

"Stub" redovi

Ivan Korać

U jednom velikom data-warehouse sistemu, podaci stižu sa svih strana sveta, sa različitim platformi i u različitim formatima. Osnovna podela podataka u takvom sistemu je na master i transakcione podatke.

Prvi podrazumevaju slogove za parent tabele, gde se smeštaju uglavnom tekstualni opisi različitih id-a (šifarnici) i/ili hijerarhijski "visoki" podaci tj. potencijalne cube-dimenzije. Drugu grupu podataka čine "pravi" podaci (za fact tabele) sa mnoštvom spuštenih ključeva (foreign key) iz ovih prvih.

U toj gunguli podataka koji se estrahuju iz svakojakih sistema, jedna od sasvim "regularnih" situacija je da vam stignu tzv. transakcioni podaci BEZ odgovarajućih master-a. Ono što vi znate, a sistem nezna je da će kasnije stići i master podaci, ali pošto imate (naravno) podignute kontrole na sistemu (constrainte) koji ne dozvoljavaju tako nešto, neće biti dozvoljena upsert (update ili insert) operacija.

Posledica: takvi transakcioni slogovi NEĆE biti ubačeni u DW.

I šta sad?

Tu počinje priča o takozvanim "stub" redovima i načinima za njihovo procesiranje.

Jedan od metoda kojim se obezbeđuje insert i ovakvih slogova u DW, uz svo poštovanje fk constraint-a, je da se dodaju nedostajući master redovi.

Za to je prvo potrebno definisati jedan user-defined exception (primer 1): npr. master_omitted, u kome će se prepoznati pokušaj narušavanja referencijelnog integriteta (oracle greška: -2291), a zatim parče koda koje treba da prepozna i "dopuni" nedostajuće master redove.

```
-----  
PROCEDURE WriteWAR_Sales_Volume (  
    p_rec_sl      WAR_SALES_VOLUME%ROWTYPE,  
    p_ParentRepLog IN MPCIROOT.MPCI_REPORT.T_MPCI_REPORT_LOG )
```

```
IS -----  
l_RepLog      MPCIROOT.MPCI_REPORT.T_MPCI_REPORT_LOG;  
l_counter      Number(9) := 0;  
master_omitted EXCEPTION;  
PRAGMA EXCEPTION_INIT(master_omitted, -2291);  
BEGIN
```

Primer 1

U exception delu code-block-a se ispituje koji master redovi nedostaju (Primer 2). Treba "izvrteti" sve postojeće fk relacije jer se može desiti da za jedan transakcioni red nedostaje više master redova. Najefikasnije je to uraditi preko **exist** klauzule (u primeru) koja će proveriti postojanje master id-a samo do njegove prve pojave i time će se izbeći FTS (full table scan) operacija na obično, ogromnoj master tabeli. Ako u master tabeli nema takvog, insertuje se slog i



time se obezbeđuje da pri ponovnom pokušaju upserta transakcionog reda referencijalni integritet bude ispoštovan.

```
EXCEPTION
WHEN master_omitted
THEN
--
select count(*) into l_counter
from dual
where exists(select NULL
              FROM WAR_DEALER
              WHERE DEALER_CODE = p_rec_sl.DEALER_CODE);
if l_counter = 0
then
  g_nCntMasterDEALER := g_nCntMasterDEALER + 1;
--
  insert into WAR_DEALER(DEALER_CODE,    MPC1_CREATE_DATE)
  values    (p_rec_sl.DEALER_CODE, p_rec_sl.MPC1_CREATE_DATE);
end if;
--
select count(*) into l_counter
from dual
where exists(select NULL
              FROM WAR_CLAIM
              WHERE CLAIM_NO = p_rec_sl.CLAIM_NO);
if l_counter = 0
then
  g_nCntMasterCLAIM := g_nCntMasterCLAIM + 1;
--
  insert into WAR_CLAIM(CLAIM_NO,    MPC1_CREATE_DATE)
  values    (p_rec_sl.CLAIM_NO, p_rec_sl.MPC1_CREATE_DATE);
end if;
--
...

```

Primer 2

I na kraju, pošto nedostatak master redova i nije baš "skroz regularna" situacija i može da nastane zbog nekog problema, zgodno je odbrojati da li je i koliko master redova nedostajalo i kroz odgovarajući GUI interface portal/software prijaviti to korisniku.