



BD201c
v200b
2018-09-25

BASES DE DONNÉES *TEMPORALITÉ*

Éléments de programmation temporelle avec ISO:2011 et PostgreSQL

Christina KHNAISSER et Luc LAVOIE
Département d'informatique
Faculté des sciences




UNIVERSITÉ DE
SHERBROOKE

Christina.Khnaisser@USherbrooke.ca
<http://info.USherbrooke.ca/ckhnaisser>
 Luc.Lavoie@USherbrooke.ca
<http://info.USherbrooke.ca/llavoie>

1

PLAN

- La temporalité et SQL
 - SQL:2011 et PostgreSQL
 - Types temporels
 - Opérateurs temporels
 - Exemples de requêtes



2018-09-25 BD201c : Introduction à la temporalité (v200b) — Christina Khnaisser et Luc Lavoie
Département d'informatique, Faculté des sciences, Université de Sherbrooke, Québec

2

2

TYPES TEMPORELS

- Types points temporels
- Type intervalle temporel

TYPES TEMPORELS

TYPES POINTS TEMPORELS

- Pour ISO 2011 et PostgreSQL
 - TIMESTAMP (WITH TIME ZONE)
 - ex. 2018-09-25 17:31:28.849346+00
 - TIME (WITH TIME ZONE)
 - ex. 17:31:28.849346+00
 - DATE
 - ex. 2018-09-25

TYPES TEMPORELS

CONSTRUCTEUR D'INTERVALLE TEMPOREL (SQL 2011)

- SQL:2011 (Teradata, DB2)
 - PERIOD notation fermé-ouvert
 - Impossible de créer des intervalle unitaire

```
CREATE TABLE Emp (
  ENo INTEGER,
  EStart DATE,
  EEnd DATE,
  EDept INTEGER,
  PERIOD FOR EPeriod (EStart, EEnd))

SELECT Ename, Edept
FROM Emp
WHERE EPeriod OVERLAPS
  PERIOD (DATE '2010-01-01', DATE '2011-01-01')
```

KULKARNI, K. AND MICHELS, J.-E. 2012. Temporal Features in SQL:2011.

TYPES TEMPORELS

TYPES INTERVALLE TEMPORELS (POSTGRESQL)

- Pour PostgreSQL (9.3+)
 - Constructeur d'intervalle : RANGE
 - Types d'intervalles prédéfinis :
 - int4range : intervalle d'entiers
 - daterange : intervalle de dates
 - tstrange : intervalle d'estampilles (timestamps)
 - Notation fermée-ouvert

<https://www.postgresql.org/docs/10/static/rangetypes.html>

OPÉRATEURS TEMPORELS

- SQL:2011
- PostgreSQL 9.3+

OPÉRATEURS TEMPORELS SQL:2011

○ OVERLAPS

(L_{start}, L_{end}) OVERLAPS (R_{start}, R_{end})

evaluates to true iff

$(L_{start} > R_{start} \text{ AND } (L_{start} < R_{end} \text{ OR } L_{end} < R_{end}))$

OR

$(R_{start} > L_{start} \text{ AND } (R_{start} < L_{end} \text{ OR } R_{end} < L_{end}))$

OR

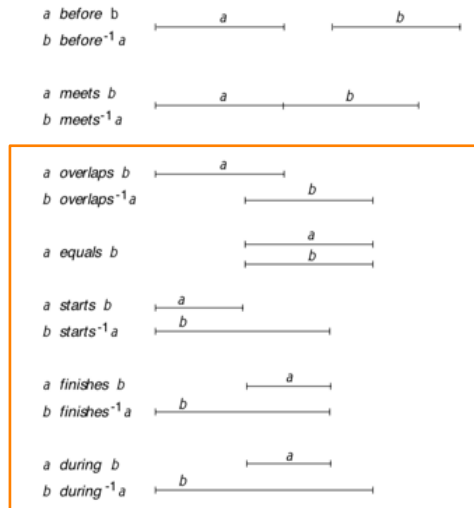
$(L_{start} = R_{start} \text{ AND } L_{end} \text{ IS NOT NULL AND } R_{end} \text{ IS NOT NULL})$

https://pages.iai.uni-bonn.de/manthey_rainer/TIS2018/

○ ATTENTION :

- un intervalle en SQL est interprété selon la notation fermé-ouvert
- OVERLAPS SQL \neq OVERLAPS ALLEN
- le «a overlaps b» de SQL est équivalent à « $a \cap b \neq \emptyset$ » en notation algébrique.

OPÉRATEURS TEMPORELS SQL:2011



SNODGRASS, R.T. 2000. *Developing time-oriented database applications in SQL*.

OPÉRATEURS TEMPORELS POSTGRESQL 9.3+

Function	Return Type	Description	Example	Result
<code>lower(anyrange)</code>	range's element type	lower bound of range	<code>lower(numrange(1.1,2.2))</code>	1.1
<code>upper(anyrange)</code>	range's element type	upper bound of range	<code>upper(numrange(1.1,2.2))</code>	2.2
<code>isempty(anyrange)</code>	boolean	is the range empty?	<code>isempty(numrange(1.1,2.2))</code>	false
<code>lower_inc(anyrange)</code>	boolean	is the lower bound inclusive?	<code>lower_inc(numrange(1.1,2.2))</code>	true
<code>upper_inc(anyrange)</code>	boolean	is the upper bound inclusive?	<code>upper_inc(numrange(1.1,2.2))</code>	false
<code>lower_inf(anyrange)</code>	boolean	is the lower bound infinite?	<code>lower_inf('(',')':daterange)</code>	true
<code>upper_inf(anyrange)</code>	boolean	is the upper bound infinite?	<code>upper_inf('(',')':daterange)</code>	true
<code>range_merge(anyrange, anyrange)</code>	anyrange	the smallest range which includes both of the given ranges	<code>range_merge('[1,2)':int4range, '[3,4)':int4range)</code>	[1,4)

<https://www.postgresql.org/docs/10/static/functions-range.html>
<https://www.postgresql.org/docs/9.3/static/btree-gist.html>

OPÉRATEURS TEMPORELS POSTGRESQL 9.3+

Operator	Description	Example	Result
=	equal	<code>int4range(1,5) = '[1,4]':int4range</code>	t
<>	not equal	<code>numrange(1.1,2.2) <> numrange(1.1,2.3)</code>	t
<	less than	<code>int4range(1,10) < int4range(2,3)</code>	t
>	greater than	<code>int4range(1,10) > int4range(1,5)</code>	t
<=	less than or equal	<code>numrange(1.1,2.2) <= numrange(1.1,2.2)</code>	t
>=	greater than or equal	<code>numrange(1.1,2.2) >= numrange(1.1,2.0)</code>	t
@>	contains range	<code>int4range(2,4) @> int4range(2,3)</code>	t
@>	contains element	<code>'[2011-01-01,2011-03-01)':tstrange @> '2011-01-10':timestamp</code>	t
<@	range is contained by	<code>int4range(2,4) <@ int4range(1,7)</code>	t
<@	element is contained by	<code>42 <@ int4range(1,7)</code>	f
&&	overlap (have points in common)	<code>int8range(3,7) && int8range(4,12)</code>	t
<<	strictly left of	<code>int8range(1,10) << int8range(100,110)</code>	t
>>	strictly right of	<code>int8range(50,60) >> int8range(20,30)</code>	t
&<	does not extend to the right of	<code>int8range(1,20) &< int8range(18,20)</code>	t
&>	does not extend to the left of	<code>int8range(7,20) &> int8range(5,10)</code>	t
- -	is adjacent to	<code>numrange(1.1,2.2) - - numrange(2.2,3.3)</code>	t
+	union	<code>numrange(5,15) + numrange(10,20)</code>	[5,20]
*	intersection	<code>int8range(5,15) * int8range(10,20)</code>	[10,15]
-	difference	<code>int8range(5,15) - int8range(10,20)</code>	[5,10]

<https://www.postgresql.org/docs/10/static/functions-range.html>
<https://www.postgresql.org/docs/9.3/static/btree-gist.html>

11

11

EXEMPLES REQUÊTES

- Quelles sont les activités auxquelles l'étudiant « 15113150 » est inscrit ?
- Quelles sont les activités auxquelles l'étudiant « 15113150 » est inscrit le 2013-09-12 ?
- ...

12

12

EXEMPLES REQUÊTES

- Quelles sont les activités auxquelles l'étudiant « 15113150 » est inscrit ?

matricule	activité	trimestre	debut	fin
15113150	IFT187	20133	2013-08-10	9999-12-31
15112354	IFT187	20123	2012-08-10	9999-12-31
15113150	IFT159	20133	2013-08-10	2013-09-10
15112354	GMQ103	20123	2012-08-10	9999-12-31
15110132	IMN117	20123	2012-08-10	9999-12-31
15110132	IFT187	20133	2013-08-10	9999-12-31
15112354	IFT159	20123	2012-08-10	9999-12-31

```
SELECT activite
FROM Resultat
WHERE matricule = '15113150'
      AND (CURRENT_DATE, CURRENT_DATE + INTERVAL '1' DAY)
      OVERLAPS (debut, fin);
```

- ATTENTION la requête est valide ssi :
 - « Fin » n'est pas nul

EXEMPLES REQUÊTES

- Quelles sont les activités auxquelles l'étudiant « 15113150 » est inscrit le 2013-09-12 ?

matricule	activité	trimestre	debut	fin
15113150	IFT187	20133	2013-08-10	9999-12-31
15112354	IFT187	20123	2012-08-10	9999-12-31
15113150	IFT159	20133	2013-08-10	2013-09-10
15112354	GMQ103	20123	2012-08-10	9999-12-31
15110132	IMN117	20123	2012-08-10	9999-12-31
15110132	IFT187	20133	2013-08-10	9999-12-31
15112354	IFT159	20123	2012-08-10	9999-12-31

```
SELECT activite
FROM Resultat
WHERE matricule = '15113150'
      AND (DATE '2013-09-12', DATE '2013-09-13')
      OVERLAPS (debut, fin);
```