|  |
| --- |
|  |
|  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 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 the task | Aksana Kuratnik | 07-DEC-2017 |  |  |

Contents

[1. Materialized Views- Basic 3](#_Toc384725184)

[1.1. Create Materialized Views - ON DEMAND 3](#_Toc384725185)

[1.2. Create Materialized Views - ON COMMIT 3](#_Toc384725186)

[2. Fact Table Loading 3](#_Toc384725187)

[3. Task Results 3](#_Toc384725188)

# Materialized Views- Basic

## Create Materialized Views - ON DEMAND

Take any "report" SQL from Lab Work 02 and create Materialized View based on it. Refresh type is ON DEMAND.

* Use Standard CREATE MATERIALIZED VIEW CLAUSE.
* Use DBMS\_MVIEW package to refresh Mat View.

## Create Materialized Views - ON COMMIT

Take any "report" SQL from Lab Work 02 and create Materialized View based on it. Refresh type is ON COMMIT.

* Use Standard CREATE MATERIALIZED VIEW CLAUSE.
* Tests ON COMMIT REFRESH.

# Fact Table Loading

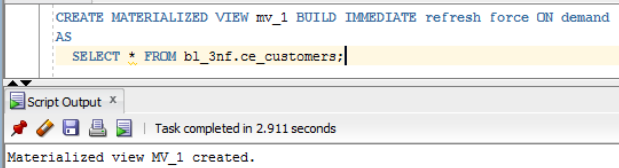
Create packages to load fact table to the presentation layer.

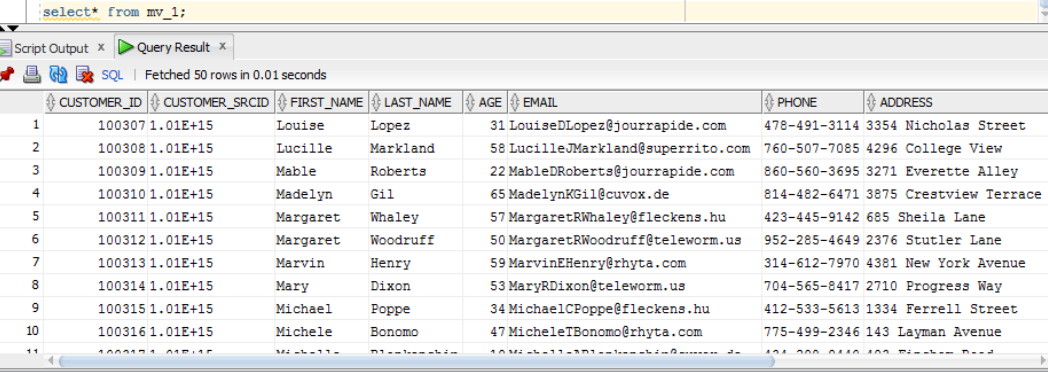
# Task Results

Create required objects:

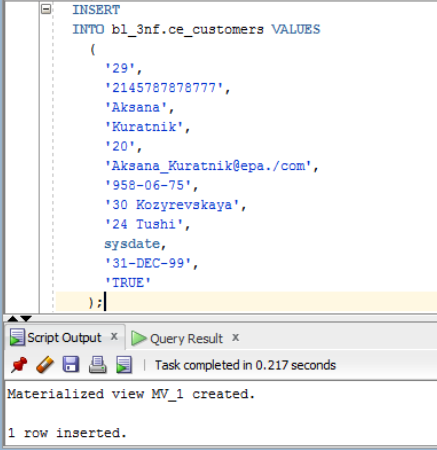
* Put objects script to Git.
* Prepare Document with Screenshot of Tests Data.
* Prepare Document with Screenshot of Refreshing Mat View script.

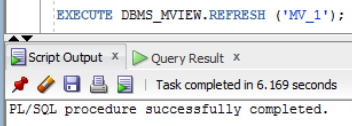
**Creation of materialized view on demand:**



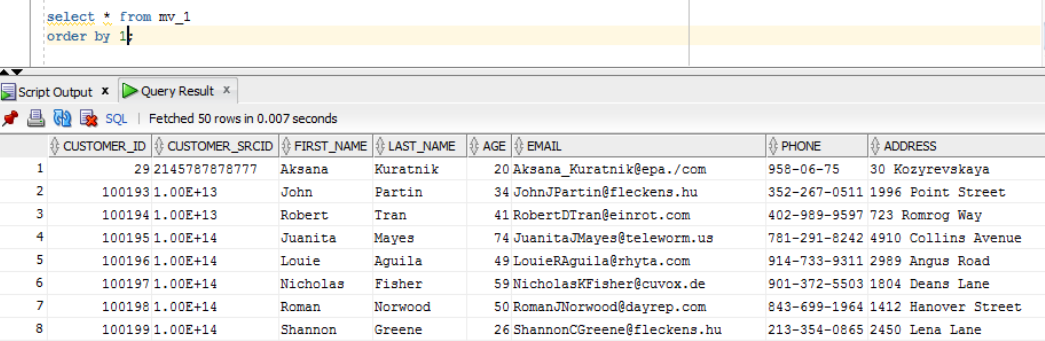


Then table was updated





The result of select from materialized view after update:



Creating MV on commit:

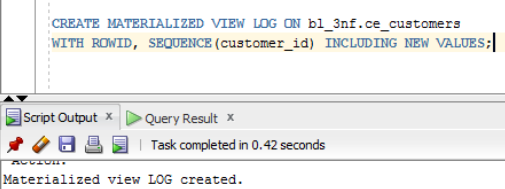
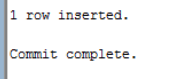


Table was updated:





The result:

