|  |
| --- |
| 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](mailto:Kiryl_Bucha@epam.com) Takushevich | 29-NOV-2017 |  |  |

Contents

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

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

[1.2. Create Materialized Views - ON COMMIT 5](#_Toc384725186)

[2. Fact Table Loading 6](#_Toc384725187)

[3. Task Results 6](#_Toc384725188)

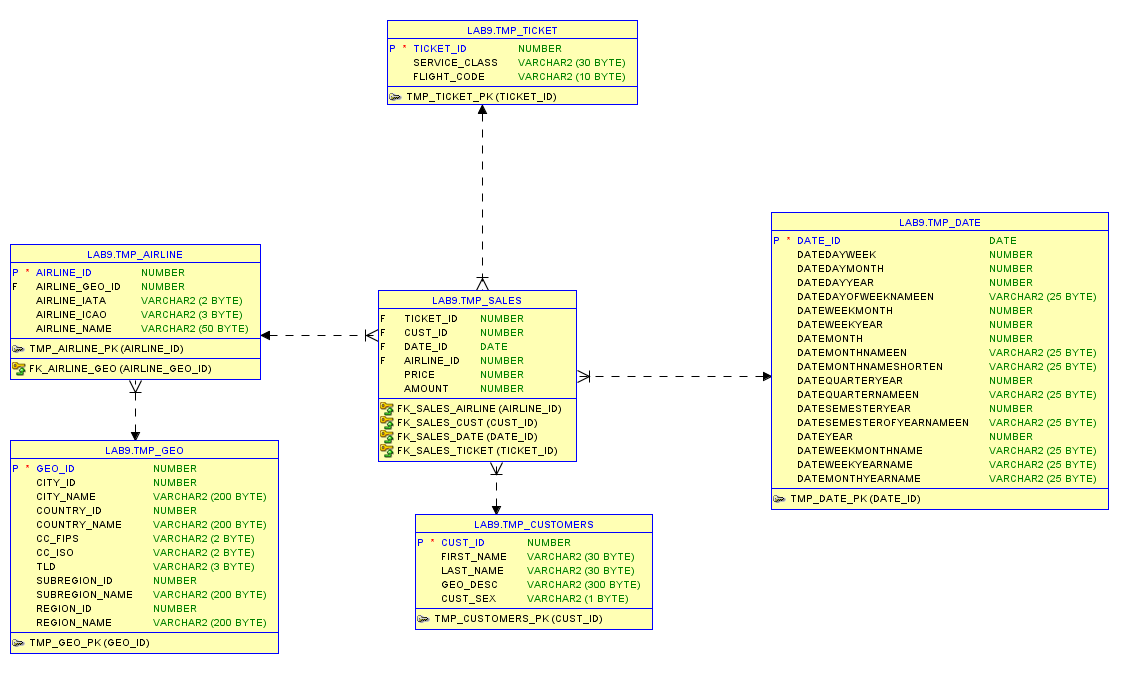
# Materialized Views- Basic

## Create Materialized Views - ON DEMAND

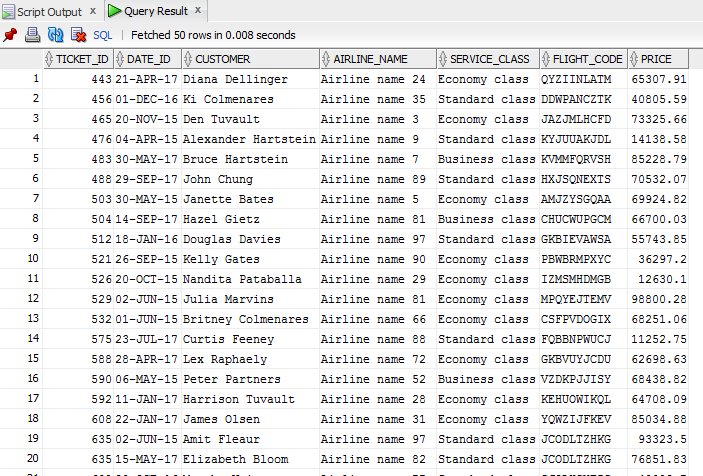
Take any "report" SQL from Lab Work 02 and create Materialized View based on it. Refresh type is ON DEMAND.

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

Звезда для второго задания:



Составленный отчёт:



Скрипт для создания материального представления:

CREATE materialized VIEW mv\_on\_demand REFRESH ON DEMAND

AS

SELECT tmp\_sales.ticket\_id,

tmp\_sales.date\_id,

first\_name

||' '

||last\_name AS customer,

airline\_name,

service\_class,

flight\_code,

price

FROM tmp\_sales

JOIN tmp\_customers

ON tmp\_sales.cust\_id = tmp\_customers.cust\_id

JOIN tmp\_airline

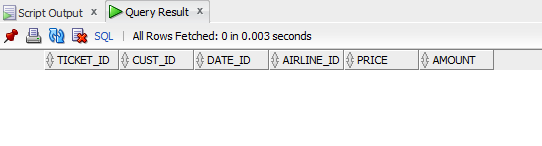
ON tmp\_sales.airline\_id=tmp\_airline.airline\_id

JOIN tmp\_ticket

ON tmp\_sales.ticket\_id=tmp\_ticket.ticket\_id;

Для проверки обновления материального представления добавим в таблицу tmp\_sales строку:

select \* from tmp\_sales where price = 999999999999;



INSERT

INTO tmp\_sales VALUES

(

1,

1,

to\_date('23/03/2014', 'dd-mm-yyyy'),

1,

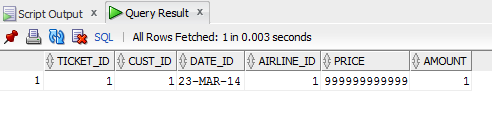
999999999999,

1

);

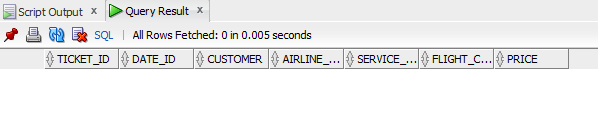
COMMIT;

SELECT \* FROM tmp\_sales WHERE price = 999999999999;



Проверяем, появилась ли эта строка в материализованном представлении:

select \* from mv\_on\_demand where price=999999999999;

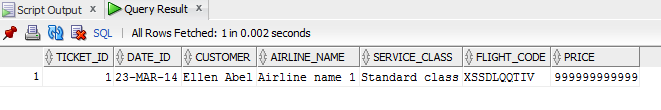


Обновляем материальное представление:

execute DBMS\_MVIEW.REFRESH( 'mv\_on\_demand' );

Находим добавленную в таблицу tmp\_sales строку в материальном представлении:

select \* from mv\_on\_demand where price=999999999999;



## 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.

Создание лога:

CREATE MATERIALIZED VIEW LOG ON tmp\_customers

WITH primary key,

ROWID,

SEQUENCE(first\_name, last\_name);

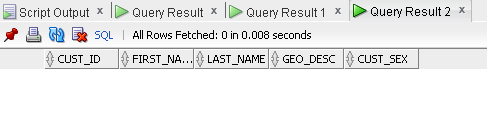
Создание материализованного представления:

CREATE materialized VIEW mv\_on\_commit REFRESH FAST ON COMMIT

AS

SELECT \* FROM tmp\_customers;

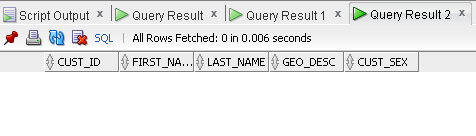
Поиск клиента с именем Hanna в таблице tmp\_customers:



Вставили строку в таблицу tmp\_customers, но не сделали коммит:

insert into tmp\_customers values (200000, 'Hanna', 'Takushevich', 'Belarus, Minsk', 'F');

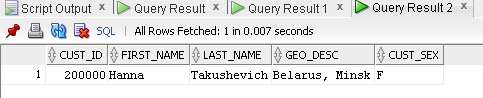
select \* from mv\_on\_commit where first\_name = 'Hanna'



После коммита:

commit;

select \* from mv\_on\_commit where first\_name = 'Hanna';



# 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.