



SQL Avancé

Fabien Coelho, Claire Medrala

Mines Paris – PSL, MESR

Janvier 2025

- SQL Avancé
- FC/CM
- VALUES
- Séries
- LATERAL
- UPSERT
- RETURNING
- Groups
- WINDOW
- WITH
- JSON
- PL/pgSQL
- EXTENSION
- Conclusion



<https://datacharmer.org/>



Valeurs directes

Séries

Construction d'une table au vol

- construire table constante au vol
- `SELECT ... UNION SELECT ... UNION ...`
- `VALUES (...), (...), ...` utilisé pour `INSERT`, expressions `ANY/ALL...`

justifie le S !

```
SELECT 1 AS jour, mois.nom AS mois
FROM (VALUES
  ('juillet'), ('août'), ('septembre'),
  ('octobre'), ('novembre'), ('décembre'),
  ('...'))
  AS mois(nom);
```

jour	mois
1	juillet
1	août
1	septembre
1	octobre
1	novembre
1	décembre
1	...

- SQL Avancé
- FC/CM
- VALUES
- Séries
- LATERAL
- UPSERT
- RETURNING
- Groups
- WINDOW
- WITH
- JSON
- PL/pgSQL
- EXTENSION
- Conclusion

Séquence énumérée

- combien de vendredi 13 au 20ème siècle ?
- quelles valeurs de clés primaires sont inutilisées ?

Génération d'une relation

generate_series

- génération d'une relation, début, fin et saut
- entiers ou TIMESTAMPTZ (bornes) et `INTERVAL`

```
SELECT COUNT(*) AS "Vendredi 13"
FROM generate_series(1901, 2000) AS year
CROSS JOIN generate_series(1, 12) AS month
WHERE EXTRACT(DOW FROM DATE (year || '-' || month || '-13')) = 5;
```

Vendredi 13
171



Requête à côté

LATERAL



UPSERT

INSERT ou UPDATE ou DELETE

Dans une jointure

- utiliser les valeurs des tuples à sa *gauche*
- nouvelle relation *fonction* de ce qui précède
- utile ? expressions, jointures, agrégations, fenêtres. .

```
-- version avec LATERAL
SELECT i, j
FROM generate_series(0, 3) AS i
CROSS JOIN
  LATERAL generate_series(0, i) AS j;
```

i	j
0	0
1	0
1	1
2	0
2	1
2	2
3	0
3	1
3	2
3	3

Fusion de données

- insertion ou mise à jour si existe déjà
économise un test et sa latence
- différentes syntaxes selon les bases de données. .
`INSERT ... ON CONFLICT ..., MERGE ..., ...`
- `INSERT ... ON CONFLICT DO NOTHING / UPDATE ...`
avec clause `WHERE`, nouvelles données `EXCLUDED`, initiales *NomTable*
- `MERGE INTO ... USING .. ON WHEN ...`
permet toutes les opérations

Exemple INSERT ... ON CONFLICT ...

Données initiales

```
INSERT INTO Heroes(name) VALUES ('Calvin')
ON CONFLICT (name) DO NOTHING;
INSERT INTO Heroes VALUES (2, 'Hobbes')
ON CONFLICT (id) DO UPDATE SET name = EXCLUDED.name;
INSERT INTO Heroes VALUES (3, 'Susie')
ON CONFLICT (name) DO NOTHING;
```

Données finales

id	name
1	Calvin
2	hbs

id	name
1	Calvin
2	Hobbes
3	Susie

Exemple MERGE

Stock initial

nom	qt
Champagne	10
Coteau du Layon	12
Riesling	3

Variations

nom	qt
Champagne	-3
Riesling	-3
Sancerre	6

Stock final

nom	qt
Champagne	7
Coteau du Layon	12
Sancerre	6

```
MERGE INTO Stock AS s
USING LivraisonVente AS lv ON s.nom = lv.nom
WHEN NOT MATCHED AND lv.qt > 0 THEN
  INSERT VALUES(lv.nom, lv.qt)
WHEN MATCHED AND s.qt + lv.qt > 0 THEN
  UPDATE SET qt = s.qt + lv.qt
WHEN MATCHED THEN
  DELETE;
```





Récupération des clés primaires créées

RETURNING



GROUPING SETS, CUBE, ROLLUP

Contexte

- INSERT UPDATE DELETE transformé en SELECT
- si conflit, retourne une relation vide

```
INSERT INTO Things(name) VALUES ('Chair')
RETURNING tid;

INSERT INTO Things(name) VALUES ('Chair')
ON CONFLICT DO NOTHING
RETURNING tid;
```

Aggrégations combinées

- aggrégation sur des sous-ensembles de GROUP BY
- GROUPING SETS liste de groupes de colonnes
- ROLLUP tous les préfixes de colonnes, y compris vide
- CUBE tous les sous-ensembles de colonnes
- équivalent à UNION, valeurs non gardées remplacées par NULL



Exemple GROUPING SETS



Cumuls par pays et par langues

```
CREATE TABLE PaysLanguePopulation(
  id SERIAL PRIMARY KEY,
  pays TEXT NOT NULL,
  langue TEXT NOT NULL,
  pop NUMERIC NOT NULL,
  UNIQUE (pays, langue)
);
```

pays	langue	pop
Allemagne	Allemand	81.5
Autriche	Allemand	8.7
Belgique	Allemand	0.1
Belgique	Français	4.1
Belgique	Néerlandais	7.0
France	Français	66.1
Italie	Italien	60.8
Pays-Bas	Néerlandais	16.9
Suisse	Allemand	5.2
Suisse	Français	1.7
Suisse	Italien	0.5

```
SELECT pays, langue, SUM(pop) AS pop
FROM PaysLanguePopulation
GROUP BY GROUPING SETS
  ((pays), (langue), ())
ORDER BY pays, langue;
```

pays	langue	pop
Allemagne		81.5
Autriche		8.7
Belgique		11.2
France		66.1
Italie		60.8
Pays-Bas		16.9
Suisse		7.4
	Allemand	95.5
	Français	71.9
	Italien	61.3
	Néerlandais	23.9
		252.6



Ou avec UNION

moins performant



Exemple ROLLUP

SQL Avancé
FC/CM
VALUES
Séries
LATERAL
UPsert
RETURNING
Groups
WINDOW
WITH
JSON
PL/pgSQL
EXTENSION
Conclusion

```
SELECT pays AS pays, NULL AS langue,
       SUM(pop) AS pop
FROM PaysLanguePopulation
GROUP BY pays -- premier groupe
UNION
SELECT NULL, langue, SUM(pop)
FROM PaysLanguePopulation
GROUP BY langue -- second groupe
UNION
SELECT NULL, NULL, SUM(pop)
FROM PaysLanguePopulation
-- dernier groupe
ORDER BY pays, langue;
```

pays	langue	pop
Allemagne		81.5
Autriche		8.7
Belgique		11.2
France		66.1
Italie		60.8
Pays-Bas		16.9
Suisse		7.4
	Allemand	95.5
	Français	71.9
	Italien	61.3
	Néerlandais	23.9
		252.6

SQL Avancé
FC/CM
VALUES
Séries
LATERAL
UPsert
RETURNING
Groups
WINDOW
WITH
JSON
PL/pgSQL
EXTENSION
Conclusion

```
SELECT pays, langue,
       SUM(pop) AS pop
FROM PaysLanguePopulation
GROUP BY ROLLUP(pays, langue)
-- équivalent à GROUPING SETS
-- ((pays, langue), (pays), ())
ORDER BY langue, pays
LIMIT 12;
```

pays	langue	pop
Allemagne	Allemand	81.5
Autriche	Allemand	8.7
Belgique	Allemand	0.1
Suisse	Allemand	5.2
Belgique	Français	4.1
France	Français	66.1
Suisse	Français	1.7
Italie	Italien	60.8
Suisse	Italien	0.5
Belgique	Néerlandais	7.0
Pays-Bas	Néerlandais	16.9
Allemagne		81.5



Exemple CUBE

pays	langue	pop
Allemagne	Allemand	81.5
Allemagne	Allemand	81.5
Autriche	Allemand	8.7
Autriche	Allemand	8.7
Belgique	Allemand	0.1
Belgique	Français	4.1
Belgique	Néerlandais	7.0
Belgique		11.2
France	Français	66.1
France		66.1
Italie	Italien	60.8
Italie		60.8
Pays-Bas	Néerlandais	16.9



Fonctions de fenêtrage

window functions

SQL Avancé
FC/CM
VALUES
Séries
LATERAL
UPsert
RETURNING
Groups
WINDOW
WITH
JSON
PL/pgSQL
EXTENSION
Conclusion

```
SELECT pays, langue,
       SUM(pop) AS pop
FROM PaysLanguePopulation
GROUP BY CUBE(pays, langue)
-- équivalent à GROUPING SETS
-- ((pays, langue),
--    (pays), (langue), ())
ORDER BY pays, langue
LIMIT 13;
```

SQL Avancé
FC/CM
VALUES
Séries
LATERAL
UPsert
RETURNING
Groups
WINDOW
WITH
JSON
PL/pgSQL
EXTENSION
Conclusion

Principe

- accès aux tuples voisins dans **SELECT**
un peu comme **GROUP BY**, mais sans le regroupement
- numérotation selon un tri, une partition, les deux. . .

Syntaxe

	fonctions et clauses
agrégation	usuelles COUNT SUM...
fonctions	spécifiques RANK LAG...
filtrage	clause FILTER (WHERE ...)
fenêtre	clause OVER (...)
nommage	pour réutilisation, clause WINDOW ... AS (...)



Aggrégations partielles

FILTER



SQL Avancé
FC/CM
VALUES
Séries
LATERAL
UPSERT
RETURNING
Groups
WINDOW
WITH
JSON
PL/pgSQL
EXTENSION
Conclusion

Syntaxe

- AGG(...) FILTER (WHERE ...)
- applique une aggrégation sur certains tuples seulement

```
-- nombre de films avant et après 1950
SELECT
  COUNT(*) FILTER (WHERE année < 1950) AS "avant",
  COUNT(*) FILTER (WHERE année >= 1950) AS "après",
  COUNT(*) AS "total"
FROM films;
```

avant	après	total
5	7	12

SQL Avancé
FC/CM
VALUES
Séries
LATERAL
UPSERT
RETURNING
Groups
WINDOW
WITH
JSON
PL/pgSQL
EXTENSION
Conclusion

Numérotation selon un tri

Syntaxe

- RANK() OVER (ORDER BY ...)

```
SELECT *,
  RANK() OVER (ORDER BY val DESC)
FROM Notes
ORDER BY id;

SELECT *,
  RANK() OVER (ORDER BY val ASC) AS r1,
  RANK() OVER (ORDER BY val ASC, id ASC) AS r2,
  RANK() OVER (ORDER BY val DESC, id DESC)
  AS r3
FROM Notes
ORDER BY id;
```

id	val	rank
1	10	5
2	15	2
3	17	1
4	13	4
5	15	2

id	val	r1	r2	r3
1	10	1	1	5
2	15	3	3	3
3	17	5	5	1
4	13	2	2	4
5	15	3	4	2



Regroupement selon une partition



SQL Avancé
FC/CM
VALUES
Séries
LATERAL
UPSERT
RETURNING
Groups
WINDOW
WITH
JSON
PL/pgSQL
EXTENSION
Conclusion

Syntaxe

- AVG(...) OVER (PARTITION BY ...)
- à comprendre comme un GROUP BY

```
SELECT eleve, cours, note,
  -- moyenne par cours
  AVG(note) OVER (PARTITION BY cours)
  AS avc,
  -- moyenne par élève
  AVG(note) OVER (PARTITION BY eleve)
  AS ave
FROM Notations
ORDER BY ave DESC, note DESC;
```

eleve	cours	note	avc	ave
Susie	Maths	19	12.25	18.5
Susie	Physics	18	13.25	18.5
Hobbes	Physics	19	13.25	17.5
Hobbes	Maths	16	12.25	17.5
Calvin	Physics	11	13.25	10.5
Calvin	Maths	10	12.25	10.5
Moe	Physics	5	13.25	4.5
Moe	Maths	4	12.25	4.5

SQL Avancé
FC/CM
VALUES
Séries
LATERAL
UPSERT
RETURNING
Groups
WINDOW
WITH
JSON
PL/pgSQL
EXTENSION
Conclusion

Calculer la valeur médiane (le tuple médian)

délais d'attente

nom	delais
calvin	8
mum	13
dad	7
suzy	10
hobbes	123450

attente moyenne

round
24697.6

tuple médian

nom	delais
suzy	10



Aggrégations sur groupes ordonnés

SQL Avancé
FC/CM

Syntaxe

- AGG(...) WITHIN GROUP (ORDER BY ...)
- Fonctions `mode` `percentile_cont` `percentile_disc`

```
SELECT *
FROM AttentePatients
WHERE delais = (
    -- calcule l'attente médiane
    SELECT percentile_disc(0.5) WITHIN GROUP (ORDER BY delais)
    FROM AttentePatients
);
```

21 / 54



Générer une somme partielle

SQL Avancé
FC/CM

VALUES
Séries
LATERAL
UPSERT
RETURNING
Groups
WINDOW
WITH
JSON
PL/pgSQL
EXTENSION
Conclusion

id	quand	montant	description	sum
1	2005-02-01 00:00:00	1000.00	versement initial	1000.00
2	2005-05-09 00:00:00	-100.00	tirage	900.00
3	2005-05-11 00:00:00	-450.00	sous-tirage	450.00
4	2006-01-01 00:00:00	200.00	ouf	650.00
5	2006-12-23 00:00:00	-700.00	joyeux noel	-50.00
6	2007-02-28 00:00:00	123.45	des sous !	73.45

22 / 54



Jointure sur opérateur ≤

SQL Avancé
FC/CM

Principe

- associe *tous* les précédents à chaque opération
- complexité en n^2 ... plus ou moins inutilisable

```
-- jointure speciale...
SELECT cc.id, cc.quand, cc.montant, cc.description,
       SUM(run.montant)
FROM CompteCheque AS cc
JOIN CompteCheque AS run ON run.id <= cc.id
GROUP BY cc.id, cc.quand, cc.montant, cc.description
ORDER BY cc.id ASC;
```

23 / 54



Avec une fenêtre

window

SQL Avancé
FC/CM

Syntaxe

- AGG(...) OVER (ORDER BY ...)

```
SELECT id, quand, montant, description,
       SUM(montant) OVER (ORDER BY id ASC)
FROM CompteCheque;
```

id	quand	montant	description	sum
1	2005-02-01 00:00:00	1000.00	versement initial	1000.00
2	2005-05-09 00:00:00	-100.00	tirage	900.00
3	2005-05-11 00:00:00	-450.00	sous-tirage	450.00
4	2006-01-01 00:00:00	200.00	ouf	650.00
5	2006-12-23 00:00:00	-700.00	joyeux noel	-50.00
6	2007-02-28 00:00:00	123.45	des sous !	73.45

24 / 54



Générer une moyenne mobile

SQL Avancé
FC/CM
VALUES
Séries
LATERAL
UPSERT
RETURNING
Groups
WINDOW
WITH
JSON
PL/pgSQL
EXTENSION
Conclusion

Syntaxe

- AGG(...) OVER (ORDER BY ... ROWS ...)
- description de la fenêtre des valeurs à prendre
- ajustement automatique au début et à la fin

```
SELECT *,
  AVG(qt) OVER (ORDER BY yr
    ROWS BETWEEN 1 PRECEDING AND 1 FOLLOWING)
    AS ma
FROM oilprod
ORDER BY yr ASC;
```

yr	qt	ma
1980	3.1	3.6
1981	4.2	4.2
1982	5.3	5.2
1983	6.2	6.2
1984	7.1	7.1
1985	7.9	7.7
1986	8.1	8.1
1987	8.3	8.1
1988	7.8	7.9
1989	7.6	7.7

25 / 54



Requête *réursive* et fermeture transitive

SQL Avancé
FC/CM
VALUES
Séries
LATERAL
UPSERT
RETURNING
Groups
WINDOW
WITH
JSON
PL/pgSQL
EXTENSION
Conclusion

Nommage d'une requête temporaire

WITH

- sorte de vue locale à une requête, de sous-requête
- calcul indépendant, stockage dans une table temporaire
- utilisation à la suite immédiate
- peut inclure INSERT, UPDATE, DELETE
- attention, barrière d'optimisation : implémentation matérialisée

Calcul itératif

WITH RECURSIVE

- calcul fermeture transitive : arrêt quand ajout vide

26 / 54



Exemple moyenne inférieure à la moyenne

WITH

SQL Avancé
FC/CM
VALUES
Séries
LATERAL
UPSERT
RETURNING
Groups
WINDOW
WITH
JSON
PL/pgSQL
EXTENSION
Conclusion

- réutilisation locale d'une sous-requête nommée...

```
WITH auteur_moy(auteur,moy) AS (
  SELECT nom, AVG(durée)
  FROM Films JOIN Personnes USING (pid)
  GROUP BY nom)
SELECT auteur, moy
FROM auteur_moy
WHERE moy < (SELECT AVG(moy) FROM auteur_moy)
ORDER BY auteur;
```

auteur	moy
Allen	01:42:00
Ozu	01:37:00

27 / 54



Fermeture transitive

WITH RECURSIVE

SQL Avancé
FC/CM
VALUES
Séries
LATERAL
UPSERT
RETURNING
Groups
WINDOW
WITH
JSON
PL/pgSQL
EXTENSION
Conclusion

- table EnfantDe des ascendants directs
 - calcul des ascendants et leur degré
 - stockage table Ascendant
- ```
CREATE TABLE Ascendant(
 enfant TEXT NOT NULL,
 parent TEXT NOT NULL,
 degre INTEGER NOT NULL,
 PRIMARY KEY(enfant,parent)
);
```

### EnfantDe

| enfant | parent       |
|--------|--------------|
| Calvin | Dad          |
| Calvin | Mum          |
| Granny | Great Granny |
| Mum    | Grand Pa     |
| Mum    | Granny       |

28 / 54



Exemple

WITH RECURSIVE



SQL Avancé  
FC/CM  
VALUES  
Séries  
LATERAL  
UPSERT  
RETURNING  
Groups  
WINDOW  
WITH  
JSON  
PL/pgSQL  
EXTENSION  
Conclusion

```
WITH RECURSIVE Ascend(enfant,parent,degre)
AS (
 -- initialisation
 SELECT enfant, parent, 1
 FROM EnfantDe
 UNION
 -- itérations
 SELECT ed.enfant, a.parent, 1+a.degre
 FROM Ascend AS a
 JOIN EnfantDe AS ed ON (ed.parent=a.enfant)
)
-- stocke le résultat dans Ascendant
INSERT INTO Ascendant(enfant, parent, degre)
SELECT enfant, parent, degre
FROM Ascend;
```

| enfant | parent       | d |
|--------|--------------|---|
| Calvin | Dad          | 1 |
| Calvin | Mum          | 1 |
| Granny | Great Granny | 1 |
| Mum    | Grand Pa     | 1 |
| Mum    | Granny       | 1 |
| Calvin | Grand Pa     | 2 |
| Calvin | Granny       | 2 |
| Mum    | Great Granny | 2 |
| Calvin | Great Granny | 3 |

SQL Avancé  
FC/CM  
VALUES  
Séries  
LATERAL  
UPSERT  
RETURNING  
Groups  
WINDOW  
WITH  
JSON  
PL/pgSQL  
EXTENSION  
Conclusion

Contraintes

WITH RECURSIVE

- WITH RECURSIVE démarrage
- SELECT simple, contenu initial
- UNION avec l'incrément d'itération
- SELECT sur la table elle-même qui ne doit apparaître qu'une fois !
- calcul fermeture transitive : implémentation par une boucle sur une table temporaire



WITH avec INSERT UPDATE DELETE



Plus longues séquences de températures croissantes

SQL Avancé  
FC/CM  
VALUES  
Séries  
LATERAL  
UPSERT  
RETURNING  
Groups  
WINDOW  
WITH  
JSON  
PL/pgSQL  
EXTENSION  
Conclusion

Retours

■ retour des tuples ajoutés, modifiés ou effacés

RETURNING ...

```
WITH archived AS (
 DELETE FROM Invoice
 WHERE paid AND sent < CURRENT_DATE - INTERVAL '1 month'
 RETURNING *
)
INSERT INTO Archive
SELECT * FROM archived;

WITH changed AS (
 UPDATE stuff
 SET something = 'changed'
 WHERE condition
 RETURNING *
)
SELECT * FROM changed;
```

SQL Avancé  
FC/CM  
VALUES  
Séries  
LATERAL  
UPSERT  
RETURNING  
Groups  
WINDOW  
WITH  
JSON  
PL/pgSQL  
EXTENSION  
Conclusion

<http://tapoueh.org/>

- variations journalières  
LAG(..., 1) OVER (ORDER BY ...)
- longueur des séquences croissantes  
WITH RECURSIVE
- sélection des plus longues  
MAX

| date       | temp |
|------------|------|
| 2024-01-01 | 0.4  |
| 2024-01-02 | 3.4  |
| 2024-01-03 | 4.6  |
| 2024-01-04 | 3.6  |
| 2024-01-05 | 0.2  |
| 2024-01-06 | 0.2  |
| 2024-01-07 | -1.7 |
| 2024-01-08 | 0.5  |
| 2024-01-09 | -0.1 |
| 2024-01-10 | 3.9  |



## Conclusion WITH et RECURSIVE

SQL Avancé

FC/CM

VALUES

Séries

LATERAL

UPSERT

RETURNING

Groups

WINDOW

WITH

JSON

PL/pgSQL

EXTENSION

Conclusion

```
WITH RECURSIVE
Delta AS (-- day-on-day temperature delta
 SELECT LAG(date, 1) OVER (ORDER BY date) AS dstart,
 date AS dend,
 temp - LAG(temp, 1) OVER (ORDER BY date) AS inc
 FROM Temperature),
RisingTemp AS (-- increasing temperature sequences
 SELECT dstart, dend, 1 AS cnt
 FROM Delta
 WHERE inc >= 0)
UNION
 SELECT p.dstart, d.dend, p.cnt + 1
 FROM RisingTemp AS p
 JOIN Delta AS d ON (p.dend = d.dstart)
 WHERE d.inc >= 0)
-- keep longest sequences found
SELECT dstart, MAX(cnt) AS cnt
FROM RisingTemp
GROUP BY dstart
ORDER BY cnt DESC, dstart ASC LIMIT 5;
```

SQL Avancé

FC/CM

VALUES

Séries

LATERAL

UPSERT

RETURNING

Groups

WINDOW

WITH

JSON

PL/pgSQL

EXTENSION

Conclusion

Options associées

■ NOT MATERIALIZED vs MATERIALIZED

précalcul ou non

■ SEARCH BREADTH vs DEPTH ...

ordre d'énumération

■ CYCLE ...

détection de cycles, construction de chemin

SQL est Turing complet

■ <https://blog.coelho.net/category/turing/>

■ WITH RECURSIVE + MAX(..) OVER + CROSS JOIN



## Support JSON

## Support JSON

SQL Avancé

FC/CM

VALUES

Séries

LATERAL

UPSERT

RETURNING

Groups

WINDOW

WITH

JSON

PL/pgSQL

EXTENSION

Conclusion

JSON dans une base de donnée relationnelle

■ représentation hiérarchique d'objects, tableaux, valeurs. ...

■ issue (natif) de JavaScript, RFC 8259

■ modèle *objet* des applications, très flexible, non relationnel

■ traités comme une valeur unique, indexable, manipulable

■ compromis pour limiter la complexité du modèle relationnel

■ intégré au standard SQL:2016 (ISO/IEC 9075:2016)

■ impl. partielles : IBM DB2, Oracle DB, Microsoft SQL Server, Postgres

■ base de données *uniquement* JSON : MongoDB, Couchbase, CouchDB

```
{
 "personnage": "Calvin",
 "age": 6,
 "auteur": "Bill Watterson"
}
```

SQL Avancé

FC/CM

VALUES

Séries

LATERAL

UPSERT

RETURNING

Groups

WINDOW

WITH

JSON

PL/pgSQL

EXTENSION

Conclusion

Implémentation Postgres

types JSON (texte), JSONB (binaire, pré-parsé)

opérateurs nombreux, quelques exemples

■ J @> D : J contient D

■ J -> 2, J -> 'a' accès indexé liste ou objets vers JSON

■ J ? t : clé dans l'object ?

fonctions nombreuses JSON\_\* JSONB\_\*, quelques exemples

■ TO\_JSON conversion vers JSON

■ JSON\_INSERT insertion dans une structure

■ JSON\_PRETTY prettyprinter

aggrégations JSON\_AGG JSONB\_AGG JSON\_OBJECT\_AGG JSONB\_OBJECT\_AGG

relations JSON\_ARRAY\_ELEMENTS JSON\_EACH ...

JSON path langage d'extraction, filtrage pour JSON

Exemples JSON

| SQL Avancé                                                                                                                  | FC/CM | VALUES | Séries | LATERAL | UPSERT | RETURNING | Groups | WINDOW | WITH | JSON | PL/pgSQL                                                                                                                                                                                                                                                                                                                 | EXTENSION | Conclusion |    |       |   |       |   |       |   |       |   |       |   |       |   |       |   |       |                                                                                                    |   |       |   |      |
|-----------------------------------------------------------------------------------------------------------------------------|-------|--------|--------|---------|--------|-----------|--------|--------|------|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|------------|----|-------|---|-------|---|-------|---|-------|---|-------|---|-------|---|-------|---|-------|----------------------------------------------------------------------------------------------------|---|-------|---|------|
| <pre>SELECT i,   JSON '["dim","lun","mar","mer","jeu","ven","sam"]' -&gt; i AS jour FROM generate_series(-1, 7) as i;</pre> |       |        |        |         |        |           |        |        |      |      |                                                                                                                                                                                                                                                                                                                          |           |            |    |       |   |       |   |       |   |       |   |       |   |       |   |       |   |       |                                                                                                    |   |       |   |      |
|                                                                                                                             |       |        |        |         |        |           |        |        |      |      | <table><tr><th>i</th><th>jour</th></tr><tr><td>-1</td><td>"sam"</td></tr><tr><td>0</td><td>"dim"</td></tr><tr><td>1</td><td>"lun"</td></tr><tr><td>2</td><td>"mar"</td></tr><tr><td>3</td><td>"mer"</td></tr><tr><td>4</td><td>"jeu"</td></tr><tr><td>5</td><td>"ven"</td></tr><tr><td>6</td><td>"sam"</td></tr></table> | i         | jour       | -1 | "sam" | 0 | "dim" | 1 | "lun" | 2 | "mar" | 3 | "mer" | 4 | "jeu" | 5 | "ven" | 6 | "sam" | <table><tr><th>v</th></tr><tr><td>"one"</td></tr><tr><td>2</td></tr><tr><td>true</td></tr></table> | v | "one" | 2 | true |
| i                                                                                                                           | jour  |        |        |         |        |           |        |        |      |      |                                                                                                                                                                                                                                                                                                                          |           |            |    |       |   |       |   |       |   |       |   |       |   |       |   |       |   |       |                                                                                                    |   |       |   |      |
| -1                                                                                                                          | "sam" |        |        |         |        |           |        |        |      |      |                                                                                                                                                                                                                                                                                                                          |           |            |    |       |   |       |   |       |   |       |   |       |   |       |   |       |   |       |                                                                                                    |   |       |   |      |
| 0                                                                                                                           | "dim" |        |        |         |        |           |        |        |      |      |                                                                                                                                                                                                                                                                                                                          |           |            |    |       |   |       |   |       |   |       |   |       |   |       |   |       |   |       |                                                                                                    |   |       |   |      |
| 1                                                                                                                           | "lun" |        |        |         |        |           |        |        |      |      |                                                                                                                                                                                                                                                                                                                          |           |            |    |       |   |       |   |       |   |       |   |       |   |       |   |       |   |       |                                                                                                    |   |       |   |      |
| 2                                                                                                                           | "mar" |        |        |         |        |           |        |        |      |      |                                                                                                                                                                                                                                                                                                                          |           |            |    |       |   |       |   |       |   |       |   |       |   |       |   |       |   |       |                                                                                                    |   |       |   |      |
| 3                                                                                                                           | "mer" |        |        |         |        |           |        |        |      |      |                                                                                                                                                                                                                                                                                                                          |           |            |    |       |   |       |   |       |   |       |   |       |   |       |   |       |   |       |                                                                                                    |   |       |   |      |
| 4                                                                                                                           | "jeu" |        |        |         |        |           |        |        |      |      |                                                                                                                                                                                                                                                                                                                          |           |            |    |       |   |       |   |       |   |       |   |       |   |       |   |       |   |       |                                                                                                    |   |       |   |      |
| 5                                                                                                                           | "ven" |        |        |         |        |           |        |        |      |      |                                                                                                                                                                                                                                                                                                                          |           |            |    |       |   |       |   |       |   |       |   |       |   |       |   |       |   |       |                                                                                                    |   |       |   |      |
| 6                                                                                                                           | "sam" |        |        |         |        |           |        |        |      |      |                                                                                                                                                                                                                                                                                                                          |           |            |    |       |   |       |   |       |   |       |   |       |   |       |   |       |   |       |                                                                                                    |   |       |   |      |
| v                                                                                                                           |       |        |        |         |        |           |        |        |      |      |                                                                                                                                                                                                                                                                                                                          |           |            |    |       |   |       |   |       |   |       |   |       |   |       |   |       |   |       |                                                                                                    |   |       |   |      |
| "one"                                                                                                                       |       |        |        |         |        |           |        |        |      |      |                                                                                                                                                                                                                                                                                                                          |           |            |    |       |   |       |   |       |   |       |   |       |   |       |   |       |   |       |                                                                                                    |   |       |   |      |
| 2                                                                                                                           |       |        |        |         |        |           |        |        |      |      |                                                                                                                                                                                                                                                                                                                          |           |            |    |       |   |       |   |       |   |       |   |       |   |       |   |       |   |       |                                                                                                    |   |       |   |      |
| true                                                                                                                        |       |        |        |         |        |           |        |        |      |      |                                                                                                                                                                                                                                                                                                                          |           |            |    |       |   |       |   |       |   |       |   |       |   |       |   |       |   |       |                                                                                                    |   |       |   |      |

37 / 54

Exemples JSONaggrégation

SQL Avancé

FC/CM

VALUES

Séries

LATERAL

UPSERT

RETURNING

Groups

WINDOW

WITH

JSON

PL/pgSQL

EXTENSION

Conclusion

Jour

| jid | jnom     |
|-----|----------|
| 1   | lundi    |
| 2   | mardi    |
| 3   | mercredi |
| 4   | jeudi    |
| 5   | vendredi |
| 6   | samedi   |
| 7   | dimanche |

```
SELECT JSON_AGG(jnom ORDER BY jid) AS jours FROM Jour;
```

| jours                                                                     |
|---------------------------------------------------------------------------|
| ["lundi", "mardi", "mercredi", "jeudi", "vendredi", "samedi", "dimanche"] |

38 / 54

Exemples JSONobjet

SQL Avancé

FC/CM

VALUES

Séries

LATERAL

UPSERT

RETURNING

Groups

WINDOW

WITH

JSON

PL/pgSQL

EXTENSION

Conclusion

```
CREATE TABLE IF NOT EXISTS Book(
 title TEXT NOT NULL,
 toc JSONB NOT NULL
);

INSERT INTO Book(title, toc) VALUES
('The Christmas Murder Game',
'{ "title": "The Christmas Murder Game",
 "contents": [
 {"n": 1, "title": "Chapter One"},
 {"n": 2, "title": "Chapter Two"},
 {"n": 3, "title": "Chapter Three"},
 {"n": 43, "title": "Chapter Forty-Three"}
]
}')
```

39 / 54

Exemples JSONobjet

SQL Avancé

FC/CM

VALUES

Séries

LATERAL

UPSERT

RETURNING

Groups

WINDOW

WITH

JSON

PL/pgSQL

EXTENSION

Conclusion

```
SELECT
 title AS "titre",
 -- titres des première et dernière parties du contenu
 toc -> 'contents' -> 0 -> 'title' AS "début",
 toc -> 'contents' -> -1 -> 'title' AS "fin"
FROM Book;
```

| titre                    | début          | fin                   |
|--------------------------|----------------|-----------------------|
| Tintin et les Picaros    | "Preface"      | "Bibliography"        |
| PostgreSQL Documentation | "Introduction" | "Révisions"           |
| Systèmes d'information   | "Chapter One"  | "Chapter Forty-Three" |

40 / 54

|                                                                                                                                                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                                   |
|------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------|
|                                                                       | PL/pgSQL                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | Postgres SQL Programming Language |
| SQL Avancé<br>FC/CM<br>VALUES<br>Séries<br>LATERAL<br>UPSERT<br>RETURNING<br>Groups<br>WINDOW<br>WITH<br>JSON<br>PL/pgSQL<br>EXTENSION<br>Conclusion | <p>Développement de nouvelles fonctions côté serveur</p> <p><b>natif</b> C et SQL</p> <p><b>intégrés</b> PL/pgSQL, PL/Tcl, PL/Perl et PL/Python</p> <p><b>externes</b> JS, Shell, PHP, Lua, Ruby, Haskell, Java, Rust. . .</p> <p><b>confiance</b> <i>trusted</i> (protections) vs <i>untrusted</i> (plein accès, admin)</p> <p>Nombreux usages</p> <ul style="list-style-type: none"> <li>■ fonctions scalaires ou relationnelles (retour d'une relation)</li> <li>■ nouveaux domaines (types constraints), attributs calculés</li> <li>■ nouveaux opérateurs, agrégations, casts, conversions</li> <li>■ gestion d'événements (triggers)</li> <li>■ fonctions applicatives : combinaison de requêtes</li> </ul> |                                   |

41 / 54

|                                                                                                                                                      |                                                                                                                                                                                                                                                                                                      |                 |
|------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|
|                                                                       | PL/pgSQL                                                                                                                                                                                                                                                                                             | generate_series |
| SQL Avancé<br>FC/CM<br>VALUES<br>Séries<br>LATERAL<br>UPSERT<br>RETURNING<br>Groups<br>WINDOW<br>WITH<br>JSON<br>PL/pgSQL<br>EXTENSION<br>Conclusion | <pre>CREATE OR REPLACE FUNCTION genser(n INTEGER) RETURNS SETOF INTEGER STRICT AS \$\$   DECLARE     i INTEGER;   BEGIN     FOR i IN 1 .. n LOOP       RETURN NEXT i;     END LOOP;     RETURN; -- fin de la relation   END; \$\$ LANGUAGE plpgsql;  SELECT * FROM genser(3); -- (1), (2), (3)</pre> |                 |

43 / 54

|                                                                                                                                                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                |                           |
|------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------|
|                                                                   | PL/pgSQL                                                                                                                                                                                                                                                                                                                                                                                                                                                       | déclarations de fonctions |
| SQL Avancé<br>FC/CM<br>VALUES<br>Séries<br>LATERAL<br>UPSERT<br>RETURNING<br>Groups<br>WINDOW<br>WITH<br>JSON<br>PL/pgSQL<br>EXTENSION<br>Conclusion | <p>Caractéristiques</p> <p><b>signature</b> nom et paramètres, retours</p> <p><b>propriétés</b> gestion <b>NULL</b>, optimisable. . .</p> <p><b>corps</b> source de la fonction (script) ou librairie dynamique (compilé)</p> <p><b>langage</b> utilisé pour interpréter la fonction</p> <pre>CREATE OR REPLACE FUNCTION plus_un(i INTEGER) RETURNS INTEGER IMMUTABLE STRICT PARALLEL SAFE AS \$\$   BEGIN     RETURN i+1;   END; \$\$ LANGUAGE plpgsql;</pre> |                           |

42 / 54

|                                                                                                                                                      |                                                                                                                                                                                                                                                                                                                                                                                           |                             |
|------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------|
|                                                                   | PL/pgSQL                                                                                                                                                                                                                                                                                                                                                                                  | opérateur + sur <b>TEXT</b> |
| SQL Avancé<br>FC/CM<br>VALUES<br>Séries<br>LATERAL<br>UPSERT<br>RETURNING<br>Groups<br>WINDOW<br>WITH<br>JSON<br>PL/pgSQL<br>EXTENSION<br>Conclusion | <pre>-- fonction de support de l'opérateur CREATE FUNCTION addtext(a TEXT, b TEXT) RETURNS INTEGER IMMUTABLE STRICT AS \$\$   BEGIN RETURN LENGTH(a) + LENGTH(b); END; \$\$ LANGUAGE plpgsql;  -- création de l'opérateur CREATE OPERATOR + ( PROCEDURE = addtext, LEFTARG = TEXT, RIGHTARG = TEXT,   COMMUTATOR = + -- + commute avec lui-même );  SELECT 'hello' + 'world'; -- 10</pre> |                             |

44 / 54

|                                                                                                                                                                                                                                                                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |                       |                                                                                     |                                                                                                                                                                                                                                                                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                                                                                                                                                          | PL/pgSQL                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | cast de DATE vers INT |  | PL/pgSQL                                                                                                                                                                                                                                                                | aggrégation prems                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <div>SQL Avancé</div> <div>FC/CM</div> <div>VALUES</div> <div>Séries</div> <div>LATERAL</div> <div>UPSERT</div> <div>RETURNING</div> <div>Groups</div> <div>WINDOW</div> <div>WITH</div> <div>JSON</div> <div>PL/pgSQL</div> <div>EXTENSION</div> <div>Conclusion</div> | <pre> -- fonction conversion date vers int CREATE OR REPLACE FUNCTION date2int(d DATE) RETURNS INT IMMUTABLE STRICT AS \$\$     SELECT EXTRACT(YEAR FROM d); \$\$ LANGUAGE SQL;  -- conversion automatique CREATE CAST(Date AS Integer) WITH FUNCTION date2int(date) AS IMPLICIT;  SELECT 2 * CURRENT_DATE; -- 4050 en 2025 </pre>                                                                                                                                                                                                                                                                                                                                   |                       |                                                                                     | <div>SQL Avancé</div> <div>FC/CM</div> <div>VALUES</div> <div>Séries</div> <div>LATERAL</div> <div>UPSERT</div> <div>RETURNING</div> <div>Groups</div> <div>WINDOW</div> <div>WITH</div> <div>JSON</div> <div>PL/pgSQL</div> <div>EXTENSION</div> <div>Conclusion</div> | <pre> -- state function: State X Item -&gt; State CREATE OR REPLACE FUNCTION premsf(s ANYELEMENT, a ANYELEMENT) RETURNS ANYELEMENT IMMUTABLE AS \$\$ SELECT COALESCE(s, a); \$\$ LANGUAGE SQL;  -- first non-null aggregation CREATE AGGREGATE prems(ANYELEMENT) (     STYPE = ANYELEMENT,     SFUNC = premsf );  -- test SELECT prems(titre) FROM oeuvre; </pre>                                                                                                                                                                                                                                                                                                                |
| 45 / 54                                                                                                                                                                                                                                                                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |                       | 46 / 54                                                                             |                                                                                                                                                                                                                                                                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|                                                                                                                                                                                          | PL/pgSQL                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | trigger               |  | PL/pgSQL                                                                                                                                                                                                                                                                | requêtes combinées                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| <div>SQL Avancé</div> <div>FC/CM</div> <div>VALUES</div> <div>Séries</div> <div>LATERAL</div> <div>UPSERT</div> <div>RETURNING</div> <div>Groups</div> <div>WINDOW</div> <div>WITH</div> <div>JSON</div> <div>PL/pgSQL</div> <div>EXTENSION</div> <div>Conclusion</div> | <pre> -- automatic update of attributes changed and who CREATE OR REPLACE FUNCTION last_change() RETURNS TRIGGER AS \$\$ BEGIN     RAISE NOTICE 'setting changed on %', TG_TABLE_NAME;     NEW.changed = NOW();     NEW.who = CURRENT_USER;     RETURN NEW; END; \$\$ LANGUAGE plpgsql;  CREATE TABLE Stuff(     id SERIAL PRIMARY KEY, data TEXT NOT NULL,     changed TIMESTAMPTZ NOT NULL, who TEXT NOT NULL);  CREATE TRIGGER stuff_last_change BEFORE INSERT OR UPDATE ON Stuff FOR EACH ROW EXECUTE PROCEDURE last_change();  INSERT INTO Stuff(data) VALUES ('Calvin'), ('Susie'), ('hobbes'); UPDATE Stuff SET data = 'Hobbes' WHERE data = 'hobbes'; </pre> |                       |                                                                                     | <div>SQL Avancé</div> <div>FC/CM</div> <div>VALUES</div> <div>Séries</div> <div>LATERAL</div> <div>UPSERT</div> <div>RETURNING</div> <div>Groups</div> <div>WINDOW</div> <div>WITH</div> <div>JSON</div> <div>PL/pgSQL</div> <div>EXTENSION</div> <div>Conclusion</div> | <pre> CREATE OR REPLACE FUNCTION ajoute_ouvrage_fr(titre TEXT, annee INT, col TEXT, ed TEXT) RETURNS INT VOLATILE STRICT AS \$\$     DECLARE         oid INT; cid INT; eid INT;     BEGIN         SELECT e.oid INTO eid FROM editeur AS e WHERE e.enom = ed;         SELECT c.cid INTO cid FROM collection AS c WHERE c.cnom = col;         INSERT INTO oeuvre AS o (titre, annee, cid, lid) VALUES (titre, annee, cid, 1)             RETURNING o.oid INTO oid;         INSERT INTO exemplaire(oid, eid) VALUES (oid, eid);         RETURN oid;     END; \$\$ LANGUAGE plpgsql;  SELECT ajoute_ouvrage_fr('Un monde un peu meilleur', 2017, 'Lapinot', 'L'association'); </pre> |
| 47 / 54                                                                                                                                                                                                                                                                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |                       | 48 / 54                                                                             |                                                                                                                                                                                                                                                                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |



# PL/pgSQL

# conclusion

SQL Avancé  
FC/CM  
VALUES  
Séries  
LATERAL  
UPSERT  
RETURNING  
Groups  
WINDOW  
WITH  
JSON  
PL/pgSQL  
EXTENSION  
Conclusion

## Langage de programmation simple basé sur SQL

- très bien intégré, accès direct à SQL
- mais syntaxe bof, debug avec RAISE NOTICE

## Disponible par défaut dans Postgres

- nombreux autres choix possibles
- moins bien intégrés

## Nouvelles fonctionnalités dans le serveur

- partage, simplification, performance...



# Exemples d'extensions disponibles

SQL Avancé  
FC/CM  
VALUES  
Séries  
LATERAL  
UPSERT  
RETURNING  
Groups  
WINDOW  
WITH  
JSON  
PL/pgSQL  
EXTENSION  
Conclusion

- pgmp entiers et rationels

units gestion d'unités SI

semver gestion de versions

pgsphere coordonnées sphériques

safeupdate requiert une clause WHERE sur DELETE UPDATE

UPDATE Mesure SET valeur = 10.0; -- Error!

cron tâches périodiques dans Postgres

dirtyread accès aux tuples cachés par le MVCC

orafce compatibilité avec Oracle

tuplock verrouillage de tuples

pgvector recherche de similarité vectorielle (eg LLM)
- POW(MPZ '2', 1024)

UNIT '3.6 km/hour'

SEMVER '1.2.3-beta'

SPOINT '(0d, 90d)'



# Extensions Postgres

SQL Avancé  
FC/CM  
VALUES  
Séries  
LATERAL  
UPSERT  
RETURNING  
Groups  
WINDOW  
WITH  
JSON  
PL/pgSQL  
EXTENSION  
Conclusion

## Extensions

- regroupement cohérent d'objets liés à une extension  
fonctions, opérateurs, agrégations, casts, tables...
- installation/désinstallation automatisée, gestion des versions...
- CREATE EXTENSION intagg;
- pgxn : PostgreSQL eXtension Network <https://pgxn.org>  
répertoire d'extensions, description standardisée, tags...



# Exemples d'utilisations d'extensions

SQL Avancé  
FC/CM  
VALUES  
Séries  
LATERAL  
UPSERT  
RETURNING  
Groups  
WINDOW  
WITH  
JSON  
PL/pgSQL  
EXTENSION  
Conclusion

## Extension pgmp

- Nombre de Mersenne si  $2^n - 1$  est premier

```
SELECT
 n AS "M",
 probab_prime((MPZ '1' << n) - 1, 10) AS prime
FROM generate_series(606, 609) AS n;
```

| M   | prime |
|-----|-------|
| 606 | 0     |
| 607 | 1     |
| 608 | 0     |
| 609 | 0     |



## Exemples d'utilisations d'extensions

SQL Avancé

FC/CM

VALUES

Séries

LATERAL

UPSERT

RETURNING

Groups

WINDOW

WITH

JSON

PL/pgSQL

EXTENSION

Conclusion

### Extension pg\_sphere

- distance entre deux coordonnées sphériques

<->

| nom      | coord (radians)                            |
|----------|--------------------------------------------|
| New-York | (4.991641660703783 , 0.7120943348136864)   |
| Paris    | (0.04014257279586957 , 0.8534660042252271) |
| Sydney   | (2.638937829015426 , -0.5759586531581288)  |
| Tokyo    | (2.4399702942880728 , 0.6230825429619757)  |

```
SELECT v1.nom, v2.nom,
 ROUND((v1.coord <-> v2.coord) * 6371) AS "km"
FROM Ville AS v1 CROSS JOIN Ville AS v2
WHERE v1.nom < v2.nom
ORDER BY 3 DESC, 1, 2;
```

| nom      | nom    | km    |
|----------|--------|-------|
| Paris    | Sydney | 16896 |
| New-York | Sydney | 15950 |
| New-York | Tokyo  | 10835 |
| Paris    | Tokyo  | 9713  |
| Sydney   | Tokyo  | 7731  |
| New-York | Paris  | 5826  |

53 / 54



## Conclusion

SQL Avancé

FC/CM

VALUES

Séries

LATERAL

UPSERT

RETURNING

Groups

WINDOW

WITH

JSON

PL/pgSQL

EXTENSION

Conclusion

### Autres sujets

- **PREPARE EXECUTE** préparation des requêtes répétées
- **INHERITS** héritage entre tables, relationnel-objet
- **FOREIGN DATA WRAPPER** accès à une autre DB  
Oracle, MySQL, MS SQL Server... Amazon S3, LDAP, Twitter...

### Bibliographie

- Documentation Postgres <https://www.postgresql.org/docs/current/>
- Planet PostgreSQL <https://planet.postgresql.org/>
- SQL Hacks, Tips & Tools for Digging into Your Data  
Andrew Cumming & Gordon Russel, O'Reilly Hacks Series

54 / 54