Práce s datumem a časem v PostgreSQL

Pavel Stěhule

P2D2 2019

Základní typy

- Date - dny, 4 bajty, od 4713BC .. 5874897 AD
- Time - čas, 8/12 bajtů, 00:00:00 .. 24:00:00
- Timestamp - den-čas, 8 bajtů, od 4713 BC .. 294276 AD
- Interval - interval, 16 bajtů, $\pm$178M let

typ Date

- integer n .. dnů od počátku (1.1.2000)
- jednoduše umožňuje přičítat/odčítat dny ±n

Ukázka	**Formát**
1999-01-08	ISO 8601 - 8 ledna 1999
~~01-08-1999~~	***MDY, DMY ??***
19990108	ISO formát
990108	ISO formát
1999.008	rok a Ntý den v roce
J2451187	pořadové číslo dne Juliánského kalendáře

typ Time

- čas - s přesností na ms
- pokud má časovou zónu, tak 12bajtů (nepoužívat)

Ukázka	Formát
04:05:06.200	ISO 8601
040506.200	ISO 8601

Ukázka

- Základní operace
- Jak se dostat k nulové hodnotě

typ Timestamp

- interně int8 - počet mikrosec od počátku 1.1.2000
- kombinace zápisů date a time
- **neukládá se časová zóna**
- vstup se transformují do UTC (pro TsTZ)
- výstup do časové zóny serveru (pro TsTZ)

Speciální konstanty

`Hodnota`	**Význam**
epoch	Unix počátek
infinity, -infinity	hodnota větší, menší než všechny ostatní
now, today, tomorrow, yesterday	

Ukázka

```
postgres=# SELECT 'tomorrow'::date, 'epoch'::timestamp;
+------------+---------------------+
|    date    |      timestamp      |
+------------+---------------------+
| 2019-02-01 | 1970-01-01 00:00:00 |
+------------+---------------------+
(1 row)
```

typ Interval

- 16 bajtová struktura - (měsíce, dny, vteřiny)
- nechtěné hodnoty (zprava) se mohou odfiltrovat
  - DAY TO MINUTE, HOUR TO MINUTE
- šílené kombinace zápisu - nejlépe specifikovat jednotky
  - Year, Months, Weeks, Days, Hours, Minutes, Seconds

Zápis intervalu - 1sec, 1hod

```
interval ‘1‘; -- 1 sec
interval ‘:1‘; -- 1 sec
interval ‘1sec‘; -- 1 sec

interval ‘1‘ hour; -- 1 hodina
‘1‘::interval hour; -- 1 hodina
‘1 hour‘:: interval; -- 1 hodina
‘1:‘::interval; -- 1 hodina
```

Pozor

```
postgres=# SELECT EXTRACT(days FROM '33days':: interval);
+-----------+
| date_part |
+-----------+
|        33 |
+-----------+
(1 row)

postgres=# SELECT EXTRACT(days FROM '1month':: interval);
+-----------+
| date_part |
+-----------+
|         0 |
+-----------+
(1 row)
```

Pozor

```
postgres=# SELECT extract(hour from '2days'::interval);
+-----------+
| date_part |
+-----------+
|         0 |
+-----------+
(1 row)

postgres=# SELECT extract(epoch from '2days'::interval)/60/60;
+----------+
| ?column? |
+----------+
|       48 |
+----------+
(1 row)
```

Otázky

- Má měsíc 30 dnů?
  - '15days' + '15days'
- Má den 24 hodin?
  - '12hours' + '12hours'

Odpovědi

- Přičtením měsíce bych měl dostat den se stejným pořadovým číslem v příštím měsíci, ale nikoliv přespříštím.
- Při operacích s intervalem se automaticky nepřevádí z hodin na dny, z dnů na měsíce - můžeme počítat s 8hodin na den, nebo s 24h.
- Jsou ale funkce, které předpokládají 30denní měsíc - justify_month a 24hodinový den - justify_hours

24hodin 30 dní

```
postgres=# SELECT justify_hours('1000hours'), justify_days('1000hours')
;
+-----------------+-------------+
|  justify_hours  | justify_days |
+-----------------+-------------+
| 41 days 16:00:00 | 1000:00:00  |
+-----------------+-------------+
(1 row)

postgres=# SELECT justify_days('41 days 16:00:00'), justify_interval('1000hours')
;
+----------------------+----------------------+
|     justify_days     |   justify_interval   |
+----------------------+----------------------+
| 1 mon 11 days 16:00:00 | 1 mon 11 days 16:00:00 |
+----------------------+----------------------+
(1 row)
```

Jak vygenerovat hodnotu

- přetypování ::
- parsovací funkce - to_date, to_timestamp
- konstruktory - make_date, make_interval, make_timestamp

Jak vygenerovat hodnotu

```
SELECT '20190214'::date;
SELECT date '20190214';
SELECT to_date('20190214', 'YYYYMMDD');
SELECT make_date(2019,02,14);

postgres=# SHOW datestyle ;
+-----------+
| DateStyle |
+-----------+
| ISO, DMY  |
+-----------+
(1 row)
```

Užitečné funkce

- age(timestamp [,timestamp]) .. rozdíl výstup(R,M,D)
- date_part(text, timestamp) | EXTRACT
  - day, dow, isodow, doy, epoch, month, year
- date_trunc(text, timestamp)
- (s,e) OVERLAP (s,e)

Změna času (letní/zimní čas)

- 31.3 ve 2:00 se změní hodiny na 3:00 .. noc bude o 1h kratší, ráno déle tma, večer déle světlo
- 27.10 ve 3:00 se hodiny posunou na 2:00 noc bude o 1h delší, ráno dříve světlo, večer dříve tma.
- Kdyby pozorovatel zapisoval čas, tak 31.3 chybí např. 2:30, a naopak 27.10, bude 2:30 2x

Časové zóny

- Čas je pořád stejný, a pouze jeden
- Naše vnímání času souvisí se střídáním dnu a noci, což souvisí s rotací planety - čas poledne, svítání, soumraku je se mění napříč poledníky (sluneční čas).
- Není ale praktické mít v Praze a Brně rozdílné časové systémy - zavedla se časová pásma (pásmový čas)

Časové zóny

- -8 .. Los Angeles
- -5 .. New York
- 0 .. UTC - Londýn, Lisabon
- +1 .. Evropa - Madrid, Varšava (standardní čas)
  - +2 Letní evropský čas (sezonní čas)
- +3 .. Moskva, Istanbul
-

Značení časových zón

- úplný název - Europe/Prague,
- zkratka - CET
- **posix offset** - pozor kladná osa je na západ od Londýna (pozor - matoucí - používá se při samostatném značení časové zóny.
- Interval - '2h'::interval
- při zobrazení se používá offset, jehož kladná osa je na východ od Londýna (i při vstupu při zápisu času).

Pozor na posix offset časové zóny

```
-- aktualni cas v Praze
SELECT CURRENT_TIMESTAMP AT TIME ZONE ‘CET‘; -- 2019-01-31 22:04:42.920639

-- aktualni cas v Praze ??
SELECT CURRENT_TIMESTAMP AT TIME ZONE ‘+01‘; -- 2019-01-31 20:04:42.920639

-- Pozor!! +01 .. je Posix offset .. Azory

SELECT CURRENT_TIMESTAMP AT TIME ZONE interval ‘1‘ hour;
```

Timestamp, Timestamptz

- V obou případech počet microsec od počátku
- V obou případech se neukládá časová zóna
- U timestamptz - na vstupu převod do UTC (pokud není časová zóna zadána, použije se zóna klienta - timestamptz jsou uložené v UTC
- U timestamptz - na výstupu je transformace z UTC do zóny klienta.
- Zóna klienta je nastavená v proměnné „timezone“

Operátor AT TIME ZONE

- Ttz **AT TIME ZONE** TZ => Timestamp
  - Co vidí v okamžik Ttz uživatel v časové zóně TZ
- T **AT TIME ZONE TZ** => TimestampTZ
  - Co vidíme my (client timezone) v okamžik T v časové zóně TZ (Timestamp with time zone se zobrazuje v zóně klienta).
- POZOR: u timetz jiné chování (ukládá se časová zóna)

Otázka - co bude výsledkem?

'2019-0-05 12:00:00' AT TIME ZONE 'Europe/Moscow'

A) timestamp - 2019-05-05 13:00:00

B) timestamptz - 2019-05-05 11:00:00+02

V letním čase je
Moskva +3 a Praha +2

správně je A)

- Postgres netypové konstanty přetypovává do typů určených kontextem - nikoliv způsobem zápisu.
- AT TIME ZONE na levé straně s nejvyšší prioritou očekává timestamptz
- timezone se do typu timestamp doplní lokální

Pozor na typy a implicitní doplnění TZ

```
postgres=# SELECT typname, typispreferred, typcategory
             FROM pg_type
            WHERE typname like 'timestamp%';
+-------------+----------------+-------------+
|   typname   | typispreferred | typcategory |
+-------------+----------------+-------------+
| timestamp   | f              | D           |
| timestamptz | t              | D           |
+-------------+----------------+-------------+
(2 rows)
```

Pozor na typy a implicitní doplnění TZ

```
-- i když neuvedeme TZ jedná se o TimestampTZ. Použije se aktuální TZ
SELECT '2019-02-07 12:00:00' AT TIME ZONE 'Europe/Moscow';

-- na otázku, jaký bude lokální čas v 12:00 moskevského času
SELECT '2019-02-07 12:00:00'::timestamp AT TIME ZONE 'Europe/Moscow';

V postgresu je typ řetězce určený kontextem - pro AT TIME ZONE
výchozí typ je timestamp with time zone
```

Plánování schůzky ve 5.5. 12:00 lokálního času

```
SELECT '2019-05-05 12:00:00'::timestamptz AT TIME ZONE 'Europe/Moscow';
-- 2019-05-05 13:00:00 Čas v Moskvě

SELECT '2019-05-05 12:00:00'::timestamptz AT TIME ZONE 'America/New_York';
-- 2019-05-05 06:00:00 Čas v NY
```

Plánování schůzky ve 5.5. 12:00 NY

```
SELECT '2019-05-05 12:00:00'::timestamp AT TIME ZONE 'America/New_York';
-- 2019-05-05 18:00:00 Čas v Praze

SELECT '2019-05-05 12:00:00'::timestamptz AT TIME ZONE 'America/New_York'
       AT TIME ZONE 'Europe/Moscow';
-- 2019-05-05 19:00:00 Čas v Moskvě
```

Otázky ?