



EDBTM

SQL Avancé I

Lætitia Avrot



Lætitia Avrot



- Field CTO – EDB
- Trésorière de PostgreSQL Europe
- Fondatrice de Postgres women
- Contributrice reconnue du projet PostgreSQL

Au programme

- Qu'est-ce que le SQL?

Qu'est-ce que le SQL?

Un langage différent

Facultatif, donc
indispensable!

- Pas objet
- Pas fonctionnel
- Pas procédural
- Déclaratif
- ... Mais reste un langage!

- Avec indentation
- Avec retours à la ligne
- Avec commentaires



Turing-complet

(depuis SQL:1999)

- Capable de récursivité
- [Preuve formelle avec une CTE récursive](#)
- [Preuve formelle avec une fonction récursive](#)
- Capable de résoudre tout problème informatique
- Résolution de [adventofcode](#)

Historique

- SQL-86
- SQL-89
 - Ajout des contraintes d'intégrité
- SQL-92
 - Celui que vous connaissez
- SQL-1999
 - Ajout des expressions régulières, des tableaux, de la récursivité, des triggers et des procédures
- SQL-2003
 - Support du XML, des fonctions de fenêtrage, des séquences et des colonnes générées
- SQL-2006
 - Support des XQuery

Historique

- **SQL-2008**
 - Ajout de l'ordre `truncate`, de la clause `fetch` et des triggers `instead of`
- **SQL-2011**
 - Ajout des tables temporales, amélioration de la clause `fetch` et des fonctions de fenêtrage
- **SQL-2016**
 - Ajout du JSON
- **SQL-2019**
 - Ajout des tableaux multidimensionnels
- **SQL-2023**
 - Amélioration de la gestion des JSON, ajout du SQL/PGQ (Property Graph Queries)

Quizz!





Comment puis-je
utiliser `values`?

- Tout seul
- Dans un insert
- Dans une clause from
- C'est quoi `values`?

```
values (1,2);
```

```
select *  
from (values (1,2));
```

```
insert into test  
values (1,2);
```

column1	column2
1	2

(1 row)

```
INSERT 0 1
```

Values est un constructeur de tables.



Dans quelle clause
peut-on mettre
une sous-requête?

- Dans un select
- Dans un from
- Dans un order by
- Dans un where

```
select (select (values(1)));
```

```
select * from (  
  select (values(1)));
```

```
select * from (values(1))  
order by (select (values(1)));
```

```
select * from (values(1))  
where (select (values(1))=1);
```

Uniquement pour
les requêtes
ramenant une seule
ligne et une seule
colonne.

column1

1

(1 row)

Uniquement pour les requêtes
ramenant un booléen.



Combien de lignes retournera la jointure entre les tables A et B ?

Table A
null
0
1
2

Table B
2
3
4
null

```
select *
from
  (values (null),(0),(1),(2))
  as a(t)
inner join
  (values (2),(3),(4),(null))
  as b(t)
on a.t=b.t;
```

t		t
2		2
(1 row)		

Inner join joint les tables lorsque les valeurs sont égales



Combien de lignes retournera la jointure à gauche entre les tables A et B ?

Table A
null
0
1
2

Table B
2
3
4
null


```

select *
from
  (values (null), (0), (1), (2))
  as a(t)
left join
  (values (2), (3), (4), (null))
  as b(t)
on a.t=b.t;

```

t	t
0	0
1	1
2	2

(4 rows)

Left join joint les tables lorsque les valeurs sont égales et ajoute toutes les valeurs de la table de gauche



Combien de lignes retournera la jointure full entre les tables A et B ?

Table A
null
0
1
2

Table B
2
3
4
null

```

select *
from
  (values (null), (0), (1), (2))
  as a(t)
Full outer join
  (values (2), (3), (4), (null))
  as b(t)
on a.t=b.t;

```

t	t
0	0
1	1
2	2
2	2
3	4
4	3
4	3

(7 rows)

Full outer join joint les tables lorsque les valeurs sont égales et ajoute toutes les valeurs de la table de gauche et de la table de droite.



Combien de lignes retournera le produit cartésien entre les tables A et B ?

Table A
null
0
1
2

Table B
2
3
4
null

```

select *
from
  (values (null), (0), (1), (2))
  as a(t)
cross join
  (values (2), (3), (4), (null))
  as b(t);

```

t	t
👩	2
👩	3
👩	4
👩	👩
0	2
0	3
0	4
0	👩
1	2
1	3
1	4
1	👩
2	2
2	3
2	4
2	👩

(16 rows)

Un produit cartésien va combiner toutes les valeurs de la table A avec toutes les valeurs de la table B



Combien de lignes et de colonnes retournera la jointure naturelle entre les tables A et B?

Table A
null
0
1
2

Table B
2
3
4
null

```
select *
from
  (values (null),(0),(1),(2))
  as a(t)
natural join
(values (2),(3),(4),(null))
  as b(t);
```

```
      t
-----
      2
(1 row)
```

Une jointure naturelle recherche des colonnes ayant le même nom et fait un `inner join` sur ces colonnes.



Combien de lignes retournera l'anti jointure entre les tables A et B?

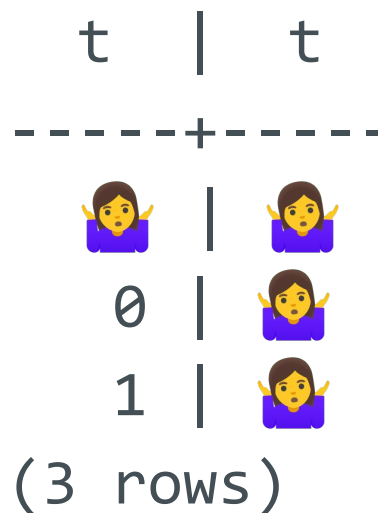
Table A
null
0
1
2

Table B
2
3
4
null



```

select *
from
  (values (null),(0),(1),(2))
  as a(t)
left join
  (values (2),(3),(4),(null))
  as b(t)
on a.t = b.t
where b.t is null;

```



Une anti jointure est une jointure à gauche pour laquelle on ne conserve que les données de la table de gauche non présents dans la table de droite.



À quoi sert
une jointure
latérale?

child	
Byron	1836-05-12
Anne	1837-09-22
Ralph	1906-08-28

```
select name, age,  
       date(birth + ((age + 1)||'years')::interval)  
       as next_birthday,  
       date(birth + ((age + 1)||'years')::interval) - date(now())  
       as days_before_birthday  
from child  
cross join lateral (  
  select extract(years from age(now(),  
    child.birth))::int as age)  
order by age;
```

```

select name, age,
       date(birth + ((age + 1) || 'years')::interval)
       as next_bday,
       date(birth + ((age +
1) || 'years')::interval) - date(now())
       as days_bf_bday
from child
  cross join lateral (
  select extract(years from age(now(),
    child.birth))::int as age)
order by age;

```

name	age	next_bday	days_bf_bday
Ralph	116	2023-08-28	116
Anne	185	2023-09-22	141
Byron	186	2023-05-12	8

(3 rows)

Une jointure latérale permet de référencer une autre table du from dans une sous-requête.



Combien de lignes retournera l'union entre les tables A et B?

Table A
null
0
1
2

Table B
2
3
4
null

```
(values (null), (0), (1), (2))  
union  
(values (2), (3), (4), (null));
```

```
column1  
-----  
4  
2  
0  
  
1  
3  
(6 rows)
```

Union ajoute les lignes résultant de la deuxième requête aux lignes résultant de la première requête tout en supprimant les doublons.



Combien de lignes retournera l'intersection entre les tables A et B?

Table A
null
0
1
2

Table B
2
3
4
null

```
(values (null),(0),(1),(2))  
intersect  
(values (2),(3),(4),(null));
```

```
column1  
-----  
                2  
(1 row)
```

`Intersect` sélectionne les lignes présentes à la fois dans les résultats de la première et la deuxième requête.



Combien de lignes retournera la différence entre les tables A et B?

Table A
null
0
1
2

Table B
2
3
4
null


```
(values (null), (0), (1), (2))  
except  
(values (2), (3), (4), (null));
```

```
column1  
-----  
      0  
      1  
(2 rows)
```

Except sélectionne les lignes résultant de la première requête qui ne sont pas dans la deuxième requête.



Quelles valeurs
retournera cette
requête?

```
select  
  'thomas' ~ 't.*ma',  
  'thomas' !~* 'T.*ma';
```

```
select
  'thomas' ~ 't.*ma',
  'thomas' !~* 'T.*ma';
```

?column?		?column?
t		f

(1 row)

L'opérateur ~ permet de vérifier qu'un texte vérifie une expression régulière en tenant compte de la casse.

L'opérateur !~* permet de vérifier qu'un texte ne vérifie pas une expression régulière sans tenir compte de la casse.

Quelles valeurs
retournera cette
requête?

Table A	
id	value
1	Ada
2	Grace
3	Marissa

```
select *  
from (  
  select id,  
  value,  
  lead(value)  
  over (order by  
id)  from a)  
where value ~ 'Grace';
```

```
select *
from (
  select id,
  value,
  lead(value)
  over (order by
id)  from a)
where value ~ 'Grace';
```

id	value	lead
2	Grace	Marissa

(1 row)

Lead est une fonction de fenêtrage qui permet de récupérer des valeurs des lignes précédentes.



Quelles valeurs
retournera cette
requête?

```
select
  count(*) as unfiltered,
  count(*) filter (
    where i < 5
  ) as filtered
from generate_series(1,10)
As s(i);
```

```
select
  count(*) as unfiltered,
  count(*) filter (
    where i < 5
  ) as filtered
from generate_series(1,10)
As s(i);
```

unfiltered		filtered
10		4

(1 row)

La clause `filter` permet d'ajouter une clause `where` sur un agrégat sans filtrer les lignes de la requête.

EDB Open Source Learning Day Paris - June 13

Marriott Champs-Élysées Hôtel



- **Time:** 8:30 - 18:00
- **Workshops:**
 - Troubleshooting Postgres
 - Running Postgres in K8s

