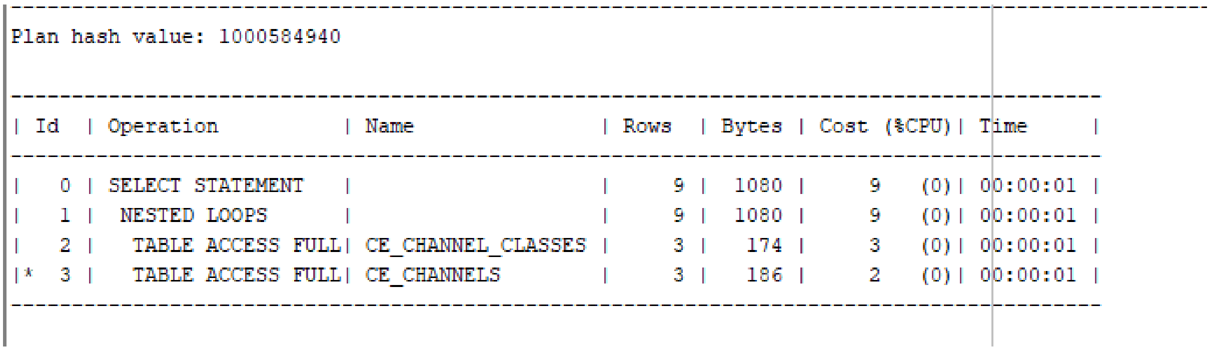
TASK 5 REPORT

I used my 3NF schema for this task

1. NESTED LOOP JOIN

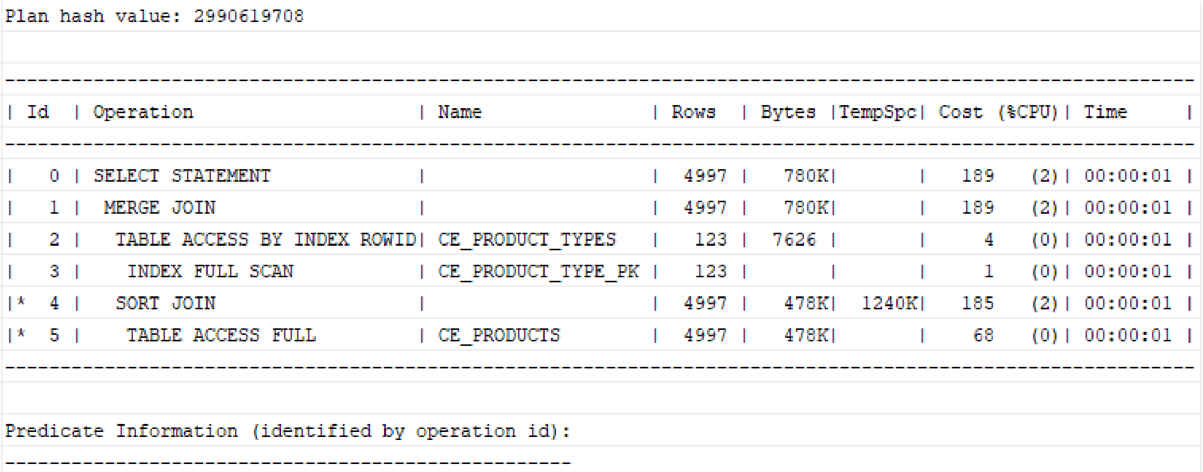
EXPLAIN PLAN FOR
Select /+ USE_NL(cs c) */ **
From ce_channel_classes cs
inner join ce_channels c
on c.channel_class_id=cs.channel_class_id;
*SELECT * FROM table (DBMS_XPLAN.DISPLAY);*



```
Plan hash value: 1000584940

---------------------------------------------------------------------------------------
| Id  | Operation           | Name               | Rows  | Bytes | Cost (%CPU)| Time     |
---------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT    |                    |     9 |  1080 |     9   (0)| 00:00:01 |
|   1 |  NESTED LOOPS       |                    |     9 |  1080 |     9   (0)| 00:00:01 |
|   2 |   TABLE ACCESS FULL| CE_CHANNEL_CLASSES |     3 |   174 |     3   (0)| 00:00:01 |
|*  3 |   TABLE ACCESS FULL| CE_CHANNELS        |     3 |   186 |     2   (0)| 00:00:01 |
---------------------------------------------------------------------------------------
```

2. SORT-MERGE JOIN

EXPLAIN PLAN FOR
Select /+ USE_MERGE(p pt)*/**
From ce_products p
inner join ce_product_types pt
on p.product_type_id=pt.product_type_id
Where p.unit_cost>1000;
*SELECT * FROM table (DBMS_XPLAN.DISPLAY);*

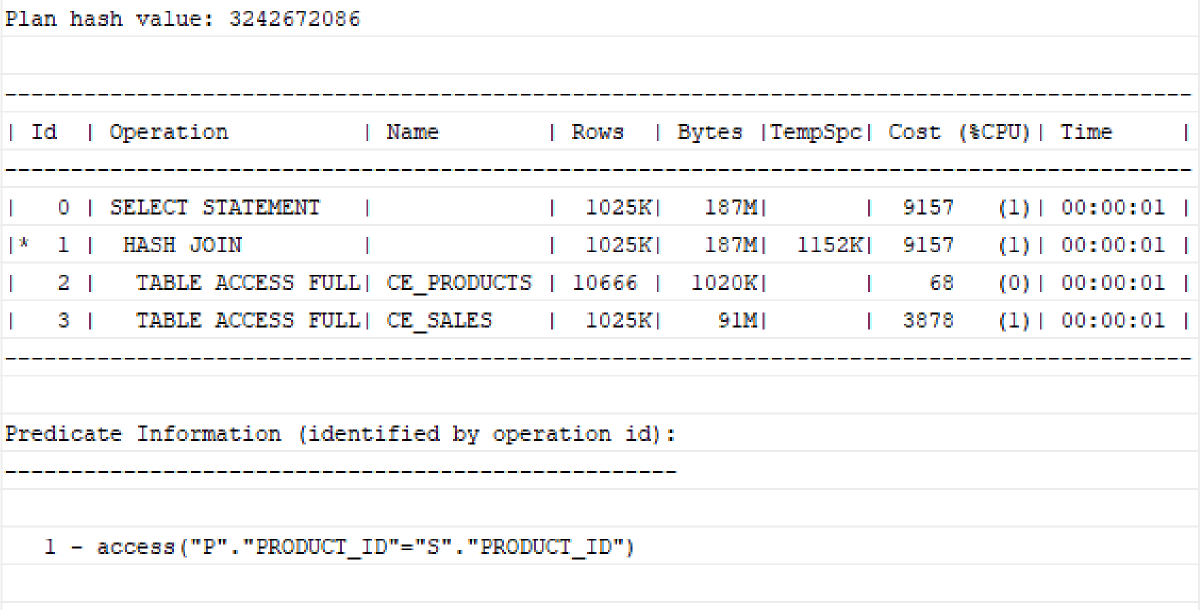


```
Plan hash value: 2990619708


----------------------------------------------------------------------------------------------------
| Id  | Operation                     | Name               | Rows  | Bytes |TempSpc| Cost (%CPU)| Time     |
----------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT              |                    |  4997 |   780K|       |   189   (2)| 00:00:01 |
|   1 |  MERGE JOIN                   |                    |  4997 |   780K|       |   189   (2)| 00:00:01 |
|   2 |   TABLE ACCESS BY INDEX ROWID| CE_PRODUCT_TYPES   |   123 |  7626 |       |     4   (0)| 00:00:01 |
|   3 |    INDEX FULL SCAN            | CE_PRODUCT_TYPE_PK |   123 |       |       |     1   (0)| 00:00:01 |
|*  4 |   SORT JOIN                   |                    |  4997 |   478K|  1240K|   185   (2)| 00:00:01 |
|*  5 |    TABLE ACCESS FULL          | CE_PRODUCTS        |  4997 |   478K|       |    68   (0)| 00:00:01 |
----------------------------------------------------------------------------------------------------

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

3: HASH JOIN

EXPLAIN PLAN FOR
Select /+ USE_HASH(p pt)*/**
From ce_sales s
inner join ce_products p

on p.product_id=s.product_id;
*SELECT * FROM table (DBMS_XPLAN.DISPLAY);*

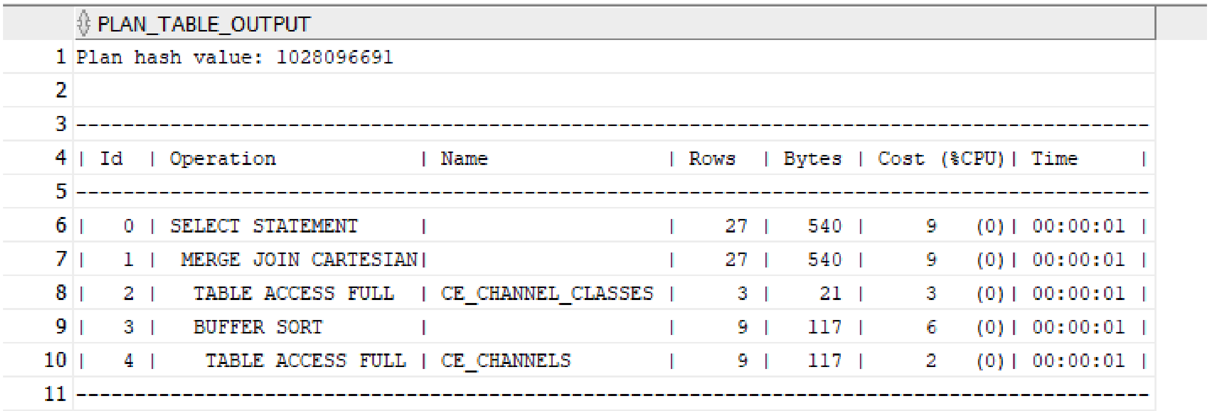


```
Plan hash value: 3242672086

-------------------------------------------------------------------------------------------
| Id  | Operation          | Name         | Rows  | Bytes |TempSpc| Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT   |              |  1025K|   187M|       |  9157   (1)| 00:00:01 |
|*  1 |  HASH JOIN         |              |  1025K|   187M|  1152K|  9157   (1)| 00:00:01 |
|   2 |   TABLE ACCESS FULL| CE_PRODUCTS  | 10666 |  1020K|       |    68   (0)| 00:00:01 |
|   3 |   TABLE ACCESS FULL| CE_SALES     |  1025K|    91M|       |  3878   (1)| 00:00:01 |
-------------------------------------------------------------------------------------------

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

   1 - access("P"."PRODUCT_ID"="S"."PRODUCT_ID")
```

4. CARTESIAN JOIN

ORDERED HINT AND WRONG PREDICATE ORDER OR JOIN TO THE TABLE WITH 0 ROWS.

Select ce_channel_classes.channel_class, ce_channels.channel_description, ce_channels.channel_id from ce_channel_classes
EXPLAIN PLAN FOR
Select /+ ORDERED*/ **
From ce_channel_classes ,ce_channels
where ce_channel_classes.channel_class_id=ce_channels.channel_class_id;
and ce_channel_classes.channel_class='tDirect';
*SELECT * FROM table (DBMS_XPLAN.DISPLAY);*

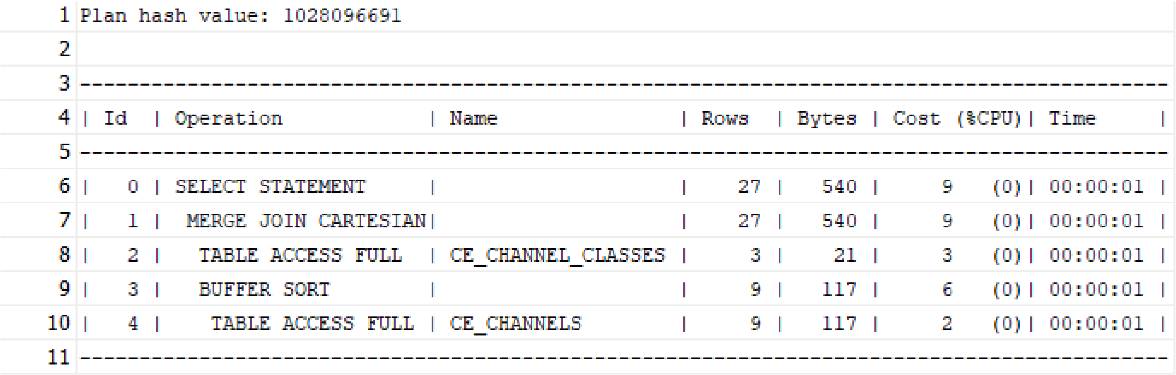


PLAN_TABLE_OUTPUT

```
Plan hash value: 1028096691

-----------------------------------------------------------------------------------------
| Id  | Operation             | Name               | Rows  | Bytes | Cost (%CPU)| Time     |
-----------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT      |                    |    27 |   540 |     9   (0)| 00:00:01 |
|   1 |  MERGE JOIN CARTESIAN|                    |    27 |   540 |     9   (0)| 00:00:01 |
|   2 |   TABLE ACCESS FULL  | CE_CHANNEL_CLASSES |     3 |    21 |     3   (0)| 00:00:01 |
|   3 |   BUFFER SORT        |                    |     9 |   117 |     6   (0)| 00:00:01 |
|   4 |    TABLE ACCESS FULL | CE_CHANNELS        |     9 |   117 |     2   (0)| 00:00:01 |
-----------------------------------------------------------------------------------------
```

W/O JOIN CONDITIONS.

EXPLAIN PLAN FOR
Select ce_channel_classes.channel_class, ce_channels.channel_description, ce_channels.channel_id
From ce_channel_classes ,ce_channels;
*SELECT * FROM table (DBMS_XPLAN.DISPLAY);*

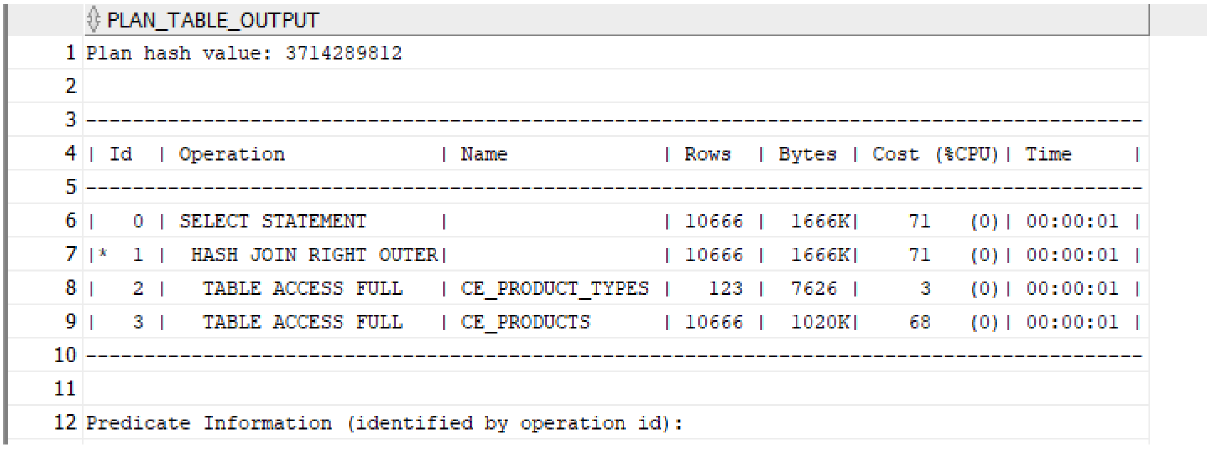


```
Plan hash value: 1028096691

-------------------------------------------------------------------------------------------
| Id  | Operation             | Name                | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT      |                     |    27 |   540 |     9   (0)| 00:00:01 |
|   1 |  MERGE JOIN CARTESIAN|                     |    27 |   540 |     9   (0)| 00:00:01 |
|   2 |   TABLE ACCESS FULL   | CE_CHANNEL_CLASSES |     3 |    21 |     3   (0)| 00:00:01 |
|   3 |   BUFFER SORT         |                     |     9 |   117 |     6   (0)| 00:00:01 |
|   4 |    TABLE ACCESS FULL  | CE_CHANNELS         |     9 |   117 |     2   (0)| 00:00:01 |
-------------------------------------------------------------------------------------------
```

5. LEFT/RIGHT OUTER JOINS

ANSI SYNTAX (LEFT/RIGHT JOIN)

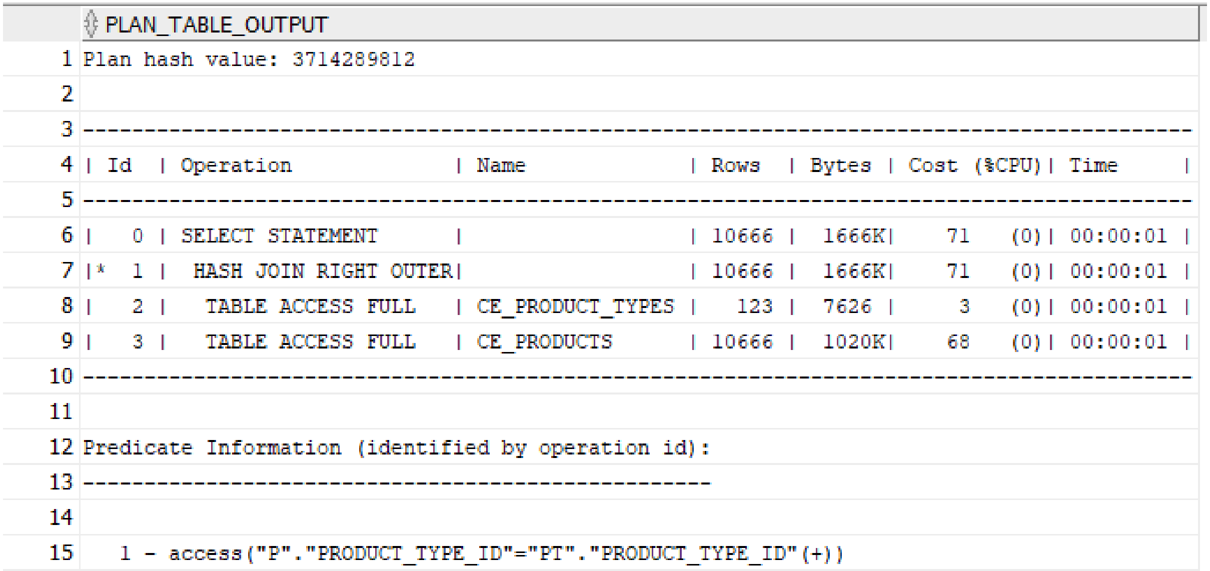
EXPLAIN PLAN FOR
*Select **
From ce_products p
left outer join ce_product_types pt
on p.product_type_id=pt.product_type_id;
*SELECT * FROM table (DBMS_XPLAN.DISPLAY);*



```
Plan hash value: 3714289812

-------------------------------------------------------------------------------------------
| Id  | Operation             | Name             | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT      |                  | 10666 |  1666K|    71   (0)| 00:00:01 |
|*  1 |  HASH JOIN RIGHT OUTER|                  | 10666 |  1666K|    71   (0)| 00:00:01 |
|   2 |   TABLE ACCESS FULL   | CE_PRODUCT_TYPES |   123 |  7626 |     3   (0)| 00:00:01 |
|   3 |   TABLE ACCESS FULL   | CE_PRODUCTS      | 10666 |  1020K|    68   (0)| 00:00:01 |
-------------------------------------------------------------------------------------------

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

ORACLE SYNTAX (+)

EXPLAIN PLAN FOR
*Select **
From ce_products p, ce_product_types pt
Where p.product_type_id=pt.product_type_id(+);
*SELECT * FROM table (DBMS_XPLAN.DISPLAY);*

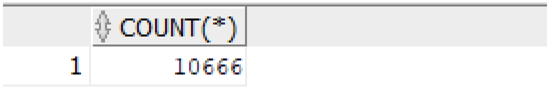


```
PLAN_TABLE_OUTPUT
Plan hash value: 3714289812

-----------------------------------------------------------------------------------------
| Id  | Operation             | Name             | Rows  | Bytes | Cost (%CPU)| Time     |
-----------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT      |                  | 10666 |  1666K|    71   (0)| 00:00:01 |
|*  1 |  HASH JOIN RIGHT OUTER|                  | 10666 |  1666K|    71   (0)| 00:00:01 |
|   2 |   TABLE ACCESS FULL   | CE_PRODUCT_TYPES |   123 |  7626 |     3   (0)| 00:00:01 |
|   3 |   TABLE ACCESS FULL   | CE_PRODUCTS      | 10666 |  1020K|    68   (0)| 00:00:01 |
-----------------------------------------------------------------------------------------

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

   1 - access("P"."PRODUCT_TYPE_ID"="PT"."PRODUCT_TYPE_ID"(+))
```

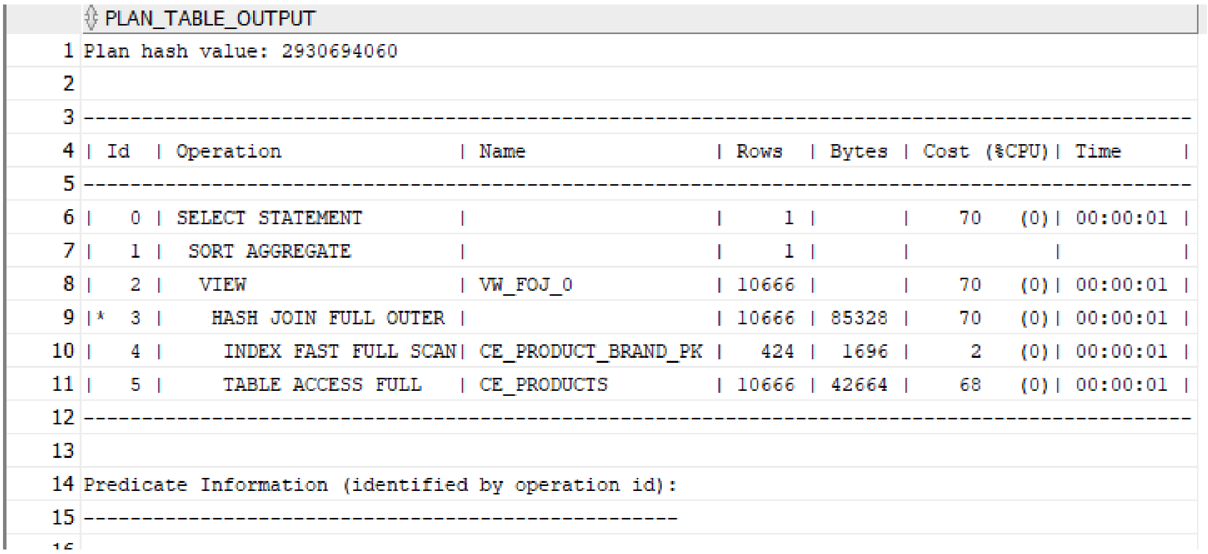
6. FULL OUTER JOIN

ANSI SYNTAX (OUTER JOIN)

EXPLAIN PLAN FOR
Select count(*)
From ce_products p
full outer join ce_brands b
on p.product_brand_id=b.product_brand_id;
*SELECT * FROM table (DBMS_XPLAN.DISPLAY);*



	COUNT(*)
1	10666

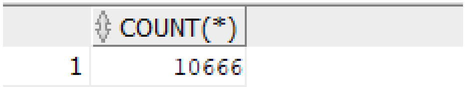


```
PLAN_TABLE_OUTPUT
Plan hash value: 2930694060

-----------------------------------------------------------------------------------------
| Id  | Operation             | Name               | Rows  | Bytes | Cost (%CPU)| Time     |
-----------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT      |                    |     1 |       |    70   (0)| 00:00:01 |
|   1 |  SORT AGGREGATE       |                    |     1 |       |            |          |
|   2 |   VIEW                | VW_FOJ_0           | 10666 |       |    70   (0)| 00:00:01 |
|*  3 |    HASH JOIN FULL OUTER |                  | 10666 | 85328 |    70   (0)| 00:00:01 |
|   4 |     INDEX FAST FULL SCAN| CE_PRODUCT_BRAND_PK |   424 |  1696 |     2   (0)| 00:00:01 |
|   5 |     TABLE ACCESS FULL  | CE_PRODUCTS        | 10666 | 42664 |    68   (0)| 00:00:01 |
-----------------------------------------------------------------------------------------

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

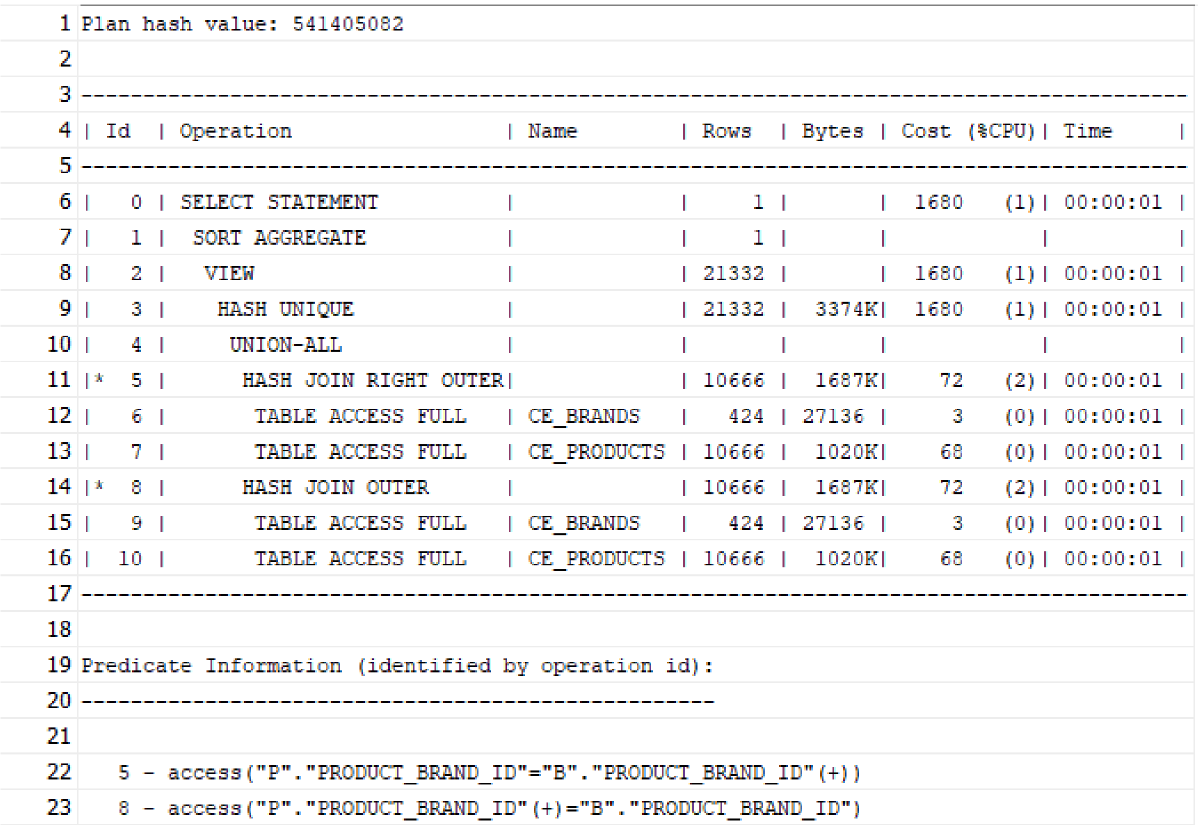
ORACLE SYNTAX (+)

EXPLAIN PLAN FOR
Select count (*)

```
From (
   Select *
   From ce_products p, ce_brands b
   Where p.product_brand_id=b.product_brand_id(+)
   union
Select *
   From ce_products p, ce_brands b
   Where p.product_brand_id(+)=b.product_brand_id);
SELECT * FROM table (DBMS_XPLAN.DISPLAY);
```



	COUNT(*)
1	10666



```
Plan hash value: 541405082

-----------------------------------------------------------------------------------------
| Id  | Operation                 | Name        | Rows  | Bytes | Cost (%CPU)| Time     |
-----------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT          |             |     1 |       |  1680   (1)| 00:00:01 |
|   1 |  SORT AGGREGATE           |             |     1 |       |            |          |
|   2 |   VIEW                    |             | 21332 |       |  1680   (1)| 00:00:01 |
|   3 |    HASH UNIQUE            |             | 21332 |  3374K|  1680   (1)| 00:00:01 |
|   4 |     UNION-ALL             |             |       |       |            |          |
|*  5 |      HASH JOIN RIGHT OUTER|             | 10666 |  1687K|    72   (2)| 00:00:01 |
|   6 |       TABLE ACCESS FULL   | CE_BRANDS   |   424 | 27136 |     3   (0)| 00:00:01 |
|   7 |       TABLE ACCESS FULL   | CE_PRODUCTS | 10666 |  1020K|    68   (0)| 00:00:01 |
|*  8 |      HASH JOIN OUTER      |             | 10666 |  1687K|    72   (2)| 00:00:01 |
|   9 |       TABLE ACCESS FULL   | CE_BRANDS   |   424 | 27136 |     3   (0)| 00:00:01 |
|  10 |       TABLE ACCESS FULL   | CE_PRODUCTS | 10666 |  1020K|    68   (0)| 00:00:01 |
-----------------------------------------------------------------------------------------

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

   5 - access("P"."PRODUCT_BRAND_ID"="B"."PRODUCT_BRAND_ID"(+))
   8 - access("P"."PRODUCT_BRAND_ID"(+)="B"."PRODUCT_BRAND_ID")
```

7. MERGE

```
MERGE INTO ce_products ch
   USING (SELECT DISTINCT
         c.product_id AS product_srcid,
         'personnel_sales' AS product_source_system,
         'src_products' AS product_source_entity,
         COALESCE(c.product_name,'N/A') AS product,
         COALESCE(c.unit_price,'-1') AS unit_price,
         COALESCE(c.unit_cost,'-1') AS unit_cost,
         COALESCE(c2.product_brand_id,-1) AS product_brand_id,
         COALESCE(c3.product_type_id,-1) AS product_type_id

   FROM  SA_SOURCE_SYSTEM_1.src_products c
   LEFT JOIN bl_3nf.ce_brands c2
```

```
    ON c.brand_id=c2.product_brand_srcid
    LEFT JOIN bl_3nf.ce_product_types c3
    on c.product_type_id=c3.product_type_srcid) t
         ON (ch.product_srcid = t.product_srcid
         AND ch.product_source_system = t.product_source_system
         AND ch.product_source_entity = t.product_source_entity)
    WHEN  MATCHED THEN
        Update set
        product=product, unit_price=unit_price, unit_cost=unit_cost,
product_brand_id=product_brand_id, product_type_id=product_type_id,ta_update_dt=SYSDATE
    WHEN NOT MATCHED THEN
         INSERT(product_id, product_srcid, product_source_system, product_source_entity,
product, unit_price, unit_cost, product_brand_id, product_type_id,ta_update_dt, ta_insert_dt)
        VALUES(seq_ce_product_id.NEXTVAL,t.product_srcid, t.product_source_system,
t.product_source_entity, t.product, t.unit_price, t.unit_cost, t.product_brand_id,
t.product_type_id,SYSDATE,SYSDATE);
```

8. A LIST OF COMBINATION OF DIFFERENT TABLES

Table “A”	Table “B”	Join type description
Small Table w/o index on join field	Small Table w/o index on join field	Nested Loops
Small Table w/o index on join field	Small Table with index on join field	Nested Loops
Small Table w/o index on join field	Big Table with index on join field	Nested Loops
Small Table w/o index on join field	Big Table w/o index on join field	Hash Join
Big table, but can fit comletely in PDA	Big Table w/o index on join field	Hash Join
Big table, but cannot fit comletely in PDA	Big Table w/o index on join field	Sort Merge Join
Big table, use condition < >	Big table	Sort Merge Join
Big table ordered by join column	Big table ordered by join column	Sort Merge Join
Big table	0 rows table	Cartesian Join
Big table	Big table	Cartesian only if we use CROSS JOIN