

SQL: Get all posts with any comments

Asked 16 years, 1 month ago Modified 3 years, 6 months ago

Viewed 3k times



1

I need to construct some rather simple SQL, I suppose, but as it's a rare event that I work with DBs these days I can't figure out the details.



I have a table 'posts' with the following columns:



id, caption, text



and a table 'comments' with the following columns:

id, name, text, post_id

What would the (single) SQL statement look like which retrieves the captions of all posts which have one or more comments associated with it through the 'post_id' key? The DBMS is MySQL if it has any relevance for the SQL query.

[sql](#)[mysql](#)[Share](#)[Improve this question](#)[Follow](#)

asked Nov 13, 2008 at 22:51

[Morten Christiansen](#)

19.5k ● 23 ● 73 ● 96

7 Answers

Sorted by:

Highest score (default)



7



```
select p.caption, count(c.id)
from posts p join comments c on p.id = c.post_id
group by p.caption
having count (c.id) > 0
```

[Share](#) [Improve this answer](#)

edited Nov 13, 2008 at 23:23

[Follow](#)

answered Nov 13, 2008 at 22:56

[Nick DeVore](#)

10.2k ● 3 ● 40 ● 41

the HAVING and COUNT is unnessesacry since he want anything that has 1 OR more comments. and use INNER JOIN for clarity, even though inner join is the default. – [jishi](#) Nov 13, 2008 at 23:02

and GROUP BY p.id instead, since that will probably the primary key. – [jishi](#) Nov 13, 2008 at 23:03

It should be "... > 0" rather than "... > 1", other than that it's looking good :) – [Morten Christiansen](#) Nov 13, 2008 at 23:08



3



```
SELECT DISTINCT p.caption, p.id
  FROM posts p,
       comments c
 WHERE c.post_ID = p.ID
```

I think using a join would be a lot faster than using the IN clause or a subquery.

Share Improve this answer

edited Nov 13, 2008 at 23:16

Follow

answered Nov 13, 2008 at 22:57



[FlySwat](#)

175k ● 75 ● 248 ● 314

I'm not sure if DISTINCT would work here, or if I'd have to nest it into a subquery ie, SELECT DISTINCT Caption FROM (my query)...but I thought I'd throw it out there. – [FlySwat](#) Nov 13, 2008 at 23:11

I believe distinct would work here since caption and p.id would be equal for each post. – [jishi](#) Nov 13, 2008 at 23:13

We are probably right, but I don't have time to play with it on a DB. I'll remove my caveat anyways => – [FlySwat](#) Nov 13, 2008 at 23:16



0

```
SELECT DISTINCT caption
FROM posts
      INNER JOIN comments ON posts.id = comments.post_id
```



Forget about counts and subqueries.



The inner join will pick up all the comments that have valid posts and exclude all the posts that have 0 comments. The DISTINCT will coalesce the duplicate caption entries for posts that have more than 1 comment.

Share Improve this answer

answered Nov 13, 2008 at 23:43

Follow



James

3,892 ● 3 ● 21 ● 15

Does MySQL not have the inferred inner join syntax (look at my post below that is T-SQL) – [FlySwat](#) Nov 13, 2008 at 23:56



0

I find this syntax to be the most readable in this situation:

```
SELECT * FROM posts P
      WHERE EXISTS (SELECT * FROM Comments WHERE post_id =
```



It expresses your intent better than most of the others in this thread - "give me all the posts ..." (select * from posts) "... that have any comments" (where exist (select * from comments ...)). It's essentially the same as the joins above, but because you're not actually doing a join, you

don't have to worry about getting duplicates of the records in Posts, so you'll just get one record per post.

Share Improve this answer

answered Nov 14, 2008 at 2:06

Follow



Ian Varley

9,437 ● 5 ● 31 ● 34

You have also posted the slowest query yet =) – [FlySwat](#) Nov 14, 2008 at 2:25

Hmm ... I think that depends on the DB engine. The query isn't *inherently* slower, because it is relationally equivalent to a join. A good optimizer should run this just as fast as a join, or faster if it can avoid materializing the join results for the distinct. Course, I haven't tested it. :) – [Ian Varley](#) Nov 15, 2008 at 18:59



0



```
SELECT caption FROM posts
INNER JOIN comments ON comments.post_id = posts.id
GROUP BY posts.id;
```

No need for a `having` clause or `count()`.



edit: Should be a `inner join` of course (to avoid nulls if a comment is orphaned), thanks to jishi.



Share Improve this answer

edited Jun 11, 2021 at 15:25



Follow



akmozo

9,829 ● 3 ● 29 ● 44



answered Nov 13, 2008 at 23:07



Tjofras

2,096 • 12 • 13

that one would give him null posts for orphaned comments, which is a bad idea. a left join would give nulls for comments. An inner join is the way to go. – [jishi](#) Nov 13, 2008 at 23:11



Just going off the top of my head here but maybe something like:

-1



```
SELECT caption FROM posts WHERE id IN (SELECT post_id
count(*) > 0)
```



Share Improve this answer

answered Nov 13, 2008 at 22:54



Follow



lomaxx

116k • 58 • 147 • 180



You're basically looking at performing a subquery --

-2

```
SELECT p.caption FROM posts p WHERE (SELECT COUNT(*)
FROM comments c WHERE c.post_id=p.id) > 1;
```



This has the effect of running the `SELECT COUNT(*)` subquery for each row in the posts table. Depending on the size of your tables, you might consider adding an additional column, `comment_count`, into your `posts` table to store the number of corresponding `comments`, such that you can simply do



```
SELECT p.caption FROM posts p WHERE comment_count >
```

1

Share Improve this answer

answered Nov 13, 2008 at 22:53

Follow



[hark](#)

2,438 ● 1 ● 16 ● 7

Nooo! Denormalization should never be advocated on something this simple. – [FlySwat](#) Nov 13, 2008 at 22:59
