



**EDB™**

Advanced SQL I

Lætitia Avrot



Lætitia Avrot



- Field CTO – EDB
- PostgreSQL Europe Treasurer
- Postgres Women Founder
- Recognized contributor to the PostgreSQL project

# Agenda

- What is SQL?
- Quiz!

# What is SQL?

# A different language

Optional yet indispensable!

- Not object
- Not functional
- Not procedural
- Declarative
- ... But still a programming language!

→ Avec indentation  
→ Avec retours à la ligne  
→ Avec commentaires



# Turing-complete

(Since SQL:1999)

- Capable of recursion
- [Formal proof with a recursive CTE](#)
- [Formal proof with a recursive function](#)
- You can solve any computer problem with it!
- [Adventofcode](#) solution

# Historique

- SQL-86
- SQL-89
  - Adding integrity constraints
- SQL-92
  - The one you probably know
- SQL-1999
  - Adding regular expression, arrays, recursion, triggers and stored procedure
- SQL-2003
  - Adding XMN, window functions, sequences and generated columns
- SQL-2006
  - Adding XQueries

# Historique

- **SQL-2008**
  - Adding `truncate`, `fetch` and `triggers` instead of
- **SQL-2011**
  - Adding des tables temporeles, amélioration de la clause `fetch` et des fonctions de fenêtrage
- **SQL-2016**
  - Adding JSON
- **SQL-2019**
  - Adding multidimensional arrays
- **SQL-2023**
  - Better management of JSONA, adding SQL/PGQ (Property Graph Queries)



# Quiz!





Where can I use  
`values`?

- By itself
- Within an insert
- Within a from
- What is `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 is a table constructor.



Where can I use  
a subquery?

- In a select
- In a from
- In an order by
- In a 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);
```

Only if your  
subquery's result is  
one row, one column.

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

Only if the subquery's result is  
a single boolean.



How many rows  
will the join  
between A et B  
return?

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 returns rows when the values are equal.



How many rows  
will the left join  
between A et B  
return?

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 returns rows when the values are equal and add all values from the left table



How many rows will the full join between A et B return?









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	
1	
2	2
	
	4
	3
(7 rows)	

Full outer join returns rows when the values are equal and add all values from the left table and from the right tables



How many rows will the cartesian product between A et B return?

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)

A cartesian product will combine all values from table A with all values from table B



How many rows and columns will the natural join between A et B return?

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)
```

A natural join looks for columns sharing the same name and performs an `inner join` based on that.



How many rows will the antijoin between A et B return?

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;

```

t	t
0	🙄
1	🙄
2	🙄

(3 rows)

An antijoin is a left join for which we will keep only values from the left table not in the right table.



What is the point of a lateral join?

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)

A lateral join is useful to use another table listed in the from clause in a subquery.



How many rows will the union between table A and B return?

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 adds the rows from a second query into the result set of a first query while removing duplicates.



How many rows will the intersect between tables A et B return?

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 selects only rows in the first and second result set.



How many rows  
will the difference  
between les  
tables A et B  
return?

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` selects only rows resulting of the first query that are in the result of the second query.



What values will  
this query return?

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

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

?column?		?column?
t		f

(1 row)

The `~` operator checks a text value matches a regular expression (case sensitive).

The `!~*` operator checks a text value matches a regular expression (case insensitive).

What result will  
this query return?

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 is a window function allowing you to use values of the previous rows.



What values will  
this query return?

```
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)		

The `filter` clause allows you to add a `where` clause on an aggregate without filtering the result set.