|  |
| --- |
| EPAM Systems, RD Dep. |
| MTN.BI.08 Materialized Views |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| REVISION HISTORY | | | | | |
| Ver. | Description of Change | Author | Date | Approved | |
| Name | Effective Date |
| 1.0 | Initial status | [Hanna\_Hul](mailto:Kiryl_Bucha@epam.com) | 12-JAN-2012 |  |  |

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)

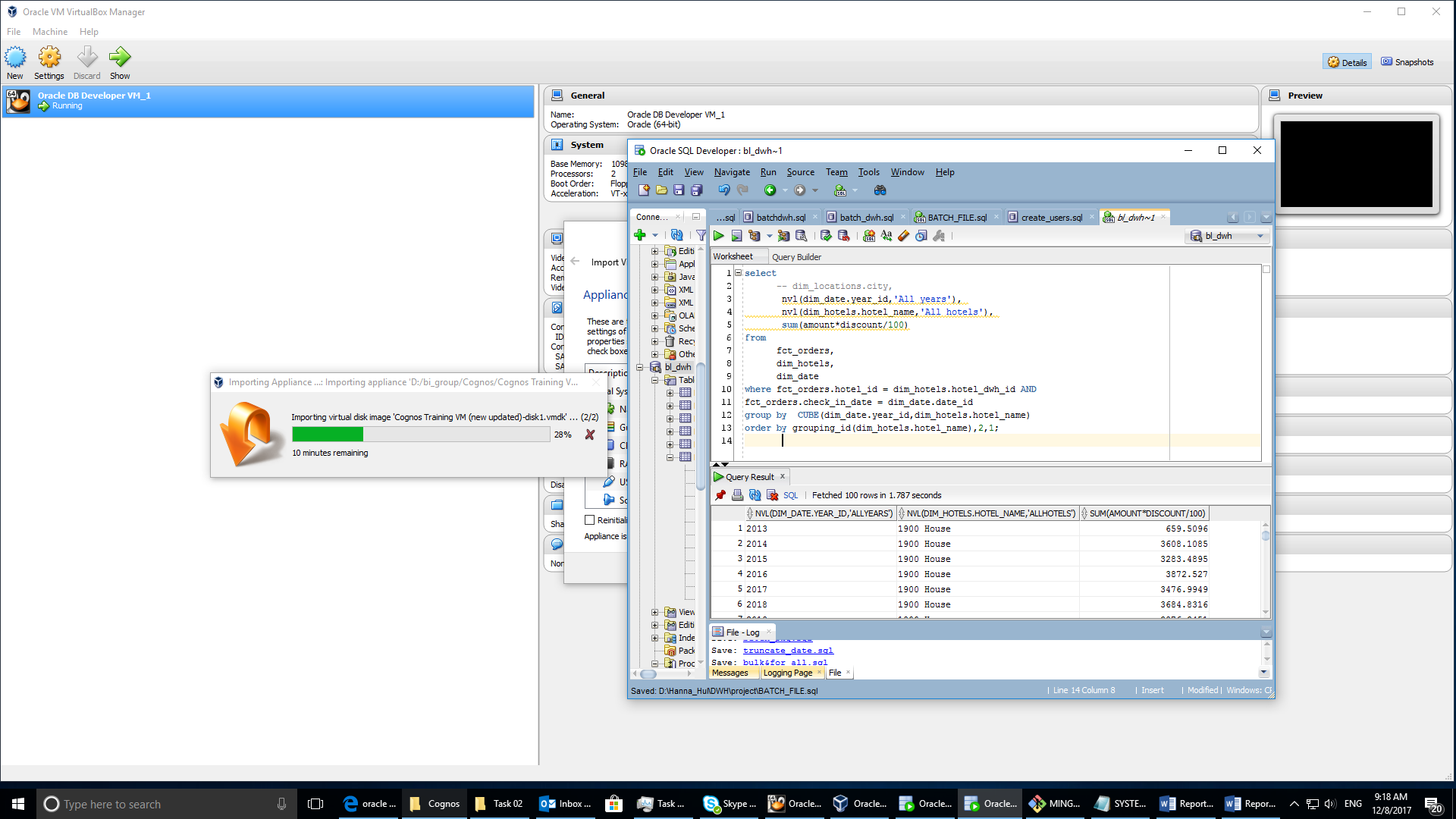
[3. Task Results 3](#_Toc384725188)

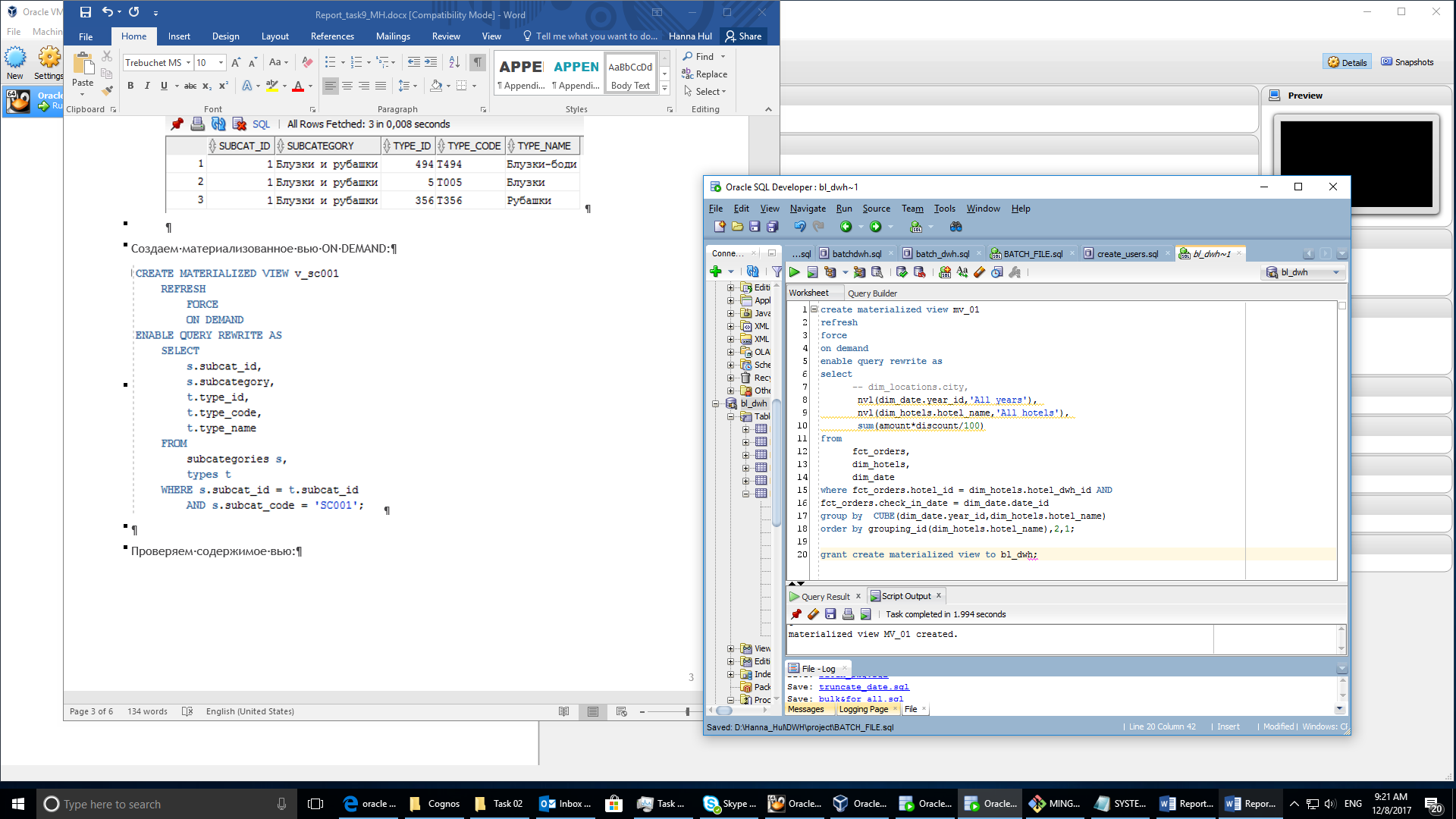
# Materialized Views- Basic

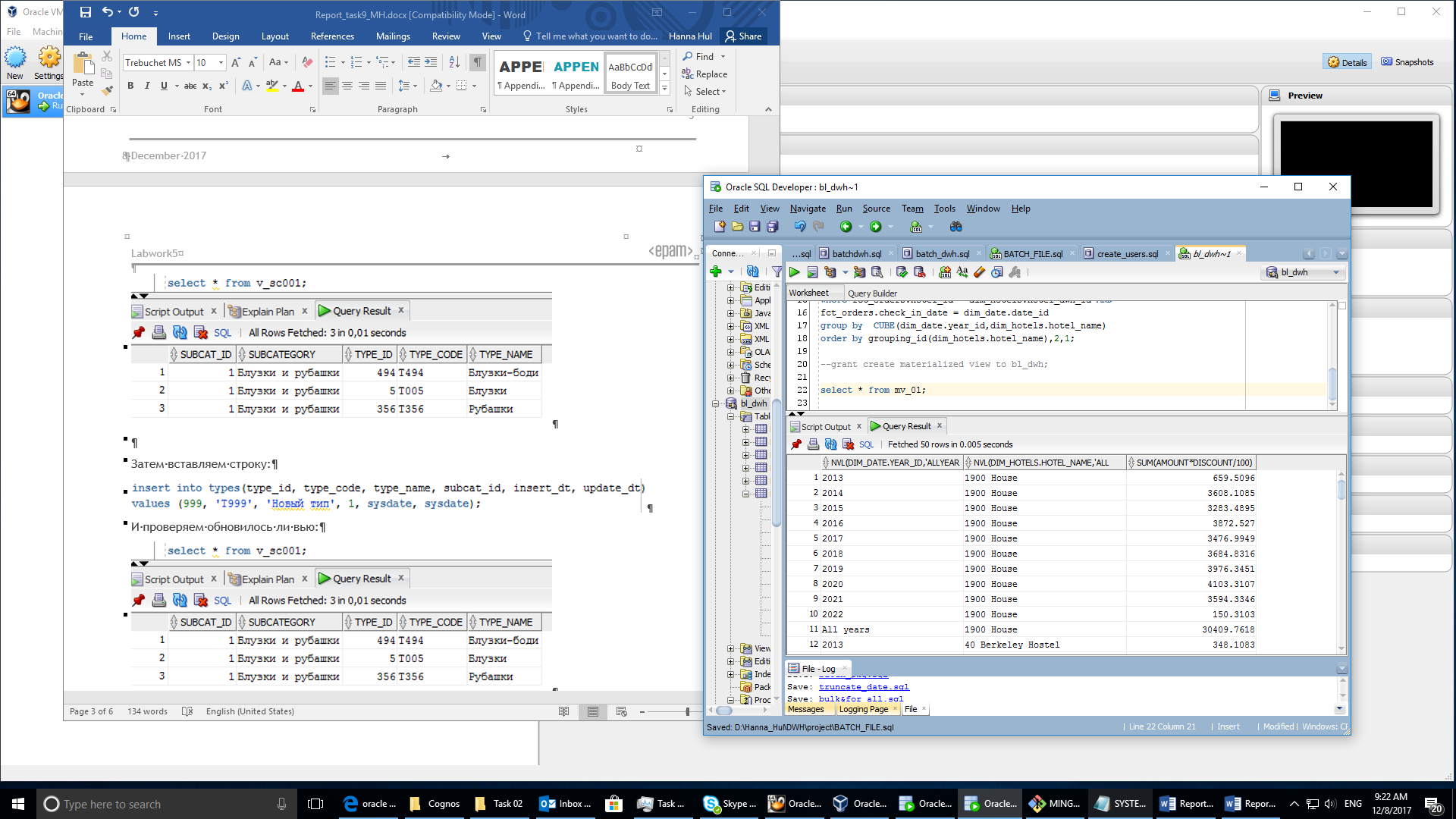
## Create Materialized Views - ON DEMAND

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

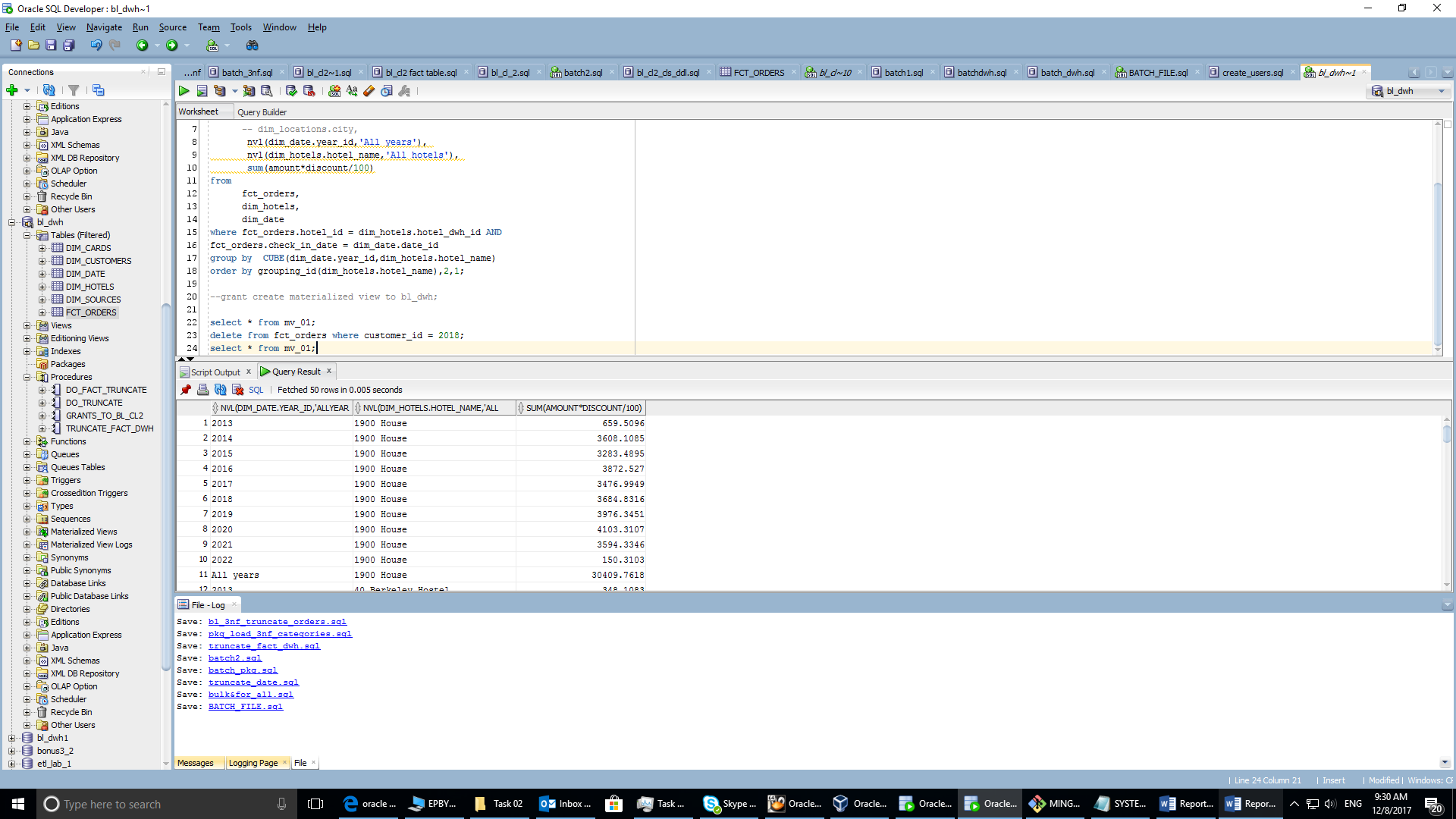
* Use Standard CREATE MATERIALIZED VIEW CLAUSE.
* Use DBMS\_MVIEW package to refresh Mat View.







After deleting data from table nothing changed.



After

* execute dbms\_mview.refresh('mv\_01');

select \* from mv\_01 order by 2,1;

you could see changes.

## 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.

