|  |
| --- |
| EPAM Systems, RD Dep. |
| MTN.BI.07 <Title> |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 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 package to load the Fact Table from the staging (BL\_3NF) to BL\_DM (star) layer 3](#_Toc499497950)

[2. Task Results 3](#_Toc499497951)

# Create package to load the Fact Table from the staging (BL\_3NF) to BL\_DM (star) layer

The Main Task is to create independent package[s] to load your fact table from BL\_3NF to BL\_DM layer, according to your DWH concept that was developed on Module 6 Introduction to DWH.

# Task Results

Create required objects:

* Create all required fact table objects on BL\_DM Layer (FCT\_...).
* Grant all required privileges to BL\_CL (Cleansing Layer).
* Create packages to load fact data (one package = one fact table) on BL\_CL.
* Prepare document with description of your load and screenshot of data in the Fact Table.
* Test Procedure for Repeatable execution (it should produce same consistent results each time on a same source data).

**PACKAGE TO LOAD THE FACT TABLE FROM BL\_3NF TO BL\_DM (bl\_3nf 🡪bl\_cl\_2nd🡪bl\_dm)**

**bl\_3nf 🡪bl\_cl\_2nd**

CREATE OR REPLACE PACKAGE pkg\_etl\_insert\_cls\_orders

AUTHID CURRENT\_USER

AS

PROCEDURE insert\_cls\_orders;

END pkg\_etl\_insert\_cls\_orders;

/

CREATE OR REPLACE PACKAGE BODY pkg\_etl\_insert\_cls\_orders

AS

PROCEDURE insert\_cls\_orders

IS

BEGIN

EXECUTE IMMEDIATE ('TRUNCATE TABLE cls\_fct\_orders');

INSERT INTO cls\_fct\_retail\_sales\_dd

SELECT order\_id,

cr.order\_dt AS event\_dt,

ce.seller\_id AS seller\_surr\_id,

cm.customer\_id AS customer\_surr\_id,

ct.store\_id AS store\_id,

cr.payment\_method\_id AS payment\_method\_surr\_id,

cr.delivery\_method\_id AS delivery\_method\_surr\_id,

cr.product\_info\_id AS product\_info\_surr\_id,

cr.order\_sum\_usd AS tot\_sale\_sum,

ROUND ( dbms\_random.value( 100, 99999), 2) AS tot\_sale\_amount,

cr.insert\_dt AS insert\_dt,

SYSDATE AS update\_dt

FROM bl\_3nf.ce\_orders cr left join bl\_3nf.ce\_sellers ce

on cr.seller\_id = ce.seller\_id

left join bl\_3nf.ce\_customers cm

on cr.customer\_id = cm.customer\_id

left join bl\_3nf.ce\_stores ct

on cr.store\_id = ct.store\_id

left join bl\_3nf.ce\_product\_info cpd

on cr.product\_info\_id = cpd.product\_info\_id;

COMMIT;

EXCEPTION

WHEN OTHERS THEN

RAISE;

END insert\_cls\_orders;

END pkg\_etl\_insert\_cls\_orders;

/

**bl\_cl\_2nd🡪bl\_dm**

CREATE OR REPLACE PACKAGE pkg\_etl\_insert\_fct\_orders

AUTHID CURRENT\_USER

AS

PROCEDURE insert\_fct\_orders;

END pkg\_etl\_insert\_fct\_orders;

/

CREATE OR REPLACE PACKAGE BODY pkg\_etl\_insert\_fct\_orders

AS

PROCEDURE insert\_fct\_orders

IS

BEGIN

INSERT INTO

bl\_dm.fct\_orders

SELECT

bl\_dm.fct\_orders\_seq.NEXTVAL AS sale\_id,

a.order\_id,

a.order\_dt,

c.seller\_surr\_id,

b.customer\_surr\_id,

e.payment\_method\_id,

g.delivery\_method\_id,

f.product\_surr\_id,

a.sum\_of\_payment,

a.insert\_dt,

a.update\_dt

FROM cls\_fct\_orders a INNER JOIN bl\_dm.dim\_customers b ON a.customer\_surr\_id = b.customer\_id

INNER JOIN bl\_dm.dim\_sellers\_ c ON a.seller\_surr\_id = c.seller\_id

INNER JOIN bl\_dm.dim\_payment\_methods e ON a.payment\_method\_surr\_id = e.payment\_method\_id

INNER JOIN bl\_dm.dim\_products f ON a.product\_surr\_id = f.product\_id

INNER JOIN bl\_dm.dim\_delivery\_methods g ON a.delivery\_method\_surr\_id = g.delivery\_method\_id;

COMMIT;

EXCEPTION

WHEN OTHERS THEN

RAISE;

END insert\_fct\_orders;

END pkg\_etl\_insert\_fct\_orders;

/