|  |
| --- |
|  |
|  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| REVISION HISTORY | | | | | |
| Ver. | Description of Change | Author | Date | Approved | |
| Name | Effective Date |
| 1.0 | Initial status | [Kiryl Bucha](mailto:Kiryl_Bucha@epam.com) | 12-JAN-2012 |  |  |
| 2.0 | Updated in accordance with renewed content | [Elias Nema](mailto:Elias_Nema@epam.com) | 20-JAN-2014 |  |  |
| 3.0 | Report | Mikita Tur | 10-DEC-2017 |  |  |

Contents

[1. Create Packages for Reload Dimension from DW to SAL 3](#_Toc384725063)

[2. Test Your Data 3](#_Toc384725064)

[3. Task Results 3](#_Toc384725065)

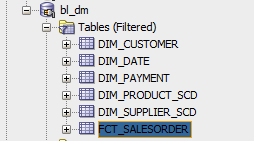
# Create packages to load dimensions from ST (staging) to SL (star layer)

The Main Task is to create independent packages to reload dimensions according to your DWH solution concept that was developed on Module 6 Introduction to DWH.

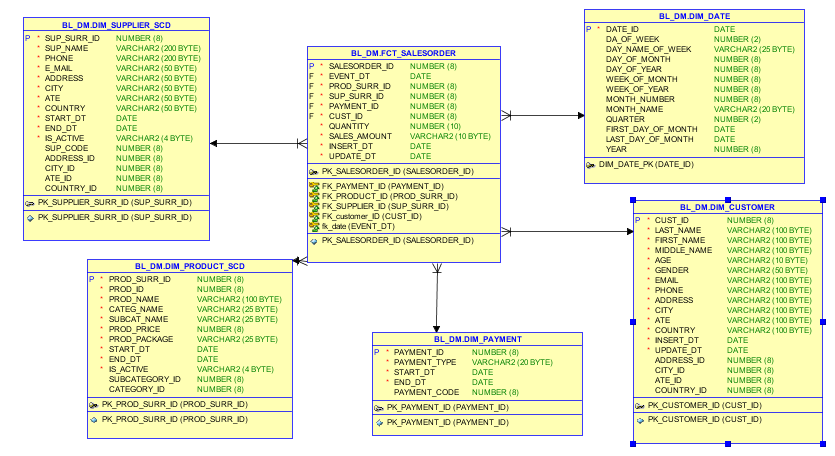
Required points:

* Create all required dim objects on SL Layer
* Grant all required privileges to SL\_CL (Cleansing Layer)
* Create packages to load dim data (one package = one dimension) on SL\_CL
* Use Bulk Collect (One or more procedures)
* Use Forall (One or more procedures)
* Use a cursor FOR loop
* Use EXECUTE IMMEDIATE INTO for dynamic queries
* Use cursor variables

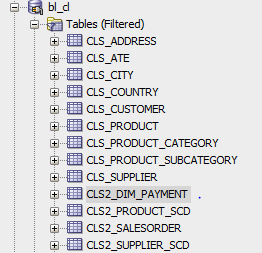
All required tables were created. The scripts are in the project folder.



STAR Shema:



Tables which are loaded from bl\_3nf to bl\_dm firstly move to bl\_cl and names BL\_CL.CLS2\_//:



Package to insert data from bl\_cl.cls2.. to bl\_dm

CREATE OR REPLACE PACKAGE pkg\_etl\_insert\_products

AUTHID CURRENT\_USER

AS

PROCEDURE insert\_table\_products;

PROCEDURE merge\_table\_products;

END pkg\_etl\_insert\_products;

CREATE OR REPLACE PACKAGE BODY pkg\_etl\_insert\_products

AS

---------------------------------------------------

PROCEDURE insert\_table\_products

IS

BEGIN

EXECUTE IMMEDIATE ('TRUNCATE TABLE cls2\_product\_scd');

insert into CLS2\_PRODUCT\_SCD(PROD\_SURR\_ID,

PROD\_id ,

PROD\_NAME ,

CATEG\_NAME ,

SUBCAT\_NAME ,

PROD\_PRICE ,

prod\_package ,

START\_DT ,

END\_DT ,

IS\_ACTIVE )

select cp.product\_id as prod\_surr\_ID,cp.PRODUCT\_srcID as prod\_id,

cp.product\_name,

ct.category\_name,

sc.subcategory\_name,

cp.product\_price,

cp.product\_package,

cp.start\_dt,

cp.end\_dt,

cp.is\_active

from BL\_3NF.CE\_PRODUCT cp left join( BL\_3NF.CE\_PRODUCT\_SUBCATEGORY

sc left join BL\_3NF.CE\_PRODUCT\_CATEGORY ct on sc.category\_srcid=ct.category\_srcid ) on cp.SUBCATEGORY\_SRCID=sc.SUBCATEGORY\_SRCID;

END;

COMMIT;

EXCEPTION

WHEN OTHERS THEN

RAISE;

END insert\_table\_products;

---------------------------------------------------

PROCEDURE merge\_table\_products

IS

BEGIN

MERGE INTO bl\_dm.dim\_product\_scd tgt USING

(SELECT prod\_surr\_id,

PROD\_NAME ,

CATEG\_NAME ,

SUBCAT\_NAME ,

PROD\_PRICE ,

prod\_package ,

START\_DT ,

END\_DT ,

IS\_ACTIVE

FROM CLS2\_PRODUCT\_SCD

MINUS

SELECT PROD\_id ,

PROD\_NAME ,

CATEG\_NAME ,

SUBCAT\_NAME ,

PROD\_PRICE ,

prod\_package ,

START\_DT ,

END\_DT ,

IS\_ACTIVE

FROM bl\_dm.dim\_product\_scd tgt

) src ON (tgt.prod\_name=src.PROD\_NAME AND tgt.prod\_price=src.prod\_price AND tgt.prod\_package=src.prod\_package)

WHEN matched THEN

UPDATE

SET tgt.prod\_id = src.prod\_surr\_id,

tgt.END\_DT =src.end\_dt,

tgt.IS\_ACTIVE =src.is\_active WHEN NOT matched THEN

INSERT

(

prod\_surr\_id,

PROD\_id ,

PROD\_NAME ,

CATEG\_NAME ,

SUBCAT\_NAME ,

PROD\_PRICE ,

prod\_package ,

START\_DT ,

END\_DT ,

IS\_ACTIVE

)

VALUES

(

bl\_dm.dim\_prod\_seq.NEXTVAL,

src.prod\_surr\_id,

src.PROD\_NAME ,

src.CATEG\_NAME ,

src.SUBCAT\_NAME ,

src.PROD\_PRICE ,

src.prod\_package ,

src.START\_DT ,

src.END\_DT ,

src.IS\_ACTIVE

) ;

COMMIT;

EXCEPTION

WHEN OTHERS THEN

RAISE;

END merge\_table\_products;

---------------------------------------------------

END pkg\_etl\_insert\_products;