**Report “Lab 6”**

**Anton Tserakhau**

# Materialized Views- Basic

## Task 01: Create Materialized Views - ON DEMAND

I have created Materialized Views, which will refresh ON DEMAND (based on first Report Layout Monthly).

CREATE MATERIALIZED VIEW mv\_sales\_city\_country\_month

BUILD DEFERRED

REFRESH COMPLETE ON DEMAND

AS SELECT restaurant\_country\_name AS country

, restaurant\_city AS city

, TO\_CHAR ( amount

, '9,999,999' )

AS amount

, TO\_CHAR ( total\_price

, '$999,999,999,999' )

AS total\_price

FROM ( SELECT DECODE ( GROUPING ( rest.restaurant\_country\_name ), 1, 'All Countries', rest.restaurant\_country\_name )

AS restaurant\_country\_name

, DECODE ( GROUPING ( rest.restaurant\_city ), 1, 'All Cities', rest.restaurant\_city ) AS restaurant\_city

, SUM ( oper.unit\_amount ) AS amount

, SUM ( oper.total\_price\_dol ) AS total\_price

FROM u\_dw\_ext\_references.cls\_operations oper

LEFT JOIN u\_dw\_ext\_references.cls\_restaurants rest

ON oper.restaurant\_code = rest.restaurant\_code

LEFT JOIN u\_dw\_ext\_references.cls\_dishes dish

ON dish.dish\_code = oper.dish\_code

WHERE TRUNC ( oper.event\_dt

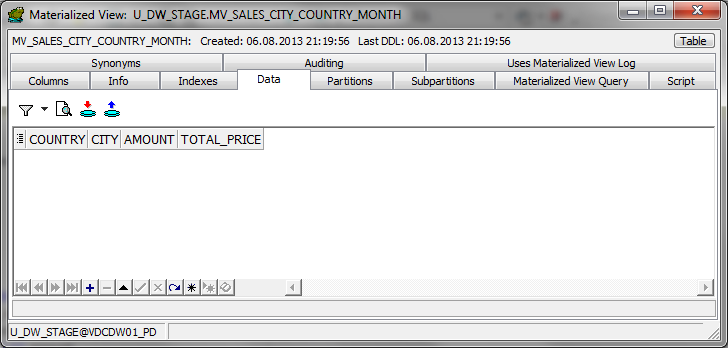
, 'MONTH' ) = TO\_DATE ( '01-JAN-2012'

, 'DD-MON-YYYY' )

GROUP BY ROLLUP ( rest.restaurant\_country\_name, rest.restaurant\_city )

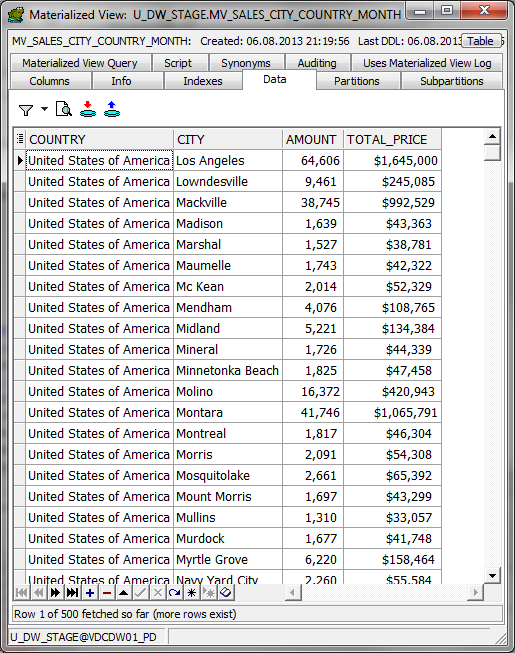
ORDER BY rest.restaurant\_country\_name

, rest.restaurant\_city);



Materialized View is empty; the data will load after script:

EXECUTE DBMS\_MVIEW.REFRESH('mv\_sales\_city\_country\_month');



## Create Materialized Views - ON COMMIT

I have created Materialized Views, which will refresh ON COMMIT (based on first Report Layout Daily).

CREATE MATERIALIZED VIEW mv\_sales\_rest\_dish\_type\_daily

BUILD IMMEDIATE

REFRESH ON COMMIT

ENABLE QUERY REWRITE

AS

SELECT rest.restaurant\_name

AS restaurant\_name

, rest.restaurant\_address

AS restaurant\_address

, SUM ( oper.unit\_amount ) AS amount

,COUNT ( oper.unit\_amount ) AS count\_amount

, SUM ( oper.total\_price\_dol ) AS total\_price

, COUNT ( oper.total\_price\_dol ) AS count\_total\_price

FROM u\_dw\_ext\_references.cls\_operations oper

LEFT JOIN u\_dw\_ext\_references.cls\_restaurants rest

ON oper.restaurant\_code = rest.restaurant\_code

WHERE TRUNC ( oper.event\_dt

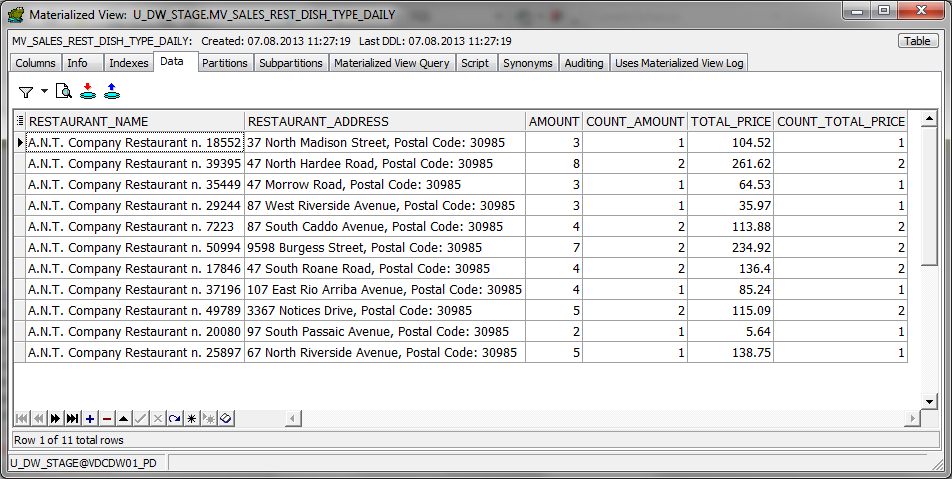
, 'DD' ) = TO\_DATE ( '06-JAN-2012'

, 'DD-MON-YYYY' )

AND rest.restaurant\_country\_name = 'Germany'

AND rest.restaurant\_city = 'Chemnitz'

GROUP BY rest.restaurant\_name, rest.restaurant\_address;



I have run script (set price=price\*2 when restaurants in city Chemnitz in Germany):

UPDATE u\_dw\_ext\_references.cls\_operations

SET total\_price\_dol = total\_price\_dol \* 2

WHERE restaurant\_code IN (SELECT DISTINCT rest.restaurant\_code

FROM u\_dw\_ext\_references.cls\_operations oper

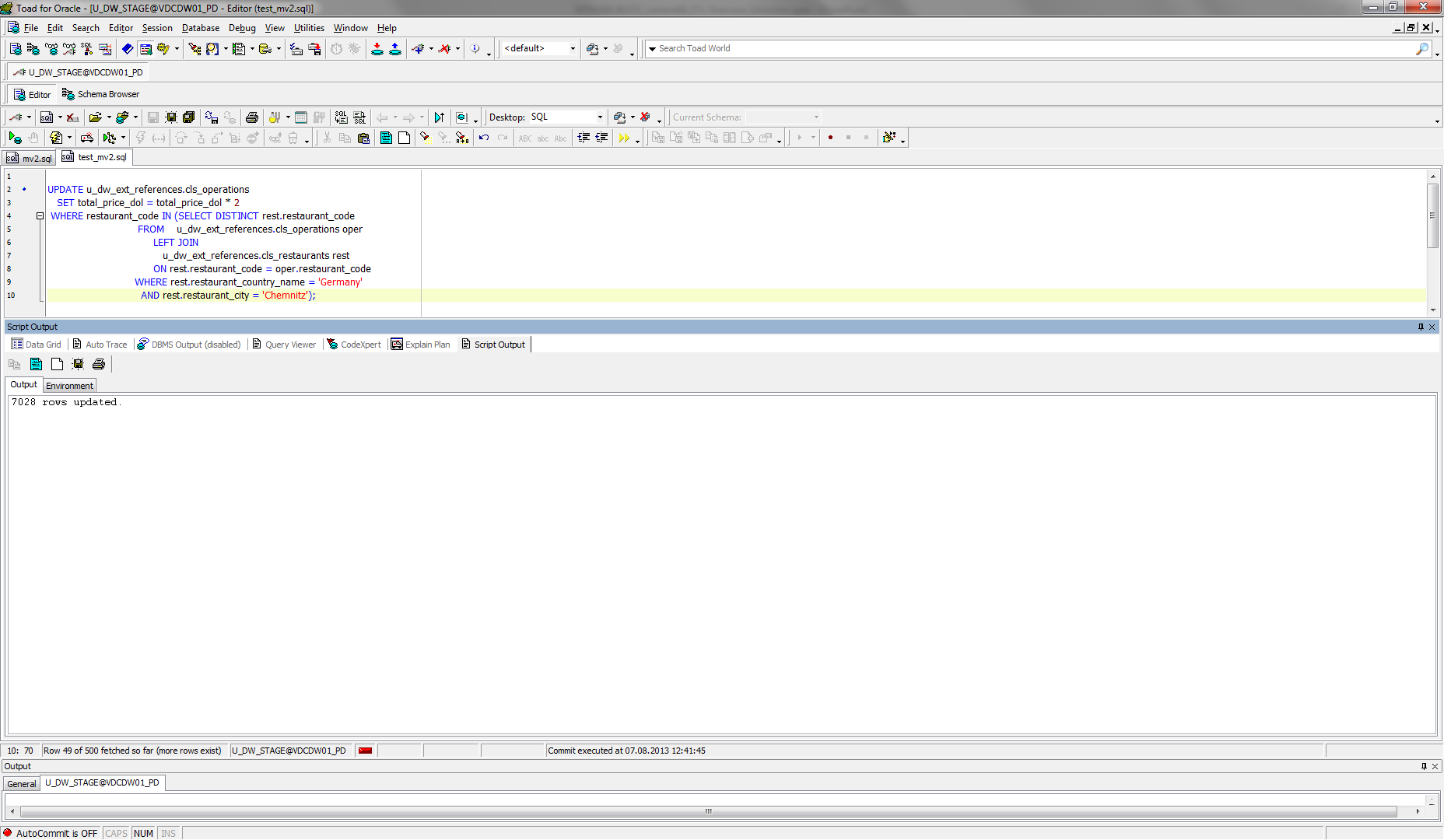
LEFT JOIN

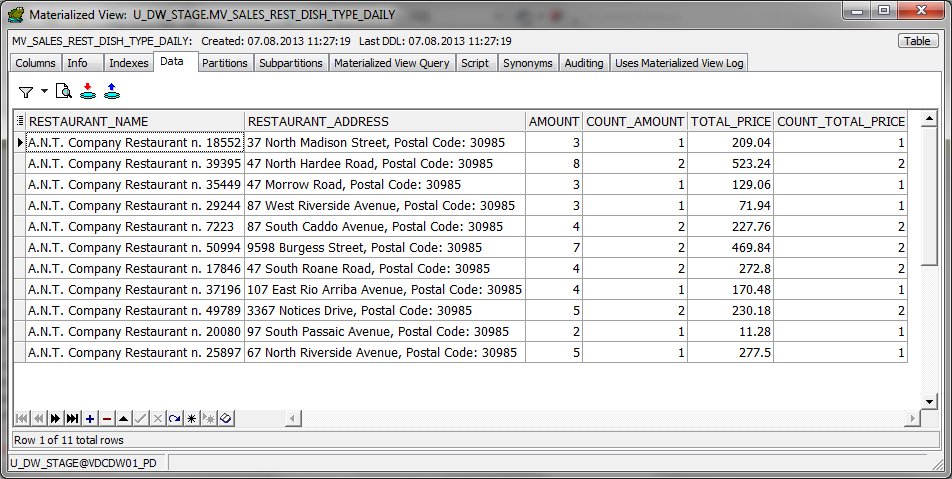
u\_dw\_ext\_references.cls\_restaurants rest

ON rest.restaurant\_code = oper.restaurant\_code

WHERE rest.restaurant\_country\_name = 'Germany'

AND rest.restaurant\_city = 'Chemnitz');





# Materialized Views- Model Clause

## Task 03: Create Materialized Views - Refreshing at definitive Time moment

I have created Materialized Views, which will refresh definitive Time moment (based on last Report Layout Monthly).

CREATE MATERIALIZED VIEW mv\_sales\_city\_cntr\_month\_delta

BUILD IMMEDIATE

REFRESH FORCE ON DEMAND START WITH SYSDATE NEXT (SYSDATE + 1/1440)

AS

SELECT NVL ( country, 'Total' ) AS country

, NVL ( city, '-' ) AS city

, NVL ( TO\_CHAR ( month\_numb ), '-' ) AS number\_of\_month

, NVL ( month\_name, '-' ) AS month\_name

, TO\_CHAR ( amount

, '9,999,999' )

AS amount

, NVL ( TO\_CHAR ( delta\_amount

, '9,999,999' )

, '-' )

AS delta\_amount

, TO\_CHAR ( total\_sales

, '$999,999,999,999' )

AS total\_sales

, NVL ( TO\_CHAR ( delta\_sales

, '$999,999,999,999' )

, '-' )

AS delta\_sales

FROM ( SELECT country\_name AS country

, city\_name AS city

, month\_numb

, month\_name

, SUM ( amount ) AS amount

, delta\_amount

, SUM ( total\_price ) AS total\_sales

, delta\_sales

FROM (SELECT \*

FROM ( SELECT TO\_NUMBER ( TO\_CHAR ( TRUNC ( oper.event\_dt

, 'MONTH' )

, 'mm' ) )

AS month\_numb

, TO\_CHAR ( TRUNC ( oper.event\_dt

, 'MONTH' )

, 'Month' )

AS month\_name

, countries.region\_desc AS country\_name

, cities.city\_desc AS city\_name

, SUM ( oper.unit\_amount ) AS amount

, SUM ( oper.total\_price\_dol ) AS total\_price

FROM t\_operations oper

LEFT JOIN t\_restaurants rest

ON oper.restaurant\_id = rest.restaurant\_id

LEFT JOIN t\_restaurant\_types

ON rest.restaurant\_type\_id = t\_restaurant\_types.restaurant\_type\_id

LEFT JOIN t\_dishes dishes

ON dishes.dish\_id = oper.dish\_id

LEFT JOIN t\_dish\_types

ON dishes.dish\_type\_id = t\_dish\_types.dish\_type\_id

LEFT JOIN t\_dish\_cuisines

ON dishes.dish\_cuisine\_id = t\_dish\_cuisines.dish\_cuisine\_id

LEFT JOIN u\_dw\_references.lc\_cities cities

ON rest.restaurant\_geo\_id = cities.geo\_id

LEFT JOIN u\_dw\_references.t\_geo\_object\_links links

ON links.child\_geo\_id = cities.geo\_id

LEFT JOIN u\_dw\_references.cu\_countries countries

ON links.parent\_geo\_id = countries.geo\_id

WHERE countries.region\_desc IN ('United States of America')

AND cities.city\_desc IN ('Hardy', 'Woodstock')

GROUP BY TO\_NUMBER ( TO\_CHAR ( TRUNC ( oper.event\_dt

, 'MONTH' )

, 'mm' ) )

, TO\_CHAR ( TRUNC ( oper.event\_dt

, 'MONTH' )

, 'Month' )

, countries.region\_desc

, cities.city\_desc

ORDER BY TO\_NUMBER ( TO\_CHAR ( TRUNC ( oper.event\_dt

, 'MONTH' )

, 'mm' ) )

, city\_name)

MODEL RETURN UPDATED ROWS

PARTITION BY ( country\_name, city\_name )

DIMENSION BY ( month\_numb )

MEASURES ( month\_name, amount, 0 delta\_amount, total\_price, 0 delta\_sales )

RULES AUTOMATIC ORDER

( delta\_amount [month\_numb] =

amount[CV ( month\_numb )]

- NVL ( amount[CV ( month\_numb ) - 1], amount[CV ( month\_numb )] ),

delta\_sales [month\_numb] =

total\_price[CV ( month\_numb )]

- NVL ( total\_price[CV ( month\_numb ) - 1], total\_price[CV ( month\_numb )] ) ))

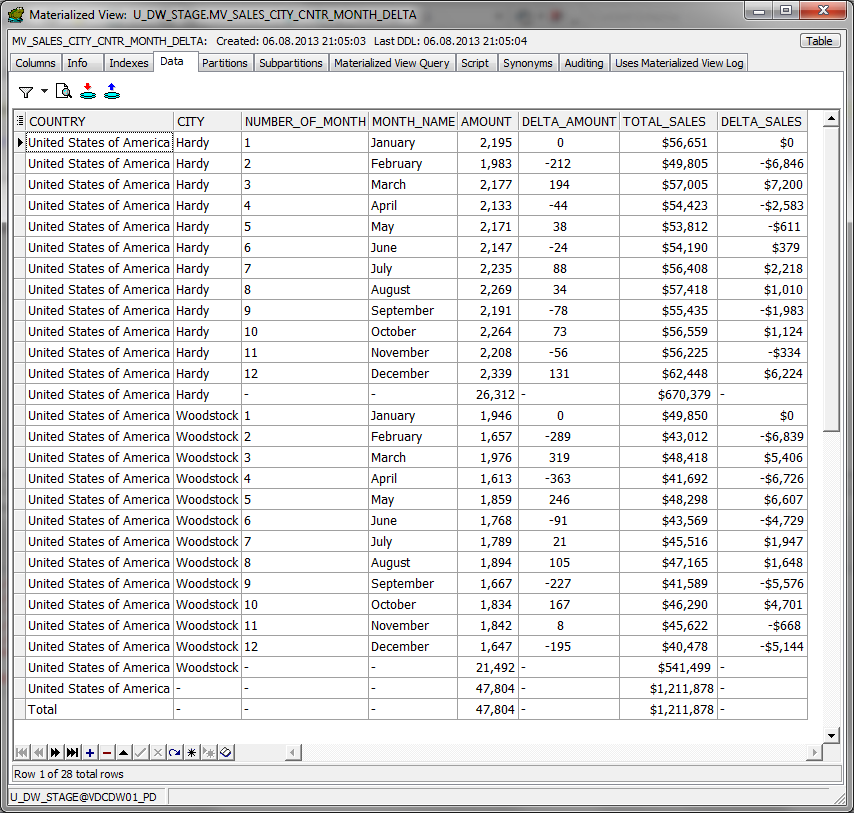
GROUP BY

ROLLUP (country\_name, city\_name, ( month\_numb, month\_name, delta\_amount, delta\_sales ) )

ORDER BY country\_name

, city\_name

, month\_numb);



I have run script (set price=price\*2 when restaurants in city Hardy):

UPDATE T\_OPERATIONS

set total\_price\_dol=total\_price\_dol\*2

where RESTAURANT\_ID in

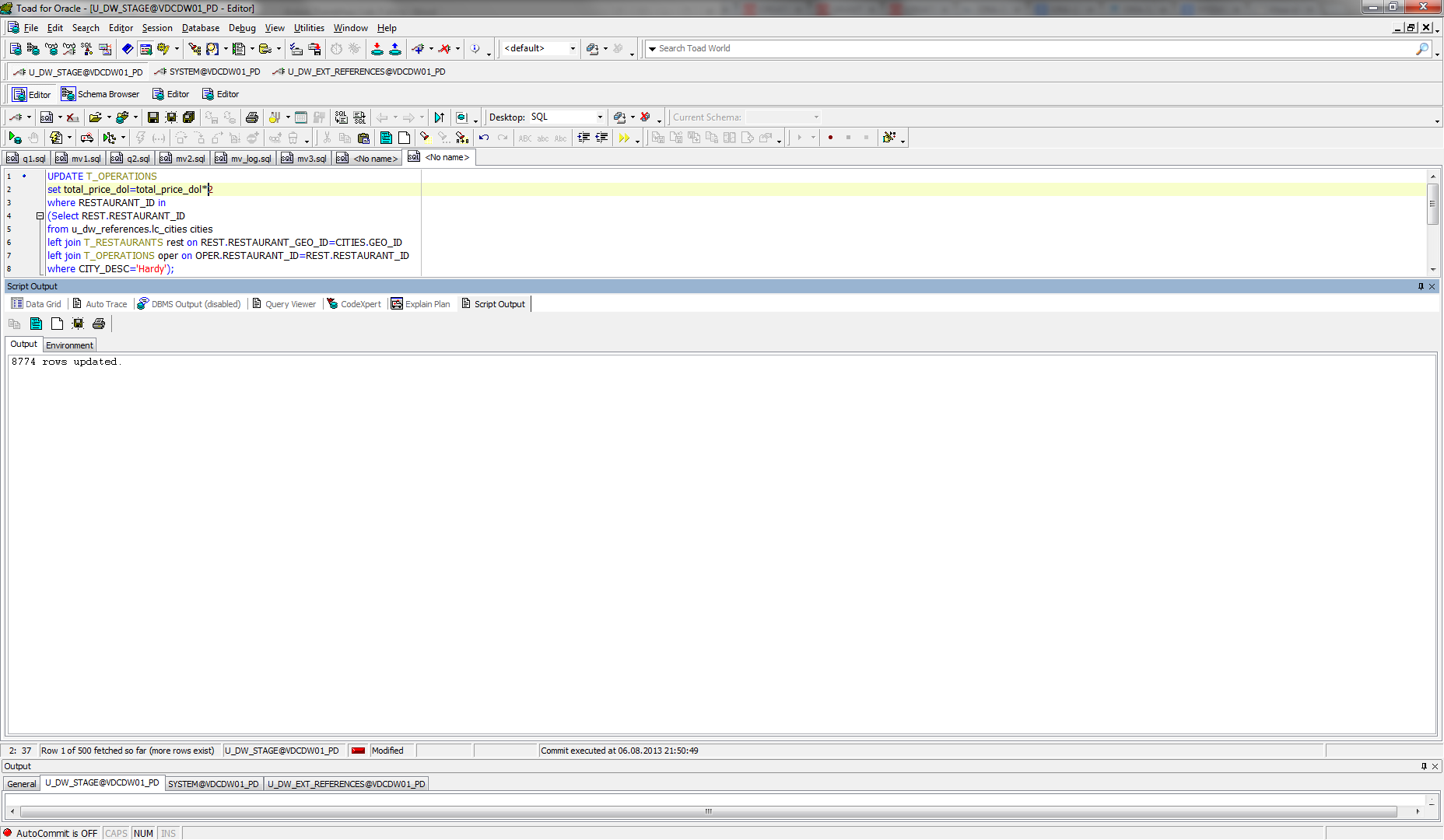
(Select REST.RESTAURANT\_ID

from u\_dw\_references.lc\_cities cities

left join T\_RESTAURANTS rest on REST.RESTAURANT\_GEO\_ID=CITIES.GEO\_ID

left join T\_OPERATIONS oper on OPER.RESTAURANT\_ID=REST.RESTAURANT\_ID

where CITY\_DESC='Hardy');



After 1 minute (=SYSDATE + 1/1440):

