

Migriranje Data Warehouse-a

Ivan Korać

Razlikujemo pre svega dva osnovna tipa DW:

1. Star Schema Model (SSM)
2. Snowflake Schema Model (SFSM)

Star Schema Model

Star schema je dominantan model u razvoju data warehouse-a pre svega zbog **performansi upita** (query). SSM se može predstaviti kao zvezda, sa velikom tabelom u centru koja sadrži **fact** (detaljne) podatke, i **dimension** (dimenzije) tabelama koje je okružuju, sa kojima je povezana preko primary i foreign ključeva. Za razliku od ostalih struktura SSM ima denormalizovane dimenzije.

Osobine SSM:

- Razumljiv korisnicima jer mu je struktura jednostavna i elementarna, sa jasnom metadata strukturom (definicije: fact-a, dimenzija, hijerarhije i relacija, pravila integriteta i kontrola-constraints).
- Ima najbrže moguće rezultate querija i optimizovan broj join-a: fact i dimension tabela.
- Nema arhivske (history) tabele, a izmene podataka se naznačuju preko indikatora (level fields)
- Podržavaju ga skoro svi DW softverski alati
- Zahteva dosta vremena za razvoj zbog visokog nivoa denormalizacije.

Snowflake Schema Model

Snowflake model je znatno bliži ERD-u (entity-relationship-diagram) jer su dimenzije normalizovane i sa klasičnom hijerarhijom. Zbog toga SFSM ima znatno bolje **load performanse** i koristi se pre svega tamo gde je potrebno što pre napuniti podatke, dok su performanse upita u drugom planu.

Osobine SFSM:

- Ima složenu metadata strukturu, ali koju je mnogo lakše prilagoditi različitim zahtevima



O M E G A

- Brže puni manje normalizovane tabele ali i ozbiljno degradira performanse zbog velikog broja join-a
- Dozvoljava izmene podataka korišćenjem arhivskih (history) tabela
- Relativno brzo se izgrađuje.

Ovo su dva najzastupljenija modela ali postoje i neki drugi (Constellation model, 3NF sa izvedenim data mart-ovima itd.). Šta god da se odlučite da koristite, preporuka je: **NE** koristite mešavinu tehnika ili modela. Ako nemate od početka jasnu strategiju u modeliranju, cena u vremenu će biti pre ili kasnije plaćena, a vreme u biznisu ima najvišu cenu.

Bez obzira koji model je u pitanju, u svakom od njih se srećemo sa velikom količinom redundantnih podataka. To znači da se isti podatak nalazi na više mesta u modelu, bilo u svom izvornom obliku, bilo kao deo neke kalkulacije (summary data).

Postoje različite tehnike kojima se postiže i održava ova redundantnost (dodatna polja, agregirane tabele, matrijalizovani view-i itd.) ali ono što pre svega karakteriše jedan DW je da se broj redundantnih podataka kroz vreme nikad ne smanjuje, već samo povećava. Zato nije nimalo jednostavno migrirati jedan DW iz starije u noviju verziju modela. Novi model ima sigurno nove tabele-dimenzije i još više redundantnih podataka, ali ono što predstavlja dobar osnov je da: **sve ono što postoji u starom modelu DW, postoji i u novom.**

Na ovoj ideji se i baziraju sledeći skriptovi za migriranje DW:

1. **TableMapping.sql** – je zadužen da iz starog DW selektuje sve tabele seta za migriranje.
2. **DataMigration3040.sql** – kreira prelazne tabele koristeći prvi skript, a zatim otvara DB link i kreira statement-e za migraciju, koje na kraju spool-uje u MigrStatements.sql
3. **MigrStatements.sql** – skriptovi spremni za migriranje koje samo treba startovati

```
-- @TableMapping.sql

define DB_SOURCE=BZ30

define DB_TARGET=BENZ

define DB_COUNTRY_CODE=9999

define DB_INSTALL_SRC_PATH=.

SPOOL &DB_INSTALL_SRC_PATH.\TableMapping.sql

select 'INSERT INTO BENZROOT.BENZ_3040_MIGR (SCHEMA40, TABLE40, TABLE30)
```

Omega magazin



O M E G A

```
VALUES ('||chr(39)||'BENZcccc' ||chr(39)||' '||rpad(chr(39)||table_name
```

```
||chr(39),30,' ')||rpad(' '||chr(39)||table_name||chr(39),30,' ')||' ');
```

```
from user_tables@BZ30_DBL
```

```
where table_name like 'WAR%' ;
```

```
SPOOL OFF
```

Rezultat izgleda ovako:

```
INSERT INTO BENZROOT.BENZ_3040_MIGR (SCHEMA40, TABLE40, TABLE30)
```

```
VALUES (' BENZccc','WAR_COST_DIST_DETAIL' , 'WAR_COST_DIST_DETAIL' );
```

```
INSERT INTO BENZROOT.BENZ_3040_MIGR (SCHEMA40, TABLE40, TABLE30)
```

```
VALUES (' BENZccc','WAR_DAMAGE_TEXT' , 'WAR_DAMAGE_TEXT' );
```

```
...
```

```
-----  
-
```

```
-- @DataMigration3040.sql
```

```
-- generates insert into / select from statements
```

```
-- the generated statements must be verified
```

```
-- The mapping oldTable->newTable is defined in TableMapping.sql
```

```
-- adapt the following assignments:
```

```
define DB_SOURCE=BZ30
```

```
define DB_TARGET=BENZ
```

```
define DB_COUNTRY_CODE=9999
```

```
define DB_INSTALL_SRC_PATH=.
```

```
CONNECT BENZROOT/BENZROOT@BENZ
```

```
DROP TABLE BENZROOT.BENZ_3040_MIGR;
```

Omega magazin



O M E G A

```
CREATE TABLE BENZROOT.BENZ_3040_MIGR (
```

```
    SCHEMA40 VARCHAR2(255),
```

```
    TABLE40 VARCHAR2(255),
```

```
    TABLE30 VARCHAR2(255)
```

```
);
```

```
DROP TABLE BENZROOT.BENZ_3040_STMT;
```

```
CREATE TABLE BENZROOT.BENZ_3040_STMT (
```

```
    TABLE40 VARCHAR2(255),
```

```
    STMT VARCHAR2(1048)
```

```
);
```

```
DROP DATABASE LINK BZ30_DBL;
```

```
CREATE DATABASE LINK BZ30_DBL CONNECT TO BENZADMIN
```

```
IDENTIFIED BY BENZADMIN USING 'BZ30' ;
```

```
@&DB_INSTALL_SRC_PATH.\TableMapping.sql
```

```
COMMIT;
```

```
CONNECT BENZ9999/BENZ9999@BENZ
```

```
INSERT INTO BENZROOT.BENZ_3040_STMT (TABLE40, STMT) (
```

```
    SELECT MAX(MIGR.TABLE40),
```

```
           DECODE(UTC1.COLUMN_ID, MIN(UTC2.COLUMN_ID), 'INSERT INTO BENZ9999.' ||
```

```
           UTC1.TABLE_NAME || ' (' &          ' ) || UTC1.COLUMN_NAME ||
```

```
           DECODE(UTC1.COLUMN_ID, MAX(UTC2.COLUMN_ID), ' )', )
```

```
FROM USER_TAB_COLUMNS@BZ30_DBL UTC1,
```

Omega magazin



O M E G A

```
USER_TAB_COLUMNS@BZ30_DBL UTC2,

BENZROOT.BENZ_3040_MIGR MIGR

WHERE UTC1.TABLE_NAME = MIGR.TABLE40

AND UTC1.TABLE_NAME = UTC2.TABLE_NAME

AND UTC1.COLUMN_NAME != 'COUNTRY_CODE'

AND UTC2.COLUMN_NAME != 'COUNTRY_CODE'

AND MIGR.SCHEMA40 = 'BENZcccc'

GROUP BY UTC1.TABLE_NAME, UTC1.COLUMN_ID, UTC1.COLUMN_NAME

);

INSERT INTO BENZROOT.BENZ_3040_STMT (TABLE40, STMT) (

SELECT MAX(MIGR.TABLE40),

DECODE(UTC1.COLUMN_ID, MIN(UTC2.COLUMN_ID), 'SELECT ' ; ' ' )

||UTC1.COLUMN_NAME||

DECODE(UTC1.COLUMN_ID, MAX(UTC2.COLUMN_ID), ' FROM '||UTC1.TABLE_NAME||

'@BZ30_DBL WHERE COUNTRY_CODE="||' 999' ' ;' ||

CHR(13)||CHR(10)||' COMMIT;' ,')

FROM USER_TAB_COLUMNS@BZ30_DBL UTC1,

USER_TAB_COLUMNS@BZ30_DBL UTC2,

BENZROOT.BENZ_3040_MIGR MIGR

WHERE UTC1.TABLE_NAME = MIGR.TABLE30

AND UTC1.TABLE_NAME = UTC2.TABLE_NAME

AND UTC1.COLUMN_NAME != 'COUNTRY_CODE'

AND UTC2.COLUMN_NAME != 'COUNTRY_CODE'

AND MIGR.SCHEMA40 = 'BENZcccc'

GROUP BY UTC1.TABLE_NAME, UTC1.COLUMN_ID, UTC1.COLUMN_NAME

);
```

Omega magazin



```
set verify off;

set trimspool on;

set termout off;

set echo off;

set heading off;

set feedback off;

set pagesize 0;

set linesize 1000;
```

```
SPOOL &DB_INSTALL_SRC_PATH.\MigrStatements.sql
```

```
SELECT STMT
FROM BENZROOT.BENZ_3040_STMT
ORDER BY TABLE40, ROWNUM;
```

```
SPOOL OFF
```

Krajnji rezultat u MigrStatements.sql izgleda ovako:

```
INSERT INTO BENZ9999.WAR_CLAIM_DECISION_S(CLAIM_DECISION_STATUS,
    CLAIM_DECISION_STATUS_TEXT,
    BENZ_CREATE_DATE,
    BENZ_UPDATE_DATE,
    STATUS_FLAG)
SELECT CLAIM_DECISION_STATUS,
    CLAIM_DECISION_STATUS_TEXT,
```

Omega magazin



O M E G A

BENZ_CREATE_DATE,

BENZ_UPDATE_DATE,

STATUS_FLAG FROM WAR_CLAIM_DECISION_S@BZ30_DBL WHERE COUNTRY_CODE = '9999' ;

COMMIT;

INSERT INTO BENZ9999.WAR_CLAIM_ERR(CREDIT_RUN_NO,

CREDIT_RUN_NO_CALC,

DEALER_CODE,

CLAIM_NO,

CLAIM_ERROR_SEQ_NO,

ERROR_CODE_CLE,

ERROR_TEXT_CLE,

ACTION_CLE,

LOAD_STATUS,

BENZ_CREATE_DATE,

BENZ_UPDATE_DATE)

SELECT CREDIT_RUN_NO,

CREDIT_RUN_NO_CALC,

DEALER_CODE,

CLAIM_NO,

CLAIM_ERROR_SEQ_NO,

ERROR_CODE_CLE,

ERROR_TEXT_CLE,

ACTION_CLE,

LOAD_STATUS,

BENZ_CREATE_DATE,

Omega magazin

Džordža Vašingtona 40, Beograd, Jugoslavija, tel/faks +381 11 30-30-666,
www.omegamagazin.com



O M E G A

```
BENZ_UPDATE_DATE FROM WAR_CLAIM_ERR@BZ30_DBL WHERE COUNTRY_CODE = '9999';
```

```
COMMIT;
```

```
INSERT INTO BENZ9999.WAR_CLAIM_EXC_RATE(CLAIM_NO,
```

```
    DAMAGE_SEQ_NO,
```

```
    CURR_ISO_CODE_CLM,
```

```
    EXCHANGE_RATE_CLM,
```

```
    REPAIR_DATE,
```

```
    COST_CARRIER_KIND_MNF,
```

```
    COST_CARRIER_CODE_MNF,
```

```
    LABOUR_RATE_MNF,
```

```
    BENZ_CREATE_DATE,
```

```
    BENZ_UPDATE_DATE,
```

```
    STATUS_FLAG)
```

```
SELECT CLAIM_NO,
```

```
    DAMAGE_SEQ_NO,
```

```
    CURR_ISO_CODE_CLM,
```

```
    EXCHANGE_RATE_CLM,
```

```
    REPAIR_DATE,
```

```
    COST_CARRIER_KIND_MNF,
```

```
    COST_CARRIER_CODE_MNF,
```

```
    LABOUR_RATE_MNF,
```

```
    BENZ_CREATE_DATE,
```

```
    BENZ_UPDATE_DATE,
```

```
    STATUS_FLAG FROM WAR_CLAIM_EXC_RATE@BZ30_DBL WHERE COUNTRY_CODE = ' 999' ;
```

Omega magazin

Džordža Vašingtona 40, Beograd, Jugoslavija, tel/faks +381 11 30-30-666,
www.omegamagazin.com



O M E G A
COMMIT;

...

Kako je nova verzija DW nadskup polja stare, po uspešnom migriranju podataka treba još dopuniti 'razliku'. Znači treba pripremiti odgovarajuće UPDATE skriptove koji će popuniti nova redodatna polja. Za "razliku" u novim agregiranim/summary tabelama je dovoljno samo prvo pokretanje loadera koji će ih iskalkulisati i inicijalno popuniti i . . . to je to. Posao je završen relativno bezbolno.

Omega magazin

Džordža Vašingtona 40, Beograd, Jugoslavija, tel/faks +381 11 30-30-666,
www.omegamagazin.com