

BDW - Fonctions et opérateurs SQL

Fabien Duchateau

fabien.duchateau [at] univ-lyon1.fr

Université Claude Bernard Lyon 1

2023 - 2024



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

Positionnement dans BDW

Modélisation

Schéma entité/
association

Niveau conceptuel

Modèle
relationnel

Niveau logique

SQL (DDL)

Niveau physique

SGBD

Concepts

Optimisation

Base de
données

...

Base de
données

Manipulation

Algèbre
relationnelle

Combinaison
d'opérateurs

Calculs
relationnels

{projetés | formule}

SQL (DML)

SELECT ...
FROM ...

Prog. web

HTML

CSS

PHP

```
<html>
...
<link ... css>
...
<?php
?>
...
</html>
```

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 !

<http://mariadb.com/kb/en/mariadb/functions-and-operators/>
<http://dev.mysql.com/doc/>
<http://docs.postgresql.fr/>
<http://docs.oracle.com/en/database/>

Ordre de priorité

Dans la suite, les fonctions et opérateurs sont donnés pour le SGBD MariaDB (et applicable à MySQL)

Comment sont gérées les priorités entre opérateurs et fonctions ?

- ▶ Les fonctions sont prioritaires sur les opérateurs
- ▶ Entre opérateurs, consulter la documentation
- ▶ Les parenthèses modifient l'ordre de priorité

<https://mariadb.com/kb/en/mariadb/operator-precedence/>
<https://mariadb.com/kb/en/mariadb/parentheses/>

Plan

Opérateurs

Fonctions chaînes

Fonctions numériques

Fonctions date

Fonctions conversion

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://mariadb.com/kb/en/mariadb/arithmetic-operators/>

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

```
18 | SELECT idE, moyenneLycée * 5 AS moyenne100
19 | FROM Élève;
```

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://mariadb.com/kb/en/mariadb/comparison-operators/>

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'

```

21 SELECT nomU
22 FROM Université
23 WHERE nomU != 'UCB';

```

nomU
INSA
UJF
UJM

Plan

Opérateurs

Fonctions chaînes

Fonctions numériques

Fonctions date

Fonctions conversion

Fonctions diverses

Fonctions chaînes de caractères

- ▶ `CONCAT(s_1 , s_2 , ...)` concatène les chaînes s_1 , s_2 , ...
- ▶ `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://mariadb.com/kb/en/mariadb/string-functions/>

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

```
25 | SELECT CONCAT(nomU, '-', ville) AS nom-ville
26 | FROM Université;
```

nom-ville

INSA-Lyon

UCB-Lyon

UJF-Grenoble

UJM-Saint-Étienne

Comparaison approximative

- ▶ `s LIKE pat` compare approximativement la chaîne `s` selon le pattern `pat` (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 expression 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://mariadb.com/kb/en/mariadb/like/>

<https://mariadb.com/kb/en/mariadb/regular-expressions-functions/>

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'

```
28 | SELECT nomE
29 | FROM Élève
30 | 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

```

33 SELECT DISTINCT e.idE, nomE
34 FROM Élève e NATURAL JOIN Candidature
35 WHERE nomU IN (SELECT nomU
36 FROM Université
37 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 conversion

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)
- ▶ $\text{TRUNCATE}(n, dec)$ tronque le nombre n à dec décimales
- ▶ ...

<https://mariadb.com/kb/en/mariadb/numeric-functions/>

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

```
40 SELECT idE, ROUND(moyenneLycée / 2, 0) AS
    moyenne10
41 FROM Élève;
```

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

Plan

Opérateurs

Fonctions chaînes

Fonctions numériques

Fonctions date

Fonctions conversion

Fonctions diverses

Fonctions sur les dates/temps

- ▶ `NOW()` et `SYSDATE()` retournent la date et temps courants (format 'YYYY-MM-DD HH :MM :SS')
- ▶ `CURDATE()` retourne la date courante (format 'YYYY-MM-DD')
- ▶ `YEAR(d)`, `MONTH(d)` et `DAY(d)` extraient respectivement l'année, le mois et le jour d'une date d
- ▶ `DATE__FORMAT(d, f)` retourne la date d au format f
- ▶ `DATEDIFF(d1, d2)` retourne le nombre de jours entre les dates d_1 et d_2
- ▶ ...

<https://mariadb.com/kb/en/mariadb/date-and-time-functions/>

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

```
44 | SELECT DATE_FORMAT(NOW(), '%d%M%Y')
45 | AS dateFormatée;
```

dateFormatée
14 septembre 2023

Plan

Opérateurs

Fonctions chaînes

Fonctions numériques

Fonctions date

Fonctions conversion

Fonctions diverses

Fonctions de conversion

- ▶ Conversion implicite entre certains types et avec certains opérateurs
- ▶ `CAST(e AS t)` convertit l'expression `e` dans le type `t` (voir les types disponibles sur la page [MariaDB data-types](#))
- ▶ Fonction `CONVERT` avec comportement similaire à `CAST`

<https://mariadb.com/kb/en/mariadb/type-conversion/>

<https://mariadb.com/kb/en/mariadb/cast/>

<https://mariadb.com/kb/en/mariadb/convert/>

Plan

Opérateurs

Fonctions chaînes

Fonctions numériques

Fonctions date

Fonctions conversion

Fonctions diverses

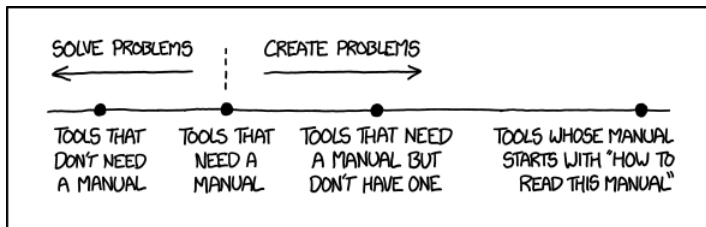
Fonctions diverses

- ▶ `DATABASE()` retourne la base de données courante
- ▶ `CURRENT_USER()` retourne l'utilisatrice courante (et sa machine)
- ▶ `CONNECTION_ID()` retourne l'identifiant de connexion (typiquement du thread)
- ▶ `LAST_INSERT_ID()` retourne l'identifiant (automatiquement généré) de la requête d'insertion la plus récente
- ▶ `ROW_COUNT()` retourne le nombre de tuples affectés par la requête la plus récente (suppression, insertion ou mise à jour)
- ▶ ...

<https://mariadb.com/kb/en/mariadb/information-functions/>

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/>