SQL JOIN Condition moved to with where clause produces differences

Asked 5 years, 1 month ago   Active 5 years, 1 month ago   Viewed 772 times

Query 1

1

1

```
select count(1)
from sdb_snmp_sysdata s
  left join sdb_snmp_entphysicaltable e on s.source = e.source  **and e.class = 3**
  left join SDB_DF_DEVICE_DNS dns on dns.source = s.source
  left join sdb_fdb_node f on upper(f.oldnodeid) = upper(dns.dns_name)
  where (regexp_like(s.descr, 'NFXS-F FANT-F ALCATEL-LUCENT|Motorola APEX3000')
  or regexp_like(e.descr, 'Motorola BSR64000 HD 100A Redundant Chassis|AS2511-RJ
chassis')
  or trim(e.ModelName) in ('RFGW1', 'ARCT01949', 'ARCT03253', 'UBR10012', 'WS-C3750-
48TS-S', 'WS-C3750V2-48TS-S')
  or e.name like '%Nexus5596 Chassis%')
```

Query 2:

```
select count(1)
    from sdb_snmp_sysdata s
      left join sdb_snmp_entphysicaltable e on s.source = e.source
      left join SDB_DF_DEVICE_DNS dns on dns.source = s.source
      left join sdb_fdb_node f on upper(f.oldnodeid) = upper(dns.dns_name)
      where (regexp_like(s.descr, 'NFXS-F FANT-F ALCATEL-LUCENT|Motorola APEX3000')
      or regexp_like(e.descr, 'Motorola BSR64000 HD 100A Redundant Chassis|AS2511-RJ
chassis')
      or trim(e.ModelName) in ('RFGW1', 'ARCT01949', 'ARCT03253', 'UBR10012', 'WS-
C3750-48TS-S', 'WS-C3750V2-48TS-S')
      or e.name like '%Nexus5596 Chassis%') **and e.class = 3**
```

The above two queries return different number of rows by changing `e.class condition from on clause to where clause`. I am unable to figure out. any help is appreciated.

My Understanding: query 1 left outer join between sysdata and entphysicaltable hash join happens after full scan of individual tables. in the second query 2 join happens after entphysicaltable is reduced to records containing only entphysicaltable.class = 3.

to me the query makes same sense but returns different results.

I can relate to this [question](#) I would like to know a concrete reason.

sql   join   where

Share Improve this question Follow

edited May 23 '17 at 10:27   asked Dec 18 '15 at 6:39

Community ♦   saikumarm
1   1   **1,543**   14   28

Where you move the `e.class = 3` predicate from `ON` to `WHERE` `LEFT JOIN` is converted into an `INNER JOIN`. You have to make it like `WHERE (e.class = 3 OR e.class IS NULL) ...` if you want it to be equivalent. – Giorgos Betsos Dec 18 '15 at 6:49 ✎

2 Answers

Active | Oldest | Votes

The best explanation is on a little example. Let have two tables

3

```
TABLE A
        C1
----------
         1
         2

TABLE B

        C1 C2
---------- -
         1 x
```

Then the query with the filter `B.c2 = 'x'` in the ON clause returns 2 rows

```
select *
from A left outer join B
on A.c1 = B.c1 and B.c2 = 'x';

        C1          C1 C2
---------- ---------- --
         1           1 x
         2
```

while when the filter is moved in the WHERE clause, only one row is delivered

```
select *
from A left outer join B
on A.c1 = B.c1
WHERE B.c2 = 'x';

        C1          C1 C2
---------- ---------- --
         1           1 x
```

The WHERE clause simple overrules the OUTER JOIN row missing logik - wee all know that NULL is not equal 'x', so the second row is discarded.

BWT if you see in the *old join syntax* constructs like `B.c2(+) = 'x'` this is the very same thema.

answered Dec 18 '15 at 8:25

Share Improve this answer Follow

Marmite Bomber

13.1k   3   18   43

If I read your question right, then it simply comes down to how a LEFT JOIN works.

2

The way a (outer) LEFT JOIN works is that it will join what's on your left side with what's on your right side. And then it being an outer join it will try to add NULL values to the right, for the situation where there is no match on the right.

However, by you adding your constraints in the WHERE clause, you're telling the query engine to filter out the rows where there is NULL because they will not match your WHERE clause. If you have the filters in your ON clause - the query engine will not remove/filter out the NULL rows. This happens because the WHERE is 'executed' after the JOINs.

That's why you get different number of rows, because an OUTER join functions differently based on whether you use the ON or the WHERE clause. So if you want the join to include NULL rows, you'll need to use the ON clause.

Share Improve this answer Follow

answered Dec 18 '15 at 6:48

Allan S. Hansen
3,853   16   21