Copyright (c) 2022, Oracle. All rights reserved. Oracle Confidential.

Materialized View (MVIEW) Not Reflecting Update (Doc ID 1417323.1)

In this Document

APPLIES TO:

Oracle Database - Enterprise Edition - Version 10.2.0.1 and later
Oracle Database Cloud Schema Service - Version N/A and later
Oracle Database Exadata Cloud Machine - Version N/A and later
Oracle Cloud Infrastructure - Database Service - Version N/A and later
Oracle Database Backup Service - Version N/A and later
Information in this document applies to any platform.

SYMPTOMS

NOTE: In the images and/or the document content below, the user information and data used represents fictitious data from the Oracle sample schema(s) or Public Documentation delivered with an Oracle database product. Any similarity to actual persons, living or dead, is purely coincidental and not intended in any manner.

MVIEW doesn't reflect the UPDATES even though "INCLUDING NEW VALUES" is mentioned.
Only the inserted records are being reflected.

```
-- Create table
SQL> CREATE TABLE TEST_TAB
2 (
3 LOG_PK NUMBER(10) CONSTRAINT NN_10020_TEST_TAB NOT NULL,
4 OFFICE_BIC_CODE VARCHAR2(16 BYTE) CONSTRAINT NN_10021_TEST_TAB NOT NULL,
5 REQUEST_REF_NO VARCHAR2(16 BYTE) CONSTRAINT NN_10022_TEST_TAB NOT NULL,
6 REQUEST_SENT_DATE DATE CONSTRAINT NN_10023_TEST_TAB NOT NULL,
7 REQUEST_TYPE VARCHAR2(30 BYTE) CONSTRAINT NN_10024_TEST_TAB NOT NULL,
8 RESPONSE_MIN_LAST_UPDATE_DATE DATE,
9 RESPONSE_ERROR_CODE VARCHAR2(40 BYTE),
10 APP_REGI_DATE DATE CONSTRAINT NN_10025_TEST_TAB NOT NULL,
11 APP_UPD_DATE DATE CONSTRAINT NN_10026_TEST_TAB NOT NULL,
12 CREATED_BY VARCHAR2(20 BYTE) CONSTRAINT NN_10027_TEST_TAB NOT NULL,
13 CREATION_DATE DATE CONSTRAINT NN_10028_TEST_TAB NOT NULL,
14 UPDATED_BY VARCHAR2(20 BYTE) CONSTRAINT NN_10029_TEST_TAB NOT NULL,
15 UPDATE_DATE DATE CONSTRAINT NN_10030_TEST_TAB NOT NULL
16 )
17 LOGGING
18 NOCOMPRESS
19 NOCACHE
20 NOPARALLEL
21 MONITORING;

Table created.

SQL> ALTER TABLE TEST_TAB ADD (CONSTRAINT PK_756_TEST_TAB PRIMARY KEY (LOG_PK));

Table altered.

-- Create MVIEW log
SQL> CREATE MATERIALIZED VIEW LOG ON TEST_TAB
```

```
2 WITH ROWID , SEQUENCE(office_bic_code, response_min_last_update_date), primary key
3 INCLUDING NEW VALUES;

Materialized view log created.

-- Create MVIEW
SQL> CREATE MATERIALIZED VIEW TEST_TAB_mv
2 BUILD IMMEDIATE
3 REFRESH FAST ON COMMIT
4 AS
5 SELECT office_bic_code, MAX(response_min_last_update_date) response_last_update_date
6 FROM TEST_TAB
7 GROUP BY office_bic_code;

Materialized view created.

-- Check the date
SQL> select to_char(sysdate,'dd-mm-yyyy hh24:mi:ss') from dual;

TO_CHAR(SYSDATE,'DD
-------------------
20-02-2012 11:22:08

-- Check the MVIEW last refresh date
SQL> select MVIEW_NAME, to_char(LAST_REFRESH_DATE,'dd-mm-yyyy hh24:mi:ss') from user_mviews
where mview_name = 'TEST_TAB_MV';

MVIEW_NAME TO_CHAR(LAST_REFRES
------------------------------ -------------------
TEST_TAB_MV 20-02-2012 11:22:03

-- Insert rerord into the table
SQL> INSERT INTO TEST_TAB ( LOG_PK, OFFICE_BIC_CODE, REQUEST_REF_NO, REQUEST_SENT_DATE,
REQUEST_TYPE,
2 RESPONSE_MIN_LAST_UPDATE_DATE, RESPONSE_ERROR_CODE, APP_REGI_DATE, APP_UPD_DATE,
CREATED_BY,
3 CREATION_DATE, UPDATED_BY, UPDATE_DATE ) VALUES (
4 17, 'TEST1', 'CRQ000000004', TO_Date( '12/28/2011 11:11:12 AM', 'MM/DD/YYYY HH:MI:SS AM')
5 , 'MultiTradeLevelRequest', TO_Date( '01/01/2012 12:07:47 PM', 'MM/DD/YYYY HH:MI:SS AM')
6 , NULL, TO_Date( '04/18/2011 12:00:00 AM', 'MM/DD/YYYY HH:MI:SS AM'), TO_Date( '04/18/2011
12:00:00 AM', 'MM/DD/YYYY HH:MI:SS AM')
7 , 'SYSTEM', TO_Date( '12/28/2011 11:11:08 AM', 'MM/DD/YYYY HH:MI:SS AM'), 'SYSTEM'
8 , TO_Date( '12/28/2011 11:11:08 AM', 'MM/DD/YYYY HH:MI:SS AM'));

1 row created.

SQL> COMMIT;

Commit complete.

-- Check the MVIEW log
SQL> select count(*) from mlog$_TEST_TAB;

COUNT(*)
----------
0

-- Check MVIEW last refresh date. It should be changed, as MVIEW is ON COMMIT refresh. It has
actually refreshed.
SQL> select MVIEW_NAME, to_char(LAST_REFRESH_DATE,'dd-mm-yyyy hh24:mi:ss') from user_mviews
where mview_name = 'TEST_TAB_MV';

MVIEW_NAME TO_CHAR(LAST_REFRES
------------------------------ -------------------
TEST_TAB_MV 20-02-2012 11:22:23

SQL> update TEST_TAB set response_min_last_update_date =sysdate +3 where log_pk=17;

1 row updated.

SQL> commit;

Commit complete.
```

```
SQL> select * from TEST_TAB_MV;

OFFICE_BIC_CODE RESPONSE_
---------------- ---------
TEST1 01-JAN-12

-- check the last refresh date. It should have changed as the MV is refresh on COMMIT, But
it's actually not refreshed.
SQL> select MVIEW_NAME, to_char(LAST_REFRESH_DATE,'dd-mm-yyyy hh24:mi:ss') from user_mviews
where mview_name = 'TEST_TAB_MV';

MVIEW_NAME TO_CHAR(LAST_REFRES
------------------------------ -------------------
TEST_TAB_MV 20-02-2012 11:22:23

SQL>
```

Why has the MVIEW not refreshed and not reflected the correct entry? To know this, see the next code section.

CHANGES

CAUSE

See the output of MV_CAPABILITIES_TABLE which explains why MVIEW was not refreshed on commit:

```
SQL> create table MV_CAPABILITIES_TABLE
2 (
3 STATEMENT_ID VARCHAR2(30),
4 MVNAME VARCHAR2(30),
5 CAPABILITY_NAME VARCHAR2(30),
6 POSSIBLE CHAR(1),
7 RELATED_TEXT VARCHAR2(2000),
8 RELATED_NUM NUMBER,
9 MSGNO INTEGER,
10 MSGTXT VARCHAR2(2000),
11 SEQ NUMBER,
12 MVOWNER VARCHAR2(30)
13 )
14 tablespace USERS
15 pctfree 10
16 initrans 1
17 maxtrans 255
18 storage
19 (
20 initial 64K
21 minextents 1
22 maxextents unlimited
23 );

Table created.

SQL> execute dbms_mview.explain_mview('TEST_TAB_mv');

PL/SQL procedure successfully completed.

SQL> set linesize 150
SQL> col msgtxt for a50
SQL> select capability_name, possible, msgtxt from mv_capabilities_table where
capability_name like 'REFRESH%' order by seq;

CAPABILITY_NAME                                    POS MSGTXT
-------------------------------------------------- ---
--------------------------------------------------
REFRESH_COMPLETE                                   Y
```

```
REFRESH_FAST                                        Y
REFRESH_FAST_AFTER_INSERT                           Y
REFRESH_FAST_AFTER_ONETAB_DML                       N   mv uses the MIN or MAX aggregate
functions
REFRESH_FAST_AFTER_ONETAB_DML                       N   COUNT(*) is not present in the select
list
REFRESH_FAST_AFTER_ANY_DML                          N   see the reason why
REFRESH_FAST_AFTER_ONETAB_DML i

                                                        s disabled


REFRESH_FAST_PCT                                    N   PCT is not possible on any of the
detail tables in

                                                         the materialized view
```

So, the cause is COUNT(*) is not being used in MVIEW creation when aggregate functions are being used.

SOLUTION

COUNT(*) must always be present to guarantee all types of fast refresh. Otherwise, you may be limited to fast refresh after inserts only.
COUNT(*) is needed for MV with aggregate as it records the number of rows in each group. The refresh operations may increase or decrease the count when incrementally apply the change data from log. When the count becomes 0 after applying the changes, the row in the MV should be deleted as the group no longer exists.
See the same testcase (after making changes):

```
-- Create MV with COUNT(*) as past of SELECT statement
SQL> CREATE MATERIALIZED VIEW TEST_TAB_mv
2 BUILD IMMEDIATE
3 REFRESH FAST ON COMMIT
4 AS
5 SELECT count(*), office_bic_code, MAX(response_min_last_update_date)
response_last_update_date
6 FROM TEST_TAB
7 GROUP BY office_bic_code;

Materialized view created.

SQL> select to_char(sysdate,'dd-mm-yyyy hh24:mi:ss') from dual;

TO_CHAR(SYSDATE,'DD
-------------------
20-02-2012 11:33:19

SQL> select MVIEW_NAME, to_char(LAST_REFRESH_DATE,'dd-mm-yyyy hh24:mi:ss') from user_mviews
where mview_name = 'TEST_TAB_MV';

MVIEW_NAME TO_CHAR(LAST_REFRES
------------------------------ -------------------
TEST_TAB_MV 20-02-2012 11:33:12

SQL> INSERT INTO TEST_TAB ( LOG_PK, OFFICE_BIC_CODE, REQUEST_REF_NO, REQUEST_SENT_DATE,
REQUEST_TYPE,
2 RESPONSE_MIN_LAST_UPDATE_DATE, RESPONSE_ERROR_CODE, APP_REGI_DATE, APP_UPD_DATE,
CREATED_BY,
3 CREATION_DATE, UPDATED_BY, UPDATE_DATE ) VALUES (
4 17, 'TEST1', 'CRQ000000004', TO_Date( '12/28/2011 11:11:12 AM', 'MM/DD/YYYY HH:MI:SS AM')
5 , 'MultiTradeLevelRequest', TO_Date( '01/01/2012 12:07:47 PM', 'MM/DD/YYYY HH:MI:SS AM')
6 , NULL, TO_Date( '04/18/2011 12:00:00 AM', 'MM/DD/YYYY HH:MI:SS AM'), TO_Date( '04/18/2011
12:00:00 AM', 'MM/DD/YYYY HH:MI:SS AM')
7 , 'SYSTEM', TO_Date( '12/28/2011 11:11:08 AM', 'MM/DD/YYYY HH:MI:SS AM'), 'SYSTEM'
8 , TO_Date( '12/28/2011 11:11:08 AM', 'MM/DD/YYYY HH:MI:SS AM'));

1 row created.

SQL>
SQL> COMMIT;

Commit complete.

SQL> select count(*) from mlog$_TEST_TAB;

COUNT(*)
----------
0

SQL> select MVIEW_NAME, to_char(LAST_REFRESH_DATE,'dd-mm-yyyy hh24:mi:ss') from user_mviews
where mview_name = 'TEST_TAB_MV';

MVIEW_NAME TO_CHAR(LAST_REFRES
------------------------------ -------------------
TEST_TAB_MV 20-02-2012 11:33:31

SQL> update TEST_TAB set response_min_last_update_date =sysdate +3 where log_pk=17;

1 row updated.

SQL> commit;

Commit complete.

SQL> select * from TEST_TAB_MV;

COUNT(*) OFFICE_BIC_CODE RESPONSE_
---------- --------------- ---------
1 TEST1 23-FEB-12

-- Check MV's LAST_REFRESH_DATE.

SQL> select MVIEW_NAME, to_char(LAST_REFRESH_DATE,'dd-mm-yyyy hh24:mi:ss') from user_mviews
where mview_name = 'TEST_TAB_MV';

MVIEW_NAME TO_CHAR(LAST_REFRES
------------------------------ -------------------
TEST_TAB_MV 20-02-2012 11:35:14

SQL>

-- See that it has changed; that means ON COMMIT MVIEW has refreshed successfully after the
last update (and commit) statement.
```

Didn't find what you are looking for?