|  |
| --- |
| EPAM Systems, RD Dep. |
| MTN.BI.08 Transportation |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 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 on task | Vitaliya Adamchuk | 02-DEC-2017 |  |  |

Contents

[1. ETL Transportation - Analytical 3](#_Toc384724511)

[1.1. Transportation Description 3](#_Toc384724512)

[2. ETL Extraction – Example of Loading FCT\_\* 3](#_Toc384724513)

[2.1. Prepare Table of Facts to SAL Layer 3](#_Toc384724514)

[3. Transportable Tablespaces and DBLinks 3](#_Toc384724515)

[4. Hierarchies 3](#_Toc384724516)

# Transportable Tablespaces and DB Links

Create sample usage of transportable tablespaces and database links and describe it in document with screenshots and description.

# Table Functions

Replace at least one cleansing (CLS) table with Oracle Table function.

Table functions are used to return PL/SQL collections that mimic tables. They can be queried like a regular table by using the TABLE operator in the FROM clause. Regular table functions require collections to be fully populated before they are returned. Since collections are held in memory, this can be a problem as large collections can waste a lot of memory and take a long time to return the first row. These potential bottlenecks make regular table functions unsuitable for large Extraction Transformation Load (ETL) operations.

--Creating object:

CREATE TYPE t\_tf\_row AS OBJECT (

id NUMBER,

description VARCHAR2(50)

);

/

CREATE TYPE t\_tf\_tab IS TABLE OF t\_tf\_row;

/

-- Build the table function itself.

CREATE OR REPLACE FUNCTION get\_tab\_tf (p\_rows IN NUMBER) RETURN t\_tf\_tab AS

l\_tab t\_tf\_tab := t\_tf\_tab();

BEGIN

FOR i IN 1 .. p\_rows LOOP

l\_tab.extend;

l\_tab(l\_tab.last) := t\_tf\_row(i, 'Description for ' || i);

END LOOP;

RETURN l\_tab;

END;

/

-- Test it.

SELECT \*

FROM TABLE(get\_tab\_tf(10))

ORDER BY id DESC;

