

République du Cameroun
Ministère de l'enseignement supérieur
Université de Yaoundé 1
Ecole Nationale Supérieure Polytechnique de Yaoundé
Département du génie informatique
3GI

Exposé de Bases de données

Thème : Les fonctions en SQL



Noms des exposants :

- 1. Sanang Sigmou Patrick Mendel*
- 2. Noyessie Hubert*
- 3. Foze Tamo Berenge*
- 4. Ngouh Mfone Aziz*

Sous la supervision du Dr Mbinkeu

Année académique 2014-2015



Introduction

Avec, l'évolution du monde, ainsi que de sa diversité, il est venu que les données concernant son environnement se sont diversifiés. D'où l'invention de langages de manipulation de bases de données comme le SQL dont dérive MYSQL. MySQL est un système de gestion de base de données (SGBD). Pour faciliter la tâche de recherche et de construction de table on retrouve certaines fonctions prédéfinies. Nous porterons notre étude sur les fonctions importantes.



I. Les fonctions de manipulation des chaînes de caractères

ASCII(c) Retourne le caractère ASCII équivalent.

CHAR(n) Retourne le caractère équivalent dans le jeu de caractères en cours.

CONCAT(c1,c2) Concatène deux chaînes.

INSERT(c1, pos, t, c2) Modifie la chaîne c1 en insérant t caractères de la sous-chaîne c2 à partir de la position pos.

INSTR(c1, c2) Premier indice d'une sous-chaîne c1 dans une chaîne c2.

LENGTH(c) Longueur de la chaîne c.

LEFT(c,n) Extrait les n premiers caractères à c en partant de la gauche

REPLACE(c1, c2, c3) Recherche les c2 présentes dans c1 et les remplace par c3

SUBSTR(c,n,[t]) Extraction de la sous-chaîne c commençant à la position n sur t caractères

TRIM(c1 FROM c2) Enlève les caractères c1 à la chaîne c2(options LEADING et TRAILING pour préciser le sens du découpage). Existent aussi LTRIM et RTRIM qui enlèvent des espaces respectivement au début ou à la fin d'une chaîne.

UPPER(c) met la chaîne c en majuscules

II. Les fonctions de manipulation numérique

ABS(n) Valeur absolue de n

ACOS(n) Arc cosinus (n de -1 à 1), retour exprimé en radians (de 0 à pi)

TRUNCATE(n,m) Coupure de n à m décimales

TAN(n) Tangente de n exprimée en radians de 0 à 2 pi

FLOOR(n) Plus grand entier \leq n

POW(m,n) m puissance n

SQRT(n) Racine carrée de n

III. Les fonctions de manipulation de bits

OR: | OU bits à bits.

AND: & ET bits à bits

XOR: ^ OU exclusif bits à bits

SHL: << Décalage à gauche de n positions

SHR: >> Décalage à droite de n positions

Complément à 1: ~

BIN(n) : chaîne binaire correspondant au code binaire de n

BIT_LENGTH(c) Taille de la chaîne en bits

HEX(ns) Chaîne en hexadécimal représentant ns(nombre ou chaîne).

OCT(n) Chaîne en octal représentant n

OCTET_LENGTH(c) Synonyme de **LENGTH()**

UNHEX(c) Fonction inverse de **HEX**

IV. Les fonctions de manipulation des dates

Fonction	Objectif	Retour
ADDDATE (date, n)	Ajoute n jours à une date (heure).	DATE ou DATETIME
ADDTIME (date1, date2)	Ajoute les deux dates avec date1 TIME ou DATETIME, et date2 TIME	TIME ou DATETIME
DAYNAME (date)	Nom du jour en anglais	VARCHAR
DAY(date) ou DAYOFMONTH (date)	Numéro du jour dans le mois (0 à 31)	INT
DAYOFYEAR (date)	Numéro du jour dans l'année (0 à 366).	INT
WEEKOFYEAR (date)	Numéro de la semaine en cours (1 à 53)	INT
TO_DAYS (date)	Retourne un nombre de jours à partir d'une date ('YYYY-MM-DD' ou YYYYMMDD). Inverse de FROM_DAYS()	INT
TIMESTAMPDIFF (intervalle,int,date)	Retranche à la date (heure) un intervalle du type (idem précédent)	TIMESTAMP
SYSDATE()	Date et heure courantes au format 'YYYYMM-DD HH:MM:SS' ou YYYYMMDDHHMMS S	DATETIME ou INT
CURDATE(), CURRENT_DATE ou CURRENT_DATE()	Date courante ('YYYY-MM-DD' ou YYYYMMDD)	INT ou DATE
CURTIME(),CURRENT_TIME ou CURRENT_TIME()	Heure courante ('HH:MM:SS' or HHMMSS)	INT ou DATE
CURRENT_TIMESTAMP, CURRENT_TIMESTAMP() ou NOW()	Date et heure courantes ('YYYY-MM-DD	INT ou DATETIME

	HH:MM:SS' ou YYYYMMDDHHMMSS)	
DATE(datet)	Extrait une date à partir d'une expression de type DATETIME	DATE

V. Conversions

MySQL autorise des conversions de types implicites ou explicites.

1. Implicites

Il est possible d'affecter, dans une expression ou dans une instruction SQL (INSERT, UPDATE...) une donnée de type numérique (ou date-heure) à une donnée de type VARCHAR (ou CHAR). Il en va de même pour l'affectation d'une colonne VARCHAR par une donnée de type date-heure (ou numérique). On parle ainsi de conversion implicite.

Pour preuve le script suivant ne renvoie aucune erreur :

```
CREATE TABLE Test (c1 DECIMAL(6,3), c2 DATE, c3 VARCHAR(1), c4
CHAR);
```

```
INSERT INTO Test VALUES ('548.45', '20060116', 3, 5);
```

2. Explicites

Une conversion est dite « explicite » quand on utilise une fonction à cet effet. Les fonctions de conversion les plus connues sont CAST et CONVERT (qui respectent la syntaxe de la norme

SQL). Les fonctions de conversion sont décrites dans le tableau suivant :

Fonction	conversion	Exemple
BINARY (expr)	L'expression en bits	Par exemple le test BINARY (brevet)= BINARY ('p1-1 ') brevet étant de type VARCHAR et prenant la valeur PL-x renverra faux
CAST (expression AS type MySQL)	L'expression dans le type en paramètre (BINARY, CHAR, DATE, DATETIME, DECIMAL, SIGNED, TIME, UNSIGNED)	CAST (2 AS CHAR) retourne '2'

CONVERT(c, jeu-car)	La chaîne c dans le jeu de caractères passé en paramètre.	CONVERT ('Ä Ê Í Ø' USING Cp850) jeu de caractère DOS, retourne "? Ê Í?".
---------------------	---	--

VI. Comparaisons

MySQL compare deux variables entre elles en suivant les règles suivantes :

- Si l'une des deux valeurs est NULL, la comparaison retourne NULL (sauf pour l'opérateur <=> qui renvoie vrai si les deux valeurs sont NULL).
- Si les deux valeurs sont des chaînes, elles sont comparées en tant que telles.
- Si les deux valeurs sont des numériques, elles sont comparées en tant que telles.
- Les valeurs hexadécimales sont traitées comme des chaînes de bits si elles ne sont pas

Comparées à des numériques.

- Si l'une des valeurs est TIMESTAMP ou DATETIME et si l'autre est une constante, cette dernière est convertie en TIMESTAMP.
- Dans les autres cas, les valeurs sont comparées comme des numériques (flottants).

Enumérations

Type ENUM

Chaque valeur de l'énumération est associée à un indice commençant à 1. Ainsi il est possible

De retrouver la position d'une valeur au sein de son énumération comme l'illustre l'exemple

Table de données colonne ENUM

UnCursus

Num	nom	{diplome}
E1	F. Brouard	BTS
E2	F. Degrelle	Licence

ENUM

BTS, DUT, Licence INSA

Extraction

```
SELECT nom, diplôme, diplôme+0 FROM UnCursus ;
```

L'indice d'une valeur vide (colonne évaluée à (") ou " dans l'INSERT) est 0, celui d'une valeur NULL est NULL.

Type SET

Il est possible d'extraire des enregistrements en comparant des ensembles entre eux ou en testant l'appartenance d'éléments au sein d'une énumération SET.

L'exemple suivant illustre deux possibilités d'extraction.

Table et données colonne SET

Cursus

num	nom	{diplomes}
E1	F. Brouard	BTS, Licence
E2	F. Degrelle	Licence, INSA, DUT
E0	F. Peyrard	INSA, DUT

SET

BTS, DUT, Licence INSA

Extraction

```
SELECT nom, diplomes FROM Cursus WHERE FIND_IN_SET ('Licence',  
diplome)>0;
```

Il est possible d'écrire des extractions basées sur l'opérateur LIKE (exemple : «
SELECT ...

FROM Cursus WHERE diplomes LIKE ('%Licence%')»). Cela n'est cependant pas recommandé, car le mot 'Licence' peut être présent dans l'ensemble non pas en tant qu'élément, mais en tant que sous-chaîne d'un élément.

VII. Autres fonctions

D'autres fonctions n'appartenant pas à la classification précédente sont présentées dans le tableau suivant :

Fonction	Objectif	Exemple
DEFAULT (colonne)	Valeur par défaut d'une colonne (NULL si aucune).	
FORMAT (numerique,nb)	Formate un nombre arrondi à nb décimales de la manière suivante : '#, ###, ###. ##'	FORMAT (1234567.8901, 1) retourne « 1.234.567.9 ».
GREATEST (expression [, expression]...)	Retourne la plus grande des expressions.	GREATEST ('Raffarin', 'Chirac', 'X-Men') retourne 'X-Men'.
LEAST (expression [, expression]...)	Retourne la plus petite des expressions.	LEAST ('Villepin', 'Sarkozy', 'X-Men') retourne 'Sarkozy'
NULLIF(expr1,expr2)	Si expr1 = expr2 retourne NULL, sinon retourne expr1.	NULLIF('Raffarine', 'Pa
IFNULL(expr1,expr2)	Convertit expr1 susceptible d'être nul en une valeur réelle (expr2).	IFNULL(diplôme, 'Aucun !') retourne 'Aucun !' si diplôme est NULL.

VIII. Regroupements

Ici, on traite à la fois des regroupements de lignes (agrégats) et des fonctions de groupe (ou multi lignes). Nous étudierons les parties surlignées de l'instruction SELECT suivante :

```
SELECT [{DISTINCT | DISTINCTROW} | ALL] listeColonnes
FROM nomTable [WHERE condition]
[ clauseRegroupement ]
[ HAVING condition]
```


[clauseOrdonnancement]

[LIMIT [rangDépart,] nbLignes] ;

- listeColonnes: peut inclure des expressions (présentes dans la clause de regroupement) ou des fonctions de groupe.
- clauseRegroupement: GROUP BY (expression1[,expression2]...) permet de regrouper des lignes selon la valeur des expressions (colonnes, fonction, constante, calcul).
- HAVING condition: pour inclure ou exclure des lignes aux groupes (la condition ne peut faire intervenir que des expressions du GROUP BY).
- ClauseOrdonnancement : déjà étudié (ORDER BY dans la section Projection/Ordonnancement).

IX. Fonctions de groupe

Nous étudions dans cette section les fonctions usuelles. D'autres sont proposées pour manipuler des cubes (datawarehouse).

Le tableau suivant présente les principales fonctions. L'option DISTINCT évite les duplicatas

(pris en compte sinon par défaut). À l'exception de COUNT, toutes les fonctions ignorent les valeurs NULL (il faudra utiliser IFNULL pour contrer cet effet).

Utilisées sans GROUP BY, ces fonctions s'appliquent à la totalité ou à une seule partie d'une table comme le montrent les exemples suivants :

Fonction	Objectif
AVG([DISTINCT] expr)	Moyenne de expr(nombre).
COUNT({* [DISTINCT] expr}) lignes, expr pour les colonnes non nulles).	Nombre de lignes (* toutes les
GROUP_CONCAT(expr)	Composition d'un ensemble de valeurs.
MAX([DISTINCT] expr)	Maximum de expr(nombre, date, chaîne).

MIN([DISTINCT] expr)
chaîne).

Minimum de expr(nombre, date,

STDDEV(expr)

Écart type de expr(nombre).

SUM([DISTINCT] expr)

Somme de expr(nombre).

VARIANCE(expr)

Variance de expr(nombre).

Conclusion

Au terme de notre analyse porte sur les fonctions il en ressort que celles – ci permettent de manipuler les chaines de caractères, les types numériques, les bits et les dates. Elles permettent aussi la conversion, l'énumération et les regroupements. Est-ce alors possible de créer nous-même nos fonctions pour ne pas réécrire les mêmes requêtes.

Sources

1. www.google.cm
2. Christian Soutou, Apprenez SQL avec MySQL Avec 40 exercices corrigés