

**EPAM Systems, RD Dep.**

# Materialized Views

---

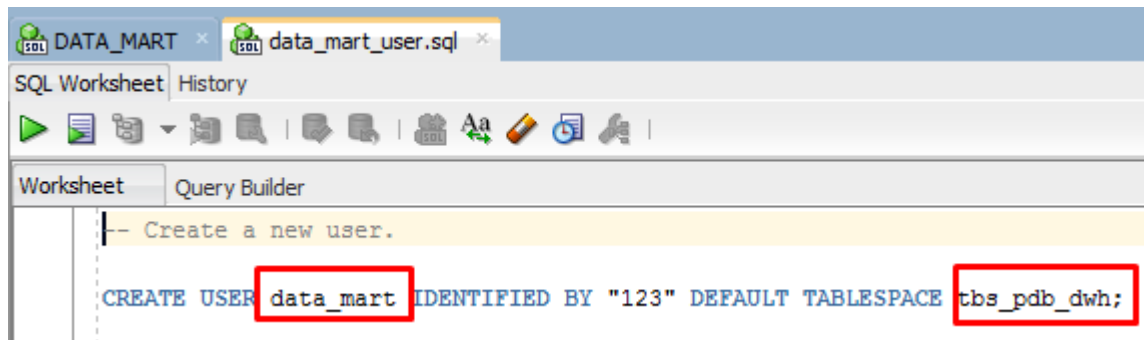
REVISION HISTORY					
Ver.	Description of Change	Author	Date	Approved	
				Name	Effective Date
1.0	Initial status	<a href="#">Valeryia_Lupanava</a>	29-NOV-2017		

## Содержание

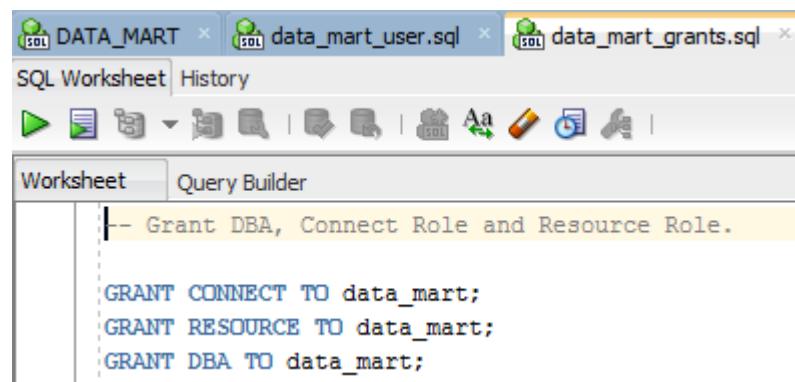
<b>1. СОЗДАНИЕ СХЕМЫ DATA_MART .....</b>	<b>3</b>
<b>2. MATERIALIZED VIEWS – ON DEMAND .....</b>	<b>4</b>
2.1. СОЗДАНИЕ MATERIALIZED VIEW .....	6
2.2. ПАКЕТ DBMS_MVIEW .....	7
<b>3. MATERIALIZED VIEWS – ON COMMIT.....</b>	<b>11</b>

## 1. Создание схемы DATA\_MART

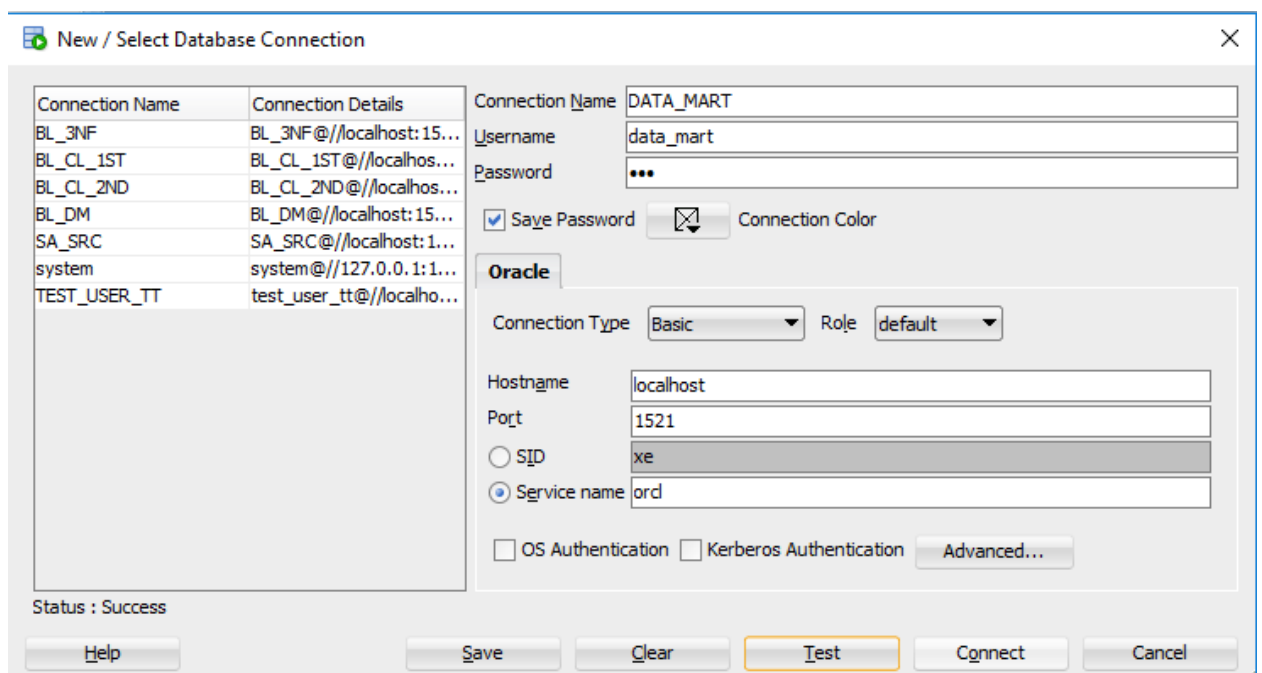
- Создаем новую схему DATA\_MART.



- Даем гранты DATA\_MART. Грант DBA использовался, поскольку этой схеме, возможно, понадобится доступ к различным таблицам 3NF и DM слоев.

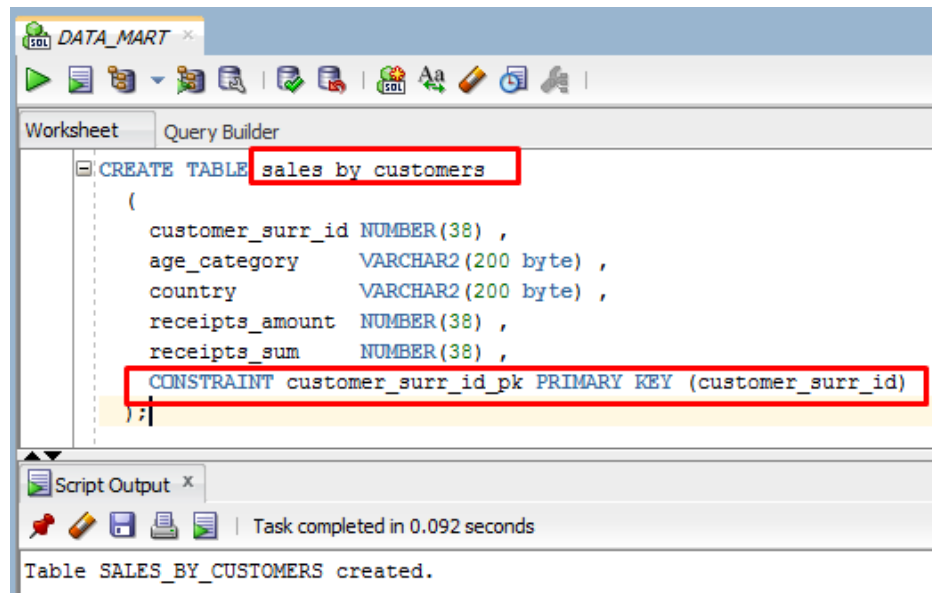


- Создаем подключение DATA\_MART.

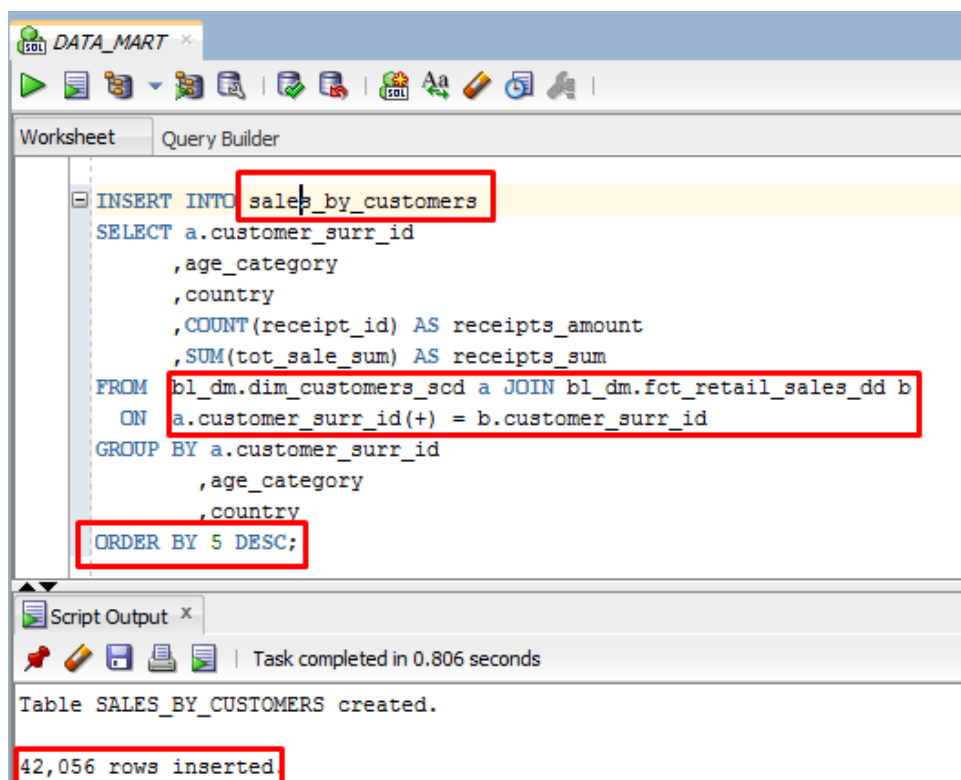


## 2. Materialized Views – ON DEMAND

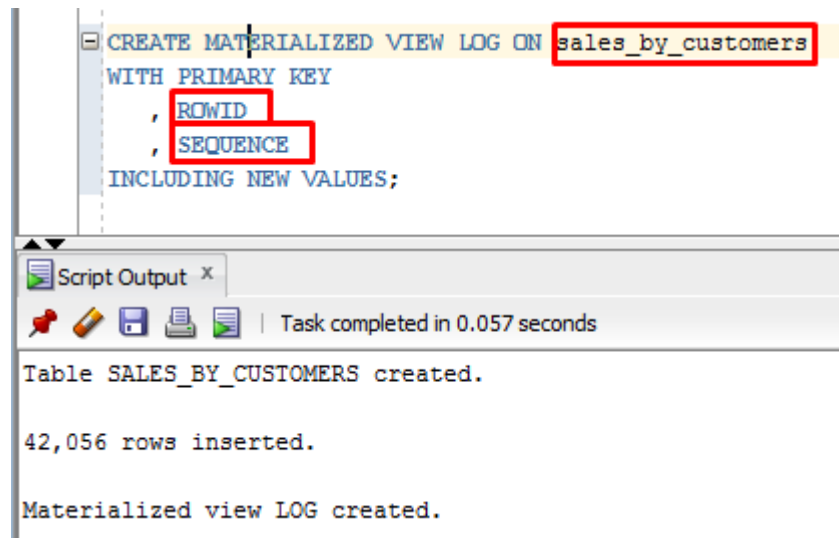
- Первым шагом создадим таблицу в нашей новой схеме. Добавим PRIMARY KEY, поскольку MATERIALIZED VIEW LOG не создастся с параметром ROWID на таблице без первичного ключа.



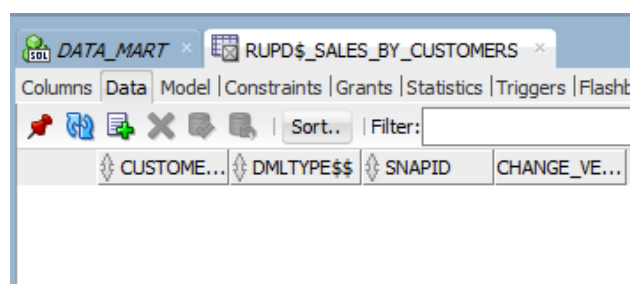
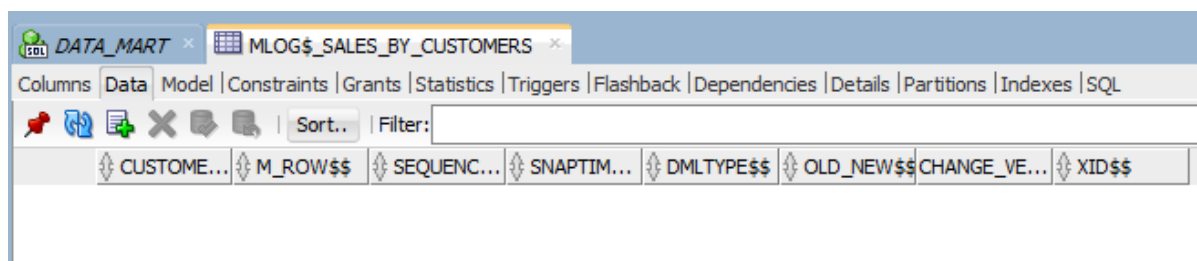
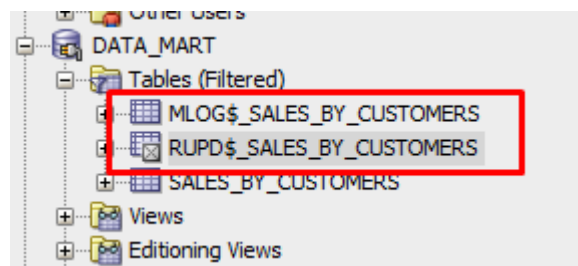
- Далее заполним нашу таблицу агрегированными значениями из таблиц DM-слоя.



- Следующим шагом создаем MATERIALIZED VIEW LOG. В параметре SEQUENCE не указываем явно поля, поскольку нас будут интересовать DML-операции со всеми полями.

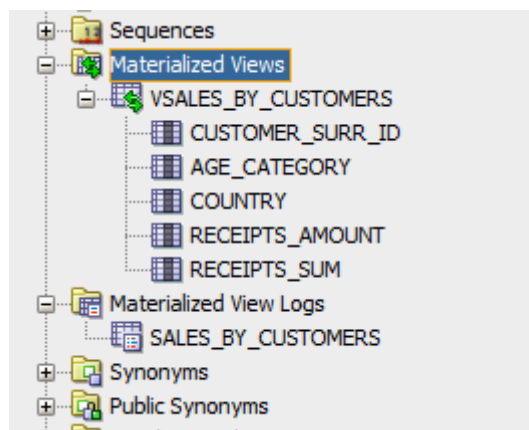
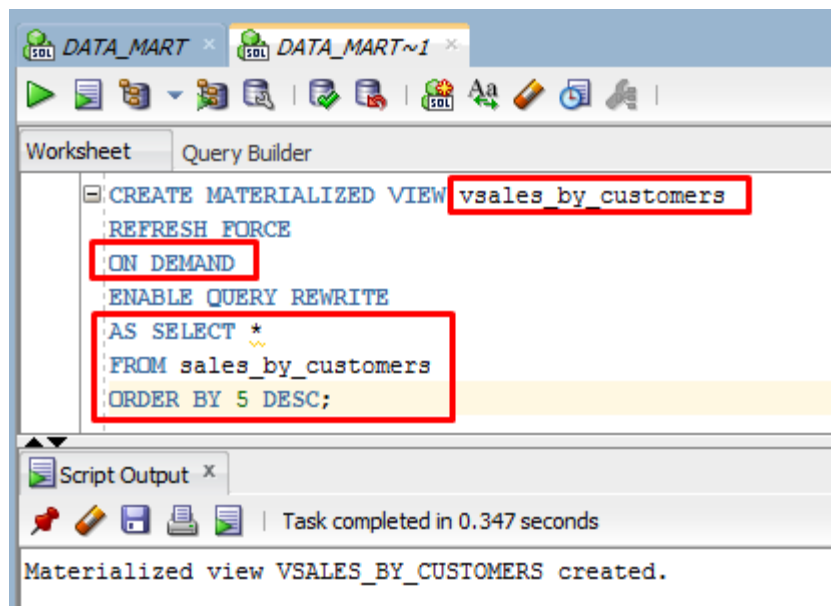


- Можно наблюдать, что сразу создались системные таблицы. На данный момент они пустые, так как никаких DML-операций не было проведено.

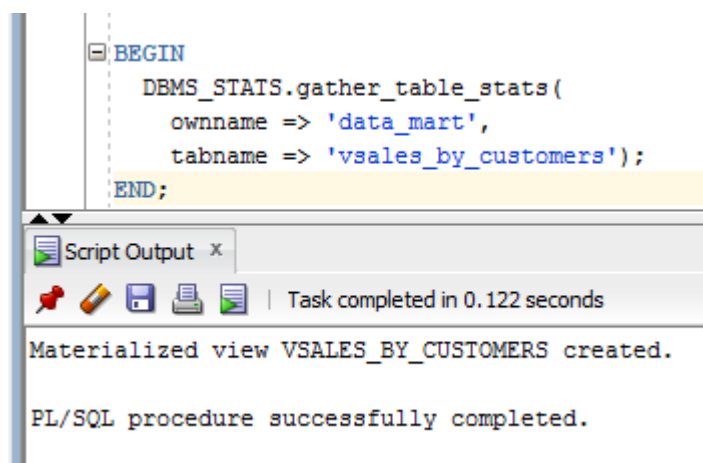


## 2.1. Создание MATERIALIZED VIEW

- Первым шагом создадим Materialized View ON DEMAND.



- Соберем статистику по созданному MATERIALIZED VIEW.



## 2.2. Пакет DBMS\_MVIEW

- Поскольку материализованное представление настроено на обновление ON DEMAND, нужно будет вручную обновлять его, но только при необходимости, так как зачастую это тяжелая операция для БД.
- Выходом может стать создание обновляемой группы представлений. В данном случае создается группа DATA\_MART.MINUTE\_REFRESH для обновления каждую минуту (для примера).

```

BEGIN
  DBMS_REFRESH.make(
    name      => 'data_mart.minute_refresh',
    list      => '',
    next_date => SYSDATE,
    interval  => '/*1:Mins*/ SYSDATE + 1/(60*24)';
END;
/

```

Script Output x

Task completed in 0.215 seconds

Materialized view VSALES\_BY\_CUSTOMERS created.

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

- Добавим в эту группу наше созданное представление.

```

BEGIN
  DBMS_REFRESH.add(
    name => 'data_mart.minute_refresh',
    list => 'data_mart.vsales_by_customers',
    lax  => TRUE);
END;
/

```

Script Output x

Task completed in 0.015 seconds

Materialized view VSALES\_BY\_CUSTOMERS created.

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

- Далее для этой группы можно настроить SHEDULE в специальных настройках

## ORACLE SCHEDULER.

- Для обычно ручного обновления можно использовать пакет DBMS\_MVIEW.REFRESH.

```
EXEC DBMS_MVIEW.refresh('vsales_by_customers');
```

Script Output x

Task completed in 0.542 seconds

Materialized view VSALES\_BY\_CUSTOMERS created.

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

- Проверим план выполнения для таблицы, на базе которой построено MATERIALIZED VIEW.

```
EXPLAIN PLAN FOR
SELECT *
FROM sales_by_customers;
```

- Можно видеть, что в действительности ORACLE обращается не к таблице, а к представлению.

```
SELECT *
FROM table dbms_xplan.display;
```

Script Output x Query Result x

All Rows Fetched: 8 in 0.18 seconds

PLAN_TABLE_OUTPUT									
1	Plan hash value: 4178981777								
2									
3	-----								
4	Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time		
5	-----								
6	0	SELECT STATEMENT		42056	1273K	62 (0)	00:00:01		
7	1	MAT_VIEW REWRITE ACCESS FULL	VSALES_BY_CUSTOMERS	42056	1273K	62 (0)	00:00:01		
8	-----								

- Удалим из нашей таблицы первую строку.



```
DELETE FROM sales_by_customers
WHERE rownum = 1;
```

Script Output x

Task completed in 0.033 seconds

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

1 row deleted.

- В LOG-таблице сразу появилась запись. Можно наблюдать, что в качестве ROWID записан ключ удаленной строки, что очень удобно для дальнейшего поиска записи с этим ключом.

CUSTOMER_SURR_ID	M_ROW\$\$	SEQUENCE\$\$	SNAPTME\$\$	DMLTYPE\$\$	OLD_NEW\$\$	CHANGE_VECTOR\$\$	XID\$\$
509098	AAAxugAAeAAAErKABR	1	01-JAN-00	D	0	00	1688931464705770

- Сделаем SELECT в нашем представлении по этому ROWID. Как видно, запись сохранилась, поскольку обновления представления не было.

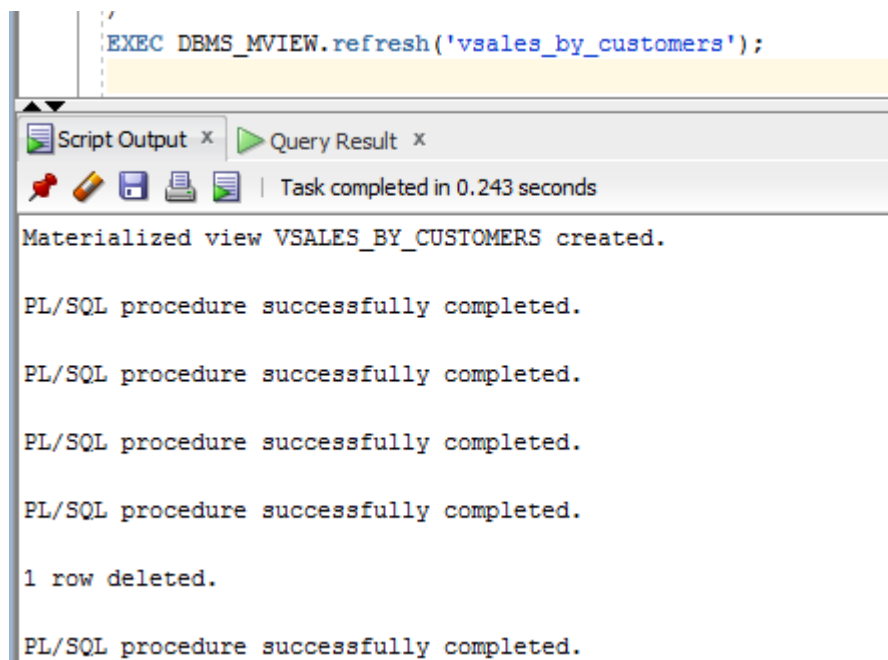
```
SELECT TO_CHAR (SYSDATE, 'DD.MM.YY HH24:MI') AS time, a.*
FROM vsales_by_customers a
WHERE customer_surr_id = 509098;
```

Script Output x Query Result x

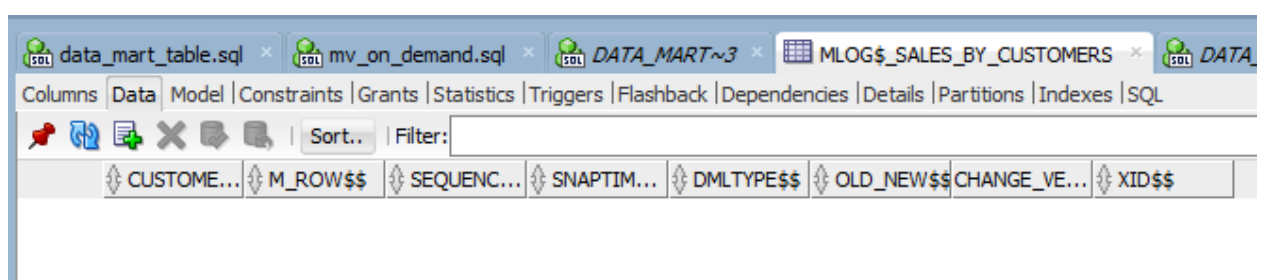
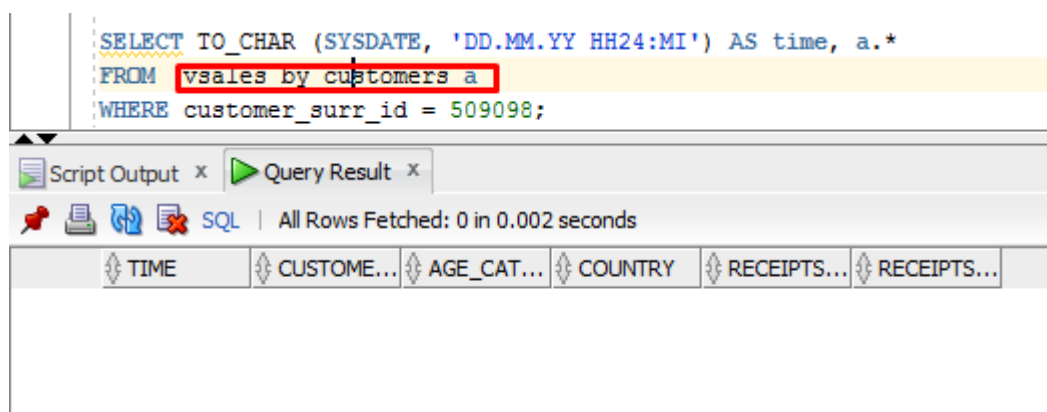
All Rows Fetched: 1 in 0.003 seconds

TIME	CUSTOMER_SURR_ID	AGE_CATEGORY	COUNTRY	RECEIPTS_AMOUNT	RECEIPTS_SUM
29.11.17 03:19	509098	old	Holy See	3	112514

- Запустим пакет на обновление представления.



- После повторного выполнения запроса можно наблюдать, что записи больше нет с данным ключом. LOG-таблица также пуста, поскольку обновление выполнилось и изменения применились к представлению.





```

SELECT TO_CHAR (SYSDATE, 'DD.MM.YY HH24:MI') AS time, a.*
FROM vsales_by_customers_c a
WHERE customer_surr_id = 509099;

```

Script Output x Query Result x

SQL | All Rows Fetched: 1 in 0.005 seconds

	TIME	CUSTOMER_SURR_ID	COUNTRY	RECEIPTS_SUM
1	29.11.17 03:27	509099	United States of America	159745

- Сделаем COMMIT.

```

COMMIT;

```

Script Output x Query Result x

Task completed in 0.147 seconds

Materialized view VSALES\_BY\_CUSTOMERS\_C created.

Materialized view VSALES\_BY\_CUSTOMERS\_C created.

1 row deleted.

Commit complete.

- Теперь при повторном выполнении запроса к нашему представлению можно наблюдать, что запись пропала и без выполнения пакета DBMS\_MVIEW.REFRESH, поскольку был сделан COMMIT.

```

SELECT TO_CHAR (SYSDATE, 'DD.MM.YY HH24:MI') AS time, a.*
FROM vsales_by_customers_c a
WHERE customer_surr_id = 509099;

```

Script Output x Query Result x

SQL | All Rows Fetched: 0 in 0.002 seconds

	TIME	CUSTOMER_SURR_ID	COUNTRY	RECEIPTS_SUM
--	------	------------------	---------	--------------

- В LOG-таблице запись об необходимых изменениях также пропала.

mv\_on\_demand.sql x DATA\_MART~3 x DATA\_MART~5 x MLOG\$\_SALES\_BY\_CUSTOMERS

Columns Data Model Constraints Grants Statistics Triggers Flashback Dependencies Details Partitions Indexes SQL

Sort.. Filter:

	CUSTOMER_SURR_ID	M_ROWID	SEQUENCE	SNAPTIMESTAMP	DMLTYPE	OLD_NEW	CHANGE_VECTOR	XID
--	------------------	---------	----------	---------------	---------	---------	---------------	-----