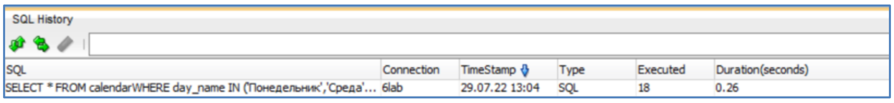
Lab report #10
Sadovskaya Veronika

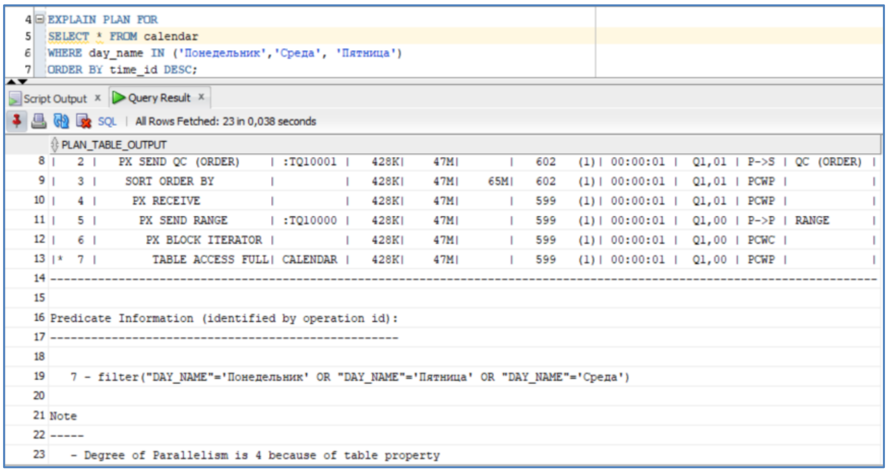
Because Parallel is designed to optimize and speed up big data, a table with 1 million records was created for this lab. Because query results are cached, the table was dropped and recreated for each query.

Task 1 - CREATE Example of Select Parallel execution

1) Select without parallel:



SQL	Connection	TimeStamp	Type	Executed	Duration(seconds)
SELECT * FROM calendarWHERE day_name IN ('Понедельник','Среда'...	6lab	29.07.22 13:04	SQL	18	0.26

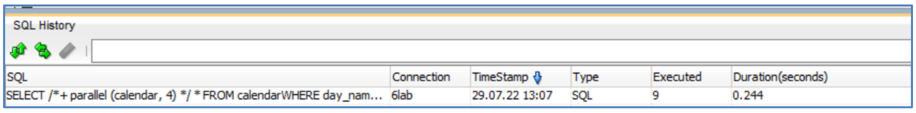


```
EXPLAIN PLAN FOR
SELECT * FROM calendar
WHERE day_name IN ('Понедельник','Среда', 'Пятница')
ORDER BY time_id DESC;
```

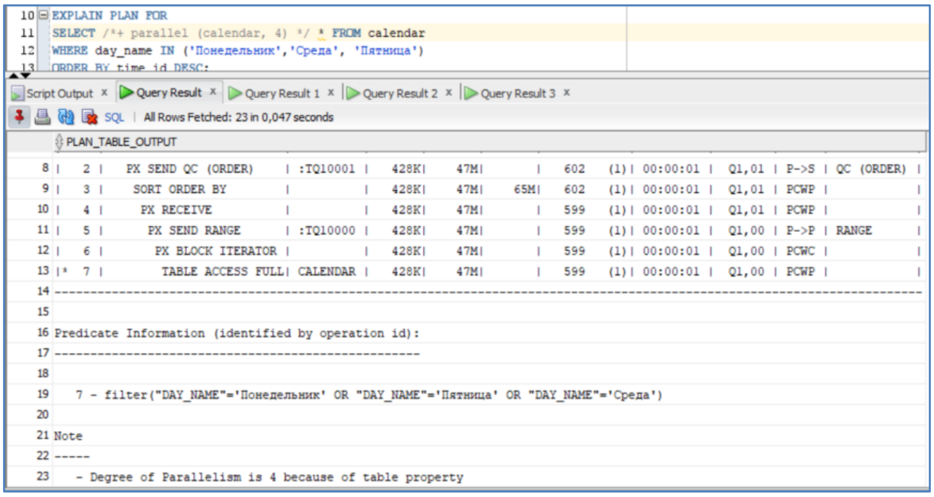
All Rows Fetched: 23 in 0,038 seconds

```
PLAN_TABLE_OUTPUT
|   2 |   PX SEND QC (ORDER)      | :TQ10001 |   428K|   47M|      |   602   (1)| 00:00:01 |  Q1,01 | P->S | QC (ORDER) |
|   3 |    SORT ORDER BY          |          |   428K|   47M|   65M|   602   (1)| 00:00:01 |  Q1,01 | PCWP |            |
|   4 |     PX RECEIVE            |          |   428K|   47M|      |   599   (1)| 00:00:01 |  Q1,01 | PCWP |            |
|   5 |      PX SEND RANGE        | :TQ10000 |   428K|   47M|      |   599   (1)| 00:00:01 |  Q1,00 | P->P | RANGE      |
|   6 |       PX BLOCK ITERATOR   |          |   428K|   47M|      |   599   (1)| 00:00:01 |  Q1,00 | PCWC |            |
|*  7 |        TABLE ACCESS FULL| CALENDAR |   428K|   47M|      |   599   (1)| 00:00:01 |  Q1,00 | PCWP |            |
-----------------------------------------------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------

   7 - filter("DAY_NAME"='Понедельник' OR "DAY_NAME"='Пятница' OR "DAY_NAME"='Среда')

Note
-----
   - Degree of Parallelism is 4 because of table property
```

2)Select with parallel:



SQL	Connection	TimeStamp	Type	Executed	Duration(seconds)
SELECT /*+ parallel (calendar, 4) */ * FROM calendarWHERE day_nam...	6lab	29.07.22 13:07	SQL	9	0.244



```
EXPLAIN PLAN FOR
SELECT /*+ parallel (calendar, 4) */ * FROM calendar
WHERE day_name IN ('Понедельник','Среда', 'Пятница')
ORDER BY time_id DESC;
```

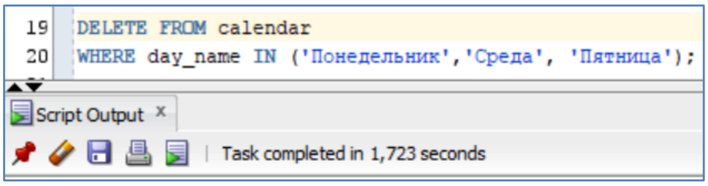
All Rows Fetched: 23 in 0,047 seconds

```
PLAN_TABLE_OUTPUT
|   2 |   PX SEND QC (ORDER)      | :TQ10001 |   428K|   47M|      |   602   (1)| 00:00:01 |  Q1,01 | P->S | QC (ORDER) |
|   3 |    SORT ORDER BY          |          |   428K|   47M|   65M|   602   (1)| 00:00:01 |  Q1,01 | PCWP |            |
|   4 |     PX RECEIVE            |          |   428K|   47M|      |   599   (1)| 00:00:01 |  Q1,01 | PCWP |            |
|   5 |      PX SEND RANGE        | :TQ10000 |   428K|   47M|      |   599   (1)| 00:00:01 |  Q1,00 | P->P | RANGE      |
|   6 |       PX BLOCK ITERATOR   |          |   428K|   47M|      |   599   (1)| 00:00:01 |  Q1,00 | PCWC |            |
|*  7 |        TABLE ACCESS FULL| CALENDAR |   428K|   47M|      |   599   (1)| 00:00:01 |  Q1,00 | PCWP |            |
-----------------------------------------------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------

   7 - filter("DAY_NAME"='Понедельник' OR "DAY_NAME"='Пятница' OR "DAY_NAME"='Среда')

Note
-----
   - Degree of Parallelism is 4 because of table property
```

As we can see from the results, the query execution time with parallel is less by 0.016 seconds, which almost does not matter. If we look at the execution plans for both queries, we can see that they are identical, that is, when fetching large amounts of data, Oracle itself chose a parallel without being explicitly indicated using hints.

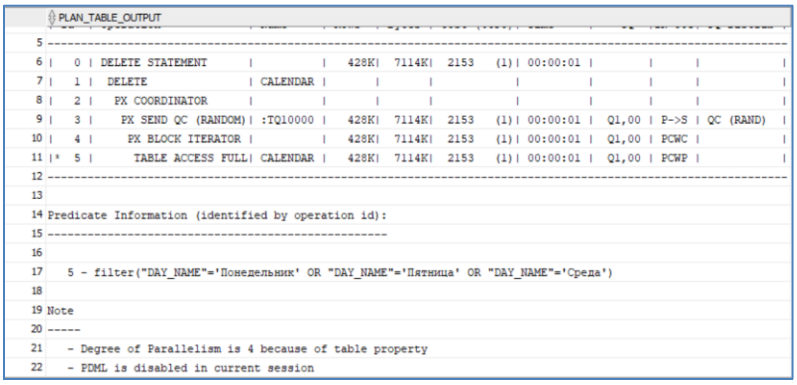
Task 2 - CREATE Example of Parallel DML

1) Without parallel DML



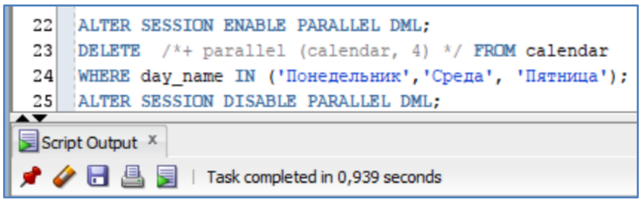
```sql
DELETE FROM calendar
WHERE day_name IN ('Понедельник','Среда', 'Пятница');
```

```
Task completed in 1,723 seconds
```



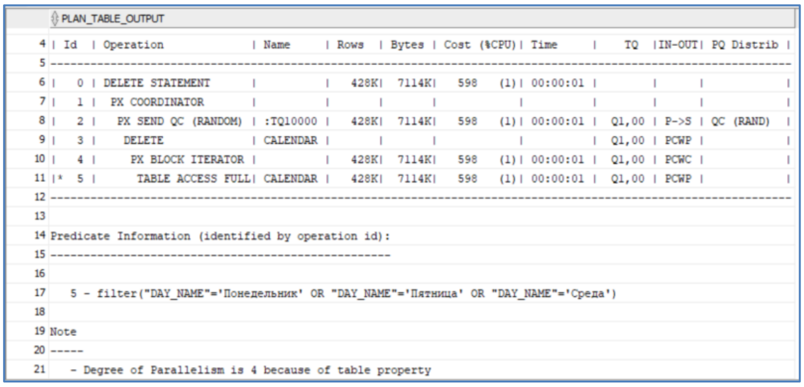
```
-------------------------------------------------------------------------------------------------------------------
|   0 | DELETE STATEMENT        |          |   428K|  7114K|  2153   (1)| 00:00:01 |        |      |            |
|   1 |  DELETE                 | CALENDAR |       |       |            |          |        |      |            |
|   2 |   PX COORDINATOR        |          |       |       |            |          |        |      |            |
|   3 |    PX SEND QC (RANDOM)| :TQ10000 |   428K|  7114K|  2153   (1)| 00:00:01 |  Q1,00 | P->S | QC (RAND)  |
|   4 |     PX BLOCK ITERATOR |          |   428K|  7114K|  2153   (1)| 00:00:01 |  Q1,00 | PCWC |            |
|*  5 |      TABLE ACCESS FULL| CALENDAR |   428K|  7114K|  2153   (1)| 00:00:01 |  Q1,00 | PCWP |            |
-------------------------------------------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------

   5 - filter("DAY_NAME"='Понедельник' OR "DAY_NAME"='Пятница' OR "DAY_NAME"='Среда')

Note
-----
   - Degree of Parallelism is 4 because of table property
   - PDML is disabled in current session
```

2) With Parallel DML



```sql
ALTER SESSION ENABLE PARALLEL DML;
DELETE  /*+ parallel (calendar, 4) */ FROM calendar
WHERE day_name IN ('Понедельник','Среда', 'Пятница');
ALTER SESSION DISABLE PARALLEL DML;
```

```
Task completed in 0,939 seconds
```

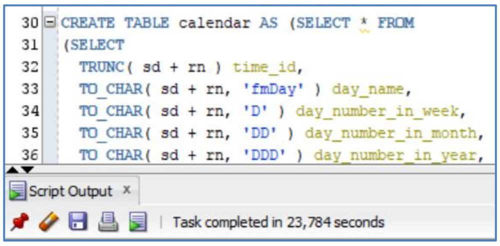


```
| Id  | Operation              | Name     | Rows  | Bytes | Cost (%CPU)| Time     |    TQ  |IN-OUT| PQ Distrib |
-------------------------------------------------------------------------------------------------------------------
|   0 | DELETE STATEMENT       |          |   428K|  7114K|   598   (1)| 00:00:01 |        |      |            |
|   1 |  PX COORDINATOR        |          |       |       |            |          |        |      |            |
|   2 |   PX SEND QC (RANDOM)  | :TQ10000 |   428K|  7114K|   598   (1)| 00:00:01 |  Q1,00 | P->S | QC (RAND)  |
|   3 |    DELETE              | CALENDAR |       |       |            |          |  Q1,00 | PCWP |            |
|   4 |     PX BLOCK ITERATOR |          |   428K|  7114K|   598   (1)| 00:00:01 |  Q1,00 | PCWC |            |
|*  5 |      TABLE ACCESS FULL| CALENDAR |   428K|  7114K|   598   (1)| 00:00:01 |  Q1,00 | PCWP |            |
-------------------------------------------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------

   5 - filter("DAY_NAME"='Понедельник' OR "DAY_NAME"='Пятница' OR "DAY_NAME"='Среда')

Note
-----
   - Degree of Parallelism is 4 because of table property
```

As we can see, the execution time of a query using parallel is less than half that of a query without parallel. Query plans differ only in the presence of PDML in the current session.

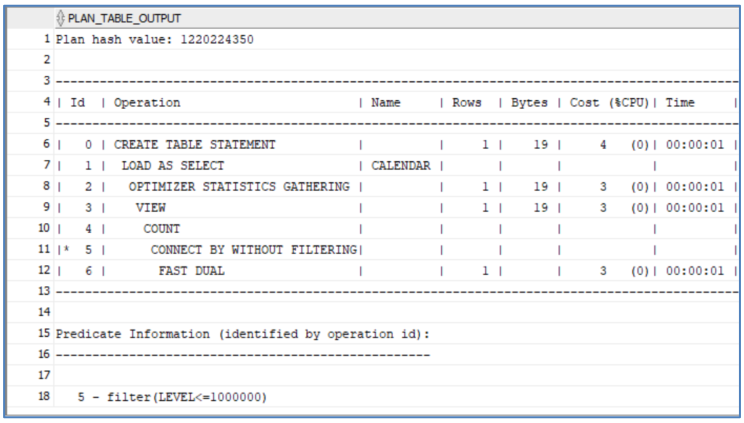
Task 3 - CREATE Example of Parallel DDL

1) Without parallel DDL



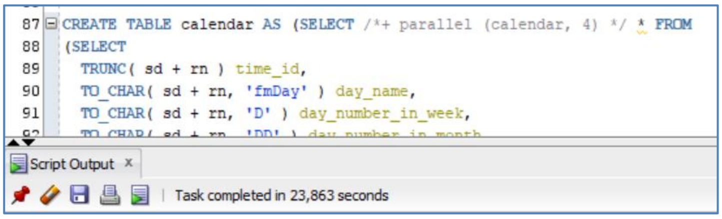
```
CREATE TABLE calendar AS (SELECT * FROM
(SELECT
  TRUNC( sd + rn ) time_id,
  TO_CHAR( sd + rn, 'fmDay' ) day_name,
  TO_CHAR( sd + rn, 'D' ) day_number_in_week,
  TO_CHAR( sd + rn, 'DD' ) day_number_in_month,
  TO_CHAR( sd + rn, 'DDD' ) day_number_in_year,
```

Task completed in 23,784 seconds



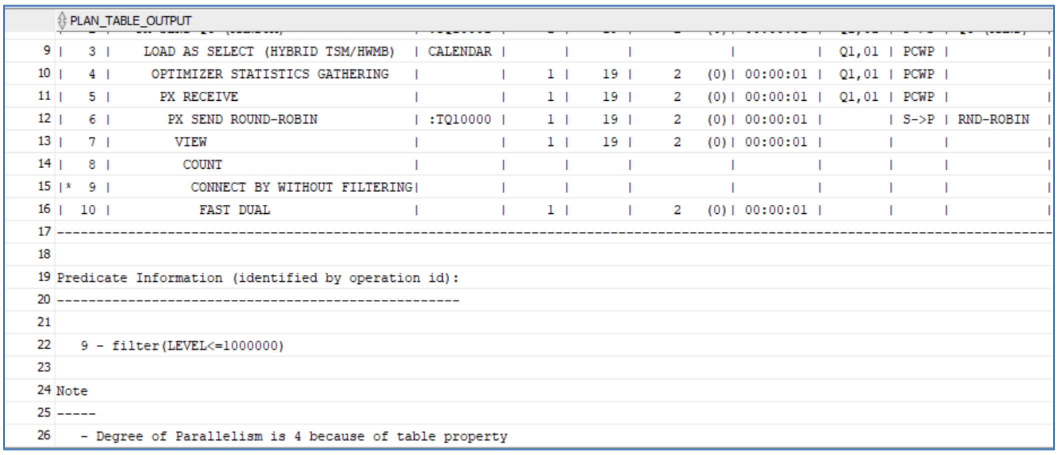
```
Plan hash value: 1220224350

-------------------------------------------------------------------------------------------
| Id  | Operation                      | Name     | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------------
|   0 | CREATE TABLE STATEMENT         |          |     1 |    19 |     4   (0)| 00:00:01 |
|   1 |  LOAD AS SELECT                | CALENDAR |       |       |            |          |
|   2 |   OPTIMIZER STATISTICS GATHERING |        |     1 |    19 |     3   (0)| 00:00:01 |
|   3 |    VIEW                        |          |     1 |    19 |     3   (0)| 00:00:01 |
|   4 |     COUNT                      |          |       |       |            |          |
|*  5 |      CONNECT BY WITHOUT FILTERING|          |       |       |            |          |
|   6 |       FAST DUAL                |          |     1 |       |     3   (0)| 00:00:01 |
-------------------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------

   5 - filter(LEVEL<=1000000)
```

2) With parallel DDL (hint)



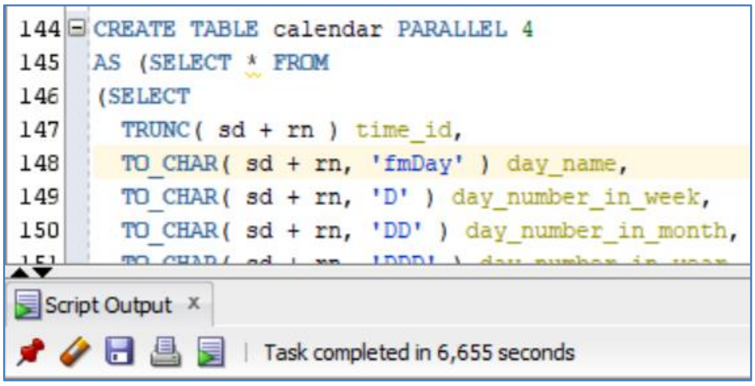
```
CREATE TABLE calendar AS (SELECT /*+ parallel (calendar, 4) */ * FROM
(SELECT
  TRUNC( sd + rn ) time_id,
  TO_CHAR( sd + rn, 'fmDay' ) day_name,
  TO_CHAR( sd + rn, 'D' ) day_number_in_week,
  TO_CHAR( sd + rn, 'DD' ) day_number_in_month
```

Task completed in 23,863 seconds



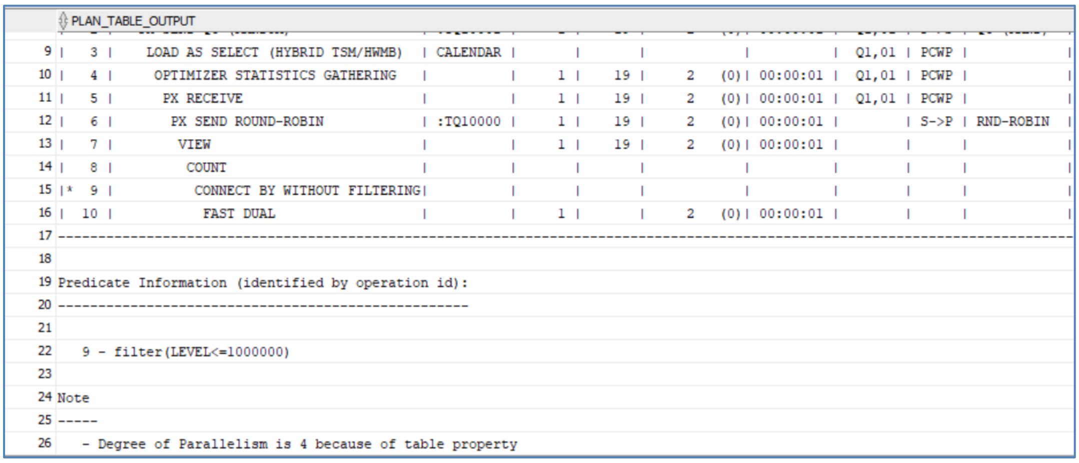
```
|   3 |      LOAD AS SELECT (HYBRID TSM/HWMB) | CALENDAR |       |       |            |          |  Q1,01 | PCWP |            |
|   4 |       OPTIMIZER STATISTICS GATHERING  |          |     1 |    19 |     2   (0)| 00:00:01 |  Q1,01 | PCWP |            |
|   5 |        PX RECEIVE                     |          |     1 |    19 |     2   (0)| 00:00:01 |  Q1,01 | PCWP |            |
|   6 |         PX SEND ROUND-ROBIN           | :TQ10000 |     1 |    19 |     2   (0)| 00:00:01 |        | S->P | RND-ROBIN  |
|   7 |          VIEW                         |          |     1 |    19 |     2   (0)| 00:00:01 |        |      |            |
|   8 |           COUNT                       |          |       |       |            |          |        |      |            |
|*  9 |            CONNECT BY WITHOUT FILTERING|         |       |       |            |          |        |      |            |
|  10 |             FAST DUAL                 |          |     1 |       |     2   (0)| 00:00:01 |        |      |            |
-----------------------------------------------------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------

   9 - filter(LEVEL<=1000000)

Note
-----
   - Degree of Parallelism is 4 because of table property
```

3) With parallel DDL (PARALLEL command)



```
CREATE TABLE calendar PARALLEL 4
AS (SELECT * FROM
 (SELECT
   TRUNC( sd + rn ) time_id,
   TO_CHAR( sd + rn, 'fmDay' ) day_name,
   TO_CHAR( sd + rn, 'D' ) day_number_in_week,
   TO_CHAR( sd + rn, 'DD' ) day_number_in_month,
   TO_CHAR( sd + rn, 'DDD' ) day_number_in_year
```

Task completed in 6,655 seconds



```
|   3 |    LOAD AS SELECT (HYBRID TSM/HWMB)  | CALENDAR |      |       |            |          |  Q1,01 | PCWP |            |
|   4 |     OPTIMIZER STATISTICS GATHERING   |          |    1 |    19 |     2   (0)| 00:00:01 |  Q1,01 | PCWP |            |
|   5 |      PX RECEIVE                      |          |    1 |    19 |     2   (0)| 00:00:01 |  Q1,01 | PCWP |            |
|   6 |       PX SEND ROUND-ROBIN            | :TQ10000 |    1 |    19 |     2   (0)| 00:00:01 |        | S->P | RND-ROBIN  |
|   7 |        VIEW                          |          |    1 |    19 |     2   (0)| 00:00:01 |        |      |            |
|   8 |         COUNT                        |          |      |       |            |          |        |      |            |
|*  9 |          CONNECT BY WITHOUT FILTERING|          |      |       |            |          |        |      |            |
|  10 |           FAST DUAL                  |          |    1 |       |     2   (0)| 00:00:01 |        |      |            |
-----------------------------------------------------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------

   9 - filter(LEVEL<=1000000)

Note
-----
   - Degree of Parallelism is 4 because of table property
```

As you can see from the query results, the query that runs the fastest is using the PARALLEL command.

The small amount of data, the parallelization does not improve the results. To benefit from parallel query execution, you need to work with a large amount of information.

Task 4 - CREATE Strategy of Parallel execution

The use of parallelism can be a good example of increasing the speed and efficiency of a data warehouse.

Since I have tables in my data warehouse that need to be constantly updated or new data added, such as ORDER_FACT, CLIENT_DIMENSION, EMPLOYEE_DIMENSION, it makes more sense to use the DML parallel to run this process more efficiently at the DW, CL, and SA levels.