# Lab4. Report made by Litvin Saveli. DataMola Student

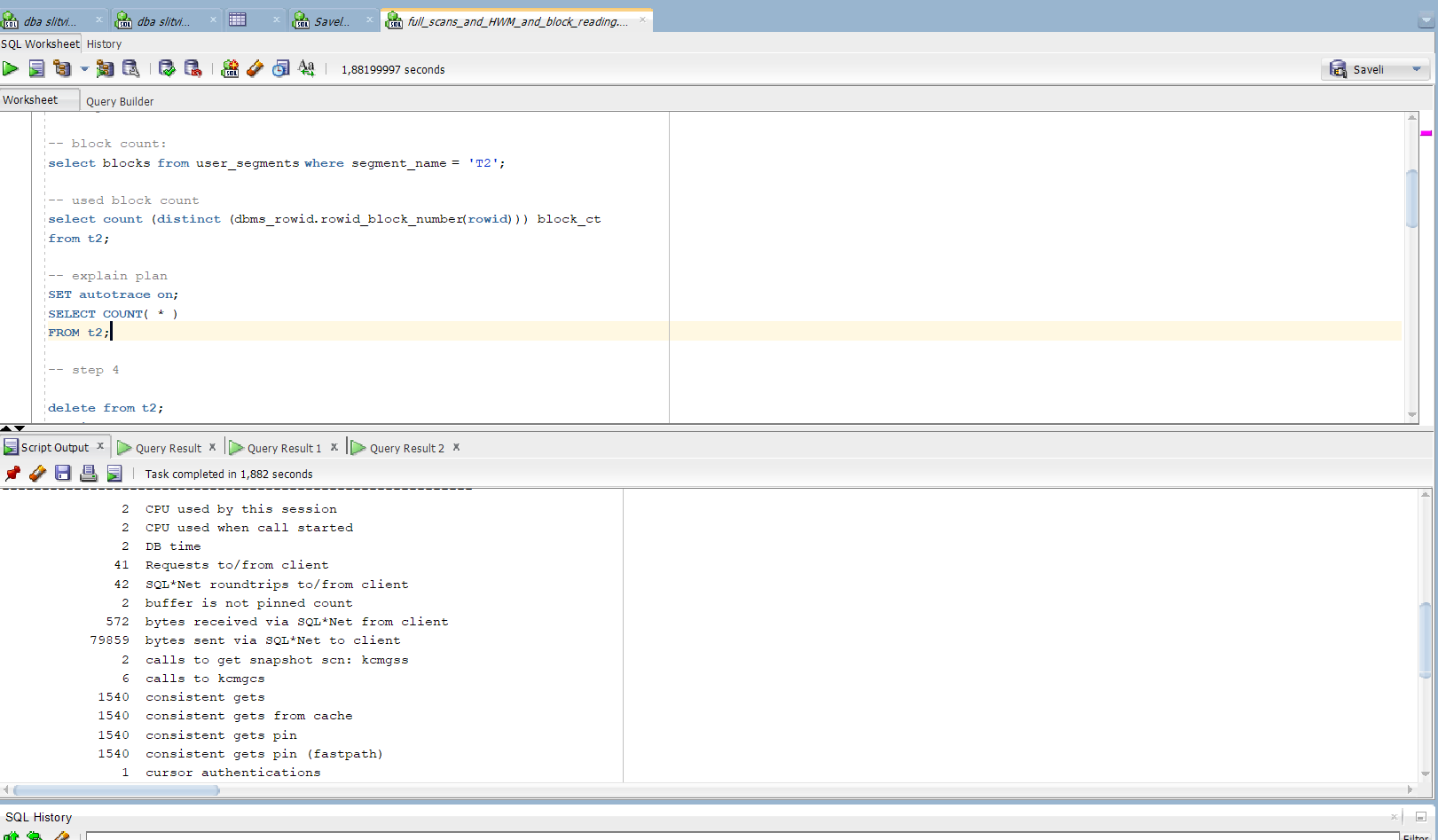
# 1. Table access full scan

## 1.1. Task 1: Full Scans and the High-water Mark and Block reading

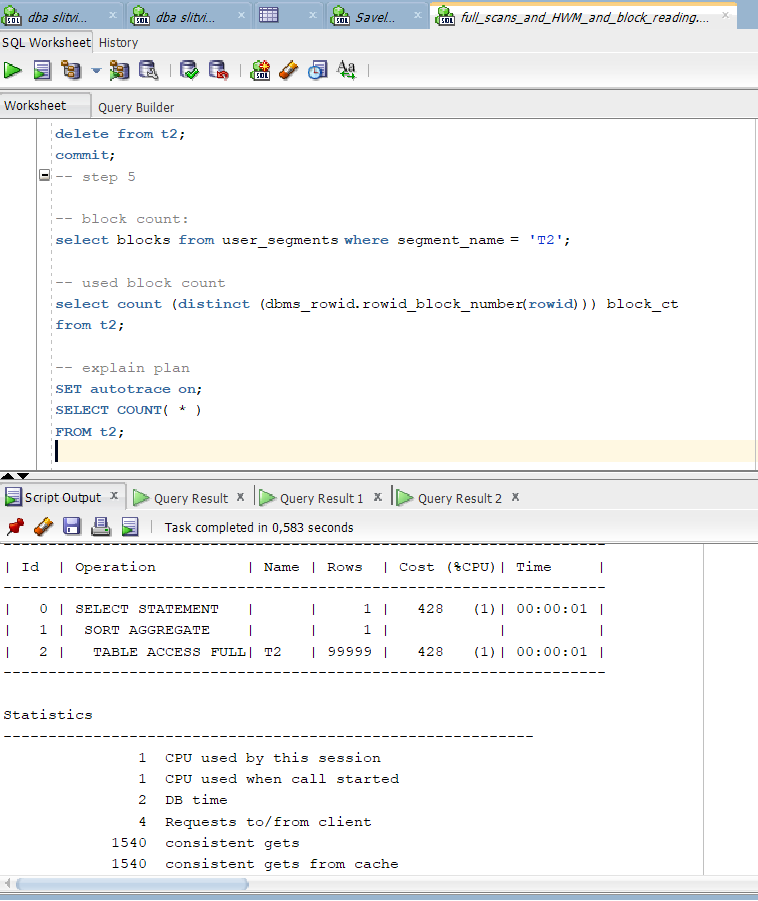
**Task Results:**

Expected:

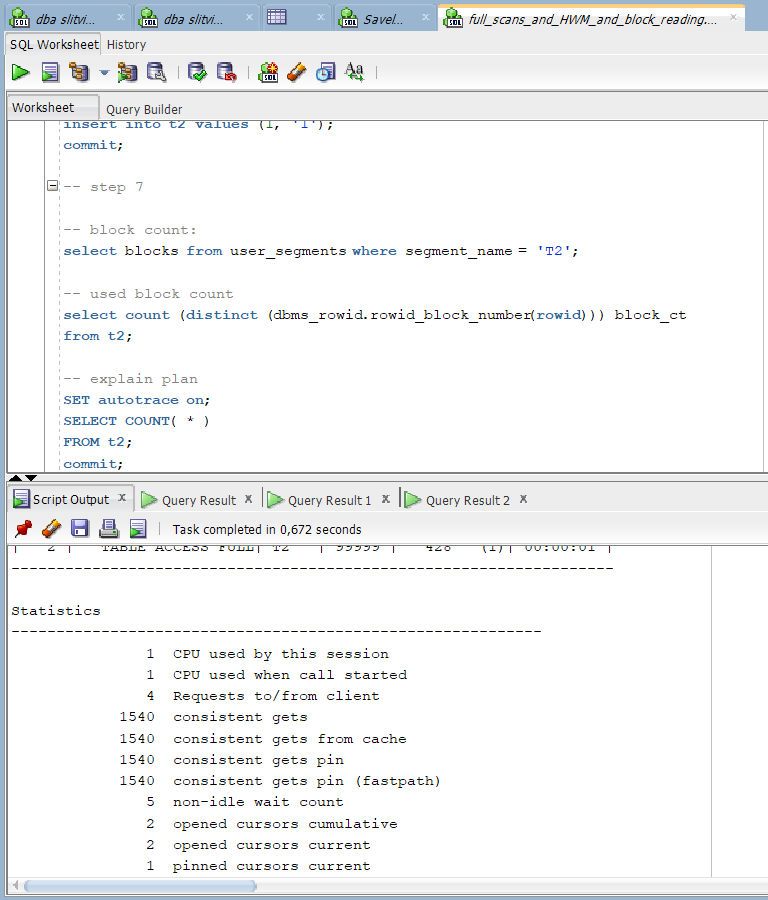
Summary table with all result and text description of analyses this results.



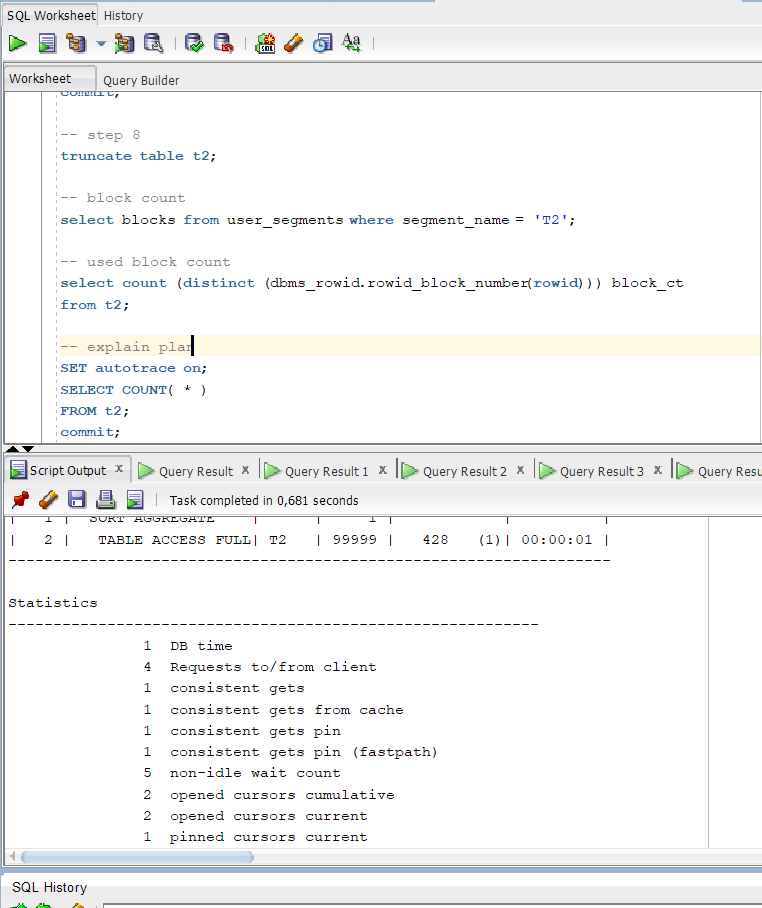
Step 1- 3



Steps 4-5



Steps 6-7



Step 8

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Count of Blocks | Count of Used Blocks | Count of Rows | Consistent gets | Description |
| 1 | 1664 | 1536 | 99999 | 1540 | The table was filled into 1536 blocks and summarized block count is 1664. Allocated consistent to get table is 1540. Max\_row\_size = 99999 |
| 2 | 1664 | 0 | 0 | 1540 | With using delete statement we can still restore our data, ‘delete’ operand saves allocated space and the object’s statistic. |
| 3 | 1664 | 1 | 1 | 1540 | Consistent gets is not change, cause allocated space were not change, even with inserting an row into our table. |
| 4 | 8 | 0 | 0 | 1 | Number of counts was decreased from 1664 to 8 (Minimal block counts). By the settings each block counts scales at 8. Truncating is deleting rows without opportunity to restore. The ‘Truncate’ deletes not only rows, but also statistic allocated space and high water mark tips. |

# 2. Index Scan types

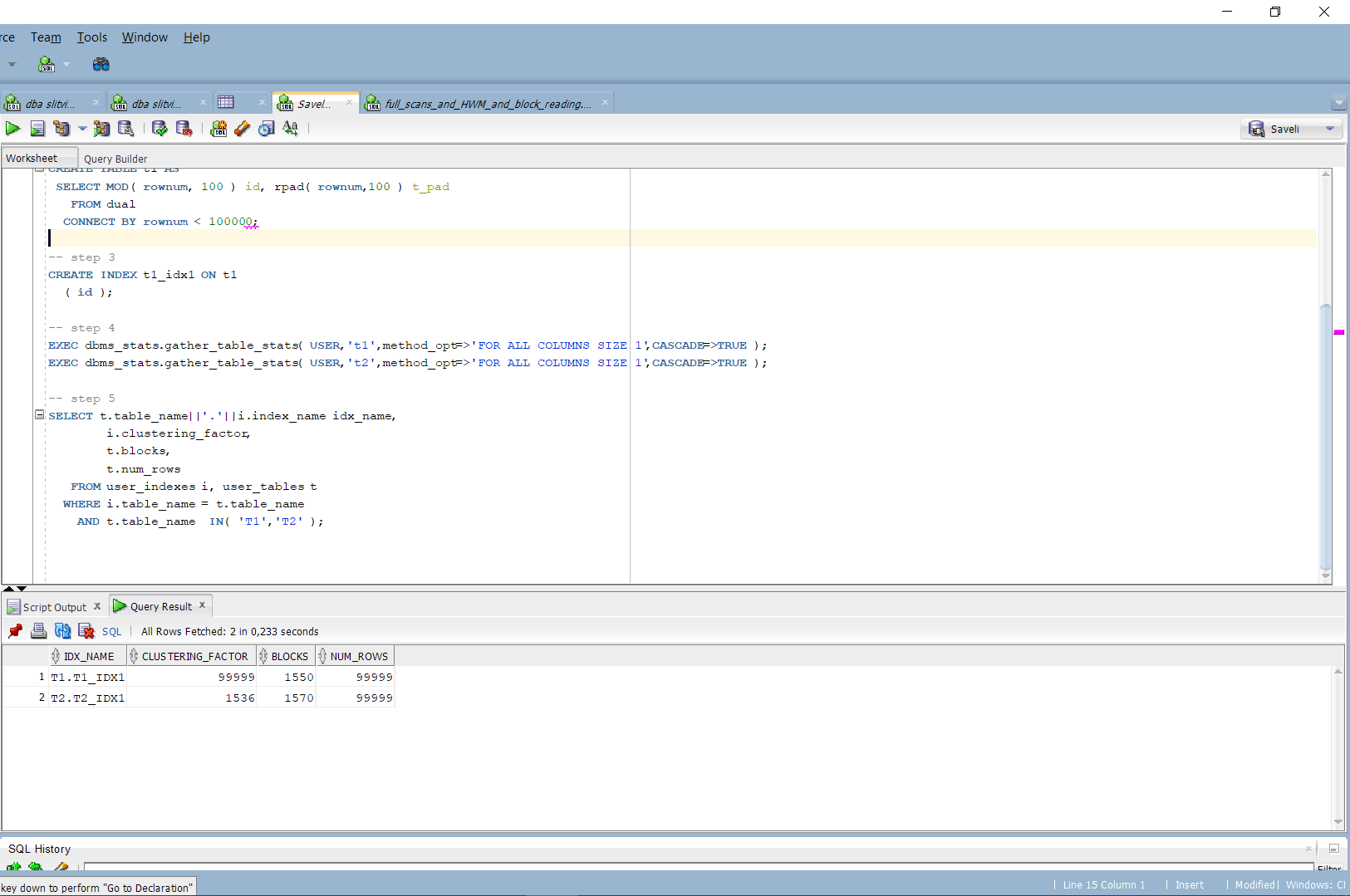
## 2.1. Task 2: Index Clustering factor parameter

**Task Results:**

Expected:

* Screenshot of the step 5;
* Description of the parameter clustering factor;
* Explanation: why for indexes *t1\_idx1* and *t2\_idx1* we have different values ;

Which Index has best selective performance in execution Select clause filtered by IN ( , list of values, );



Step 5

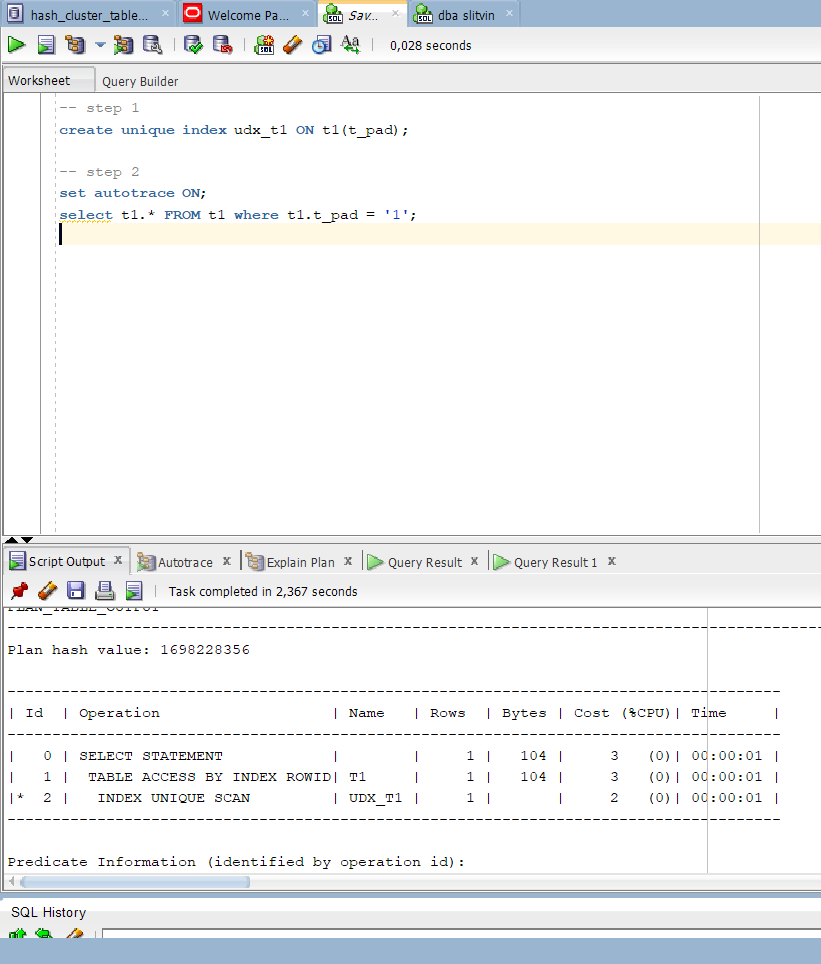
Clustering factor records the number of blocks that will be read when scanning the index. The biggest advantage is that correctly setuped clustering could increase speed while searching and decrease costs. The clustering of data within the table can be used to improve the performance of statements that perform range scan–type operations. By determining how the column is being used in the statements, indexing these column(s) may provide a great benefit.

The clustering index is computed by the following statements:

* The index is scanned in order.
* The block portion of the ROWID pointed at by the current indexed valued is compared to the previous indexed value (comparing adjacent rows in the index).
* If the ROWIDs point to different TABLE blocks, the clustering factor is incremented (this is done for the entire index).

We’ve receive different values for indexes, cause by creating tables we’ve used different operands ‘MOD’ in t1, instead of ‘TRUNC in t2.

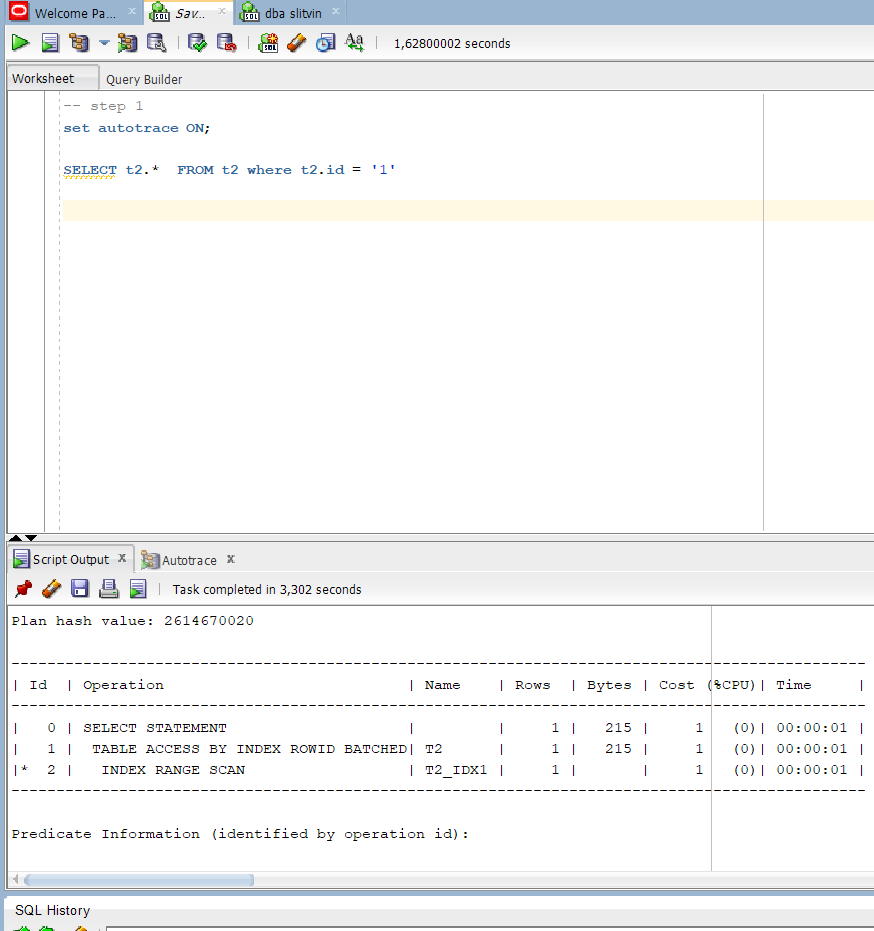
## 2.2. Task 3: Index Unique Scan



Step 2

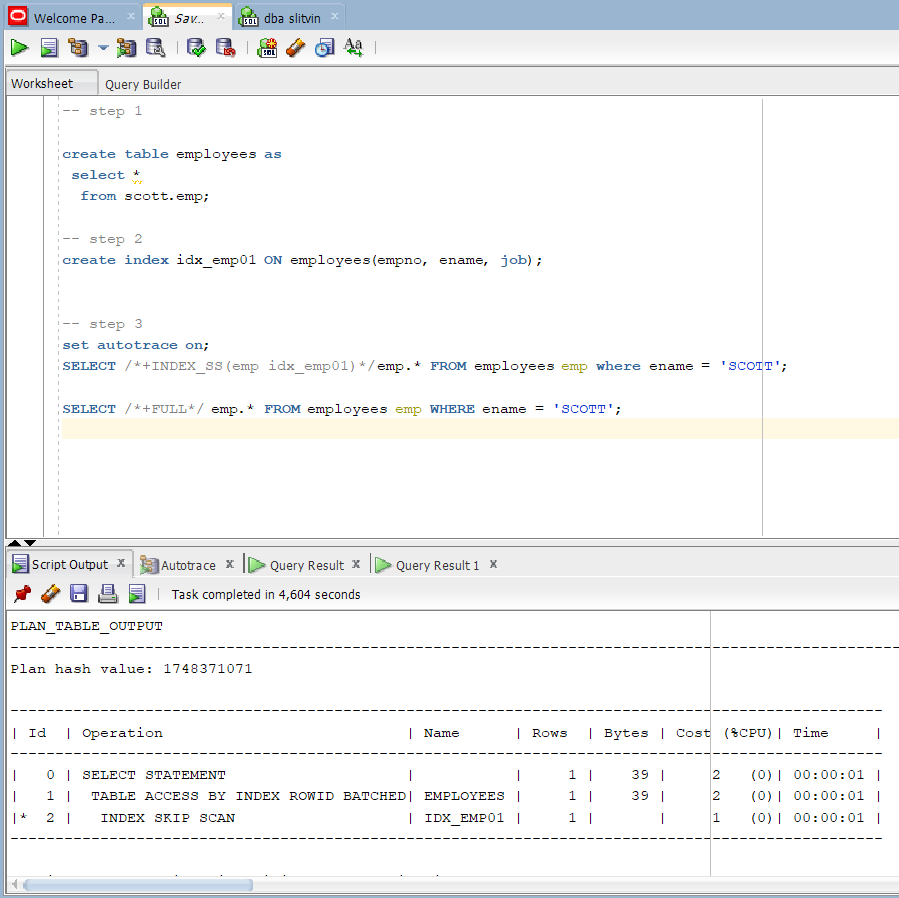
Oracle reads the index nodes by accessing the table and the returns the ROWID for the corresponding index row.

## 2.3. Task 4: Index Range Scan



The Oracle reads adjacent index entries in the table and then returns ROWID values corresponding to this index to retrieve tows.

## 2.4. Task 5: Index Skip Scan



Oracle simply scans the index sub-trees for each of the possible values of the leading column.

Oracle index full scan only reads all of the leaf blocks of an index, but only enough of the branch blocks to find the first leaf block.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Count of Blocks | Count of Used Blocks | Count of Rows | Consistent gets | Description |
| 1 | 1664 | 1516 | 99999 | 1520 | Table t1 |
| 2 | 8 | 0 | 0 | 1 | Table t2 |
| 3 | 8 | 1 | 12 | 2 | Table employees |