Ambiguity in Left joins (oracle only?)

Asked 16 years, 3 months ago Modified 13 years, 4 months ago Viewed 4k times

4

My boss found a bug in a query I created, and I don't understand the reasoning behind the bug, although the query results prove he's correct. Here's the query (simplified version) before the fix:

```
select PTNO,PTNM,CATCD
from PARTS
left join CATEGORIES on (CATEGORIES.CATCD=PARTS.CATCD);
```

and here it is after the fix:

```
select PTNO,PTNM,PARTS.CATCD
from PARTS
left join CATEGORIES on (CATEGORIES.CATCD=PARTS.CATCD);
```

The bug was, that null values were being shown for column CATCD, i.e. the query results included results from table CATEGORIES instead of PARTS. Here's what I don't understand: if there was ambiguity in the original query, why didn't Oracle throw an error? As far as I understood, in the case of left joins, the "main" table in the query (PARTS) has precedence in ambiguity. Am I wrong, or just not thinking about this problem correctly?

Update:

Here's a revised example, where the ambiguity error is not thrown:

```
CREATE TABLE PARTS (PTNO NUMBER, CATCD NUMBER, SECCD NUMBER);

CREATE TABLE CATEGORIES(CATCD NUMBER);

CREATE TABLE SECTIONS(SECCD NUMBER, CATCD NUMBER);


select PTNO,CATCD
from PARTS
left join CATEGORIES on (CATEGORIES.CATCD=PARTS.CATCD)
left join SECTIONS on (SECTIONS.SECCD=PARTS.SECCD) ;
```

Anybody have a clue?

sql oracle-database join

Share

Improve this question

Follow

edited Sep 12, 2008 at 14:15

asked Sep 12, 2008 at 12:37

Ovesh

5,369 ● 11 ● 56 ● 77

12 Answers

Sorted by: Highest score (default)

6

> Here's the query (simplified version)

I think by simplifying the query you removed the real cause of the bug :-)

What oracle version are you using? Oracle 10g (10.2.0.1.0) gives:

```
create table parts (ptno number , ptnm number , catcd number);
create table CATEGORIES (catcd number);

select PTNO,PTNM,CATCD from PARTS
left join CATEGORIES on (CATEGORIES.CATCD=PARTS.CATCD);
```

I get ORA-00918: column ambiguously defined

Share

Improve this answer

Follow

edited Sep 12, 2008 at 13:30

answered Sep 12, 2008 at 13:06

Pat

36.7k ● 18 ● 74 ● 87

Interesting in SQL server that throws an error (as it should)

```
select id
from sysobjects s
left join syscolumns c on s.id = c.id
```

Server: Msg 209, Level 16, State 1, Line 1 Ambiguous column name 'id'.

```
select id
from sysobjects
left join syscolumns  on sysobjects.id = syscolumns.id
```

Server: Msg 209, Level 16, State 1, Line 1 Ambiguous column name 'id'.

Share Improve this answer Follow

answered Sep 12, 2008 at 13:04

SQLMenace

135k ● 25 ● 211 ● 225

From my experience if you create a query like this the data result will pull CATCD from the right side of the join not the left when there is a field overlap like this.

So since this join will have all records from PARTS with only some pull through from CATEGORIES you will have NULL in the CATCD field any time there is no data on the right side.

By explicitly defining the column as from PARTS (ie left side) you will get a non null value assuming that the field has data in PARTS.

Remember that with LEFT JOIN you are only guarantied data in fields from the left table, there may well be empty columns to the right.

Share Improve this answer Follow

answered Sep 12, 2008 at 13:29

Laith

349 ● 1 ● 7

I understand your answer. But why didn't I get an ambiguity error in the first place? – Ovesh Sep 12, 2008 at 13:45

I'm uncertain why. that was the only strange thing about your question. – Laith Sep 18, 2008 at 20:42

This may be a bug in the Oracle optimizer. I can reproduce the same behavior on the query with 3 tables. Intuitively it does seem that it should produce an error. If I rewrite

2

it in either of the following ways, it does generate an error:

(1) Using old-style outer join

```
select ptno, catcd
from parts, categories, sections
where categories.catcd (+) = parts.catcd
  and sections.seccd (+) = parts.seccd
```

(2) Explicitly isolating the two joins

```
select ptno, catcd
from (
  select ptno, seccd, catcd
  from parts
  left join categories on (categories.CATCD=parts.CATCD)
)
left join sections on (sections.SECCD=parts.SECCD)
```

I used DBMS_XPLAN to get details on the execution of the query, which did show something interesting. The plan is basically to outer join PARTS and CATEGORIES, project that result set, then outer join it to SECTIONS. The interesting part is that in the projection of the first outer join, it is only including PTNO and SECCD -- it is NOT including the CATCD from either of the first two tables. Therefore the final result is getting CATCD from the third table.

But I don't know whether this is a cause or an effect.

Share

Improve this answer

Follow

edited Sep 16, 2008 at 12:45

answered Sep 15, 2008 at 15:55

Dave Costa

48.1k ● 8 ● 60 ● 73

1

I'm afraid I can't tell you why you're not getting an exception, but I can postulate as to why it chose CATEGORIES' version of the column over PARTS' version.

> As far as I understood, in the case of left joins, the "main" table in the query (PARTS) has precedence in ambiguity

It's not clear whether by "main" you mean simply the left table in a left join, or the "driving" table, as you see the query conceptually... But in either case, what you see as the "main" table in the query as you've written it will not necessarily be the "main" table in the actual execution of that query.

My guess is that Oracle is simply using the column from the first table it hits in executing the query. And since most individual operations in SQL do not require one table to be hit before the other, the DBMS will decide at parse time which is the most efficient one to scan first. Try getting an execution plan for the query. I suspect it may reveal that it's hitting CATEGORIES first and then PARTS.

Share Improve this answer Follow

answered Sep 12, 2008 at 18:44

Chris Ammerman
15.3k ● 8 ● 42 ● 42

1

I am using Oracle 9.2.0.8.0. and it does give the error "ORA-00918: column ambiguously defined".

Share Improve this answer Follow

answered Sep 12, 2008 at 13:17

user2752
31 ● 2

1

This is a known bug with some Oracle versions when using ANSI-style joins. The correct behavior would be to get an ORA-00918 error.

It's always best to specify your table names anyway; that way your queries don't break when you happen to add a new column with a name that is also used in another table.

Share Improve this answer Follow

answered Sep 18, 2008 at 15:55

Myto
5,291 ● 2 ● 17 ● 4

0

It is generally advised to be specific and fully qualify all column names anyway, as it saves the optimizer a little work. Certainly in SQL Server.

From what I can gleen from the Oracle docs, it seems it will only throw if you select the column name twice in the select list, or once in the select list and then again elsewhere like an order by clause.

Perhaps you have uncovered an 'undocumented feature' :)

Share Improve this answer Follow

answered Sep 12, 2008 at 13:00

hollystyles

5,029 ● 2 ● 37 ● 38

0

Like HollyStyles, I cannot find anything in the Oracle docs which can explain what you are seeing.

PostgreSQL, DB2, MySQL and MSSQL all refuse to run the first query, as it's ambiguous.

Share Improve this answer Follow

answered Sep 12, 2008 at 13:19

Troels Arvin

6,392 ● 2 ● 27 ● 28

0

@Pat: I get the same error here for your query. My query is just a little bit more complicated than what I originally posted. I'm working on a reproducible simple example now.

Share Improve this answer Follow

answered Sep 12, 2008 at 13:42

Ovesh

5,369 ● 11 ● 56 ● 77

0

A bigger question you should be asking yourself is - why do I have a category code in the parts table that doesn't exist in the categories table?

Share Improve this answer Follow

answered Sep 17, 2008 at 15:57

Mike McAllister

1,549 ● 2 ● 12 ● 15

0

This is a bug in Oracle 9i. If you join more than 2 tables using ANSI notation, it will not detect ambiguities in column names, and can return the wrong column if an alias isn't used.

As has been mentioned already, it is fixed in 10g, so if an alias isn't used, an error will be returned.

Share

edited Aug 11, 2011 at 21:22

answered Aug 11, 2011 at 18:23

Improve this answer

Follow

steve godfrey

1,234 ● 7 ● 15