Query #1

```
SELECT *
FROM emp, dept
WHERE emp.deptno = dept.deptno AND emp.job = 'ENGINEER';
```

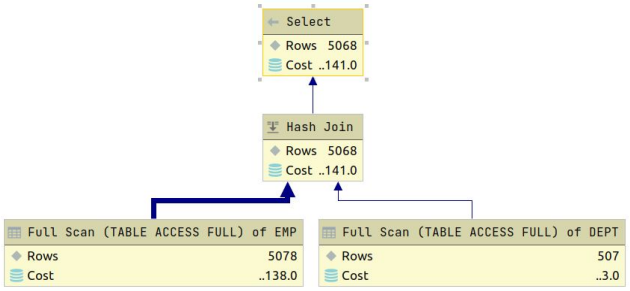
Cambiare l'obiettivo di ottimizzazione dalla modalità ALL ROWS (best throughput) alla modalità FIRST_ROWS (best response time) attraverso l'uso di hint (/*+ FIRST_ROWS(n) */). n è una variabile numerica intera che può assumere valori maggiori o uguali a 1. Assegnare diversi valori ad n e verificare come variano il piano d'esecuzione e i costi delle diverse operazioni.

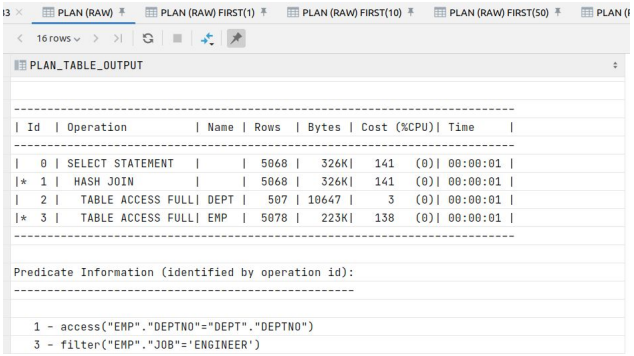
ALL ROWS: piano esecuzione ottimizzato in modo da minimizzare il tempo di esecuzione di tutta la query.

FIRST_ROWS(n): piano esecuzione ottimizzato in modo da minimizzare il tempo di esecuzione per i primi n record del risultato.

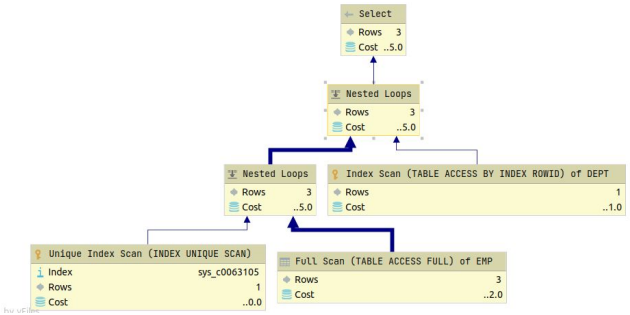
```
SELECT /*+ FIRST_ROWS(n) */ *
FROM emp, dept
WHERE emp.deptno = dept.deptno AND emp.job = 'ENGINEER';
```

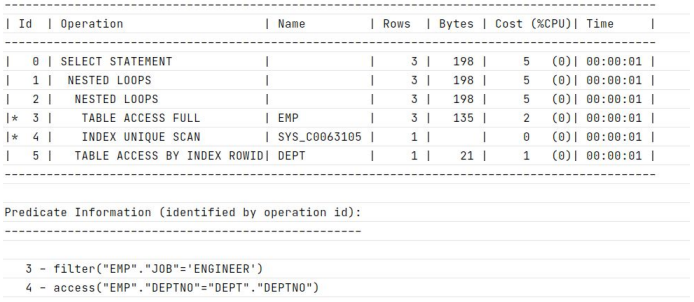
ALL ROWS





FIRST_ROWS(1)





FIRST_ROWS(10,50,150,2500,5000)

Piano uguale a all_rows

first_rows(10)->TAF-> 12

first_rows(50)->TAF-> 52

first_rows(150)->TAF-> 152

hash join->563

first_rows(2500)->TAF-> 2506

hash join->2511

first_rows(5000)->TAF-> 5011

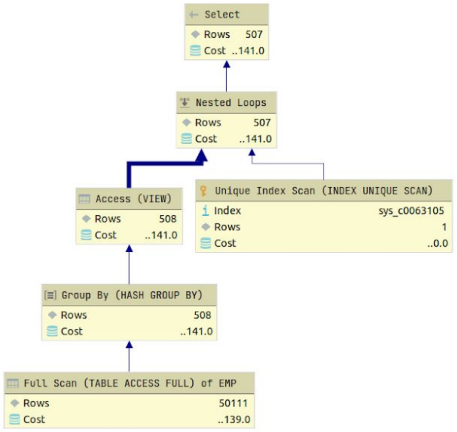
hash join->5001

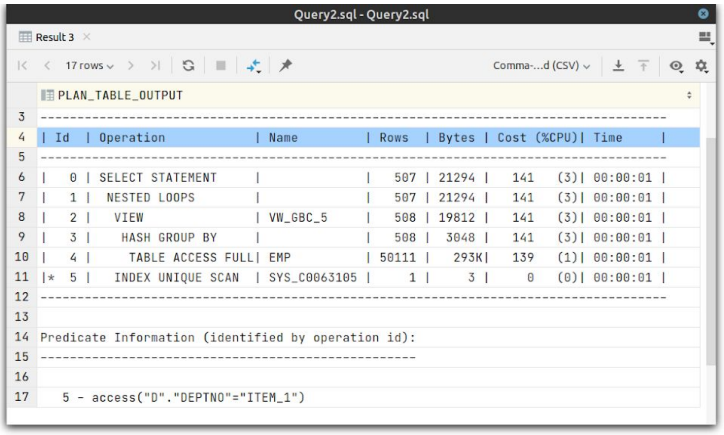
Query #2

Confrontare i costi di hash join e nested loop usando l'hint USE / NO_USE_HASH

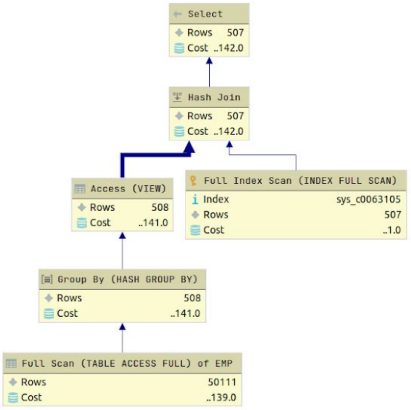
```
SELECT /*+ NO_USE_HASH(e d) */ d.deptno, AVG(e.sal)
FROM emp e, dept d
WHERE d.deptno = e.deptno
GROUP BY d.deptno;
```

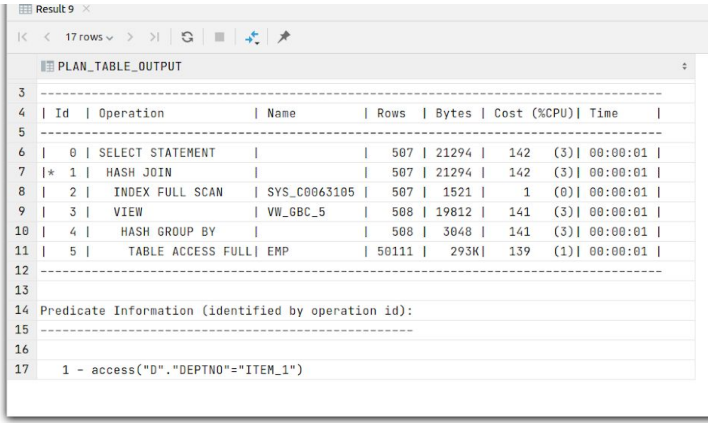
Default=No_Use_Hash(e d)





Use_Hash(e d)



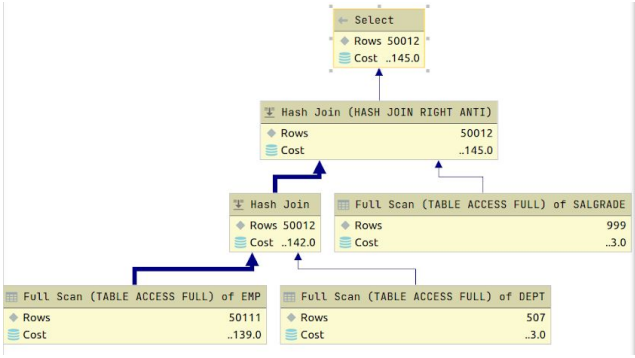


Query #3

Disabilitare il metodo hash join mediante l'uso di hint (/*+ NO_USE_HASH(e d) */)

```
SELECT /*+ NO_USE_HASH(e d) */ ename, job, sal, dname
FROM emp e, dept d
WHERE e.deptno = d.deptno
AND NOT EXISTS
(SELECT * FROM salgrade WHERE e.sal = hisal);
```

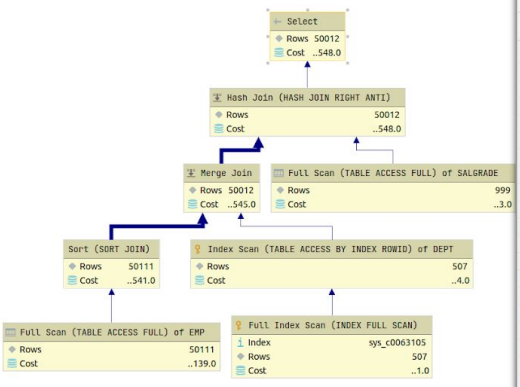
Default=Use_Hash(e d)



PLAN_TABLE_OUTPUT

```
Plan hash value: 2000531936

-------------------------------------------------------------------------------
| Id  | Operation             | Name     | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------
|   0 | SELECT STATEMENT      |          | 50012 |  2246K|   145   (1)| 00:00:01 |
|*  1 |  HASH JOIN RIGHT ANTI |          | 50012 |  2246K|   145   (1)| 00:00:01 |
|   2 |   TABLE ACCESS FULL   | SALGRADE |   999 |  2997 |     3   (0)| 00:00:01 |
|*  3 |   HASH JOIN           |          | 50012 |  2100K|   142   (1)| 00:00:01 |
|   4 |    TABLE ACCESS FULL  | DEPT     |   507 |  7098 |     3   (0)| 00:00:01 |
|   5 |    TABLE ACCESS FULL  | EMP      | 50111 |  1419K|   139   (1)| 00:00:01 |
-------------------------------------------------------------------------------

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

   1 - access("E"."SAL"="HISAL")
   3 - access("E"."DEPTNO"="D"."DEPTNO")
```

No_Use_Hash(e d)

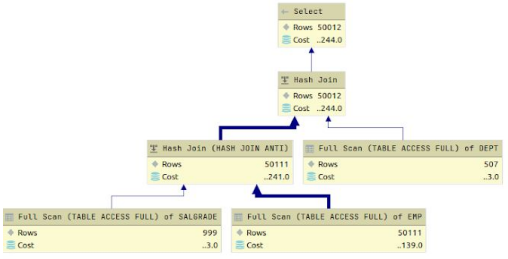


Result 7 — 21 rows — Comma-...d (CSV)

PLAN_TABLE_OUTPUT

```
Plan hash value: 2287891773

-----------------------------------------------------------------------------------------
| Id  | Operation                     | Name         | Rows  | Bytes |TempSpc| Cost (%CPU)| Time
-----------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT              |              | 50012 |  2246K|       |   548   (1)| 00:00:01
|*  1 |  HASH JOIN RIGHT ANTI         |              | 50012 |  2246K|       |   548   (1)| 00:00:01
|   2 |   TABLE ACCESS FULL           | SALGRADE     |   999 |  2997 |       |     3   (0)| 00:00:01
|   3 |   MERGE JOIN                  |              | 50012 |  2100K|       |   545   (1)| 00:00:01
|   4 |    TABLE ACCESS BY INDEX ROWID| DEPT         |   507 |  7098 |       |     4   (0)| 00:00:01
|   5 |     INDEX FULL SCAN           | SYS_C0063105 |   507 |       |       |     1   (0)| 00:00:01
|*  6 |    SORT JOIN                  |              | 50111 |  1419K|  4344K|   541   (1)| 00:00:01
|   7 |     TABLE ACCESS FULL         | EMP          | 50111 |  1419K|       |   139   (1)| 00:00:01
-----------------------------------------------------------------------------------------

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

   1 - access("E"."SAL"="HISAL")
   6 - access("E"."DEPTNO"="D"."DEPTNO")
       filter("E"."DEPTNO"="D"."DEPTNO")
```

/*+ ORDERED */(prima join tra emp e salgrade, e poi join con dept)



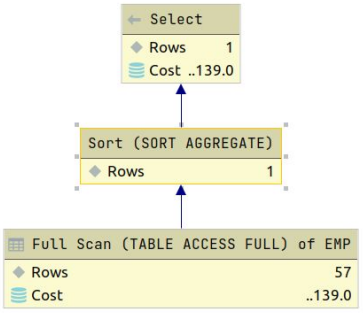

```
-------------------------------------------------------------------------------
| Id  | Operation             | Name     | Rows  | Bytes |TempSpc| Cost (%CPU)| Time     |
-------------------------------------------------------------------------------
|   0 | SELECT STATEMENT      |          | 50012 |  2246K|       |   244   (1)| 00:00:01 |
|*  1 |  HASH JOIN            |          | 50012 |  2246K|       |   244   (1)| 00:00:01 |
|   2 |   TABLE ACCESS FULL   | DEPT     |   507 |  7098 |       |     3   (0)| 00:00:01 |
|*  3 |   HASH JOIN ANTI      |          | 50111 |  1565K|  2008K|   241   (1)| 00:00:01 |
|   4 |    TABLE ACCESS FULL  | EMP      | 50111 |  1419K|       |   139   (1)| 00:00:01 |
|   5 |    TABLE ACCESS FULL  | SALGRADE |   999 |  2997 |       |     3   (0)| 00:00:01 |
-------------------------------------------------------------------------------

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

   1 - access("E"."DEPTNO"="D"."DEPTNO")
   3 - access("E"."SAL"="HISAL")
```

Query #4

Si definiscano una o più strutture secondarie (indici) che permettano l'ottimizzazione della seguente query. Si analizzi con particolare attenzione il cambiamento nel piano di esecuzione creando due indici sugli attributi interessati dall'interrogazione.

```
select avg(e.sal)
from emp e
where e.deptno < 10 and
e.sal > 100 and e.sal < 200;
```

Default oppure deptno B-tree index oppure sal B-tree index



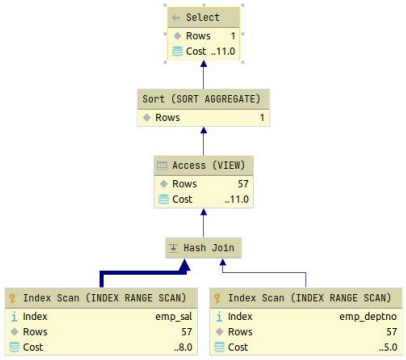
14 rows

PLAN_TABLE_OUTPUT

```
| Id  | Operation          | Name | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------
|   0 | SELECT STATEMENT   |      |     1 |     6 |   139   (1)| 00:00:01 |
|   1 |  SORT AGGREGATE    |      |     1 |     6 |            |          |
|*  2 |   TABLE ACCESS FULL| EMP  |    57 |   342 |   139   (1)| 00:00:01 |
-------------------------------------------------------------------------------

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

   2 - filter("E"."DEPTNO"<10 AND "E"."SAL"<200 AND "E"."SAL">100)
```

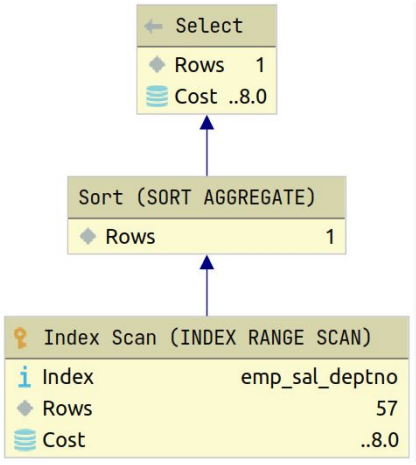
(SORT AGGREGATE does not actually involve a sort. It is used when aggregates are being computed across the whole set of rows)

Deptno B-tree index + sal B-tree index




```
| Id  | Operation           | Name              | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT    |                   |     1 |     6 |    11   (0)| 00:00:01 |
|   1 |  SORT AGGREGATE     |                   |     1 |     6 |            |          |
|*  2 |   VIEW              | index$_join$_001  |    57 |   342 |    11   (0)| 00:00:01 |
|*  3 |    HASH JOIN        |                   |       |       |            |          |
|*  4 |     INDEX RANGE SCAN| EMP_DEPTNO        |    57 |   342 |     5  (20)| 00:00:01 |
|*  5 |     INDEX RANGE SCAN| EMP_SAL           |    57 |   342 |     8  (13)| 00:00:01 |
-------------------------------------------------------------------------------------

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

   2 - filter("E"."DEPTNO"<10 AND "E"."SAL"<200 AND "E"."SAL">100)
   3 - access(ROWID=ROWID)
   4 - access("E"."DEPTNO"<10)
   5 - access("E"."SAL">100 AND "E"."SAL"<200)
```

Sal,deptno B-tree index



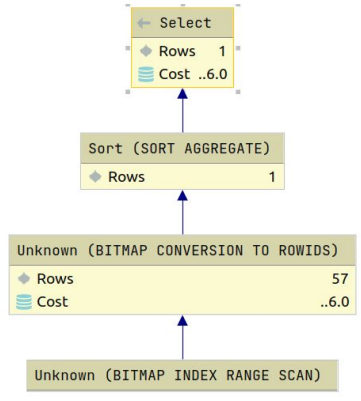
Result 30

15 rows

PLAN_TABLE_OUTPUT

```
| Id  | Operation          | Name             | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT   |                  |     1 |     6 |     8   (0)| 00:00:01 |
|   1 |  SORT AGGREGATE    |                  |     1 |     6 |            |          |
|*  2 |   INDEX RANGE SCAN| EMP_SAL_DEPTNO    |    57 |   342 |     8   (0)| 00:00:01 |
-------------------------------------------------------------------------------------

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

   2 - access("E"."SAL">100 AND "E"."SAL"<200 AND "E"."DEPTNO"<10)
       filter("E"."DEPTNO"<10)
```

Deptno,sal BITMAP index (deptno più selettivo!)

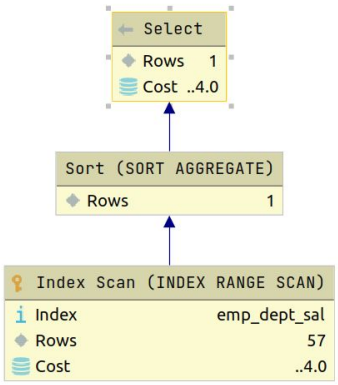


16 rows

PLAN_TABLE_OUTPUT

```
Plan hash value: 1840577623

-------------------------------------------------------------------------------------
| Id  | Operation                    | Name           | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT             |                |     1 |     6 |     6   (0)| 00:00:01 |
|   1 |  SORT AGGREGATE              |                |     1 |     6 |            |          |
|   2 |   BITMAP CONVERSION TO ROWIDS|                |    57 |   342 |     6   (0)| 00:00:01 |
|*  3 |    BITMAP INDEX RANGE SCAN   | EMP_DEPTNO_SAL |       |       |            |          |
-------------------------------------------------------------------------------------

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

   3 - access("E"."DEPTNO"<10 AND "E"."SAL"<200)
       filter("E"."SAL">100 AND "E"."DEPTNO"<10 AND "E"."SAL"<200)
```

Deptno,sal B-tree index (deptno più selettivo!)



Result 17

15 rows | Comma-...d (CSV)

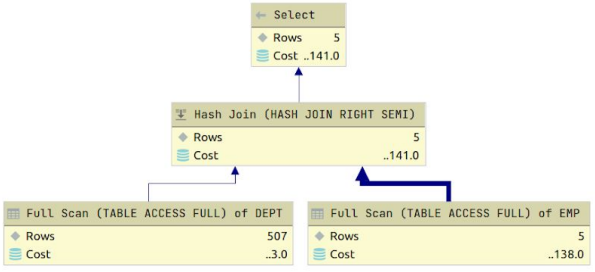
```
PLAN_TABLE_OUTPUT
Plan hash value: 531818999

-------------------------------------------------------------------------------
| Id  | Operation          | Name          | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------
|   0 | SELECT STATEMENT   |               |     1 |     6 |     4   (0)| 00:00:01 |
|   1 |  SORT AGGREGATE    |               |     1 |     6 |            |          |
|*  2 |   INDEX RANGE SCAN| EMP_DEPT_SAL |    57 |   342 |     4   (0)| 00:00:01 |
-------------------------------------------------------------------------------

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

   2 - access("E"."SAL">100 AND "E"."DEPTNO"<10 AND "E"."SAL"<200)
       filter("E"."SAL"<200 AND "E"."SAL">100)
```

Query #5

Si definiscano una o più strutture secondarie (indici) che permettano l'ottimizzazione della seguente query:

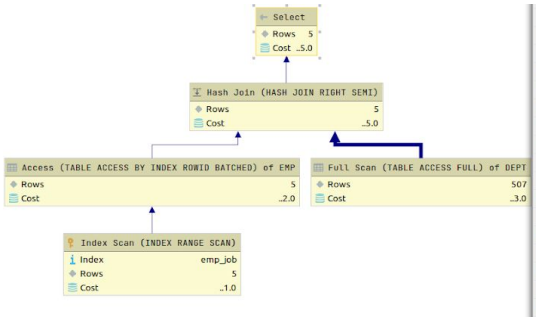
```
select dname
from dept
where deptno in (select deptno
from emp
where job = 'PHILOSOPHER');
```

Default o indice su dept(dname)



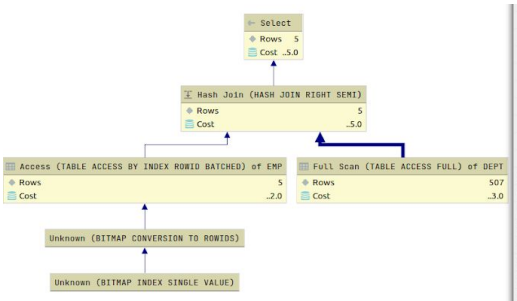

```
Plan hash value: 1870428981

-------------------------------------------------------------------------------
| Id  | Operation           | Name | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------
|   0 | SELECT STATEMENT    |      |     5 |   130 |   141   (0)| 00:00:01 |
|*  1 |  HASH JOIN RIGHT SEMI|      |     5 |   130 |   141   (0)| 00:00:01 |
|*  2 |   TABLE ACCESS FULL | EMP  |     5 |    60 |   138   (0)| 00:00:01 |
|   3 |   TABLE ACCESS FULL | DEPT |   507 |  7098 |     3   (0)| 00:00:01 |
-------------------------------------------------------------------------------

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

   1 - access("DEPTNO"="DEPTNO")
   2 - filter("JOB"='PHILOSOPHER')
```

Job B-tree index



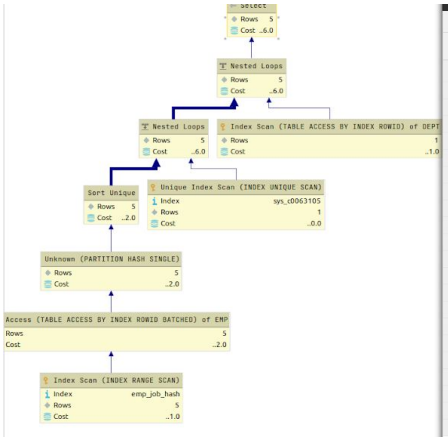

```
Plan hash value: 262934622

-------------------------------------------------------------------------------------------
| Id  | Operation                           | Name    | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT                    |         |     5 |   130 |     5   (0)| 00:00:01 |
|*  1 |  HASH JOIN RIGHT SEMI               |         |     5 |   130 |     5   (0)| 00:00:01 |
|   2 |   TABLE ACCESS BY INDEX ROWID BATCHED| EMP     |     5 |    60 |     2   (0)| 00:00:01 |
|*  3 |    INDEX RANGE SCAN                 | EMP_JOB |     5 |       |     1   (0)| 00:00:01 |
|   4 |   TABLE ACCESS FULL                 | DEPT    |   507 |  7098 |     3   (0)| 00:00:01 |
-------------------------------------------------------------------------------------------

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

   1 - access("DEPTNO"="DEPTNO")
   3 - access("JOB"='PHILOSOPHER')
```

Job BITMAP index




```
Plan hash value: 2288164076

-----------------------------------------------------------------------------------------------
| Id  | Operation                           | Name           | Rows  | Bytes | Cost (%CPU)| Time     |
-----------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT                    |                |     5 |   130 |     5   (0)| 00:00:01 |
|*  1 |  HASH JOIN RIGHT SEMI               |                |     5 |   130 |     5   (0)| 00:00:01 |
|   2 |   TABLE ACCESS BY INDEX ROWID BATCHED| EMP            |     5 |    60 |     2   (0)| 00:00:01 |
|   3 |    BITMAP CONVERSION TO ROWIDS      |                |       |       |            |          |
|*  4 |     BITMAP INDEX SINGLE VALUE       | EMP_JOB_BITMAP |       |       |            |          |
|   5 |   TABLE ACCESS FULL                 | DEPT           |   507 |  7098 |     3   (0)| 00:00:01 |
-----------------------------------------------------------------------------------------------

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

   1 - access("DEPTNO"="DEPTNO")
   4 - access("JOB"='PHILOSOPHER')
```

Job HASH index Nested Loop

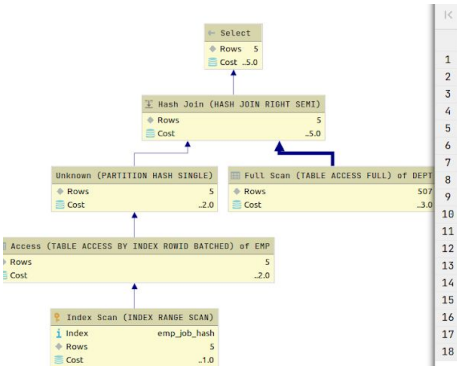


Result 44

21 rows · Comma-...d (CSV)

```
PLAN_TABLE_OUTPUT
n hash value: 3293949767

-------------------------------------------------------------------------------------------------------------
d  | Operation                           | Name          | Rows  | Bytes | Cost (%CPU)| Time     | Pstart| Pstop
-------------------------------------------------------------------------------------------------------------
 0 | SELECT STATEMENT                    |               |     5 |   130 |     6  (17)| 00:00:01 |       |
 1 |  NESTED LOOPS                       |               |     5 |   130 |     6  (17)| 00:00:01 |       |
 2 |   NESTED LOOPS                      |               |     5 |   130 |     6  (17)| 00:00:01 |       |
 3 |    SORT UNIQUE                      |               |     5 |    60 |     2   (0)| 00:00:01 |       |
 4 |     PARTITION HASH SINGLE           |               |     5 |    60 |     2   (0)| 00:00:01 |     1 |     1
 5 |      TABLE ACCESS BY INDEX ROWID BATCHED| EMP       |     5 |    60 |     2   (0)| 00:00:01 |       |
 6 |       INDEX RANGE SCAN              | EMP_JOB_HASH  |     5 |       |     1   (0)| 00:00:01 |     1 |     1
 7 |    INDEX UNIQUE SCAN                | SYS_C0063105  |     1 |       |     0   (0)| 00:00:01 |       |
 8 |   TABLE ACCESS BY INDEX ROWID       | DEPT          |     1 |    14 |     1   (0)| 00:00:01 |       |
-------------------------------------------------------------------------------------------------------------

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

 6 - access("JOB"='PHILOSOPHER')
 7 - access("DEPTNO"="DEPTNO")
```

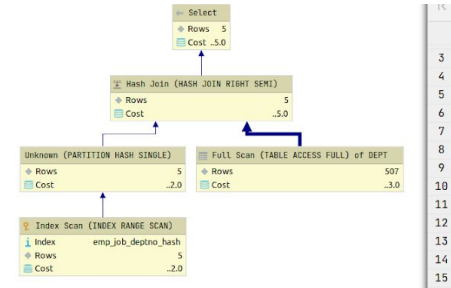
Job HASH index Hash Join



18 rows · Comma-...d (CSV)

```
PLAN_TABLE_OUTPUT
lan hash value: 4254969650

-------------------------------------------------------------------------------------------------------------
 Id  | Operation                           | Name          | Rows  | Bytes | Cost (%CPU)| Time     | Pstart| Pstop
-------------------------------------------------------------------------------------------------------------
   0 | SELECT STATEMENT                    |               |     5 |   130 |     5   (0)| 00:00:01 |       |
*  1 |  HASH JOIN RIGHT SEMI               |               |     5 |   130 |     5   (0)| 00:00:01 |       |
   2 |   PARTITION HASH SINGLE             |               |     5 |    60 |     2   (0)| 00:00:01 |     1 |     1
   3 |    TABLE ACCESS BY INDEX ROWID BATCHED| EMP         |     5 |    60 |     2   (0)| 00:00:01 |       |
*  4 |     INDEX RANGE SCAN                | EMP_JOB_HASH  |     5 |       |     1   (0)| 00:00:01 |     1 |     1
   5 |   TABLE ACCESS FULL                 | DEPT          |   507 |  7098 |     3   (0)| 00:00:01 |       |
-------------------------------------------------------------------------------------------------------------

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

  1 - access("DEPTNO"="DEPTNO")
  4 - access("JOB"='PHILOSOPHER')
```

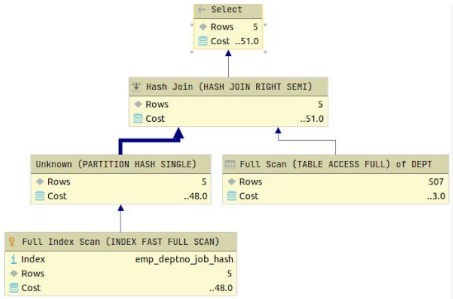
Job,Deptno HASH index Hash Join



17 rows · Comma-...d (CSV)

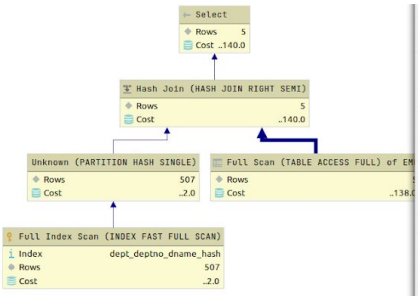
```
PLAN_TABLE_OUTPUT

-------------------------------------------------------------------------------------------------------------
| Id  | Operation              | Name                | Rows  | Bytes | Cost (%CPU)| Time     | Pstart| Pstop |
-------------------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT       |                     |     5 |   130 |     5   (0)| 00:00:01 |       |       |
|*  1 |  HASH JOIN RIGHT SEMI  |                     |     5 |   130 |     5   (0)| 00:00:01 |       |       |
|   2 |   PARTITION HASH SINGLE|                     |     5 |    60 |     2   (0)| 00:00:01 |     1 |     1 |
|*  3 |    INDEX RANGE SCAN    | EMP_JOB_DEPTNO_HASH |     5 |    60 |     2   (0)| 00:00:01 |     1 |     1 |
|   4 |   TABLE ACCESS FULL    | DEPT                |   507 |  7098 |     3   (0)| 00:00:01 |       |       |
-------------------------------------------------------------------------------------------------------------

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

   1 - access("DEPTNO"="DEPTNO")
   3 - access("JOB"='PHILOSOPHER')
```

Deptno,Job HASH index Hash Join



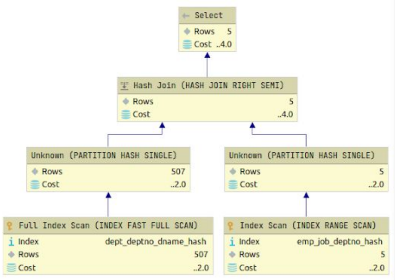

```
PLAN_TABLE_OUTPUT
Plan hash value: 1140949081

-------------------------------------------------------------------------------------------------------
| Id  | Operation              | Name                | Rows  | Bytes | Cost (%CPU)| Time     | Pstart| Pstop |
-------------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT       |                     |     5 |   130 |    51   (0)| 00:00:01 |       |       |
|*  1 |  HASH JOIN RIGHT SEMI  |                     |     5 |   130 |    51   (0)| 00:00:01 |       |       |
|   2 |   PARTITION HASH SINGLE|                     |     5 |    60 |    48   (0)| 00:00:01 |     1 |     1 |
|*  3 |    INDEX FAST FULL SCAN| EMP_DEPTNO_JOB_HASH |     5 |    60 |    48   (0)| 00:00:01 |     1 |     1 |
|   4 |   TABLE ACCESS FULL    | DEPT                |   507 |  7098 |     3   (0)| 00:00:01 |       |       |
-------------------------------------------------------------------------------------------------------

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

   1 - access("DEPTNO"="DEPTNO")
   3 - filter("JOB"='PHILOSOPHER')
```

Deptno,Dname HASH index Hash Join




```
Plan hash value: 1763987074

-------------------------------------------------------------------------------------------------------
| Id  | Operation              | Name                | Rows  | Bytes | Cost (%CPU)| Time     | Pstart| Pstop |
-------------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT       |                     |     5 |   130 |   140   (0)| 00:00:01 |       |       |
|*  1 |  HASH JOIN RIGHT SEMI  |                     |     5 |   130 |   140   (0)| 00:00:01 |       |       |
|*  2 |   TABLE ACCESS FULL    | EMP                 |     5 |    60 |   138   (0)| 00:00:01 |       |       |
|   3 |   PARTITION HASH SINGLE|                     |   507 |  7098 |     2   (0)| 00:00:01 |     1 |     1 |
|   4 |    INDEX FAST FULL SCAN| DEPT_DEPTNO_DNAME_HASH |   507 |  7098 |     2   (0)| 00:00:01 |     1 |     1 |
-------------------------------------------------------------------------------------------------------

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

   1 - access("DEPTNO"="DEPTNO")
   2 - filter("JOB"='PHILOSOPHER')
```

Job,Deptno HASH index + Deptno,Dname HASH index Hash Join



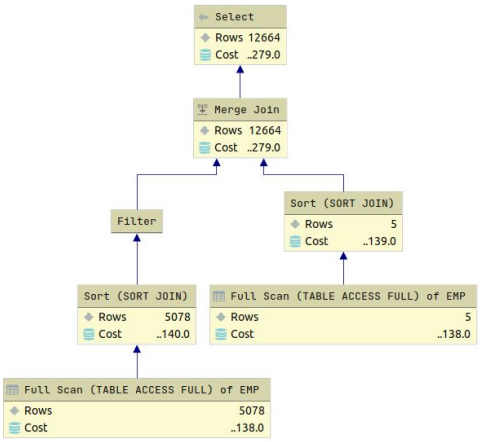

```
Plan hash value: 1825229883

-------------------------------------------------------------------------------------------------------
| Id  | Operation              | Name                | Rows  | Bytes | Cost (%CPU)| Time     | Pstart| Pstop |
-------------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT       |                     |     5 |   130 |     4   (0)| 00:00:01 |       |       |
|*  1 |  HASH JOIN RIGHT SEMI  |                     |     5 |   130 |     4   (0)| 00:00:01 |       |       |
|   2 |   PARTITION HASH SINGLE|                     |     5 |    60 |     2   (0)| 00:00:01 |     1 |     1 |
|*  3 |    INDEX RANGE SCAN    | EMP_JOB_DEPTNO_HASH |     5 |    60 |     2   (0)| 00:00:01 |     1 |     1 |
|   4 |   PARTITION HASH SINGLE|                     |   507 |  7098 |     2   (0)| 00:00:01 |     1 |     1 |
|   5 |    INDEX FAST FULL SCAN| DEPT_DEPTNO_DNAME_HASH |   507 |  7098 |     2   (0)| 00:00:01 |     1 |     1 |
-------------------------------------------------------------------------------------------------------

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

   1 - access("DEPTNO"="DEPTNO")
   3 - access("JOB"='PHILOSOPHER')
```

Query #6

Si definiscano una o più strutture secondarie (indici) che permettano l'ottimizzazione della seguente query (rimuovere eventuali indici già esistenti per confrontare le performance della query con e senza indici):

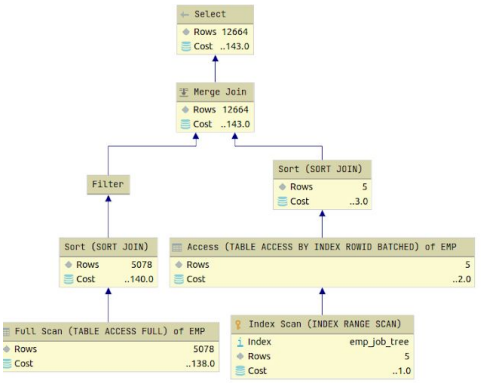
```
select e1.ename, e1.empno, e1.sal, e2.ename, e2.empno, e2.sal
from emp e1, emp e2
where e1.ename <> e2.ename and e1.sal < e2.sal
and e1.job = 'PHILOSOPHER' and e2.job = 'ENGINEER';
```

Default o SAL + Ename B-tree index



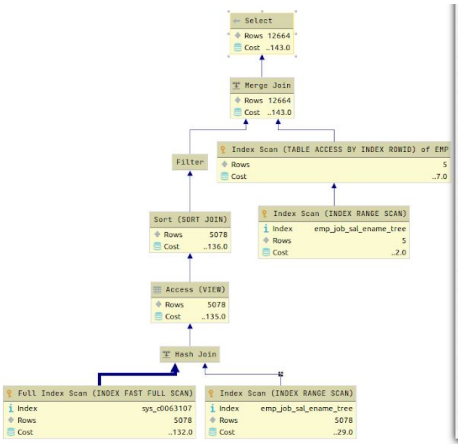

```
Plan hash value: 3733349388

-------------------------------------------------------------------------------
| Id  | Operation            | Name | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------
|   0 | SELECT STATEMENT     |      | 12664 |   742K|   279   (2)| 00:00:01 |
|   1 |  MERGE JOIN          |      | 12664 |   742K|   279   (2)| 00:00:01 |
|   2 |   SORT JOIN          |      |     5 |   150 |   139   (1)| 00:00:01 |
|*  3 |    TABLE ACCESS FULL | EMP  |     5 |   150 |   138   (0)| 00:00:01 |
|*  4 |   FILTER             |      |       |       |            |          |
|*  5 |    SORT JOIN         |      |  5078 |   148K|   140   (2)| 00:00:01 |
|*  6 |     TABLE ACCESS FULL| EMP  |  5078 |   148K|   138   (0)| 00:00:01 |
-------------------------------------------------------------------------------

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

   3 - filter("E1"."JOB"='PHILOSOPHER')
   4 - filter("E1"."ENAME"<>"E2"."ENAME")
   5 - access("E1"."SAL"<"E2"."SAL")
       filter("E1"."SAL"<"E2"."SAL")
   6 - filter("E2"."JOB"='ENGINEER')
```

Job B-Tree index




```

-------------------------------------------------------------------------------------------
| Id  | Operation                           | Name         | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT                    |              | 12664 |   742K|   143   (3)| 00:00:01 |
|   1 |  MERGE JOIN                         |              | 12664 |   742K|   143   (3)| 00:00:01 |
|   2 |   SORT JOIN                         |              |     5 |   150 |     3  (34)| 00:00:01 |
|   3 |    TABLE ACCESS BY INDEX ROWID BATCHED| EMP        |     5 |   150 |     2   (0)| 00:00:01 |
|*  4 |     INDEX RANGE SCAN                | EMP_JOB_TREE |     5 |       |     1   (0)| 00:00:01 |
|*  5 |   FILTER                            |              |       |       |            |          |
|*  6 |    SORT JOIN                        |              |  5078 |   148K|   140   (2)| 00:00:01 |
|*  7 |     TABLE ACCESS FULL               | EMP          |  5078 |   148K|   138   (0)| 00:00:01 |
-------------------------------------------------------------------------------------------

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

   4 - access("E1"."JOB"='PHILOSOPHER')
   5 - filter("E1"."ENAME"<>"E2"."ENAME")
   6 - access("E1"."SAL"<"E2"."SAL")
       filter("E1"."SAL"<"E2"."SAL")
   7 - filter("E2"."JOB"='ENGINEER')
```

Job,Sal,Ename B-Tree index




```
| Id  | Operation                     | Name                  | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT              |                       | 12664 |   742K|   143   (2)| 00:00:01 |
|   1 |  MERGE JOIN                   |                       | 12664 |   742K|   143   (2)| 00:00:01 |
|   2 |   TABLE ACCESS BY INDEX ROWID| EMP                   |     5 |   150 |     7   (0)| 00:00:01 |
|*  3 |    INDEX RANGE SCAN           | EMP_JOB_SAL_ENAME_TREE|     5 |       |     2   (0)| 00:00:01 |
|*  4 |   FILTER                      |                       |       |       |            |          |
|*  5 |    SORT JOIN                  |                       |  5078 |   148K|   136   (2)| 00:00:01 |
|*  6 |     VIEW                      | index$_join$_002      |  5078 |   148K|   135   (1)| 00:00:01 |
|*  7 |      HASH JOIN                |                       |       |       |            |          |
|*  8 |       INDEX RANGE SCAN        | EMP_JOB_SAL_ENAME_TREE|  5078 |   148K|    29   (0)| 00:00:01 |
|   9 |       INDEX FAST FULL SCAN    | SYS_C0063107          |  5078 |   148K|   132   (1)| 00:00:01 |
-------------------------------------------------------------------------------------------

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

   3 - access("E1"."JOB"='PHILOSOPHER')
   4 - filter("E1"."ENAME"<>"E2"."ENAME")
   5 - access("E1"."SAL"<"E2"."SAL")
       filter("E1"."SAL"<"E2"."SAL")
   6 - filter("E2"."JOB"='ENGINEER')
   7 - access(ROWID=ROWID)
   8 - access("E2"."JOB"='ENGINEER')
```