NEXTDATE,
            SR_SCHEDULED_FIRST, SR_SCHEDULED_OVERDUE,
            SR_JOB_NEXTDATE, SR_JOB_FAILURES, SR_JOB_BROKEN
from table (REPLIC.SmartStatusFct('%', '%', null, 1))
where nvl(SR_SCHEDULED_FIRST, SYSDATE) < SYSDATE - 5/1440
or nvl(SR_JOB_FAILURES, 0) > 0
or nvl(SR_JOB_BROKEN, 'N') != 'N'
or nvl(SR_JOB_NEXTDATE,to_date('9999','yyyy')) >
greatest(SR_SCHEDULED_FIRST, SR_DATEREF + 5/1440)
order by 1, 2;
```

Listing 3

```
with info as (select OBJEKT, ZEIT, SEKUNDEN, ZEILEN from (
select VON ZEIT, (BIS-VON)*24*60*60 SEKUNDEN, OBJEKT, ZEILEN from (
select OBJEKT, ZEIT VON,
lead(Zeit, 1) over (partition by OBJEKT
order by ZEIT, ISTSTART desc, ISTENDE) BIS,
ZEILEN, IstStart, IstEnde, TEXT from (
select ZEIT, OBJEKT, TEXT,
case when TEXT like '%will now be transferred%'
then 1 else 0 end IstStart,
case when TEXT like '%arget(s) completed.%'
then 1 else 0 end IstEnde,
case when TEXT like '%will now be transferred%'
then to_number(substr(TEXT, 1, instr(TEXT, ' ')))
else NULL end ZEILEN from (
select to_date(substr(TEXT, 1, 20), 'dd.mm.yyyy, hh24:mi:ss') ZEIT,
substr(TEXT, 24, instr(TEXT, ' SmartExecute')-24) OBJEKT,
substr(TEXT, instr(TEXT, ' SmartExecute: ')+
length(' SmartExecute: ')) TEXT
from REPLIC.ALERT_REPLIC_LOG
where TEXT like '%SmartExecute%')
where IstStart = 1 or IstEnde = 1 )
where IstStart = 1
order by OBJEKT, VON))
select OBJEKT, trunc(ZEIT, 'HH'), sum(Zeilen) ZEILEN,
sum(Sekunden) SEKUNDEN, count(1) AUFRUFE
from info
group by OBJEKT, trunc(ZEIT, 'HH')
order by 1, 2;
```

Listing 4

erhaltene Skript wird dann auf der Ziel-Datenbank ausgeführt und überträgt die Daten aus der Quelle.

## Reporting und Monitoring

Abfragen auf „SmartReport“, „SmartStatusFct“ und „ALERT\_REPLIC\_LOG“ geben die Informationen zu Status und Konfiguration der Smart Replications. Eine Auswahl dazu ist die umfassende Status- und Konfigurations-Abfrage (Aufruf in der Quell- oder in der Zieldatenbank, *siehe Listing 2*), die schnelle Abfrage auf eventuelle Probleme (Aufruf in der Quelldatenbank, *siehe Listing 3*) sowie die Analyse der Logdatei nach „Übertragungsvolumen pro Stunde“ (Aufruf in der Quelldatenbank, *siehe Listing 4*).

## Gezieltes Unterbrechen

Smart Replication erlaubt das Unterbrechen von Daten-Übertragungen in ein oder mehrere Senken (etwa bei Verbindungsproblemen) und das anschließende kontrollierte Nachfahren. Das Stoppen der Replikation in eine Senke (die anderen Transfers bleiben aktiv) erfolgt mit „ADMIN.DBA\_REPLIC.SmartTransferSetStatus('«SCHEMA», '«TABELLE», '«ZIELDB\_2», '%', '%', 0);“. Zum Starten aller Replikationen und Nachfahren der angefallenen Änderungen, soweit und wohin nötig, gibt man „ADMIN.DBA\_REPLIC.SmartTransferSetStatus('%', '%', '%', '%', 1);“ ein.

## Reparatur

Grundsätzlich können alle Installations- und Einrichtungs-Prozeduren (auch die Deinstallations- und Entfernungs-Prozeduren) beliebig oft aufgerufen und wiederholt werden; der jeweils erreichte Status wird überprüft und nur die eventuell noch ausstehenden Aktivitäten durchgeführt. Falls etwa während der Einrichtung einer Smart Replication ein Fehler aufgetreten ist, wird nach Behebung der Fehlerursache der verwendete Aufruf einfach wiederholt. Falls beispielsweise Smart-Replication-Objekte versehentlich entfernt wurden, kann zur Reparatur die Prozedur SmartSetup verwendet werden „ADMIN.DBA\_REPLIC.SmartSetup;“. Fehlende Installationsobjekte werden dann wieder eingerichtet.

## Eigenschaften

Die Schlüssel-Qualitäten von Smart Replication sind:

- Massenänderungen werden zügig abgearbeitet und führen nicht zu Ressourcen-Engpässen.
- Die Übertragung von Änderungen lässt sich priorisieren – etwa bei Online-Anwendungen gegenüber Batch-Abläufen – und zeitlich steuern.
- Einfache Konfiguration: Weitgehend automatische Einrichtung, jederzeitige Anpassung der Einstellungen, unkomplizierte Änderung von Datenstrukturen, problemlose Erweiterung etwa um neue Zieldatenbanken, einfaches Kopieren, Reparieren etc.
- Sichere und flexible Administration: Durchgängige Kontrolle über den aktuellen Status und direkter Zugriff auf die Smart-Replication-Komponenten. Im Bedarfsfall sind eine gezielte Drosselung der Übertragungsraten, die Unterbrechung und Umleitung von Datenübertragungen etc. möglich. Im Status „InReorganisation“ können auf Session-Ebene Datenänderungen ohne Folge-Replikation durchgeführt werden.

Wesentliche Unterschiede zu Streams beziehungsweise Advanced Replication sind:

- Die zu übertragenden Inhalte stehen nicht von vornherein fest, sondern nur die den Datensatz identifizierenden Schlüsselwerte. Die zum Übertragungszeitpunkt – nicht dem Änderungszeitpunkt – in der Quelle konkret vorhandenen Werte werden im Ziel eingefügt.

- Hohe Fehlertoleranz: Beispielsweise gibt es keinen Feldinhalts-Vergleich zwischen alten und neuen Einträgen, der gegebenenfalls zu einer Fehlermeldung führt – der Inhalt zum Zeitpunkt der letzten Datensatz-Übertragung liegt vorn. Ein Kollisionshandling ist deshalb nicht ohne Weiteres möglich; bei Bedarf lässt sich der Übertragungsschlüssel geeignet erweitern, etwa unter Einbeziehung des letzten Datensatz-Update-Zeitpunkts, um diese Thematik abzudecken.
- Keine Transaktionstreue und keine Einzelsatzverarbeitung: Smart Replication überträgt in unabhängigen Transaktionen mehrere, eventuell geeignet zusammengefasste Änderungen auf einmal.

Weitere Merkmale sind:

- Abweichend vom Default (der für bi- und multidirektionale Replikationen geeignet ist) kann konfiguriert werden, dass von Smart Replication übertragene Änderungen in einer anschließenden Smart Replication weiter übertragen werden; damit sind Ketten möglich.
- Mit der Verwendung von (Editable) Views lassen sich zu denselben Tabellen völlig verschiedene Smart Replications einrichten.
- Änderungen werden in den vorgesehenen Übertragungsfenstern sofort übertragen, es gibt kein Warten auf den Ablauf eines regelmäßigen Refresh-Intervalls oder Ähnliches.

## Fazit

Smart Replication ist ein individuell entwickeltes Projekt; der Autor hat zwar die Installation allgemeingültig gestaltet und beschrieben, aber es handelt sich nicht um ein Produkt. Vor einer Verwendung sind Prüfungs-, Anpassungs- und Testaufwände fällig. Trotzdem ist der Autor davon überzeugt, dass in Fällen, in denen über den Ersatz der mittelfristig obsoleten Oracle-Techniken durch selbstgeschriebenen Code nachgedacht wird, die Adaption eines seit Jahren produktiven Verfahrens einigen Entwicklungsaufwand sparen kann.



Carsten Kaftan  
c.kaftan.sanacorp@myway.de

# Delegiertenversammlung: DOAG soll offener und internationaler werden

Die Gewinnung von neuen Mitgliedern und Referenten wurde auf der Delegiertenversammlung stark diskutiert. Als eine Maßnahme der Mitgliederakquise sollen digitale Inhalte auf den Webseiten der DOAG zukünftig für jeden frei im Internet verfügbar sein. Die neue, offenere Ausrichtung der DOAG sowie weitere Maßnahmen und Details werden nachfolgend im Vorstand besprochen.

Besonderen Wert legen die Delegierten auch auf die Analyse und Verbesserung des Prozesses zur Gewinnung von Referenten für alle Veranstaltungen der DOAG. Hier wurde eine entsprechende Arbeitsgruppe mit fünf Delegierten eingerichtet, die zusammen mit je zwei Vertretern des Vorstands und der Geschäftsstelle den gesamten Prozess zur Referentengewinnung analysieren und dem Vorstand bis Ende des

Jahres entsprechende Verbesserungsvorschläge präsentieren soll.

Weiterhin stimmten die Delegierten einstimmig der Internationalisierungsstrategie des Vorstands zu. Insbesondere die Jahreskonferenz der DOAG, die DOAG 2015 Konferenz + Ausstellung, soll internationaler werden. Auch eine noch intensivere Vernetzung mit den Usergroups aus den USA und aus Europa ist geplant.