Database Administrators

How to get the timestamp column in only milliseconds from PostgreSQL?

Asked 8 years, 5 months ago Active 1 year, 2 months ago Viewed 101k times

28

I have a column "created" with type `timestamp without time zone default now()` in a PostgreSQL database.

If I select colums, it has a nice and readable format per default:

```
SELECT created FROM mytable;

         created
-------------------------
2011-05-17 10:40:28.876944
```

But I would like to get the timestamp in only milliseconds (as a Long). Something like this:

SELECT myformat(created) FROM mytable;

```
2432432343876944
```

How can I get the timestamp column in only milliseconds from PostgreSQL?

Response to Jack:

I do get the same difference as you (-3600), but if I use `timestamp with time zone` I can see that the "error" or difference is because '1970-01-01' gets time zone `+01`.

```
create table my_table_2(created timestamp with time zone);
CREATE TABLE
insert into my_table_2 (created) values (now()), ('1970-01-01');
INSERT 0 2
select created, extract(epoch from created) from my_table_2;
            created            |    date_part
-------------------------------+------------------
 2011-05-18 11:03:16.909338+02 | 1305709396.90934
 1970-01-01 00:00:00+01        |            -3600
(2 rows)
```

Is the difference a bug? I may be because of "Daylight saving times" at the moment?

Also interesting while using `to_timestamp()` to insert timestamp 0 and 1.

```
insert into my_table_2 (created) values (to_timestamp(0));
INSERT 0 1

insert into my_table_2 (created) values (to_timestamp(1));
INSERT 0 1
select created, extract(epoch from created) from my_table_2;
            created            |    date_part
-------------------------------+------------------
 2011-05-18 11:03:16.909338+02 | 1305709396.90934
 1970-01-01 00:00:00+01        |            -3600
 1970-01-01 01:00:00+01        |                0
 1970-01-01 01:00:01+01        |                1
```

postgresql timestamp date-format

edited May 18 '11 at 9:15

asked May 17 '11 at 9:16

Jonas

12.2k 24 50 63

2 Answers

Use `EXTRACT` and the UNIX-Timestamp

34

```
SELECT EXTRACT(EPOCH FROM TIMESTAMP '2011-05-17 10:40:28.876944') * 1000;
```

would give

Multiply it by `1000` to turn it into milliseconds. You can then convert it to whatever you want (decimal would be a good choice). Don't forget to keep the timezone in mind. JackPDouglas has such an example in his answer. Here is an excerpt from his answer (`created` being the column with your timetamp) that illustrates how to work with timezones:

```
SELECT EXTRACT(EPOCH FROM created AT TIME ZONE 'UTC') FROM my_table;
```

edited Aug 9 '18 at 15:49

answered May 17 '11 at 10:06

DrColossos

4,439 2 26 31

5

--EDIT--

I've discovered this (see below) is basically wrong. See How do I get the current unix timestamp from PostgreSQL? for the source of my confusion...

--END EDIT--

Posting as an answer because it won't work as a comment.

testbed:

```
create role stack;
grant stack to dba;
create schema authorization stack;
set role stack;

create table my_table(created timestamp);
insert into my_table(created) values(now()),('1970-01-01');
\d my_table
             Table "stack.my_table"
 Column  |            Type             | Modifiers
---------+-----------------------------+-----------
 created | timestamp without time zone |
```

queries:

```
select created, extract(epoch from created) from my_table;

          created          |    date_part
---------------------------+------------------
 2011-05-17 13:18:48.03266 | 1305634728.03266
 1970-01-01 00:00:00       |            -3600


select created, extract(epoch from date_trunc('milliseconds', created))
from my_table;
```

```
select created, extract(epoch from created at time zone 'UTC') from my_table;

          created          |    date_part
---------------------------+-----------------
 2011-05-17 13:18:48.03266 | 1305638328.03266
 1970-01-01 00:00:00       |               0
```

note `date_part` in the third query is: 130563**83**28.03266 - 3600 different.

edited Apr 13 '17 at 12:42
Community ♦
1

answered May 17 '11 at 12:22

Jack Douglas ♦
30k 13 83 159