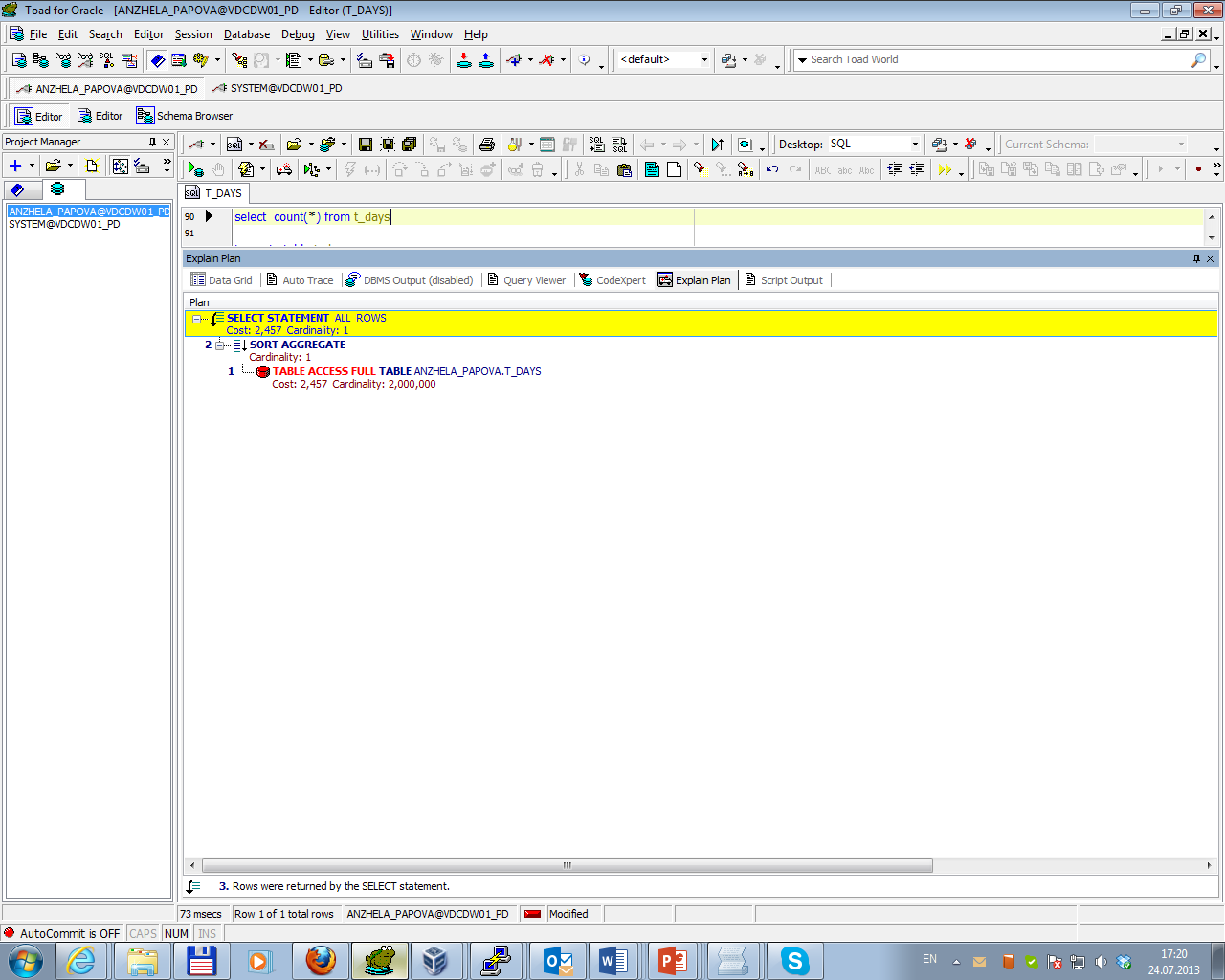
**Oracle Architecture - Parallel execution**

The application of parallel processing leads to increased resource consumption, as parallel execution attempts to use all available resources. Applying parallel execution is reasonable with a very large task and sufficient available resources.

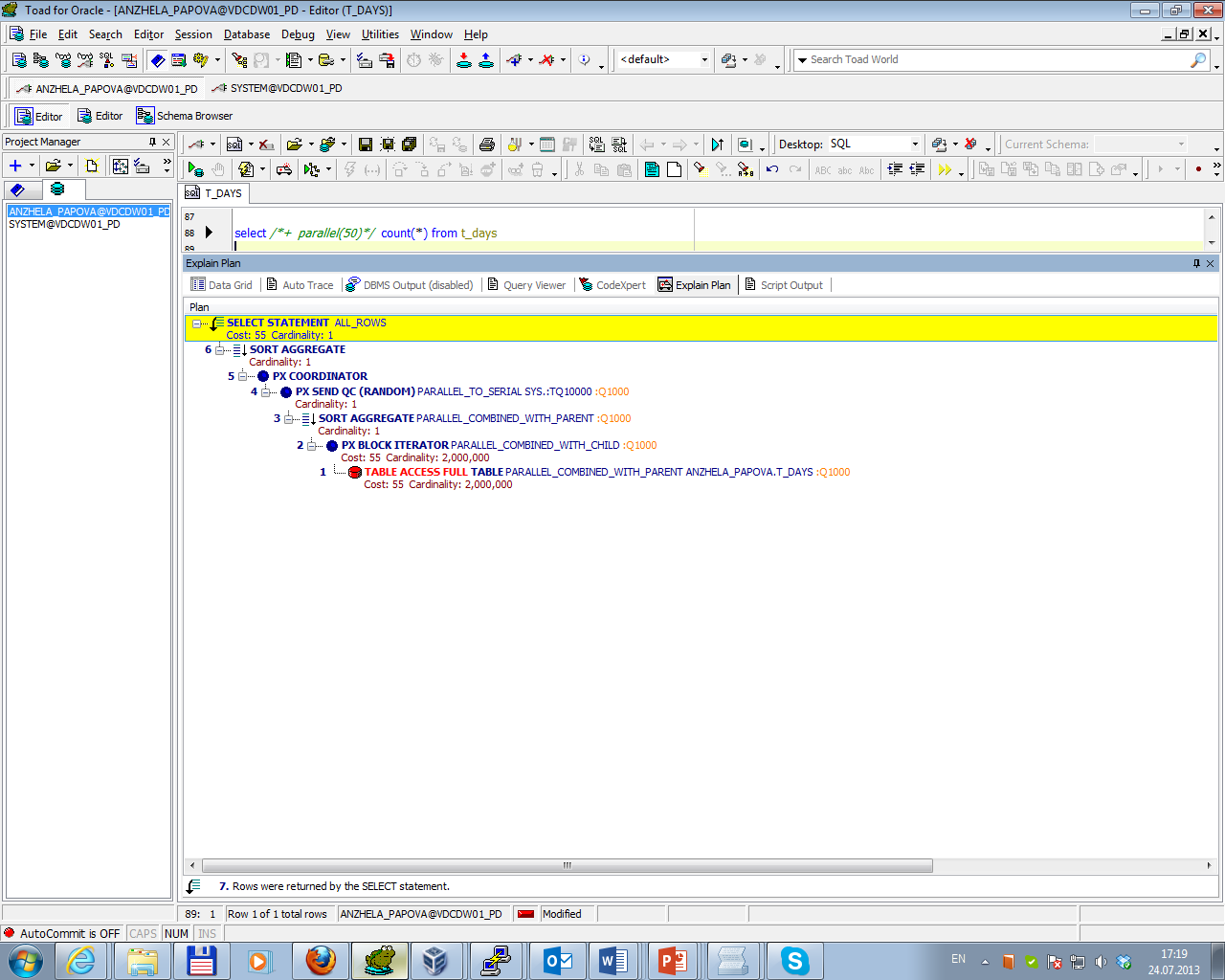
Task 01. CREATE Example of Select Parallel execution

To create example of select parallel execution table with 2 000 000 rows was created.

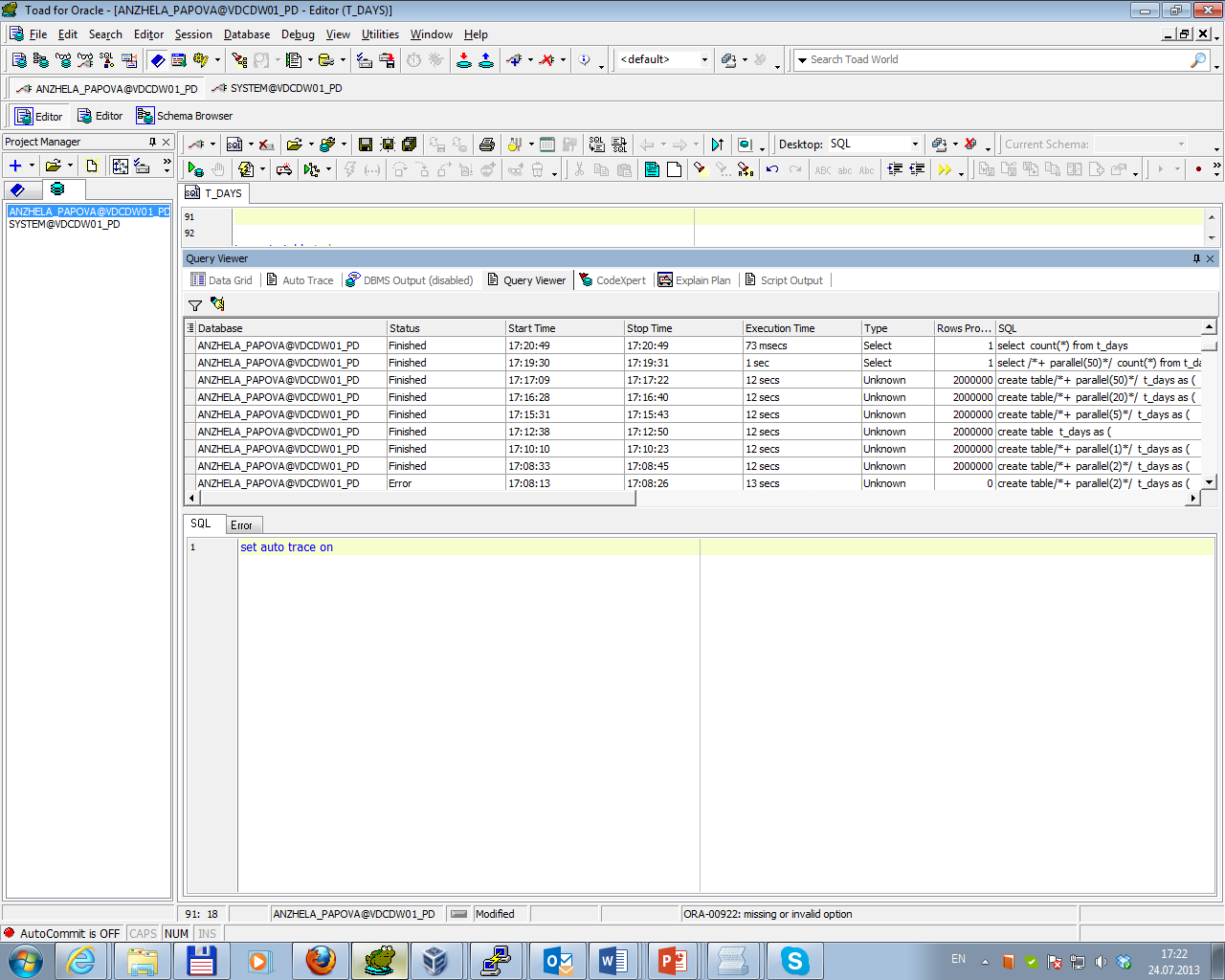
Common execution plan (single serial process) for select statement is presented below.



Then parallel query was running by using of a hint directly in the select statement. Execution plan for this case is presented below.



Difference in time query running is presented in Query Viewer below.



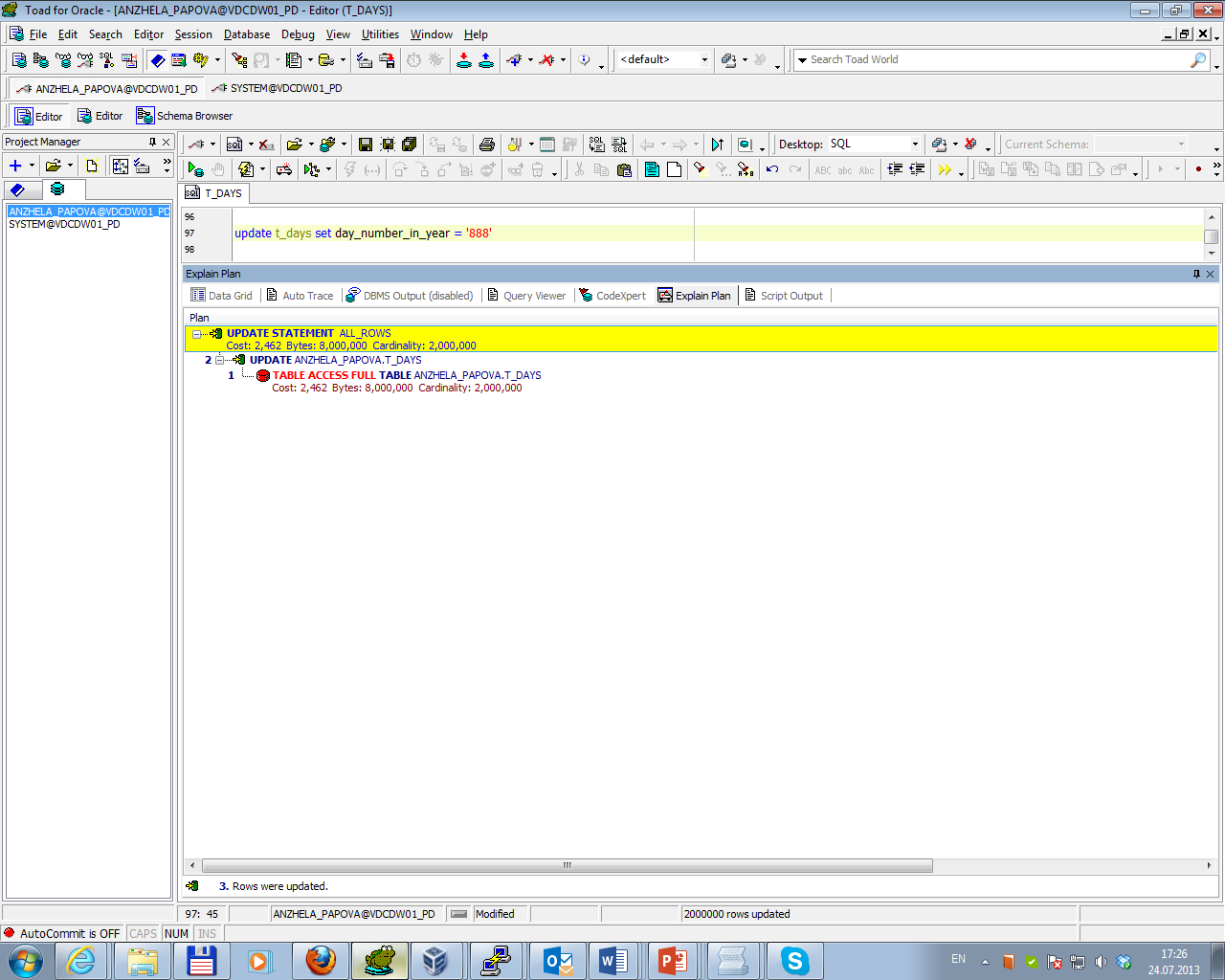
Comparison of operation execution is presented in [Summarize table](#Summ_table).

Task 02. CREATE Example of Parallel DML

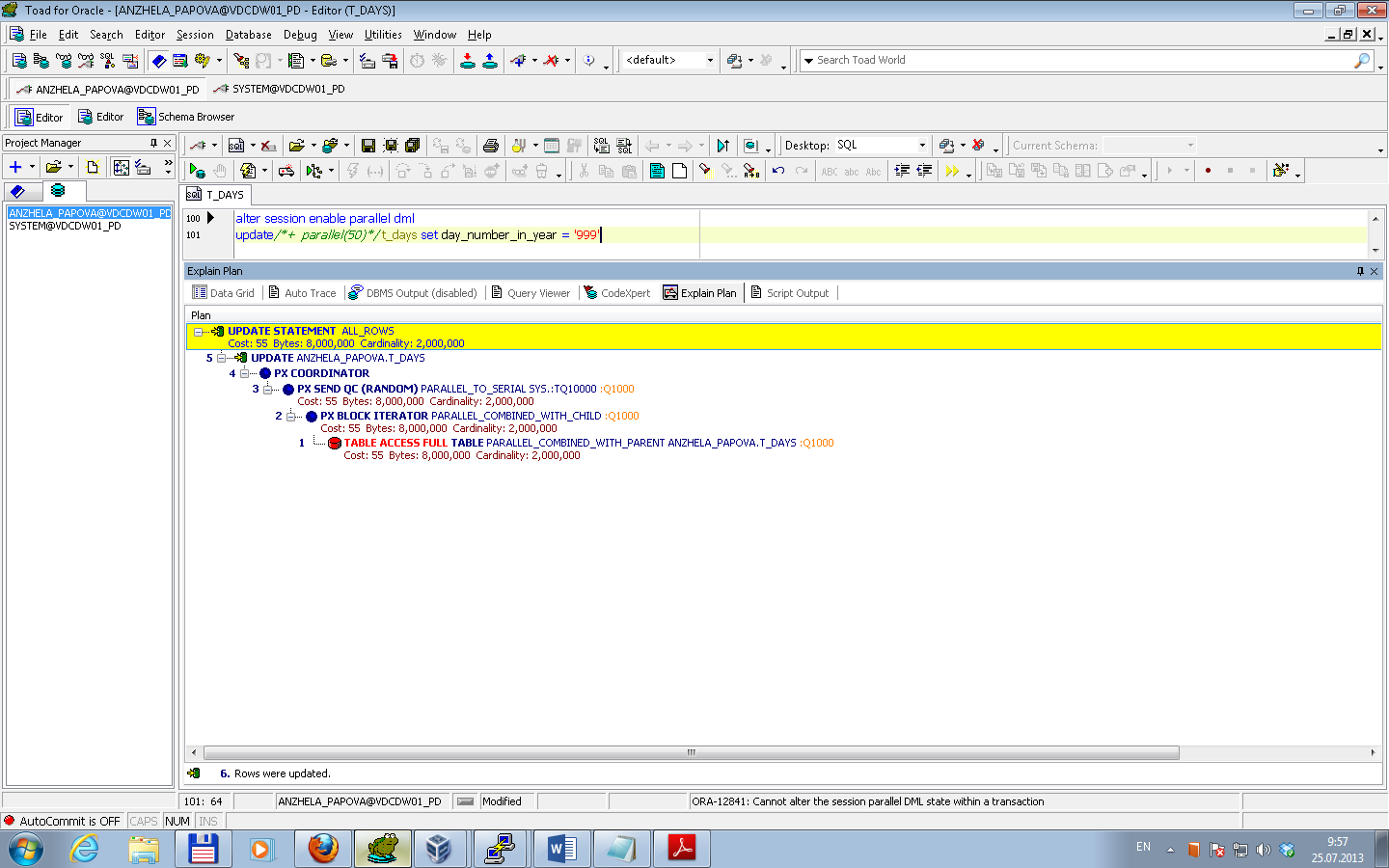
Parallel DML (PDML) is used in reference to performing modifications (INSERT, UPDATE, DELETE, and MERGE) using parallel processing. PDML is useful in a large data warehousing environment to facilitate bulk updates to massive amounts of data.

Update statement was chosen to test using Parallel DML.

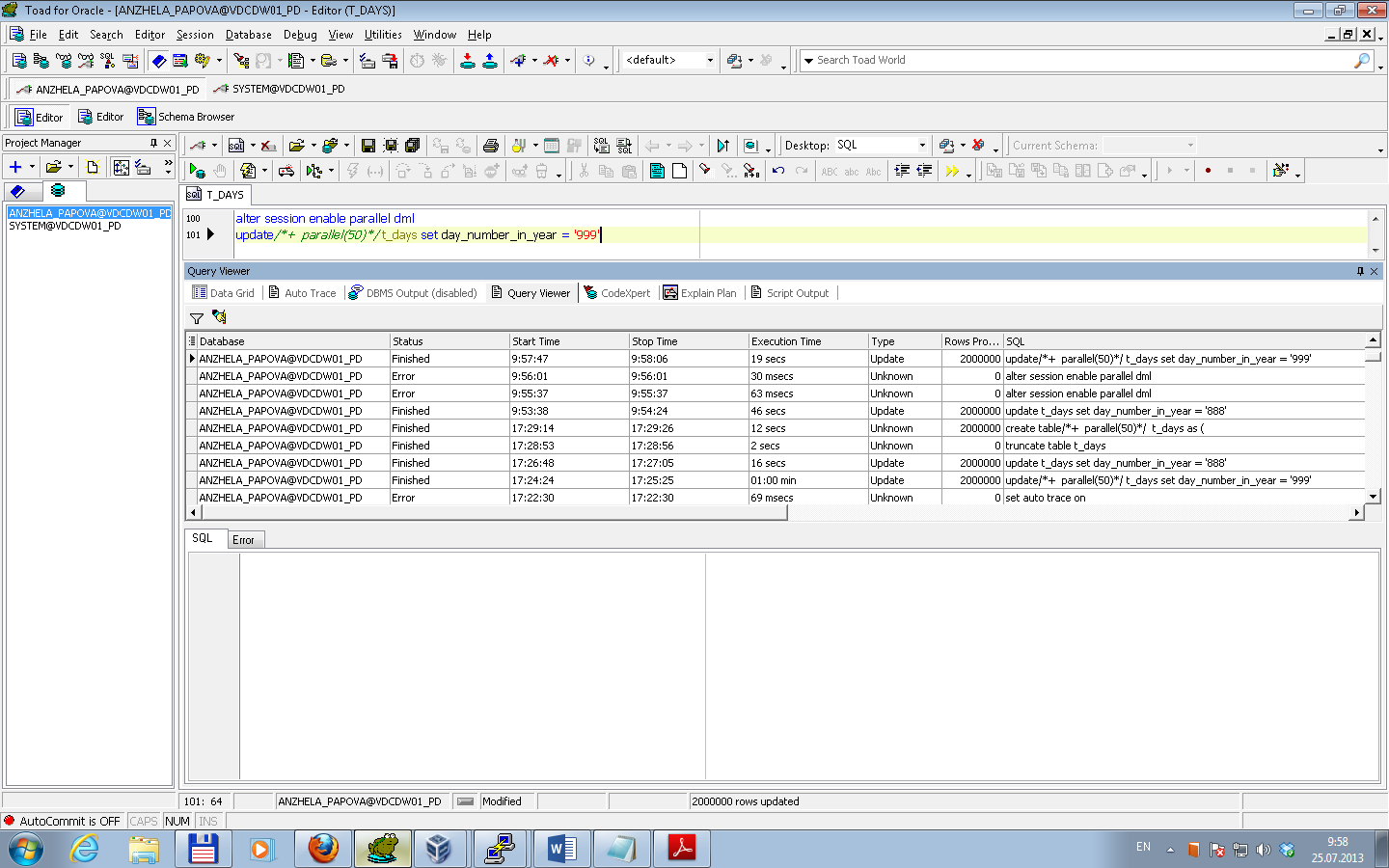
Execution plan for single serial process of update statement is presented below.



Then parallel DML was running by using of a hint directly in the update statement. Execution plan for this case is presented below.



Difference in time statement running is presented in Query Viewer below.



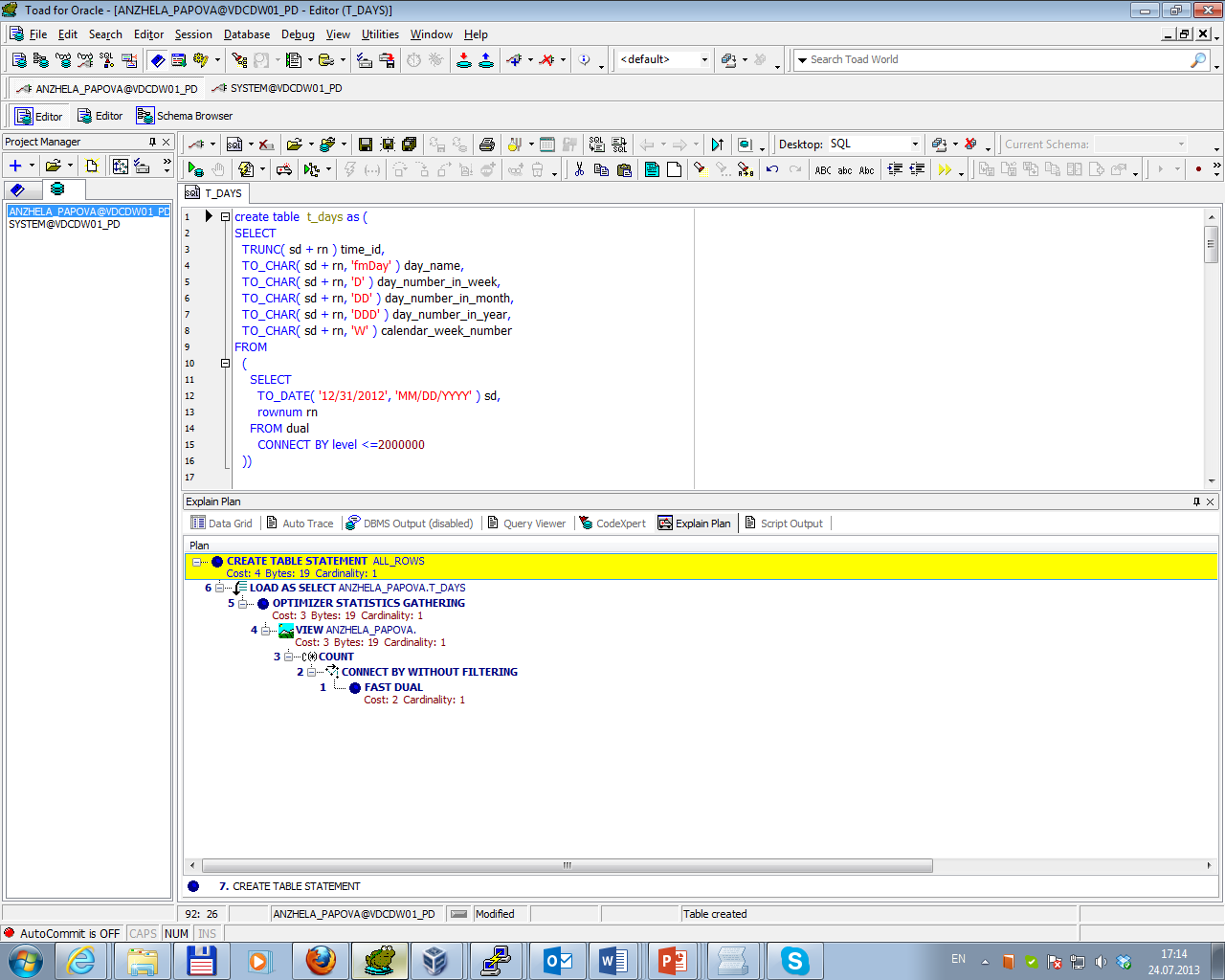
Comparison of operation execution is presented in [Summarize table](#Summ_table).

Task 03. CREATE Example of Parallel DDL

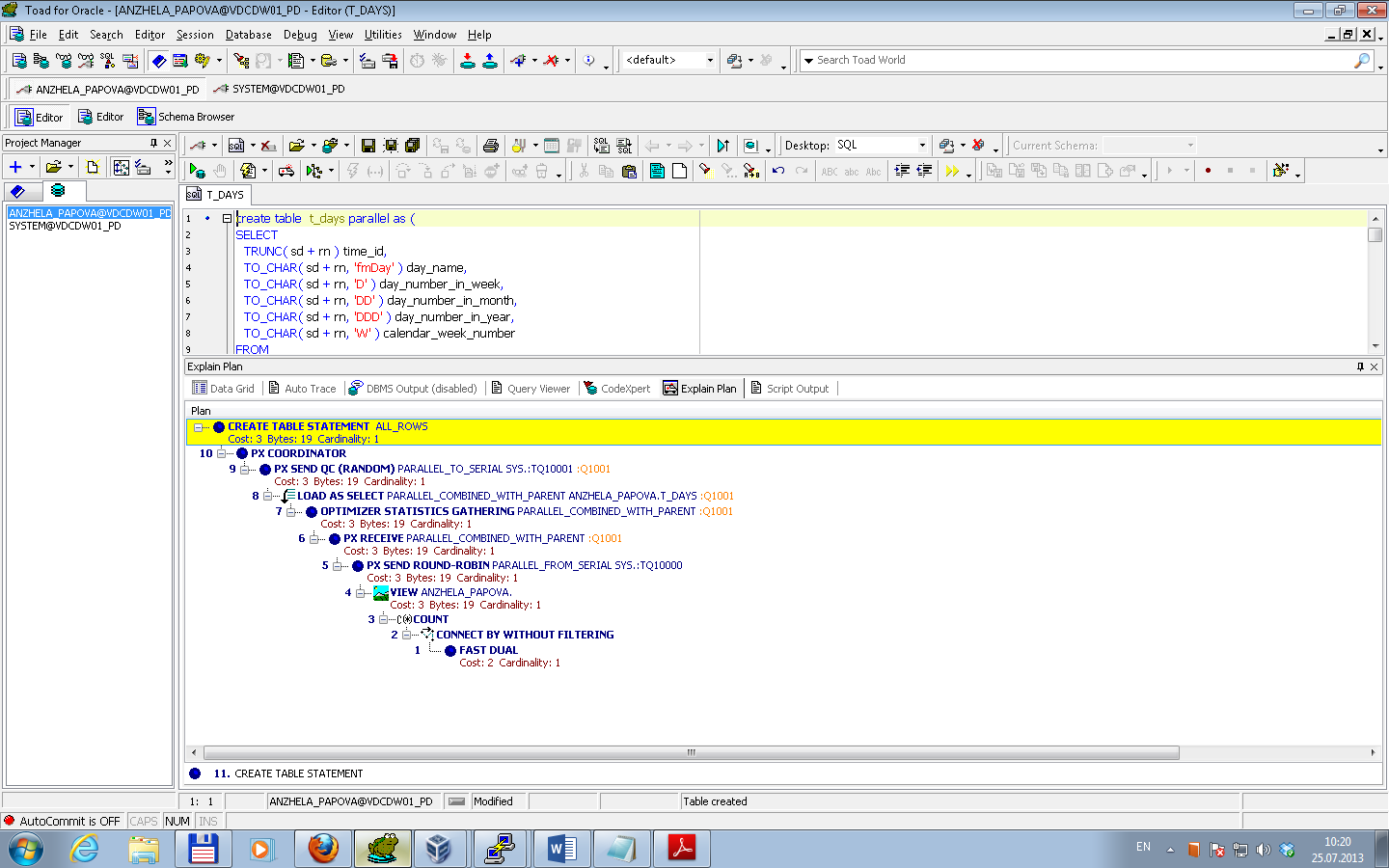
Parallel DDL is a tool for the DBA and developer alike to quickly perform those large maintenance operations typically done during off-peak times when resources are available. Parallel DDL performs large DDL operations in parallel.

Statement CREATE TABLE … AS SELECT was chosen to test using Parallel DDL.

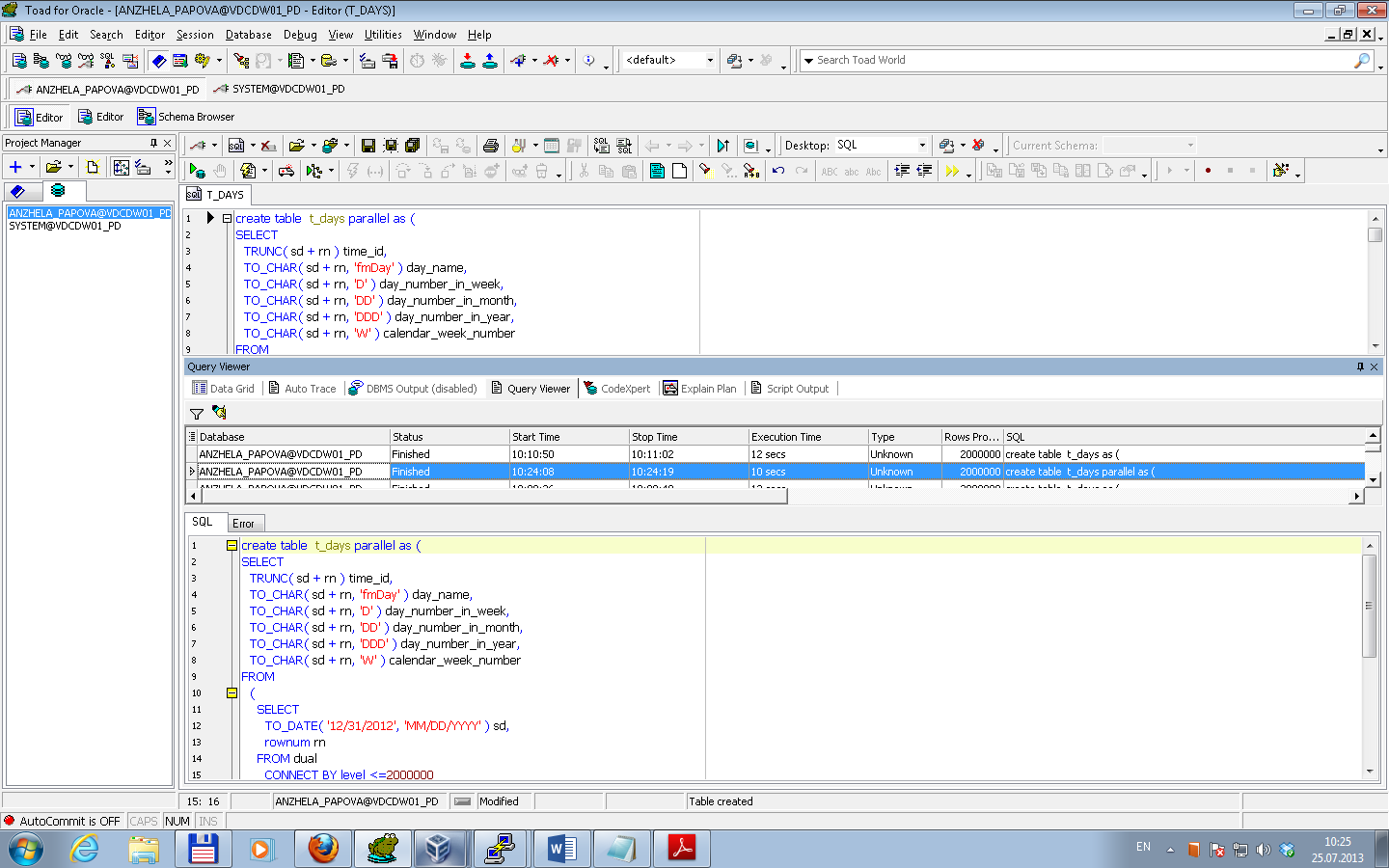
Execution plan for single serial process of such statement is presented below.



Then parallel DDL was running by command create table <…> parallel…. Execution plan for this case is presented below.



Time statement running is presented in Query Viewer below.



Comparison of operation execution is presented in [Summarize table](#Summ_table).

**Summarize table**

| Type | Parallel degree | Cost | Time | Description |
| --- | --- | --- | --- | --- |
| Parallel Query | single | 2 457 | 73msecs | Query uses number of parallel sessions (equal to parallel degree), break table into small, non-overlapping slices, and then ask each parallel session to read the table and count its section of rows. The parallel query coordinator receives each of the aggregated counts from the individual parallel sessions and further aggregate them, returning the final answer to the client application  Thus in case full table access cost for single serial process was 2 457 and in case partially table access – 55  *So, Parallel query execution works best when given access to as many resources as possible* |
| parallel(50) | 55 | 1sec |
| Parallel DML | single | 2 462 | 46secs | When parallel DML is used each slice of the table is modified as a separate transaction in the separate session. After they are all done is performed to commit the separate, independent transactions.  Thus full table access cost for single serial process was 2 462 and in case partially table access – 55.  But there are some of limitations associated with PDML (triggers are not supported and others).  *So, using Parallel DML is reasonable considering limitations associated with it* |
| parallel(50) | 55 | 19secs |
| Parallel DDL | single | 4 | 12secs | With Parallel DDL the query that executes the SELECT was executed using parallel query, and the table load itself was done in parallel.  Cost of statement execution as a single process was 4; as a parallel process – 3.  *So, Parallel DDL is the best choice for DBAs and developers to improve database administration* |
| parallel(auto) | 3 | 10secs |