PostgreSQL Indexes

Last updated by | Lisa Liu | Nov 6, 2020 at 10:35 AM PST

PostgreSQL Indexes

Monday, January 13, 2020
4:34 PM

SQL query List all indexes on postgresql database

```
SELECT
   tablename,
   indexname,
   indexdef
FROM
   pg_indexes
ORDER BY
   tablename,
   indexname;
```

Indexes are most valuable across large tables as well. While accessing data from cache is faster than disk, even data within memory can be slow if Postgres must parse through hundreds of thousands of rows to identify if they meet a certain condition. To generate a list of your tables in your database with the largest ones first and the percentage of time which they use an index you can run:

```
SELECT
  relname,
  100 * idx_scan / (seq_scan + idx_scan) percent_of_times_index_used,
  n_live_tup rows_in_table
FROM
  pg_stat_user_tables
WHERE
    seq_scan + idx_scan > 0
ORDER BY
  n_live_tup DESC;
```

While there is no perfect answer, if you're not somewhere around 99% on any table over 10,000 rows you may want to consider adding an index. When examining where to add an index you should look at what kind of queries you're running. Generally you'll want to add indexes where you're looking up by some other id or on values that you're commonly filtering on such as created_at fields.

Pro tip: If you're adding an index on a production database use CREATE INDEX CONCURRENTLY to have it build your index in the background and not hold a lock on your table. The limitation to creating indexes concurrently is they can typically take 2-3 times longer to create and can't be run within a transaction. Though for any large production site these trade-offs are worth the trade-off in experience to your end users.

Created with Microsoft OneNote 2016.

How good have you found this content?

☺ ☹