**Report “Lab 4”**

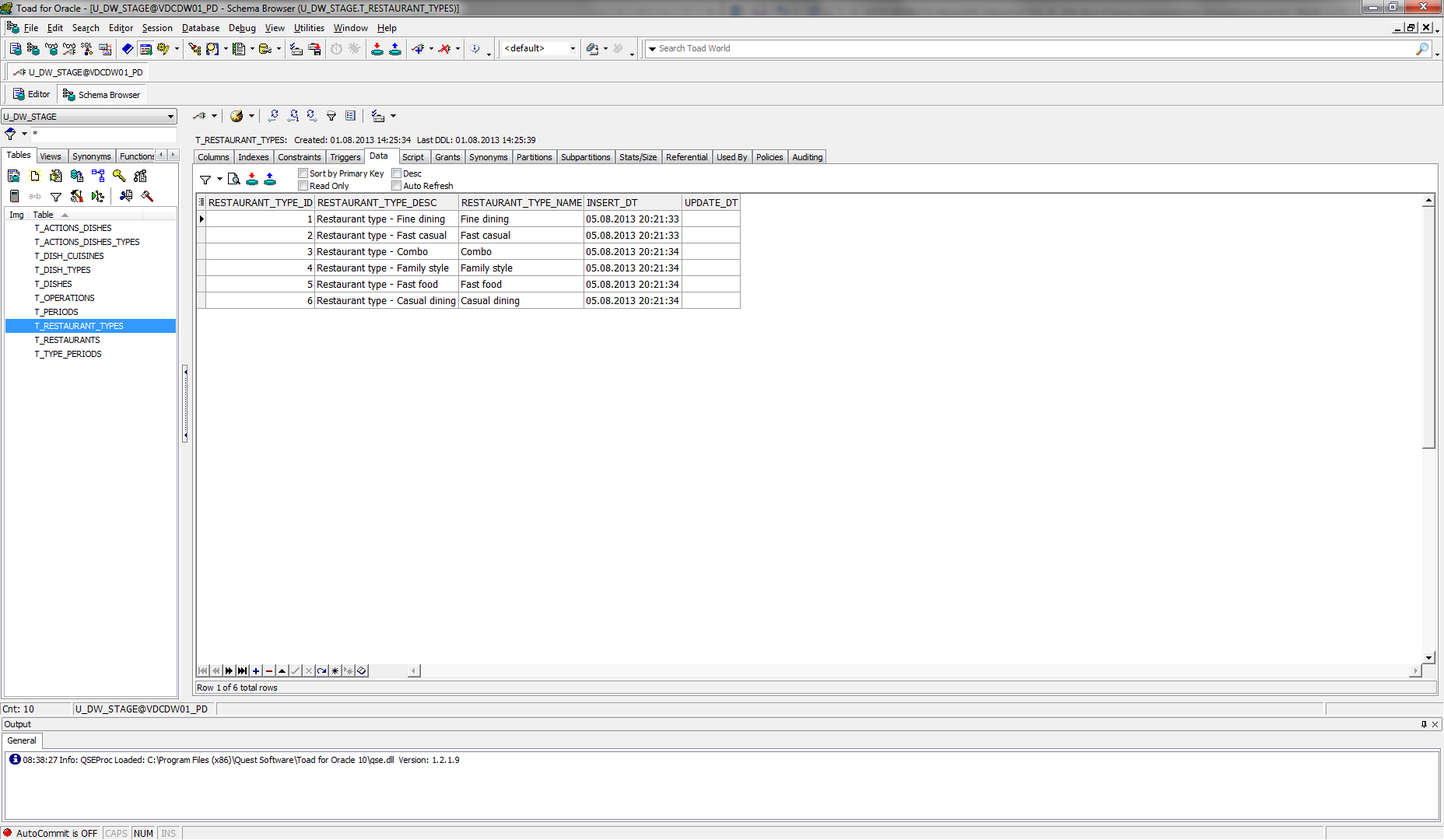
**Anton Tserakhau**

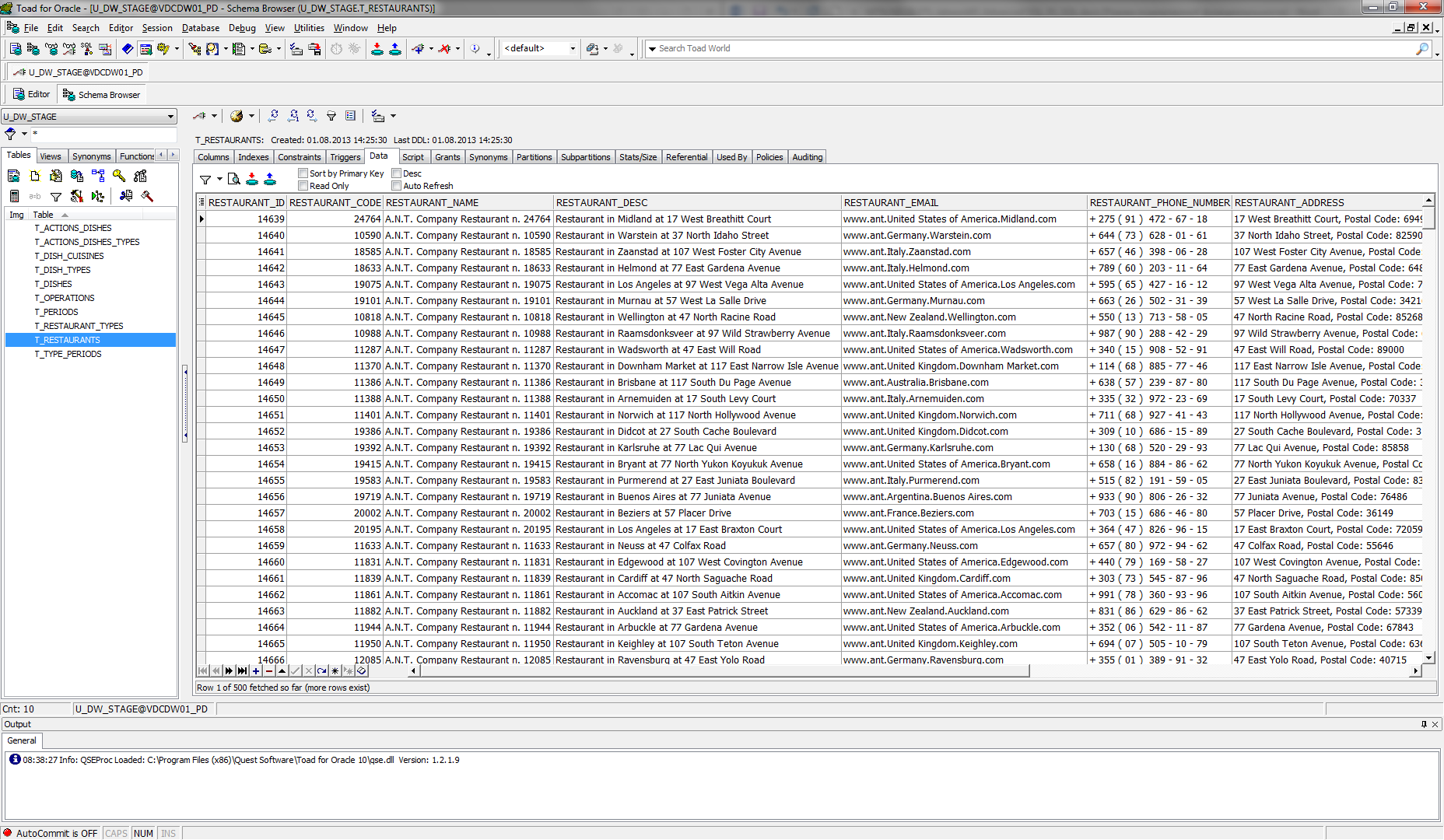
# Business analyses tasks – Dimensions

## Task 01: Create Packages for Reload Dimension from SA\_\*

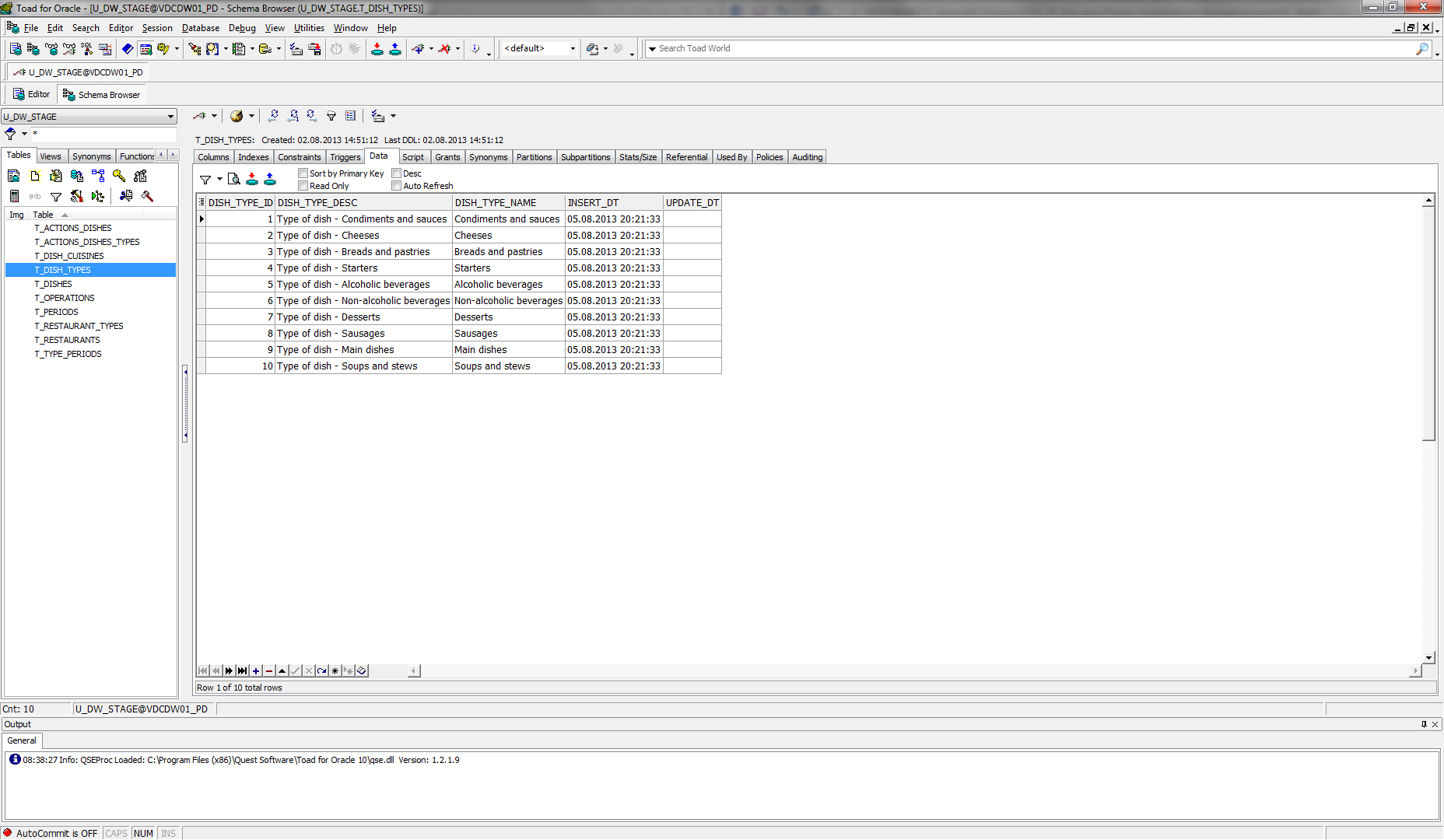
I have created independent packages to reload dimension.

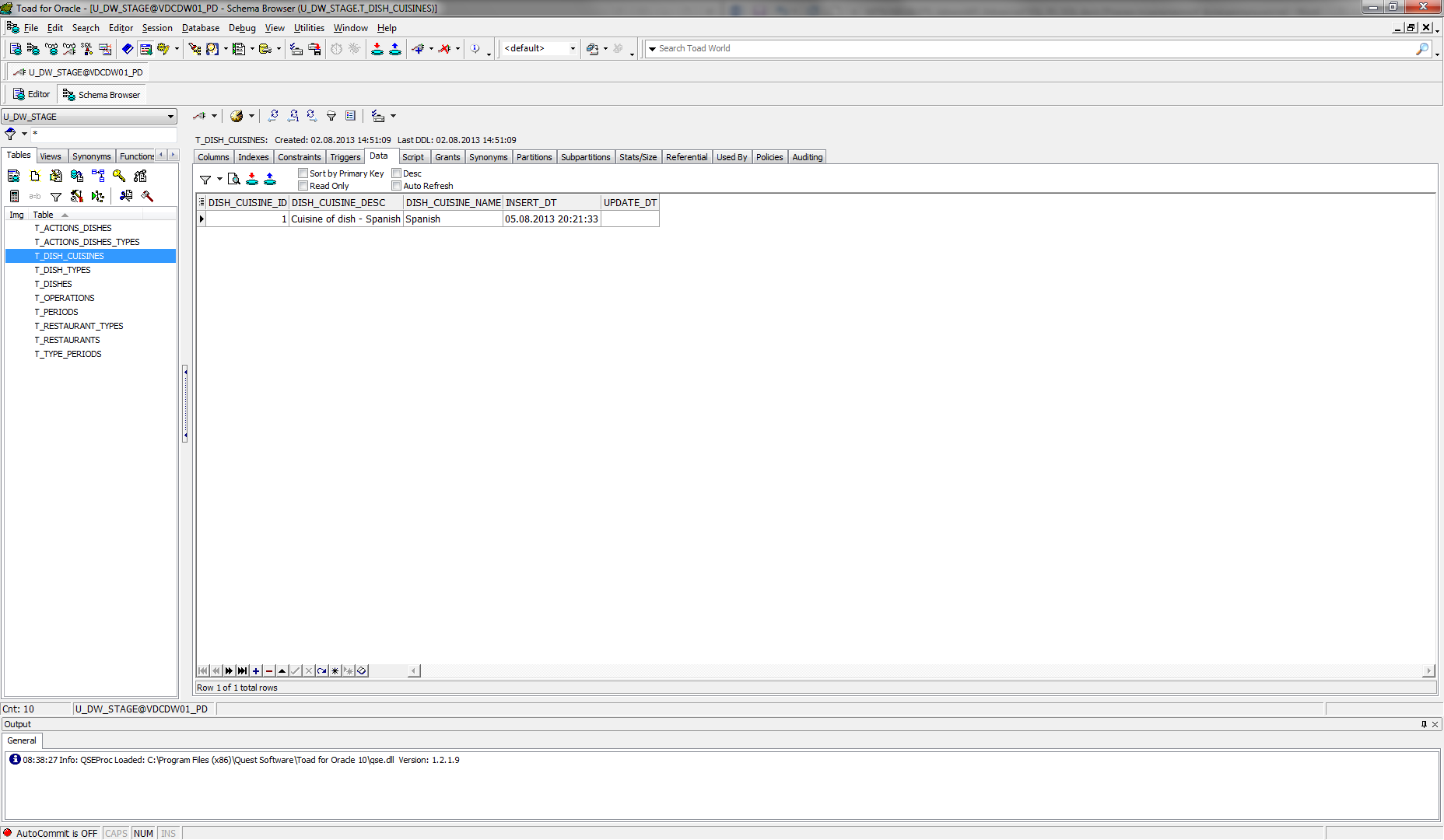
1. I have reload data to tables T\_RESTAURANT\_TYPES and T\_RESTAURANTS with using Execute Immediate with Bind Parameters (package *pkg\_etl\_restaurants\_dw\_stage*):

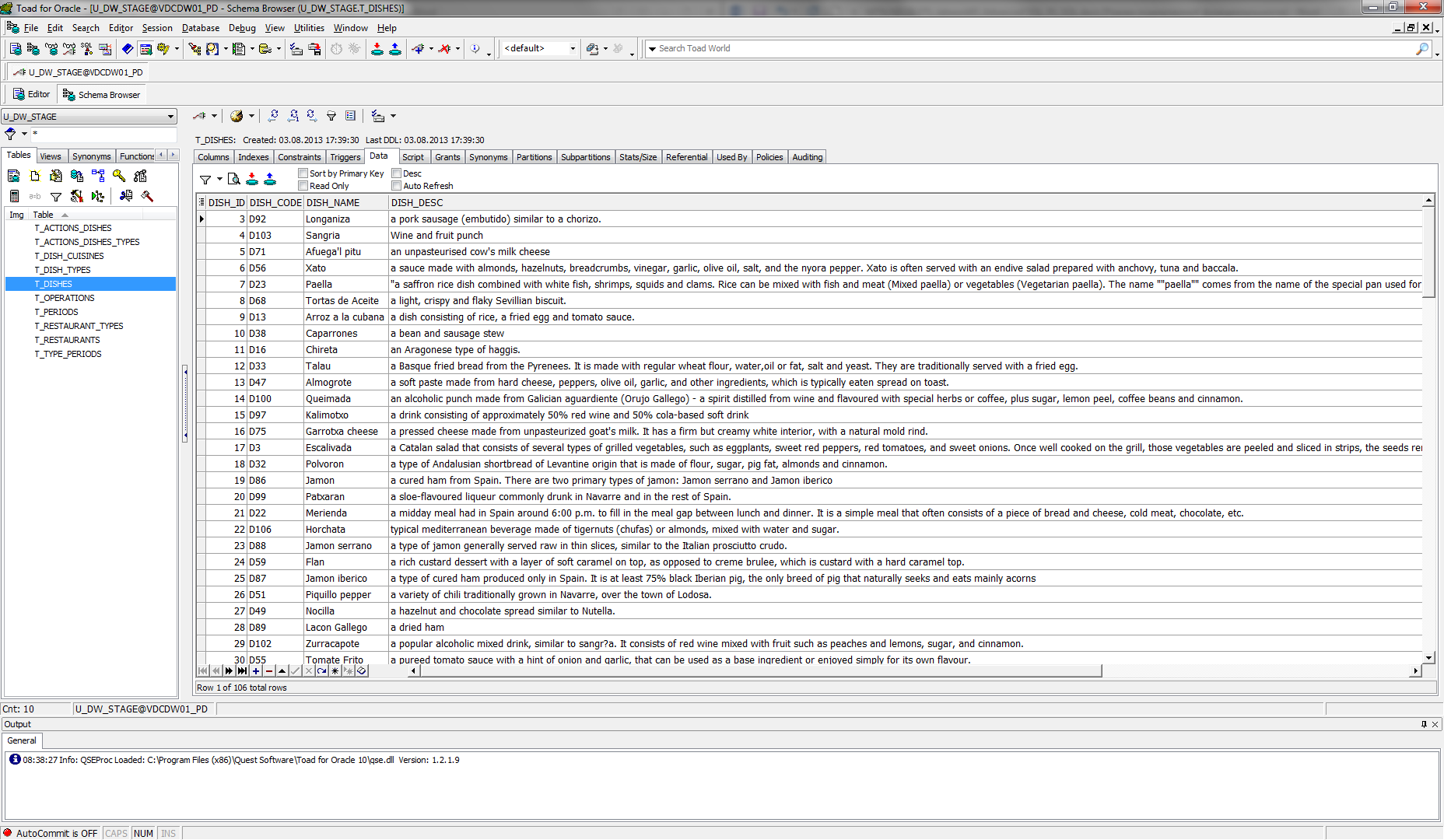




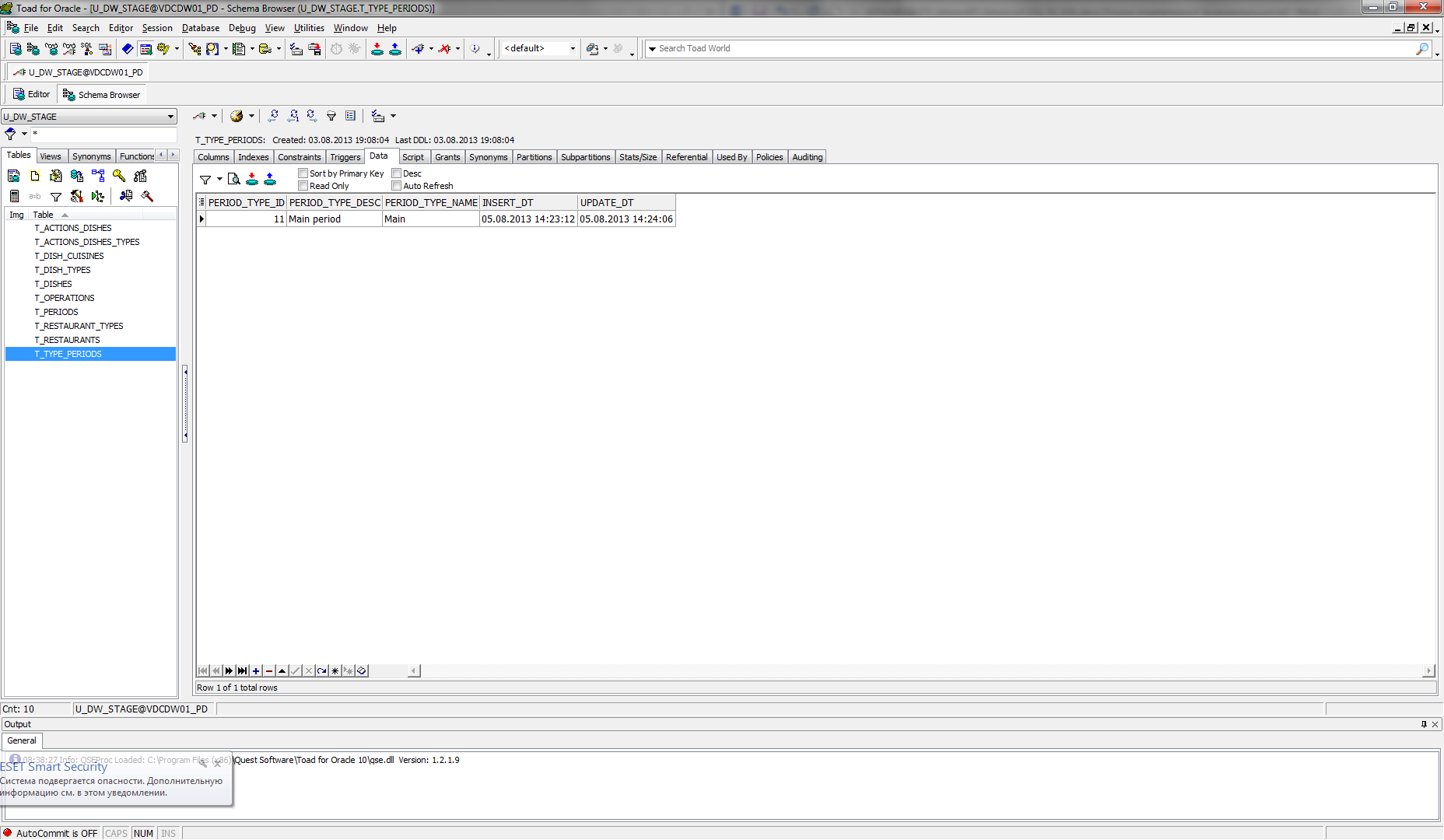
1. I have reload data to tables T\_DISH\_TYPES, T\_DISH\_CUISINES and T\_DISHES with using DBMS\_SQL.TO\_REFCURSOR Function (package *pkg\_etl\_dishes\_dw\_stage*):

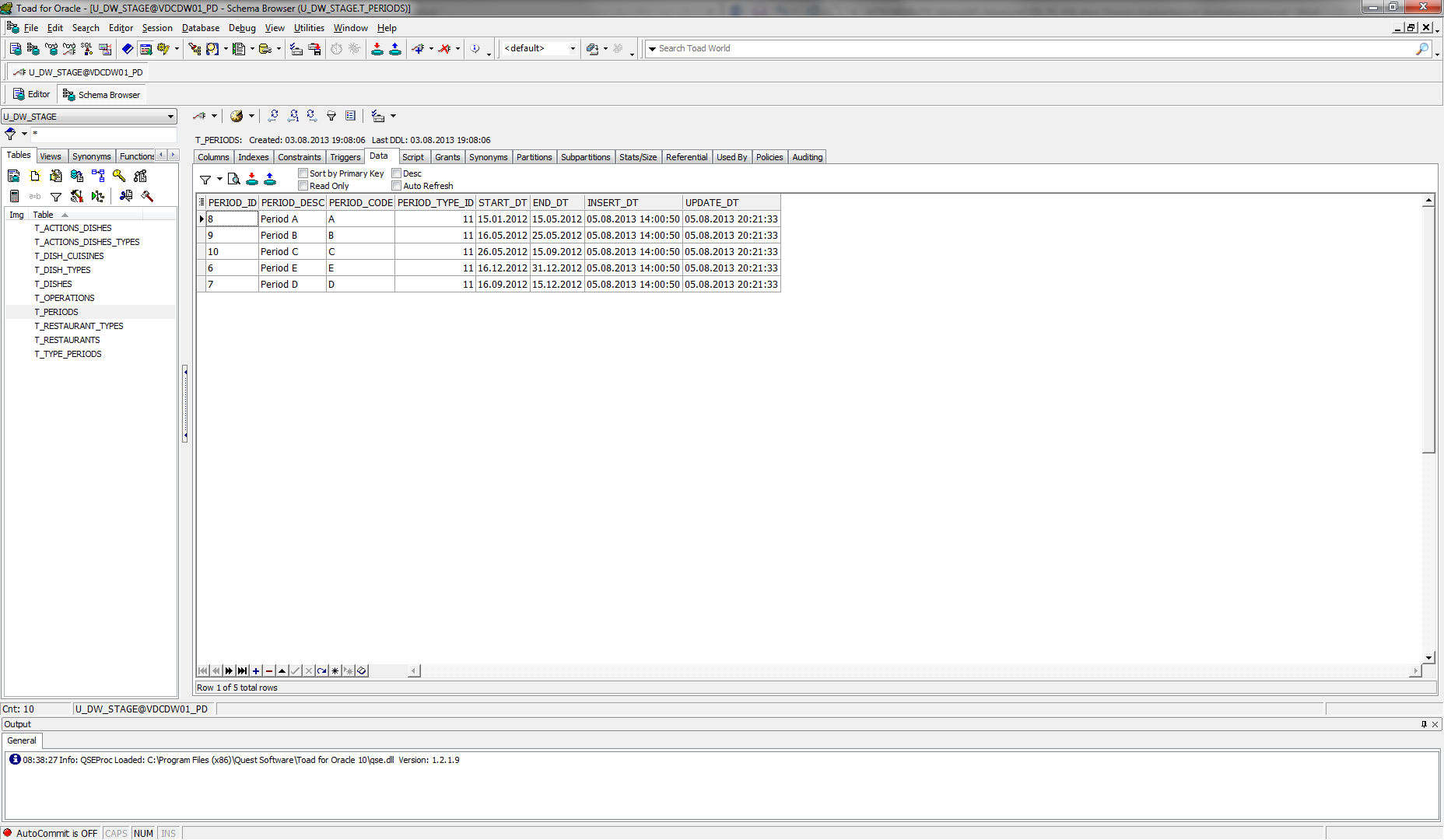






1. I have reload data to tables T\_TYPE\_PERIODS and T\_PERIODS with using DBMS\_SQL.TO\_CURSOR\_NUMBER Function (package *pkg\_etl\_periods\_dw\_stage*):





1. I have tested data for consistent. Scripts are in folder *test\_data.*

Table T\_DISH\_CUISINES:

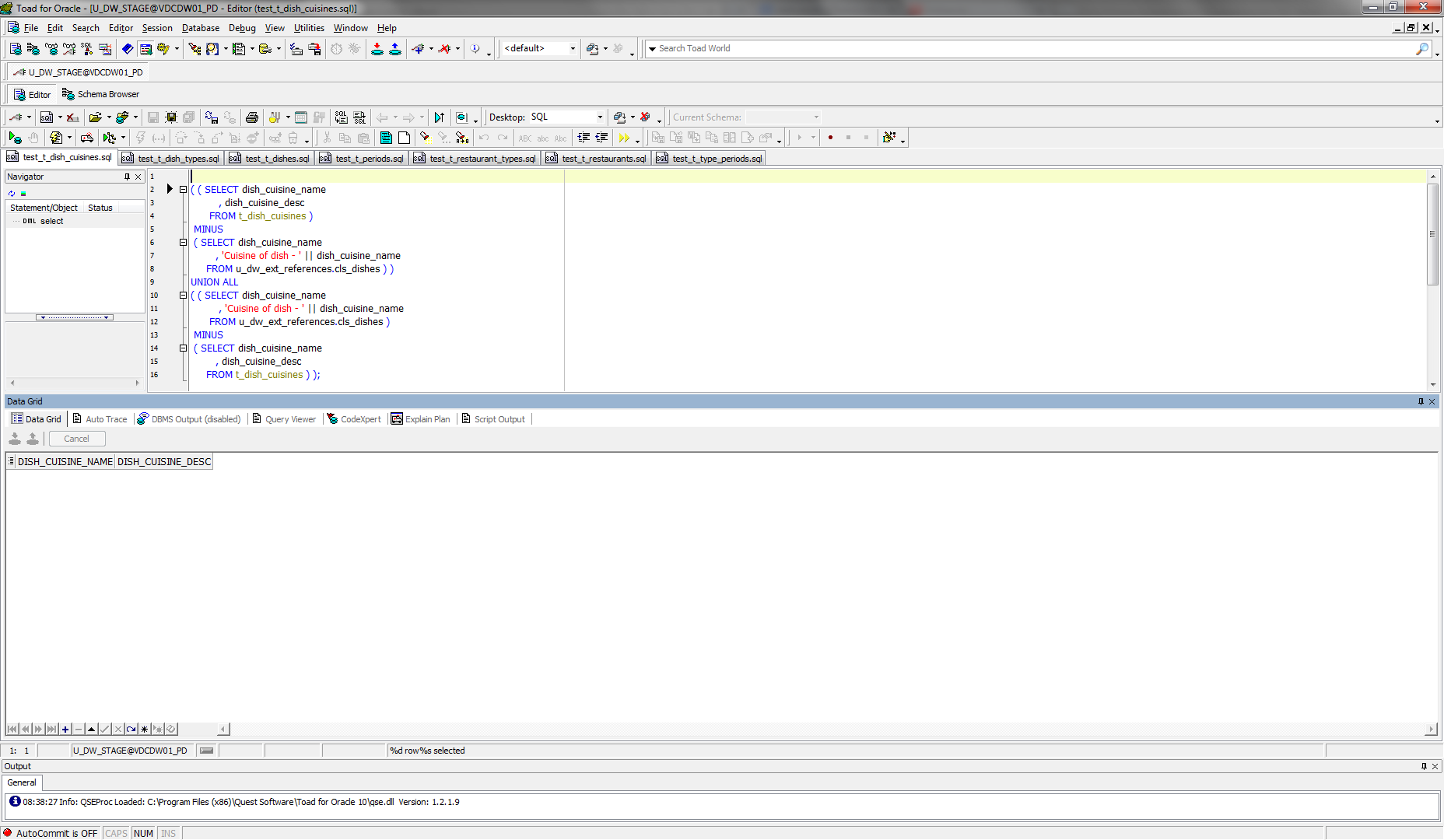


Table T\_DISH\_TYPES:

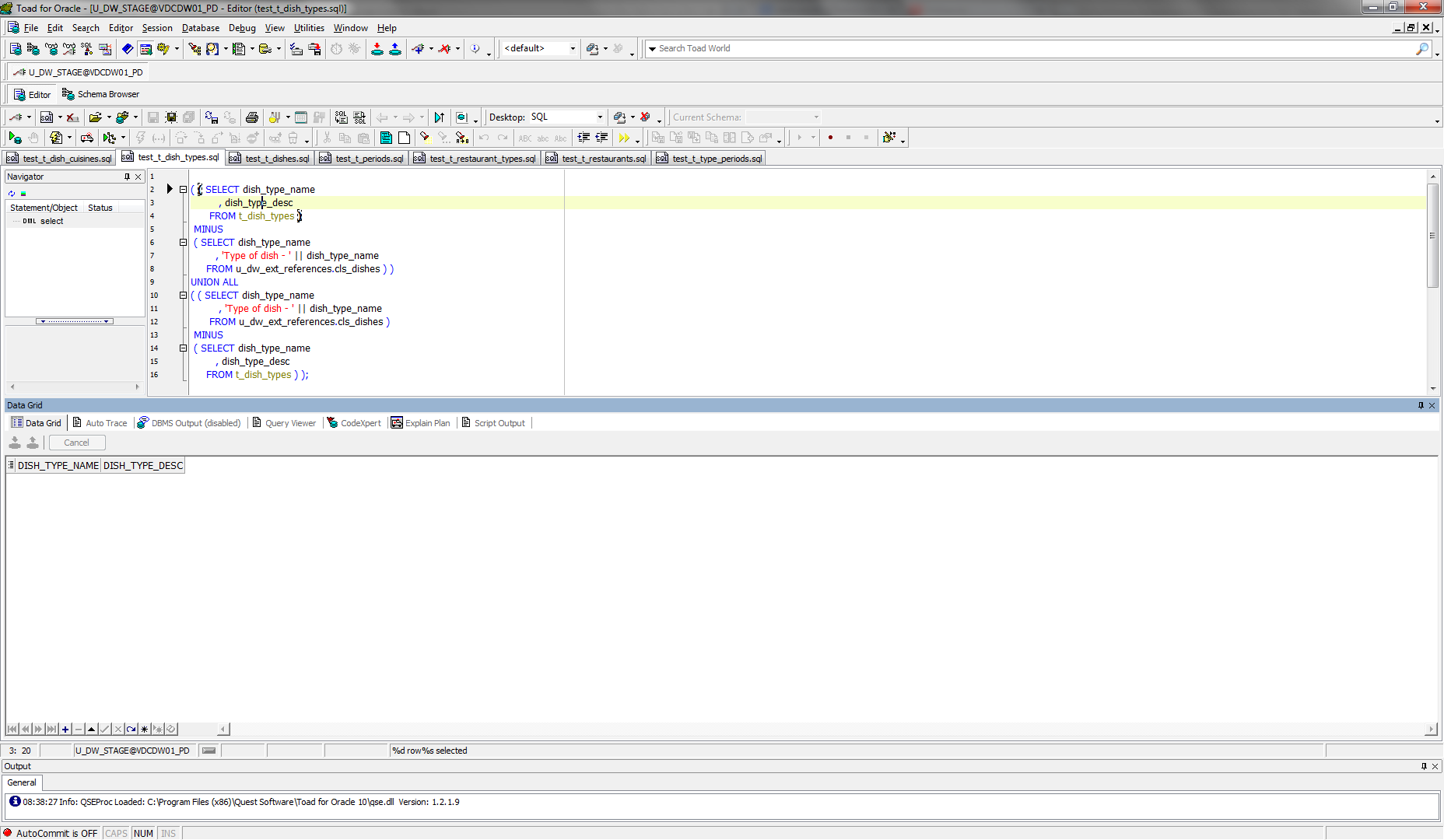


Table T\_DISHES:

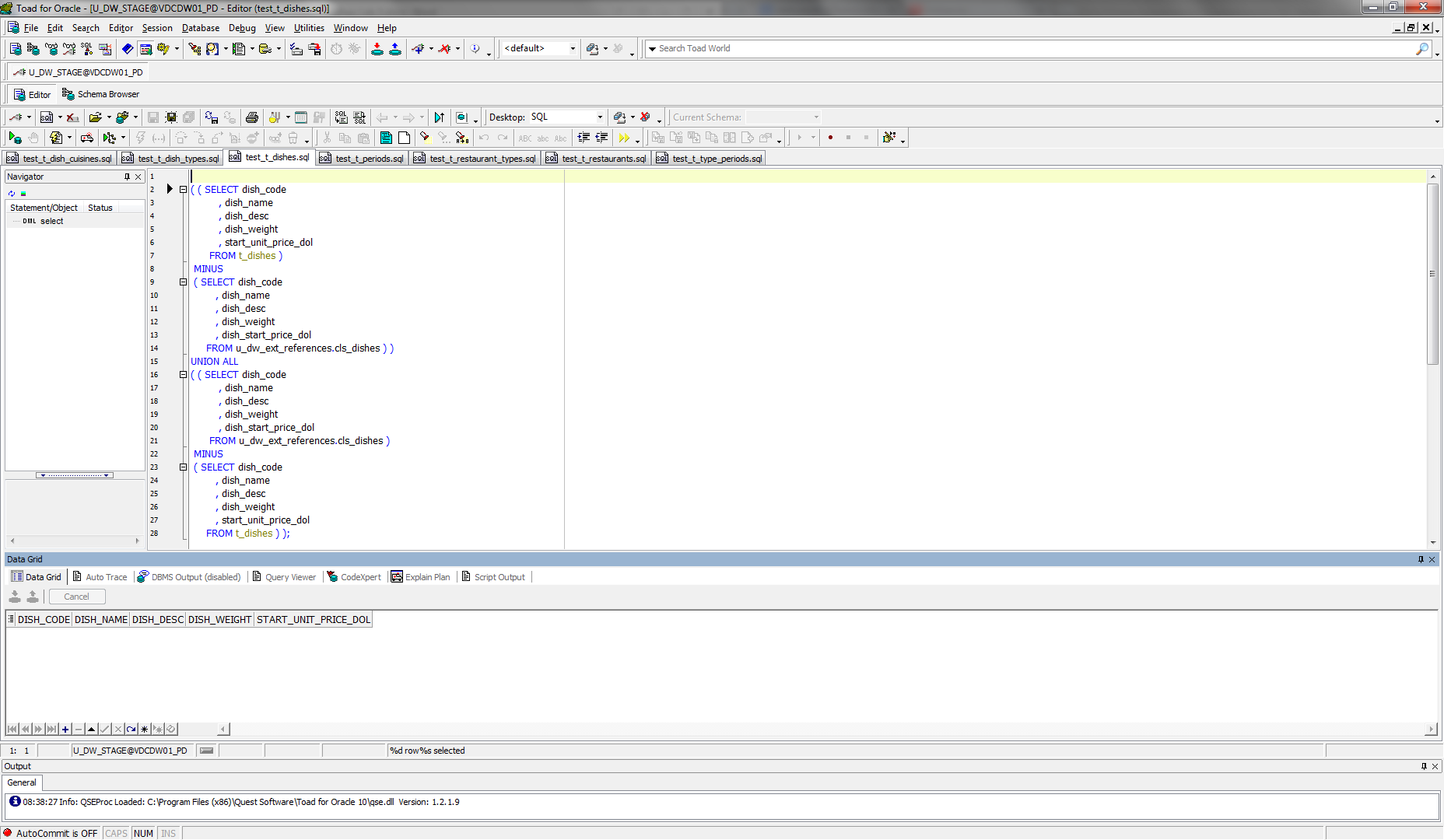


Table T\_PERIODS:

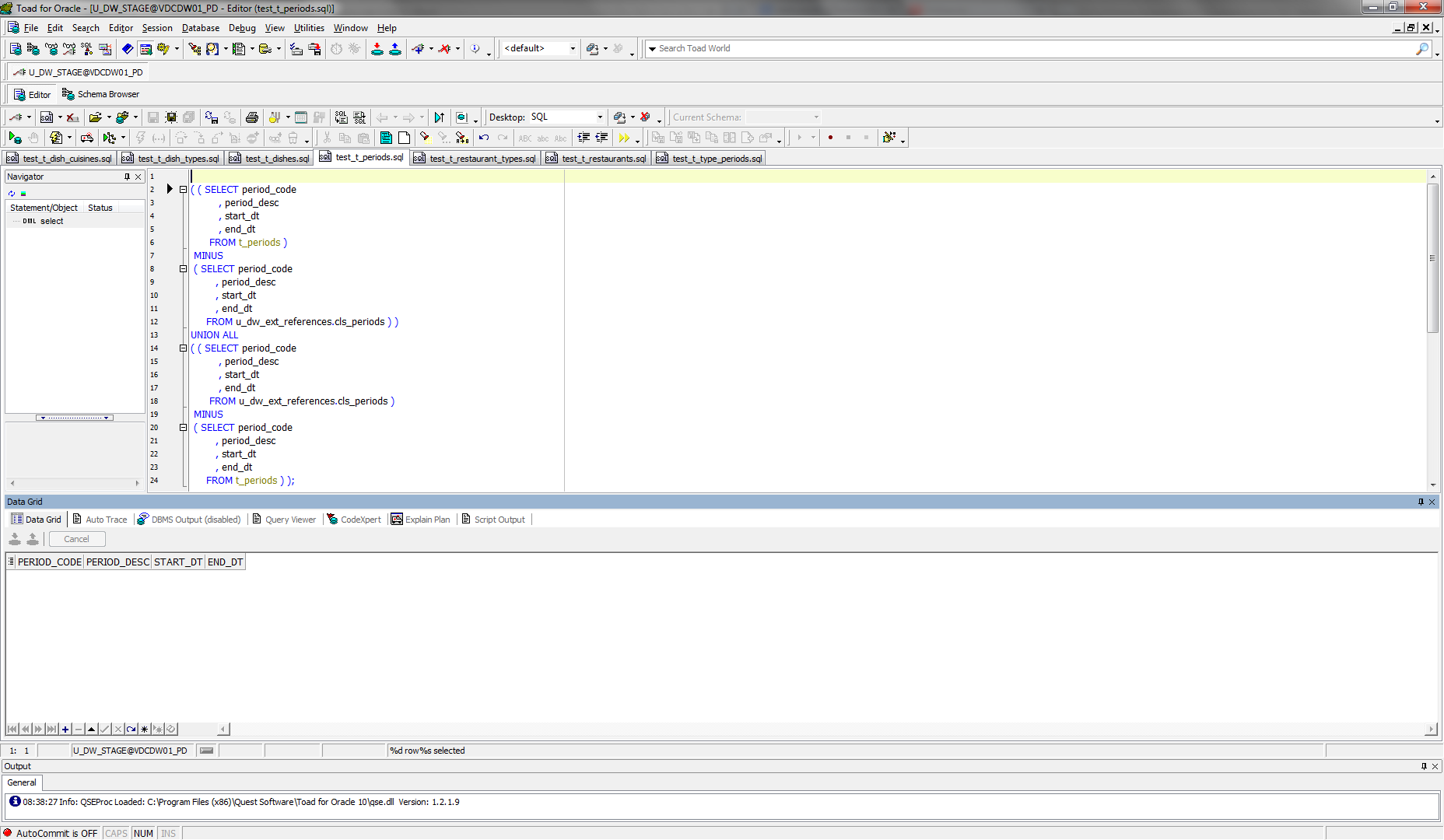


Table T\_RESTAURANT\_TYPES:

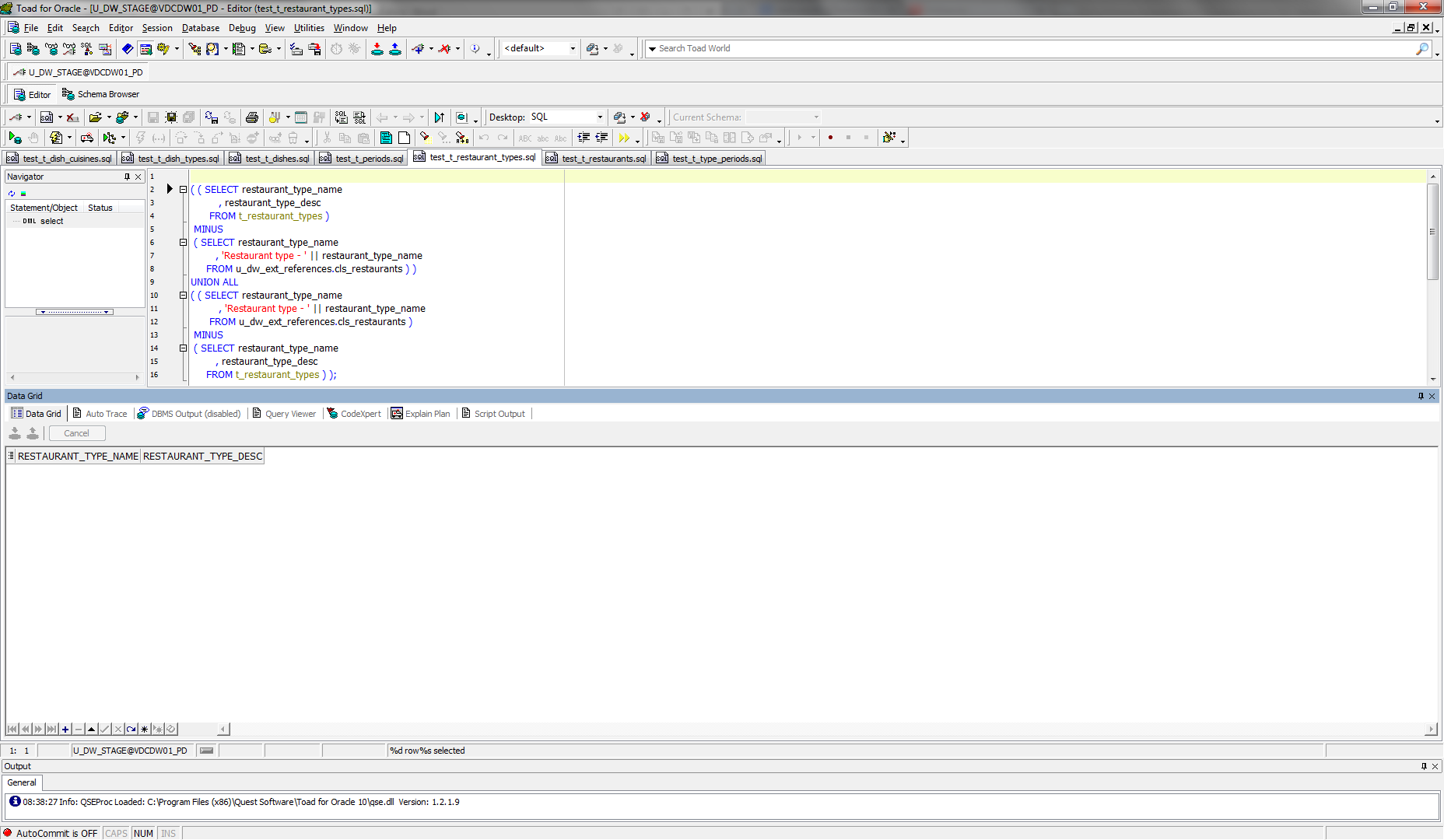


Table T\_RESTAURANTS:

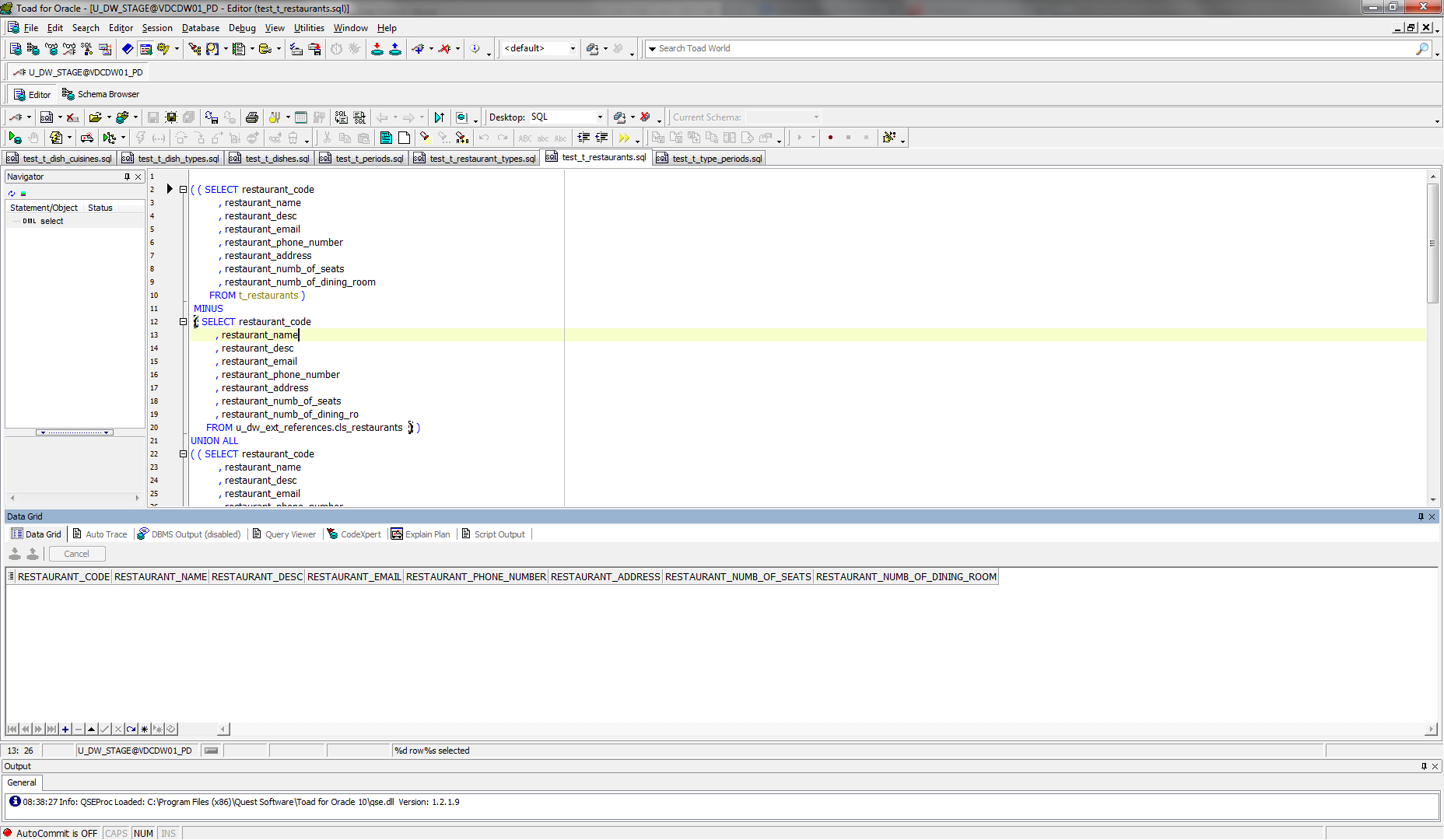
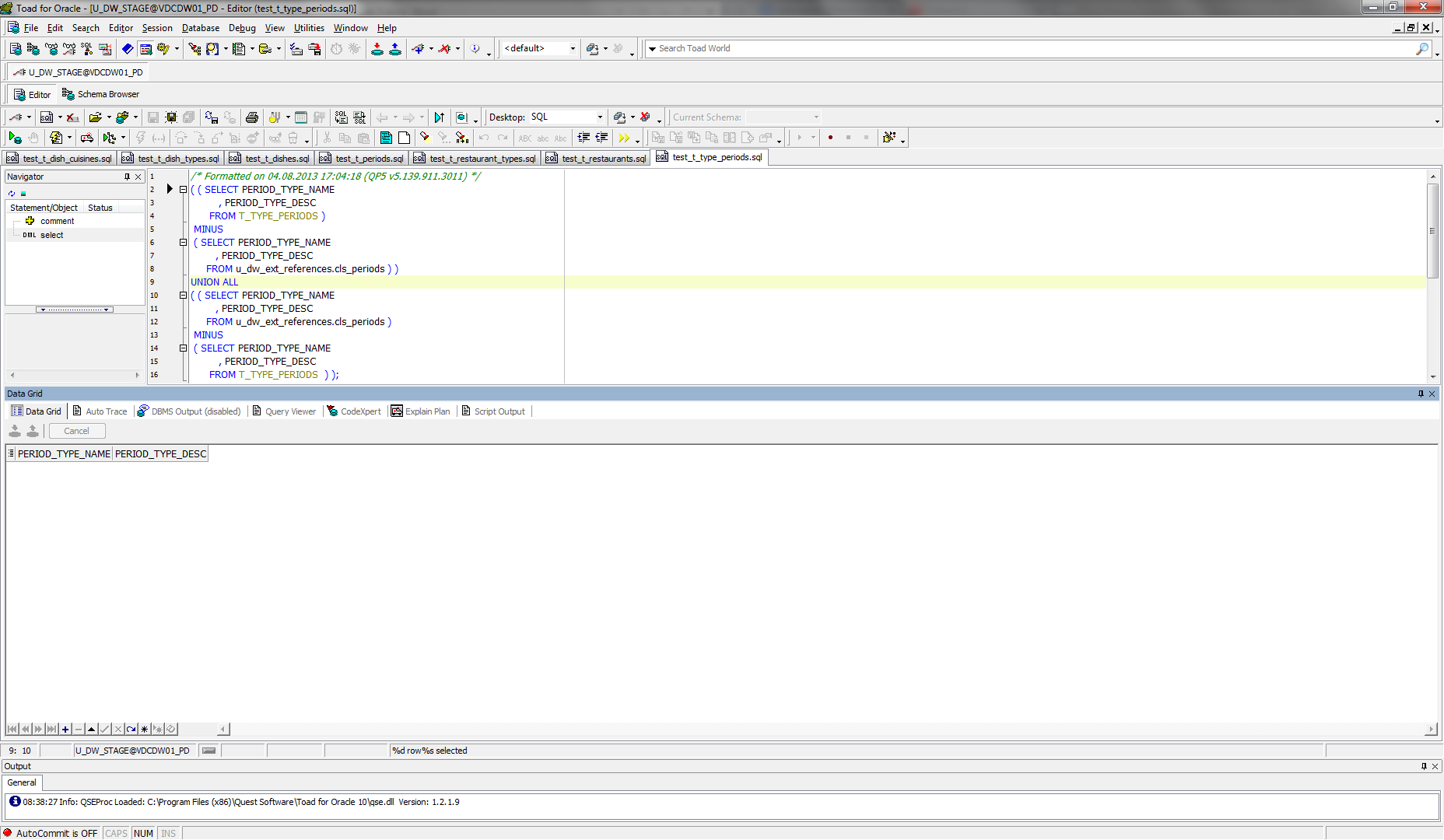


Table T\_TYPE\_PERIODS:



# Business analyses tasks – Reports

## Task 02: CREATE Monthly Reports Layouts

I have created Refactoring Adhoc SQL with using Module Clause.

Select necessary country and city (you can select all), month, the amount of dishes sold, the delta of dishes sold (the difference between the amounts of this and the previous month), the sales and delta of the sales (the difference between the sales of this and the previous month); display the total sums by the city, the country, all countries; sort by country in alphabetical order.

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)

