



## Example to Unveil the Power of Oracle Cost-Based SQL Optimizer

A user who has a SQL statement takes a long time to execute, actually the SQL is not very complicated, but it has a very complex execution plan.

Mimic SQL text :

```
SELECT TO_CHAR(SYSDATE, 'yyyy-mm-dd') AS STAT_DATE,
       SYSDATE AS STAT_TIME,
       X.TABLE_NAME,
       NVL(X.NUM_ROWS, 0) AS TABLE_ROWS,
       NVL(ROUND(X.NUM_ROWS * X.AVG_ROW_LEN / 1024, 2), 0) AS TABLE_SIZES,
       Y.CREATED AS CREATE_TIME,
       Z.COMMENTS AS TABLE_COMMENT,
       H.COL_CNT
FROM ALL_TABLES X,
     ALL_OBJECTS Y,
     ALL_TAB_COMMENTS Z,
     (SELECT H.TABLE_NAME, COUNT(1) AS COL_CNT
      FROM ALL_TAB_COLS H
      WHERE H.OWNER = 'TOSSKA'
      GROUP BY H.TABLE_NAME) H
WHERE X.TABLE_NAME = Y.OBJECT_NAME
     AND X.TABLE_NAME(+) = Z.TABLE_NAME
     AND X.TABLE_NAME = H.TABLE_NAME
     AND Y.OBJECT_TYPE IN ('TABLE PARTITION', 'TABLE')
     AND X.OWNER = 'TOSSKA'
     AND Y.OWNER = 'TOSSKA'
     AND Z.OWNER = 'TOSSKA'
ORDER BY X.TABLE_NAME
```



## Execution Plan :

The following partial plan steps list about 10% of a total of 214 steps of execution plan.

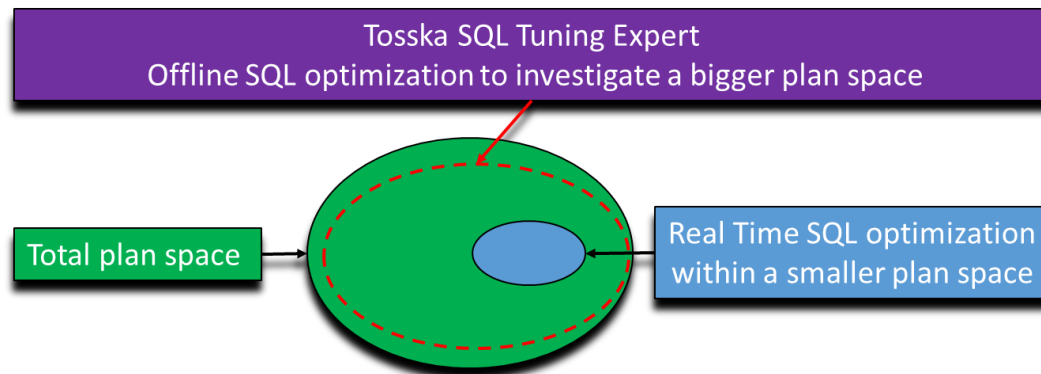


User input the SQL into Toska SQL Tuning Expert for Oracle, and press Tune to start exploring if there are only potential better query plans from Oracle. User goes back to his daily work and let the computer do the rest of tuning job on his behalf. Finally, there are 5 better alternatives of SQL which are found within an hour.



## Product Highlight

Toska SQL Tuning Expert equipped with an AI engine which can try most effective combinations of Hints injection to the problematic SQL. The engine can deeply explore any hidden good execution plans that Oracle CBO cannot discover at the stage of online SQL optimization, but those good plans actually exist in the plans space for the given SQL statement.



Toska SQL Tuning Expert for Oracle 2.0.3

Open Session Tuning-16 Tuning-17 \* New Session

Tune Explain Plan Explore Alternatives Test Run Compare SQL Find SQL Abort Save Clear

SQL Editor Current Schema TOSSKA

```
SELECT /*+ FULL(@SEL$134 0) */ to_char(SYSDATE, 'yyyy-mm-dd') AS stat_date,
SYSDATE AS stat_time,
x.table_name,
nvl(x.num_rows,0) AS table_rows
nvl(round(x.num_rows * x.avg_row_len / 1024,2),0) AS table_size
```

SQL Alternative	Cost	Row Count	Status	Elapsed Time	Response Time	Physical Reads	Logical Reads
Original		1,330	1750 ✓	00:14:07.93	00:14:07.90	34,821	65,118
SQL 130		1,178	1750 ✓	00:01:07.83	00:01:07.82	1,518	11,118
SQL 135		1,178	1750 ✓	00:00:06.18	00:00:06.17	1,517	40,118
SQL 127		3,252	1750 ✓	00:00:05.18	00:00:05.17	455	68,118
SQL 129		3,974	1750 ✓	00:00:05.02	00:00:05.01	103	62,118
SQL 45		1,329	1750 ✓	00:00:04.42	00:00:04.41	345	60,118
SQL 141		1,333	Test run was...	>00:00:05.42			

## Benchmark Result:

Original SQL takes 14 minutes and 7 seconds.

The best alternative SQL with hints injected is SQL 45 and it takes 4 seconds only.

SQL45 is running more than 99 times faster than the Original SQL



### **Observation of the query plans generated by Oracle CBO**

Tosska SQL Tuning Expert is a Hints-Injection-Based SQL tuning tool without the need of rewriting user's SQL text. So, various hints injected to a SQL statement and the corresponding query plans generated by Oracle are all potential query plans that Oracle SQL optimizer can provide for the SQL statement.

Let's review those generated query plans and why Oracle cannot find the best query plan at the beginning.

#### **First observation:**

Original SQL's cost is 1330, but the cost of SQL 130, SQL 135 and SQL 45 are all lower than Original SQL's cost, why can't Oracle pick up these lower cost plans?

#### **Reason:**

Oracle cannot explore all potential query plans that it can generate within a short time, otherwise it will take even longer time to optimize a given SQL that might not be fully compensated by an unforeseeable better query plan.

#### **Second observation:**

SQL 127 and SQL 129 have 3 times higher cost than Original SQL, but the speed is much faster than Original SQL, it means the cost estimation of these two SQL are exceptionally wrong.

#### **Reason:**

It is the limitation of cost estimation algorithm used in database SQL optimizer. Theoretically, there is no 100% accurate SQL cost estimation algorithm in the market that can handle various environments, and the problem is especially true for complex SQL statements like the above SQL statement.

#### **Conclusion:**

Oracle has the most sophisticated SQL optimizer in the market. There are a lot of better query plans that Oracle are potential can run faster for your SQL statements. So, whenever you are thinking to upgrade your hardware or cloud service spending, you should first explore the potential power that Oracle SQL optimizer can provide for your SQL statements, and what you need is only a right tool that can unveil the potential power of your Oracle SQL optimizer.

<https://tosska.com/tosska-sql-tuning-expert-tse-oracle/>

<https://tosska.com/tosska-sql-tuning-expert-pro-tse-pro-for-oracle/>