

Cross join Redshift with sequence of dates

Asked 1 year, 2 months ago Modified 1 year, 2 months ago Viewed 790 times  Part of [AWS Collective](#)

I'd like to do following athena query with redshift, but so far it's been impossible to reproduce it. The query should be called in a `CREATE TABLE AS ()` statement, so `generate_sequence()` ideas might not work. Any ideas?

Athena query:

```
SELECT
  *
FROM table_one t1
CROSS JOIN UNNEST(slice(sequence(t1.effective_date, t1.expiration_date,
INTERVAL '1' MONTH), 1 ,12)) AS t (sequence_date)
```

As requested I add an example to show what I'm trying to do. Basically I have a record with a validity interval (year units 1, 2, 3...) and I'd like to replicate it N time times such that each new record is assigned to date YYYY-MM-DD + interval*12/N months (See examples)

Original record:

Date	variables
2021-05-06	values

To be (N=12 and interval of 1 year)

Date	variables
2021-05-06	values/12
2021-06-06	values/12
2021-07-06	values/12
2021-08-06	values/12
2021-08-06	values/12
2021-10-06	values/12
2021-11-06	values/12
2021-12-06	values/12
2022-01-06	values/12
2022-02-06	values/12
2022-03-06	values/12

Date	variables
2022-04-06	values/12

To be (N=4 and interval of two years)

Date	variables
2021-05-06	values/2
2021-11-06	values/2
2022-05-06	values/2
2022-11-06	values/2


Thanks for the help

amazon-redshift amazon-athena

Share Edit Follow

edited Dec 14, 2021 at 18:24

asked Dec 9, 2021 at 13:21

 **Ignasi Mañé**
43 8

Sorted by:

Highest score (default)

1 Answer



1



Likely the best way to do this is with a recursive CTE -

https://docs.aws.amazon.com/redshift/latest/dg/r_WITH_clause.html

Example - [Generate rows with incrementing dates based on just a starting date in Redshift](#)

What you seem to be doing is a little more complex than this example. If you can't get it post some sample data that the experts here can use to create a sample query for you.

=====

With the new info and the above recursive CTE process I came up with this:

```
drop table if exists table_one ;
create table table_one (
dt date,
info varchar(32),
n int,
y int);

insert into table_one values ('2021-05-06', 'record info here', 12, 1);
commit;

with recursive dates(dt, info, n, y, end_dt) as
```

```
( select dt::date, info, n, y, dateadd(year, y, dt)::date as end_dt
  from table_one
 union all
 select dateadd(months, 12 * y / n, dt)::date as dt, info, n, y, end_dt::date
  from dates d
 where d.dt < dateadd(month, -12 * y /n, end_dt)
 )
select dt, info from dates;
```

I'm not sure this is how you want to get N and year into the process but hopefully you can modify from here. Just change the values of N and year in table_one insert statement and rerun the whole process to get your 2nd result.

Share Edit Follow

edited Dec 10, 2021 at 17:39

answered Dec 9, 2021 at 22:41



Bill Weiner

7,672 2 5 17

Thanks for your help I already explored this path, but not sure whether it is going to work. I added some clarifications in the post so that it is more clear what I'm trying to do. – [Ignasi Mañé](#) Dec 10, 2021 at 9:45

De nada. Just paying it forward. – [Bill Weiner](#) Dec 13, 2021 at 17:21
