PRIMJENA ORACLE CDC-a (CAPTURE DATA CHANGE) u praksi

Zvonimir Hrkać
Neos d.o.o., Gundulićeva 63 10000 Zagreb
++385 91 488 1359
zvonimir.hrkac@neos.hr
www.neos.hr

SAŽETAK

Tijekom razvoja skladišta podataka redovno se susrećemo sa velikim količinama podataka koje je potrebno efikasno učitati u skladišta. Pri tome se često pojavljuje problem kako identificirati podatke koji su od zadnjeg učitavanja na izvoru promijenjeni, te tako ostvariti inkrementalno učitavanje podataka. Tijekom predavanja biti će obrađena metoda rudarenja log datoteka baznog servera radi utvrđivanje promjena na izvoru podataka korištenjem CDC (Capture Data Change) metode Oracle baze podataka za identifikaciju i praćenje promjena na izvoru. Upoznati ćemo se sa ovom metodom identifikacija promjena na izvoru, vidjeti kako konfigurirati bazu podataka da ju koristi, osvrnuti sa na performanse ovakvih rješenja, pokazati prednosti i upozoriti na nedostatke. Na kratkom primjeru vidjeti ćemo i kako koristiti Oracle ETL/ELT alate (Oracle Warehouse Builder, Oracle Data Integrator) u sprezi sa ovom tehnologijom.

UVOD

Da bismo ostvarili prihvatljive performanse prilikom učitavanja podataka u skladište, gotovo redovno moramo pribjegavati inkrementalnom učitavanju podataka. To znači da priliko ekstrakcije sa izvornog sustava, moramo biti u mogućnosti identificirati podatke koji su se promijenili od zadnje ekstrakcije. U pravilu se prilikom identifikacije oslanjamo ili na audit kolone, pridijeljene izvornim tablicama, ili na specijalizirane log tablice, u kojima se prate promjene na izvoru. I u jednom i u drugom slučaju podaci o promjenama se popunjavaju putem specijaliziranih triggera na izvornim tablicama. Naravno ovakav način praćenja promjena često je nepraktičan. Moguće je da na izvornom sustavu ne postoje audit kolone, da se triggeri u slučajevima različitih intervencija u same podatke od strane programera ne koriste ili najčešće, da nije moguće identificirati obrisane retke. Tako nerijetko, da bismo garantirali da smo u skladištu podataka identificirali sve promjerene na izvornom sustavu, moramo na izvornom sustavu pratiti i brisanje redaka, pomoću zasebnih log tablica, ili u skladištu podataka izračunavati sve promjene koje su se desile uz pomoć minus operatora, ili hash funkcija. Ovakve su metode nepraktične, jer zahtijevaju intervenciju na izvornim sustavima, ili zahtjevno izračunavanje razlika između starog i novog stanja u skladištu podataka.

Kao bolju alternativu baze podataka često za identifikaciju promjena na izvoru nude različite ugrađene mehanizme, koji omogućavaju jednostavno i pouzdano identificiranje promjena na izvoru. Tema ovog predavanja biti će Oracle CDC, mehanizam Oracle baze podataka, koji omogućava praćenje svih promjena od interesa na izvornoj bazi podataka, te njegova integracija u alate Oracle Warehouse Builder i Oracle Data Integrator

ORACLE CDC

Oracle Capture Data Change je mehanizam putem kojega oracle baza podataka omogućava praćenje promjena na tablicama, a bez direktne intervencije u kod objekata.

Oracle baza podataka nudi četiri načina na koji je moguće implementirati Oracle CDC.

- Synchronous CDC

  Pomoću internih baznih triggera, koji se okidaju prilikom izmjena na izvornim tablicama, baza prati promjene i zapisuje ih u za tu svrhu kreirane log tablice. Ova metoda praćenja promjena je dosta slična standardnom praćenju promjena pomoću klasičnih trigera na baznim tablicama. Osnovni nedostatak ovakve konfiguracije, je utjecaj na performanse izvornog sustava, zato jer praćenje promjena postaje dio originalne transakcije. Prednost u odnosu na klasične bazne triggere, koji modificiraju originalnu tablicu, je mogućnost praćenja brisanja redaka, te nemogućnost korisnika objekata (npr. developera) da privremeno onemogući triggere

- Asynchronous Autolog CDC

  Za razliku od sinkronog CDC asinkroni CDC tipično nema veze sa transakcijom koja je učinila promjene na izvornim objektima. Promjene u tablicama se prate nezavisno, što u tipičnom slučaju da postoji mali vremenski odmak(nekoliko sekundi) između vremena kada je promjena nastala, i vremena kada je promjena registrirana. U slučaju Asynchronous Autolog CDC-a redo log file izvorne baze podataka, prebacuje se na odredišnu bazu podataka. Tamo se iz log datoteke, pomoću zasebnog procesa identificiraju sve promjene do kojih je došlo na izvornim tablicama. Utjecaja na performanse izvornog sustava u principu nema, jer se sve obrade izvršavaju na odredišnoj bazi podataka . Jedini sistemski resurs koji se troše su mrežni kapaciteti potrebni za prebacivanje log datoteke.

- Asynchronous Hotlog CDC

  Za razliku od Asynchronous Autolog CDC-a redo log file se ne prebacuje na odredišnu bazu podataka, nego se sve promjene pomoću zasebnog procesa identificiraju na izvornoj bazi, te se tamo pohranjuju u za u tu svrhu kreirane tablice.

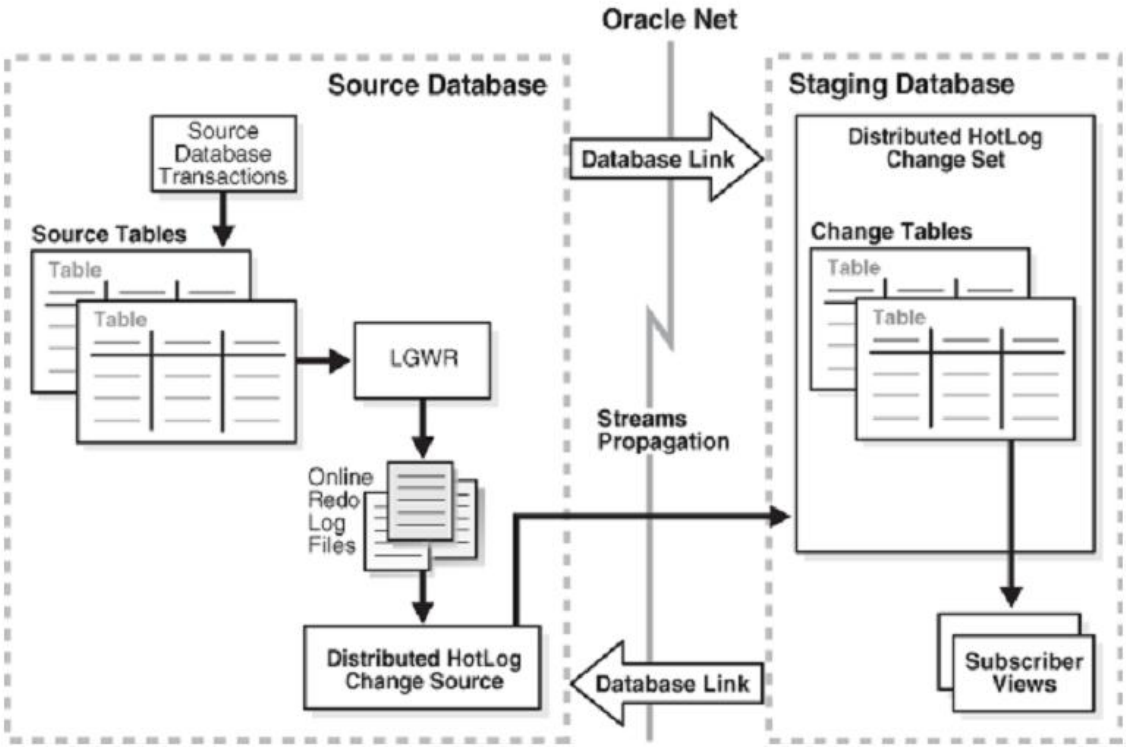
- Asynchronous Distributed Hotlog CDC

  Jedina razlika u odnosu na Hotlog CDC je to što se promjene, kada su jednom identificirane putem dblinka prebacuju na odredišnu bazu podataka.

Svaka od navedenih CDC konfiguracija nudi neke prednosti i nedostatke, međutim jedino se Synchronous CDC izdvaja kao relativno nepraktična CDC postavka, prvenstveno zbog svog utjecaja na performanse izvornog sustava. Sve ostale metode imaju maleni ili nikakav utjecaj na izvorni sustav. Za potrebe ovog rada detaljnije ćemo opisati Asynchronous Distributed Hotlog CDC.

FIZIČKA IMPLEMETACIJA ORACLE ASYNCHRONOUS DISTRIBUTED HOTLOG CDC

Pod fizičkom konfiguracijom podrazumijevamo proces konfiguriranja izvorne i odredišne baze a koji uključuje izmjene na inicijalizacijskim parametrima, te izvršavanje niza procedura iz paketa dbms_cdc_publish, kojima se inicijalizira potrebna bazna infrastruktura(različite tablice, dblinkovi i slično). Detaljan opis procesa konfiguracije oracle baze podataka za Asynchronous Distributed Hotlog CDC može se naći u 16 poglavlju Oracle® Database Data Warehousing Guide koji je standardni dio dokumentacije oracle baze podataka. U ovom dokumentu dati ćemo samo općenite opaske vezane uz konfiguraciju.



Slika 1 ASYNCHRONOUS DISTRIBUTED HOTLOG CDC

Sam proces identifikacije promjene na izvoru teče ovako:

1. Transakcija na izvornoj bazi mijenja sadržaj tablice, koju smo tijekom konfiguracije označili tablicom od interesa(Uključili je u CDC)
2. Pošto se baza nalazi u ARHIVELOG modu, što je preduvjet za funkcioniranje za Asynchronous Distributed Hotlog CDC-a, LOGWRITER proces zapisuje promjene koje su nastupile u online redo log file baze.
3. Zasebni Oracle CDC proces identificira promjene, te koristeći oracle data stream i dblink do odredišne baze, prebacuje retke(što uključuje sve kolone izvorne tablice, koje smo označili interesantnima prilikom konfiguracije) identificirane kao promijenjene na odredišnu bazu podataka
4. Podaci se na odredišnoj bazi zapisuju u zasebne tablice, koje su kreirane tijekom procesa konfiguracije oracle CDC-a. Tablice u sebi sadrže podatke o tome tko je promijenio redak, koja je operacija učinjena na njemu, a u slučaju operacije mijenjanja(UPDATE), tablice mogu sadržavati i vrijednosti prije i nakon promjene.

Za ovaj dio procesa CDC-a, zadužen je specijalno za tu svrhu kreirani bazni korisnik, tzv. PUBLISHER. PUBLISHER je bazni korisnik, zadužen da podatke o promjenama objavi drugim korisnicima koji su zainteresirani za njih(u kontekstu CDC-a zovemo ih SUBSCRIBERIMA). Kako se u opisanom slučaju radi o CDC procesu koji se obavlja istodobno na dvije baze, izvorišnoj i odredišnoj, tako je i publisher kreiran kao zaseban korisnik na obje baze. Publisher na izvornoj bazi prati promjene u tablicama koje su tijekom konfiguracije identificirane kao tablice od interesa, a na odredišnoj bazi pohranjuje identificirane promjene u za to kreirane tablice, te kontrolira pristup do podataka zainteresiranim korisnicima(SUSCRIBERIMA).

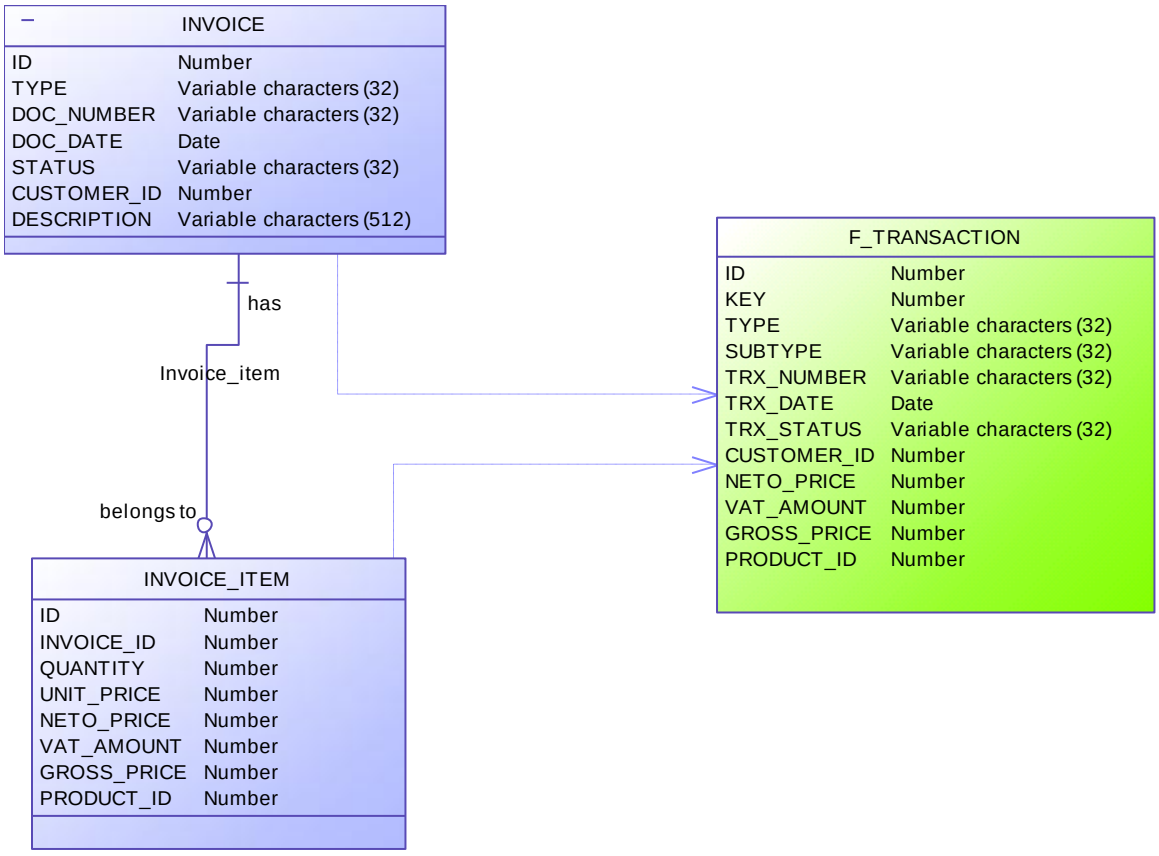
5. Sada korisnici, kojima je dano to pravo(korisnici koji su registrirani za praćenje CDC promjena) putem viewva, koji su automatski kreirani prilikom registracije mogu pratiti promjene na podacima izvorne baze

Sam proces fizičke implementacije je vremenski zahtjevan ali dobro dokumentiran. Pravi izazov predstavlja odlučiti se na kojima ćemo tablicama pratiti promjene i u kojoj količini(jedan od koraka prilikom konfiguracije predstavlja i definiranje kolona čije vrijednosti želimo zapisivati u tablice unutar kojih se prate promjene)

Po mišljenju autora, kao nešto povoljniji pristupi praćenju promjena u okolini skladišta podataka, izdvajaju se Asynchronous Autolog CDC, koji gotovo u potpunosti oslobađa izvorni sustav od izvođenja bilo kakvih dodatnih zadataka(ali uz cijenu nešto veće latencije do identifikacije promjene na izvoru), te Asynchronous Distributed Hotlog CDC koji nudi brzu identifikaciju promjena, ali uz blago opterećenje izvornog sustava(Jer se proces zadužen za rudarenje log datoteke izvršava na izvornoj bazi). Naravno kako konfiguriranje svih vrsta Oracle CDC-a zahtjeva manje ili veće intervencije u konfiguracijske parametre izvorne baze, mora postojati i zainteresiranost administratora izvornog stava da podrži nužne promjene.

LOGIČKA IMPLEMENTACIJA ORACLE CDC

Kada počnemo koristiti Oracle CDC kao sredstvo za praćenje promjena na izvornoj bazi podataka, brzo ćemo uvidjeti i njegovu osnovnu manu. Naime Oracle CDC omogućava brzo i jednostavno praćenje promjena u bazi na fizičkoj razini(na razini pojedine tablice). Ono što ne omogućava, a što je ključno u razvoju skladišta podataka, je praćenje izmjena na logičkoj razini(na razini logičke cjeline što u kontekstu skladišta podataka najčešće znači na razini jedne denormalizirane cjeline). Da bismo to bolje ilustrirali poslužiti ćemo se primjerom:



Slika 2

Pretpostavimo da u našem skladištu podataka između ostalog želimo podacima popunjavati fakt F_TRANSACTION. Ova se fakt tablica puni podacima o različitim transakcijama sa izvornog sustava.

Kao što slika pokazuje jedna takva transakcija može biti račun. Međutim kako se na izvornom sustavu podaci o jednom računu nalaze normalizirani, u dvije tablice: INVOICE i INVOICE_ITEM jasno je da ih moramo denormalizirati da bismo ih pohranili u naše skladište podataka.

Unos jednog računa može se dešavati kroz dulje vrijeme. U takvim slučajevima naš scenarij praćenja promjena unificiranim podacima putem CDC-a pokazati svoje nedostatke.

1. Na izvornoj bazi korisnik kreira račun i dodaje dvije stavke u njega, u sklopu jedne transakcije.

Tabela 1 Invoice

ID	DOC_NUMBER	DOC_DATE	STATUS	CUSTOMER_ID	DESCRIPTION
100	10000-100001	26.9.2009	UNOS	200	NULL

Tabela 2 Invoice item

ID	INVOICE ID	QUANTITY	UNIT PRICE	NETO PRICE	VAT AMOUNT	GROSS PRICE	PRODUCT ID
1000	100	10	100	1000	220	1220	99
1001	100	5	30	150	33	183	65

2. Oracle CDC proces identificira promjene koje su nastupile u sklopu transakcije te ih prebacuje u skladište podataka. Kako kroz CDC pratimo izmjene na svim kolonama, nema potrebe za učitavanjem dodatnih podataka sa izvorne baze podataka. Denormalizacijom pristiglih podataka spremni smo za unos istih u odredišnu fakt tablicu:

Tabela 3 F_TRANSACTION

KEY	TYPE	TRX NUMBER	TRX DATE	TRX STATUS	CUSTOMER ID	NETO PRICE	VAT AMOUNT	GROSS PRICE	PRODUCT ID
1000	Invoice	10000-100001	26.9.2009	UNOS	200	1000	220	1220	99
1001	Invoice	10000-100001	26.9.2009	UNOS	200	150	33	183	65

3. Sljedeći dan, na zahtjev kupca korisniku za proizvod 99 je odobreno 50% rabata. Nakon što CDC identificira promijenjene podatke, dobivamo sljedeći denormalizirani redak za unos u našu fakt tablicu:

KEY	TYPE	TRX NUMBER	TRX DATE	TRX STATUS	CUSTOMER ID	NETO PRICE	VAT AMOUNT	GROSS PRICE	PRODUCT ID
1000						500	110	510	99

Kao što vidimo, nismo dobili punu informaciju o retku koji se promijenio, zato što na zaglavlju računa nije došlo do promjena.

4. Nakon što je sljedeći dan kupac potvrdio da je zadovoljan sa našim računom/ponudom, korisnik potvrđuje račun. Nakon što CDC identificira promijenjene podatke, dobivamo sljedeći denormalizirani redak za unos u našu fakt tablicu:

KEY	TYPE	TRX NUMBER	TRX DATE	TRX STATUS	CUSTOMER ID	NETO PRICE	VAT AMOUNT	GROSS PRICE	PRODUCT ID
	Invoice	10000-100001	26.9.2009	POTVRDA	200				

Ovaj put smo, informaciju dobili samo o zaglavlju računa.

Do sličnog problema može doći i prilikom klasičnog inkrementalnog učitavanja podataka(npr. putem audit kolona), međutim uvijek imamo mogućnost koristiti viewve da kreiramo logičke cjeline podataka koje želimo učitati. U takvoj situaciji, na izvoru bi kao izvor podataka koristili view, koji objedinjuje račun i stavku, pa bi se svaka promjena, bilo na razini zaglavlja ili stavke gledala u kontekstu logičke cjeline kojoj promjena pripada.

Ovaj problem moguće je riješiti na različite načine, jedna od opcija je svakako i parcijalno mijenjanje podataka u odredišnoj fakt tablici,a u odnosu na to koji su se podaci i u kojem kontekstu promijenili. Tako bi u gornjoj situaciji, u slučaju promjene zaglavlja mijenjali podatke u fakt tablici za dva retka, a koji se tiču zaglavlja, u slučaju promjene jedne od stavaka mijenjali bi samo redak koji se tiče konkretne stavke. Ovakvo mijenjanje podataka osim što može biti neprihvatljivo što se tiče performansi , također je sve kompliciranije implementirati kako raste broj tablica koje sudjeluju u jednoj logičkoj cjelini.

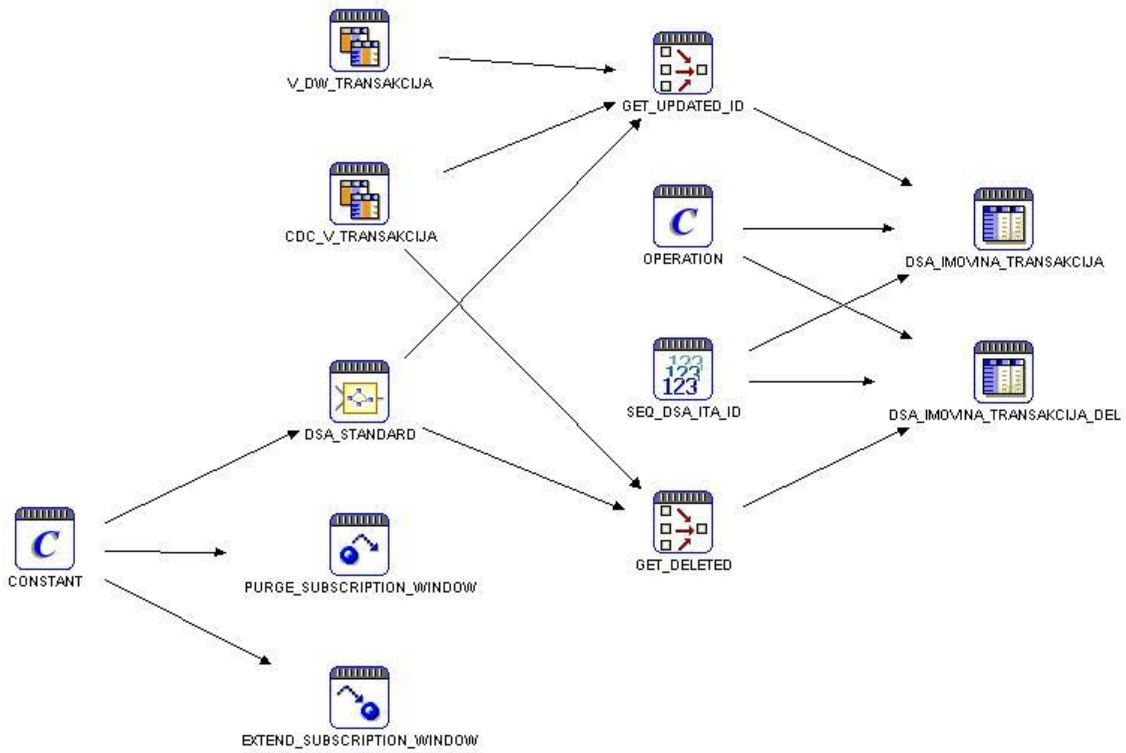
Drugo preporučeno rješenje je korištenje CDC-a samo radi identificiranja redaka koji su se promijenili u izvornim tablicama, a ne za prebacivanje podataka na odredišnu bazu. U takvom kontekstu, CDC bi na odredišnu bazu prebacio samo primarni ključ retka koji se mijenjao. Na izvoru bi zadržali viewve, koji definiraju logičke cjeline, a do promijenjenih podataka bi došli, povezivanjem tablica sa primarnim ključevima mijenjanih redaka i viewva na izvoru. Naravno korištenjem ovakve logike učitavanja podataka odričemo se praćenja slijednih izmjena na izvornim recima koje su nastupile između dva periodu učitavanja. Tako ukoliko je tijekom dana, dva puta mijenjano zaglavlje računa, a učitavanje obavljamo samo krajem dana, identificirati ćemo samo zadnju promjenu. Ovaj problem možemo zaobići češćim učitavanjima podataka.

ORACE CDC I ORACLE WAREHOUSE BUILDER 10G2R

Još prije nekoliko godina, kada je izlazila verzija OWB-a 10.2.0.1 kodnog imena Paris, Oracle je najavio podršku za Oracle CDC ugrađenu u sam alat. Nažalost, zbog tehničkih problema ova značajka alata je izbačena iz konačne verzije OWB-a. Tek sa verzijom 11gR2 OWB će dobiti podršku za CDC. Ta će podrška biti bazirana na mogućnostima koje na području CDC-a nudi novi Oraclov alat ODI(Oracle Data Integrator), a na koji ćemo se detaljnije osvrnuti u sljedećem poglavlju. Usprkos tome i sa starijim verzijama OWB-a moguće je koristiti Oracle CDC i to kombinacijom ručne konfiguracije baze podataka, te pozivima različitih procedura u kontekstu mapiranja Sam proces konfiguracije baze opisan je u prethodnim poglavljima, a rezultat je da bazni korisnik na odredišnoj bazi(kojega u kontekstu CDC-a zovemo publisher) u nizu tablica čuva podatke o svim promjenama koje su nastupile u izvornoj bazi podataka. Da bismo te podatke koristili unutar OWB-a, moramo kreirati novog baznog korisnika koji će biti zadužen na konzumiranje podataka o promjenama. Takvog korisnika nazivamo subscriberom. Subscriber podacima o promjenama koje su nastupile pristupa putem niza viewva, koji se automatski kreiraju za svaku tablicu koja sudjeluje u CDC procesu, a na čije smo se praćenje odlučili.

```
CREATE OR REPLACE FORCE VIEW DSA.CDC_V_TRANSAKCIJA
(
  OPERATION$,
  CSCN$,
  COMMIT_TIMESTAMP$,
  XIDUSN$,
  XIDSLT$,
  XIDSEQ$,
  RSID$,
  TARGET_COLMAP$,
  ID
)
AS
  SELECT   OPERATION$,
        CSCN$,
        COMMIT_TIMESTAMP$,
        XIDUSN$,
        XIDSLT$,
        XIDSEQ$,
        RSID$,
        TARGET_COLMAP$,
        "ID"
   FROM   "CDCPUB"."TRANSAKCIJA_CT"
  WHERE   CSCN$ >= 539743528 AND CSCN$ <= 539743527
```

Kao što vidimo na primjeru,, koji predstavlja standardni view za pristup promjenama identificiranim putem CDC–a, subscriber može pristupiti samo setu podataka koje još nije učitao. O tome se brine where uvjet viewa koji se nanovo generira svaku put kada publisher pozove procedure *DBMS_CDC_SUBSCRIBE.EXTEND_*WINDOW ili DBMS_CDC_SUBSCRIBE.PURGE_WINDOW. Ove procedure modificiraju(regeneriraju) gornji view tj. njezin where uvjet na način da u pogled uključe samo one podatke, koje korisnik još nije učitao. Publisher vodi računa o tome što je za pojedinog subscribera učitano, a što nije. Samo mapiranje kroz koje ćemo konzumirati promijenjene podatke jednostavno je. Korisnik koji pokreće OWB mapiranje ima u kontekstu CDC-a status subscribera. On koristi viewe koji su automatski kreirani prilikom konfiguriranja CDC-a da bi pročitao podatke koji su se promijenili od zadnjeg učitavanja, te ih učitao u bazu.



Slika 3 Owb cdc mapping

U tu svrhu kroz PREMAP proceduru EXTEND_SUBSCRIBTION_WINDOW prije učitavanja podataka modificira se view da uključi nove podatke, a nakon učitavanja podataka sa procedurom PURGE_SUBSCRIPTION_WINDOW označavamo podatke učitanima.

Isto tako vidljivo je da u skladu sa opisanim u prethodnom poglavlju spajamo view V_DW_TRANSAKCIJA koji sadrži čitav logički kontekst transakcije sa viewom CDC_V_TRANSAKCIJA koji sadrži samo identifikator promijenjene transakcije. Na taj način učitavamo potpunu informaciju o transakciji(koja se nalazi u više različitih tablica), a podatke o promjenama pratimo isključivo putem primarnog ključa izvorne tablice transakcija.

ORACLE CDC U ORACLE DATA INTEGRATORU

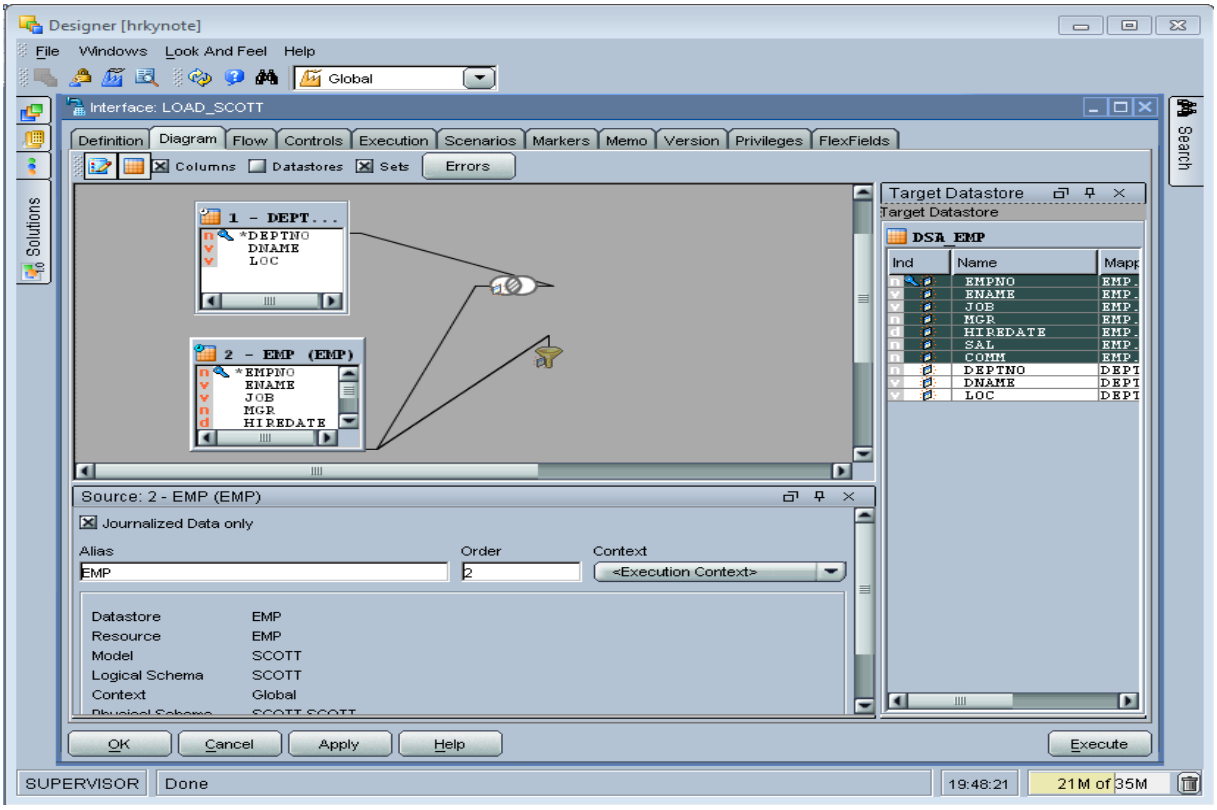
U sklopu niza akvizija obavljenih prošlih godina, Oracle je kupio u Sunopsis, te njihov alat Data Conductor. Funkcionalnost ovog alata uvelike se preklapa sa Oracle Warehouse Builderom, a njegova glavna prednost je vrlo fleksibilna arhitektura koja omogućava podršku za gotovo sve poznate baze podataka. Za razliku od Oracle Warehouse Buildera Oracle Data Integrator ima integriranu podršku za CDC. Čitav koncept rada ODI-ja zasniva se na Knowledge Modulima, konceptu predložaka koda na temelju kojih je ODI u stanju generirati naredbe u skladu sa sintaksom sustava na kojemu radi. Knowledge moduli podržavaju različite akcije koje je moguće izvoditi uz pomoć ODI-ja. Tako imamo module za učitavanje, integraciju, reverzni inženjering i slično. Jedna od takvih vrsta modula su i moduli za journalizing. Journalizing je proces kroz koji Oracle Data Integrator prati promjene do kojih je došlo na podacima, te omogućava inkrementalno učitavanje podataka. Putem journalizinga unutar ODI-ja omogućeno je praćenje promjena na izvoru, tj CDC. Način kako je CDC izveden, pošto je ODI alat namijenjen korištenju u heterogenim okolinama, ovisi o bazu i konkretnom Knowledge Modelu koji je izabran. Za većinu baza to znači kreiranje baznih triggera na tablicama i praćenje promjena putem tih triggera. Među Knowledge modulima koji su na raspolaganju korisniku, odmah poslije instalacije ODI-ja(a kada je baza sa izvornim podacima Oracle) nalazi se i specifičan Oracle JKM(Journaling Knowledge Model) koji podržava učitavanje promijenjenih podataka putem Oracle CDC-a. Ovaj modul između ostalog podržava i automatsko konfiguriranje baze podataka za CDC. Ukoliko želimo upotrijebiti ovu opciju korisnik s kojim startamo Journalizing mora imati DBA prava(da bi mogao mijenjati različite bazne parametre, poput broja dopuštenih paralelnih procesa i sl.). Isto tako prije nego pokrenemo journalizing, baza se mora nalaziti u arhive log modu(ako smo se odlučili za asinkroni journaling). Ukoliko su ova dva uvjeta zadovoljena, ODI će biti u stanju sam konfigurirati bazu za CDC i startati isti.

Nakon što se kroz ODI-jevo sučelje starta CDC(o čemu detaljnije možete pročitati u korisničkoj dokumentaciji), možemo vidjeti da su u shemi korisnika koji je vlasnik tablica čije promjene pratimo kreirane tablice pomoću kojih se prati CDC. Pošto su tablice kreirane na istoj shemi u kojoj se nalazi izvorna baza, zaključujemo da ODI ne barata sa distribuiranim asinkronim CDC-om. Primjer jedne takve tablice vidimo nešto niže. Možemo primijetiti da je i ODI prilikom generiranja CDC tablica uključio samo praćenje vrijednosti ID kolone. Na taj način će i ODI ostvariti logičku konzistentnost podataka u CDC-u na sličan način kao i mi ručno u ranijem OWB primjeru.

Tabela 4

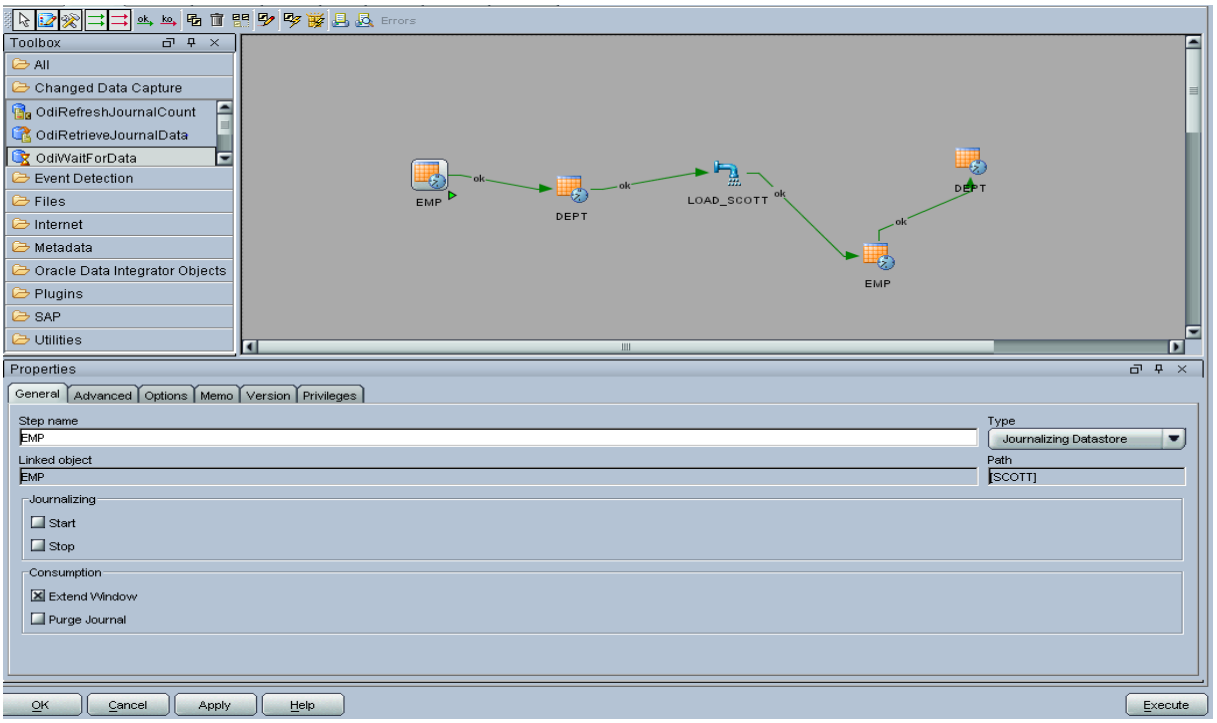
OPERATION$	CSCN$	COMMIT TIMESTAMP$	XIDUSN$	XIDSLT$	XIDSEQ$	TIMESTAMP$	DEPTNO
UN	1.053.780	09.27.2009 18:04:53	8	10	541	09.27.2009 18:04:53	20
UN	1.081.027	09.27.2009 21:34:07	4	9	478	09.27.2009 21:34:07	20
D	1.081.084	09.27.2009 21:35:04	4	3	478	09.27.2009 21:35:04	40

Sam proces journalizinga, u ODI-ju je u potpunosti odvojen od procesa učitavanja podataka. Preostaje nam još kreirati sučelja kroz koja ćemo učitati podatke u odredišnu tablicu.



Slika 4:ODI Jurnalized inteface

Slično kao i u warehouse builderu, još moramo u proces učitavanja uključiti procedure za proširivanje i brisanje prozora u kojem vršimo učitavanje. U ODI-ju se te akcije definiraju tijekom definicije procesa.



Slika 5: ODI Process

Kao što vidimo koncept uključivanja journalizinga u ODI je vrlo sličan onom od OWB-a. Jedina prava razlika je količina automatizma(što se posebno tiče konfiguracije baze) koju odi obavlja samostalno. U scenariju iz stvarnog života konfiguraciju pogotovo produkcije baze ionako nećemo nikada prepustiti alatu na automatsko korištenje, tako da je razlika između ova dva alata prilikom definiranja inkrementalno CDC učitavanja zanemariva.

ZAKLJUČAK

Oracle CDC napredna je mogućnost Oracle baze podataka, koja nam otvara čitav niz mogućnosti prilikom rada sa skladištima podataka. Korištenjem iste, u mogućnosti smo implementirati skladišta podatka koja funkcioniraju u realnom vremenu(bez kašnjenja učitavanja podataka), te istovremeno znatno preciznije pratiti promjene koje su se dešavale na izvoru. Također kroz koncept subscribera omogućava nam jednostavno praćenje koji su podaci učitani, a koji tek trebaju biti učitani.

Još donedavno ova mogućnost Oracle baze podataka nije bila kvalitetno implementirana o alatima namijenjenima ETL-u. Nasreću, pojavljivanjem ODI-ja kao novog Oraclovog alata namijenjenog integraciji podataka, te izdavanjem nove verzije Oracle Warehouse Buildera(11gR2), stanje se znatno popravilo. Može se reći da ovaj način učitavanja podataka zasigurno očekuje bogata budućnost, sada kada je njegova implementacija postala znatno jednostavnija.