Data Vault alapú adattárház - Fél óra alatt

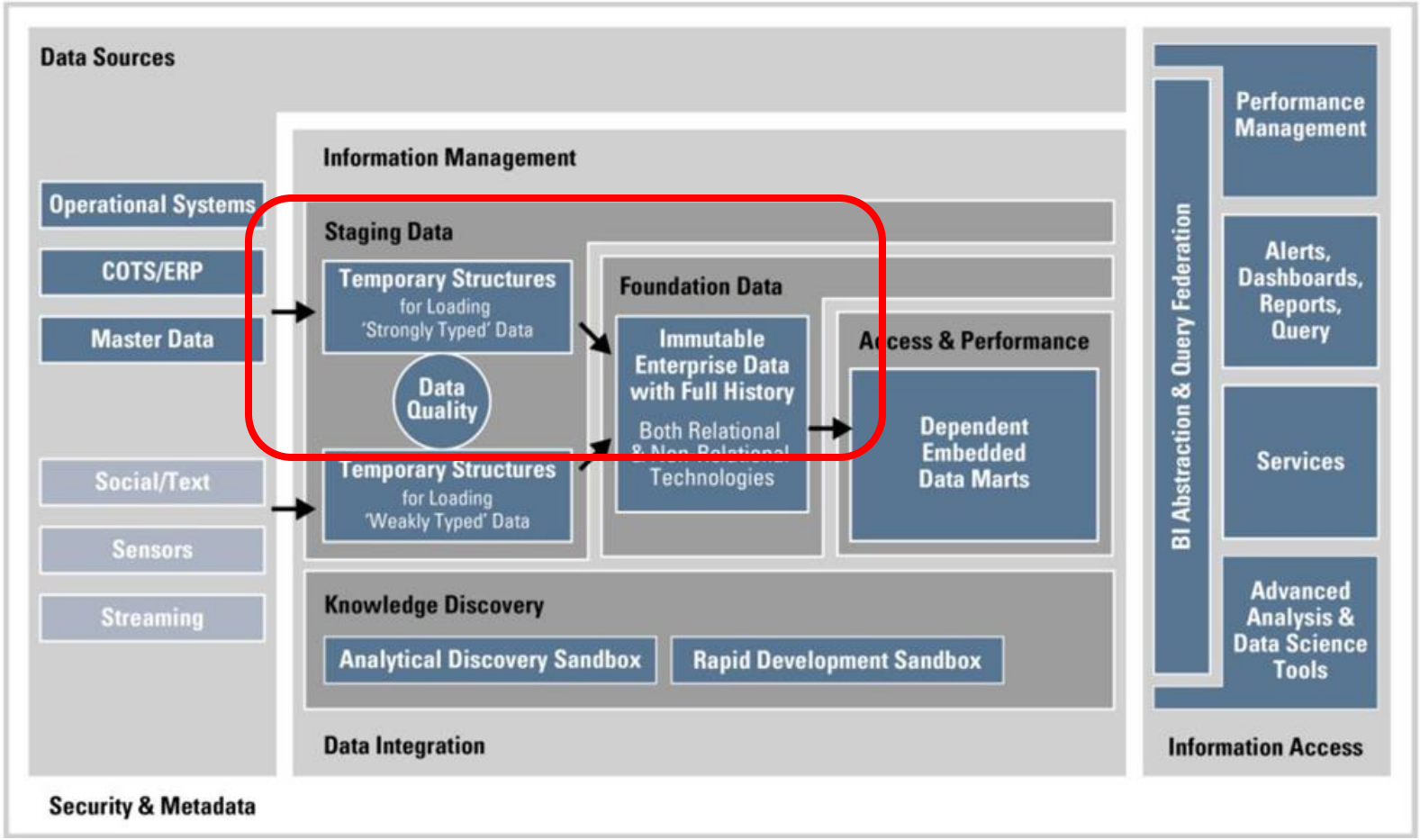
Döbrönte Zoltán

DMS Consulting Kft.

Miről lesz szó

- Adattárház automatizálás
  - Hol alkalmazható a leghatékonyabban
  - Célok, funkcionalitás, előnyök
- Data Vault modellezés a történetkezelt rétegben
- DW tervezés és fejlesztés automatizálás
  - Forrás elemzés és modell generálás
  - ETL generálás

A DW automatizálás helye az architektúrában
Oracle IM & Big Data Referencia Architektúra



Forrás: Oracle Information Management & Big Data Reference Architecture

Modell alapú DW tervezés/fejlesztés automatizálás

DW tervezés „automatizálás” célja:

- Az adattárházak tervezésének támogatása
- Az adattárház specifikus modellezési elemek kezelése
- ETL tervezés jobb támogatása

Fejlesztés automatizálás célja:

- DW fejlesztés gyorsítása
- Egyenletes, jó minőségű megoldás készítés támogatása
- Hatékony, jó minőségű ETL kód generálás

Funkciócsoportjai

- Forrásmodell elemzés
- DW modell és ETL generálás

A kialakított megoldás funkcionalitása

Adattárház modell generálás

- Data Vault módszertan szerinti modell
  (automatikus elemzés és tervezői beállítások alapján)
- Stage és DW szintű táblák
- Kulcsok (PK,FK,BK) és szekvenciák/identity oszlopok
- Alap metaadat oszlopok (auditálhatósághoz)
- Fizikai modell (Oracle, MSSQL) és névütközések feloldása

ETL elemek/eljárások generálása

- Mapping (tábla, oszlop szintű) információk (impact/lineage)
- ETL view-k és SQL eljárások
- Format/control fájlok (fájl alapú töltésekhez)
- DW futtató keretrendszerhez szükséges metaadatok

DW modell és ETL tervezés támogatás
Tervezői előnyök

Jobb minőségű, átfogó, részletes tervek készítése

- Adatbázis és ETL tervezés egy helyen
- Beépített minőség-ellenőrzés

Gyorsabb, hatékonyabb tervezés

- Kiemelt DW specifikus elemek
- DW best-practice választólisták

Részletes dokumentáció generálás

Testre szabható, egyszerűen bővíthető

DW modell és ETL tervezés támogatás Fejlesztői előnyök

Gyors fejlesztést tesz lehetővé

- 1-2 hét alatt működőképes pilot/1.verzió készíthető

Futtatható kód

- A generálás eredménye azonnal felhasználható

Egységes, hibamentes és jól strukturált kód

- Rövidebb a funkcionális tesztelés
- Kisebb idő- és erőforrásigény
- Könnyű továbbfejleszteni, módosítani a generált kódot

Teljeskörű DW modell és dokumentáció

- Hatás (impact) és lineage elemzés a teljes adatfolyamon

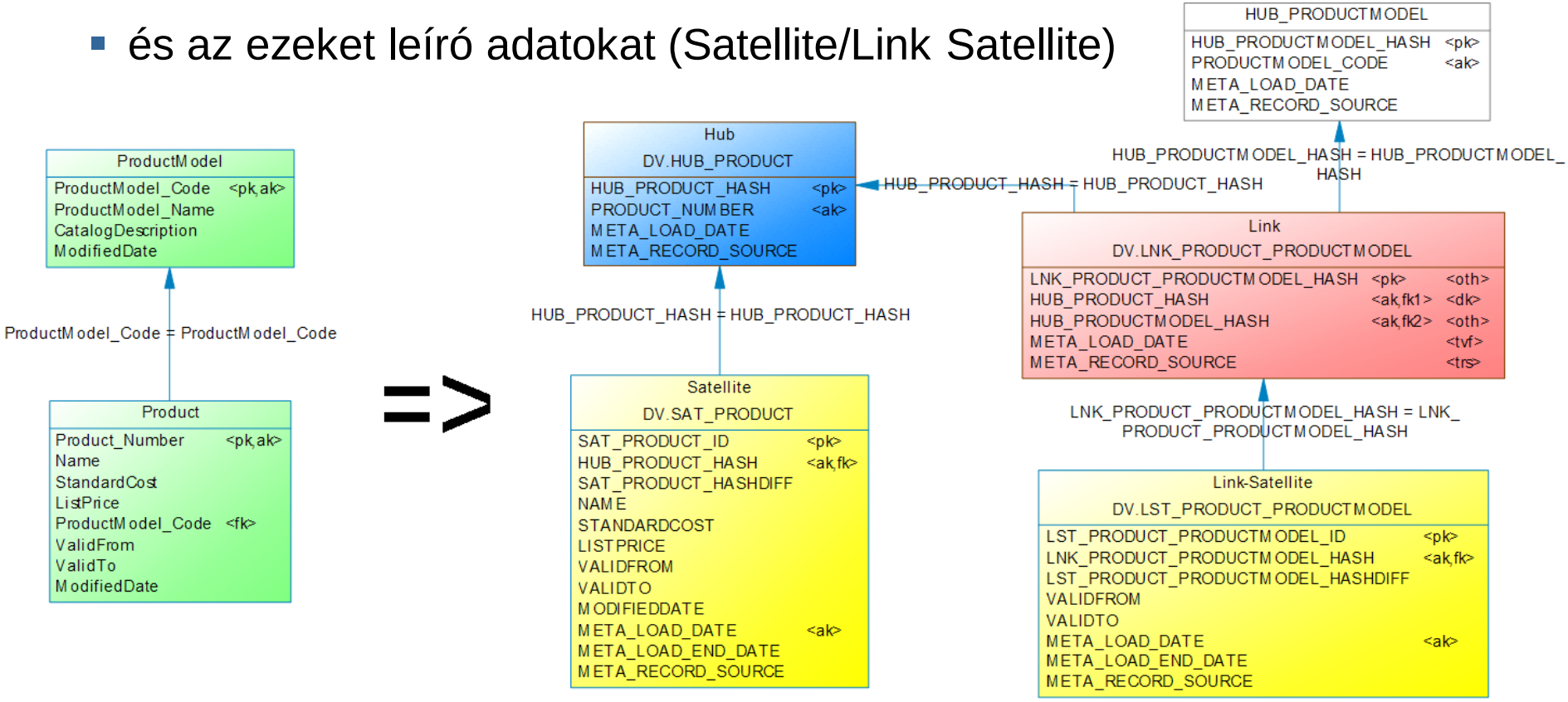
Miről lesz szó

- Adattárház automatizálás
  - Hol alkalmazható a leghatékonyabban
  - Célok, funkcionalitás, előnyök
- Data Vault modellezés a történetkezelt rétegben
- DW tervezés és fejlesztés automatizálás
  - Forrás elemzés és modell generálás
  - ETL generálás

Modell alapú adattárház generálás
Data Vault (DV) módszertan áttekintése

Lényege, hogy különválasztja

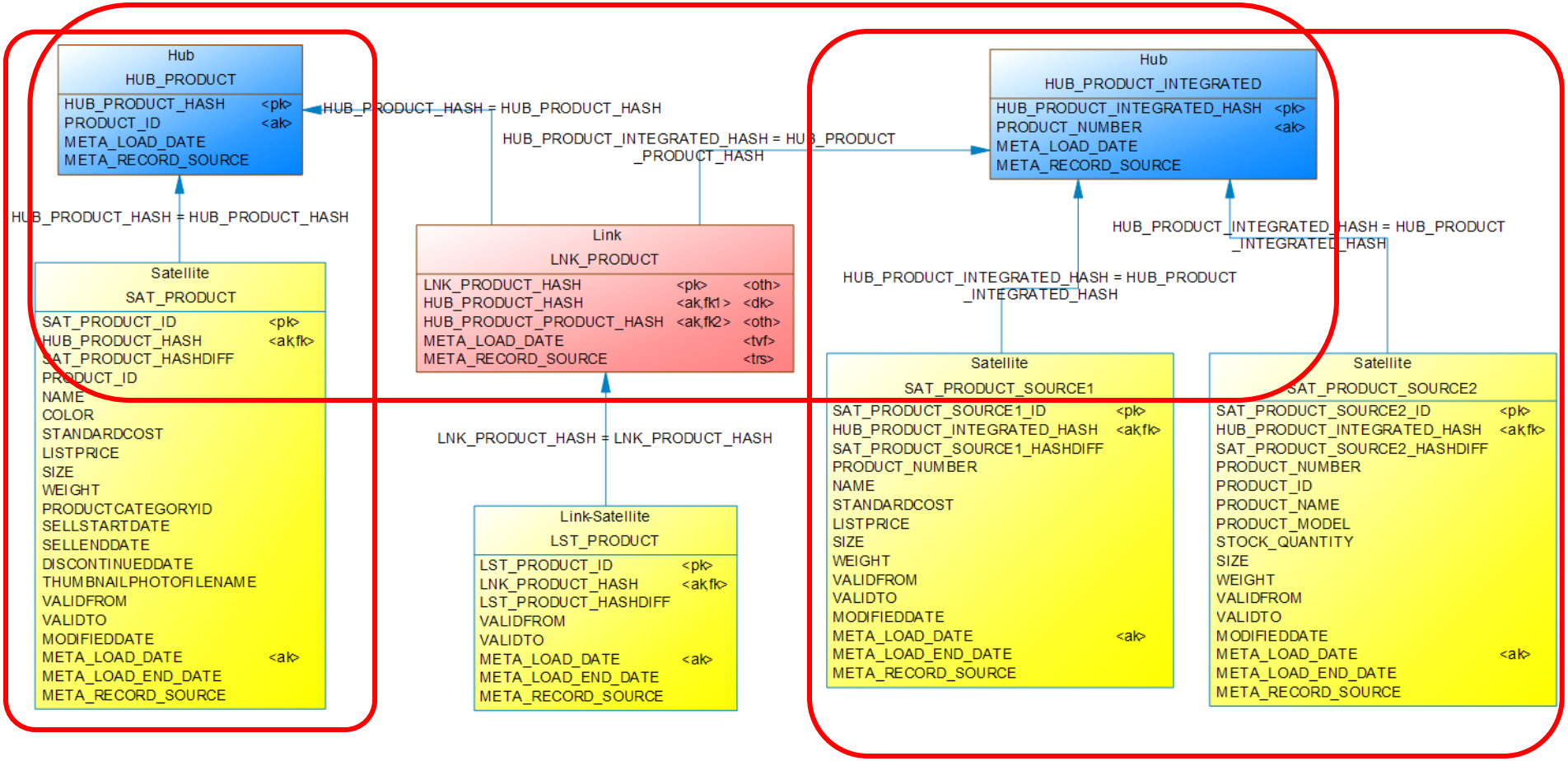
- az üzleti egyed azonosítókat (Hub)
- ezek kapcsolatait (Link)
- és az ezeket leíró adatokat (Satellite/Link Satellite)



Modell alapú adattárház generálás
DV módszertan előnyei

- Teljes történetiségkezelést biztosít, auditálható
- A folyamatosan változó üzleti szabályok, kalkulációk akár visszamenőleges futtatásához is biztosítja az alapadatokat
- Megvalósítja a különböző forrású adatok integrációját
- Alkalmas extrém nagy adatmennyiség kezelésére, nagyfokú párhuzamosíthatóság a betöltéseknél
- Agilis fejlesztést, követést tesz lehetővé
- Könnyen bővíthető újabb forrásrendszerekkel, adatkörökkel
- Egyszerű, tipizálható, generálható ETL eljárások
- Gyors implementáció

Data Vault – több forrás integrációja



Miről lesz szó

- Adattárház automatizálás
  - Hol alkalmazható a leghatékonyabban
  - Célok, funkcionalitás, előnyök
- Data Vault modellezés a történetkezelt rétegben
- DW tervezés és fejlesztés automatizálás
  - Forrás elemzés és modell generálás
  - ETL generálás

DV modell generálás
Lépések áttekintése

Interfészmodell összeállítása

- A forrásrendszerből kinyerendő adatelemek azonosítása => interfészspecifikáció => interfészmodell összeállítása

Kiinduló forrásmodell beállítása, paraméterezése

- DV stereotype-ok beállítása az interfészmodellen

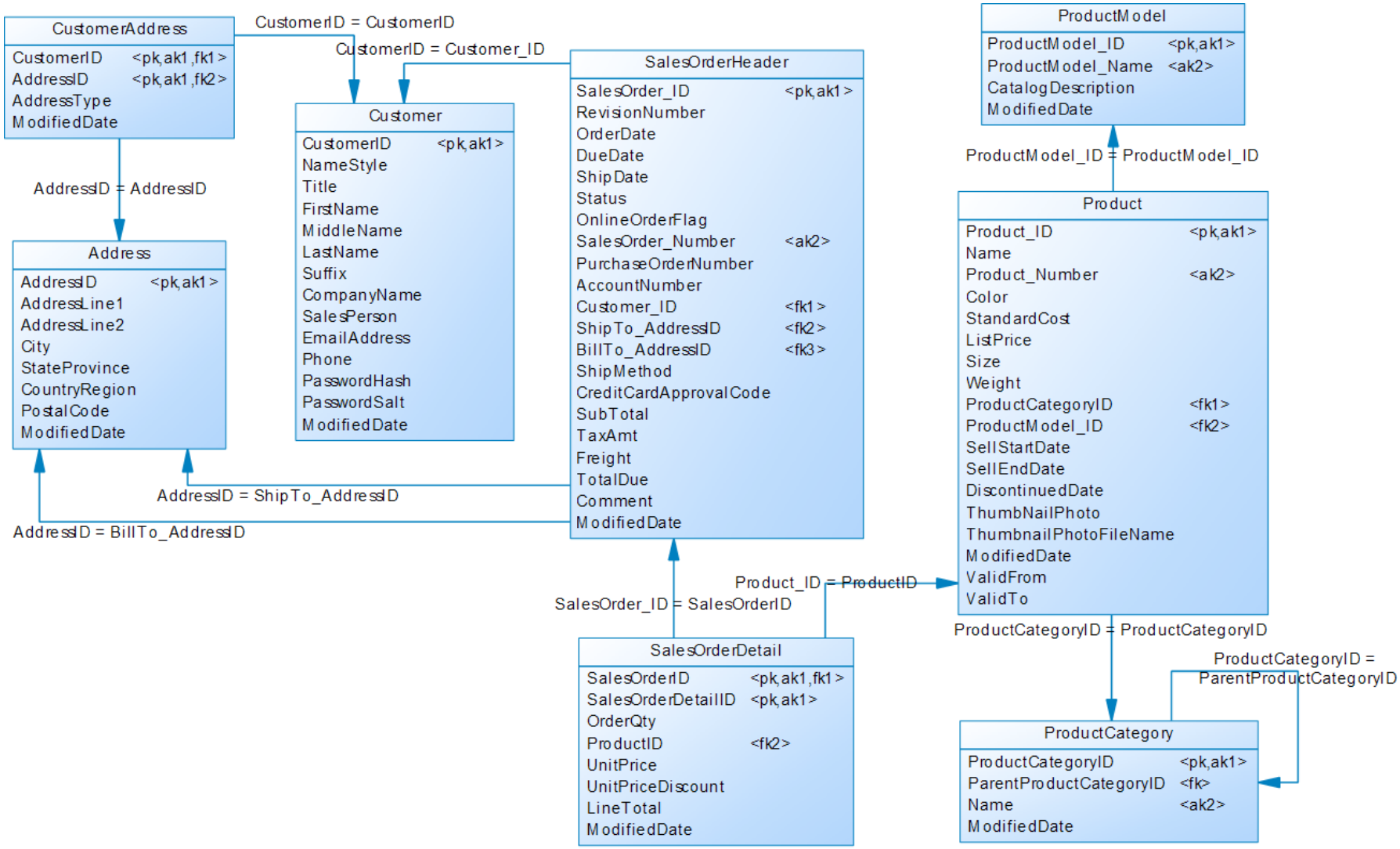
DW réteg modellek előállítása

- IFACE, LANDING, STAGE modellek
- DV modell

ETL generálás

- LANDING, STAGE, DV rétegek

DV modell generálás
Interfészmodell



DV modell generálás
Interfészmodell

Miket kell azonosítani a forrásrendszer modellen:

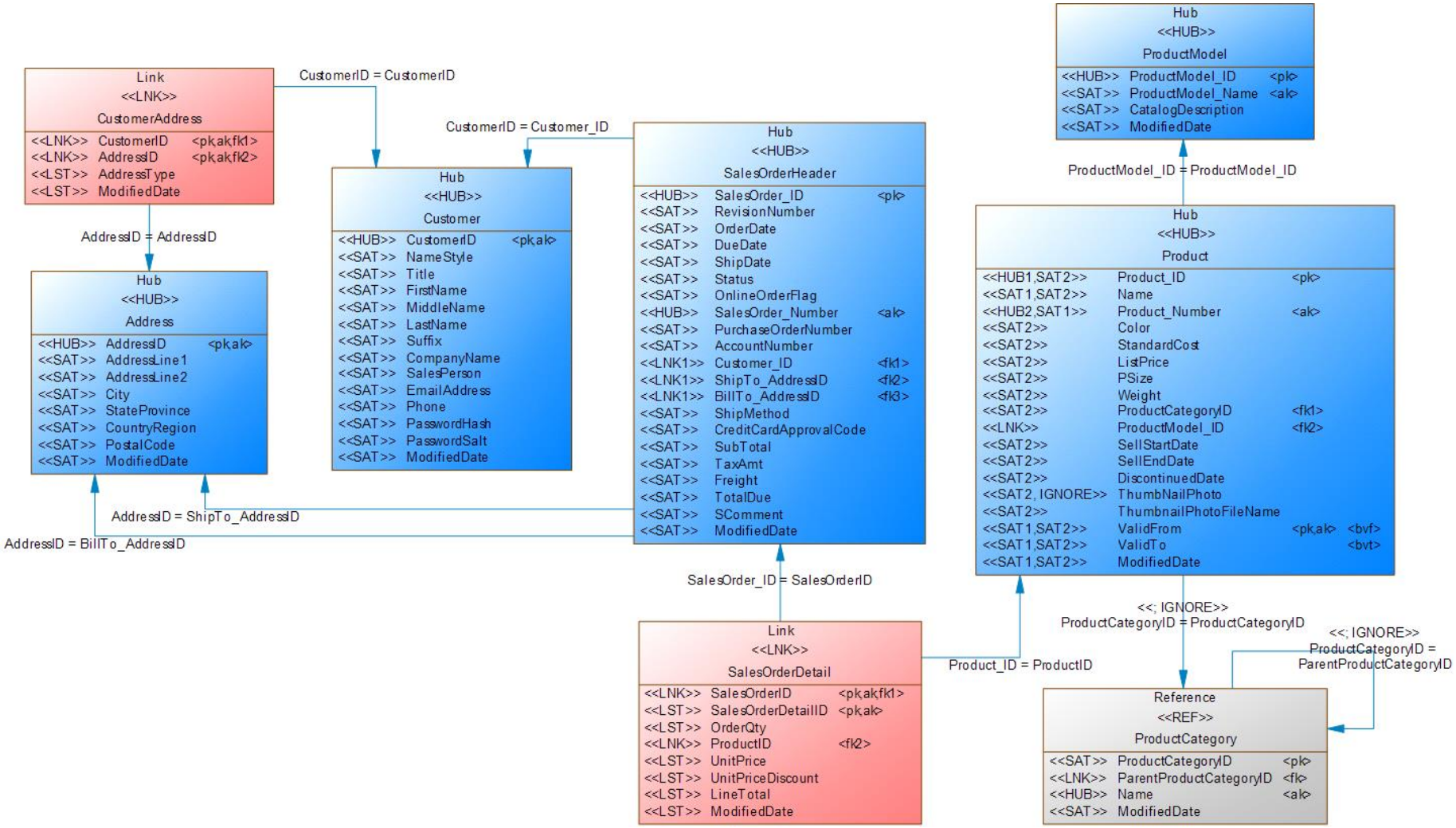
- PK – Primary Key, elsődleges kulcs
- BK – Business Key, üzleti kulcs
- FK – Foreign Key, idegen kulcs
- forrásrendszeri történetiségkövető attribútumok

Kiinduló forrásmodell előállítása: Manuális lépések

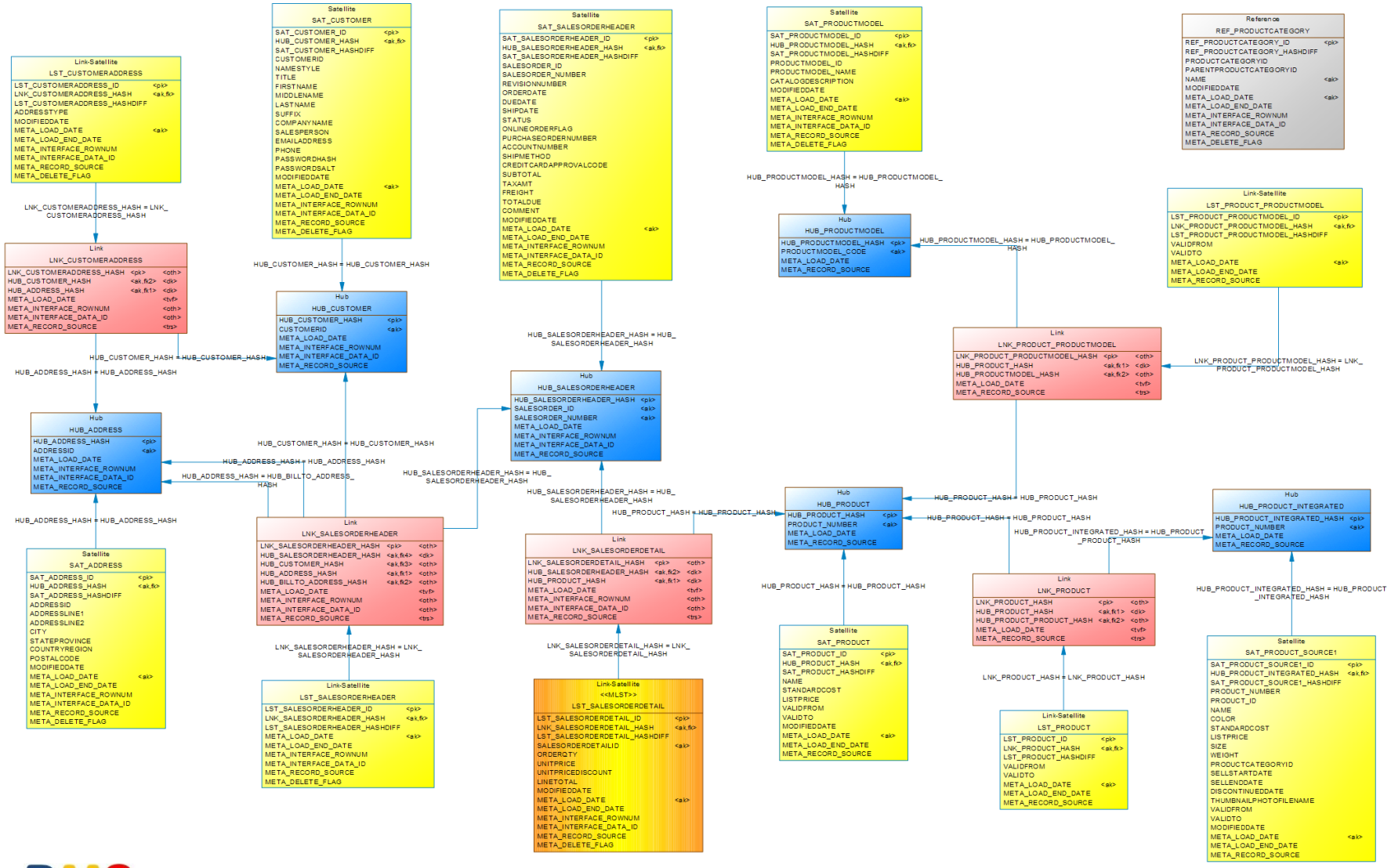
Manuális átállítások, hogy a kívánt DV modell álljon elő

- HUB több kulcs - integráció
  - hivatkozások használhatják a BK helyett a PK-t (pl. Product)
- SAT tervezés
  - forrásrendszer / frissítási gyakoriság / karakterisztika jelleg szerint
- LNK tervezés
  - melyik entitásból legyen HUB-ok és LNK-k (pl. SalesOrderDetail)
  - melyik kapcsolatok kerüljenek egy LNK-be, melyek külön (pl. SalesOrderHeader)
- REF táblák jelölése (pl. ProductCategory)
- Forrásrendszeri történetet kezelő attribútumok jelölése
  - a generált kód ősfeltöltő része fogja ezt felhasználni

DV modell generálás
Kiinduló forrásmodell előállítása



DV modell generálás
DW réteg modellek előállítása – DV



DV modell generálás
DW réteg modellek előállítása

Az eredmény:

- Tábla struktúrák
- PK & BK kulcsok, driving key, referenciák (FK)
- Mappingek

Mapping: stage-dv modell.LNK_SALESORDERHEADER.LSLSALES_GSLSALE0

	Column	Mapped to	Column Ma
→	HUB_ADDRESS_HASH	STAGE.ST_SAL_SALESORDERHEADER.HUB_ADDRESS_HASH	direct
2	HUB_BILLTO_ADDRESS_HASH	STAGE.ST_SAL_SALESORDERHEADER.HUB_BILLTO_ADDRESS_HASH	direct
3	HUB_CUSTOMER_HASH	STAGE.ST_SAL_SALESORDERHEADER.HUB_CUSTOMER_HASH	direct
4	HUB_SALESORDERHEADER_HASH	STAGE.ST_SAL_SALESORDERHEADER.HUB_SALESORDERHEADER_HASH	direct
5	LNK_SALESORDERHEADER_HASH	STAGE.ST_SAL_SALESORDERHEADER.LNK_SALESORDERHEADER_HASH	direct
6	META_INTERFACE_DATA_ID	STAGE.ST_SAL_SALESORDERHEADER.META_INTERFACE_DATA_ID	direct
7	META_INTERFACE_ROWNUM	STAGE.ST_SAL_SALESORDERHEADER.META_INTERFACE_ROWNUM	direct
8	META_LOAD_DATE	get_interface_datetime(interface_data)	function
9	META_RECORD_SOURCE	STAGE.ST_SAL_SALESORDERHEADER.META_RECORD_SOURCE	direct

Table sources | Mapped to | Base Column Mappings | Criteria | Select | Insert | Update | Delete

ETL generálás

Adatbázisobjektumok generálása

ETL eljárások generálása

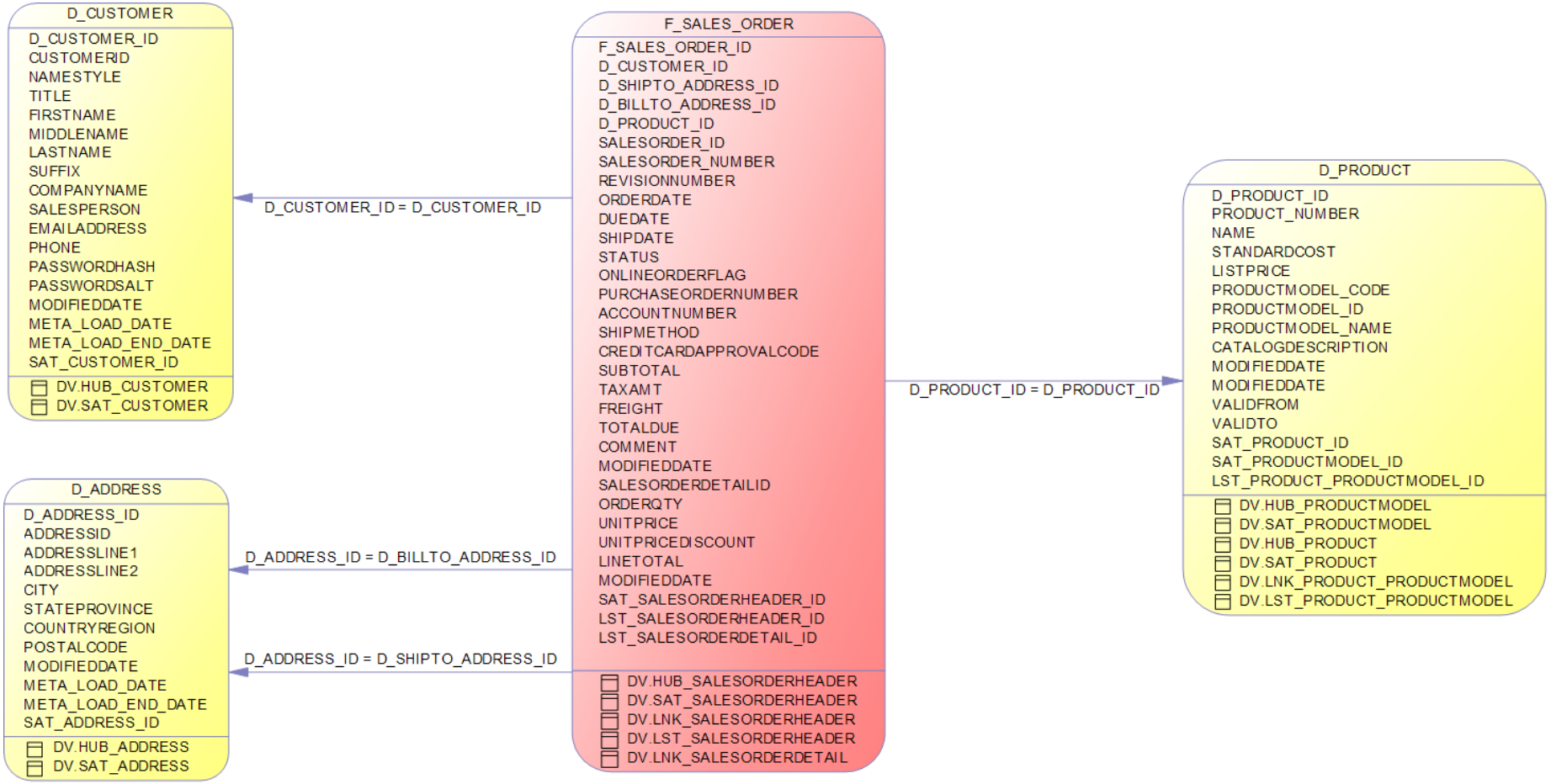
- Extraktáló eljárások
- Stage eljárások
- DV töltő eljárások
- ODS töltő eljárások

Generált ETL eljárás minta

```
CREATE OR REPLACE PACKAGE BODY STAGE_GDV_LOADTOSAT is

...

PROCEDURE PUS_SDVUGYFE_GDVSYNUG (p_target_system_code varchar2, p_target_instance_code varchar2, p_etl_program_
...

  begin
...
    create index I_TempVUS_SDVUGYFE_GDVSYNUG_historder on tVUS_SDVUGYFE_GDVSYNUG (HUB_UGYFEL_ID, UZL_ERV_KEZD)
...

    merge into DV.SAT_UGYFEL trg
    using (select s.SAT_UGYFEL_ID
                 ,nvl( i.interface_datetime, to_date('99991231','YYYYMMDD')) as meta_load_end_date
              from DV.SAT_UGYFEL s
                   inner join tVUS_SDVUGYFE_GDVSYNUG v
                      on v.HUB_UGYFEL_ID = s.HUB_UGYFEL_ID
                     and s.meta_load_end_date = to_date('99991231','YYYYMMDD')
                   inner join DW_META.interface_data i
                      on i.interface_data_id = v.meta_interface_data_id
          ) src
       on (    src.SAT_UGYFEL_ID = trg.SAT_UGYFEL_ID
          )
     when matched then
          update set trg.meta_load_end_date = src.meta_load_end_date;

     if ( SQL%ROWCOUNT > 0 ) then v_SCD2_Closed := SQL%ROWCOUNT; end if;

    insert /*+ APPEND NOLOGGING PARALLEL */ into DV.SAT_UGYFEL
           (SAT_UGYFEL_ID
           ,HUB_UGYFEL_ID
           ,UGYFEL_BK
           ,NEV
           ,SZULETESI_DATUM
           ,SZULETESI_HELY
           ,ADO_AZONOSITO
           ,TELEFONSZAM
           .MOBIL TELEFONSZAM
```

További lehetőségek

- Üzleti nézetek (view), lehet akár materializált is



További Oracle specifikus lehetőségek

A generálás eredménye:

- Tábladefiníciók kulcsokkal, szekvenciákkal
- PL/SQL kódok

Az eredmény emiatt:

- bármilyen Oracle Edition alatt használható

A rendelkezésre álló Oracle Edition függvényében:

- beállíthatók egyéb fizikai tárolási paraméterek (pl.: táblaterek, partíciók)
- RAC-on is futtatható további hozzányúlás nélkül
- Adatbázis parallel futtatási képessége is kihasználható

Összefoglalás
Data Vault modellezés és DW automatizálás

Alkalmazása:

- Klasszikus DW architektúrában elsősorban a történeti adattár rétegig
  - 99%-ban azonnal használható generált adatmodell és ETL kódok
  - A üzleti felhasználók által lekérdezhető DM réteghez ETL előkészíthető

Előnyei:

- Agilis és rugalmas, a Data Vault miatt egy üzleti igény változása a lehető legkisebb módosítást vonja maga után
- Gyors reakció az üzleti igények kielégítésére
- Nagy teljesítmény, a magas fokú párhuzamosíthatóság miatt
- Jó minőségű, egységes kódok a template-ek és generálás miatt
- A kódok a rendelkezésre álló bármely Oracle Edition alatt használhatók, a választott Edition előnyeinek kihasználása mellett

Kérdések & válaszok

zoltan.dobronte@datagovernance.hu

Bónusz
Oracle DB Standard Edition (SE) és DW

Adattárházhoz is használható az SE, de nincs:

- Parallel query
- Tömörítés
- Particionálás
- Bitmap index
- Mat.view query rewrite
- HeatMap
- Virtual Private Database (VPD) és Fine-grained Auditing (FGA)
- stb.

Bónusz
Oracle DB Standard Edition 2

Szeptember elejétől van a DB 12.1.0.2 verzióval Standard Edition Two (SE2)

A Standard Edition és Standard Edition 1 (SE/SE1) decembertől nem lesz és support/patch-ek is csak 2016. aug-ig

Változtak a licenc feltételek:

- az SE2 ára az SE árával egyező (drágább mint SE1)
- minimum named user szám emelkedett:
  5-ről (SE/SE1) 10-re (SE2 és ez szerverenként)
- max. 2 node (SE: 4node) és csak 1 foglalat/node
- RAC benne marad, de max. 2 foglalat (azaz pl. 2 node x 1 foglalat)
- csökkentett párhuzamos futás: max. 16 CPU thread (ha RAC, akkor 2x8)
- egy jó összefoglaló ábra:
  http://www.xynomix.com/oracle-standard-edition-2-what-you-need-know