Savepoint

Ideas not commited yet

DATA no PostgreSQL – Brincando com funções

2012/12/20 Banco de Dados, PostgreSQL PostgreSQL, Timbira

Após brincar de inserir e exibir datas e de fazer um pouco de aritimética com datas, chegou a hora de trabalhar com problemas mais complexos. Se você tiver alguma sugestão para um 4º ou até 5º post, deixe um comentário aqui.

Nesse ponto, você já deveria se sentir mais à vontade trabalhando com datas no PostgreSQL. Lembre-se sempre de consultar o capítulo 8 – "Data Types" e o capítulo 9 – "Functions and Operators" para conhecer um pouco mais sobre os tipos de dados disponíveis e as funções existentes para lidar com eles. Para hoje vamos fazer uma brincadeira simples com datas: vamos pegar o primeiro e o último dia útil do mês atual. Antes de mais nada, RTFM…. vá dar uma lida no manual para conhecer as funções do Postgres. Quanto mais eu releio a documentação, mais eu aprendo.

Mas vamos lá. Para começar, vamos pegar o Mês atual, com a função CURRENT_DATE que serve para pegar a data atual. Se você quiser utilizar as função semelhante, que retorna um TIMESTAMP ao invés de um DATE, você terá outras preocupações, como saber em que momento exato será considerado como "data atual". Por enquanto é tudo que precisamos saber.

PgSQL

```
1 # SELECT current_date;
2     date
3 ------------
4  2012-12-19
5 (1 row)
```

Bom, e se quisermos saber qual é o primeiro dia do mês atual? OK, podemos marretar que todo mês começa no dia primeiro, mas vamos trabalhar com SQL um pouco e conhecer outra função que serve para "arredondar" datas, o DATE_TRUNC:

PgSQL

```
1 # SELECT date_trunc('month',current_date);
2         date_trunc
3 ------------------------
4  2012-12-01 00:00:00-02
5 (1 row)
```

Simples não? Se fossemos arredondar um número, pensaríamos em qual casa decimal (dédimos, centésimos, milésimos) iríamos arredondar, como na função TRUNC. Com datas, podemos definir

arredondar no ano, mês, dia, hora, minuto, segundo, etc.

Mas vamos começar a complicar um pouco… e se nós quisermos pegar o último dia do mês? Bom, temos meses com 30, 31, 28 e 29 dias. O postgres já conhece bem o calendário, portanto, o que podemos fazer (sei que tem outra forma de fazer isso) é pegar o primeiro dia do mês seguinte e subtrair um dia. Vejamos alguns exemplos:

```PgSQL
# SELECT date_trunc('month',current_date) + INTERVAL'1 month' - INTERVAL'1
        ?column?
------------------------
 2012-12-31 00:00:00-02

# -- Vamos pegar o último dia de novembro
# SELECT date_trunc('month',DATE'2012-11-19') + INTERVAL'1 month' - INTERV
        ?column?
------------------------
 2012-11-30 00:00:00-02
(1 row)

# -- Vamos pegar o último dia de fevereiro num ano bisexto
# SELECT date_trunc('month',DATE'2012-02-16') + INTERVAL'1 month' - INTERV
        ?column?
------------------------
 2012-02-29 00:00:00-03
(1 row)

# -- Vamos pegar o último dia de fevereiro num ano não bisexto
# SELECT date_trunc('month',DATE'2011-02-16') + INTERVAL'1 month' - INTERV
        ?column?
------------------------
 2011-02-28 00:00:00-03
(1 row)
```

Ótimo, agora temos o primeiro e o último dia do mês atual, correto? Agora vamos gerar uma tabela com todos os dias do mês usando a função GENERATE_SERIES:

```PgSQL
=# SELECT *
-# FROM generate_series(DATE'2012-12-01',DATE'2012-12-31',INTERVAL'1 day')
    generate_series
------------------------
 2012-12-01 00:00:00-02
 2012-12-02 00:00:00-02
 2012-12-03 00:00:00-02
 2012-12-04 00:00:00-02
 2012-12-05 00:00:00-02
 2012-12-06 00:00:00-02
 2012-12-07 00:00:00-02
 2012-12-08 00:00:00-02
 2012-12-09 00:00:00-02
 2012-12-10 00:00:00-02
 2012-12-11 00:00:00-02
 2012-12-12 00:00:00-02
 2012-12-13 00:00:00-02
 2012-12-14 00:00:00-02
 2012-12-15 00:00:00-02
 2012-12-16 00:00:00-02
 2012-12-17 00:00:00-02
 2012-12-18 00:00:00-02
 2012-12-19 00:00:00-02
```

```
 2012-12-20 00:00:00-02
 2012-12-21 00:00:00-02
 2012-12-22 00:00:00-02
 2012-12-23 00:00:00-02
 2012-12-24 00:00:00-02
 2012-12-25 00:00:00-02
 2012-12-26 00:00:00-02
 2012-12-27 00:00:00-02
 2012-12-28 00:00:00-02
 2012-12-29 00:00:00-02
 2012-12-30 00:00:00-02
 2012-12-31 00:00:00-02
(31 rows)
```

Agora vamos fazer a mesma coisa, mas vamos substituir o primeiro e o último dia do mês pelas expressões que criamos antes:

PgSQL

```
=# SELECT * FROM generate_series(
-#    date_trunc('month',current_date),
-#    date_trunc('month',current_date) + INTERVAL'1 month' - INTERVAL'1 da
-#    INTERVAL'1 day');
    generate_series
------------------------
 2012-12-01 00:00:00-02
 2012-12-02 00:00:00-02
 2012-12-03 00:00:00-02
 2012-12-04 00:00:00-02
 2012-12-05 00:00:00-02
 2012-12-06 00:00:00-02
 2012-12-07 00:00:00-02
 2012-12-08 00:00:00-02
 2012-12-09 00:00:00-02
 2012-12-10 00:00:00-02
 2012-12-11 00:00:00-02
 2012-12-12 00:00:00-02
 2012-12-13 00:00:00-02
 2012-12-14 00:00:00-02
 2012-12-15 00:00:00-02
 2012-12-16 00:00:00-02
 2012-12-17 00:00:00-02
 2012-12-18 00:00:00-02
 2012-12-19 00:00:00-02
 2012-12-20 00:00:00-02
 2012-12-21 00:00:00-02
 2012-12-22 00:00:00-02
 2012-12-23 00:00:00-02
 2012-12-24 00:00:00-02
 2012-12-25 00:00:00-02
 2012-12-26 00:00:00-02
 2012-12-27 00:00:00-02
 2012-12-28 00:00:00-02
 2012-12-29 00:00:00-02
 2012-12-30 00:00:00-02
 2012-12-31 00:00:00-02
(31 rows)
```

Pronto… estamos quase lá agora. Vamos pegar apenas o primeiro e o último dia usando as funções de agregação MIN e MAX:

PgSQL

```
=# SELECT min(dias), max(dias)
```

```
-# FROM generate_series(date_trunc('month',current_date),date_trunc('month',c
          min           |          max
------------------------+------------------------
 2012-12-01 00:00:00-02 | 2012-12-31 00:00:00-02
(1 row)
```

Sim, parece que estamos dando voltas, não? Mas veja, para pegar o primeiro e o último dia do ano, só falta para nós filtrar o sábado e o domingo. E isso é fácil se utilizarmos a função EXTRACT (similar a função DATE_PART) que extrai apenas uma parte de uma data. No caso, queremos saber de qual dia da semana ela é:

PgSQL

```
=# SELECT
-#     EXTRACT(DAY FROM min(dias)) AS primeiro_dia_util,
-#     EXTRACT(DAY FROM max(dias)) AS ultimo_dia_util
-# FROM generate_series(
(#     date_trunc('month',current_date),
(#     date_trunc('month',current_date) + INTERVAL'1 month' - INTERVAL'1 day'
(#     INTERVAL'1 day') AS dias
-# WHERE EXTRACT(ISODOW FROM dias) < 6;
 primeiro_dia_util | ultimo_dia_util
-------------------+-----------------
                 3 |              31
(1 row)
```

Aqui utilizei o ISODOW que é o dia da semana, onde a segunda-feira é o dia 1 e o domingo é o dia 7, conforme o padrão ISO. Também aproveitei para extrair apenas o dia do mês na hora de exibir a informação, o que é absolutamente opcional. Você pode testar esta consulta para outros intervalos, substituindo o CURRENT_DATE por outras datas.

Como sugestão para uma aplicação real você pode criar uma tabela de feriados e filtrar as datas dos feriados também:

PgSQL

```
1  CREATE TABLE feriados (data date, descr text);
2  INSERT INTO feriados VALUES ('2012-12-25','Natal');
3  INSERT INTO feriados VALUES ('2013-01-01','Confraternização Universal');
4  INSERT INTO feriados VALUES ('2012-11-15','Proclamação da República');
5  INSERT INTO feriados VALUES ('2012-11-02','Finados');
6
7  SELECT
8      EXTRACT(DAY FROM min(dias)) AS primeiro_dia_util,
9      EXTRACT(DAY FROM max(dias)) AS ultimo_dia_util
10 FROM generate_series(
11     date_trunc('month',current_date),
12     date_trunc('month',current_date) + INTERVAL'1 month' - INTERVAL'1 day'
13     INTERVAL'1 day') AS dias
14 WHERE
15     EXTRACT(ISODOW FROM dias) < 6 AND
16     NOT EXISTS (SELECT data FROM feriados WHERE dias = data);
```

Você sofisticar mais o problema criando uma tabela que diferencie com feriados nacionais, estaduais e municipais. Mas isso fica para você pensar, se realmente tiver que lidar com este tipo de problema.

Mais

Curtir isso:

★ Curtir

Seja o primeiro a curtir este post.

Relacionado

Entrevista com Euler Taveira de Oliveira
Após a Entrevista com Josh Berkus, estou trazendo aqui uma entrevista com um desenvolvedor brasileiro de peso, o Sr. Euler Taveira que
Em "PostgreSQL"

Um pouco de aritmética com Data/Hora no PostgreSQL
No último post, comentei que para trabalhar corretamente no PostgreSQL, você tem de começar inserindo e exibindo
Em "PostgreSQL"

Trabalhando com logs no PostgreSQL
Em "PostgreSQL"

4 comentários sobre "DATA no PostgreSQL – Brincando com funções"

Eduardo Royer

2014/09/15 às 10:01

Porra, show de bola em cara. Estou inciando em funções plpgsql e cara, tu me ajudou bastante. Já adicionei aos favoritos 😉

Parabéns pelo site! Abraços.

Luis Romano

2015/02/23 às 16:43

Muito bom seu post, mas eu tenho uma dúvida. Como pegar todos os registros de uma tabela do mês passado sem informar a data?

2015/02/24 às 17:33

★ telles

Luis, tente algo como:

SELECT * FROM tabela_com_data
WHERE campo_data BETWEEN
date_trunc('month',current_date) – INTERVAL'1 month' AND
date_trunc('month',current_date)

Luis Romano

2015/02/24 às 17:47

Telles, muito obrigado por ter respondido, a alguns minutos essa minha dúvida também foi respondia através de um fórum, mas foi exatamente a mesma solução.

Muito obrigado novamente.