



EDBTM

SQL Avancé II

Lætitia Avrot



Laetitia Avrot

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

Résumé des épisodes précédents?

SQL est...

- Un langage déclaratif
- Turing-complet
- Mal connu
- Basé à l'origine sur
l'algèbre relationnelle

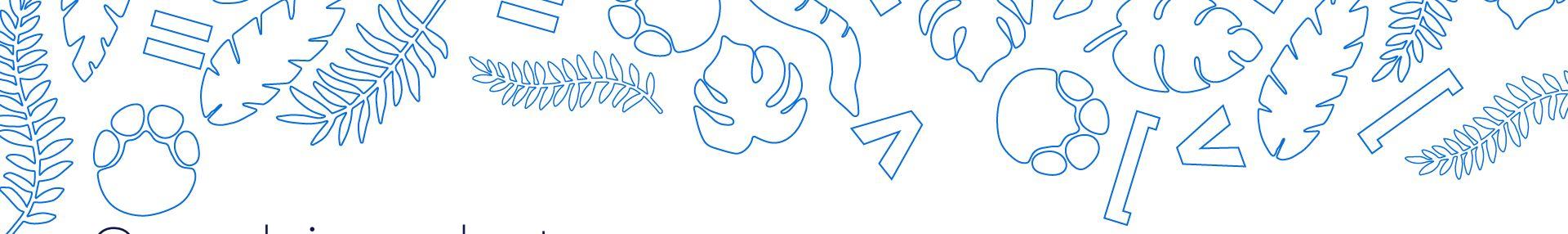
NULL est...

NULL n'est pas...

- Une absence de valeur
- Existant pour tout type de donnée

- Une chaîne vide
- Une chaîne avec uniquement des espaces
- La chaîne NULL

Quizz!



Combien de types
de jointures
existent en SQL?

- 2
- 4
- 7
- 12

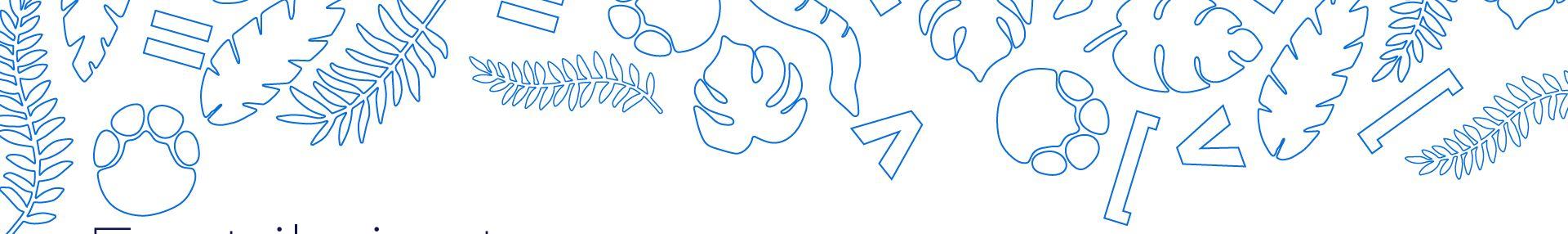


Combien de types de jointures existent en SQL?

- ⓘ Click **Present with Slido** or install our [Chrome extension](#) to activate this poll while presenting.

Il y a 7 sortes de jointures:

- Jointure interne
- Jointure externe à droite ou à gauche
- Jointure externe totale
 - Produit cartésien
 - Jointure naturelle
 - Jointure latérale
 - Anti-jointure



Faut-il ajouter un
group by à cette
requête?

```
select  
    name,  
    avg(salary)  
from employeeHistory
```

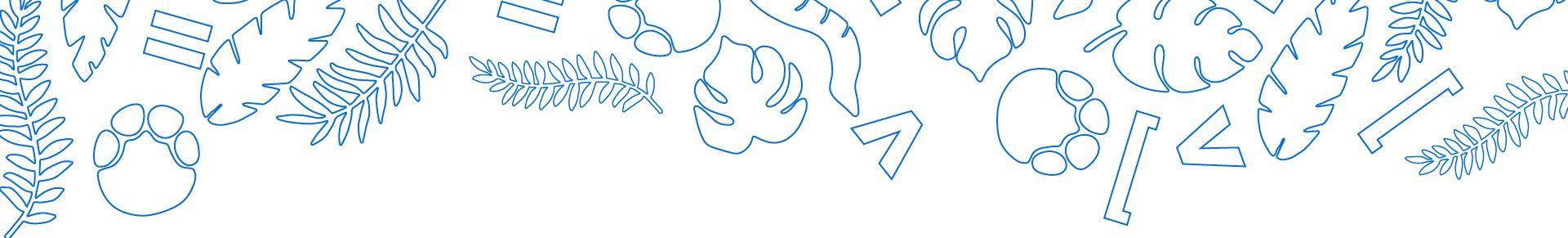


Faut-il ajouter un group by à cette requête?

- ⓘ Click **Present with Slido** or install our [Chrome extension](#) to activate this poll while presenting.

```
select  
    name,  
    avg(salary)  
from employeeHistory  
group by name
```

Group by n'est pas implicite en SQL.
Il est toujours possible de grouper sur des colonnes non
présentes dans le select.
Il est interdit de ne pas mentionner les colonnes non
agrégées du select dans le group by.



Comment peut-on filtrer les résultats d'un agrégat?

- Dans le where
- Dans le select
- Dans une clause having
- Toutes ces réponses sont justes



Comment peut-on filtrer les résultats d'un agrégat?

- ⓘ Click **Present with Slido** or install our [Chrome extension](#) to activate this poll while presenting.

```
select  
    name,  
    avg(salary)  
from employeeHistory  
group by name  
having avg(salary) < 10000
```

Having a été inventé dans ce but. C'est la manière la plus simple de faire.

```
With avgSalary(name, avgSalary) as (
    select
        name,
        avg(salary)
    from employeeHistory
    group by name
)
select *
from avgSalary
where avgSalary < 10000
```

Un peu plus torturé mais faisable.

```
with avgSalary(name, avgSalary) as (
    select
        name,
        avg(salary)
    from employeeHistory
    group by name
)
select name filter
from avgSalary
where avgSalary < 10000
```

Un peu plus torturé mais faisable.

```
with avgSalary(name, avgSalary) as (
    select
        name,
        avg(salary)
    from employeeHistory
    group by name
)
select name filter (where avgSalary < 10000),
       avgSalary (where avgSalary < 10000)
from avgSalary
```



Beaucoup plus torturé.



Comment peut-on faire un auto-incrémentation avec PostgreSQL?

- Manuellement avec une séquence
- Automatiquement avec une séquence et une valeur par défaut
- Automatiquement avec le type serial
- Automatiquement avec une colonne générée
- Toutes les réponses ci-dessus



Comment peut-on faire un auto-incrémentation avec PostgreSQL?

- ⓘ Click **Present with Slido** or install our [Chrome extension](#) to activate this poll while presenting.

```
laetitia=# create table test(id integer primary key, value text);
CREATE TABLE
laetitia=# create sequence my_seq;
CREATE SEQUENCE
laetitia=# insert into test (select nextval('my_seq'), 'blabla');
INSERT 0 1
```

```
laetitia=# create sequence my_seq;  
CREATE SEQUENCE  
laetitia=# create table test (id integer  
default nextval('my_seq') primary key,  
value text);  
CREATE TABLE  
laetitia=# insert into test(value) values  
('blabla');  
INSERT 0 1
```

```
laetitia=# create table test (id serial primary  
key, value text);  
CREATE TABLE  
laetitia=# insert into test (value) values  
( 'blabla');  
INSERT 0 1
```

```
laetitia=# create table test (id integer generated by default as identity primary key, value text);
CREATE TABLE
laetitia=# \d test
              Table "public.test"
   Column |  Type   | Collation | Nullable |          Default
-----+-----+-----+-----+-----+
    id   | integer |           | not null | generated by default as identity
  value | text    |           |           |
Indexes:
  "test_pkey" PRIMARY KEY, btree (id)

laetitia=# insert into test (value) values ('blabla');
INSERT 0 1
```

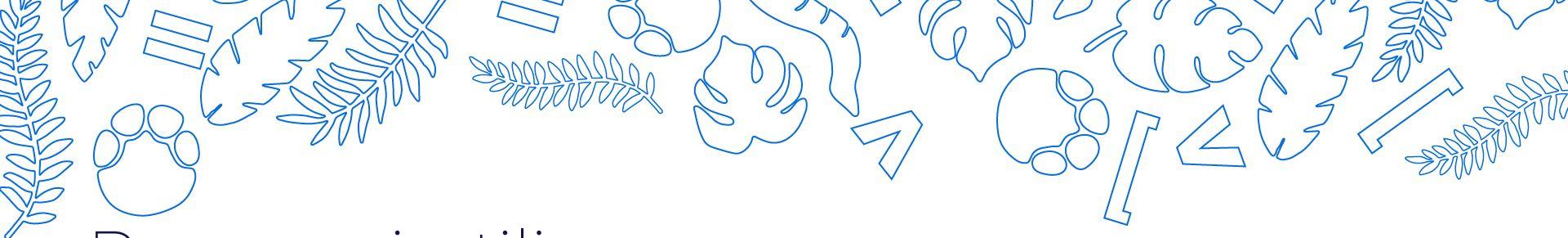
```
laetitia=# insert into test (id, value) values  
(2,'blabla');
```

ERROR: cannot **insert** a non-**DEFAULT** value into
column "id"

DETAIL: Column "id" is an **identity column**
defined as **GENERATED ALWAYS**.

HINT: Use **OVERRIDING SYSTEM VALUE** to override.

	Sequence	Serial	Identity column
Nextval automatically as default value	No	Yes	Yes
Not null constraint	No	Yes	Yes
Prevent manual insert	No	No	With `always`



Pourquoi utiliser les CTEs (Common Table Expressions)?

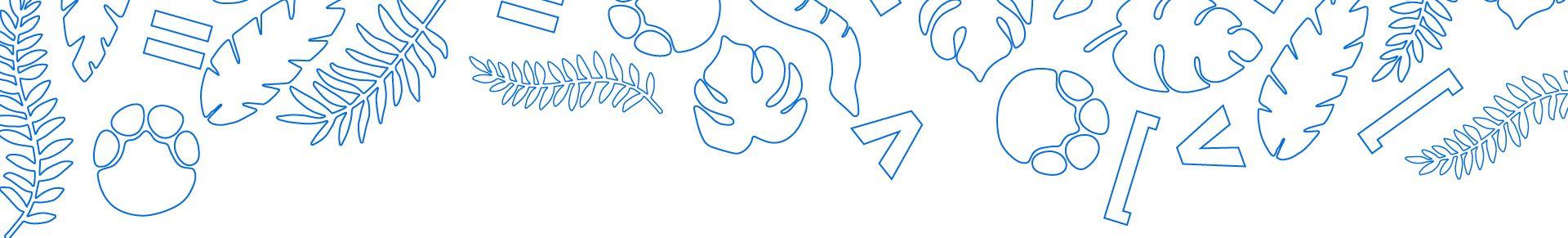
- Pour montrer aux devs tout ce qu'on sait faire
- Pour une meilleure lecture naturelle du code
- Pour une embrouiller l'optimiseur



Pourquoi utiliser les CTEs (Common Table Expressions)?

- ⓘ Click **Present with Slido** or install our [Chrome extension](#) to activate this poll while presenting.

```
with CTENName1 (list of CTE columns) as (
...
),
CTENName2 (list of CTE columns) as (
...
)
Select columnName
From CTENName2
```



Not in est souvent
plus rapide que
Not exist

- Vrai
- Faux



Not in est souvent plus rapide que Not exist

- ⓘ Click **Present with Slido** or install our [Chrome extension](#) to activate this poll while presenting.

```
select surname,  
      firstname  
from members  
where "memid" not in  
(  
    select memid  
    from bookings  
)
```

```
select surname,  
      firstname  
from members  
where not exists  
(  
    select 1  
    from bookings  
    where bookings.memid = members.memid  
)
```



Quelle différence y a-t-il entre cube et rollup?

- Cube est hiérarchisé alors que Rollup prend une combinaison de toutes les colonnes
- Cube contrairement à rollup demande un grouping set
- Cube n'a pas d'application réelle contrairement à rollup
- Toutes les réponses ci-dessus



Quelle différence y a-t-il entre cube et rollup

- ⓘ Click **Present with Slido** or install our [Chrome extension](#) to activate this poll while presenting.

Cube permet de faire des agrégats hiérarchisés.

Par exemple, on peut calculer les sommes des ventes avec des sous-totaux par mois et années.

Rollup va faire des sous-totaux sur toutes les combinaisons de colonnes possibles.



Quelques ressources

(Si vous voulez améliorer votre
SQL)

- <https://mystery.knightlab.com/>
- <https://pqexercises.com/>
- <https://modern-sql.com/>
- <https://theartofpostgresql.com/>



EDB™

Merci!