

BDW - Fonctions et opérateurs SQL

Fabien Duchateau

fabien.duchateau [at] univ-lyon1.fr

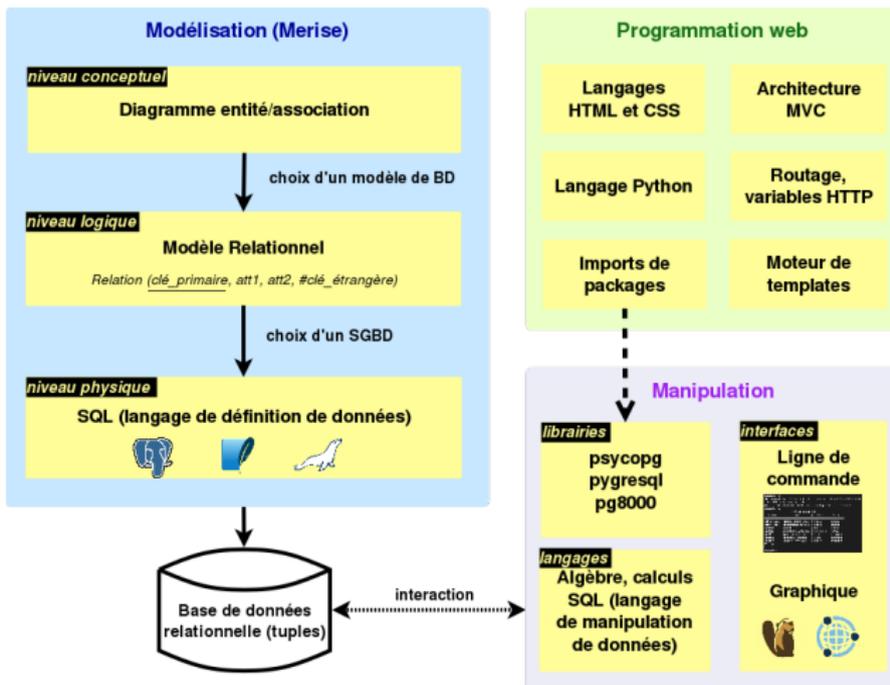
Université Claude Bernard Lyon 1

2024 - 2025



<https://perso.liris.cnrs.fr/fabien.duchateau/BDW/>

Positionnement dans BDW



Ces diapositives utilisent **le genre féminin** (e.g., chercheuse, développeuses) plutôt que **l'écriture inclusive** (moins accessible, moins concise, et pas totalement inclusive)

Pourquoi des fonctions et opérateurs ?

- ▶ Manipuler les chaînes de caractères (e.g., recherche de sous-chaine, concaténation)
- ▶ Réaliser des calculs mathématiques (e.g., somme, division, moyenne)
- ▶ Manipuler des dates (e.g., différence entre dates, formatage)
- ▶ Convertir entre types de données (e.g., caractère vers entier)

Se référer à la documentation spécifique au SGBD utilisé !

<https://www.postgresql.org/docs/current/functions.html>

Plan

Opérateurs

Fonctions chaînes

Fonctions numériques

Fonctions date

Fonctions diverses

Opérateurs arithmétiques

- ▶ + (addition)
- ▶ - (soustraction)
- ▶ * (multiplication)
- ▶ / division (4 chiffres après la virgule par défaut)
- ▶ DIV (division entière)
- ▶ MOD ou % (modulo, i.e., reste de la division entière)
- ▶ ...

<https://www.postgresql.org/docs/current/functions-math.html>

Exemple d'opérateur arithmétique

idE	nomE	moyenneLycée	effectifLycée
123	Ana	19.5	1000
234	Bob	18	1500
345	Chloé	17.5	500
456	Damien	19.5	1000
543	Chloé	17	2000
567	Éléonore	14.5	2000
654	Ana	19.5	1000
678	Farid	19	200
765	Joana	14.5	1500
789	Gisèle	17	800
876	Irène	19.5	400
898	Hector	18.5	800

Table ÉLÈVE

nomU	ville	effectif
INSA	Lyon	36000
UCB	Lyon	15000
UJF	Grenoble	10000
UJM	Saint-Étienne	21000

Table UNIVERSITÉ

idE	nomU	département	décision
123	INSA	informatique	O
123	UCB	électronique	N
123	UCB	informatique	O
123	UJM	électronique	O
234	INSA	biologie	N
345	UJF	bioinformatique	O
345	UJM	bioinformatique	N
345	UJM	électronique	N
345	UJM	informatique	O
543	UJF	informatique	N
678	UCB	histoire	O
765	UCB	histoire	O
765	UJM	histoire	N
765	UJM	psychologie	O
876	UCB	informatique	N
876	UJF	biologie	O
876	UJF	biologie marine	N
898	INSA	informatique	O
898	UCB	informatique	O

Table CANDIDATURE

La moyenne des élèves sur 100

```
SELECT idE, moyenneLycee * 5 AS moyenne100
FROM Eleve;
```

idE	moyenne100
123	97.5
234	90
345	87.5
456	97.5
...	...

Opérateurs de comparaison

Comparaison de deux expressions selon leur ordre naturel :

- ▶ $e_1 \Theta e_2$ avec Θ un opérateur parmi $\{=, \neq, <, >, \geq, \leq\}$
- ▶ `[NOT] BETWEEN ... AND` teste une valeur dans un intervalle
- ▶ `IS [NOT] NULL` teste une valeur nulle
- ▶ `[NOT] IN` teste si une valeur se trouve dans une liste
- ▶ `IS [NOT] TRUE`, `IS [NOT] FALSE`, et `IS [NOT] UNKNOWN`
- ▶ ...

<https://www.postgresql.org/docs/current/functions-comparison.html>

Exemple de comparaison

idE	nomE	moyenneLycée	effectifLycée
123	Ana	19.5	1000
234	Bob	18	1500
345	Chloé	17.5	500
456	Damien	19.5	1000
543	Chloé	17	2000
567	Éléonore	14.5	2000
654	Ana	19.5	1000
678	Farid	19	200
765	Joana	14.5	1500
789	Gisèle	17	800
876	Irène	19.5	400
898	Hector	18.5	800

Table ÉLÈVE

nomU	ville	effectif
INSA	Lyon	36000
UCB	Lyon	15000
UJF	Grenoble	10000
UJM	Saint-Étienne	21000

Table UNIVERSITÉ

idE	nomU	département	décision
123	INSA	informatique	O
123	UCB	électronique	N
123	UCB	informatique	O
123	UJM	électronique	O
234	INSA	biologie	N
345	UJF	bioinformatique	O
345	UJM	bioinformatique	N
345	UJM	électronique	N
345	UJM	informatique	O
543	UJF	informatique	N
678	UCB	histoire	O
765	UCB	histoire	O
765	UJM	histoire	N
765	UJM	psychologie	O
876	UCB	informatique	N
876	UJF	biologie	O
876	UJF	biologie marine	N
898	INSA	informatique	O
898	UCB	informatique	O

Table CANDIDATURE

Les universités dont le nom diffère de 'UCB'

```
SELECT nomU
FROM Universite
WHERE nomU != 'UCB';
```

nomU
INSA
UJF
UJM

Plan

Opérateurs

Fonctions chaînes

Fonctions numériques

Fonctions date

Fonctions diverses

Fonctions chaînes de caractères

- ▶ `CONCAT(s1, s2, ...)` concatène les chaînes *s₁*, *s₂*, ...
- ▶ `LENGTH(s)` retourne la longueur de la chaîne *s*
- ▶ `SUBSTR(s, pos, len)` extrait de *s* la sous-chaîne qui débute à l'index *pos* et de longueur *len*
- ▶ `REPLACE(s, old, new)` remplace dans *s* toutes les occurrences de la chaîne *old* par la chaîne *new*
- ▶ ...

<https://www.postgresql.org/docs/current/functions-string.html>

Exemple de fonctions chaînes de caractères

idE	nomE	moyenneLycée	effectifLycée
123	Ana	19.5	1000
234	Bob	18	1500
345	Chloé	17.5	500
456	Damien	19.5	1000
543	Chloé	17	2000
567	Éléonore	14.5	2000
654	Ana	19.5	1000
678	Farid	19	200
765	Joana	14.5	1500
789	Gisèle	17	800
876	Irène	19.5	400
898	Hector	18.5	800

Table ÉLÈVE

nomU	ville	effectif
INSA	Lyon	36000
UCB	Lyon	15000
UJF	Grenoble	10000
UJM	Saint-Étienne	21000

Table UNIVERSITÉ

idE	nomU	département	décision
123	INSA	informatique	O
123	UCB	électronique	N
123	UCB	informatique	O
123	UJM	électronique	O
234	INSA	biologie	N
345	UJF	bioinformatique	O
345	UJM	bioinformatique	N
345	UJM	électronique	N
345	UJM	informatique	O
543	UJF	informatique	N
678	UCB	histoire	O
765	UCB	histoire	O
765	UJM	histoire	N
765	UJM	psychologie	O
876	UCB	informatique	N
876	UJF	biologie	O
876	UJF	biologie marine	N
898	INSA	informatique	O
898	UCB	informatique	O

Table CANDIDATURE

Le nom et la ville des universités, séparés par un tiret

```
SELECT CONCAT(nomU, '-', ville) AS nomville
FROM Université;
```

nomville
INSA-Lyon
UCB-Lyon
UJF-Grenoble
UJM-Saint-Étienne

Comparaison approximative

- ▶ `s LIKE pat` compare approximativement la chaîne `s` selon le pattern `pat` (`ILIKE` pour la version insensible à la casse)
- ▶ Deux caractères joker pour le pattern `pat` :
 - ▶ `'_'` pour un caractère alphanumérique quelconque
 - ▶ `'%'` pour une chaîne alphanumérique quelconque
- ▶ Fonctions plus complexes avec des expressions régulières

Exemples de patterns :

`'a__'` \equiv toutes les chaînes de 3 caractères commençant par un 'a'

`'a%'` \equiv toutes les chaînes commençant par un 'a'

`'_n_'` \equiv toutes les chaînes de 3 caractères avec un 'n' en deuxième position

<https://www.postgresql.org/docs/current/functions-matching.html>

Exemple de comparaison approximative

idE	nomE	moyenneLycée	effectifLycée
123	Ana	19.5	1000
234	Bob	18	1500
345	Chloé	17.5	500
456	Damien	19.5	1000
543	Chloé	17	2000
567	Éléonore	14.5	2000
654	Ana	19.5	1000
678	Farid	19	200
765	Joana	14.5	1500
789	Gisèle	17	800
876	Irène	19.5	400
898	Hector	18.5	800

Table ÉLÈVE

nomU	ville	effectif
INSA	Lyon	36000
UCB	Lyon	15000
UJF	Grenoble	10000
UJM	Saint-Étienne	21000

Table UNIVERSITÉ

idE	nomU	département	décision
123	INSA	informatique	O
123	UCB	électronique	N
123	UCB	informatique	O
123	UJM	électronique	O
234	INSA	biologie	N
345	UJF	bioinformatique	O
345	UJM	bioinformatique	N
345	UJM	électronique	N
345	UJM	informatique	O
543	UJF	informatique	N
678	UCB	histoire	O
765	UCB	histoire	O
765	UJM	histoire	N
765	UJM	psychologie	O
876	UCB	informatique	N
876	UJF	biologie	O
876	UJF	biologie marine	N
898	INSA	informatique	O
898	UCB	informatique	O

Table CANDIDATURE

Le nom des élèves qui contient un 'e'

```
SELECT nomE
FROM Eleve
WHERE nomE LIKE '%e%';
```

nomE
Chloé
Hector
Damien
Éléonore
Gisèle
Chloé
Irène

Exemple de comparaison approximative (2)

idE	nomE	moyenneLycée	effectifLycée
123	Ana	19.5	1000
234	Bob	18	1500
345	Chloé	17.5	500
456	Damien	19.5	1000
543	Chloé	17	2000
567	Éléonore	14.5	2000
654	Ana	19.5	1000
678	Farid	19	200
765	Joana	14.5	1500
789	Gisèle	17	800
876	Irène	19.5	400
898	Hector	18.5	800

Table ÉLÈVE

nomU	ville	effectif
INSA	Lyon	36000
UCB	Lyon	15000
UJF	Grenoble	10000
UJM	Saint-Étienne	21000

Table UNIVERSITÉ

idE	nomU	département	décision
123	INSA	informatique	O
123	UCB	électronique	N
123	UCB	informatique	O
123	UJM	électronique	O
234	INSA	biologie	N
345	UJF	bioinformatique	O
345	UJM	bioinformatique	N
345	UJM	électronique	N
345	UJM	informatique	O
543	UJF	informatique	N
678	UCB	histoire	O
765	UCB	histoire	O
765	UJM	histoire	N
765	UJM	psychologie	O
876	UCB	informatique	N
876	UJF	biologie	O
876	UJF	biologie marine	N
898	INSA	informatique	O
898	UCB	informatique	O

Table CANDIDATURE

Le nom et identifiant des élèves qui ont candidaté dans une université avec un nom de trois lettres

```
SELECT DISTINCT e.idE, nomE
FROM Eleve e NATURAL JOIN Candidature
WHERE nomU IN (SELECT nomU
                FROM Universite
                WHERE nomU LIKE '___');
```

e.idE	nomE
123	Ana
345	Chloé
543	Chloé
678	Farid
765	Joana
876	Irène
898	Hector

Plan

Opérateurs

Fonctions chaînes

Fonctions numériques

Fonctions date

Fonctions diverses

Fonctions numériques

- ▶ $\text{ABS}(e)$, $\text{COS}(e)$, $\text{SIN}(e)$, $\text{LOG2}(e)$, $\text{SQRT}(e)$, ...
- ▶ $\text{RAND}()$ retourne un nombre aléatoire dans $[0, 1]$
- ▶ $\text{ROUND}(n, dec)$ arrondit le nombre n à dec décimales (dec peut être négatif pour arrondir la partie entière)
- ▶ $\text{TRUNC}(n, dec)$ tronque le nombre n à dec décimales
- ▶ ...

<https://www.postgresql.org/docs/current/functions-math.html>

Exemple de fonction numérique

idE	nomE	moyenneLycée	effectifLycée
123	Ana	19.5	1000
234	Bob	18	1500
345	Chloé	17.5	500
456	Damien	19.5	1000
543	Chloé	17	2000
567	Éléonore	14.5	2000
654	Ana	19.5	1000
678	Farid	19	200
765	Joana	14.5	1500
789	Gisèle	17	800
876	Irène	19.5	400
898	Hector	18.5	800

Table ÉLÈVE

nomU	ville	effectif
INSA	Lyon	36000
UCB	Lyon	15000
UJF	Grenoble	10000
UJM	Saint-Étienne	21000

Table UNIVERSITÉ

idE	nomU	département	décision
123	INSA	informatique	O
123	UCB	électronique	N
123	UCB	informatique	O
123	UJM	électronique	O
234	INSA	biologie	N
345	UJF	bioinformatique	O
345	UJM	bioinformatique	N
345	UJM	électronique	N
345	UJM	informatique	O
543	UJF	informatique	N
678	UCB	histoire	O
765	UCB	histoire	O
765	UJM	histoire	N
765	UJM	psychologie	O
876	UCB	informatique	N
876	UJF	biologie	O
876	UJF	biologie marine	N
898	INSA	informatique	O
898	UCB	informatique	O

Table CANDIDATURE

La moyenne des élèves sur 10, arrondie à l'entier

```
SELECT idE, round(CAST(moyenneLycée / 2 AS
    NUMERIC), 0) AS moyenne10
FROM Eleve;
```

idE	moyenne10
123	10
234	9
345	9
456	10
...	...

Plan

Opérateurs

Fonctions chaînes

Fonctions numériques

Fonctions date

Fonctions diverses

Fonctions sur les dates/temps

- ▶ `CURRENTTIMESTAMP` et `NOW()` retournent la date et temps courants (format 'YYYY-MM-DD HH:MM:SS.sss')
- ▶ `CURRENTDATE` retourne la date courante (format 'YYYY-MM-DD')
- ▶ `EXTRACT(champ FROM d)` extrait le *champ* donné (YEAR, DAY, SECOND, ...) d'une date *d*
- ▶ `TO__DATE(d, f)` retourne la date *d* au format *f*
- ▶ $d_1 - d_2$ retourne le nombre de jours entre les dates d_1 et d_2
- ▶ ...

<https://www.postgresql.org/docs/current/functions-formatting.html>

<https://www.postgresql.org/docs/current/functions-datetime.html>

Exemple de fonction date

idE	nomE	moyenneLycée	effectifLycée
123	Ana	19.5	1000
234	Bob	18	1500
345	Chloé	17.5	500
456	Damien	19.5	1000
543	Chloé	17	2000
567	Éléonore	14.5	2000
654	Ana	19.5	1000
678	Farid	19	200
765	Joana	14.5	1500
789	Gisèle	17	800
876	Irène	19.5	400
898	Hector	18.5	800

Table ÉLÈVE

nomU	ville	effectif
INSA	Lyon	36000
UCB	Lyon	15000
UJF	Grenoble	10000
UJM	Saint-Étienne	21000

Table UNIVERSITÉ

idE	nomU	département	décision
123	INSA	informatique	O
123	UCB	électronique	N
123	UCB	informatique	O
123	UJM	électronique	O
234	INSA	biologie	N
345	UJF	bioinformatique	O
345	UJM	bioinformatique	N
345	UJM	électronique	N
345	UJM	informatique	O
543	UJF	informatique	N
678	UCB	histoire	O
765	UCB	histoire	O
765	UJM	histoire	N
765	UJM	psychologie	O
876	UCB	informatique	N
876	UJF	biologie	O
876	UJF	biologie marine	N
898	INSA	informatique	O
898	UCB	informatique	O

Table CANDIDATURE

La date courante formatée

```
SELECT TO_CHAR(NOW(), 'DD Mon YYYY')
AS dateFormatee;
```

dateFormatee
15 septembre 2024

Plan

Opérateurs

Fonctions chaînes

Fonctions numériques

Fonctions date

Fonctions diverses

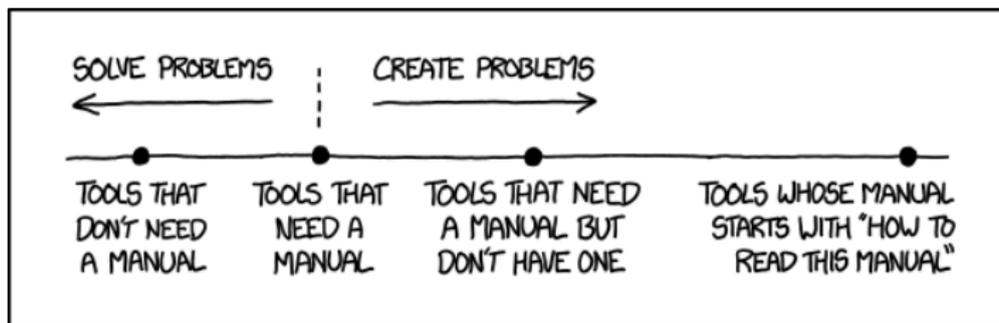
Fonctions diverses

- ▶ `CURRENT_DATABASE()` pour la base de données courante
- ▶ `CURRENT_SCHEMA()` pour le répertoire schéma courant
- ▶ `CURRENT_USER()` pour l'utilisateur de la session
- ▶ `COALESCE(s1, s2, ...)` retourne le premier argument non nul parmi *s₁*, *s₂*, ...
- ▶ ...

<https://www.postgresql.org/docs/current/functions.html>

En résumé

- ▶ Opérateurs et fonctions pour les chaînes, numériques, dates, etc.
- ▶ Autres fonctions (géographiques, hachage, RI, etc.)
- ▶ Fonctions associées aux regroupements (clause GROUP BY)



<https://xkcd.com/>