|  |
| --- |
| MTN.\*NIX.07 Labs - Access and Join Methods Part 2 |

# Auto Trace & Explain Plan

## Task 1: Auto Trace configuration training

Below all possible variants of SQL plus utilities autotrace:

set autotrace off

set autotrace on

set autotrace traceonly

set autotrace on explain

set autotrace on statistics

set autotrace on explain statistics

set autotrace traceonly explain

set autotrace traceonly statistics

set autotrace traceonly explain statistics

set autotrace off explain

set autotrace off statistics

set autotrace off explain statistics

**NOTE:** If you received next error: Check PLUSTRACE role is enabled. Please make next steps:

1. Run next script connected as sysdba:

# @ $ORACLE\_HOME/sqlplus/admin/plustrce.sql;

1. Grant role PLUSTRACE to $UserName$

# grant plustrace to $UserName$;

**Task Results:**

Expected:

Summary table with all result and text description of analyses this results. SELECT empno from employees;

|  |  |  |  |
| --- | --- | --- | --- |
| № | Auto Trace Configuration Options | Expected Results | Description |
|  | set autotrace on | Result Set  Execution Plan,  Statistics |  |
|  | set autotrace traceonly | Execution Plan  Statistics |  |
|  | set autotrace off | Result Set |  |
|  | set autotrace on explain | Result Set  Execution Plan |  |
|  | set autotrace on statistics | Result Set  Statistics |  |
|  | set autotrace on explain statistics | Result Set  Execution Plan  Statistics |  |
|  | set autotrace traceonly explain | Execution Plan |  |
|  | set autotrace traceonly statistics | Statistics |  |
|  | set autotrace traceonly explain statistics | Execution Plan  Statistics |  |
|  | set autotrace off explain | Result Set |  |
|  | set autotrace off statistics | Result Set |  |
|  | set autotrace off explain statistics | Result Set |  |

# Join Methods

**The Main Task** is to create SQL and prepare execution plan of statements with join methods on Task Topics (Task 2 - 9)

**Task Results:**

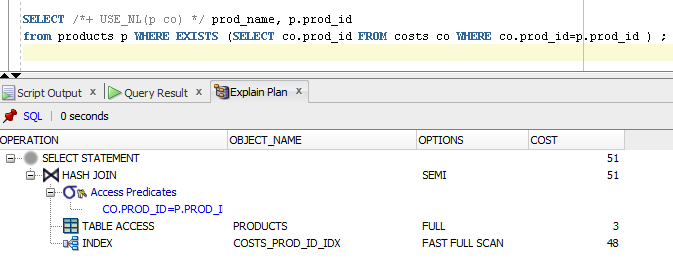
There are several tasks below with the same main expected result points:

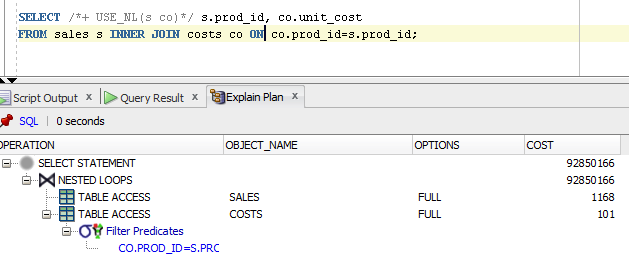
* Create SQL using next tables: scott.emp, scott.dept
* Create additional needed Tables and Indexes
* Prepare screenshots of execution plan

In queries were used tables from scheme SH and created Indexes.

|  |  |  |
| --- | --- | --- |
| № | TABLE NAME/INDEX NAME | STATISTICS |
| 1 | Products |  |
| 2 | Product\_subcategories  Result of a query –  CREATE TABLE prod\_sub AS  SELECT rownum r\_id, ps prod  FROM(  SELECT distinct prod\_subcategory as ps  FROM products); |  |
| 3 | Prod\_sub\_idx  (non\_unique index on sh.products (prod\_subcategory) ) |  |
| 4 | Costs |  |
| 5 | Sales |  |
| 6 | Sales\_prod\_id\_idx  (non\_unique index on sh.sales (prod\_id) ) |  |
| 7 | Costs\_prod\_id\_idx  (non\_unique index on sh.costs(prod\_id) ) |  |

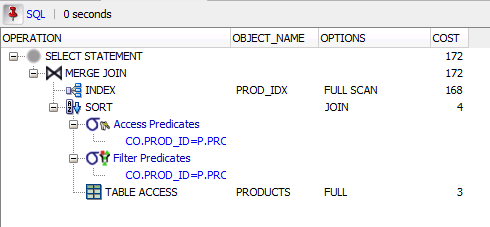
## Task 2: Nested Loops Joins





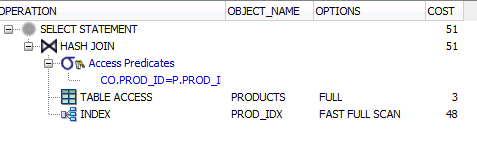
**Note:**  If you would like change in execution plan the type of join method use oracle performance hints. (USE\_NL)

## Task 3: Sort-Merge Joins



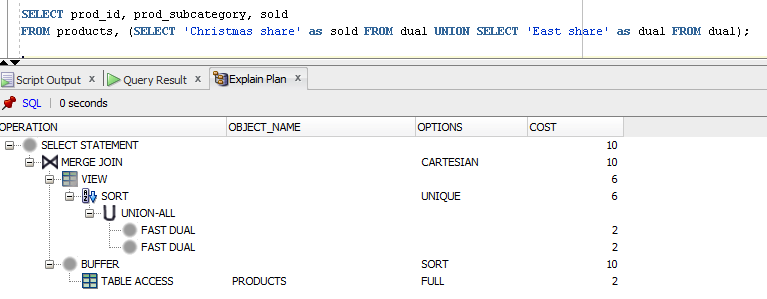
**Note:**  If you would like change in execution plan the type of join method use oracle performance hints. (USE\_MERGE)

## Task 4: Hash Joins

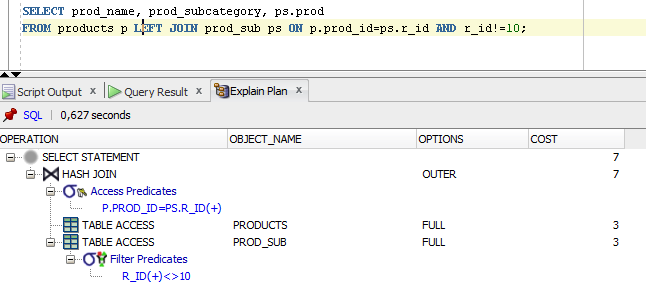


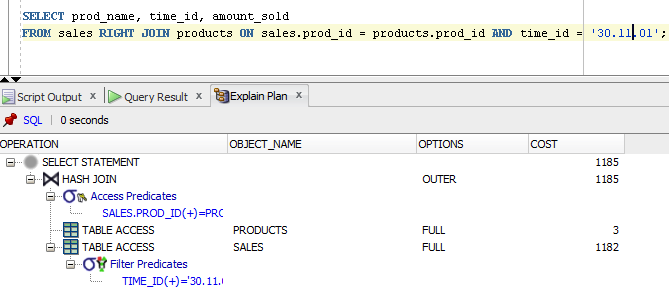
**Note:**  If you would like change in execution plan the type of join method use oracle performance hints. (USE\_HASH)

## Task 5: Cartesian Joins

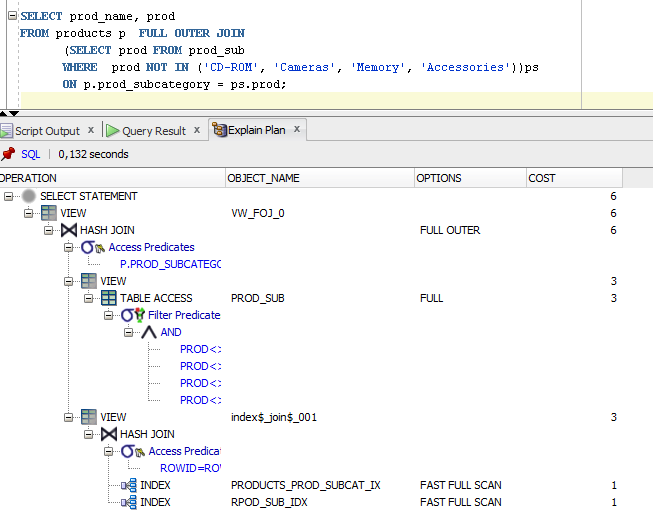


## Task 6: Left/Right Outer Joins

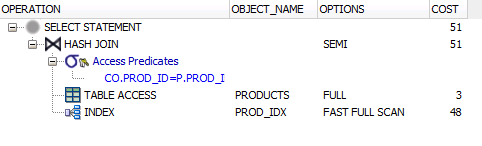




## Task 7: Full Outer Join



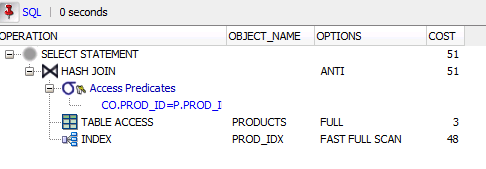
## Task 8: Semi Joins



**Note:**  If you would like change in execution plan the type of join method use oracle performance hints.

1. SEMIJOIN – perform a semi-join (the optimizer gets to pick which kind)
2. NO\_SEMIJOIN – obviously means don’t perform a semi-join
3. NL\_SJ – perform a nested loops semi-join (deprecated as of 10g)
4. HASH\_SJ – perform a hash semi-join (deprecated as of 10g)
5. MERGE\_SJ – perform a merge semi-join (deprecated as of 10g)

## Task 9: Anti Joins



**Note:**  If you would like change in execution plan the type of join method use oracle performance hints.

1. ANTIJOIN – perform an anti-join (the optimizer gets to pick which kind)
2. USE\_ANTI – older version of ANTIJOIN hint
3. NL\_AJ – perform a NESTED LOOPS anti-join (deprecated as of 10g)
4. HASH\_AJ – perform a HASH anti-join (deprecated as of 10g)
5. MERGE\_AJ – perform a MERGE anti-join (deprecated as of 10g)

## Task 10: Prepare summary table

**Task:** Make comparison of all possible variant of join methods and join access methods and fill the table below:

To build queries were used tables from sheme SH.

|  |  |  |  |
| --- | --- | --- | --- |
| № | TABLE TYPE | TABLE NAME/INDEX NAME | STATISTICS |
| 1 | Small table | Products |  |
| 2 | Small table | Product\_subcategories  Result of a query –  CREATE TABLE prod\_sub AS  SELECT rownum r\_id, ps prod  FROM(  SELECT distinct prod\_subcategory as ps  FROM products); |  |
| 3 | Small Indexed Table | Products/Prod\_sub\_idx  (non\_unique index(prod\_subcategory) ) |  |
| 4 | Big Table | Costs |  |
| 5 | Big Table | Sales |  |
| 6 | Big Indexed Table | Sales/Sales\_prod\_id\_idx  (non\_unique index(prod\_id) ) |  |

Query cost used in this table like a scale.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Join Access “A” | Join Access “B” | Nested Loop | Hash Join | Sort-Merge Join | Anti-Join | Semi-Join |
| Small Table(1) | Small Table(2) | 24 | 7 | 8 | 7 | 7 |
| Small Table(2) | Indexed Small Table(3) | 24 | 7 | 8 | 7 | 7 |
| Small Table(1) | Big Table(5) | 7290 | 107 | 346 | 107 | 107 |
| Small Table(1) | Indexed Big Table(6) | 148 | 51 | 172 | 51 | 51 |
| Big Table(4) | Big Table(5) | 92981413 | 7358 | 9531 | 2171 | 1517 |
| Big Table(4) | Big Indexed Table(6) | 2130514 | 6651 | 7584 | 2116 | 1462 |