Bases de données
Temporalité

Éléments de programmation temporelle avec ISO:2011 et PostgreSQL

BD201c
v200b

2018-09-25

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

PLAN

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

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
    - tsrange : intervalle d'estampilles (timestamps)
  - Notation fermée-ouvert

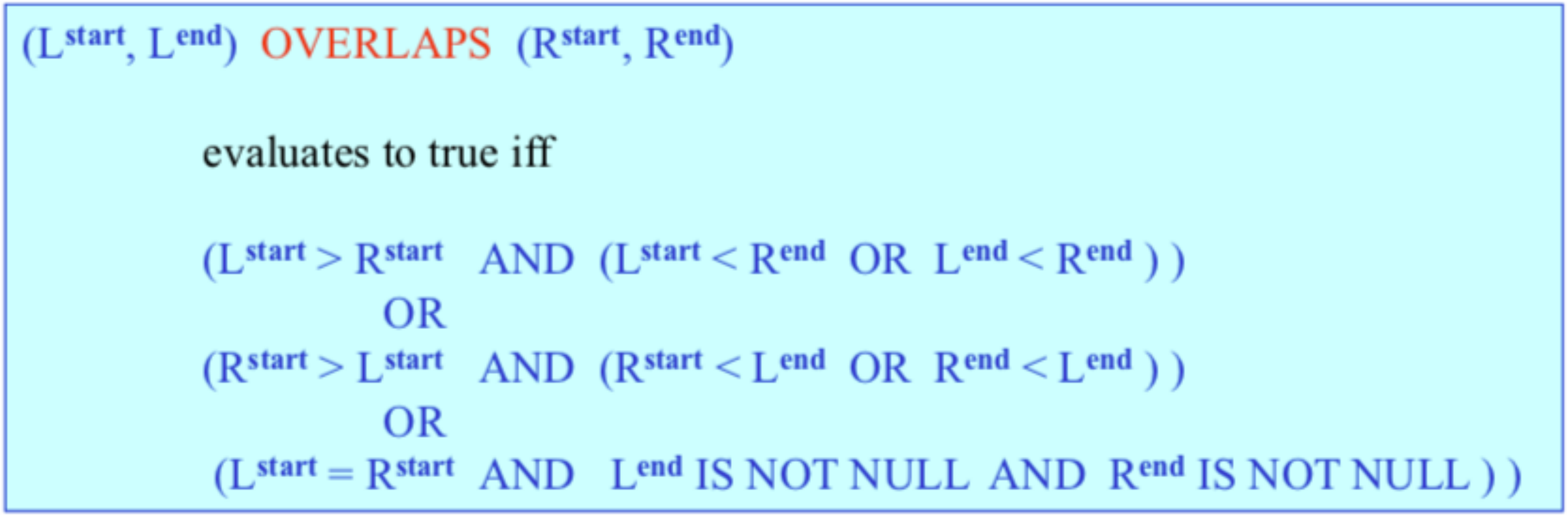
https://www.postgresql.org/docs/10/static/rangetypes.html

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

$(L^{start} > R^{start}$ AND $(L^{start} < R^{end}$ OR $L^{end} < R^{end}))$

OR

$(R^{start} > L^{start}$ AND $(R^{start} < L^{end}$ OR $R^{end} < L^{end}))$

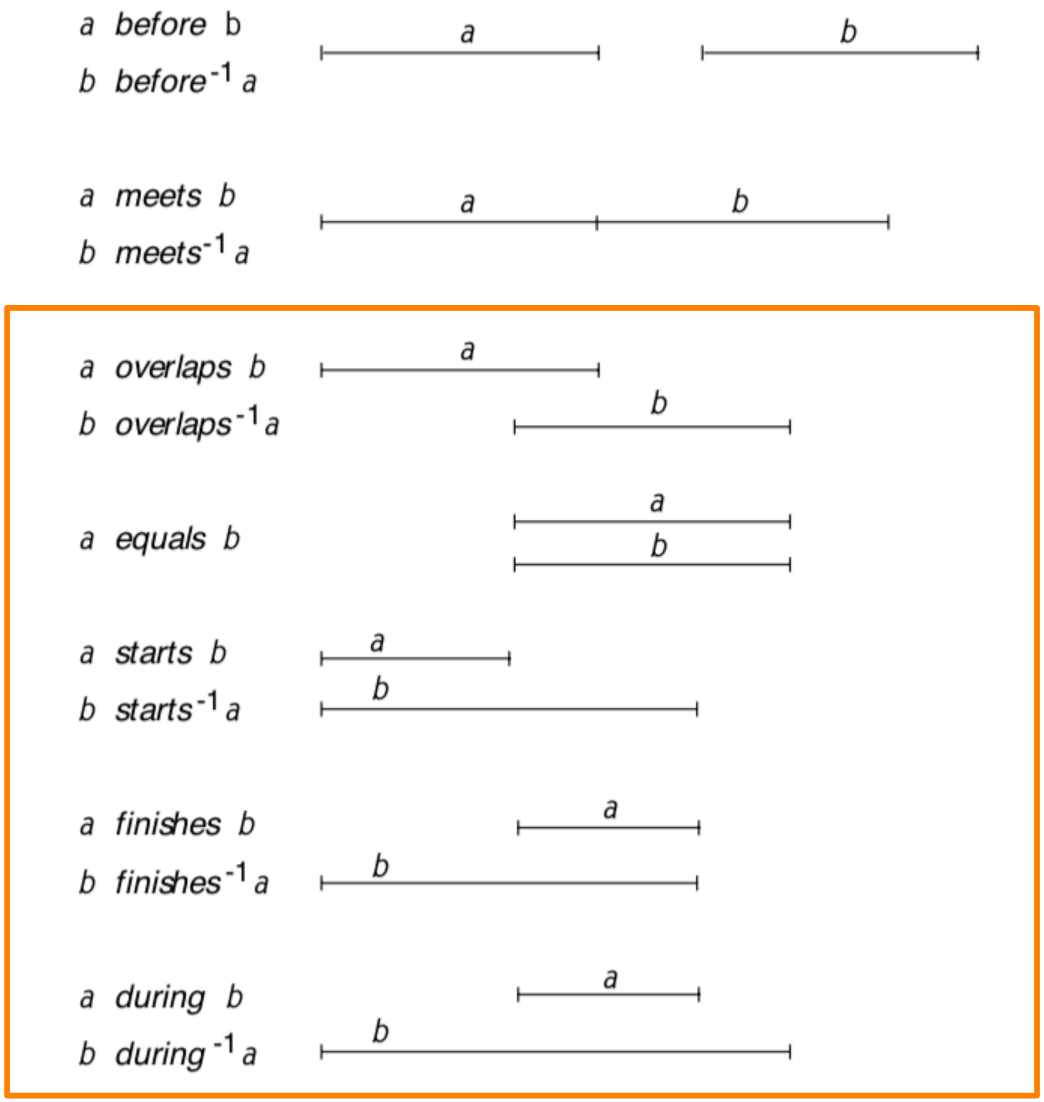
OR

$(L^{start} = R^{start}$ AND L^{end} IS NOT NULL AND R^{end} IS NOT NULL))

- ATTENTION :
  - un intervalle en SQL est interprété selon la notation fermé-ouvert
  - OVERLAP SQL ≠ OVERLAPS ALLEN

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

Opérateurs temporels SQL:2011



a before b

b before^{-1} a

a meets b

b meets^{-1} a

a overlaps b

b overlaps^{-1} a

a equals b

a starts b

b starts^{-1} a

a finishes b

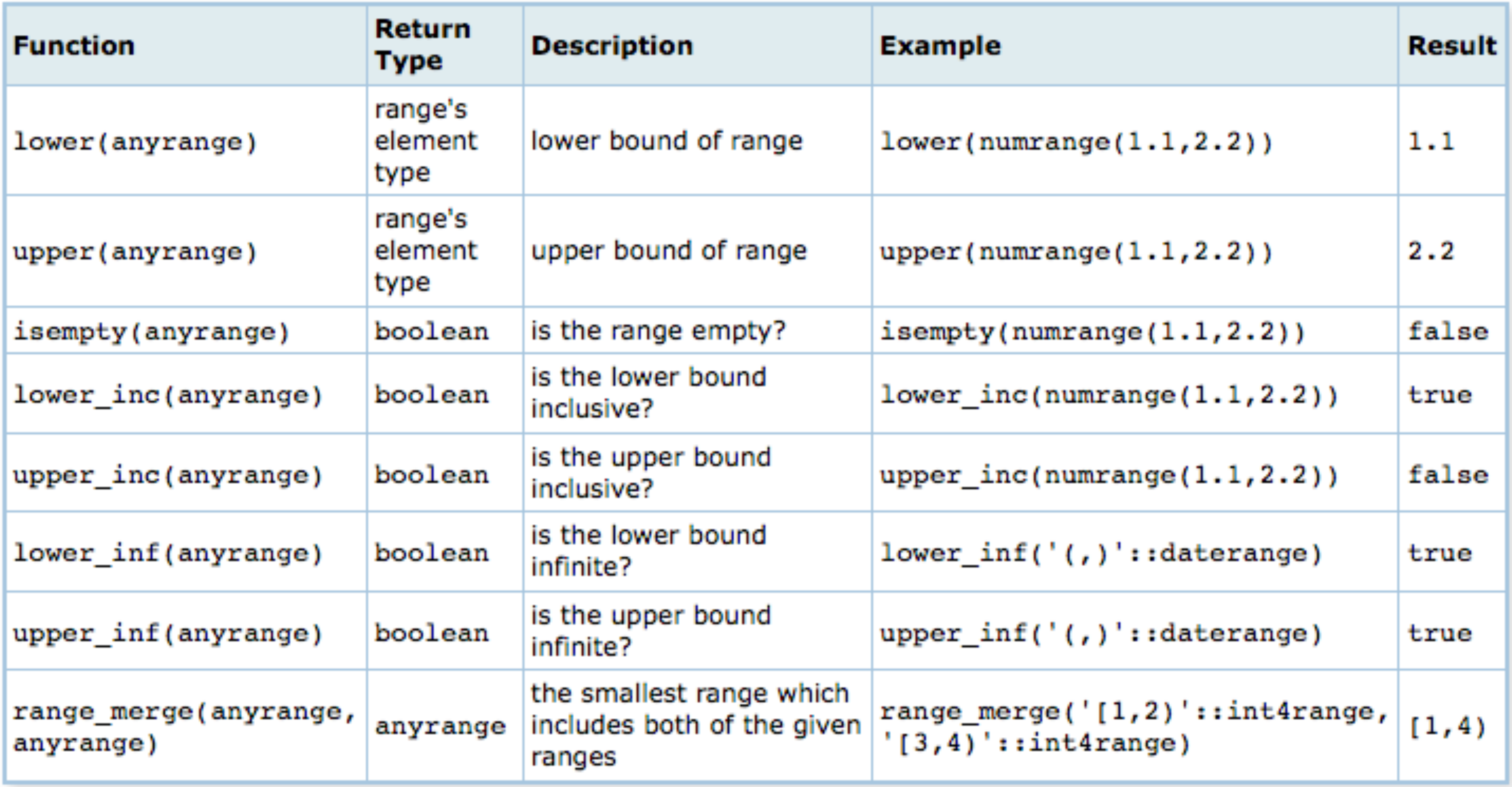
b finishes^{-1} a

a during b

b during^{-1} a

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

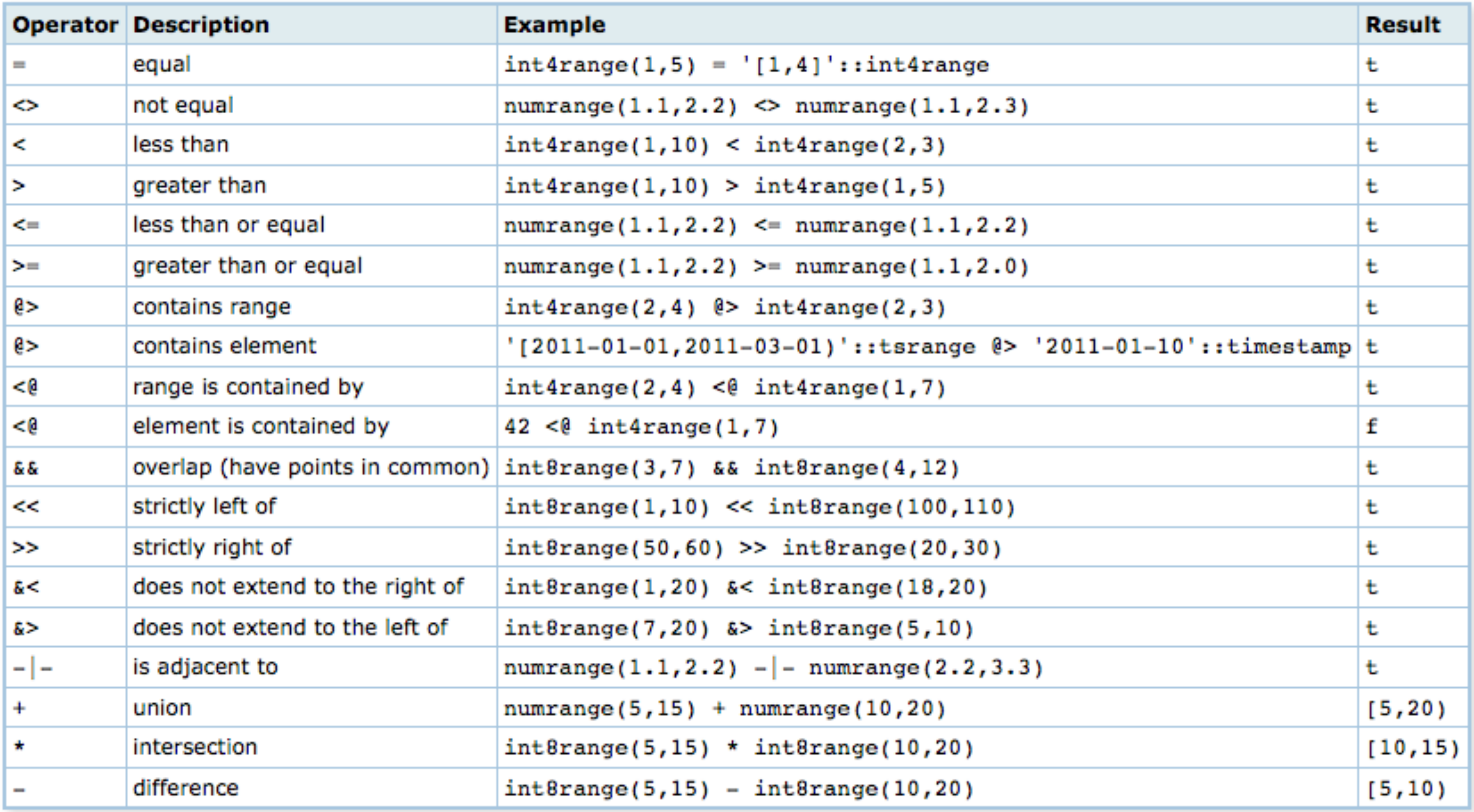
Opérateurs temporels PostgreSQL 9.3+



Function	Return Type	Description	Example	Result
`lower(anyrange)`	range's element type	lower bound of range	`lower(numrange(1.1,2.2))`	`1.1`
`upper(anyrange)`	range's element type	upper bound of range	`upper(numrange(1.1,2.2))`	`2.2`
`isempty(anyrange)`	`boolean`	is the range empty?	`isempty(numrange(1.1,2.2))`	`false`
`lower_inc(anyrange)`	`boolean`	is the lower bound inclusive?	`lower_inc(numrange(1.1,2.2))`	`true`
`upper_inc(anyrange)`	`boolean`	is the upper bound inclusive?	`upper_inc(numrange(1.1,2.2))`	`false`
`lower_inf(anyrange)`	`boolean`	is the lower bound infinite?	`lower_inf('(,)'::daterange)`	`true`
`upper_inf(anyrange)`	`boolean`	is the upper bound infinite?	`upper_inf('(,)'::daterange)`	`true`
`range_merge(anyrange, anyrange)`	`anyrange`	the smallest range which includes both of the given ranges	`range_merge('[1,2)'::int4range, '[3,4)'::int4range)`	`[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	`int4range(1,5) = '[1,4]'::int4range`	t
<>	not equal	`numrange(1.1,2.2) <> numrange(1.1,2.3)`	t
<	less than	`int4range(1,10) < int4range(2,3)`	t
>	greater than	`int4range(1,10) > int4range(1,5)`	t
<=	less than or equal	`numrange(1.1,2.2) <= numrange(1.1,2.2)`	t
>=	greater than or equal	`numrange(1.1,2.2) >= numrange(1.1,2.0)`	t
@>	contains range	`int4range(2,4) @> int4range(2,3)`	t
@>	contains element	`'[2011-01-01,2011-03-01)'::tsrange @> '2011-01-10'::timestamp`	t
<@	range is contained by	`int4range(2,4) <@ int4range(1,7)`	t
<@	element is contained by	`42 <@ int4range(1,7)`	f
&&	overlap (have points in common)	`int8range(3,7) && int8range(4,12)`	t
<<	strictly left of	`int8range(1,10) << int8range(100,110)`	t
>>	strictly right of	`int8range(50,60) >> int8range(20,30)`	t
&<	does not extend to the right of	`int8range(1,20) &< int8range(18,20)`	t
&>	does not extend to the left of	`int8range(7,20) &> int8range(5,10)`	t
-|-	is adjacent to	`numrange(1.1,2.2) -|- numrange(2.2,3.3)`	t
+	union	`numrange(5,15) + numrange(10,20)`	[5,20)
*	intersection	`int8range(5,15) * int8range(10,20)`	[10,15)
-	difference	`int8range(5,15) - int8range(10,20)`	[5,10)

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

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 ?
- …

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);
```