How to select posts with specific tags/categories in WordPress

Asked 16 years, 4 months ago   Modified 3 years, 4 months ago

Viewed 8k times   ⬖ Part of PHP Collective

10

This is a very specific question regarding **MySQL** as implemented in **WordPress**.
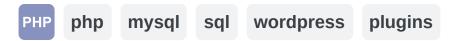
I'm trying to develop a plugin that will show (select) posts that have specific **'tags'** and belong to specific **'categories'** (both multiple)

I was told it's impossible because of the way categories and tags are stored:

1. `wp_posts` contains a list of posts, each post have an "ID"

2. `wp_terms` contains a list of terms (both categories and tags). Each term has a TERM_ID

3. `wp_term_taxonomy` has a list of terms with their TERM_IDs and has a Taxonomy definition for each one of those (either a Category or a Tag)

4. `wp_term_relationships` has associations between terms and posts

How can I join the tables to get all posts with tags "Nuclear" **and** "Deals" that also belong to the category "Category1"?
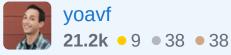
Share

Improve this question

Follow

6 Answers

Sorted by: Highest score (default) ⇕

▲

5

▼

I misunderstood you. I thought you wanted Nuclear or Deals. The below should give you only Nuclear and Deals.

```sql
select p.*
from wp_posts p, wp_terms t, wp_term_taxonomy tt, wp_t
wp_terms t2, wp_term_taxonomy tt2, wp_term_relationshi
wp_terms t2, wp_term_taxonomy tt2, wp_term_relationshi

where p.id = tr.object_id and t.term_id = tt.term_id a
tt.term_taxonomy_id

and p.id = tr2.object_id and t2.term_id = tt2.term_id
= tt2.term_taxonomy_id

and p.id = tr3.object_id and t3.term_id = tt3.term_id
= tt3.term_taxonomy_id

and (tt.taxonomy = 'category' and tt.term_id = t.term_
'Category1')
and (tt2.taxonomy = 'post_tag' and tt2.term_id = t2.te
```

```
  'Nuclear')
and (tt3.taxonomy = 'post_tag' and tt3.term_id = t3.te
  'Deals')
```

Try this:

3

```
select p.*
from wp_posts p,
wp_terms t, wp_term_taxonomy tt, wp_term_relationship
wp_terms t2, wp_term_taxonomy tt2, wp_term_relationshi

where p.id = tr.object_id
and t.term_id = tt.term_id
and tr.term_taxonomy_id = tt.term_taxonomy_id

and p.id = tr2.object_id
and t2.term_id = tt2.term_id
and tr2.term_taxonomy_id = tt2.term_taxonomy_id

and (tt.taxonomy = 'category' and tt.term_id = t.term_
  'Category1')
and (tt2.taxonomy = 'post_tag' and tt2.term_id = t2.te
  ('Nuclear', 'Deals'))
```

Essentially I'm employing 2 copies of the pertinent child tables - terms, term_taxonomy, and term_relationship. One copy applies the 'Category1' restriction, the other the 'Nuclear' or 'Deals' restriction.

BTW, what kind of project is this with posts all about nuclear deals? You trying to get us on some government list? ;)

Share Improve this answer

Follow

This is a fantastic solution – LΞИIИ Jun 25, 2019 at 6:17

▲

2

▼

🔖

🕘

What a gross DB structure.

Anyway, I'd do something like this (note I prefer EXISTS to joins, but you can re-write them as joins if you like; most query analyzers will collapse them to the same query plan anyway). You may have to do some additional juggling one way or another to make it work...

```
SELECT *
  FROM wp_posts p
 WHERE EXISTS( SELECT *
                 FROM wp_term_relationship tr
                WHERE tr.object_id = p.id
                  AND EXISTS( SELECT *
                                FROM wp_term_taxonomy
                               WHERE tt.term_taxonomy_
                                 AND tt.taxonomy
                                 AND EXISTS( SELECT *
                                               FROM wp
```

```
                                                    WHERE t.
                                                      AND t.
                                                 )
                              )
                    AND EXISTS( SELECT *
                                  FROM wp_term_taxonomy
                                 WHERE tt.term_taxonomy_
                                   AND tt.taxonomy
                                   AND EXISTS( SELECT *
                                                 FROM wp
                                                WHERE t.
                                                  AND t.
                                              )
                                   AND EXISTS( SELECT *
                                                 FROM wp
                                                WHERE t.
                                                  AND t.
                                              )
                              )
                    )
```

Share   Improve this answer

Follow

answered Aug 26, 2008 at 14:41

Matt Rogish
24.9k ● 11 ● 78 ● 93

1

So I tried both options on my WordPress db. I looked for the category "Tech" in my posts with the tags "Perl" AND "Programming".

Eric's worked once I added a missing comma in the initial select statement. It returned 3 records. The problem is that the section that is looking for the "post_tag" is actually working as an OR option. One of my posts only had one tag not both. Also it would be good to make the SELECT DISTINCT.

I tried [Matt's](#) version, but it kept returning an empty set. I may try to "juggle" with it.

Thanks @Eric it works! Just a few code corrections for future reference:

0

- the first select statements misses a coma after wp_term_relationship tr2

- In the same select statemt the following must be change:

```
wp_terms t2, wp_term_taxonomy tt2, wp_term_relationshi

tr2
```

should be

```
wp_terms t3, wp_term_taxonomy tt3, wp_term_relationshi

tr3
```

Really so great answer .. helped me a lot..

0

great bcoz., it gave me basic approach to build my complex query !

one small correction, for ready users like me :)

"wp_term_relationship" will give 'doesn't exist error' .. use **wp_term_relationships** as it is the correct table name.

Thanks Eric

Share   Improve this answer

Follow

answered Feb 5, 2009 at 15:26

Raghu