|  |
| --- |
| EPAM Systems, RD Dep. |
| MTN.BI.08 PL/SQL Techniques |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 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 | Aksana Kuratnik | 07-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:

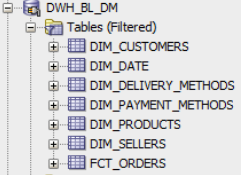
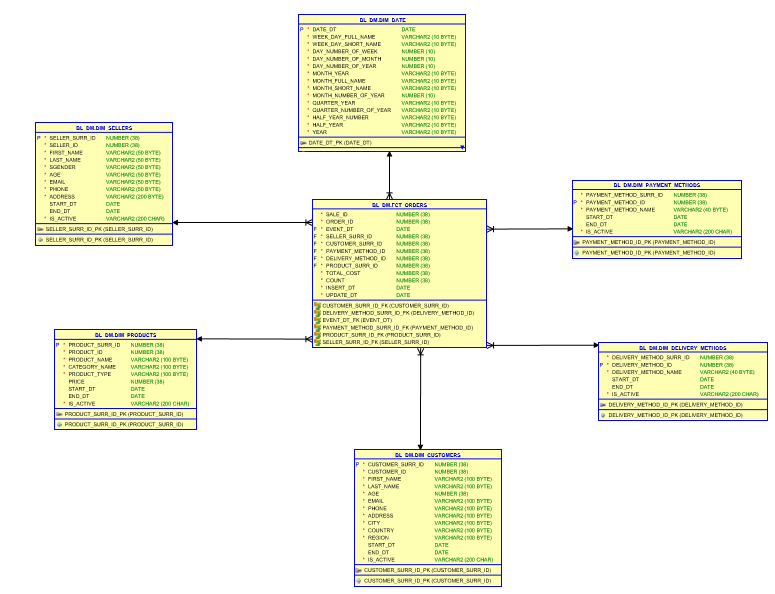
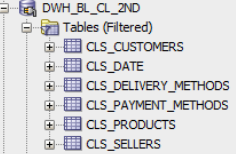
* Create all required dim objects on SL Layer
* Grant all required privileges to SL\_CL (Cleansing Layer)
* Create packages to load dim data (one package = one dimension) on SL\_CL
* Use Bulk Collect (One or more procedures)
* Use Forall (One or more procedures)
* Use a cursor FOR loop
* Use EXECUTE IMMEDIATE INTO for dynamic queries
* Use cursor variables
* All required tables were created. The scripts are in the project folder.
* 
* STAR Shema:
* 

Table on 2ND BL\_CL layer were created and data was load:



Package to insert data from bl\_cl\_2nd to bl\_dm

CREATE OR REPLACE PACKAGE pkg\_etl\_insert\_payment\_methods

AUTHID CURRENT\_USER

AS

PROCEDURE merge\_dim\_payment\_methods;

END pkg\_etl\_insert\_payment\_methods;

/

CREATE OR REPLACE PACKAGE BODY pkg\_etl\_insert\_payment\_methods

AS

PROCEDURE merge\_dimpayment\_methods

IS

BEGIN

MERGE INTO bl\_dm.dim\_payment\_methods\_scd t USING

( SELECT payment\_method\_surr\_id,

payment\_method\_desc,

start\_dt,

end\_dt,

is\_active

FROM cls\_payment\_methods\_scd

MINUS

SELECT payment\_method\_id,

payment\_method\_desc,

start\_dt,

end\_dt,

is\_active

FROM bl\_dm.dim\_payment\_methods\_scd

) c ON ( t.payment\_method\_desc = c.payment\_method\_desc

AND t.start\_dt = c.start\_dt)

WHEN matched THEN

UPDATE SET

t.payment\_method\_id = c.payment\_method\_surr\_id,

t.end\_dt = c.end\_dt,

t.is\_active = c.is\_active

WHEN NOT matched THEN

INSERT

(

payment\_method\_surr\_id,

payment\_method\_id,

payment\_method\_desc,

start\_dt,

end\_dt,

is\_active

)

VALUES

(

bl\_dm.dim\_payment\_methods\_seq.NEXTVAL,

c.payment\_method\_surr\_id,

c.payment\_method\_desc,

c.start\_dt,

c.end\_dt,

c.is\_active

) ;

COMMIT;

EXCEPTION

WHEN OTHERS THEN

RAISE;

END merge\_dim\_payment\_methods;

END pkg\_etl\_insert\_payment\_methods;

/

