ORACLE Change Data Capture (CDC) – Die Änderungen entscheiden

Günther herzog
metafinanz
München

Schlüsselworte:

Data Warehouse , Change Data Capture, ETL, Business Intelligence

Einleitung

Dieser Vortrag gibt eine Übersicht über die Change Data Capture (CDC) – Technologie, sprich das Bereitstellen von DML-Datenänderungen. Es werden die unterschiedlichen CDC-Methoden sowie praktische Betriebsaspekte aufgezeigt.

Problemstellung Performance / Zeitbedarf der Befüllung eines DWHs

In den meisten DWHs werden Daten historisiert bzw. versioniert. Dafür werden die neuen Versionen der Daten aus den Quelldatenbanken benötigt. Doch bei der periodischen Beschickung eines Data Warehouses mit den neuen Daten ist oft die Zeit die dafür zur Verfügung steht knapp bemessen. Deswegen ist es eine der großen Aufgaben bei der Entwicklung einer DWH-Architektur, die Befüllung so effizient wie möglich zu gestalten.
Es gilt hier neben dem klassischen Datenbanktuning auch die zu Verarbeitenden Datenmengen so klein wie möglich zu halten. Hierbei hilft Change Data Capture bzw. CDC.
CDC ist eine Methodik um nur die Datenänderungen zu erkennen, zur Verfügung zu stellen und dabei so wenig wie möglich Ressourcen zu verbrauchen.

Unterschiedliche Lösungsmöglichkeiten

Es gibt mehrere Möglichkeiten Datenänderungen mit und ohne Einsatz von CDC zu erkennen. Einige seien hier beispielhaft aufgeführt:

Tabellenvergleich

Ermitteln der Datenänderungen indem die gesamte Quell-Tabelle mit dem Inhalt der DWH-Tabelle verglichen wird.
Hierbei ist u.a. ein hoher Zeitaufwand für die Datenübertragung der kompletten Tabelle(n) und das Ermitteln des Deltas einzuplanen. Zwischenzeitliche Updates werden nicht erkannt.
Die Datenkonsistenz ist nicht gewährleistet.

Attributsabhängige Selektion

Erkennung der geänderten Daten aufgrund fachlicher Spalteninhalte mit z.B. dem letzten Änderungsdatum oder Versionsnummer oder ähnliches.
Erfordert eventuell Änderungen an der Quelldatenbank. Zwischenzeitliche Updates werden nicht erkannt oder können nicht zur Verfügung gestellt werden.
Die Datenkonsistenz ist nicht gewährleistet.

Trigger
Datenänderungen werden ermittelt, indem auf jeder zu überwachenden Quelltabelle Trigger definiert werden. Dadurch entsteht jedoch ein hoher Verwaltungsaufwand und eine Beeinträchtigung der Transaktion der Quelldatenbank. Diese Lösung ist nur resourcenschonend wenn es gut programmiert ist. Die Datenkonsistenz ist nicht gewährleistet.

Change Data Capture von ORACLE

Oracle bietet für diesen Zweck seit der Version 9i ein Change Data Capture Framework an, das bei der Datenbankinstallation bereits enthalten ist.
Die Vorteile:

- Weniger Hardwareresourcenverbrauch
- Weniger Datenübertragung
- Leichtes Wiederaufsetzen nach Connection-Abbrüchen
- Beinhaltet Auditing
- Beinhaltet Exception Handling
- Berücksichtigt nur die notwendigen Spalten, es müssen nicht komplette Tabellen überwacht werden
- Transaktionelle Konsistenz für Änderungen über mehrere Quelltabellen

Versionsgeschichte

- **DB 9i**:
  Einführung Synchronous Change Data Capture
- **DB 10gR2**:
  Einführung Distributed Asynchronous Change Data Capture (CDC)
- **Data Integrator (ODI) 10gR2**:
  Einführung Change Data Capture mit den Knowledge Modulen
- **DB 11gR1**:
  CDC ist nach Bedarf zu aktivieren oder zu deaktivieren.
  Das Purgen der Subscriber Views wurde flexibilisiert, so daß man den Zeitpunkt angeben kann, bis zu dem gepurged warden soll.
- **Data Integrator (ODI) 11gR1**:
  Neue Knowledge Modules für Oracle GoldenGate
  ODI benutzt ORACLE GoldenGate, um Daten von einer Quelle zur Staging-Datenbank zu replizieren. Ein journalizing Knowledge Modul managt die ODI-CDC-Infrastruktur und generiert automatisch die Konfiguration für ORACLE Golden Gate
- **Warehouse Builder (OWB) 11gR2**:
  Code template mappings für Change Data Capture auf Basis der Data Integrator (ODI) Knowledge Module

Change Data Capture Implementierungsarten

Es gibt synchrones und asynchrones CDC. Das asynchrone CDC lässt sich nochmals in HotLog, Distributed HotLog und AutoLog unterscheiden.
Nachfolgende Tabelle gibt Aufschluss über die unterschiedlichen Arten:

	Synchronous Hotlog CDC	Asyncronous Hotlog CDC	Asynchronous Distributed Hotlog CDC	Asynchronous Autolog CDC
Quelle/ Mechanismus	Systemtrigger	Online Redo Logfile	Online Redo Logfile, Übertragung durch Streams	Redo Log Files (Transport managed by Redo transport services)
Teil der Quell-transaktion	**Ja**	**Nein**	**nein**	**Nein**
Anzahl be-teiligter Systeme	**1**	**1**	**2**	**2**
Zeitversatz zur Quelltransaktion	kein	Nahe Echtzeit. ChangeTabelle wird befüllt, sobald die neue abgeschlossene Transaktion ankommt	Abhängig von der Topologie. ChangeTabelle wird befüllt, sobald die neue abgeschlossene Transaktion ankommt	Abhängig von der Topologie und der LogSwitch-Intervalle
Seit DB Version	**9i**	**10g**	**10g**	**10g**

Synchronous Change Data Capture
Synchrones Change Data Capture benutzt Systemtrigger um die geänderten und neuen Datensätze vor und nach der Änderung zu erkennen. Sync CDC ermöglicht Echtzeit-Capturing und hat die gleichen Performance-Auswirkungen wie User-Trigger. Es wird das gleiche CDC-Framework-Interface wie beim asynchronen CDC benutzt.

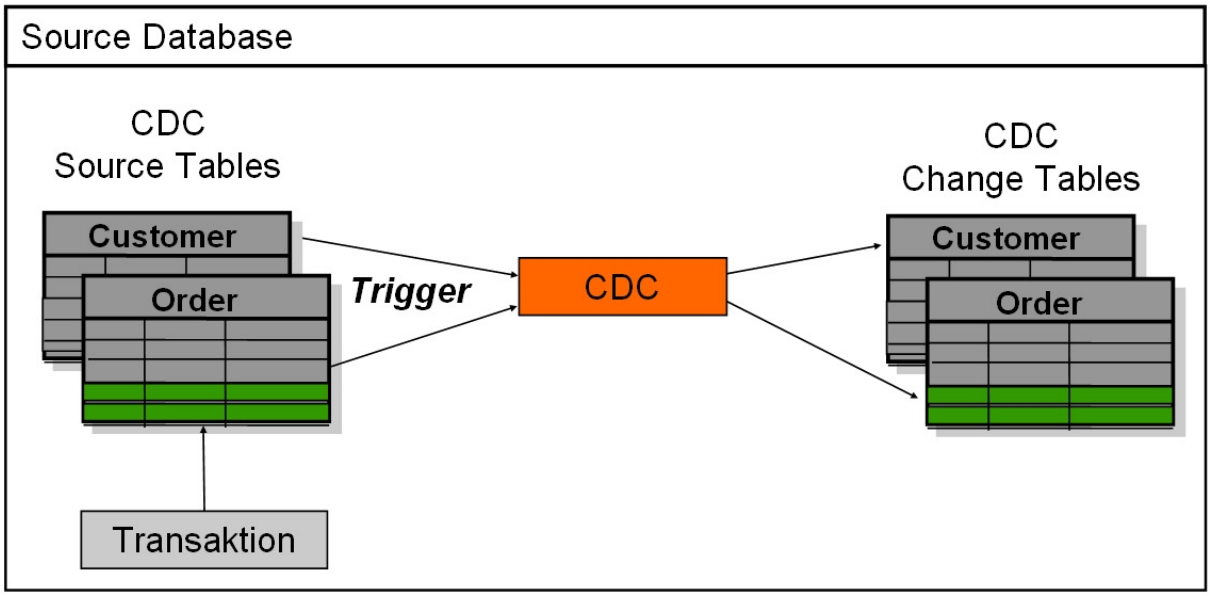

Abb. 1:Synchronous Hotlog Change Data Capture

Asynchronous Change Data Capture

Asynchronous CDC captured die Daten vom online oder den archivierten Redo Logfiles. Asynchrones CDC kann die Änderungsdaten auf der Quell-Datenbank oder auf der Staging-Datenbank zur Verfügung stellen.

Asynchronous Hotlog Change Data Capture
Im Asynchronous Hotlog Mode werden die Änderungsdaten vom Online Redo Log der Quelldatenbank ausgelesen. Es entsteht dadurch nur eine kurze Latenz zwischen dem Commit der Quell-Tabellen-Transaktion und dem Erscheinen der Änderungsdaten in den ChangeTabellen.
Die geänderten Daten werden auf derselben Datenbank zur Verfügung gestellt.

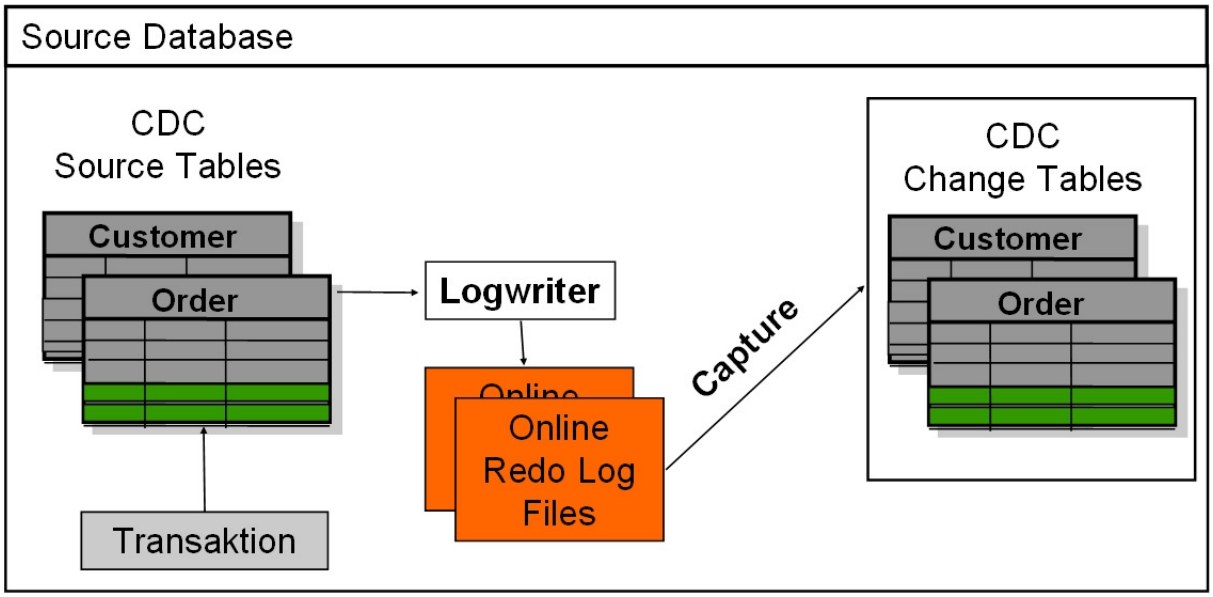

Abb. 1:Asynchronous Hotlog Change Data Capture

Asynchrones Distibuted Hotlog Change Data Capture
Im Asynchronous Distributed Hotlog Mode werden die Änderungsdaten vom Online Redo Log der Quelldatenbank ausgelesen und per Streams an die Staging Datenbank übertragen.

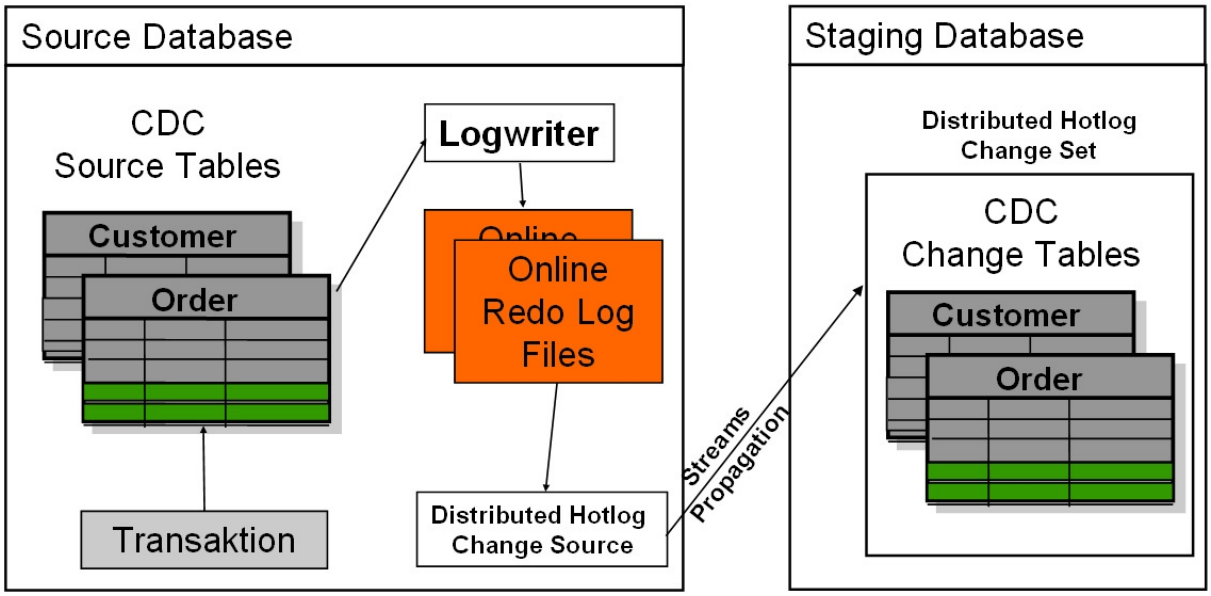

Abb. 1:Asynchronous Distributed Hotlog Change Data Capture

Asynchronous Autolog Change Data Capture
Im Asynchronous Autolog CDC wird als Quelle für die Änderungsdaten die RedoFiles verwendet, die durch den RedoTransportService von der Quell- zur Staging-DB übertragen werden. Unterschieden kann noch zwischen **Autolog Online CDC Mode** (Logwriter schreibt Daten für RedoLog local und auf die StagingDB) und **Autolog Archive CDC Mode** (ArchiverProzess schreibt bei einem Redo Log File Switch die Daten in ein locales Archived Redo Log und ein Redo Logfile auf der Staging DB)

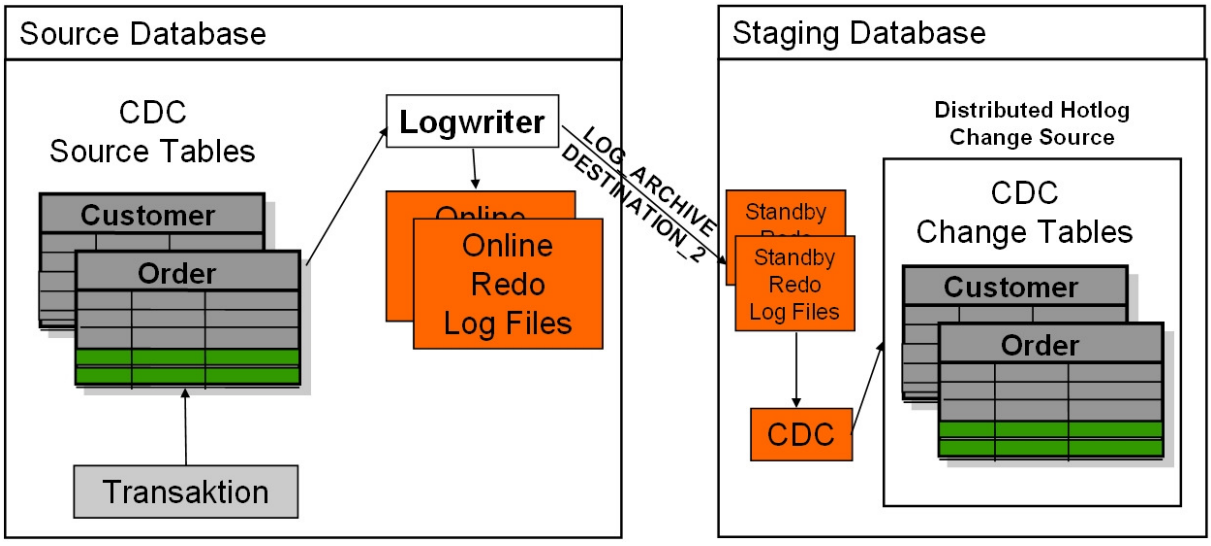

Abb. 1:Asynchronous Autolog Change Data Capture

Publisher/Subscriber Model

ORACLE CDC ist eine Implementation des Observer-Patterns. Hierbei stellt der Publisher die Änderungsdaten zur Verfügung und der oder die Subscriber abonnieren und konsumieren die Daten. Dieses Vorgehen erlaubt die gemeinsame Benutzung der Änderungsdaten und gewährleistet, daß die Änderungsdaten kontrolliert und nicht versehentlich mehrmalig verarbeitet werden. Es wird transaktionelle Konsistenz innerhalb der ChangeTabellen eines Change Sets sichergestellt und ermöglicht dass die bereits verarbeiteten Daten kontrolliert logisch gelöscht werden.

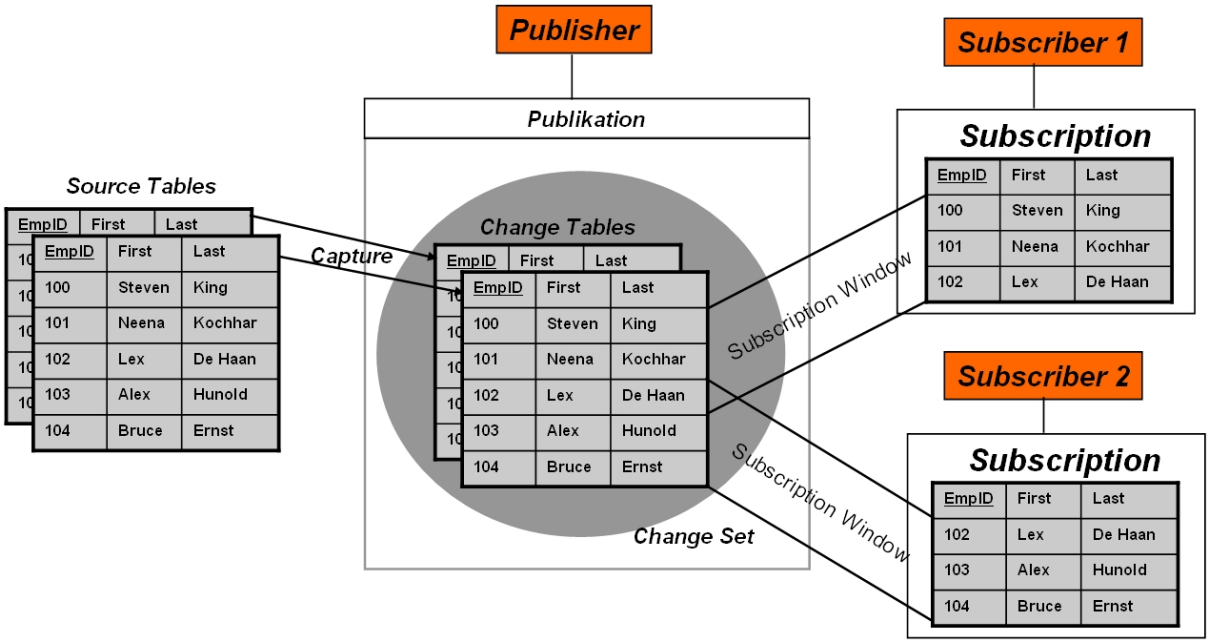

Abb. 1:Publisher/Subscriber Model

Der Publisher
Der Publisher erstellt die ChangeSets – eine logische Sammlung von Änderungsdaten die transaktionell konsistent benötigt werden. Diesen ChangeSets werden die benötigten ChangeTables zugeordnet, die jeweils eine Untermenge der Quelltabellenattribute enthalten.

Übliches Vorgehen des Publishers in einer vorbereiteten Umgebung:

1. Erstellung eines ChangeSets: `DBMS_CDC_PUBLISH.CREATE_CHANGE_SET`
2. Erstellen der ChangeTables: `DBMS_CDC_PUBLISH.CREATE_CHANGE_TABLE`
3. Aktivieren des ChangeSets: `DBMS_CDC_PUBLISH.ALTER_CHANGE_SET`
4. Select-Rechte für Subscriber: `GRANT SELECT ON <ChangeTable> to <Subscriber>`

Der Subscriber
Der Subscriber abonniert mit der Erstellung einer Subscription die Änderungsdaten genau eines ChangeSets. Diese Subscription erstellt eine bewegliche View (Subscription_Window) für die Änderungsdaten, die transaktionell konsistent zur Verfügung gestellt wird. Wenn alle Subscriber die Änderungsdaten in ihren Subscription-Windows logisch gelöscht haben, werden diese Daten auch physikalisch aus den ChangeTables gelöscht.

Übliches Vorgehen eines Subscribers für den Zugriff auf CDC-Daten:

1. Erstellen einer Subscription: `DBMS_CDC_SUBSCRIBE.CREATE_SUPSCRIPTION`
2. Abbonnierten der Daten: `DBMS_CDC_SUBSCRIBE.SUBSCRIBE`
3. Subscription aktivieren: `DBMS_CDC_SUBSCRIBE.ATIVATE_SUBSCRIPTION`
4. Daten in SUBSCRIPTION_WINDOW laden: `DBMS_CDC_SUBSCRIBE.EXTEND_WINDOW`
   Hiermit werden alle Datenänderungen in das Subscription-Window geladen, die seit dem letzten Aufruf von `EXTEND_WINDOW` oder `PURGE_WINDOW` aufgelaufen sind.
5. Daten selektieren und verarbeiten.
6. Daten logisch aus dem SUBSCRIPTION_WINDOW löschen: `DBMS_CDC_SUBSCRIBE.PURGE_WINDOW`
   Die Änderungsdaten sind in den ChangeTabellen physikalisch noch vorhanden.

Change Data Capture Meta-Attribute der ChangeTabellen

Die ChangeTabellen enthalten neben den CDC-Attributen, die gecaptured werden sollen zusätzliche CDC-Metadaten-Attribute.
Hier eine Übersicht über wichtige CDC-Metaattribute in der ChangeTable:

- OPERATION$ (CHAR(2)):
  - 'I' : Insert Operation
  - 'UO': Version vor Update
  - 'UN': Version nach Update
  - 'D' : Delete Operation
- CSCN$ (NUMBER): Commit System Change Number
  - Synchronous CDC: CSCN der EXTEND_WINDOW - Operation.
  - Asynchronous CDC: CSCN der DML operation
- RSID$ (NUMBER):
  - Unique Row Sequenz-ID der Transaktion (nicht transaktionsübergreifend)
  - Möglichkeit, die Source-Table-Version vor dem Update zu identifizieren.
  - Die RSID$ des Updates entspicht der der Version vor dem Update.
- COMMIT_TIMESTAMP$ (DATE): Commit-Zeitpunkt der Transaktion
- TIMESTAMP$ (DATE): Zeitpunkt der Operation in der Quelltabelle
- USERNAME$ (VARCHAR2(30)):
  - Username des Users, der die Operation angestoßen hat.
  - Vor 10.2 NULL.
- ROW_ID$ (ROW_ID): ROW-ID des betroffenen Datensatzes in der Quelltabelle.

Reihenfolge der aufgetretenen Änderungen

SELECT * FROM CHANGETABLE ORDER BY CSCN$,RSID$;

Änderungen im CDC-System

Attribute in Quelltabellen hinzufügen/löschen
Falls in Quelltabellen Attribute gelöscht (drop) oder hinzugefügt (add) wurden, können diese Änderungen in den ChangeTables mit `DBMS_CDC_PUBLISH.ALTER_CHANGE_TABLE` nachgezogen werden.

Subscription ändern / Tabellen reorganisieren
Durch die notwendige Aktivierung der Subscription werden die Subscription-Views angelegt.Es wird dabei implizit festgelegt, daß der Subscriber zu allen notwendigen Tabellen subscribed hat. Einer Subscription kann deswegen nach der Aktivierung keine zusätzliche ChangeTable hinzugefügt werden. Nichtsdestotrotz kann der Publisher zu dem der Subscription zugrundeligenden ChangeSet eine ChangeTable hinzufügen.
Um eine Tabelle zu einer Subscription hinzuzufügen sind folgende Schritte vorzunehmen:

1. Gewährleisten, daß auf den publizierten Quelltabellen keine Daten geändert werden.
2. Das Subscription-Window nochmals um alle CDC-Daten mit `DBMS_CDC_SUBSCRIBE.EXTEND_WINDOW` erweitern.
3. Eventuelle neue/verbliebene CDC-Daten verarbeiten.
4. Subscription-Window löschen (`DBMS_CDC_SUBSCRIBE.PURGE_WINDOW`)
5. Subscription löschen (`DBMS_CDC_SUBSCRIBE.DROP_SUBSCRIPTION`)
6.
    - Tabelle zur Subscription hinzufügen:
      Tabelle publizieren (`DBMS_CDC_PUBLISH.CREATE_CHANGE_TABLE`)
    - Tabelle reorganisieren:
      Die Quelltabelle wird an der OID identifiziert, deswegen ist es notwendig, die ChangeTable zu löschen (`DBMS_CDC_PUBLISH.DROP_CHANGE_TABLE`), danach die Quelltabelle reorganisieren mit z.B. CreateTableAsSelect, Originaltabelle droppen, Kopie Umbenennen, und danach die neue/überarbeitete QuellTabelle publizieren (`DBMS_CDC_PUBLISH.CREATE_CHANGE_TABLE`)
7. Die Subscription neu erstellen `DBMS_CDC_SUBSCRIBE.CREATE_SUBSCRIPTION`
8. Zu allen Tabellen subscriben `DBMS_CDC_SUBSCRIBE.SUBSCRIBE`
9. Subscription aktivieren `BMS_CDC_SUBSCRIBE.ACTIVATE_SUBSCRIPTION`

Lizenzbedingungen für ORACLE Change Data Capture
Synchronous CDC ist ab der Standart Edition enthalten
Asynchronous CDC ist ab der Enterprise Edition enthalten

Kontaktadresse:

Günther Herzog
metafinanz
Leopoldstrasse, 146
D-80804 München

Telefon: +49 (0) 89-360 531 5138
Fax: +49 (0) 89-360 531 15
E-Mail guenther.herzog@metafinanz.de
Internet: www.metafinanz.de