

Brzina je potrebna

Ivan Korać

Vrlo često se u praksi dešava sledeće: Da testni sistem u nekoj zemlji (sa svim atributima produkcionog) treba da postane produkcioni, al pod odma', bez ponovnog reinstaliranja.

“Mi smo testirali i sad hoćemo da startujemo sa prvim podacima ali treba nam to što pre, sve čisto, da bi mogli da punimo prave podatke”.

(Z)Briši ako možeš

I to i nije neki problem, kad treba sve tabele počistiti. S obzrom da truncate comanda NE radi kad su podignuti "foreign constrainti", potrebno je iskoristiti sledeća 3 dinamička skripta za kreiranje: alter table skriptova za: disabling, truncating i enabling FK constraint-a za naprimer WARRANTY modul.

```
set heading off
set feedback off
spool truncate.tmp
```

```
select 'alter table ' || table_name || ' disable constraint ' || constraint_name || ';'
from user_constraints where constraint_type = 'R'
and table_name like 'WAR_%'
/
```

```
select 'truncate table ' || object_name || ';'
from user_objects where object_type = 'TABLE'
and object_name like 'WAR_%'
/
```

```
select 'alter table ' || table_name ||
      ' enable constraint ' || constraint_name || ';'
from user_constraints where constraint_type = 'R'
and table_name like 'WAR_%'
/
```

```
spool off
set heading on
set feedback on
```

Sad skriptove treba izvući iz fajla: truncate.tmp i iskrojiti u zasebne sql fajlove koji se izvršavaju jedan za drugim. Primeri rezultujućih skriptovi su redom slika 1,2,3



OMEGA

```
alter table WAR_CLAIM disable constraint WAR_CLAIM_DEALER_FK;
alter table WAR_CLAIM_ERR disable constraint WAR_CLAIM_ERR_CLAIM_FK;
alter table WAR_COST_DIST_DETAIL disable constraint WAR_COST_DIST_DETAIL_FACT_FK;
alter table WAR_COST_DIST_DETAIL disable constraint WAR_COSTDIST_COSTCARRCODE_FK;
alter table WAR_DAMAGE_ERR disable constraint WAR_DAMAGE_ERR_FACT_FK;
alter table WAR_DEALER_CATEGORY disable constraint WAR_DEALER_CATEGORY_FK;
alter table WAR_DEALER_DETAIL_H disable constraint WAR_DEALER_DETAIL_H_DEALER_FK;
alter table WAR_DEALER_GROUP disable constraint WAR_DEALER_GROUP_FK;
alter table WAR_DECISION disable constraint WAR_DECISION_FACT_FK;
alter table WAR_DECISION_CODE disable constraint WAR_DECISION_CODE_FACT_FK;
alter table WAR_FACT disable constraint WAR_FACT_CLAIM_FK;
alter table WAR_FACT disable constraint WAR_FACT_CONST_GROUP_TEXT_FK;
alter table WAR_FACT disable constraint WAR_FACT_DAMAGE_PART_TEXT_FK;
alter table WAR_FACT disable constraint WAR_FACT_DAMAGE_TYPE_TEXT_FK;
alter table WAR_FACT disable constraint WAR_FACT_PARTS_TEXT_FK;
alter table WAR_FACT disable constraint WAR_FACT_FTL_DAMAGE_TEXT_FK;
alter table WAR_LABOUR_RATES disable constraint WAR_LABOUR_RATES_DEALER_FK;
alter table WAR_OPER_POS disable constraint WAR_OPER_POS_FACT_FK;
alter table WAR_OPER_POS disable constraint WAR_OPER_POS_OPER_CODE_TEXT_FK;
alter table WAR_PARTS_POS disable constraint WAR_PARTS_POS_FACT_FK;
alter table WAR_PARTS_POS disable constraint WAR_PARTS_POS_PARTS_TEXT_FK;
alter table WAR_PARTS_TEST disable constraint WAR_PARTS_TEST_FACT_FK;
alter table WAR_SALES_VOLUME disable constraint WAR_SALES_VOL_SUBCATEGORY_FK;
alter table WAR_SUBLET_POS disable constraint WAR_SUBLET_POS_FACT_FK;
alter table WAR_TEST_REMARK disable constraint WAR_TEST_REMARK_FACT_FK;
alter table WAR_VEHICLE disable constraint WAR_VEHICLE_CLAIM_FK;
alter table WAR_VEHICLE disable constraint WAR_VEHICLE_SUBCATEGORY_FK;
alter table WAR_VEH_SERV_PKG disable constraint WAR_VEH_SERV_PKG_VEHICLE_FK;
```

Slika 1



O M E G A

```
truncate table WAR_CLAIM;  
truncate table WAR_CLAIM_CREDITNO_S;  
truncate table WAR_CLAIM_DECISION_S;  
truncate table WAR_CLAIM_ERR;  
truncate table WAR_CLAIM_EXC_RATE;  
truncate table WAR_CLAIM_MB99930;  
truncate table WAR_CLAIM_MB99980;  
truncate table WAR_CONST_GROUP_ASSIG;  
truncate table WAR_CONST_GROUP_TEXT;  
truncate table WAR_COST_CARRIER_CODE;  
truncate table WAR_COST_CARRIER_KIND;  
truncate table WAR_COST_DIST_DETAIL;  
truncate table WAR_CURR_EXC_RATE;  
truncate table WAR_DAMAGE_COSTDIST_S;  
truncate table WAR_DAMAGE_CREDITRUN_S;  
truncate table WAR_DAMAGE_DEBITRUN_S;  
truncate table WAR_DAMAGE_DECISION_S;  
truncate table WAR_DAMAGE_ERR;  
truncate table WAR_DAMAGE_PART_TEXT;  
truncate table WAR_DAMAGE_TEXT;  
truncate table WAR_DAMAGE_TYPE_TEXT;  
truncate table WAR_DEALER;  
truncate table WAR_DEALER_CATEGORY;  
truncate table WAR_DEALER_DETAIL;  
truncate table WAR_DEALER_DETAIL_H;  
truncate table WAR_DEALER_GROUP;  
truncate table WAR_DECISION;  
truncate table WAR_DECISION_CODE;  
truncate table WAR_EVALUATION_METHOD;  
truncate table WAR_FACT;  
truncate table WAR_FTL_DAMAGE_TEXT;  
truncate table WAR_LABOUR_RATES;  
truncate table WAR_LANGUAGE_TRANSLATE;  
truncate table WAR_LATEST_CREDIT_RUN;  
truncate table WAR_MANUFACTURER;  
truncate table WAR_MANUF_LABOUR;  
truncate table WAR_MILEAGE_IND;  
truncate table WAR_OPER_CODE_TEXT;  
truncate table WAR_OPER_POS;  
truncate table WAR_PARTS_POS;  
truncate table WAR_PARTS_PRICE;  
truncate table WAR_PARTS_TEST;  
truncate table WAR_PARTS_TEXT;  
truncate table WAR_PRODUCT_TREE;  
truncate table WAR_REPAIR_TYPE;  
truncate table WAR_SALES_VOLUME;  
truncate table WAR_SERVICE_TYPE;  
truncate table WAR_SUBCATEGORY_TEXT;  
truncate table WAR_SUBLET_POS;  
truncate table WAR_TEST_REMARK;  
truncate table WAR_TYPE_USP_TEXT;  
truncate table WAR_VEHICLE;  
truncate table WAR_VEH_SERV_PKG;
```

Slika 2

```

alter table WAR_CLAIM enable constraint WAR_CLAIM_DEALER_FK;
alter table WAR_CLAIM_ERR enable constraint WAR_CLAIM_ERR_CLAIM_FK;
alter table WAR_COST_DIST_DETAIL enable constraint WAR_COST_DIST_DETAIL_FACT_FK;
alter table WAR_COST_DIST_DETAIL enable constraint WAR_COSTDIST_COSTCARRCODE_FK;
alter table WAR_DAMAGE_ERR enable constraint WAR_DAMAGE_ERR_FACT_FK;
alter table WAR_DEALER_CATEGORY enable constraint WAR_DEALER_CATEGORY_FK;
alter table WAR_DEALER_DETAIL_H enable constraint WAR_DEALER_DETAIL_H_DEALER_FK;
alter table WAR_DEALER_GROUP enable constraint WAR_DEALER_GROUP_FK;
alter table WAR_DECISION enable constraint WAR_DECISION_FACT_FK;
alter table WAR_DECISION_CODE enable constraint WAR_DECISION_CODE_FACT_FK;
alter table WAR_FACT enable constraint WAR_FACT_CLAIM_FK;
alter table WAR_FACT enable constraint WAR_FACT_CONST_GROUP_TEXT_FK;
alter table WAR_FACT enable constraint WAR_FACT_DAMAGE_PART_TEXT_FK;
alter table WAR_FACT enable constraint WAR_FACT_DAMAGE_TYPE_TEXT_FK;
alter table WAR_FACT enable constraint WAR_FACT_PARTS_TEXT_FK;
alter table WAR_FACT enable constraint WAR_FACT_FTL_DAMAGE_TEXT_FK;
alter table WAR_LABOUR_RATES enable constraint WAR_LABOUR_RATES_DEALER_FK;
alter table WAR_OPER_POS enable constraint WAR_OPER_POS_FACT_FK;
alter table WAR_OPER_POS enable constraint WAR_OPER_POS_OPER_CODE_TEXT_FK;
alter table WAR_PARTS_POS enable constraint WAR_PARTS_POS_FACT_FK;
alter table WAR_PARTS_POS enable constraint WAR_PARTS_POS_PARTS_TEXT_FK;
alter table WAR_PARTS_TEST enable constraint WAR_PARTS_TEST_FACT_FK;
alter table WAR_SALES_VOLUME enable constraint WAR_SALES_VOL_SUBCATEGORY_FK;
alter table WAR_SUBLET_POS enable constraint WAR_SUBLET_POS_FACT_FK;
alter table WAR_TEST_REMARK enable constraint WAR_TEST_REMARK_FACT_FK;
alter table WAR_VEHICLE enable constraint WAR_VEHICLE_CLAIM_FK;
alter table WAR_VEHICLE enable constraint WAR_VEHICLE_SUBCATEGORY_FK;
alter table WAR_VEH_SERV_PKG enable constraint WAR_VEH_SERV_PKG_VEHICLE_FK;

```

Slika 3

Malo hoću, malo neću

Pošto podaci stižu iz različitih izvora i koristi se mnoštvo različitih "loader-a", nije nimalo retko da se otkrije problem u jednom delu podataka. Tada dolazimo do mnogo "zapatlijenijeg" zahteva: korisnici hoće da im izbrišete podatke iz nekih tabela, a pri tome, ni oni ni vi, nemate pojma šta sve ima od child tabela, i kog nivoa su ta ugnježdenja (child tabela je dalje parent za neku drugi itd.).

Apsolutno najbolji način je imati proceduru u kernel package-u koja će te provere i truncate child tabela da odradi za vas. Rešenje koje ja koristim se zasniva na primeru koji sam dobio od Thomasa Kyta:

```

SQL> create table p ( x int primary key );
Table created.
SQL> create table c1 ( c1x int primary key, x references p );
Table created.
SQL> create table c2 ( c2x int primary key, x references p );
Table created.
SQL> create table c3 ( c3x int primary key, x references p );
Table created.
SQL> create table c4 ( c4x int primary key, x references c1 );
Table created.
SQL> create table c5 ( c5x int primary key, x references c4 );

```



Table created.

```
SQL> create table c6 ( c6x int primary key, x references c5 );
```

Table created.

```
SQL> create table c7 ( c7x int primary key, x references c7, y references c6 );
```

Table created.

```
SQL> create or replace
```

```
2 procedure recursive_truncate( p_tname in varchar2,
3                               p_storage in varchar2 default NULL,
4                               p_recurse_lvl in number default 0)
5 as
6   type array is table of varchar2(255) index by binary_integer;
7   l_stmts array;
8
9   procedure p( p_str in varchar2 )
10  is
11  begin
12    dbms_output.put_line
13    ( rpad(chr(9), p_recurse_lvl, chr(9)) || p_str );
14  end;
15  procedure exec( p_str in varchar2 )
16  is
17  begin
18    p( p_str );
19    execute immediate p_str;
20  end;
21  begin
22    p( 'enter: ' || p_tname );
23    for x in ( select *
24              from user_constraints
25              where r_constraint_name in
26                    ( select constraint_name
27                      from user_constraints
28                      where table_name = upper(p_tname)
29                        and constraint_type in ('P','U') )
30              and constraint_type = 'R'
31              and status = 'ENABLED' )
32    loop
33      exec( 'alter table ' || x.table_name ||
34            ' disable constraint "' || x.constraint_name || '"' );
35      recursive_truncate( x.table_name, p_storage, p_recurse_lvl+1 );
36      l_stmts( l_stmts.count+1 ) := 'alter table ' || x.table_name ||
37                                    ' enable constraint "' ||
38                                    x.constraint_name || '"';
39    end loop;
40
```



```
41  exec( 'truncate table ' || p_tname || ' ' || p_storage );
42
43  for i in 1 .. l_stmts.count
44  loop
45      exec( l_stmts(i) );
46  end loop;
47 end;
48 /
```

Procedure created.

SQL>

SQL> truncate table p;

truncate table p

*

ERROR at line 1:

ORA-02266: unique/primary keys in table referenced by enabled foreign keys

SQL> exec recursive_truncate('P')

enter: P

alter table C1 disable constraint "SYS_C007908"

enter: C1

alter table C4 disable constraint "SYS_C007914"

enter: C4

alter table C5 disable constraint "SYS_C007916"

enter: C5

alter table C6 disable constraint "SYS_C007918"

enter: C6

alter table C7 disable constraint "SYS_C007921"

enter: C7

alter table C7 disable constraint "SYS_C007920"

enter: C7

truncate table C7

truncate table C7

alter table C7 enable constraint "SYS_C007920"

truncate table C6

alter table C7 enable constraint "SYS_C007921"

truncate table C5

alter table C6 enable constraint "SYS_C007918"

truncate table C4

alter table C5 enable constraint "SYS_C007916"

truncate table C1

alter table C4 enable constraint "SYS_C007914"

alter table C2 disable constraint "SYS_C007910"

enter: C2

truncate table C2

alter table C3 disable constraint "SYS_C007912"



```
enter: C3
truncate table C3
truncate table P
alter table C1 enable constraint "SYS_C007908"
alter table C2 enable constraint "SYS_C007910"
alter table C3 enable constraint "SYS_C007912"
```

```
PL/SQL procedure successfully completed.
SQL>
```