

FONCTIONS SQL, VUES, PROCÉDURES STOCKÉES dans MYSQL

DUT1 – semestre 2

Sommaire

Introduction	3
1. Les fonctions usuelles SQL dans MYSQL.....	3
1.1. Les fonctions pour les caractères	
1.2. Les fonctions numériques	
1.3. Les fonctions pour les dates	
2. Les utilisateurs et privilèges.....	11
2.1. La gestion des utilisateurs	
2.2. Les privilèges	
3. Les vues	14
4. Les bases du langage de programmation	16
5. La programmation avancée	18
5.1. Les procédures stockées	
5.2. Les curseurs	
5.3. Les exceptions	

Introduction

Ce cours présente les différents aspects du langage SQL de MySQL, en expliquant notamment comment faire évoluer et interroger des tables avec leurs éléments associés (fonctions usuelles, vues) et en décrivant les caractéristiques du langage procédural de MySQL à l'aide des éléments de base (structure d'un programme, variables, structures de contrôle).

1. Les fonctions usuelles SQL dans MYSQL

MySQL propose un grand nombre de fonctions qui s'appliquent dans les clauses SELECT ou WHERE d'une requête.

nomFonction(colonne1 | expression1 [,colonne2 | expression2 ...])

[...] : paramètre facultatif

1.1. Les fonctions pour les caractères

Fonction	Exemple	Résultat
ASCII(<i>car</i>)	ASCII('A')	65
CHAR(<i>n</i>)	CHAR(65)	'A'
CONCAT(<i>str1, str2, ...,strn</i>)	CONCAT ('La moyenne de ',nom, ' est ', moy)	'La moyenne de DUPOND est 14'
INSERT(<i>str1,pos,n,str2</i>)	INSERT('compxxxie Air France',5,3,'agn')	'compagnie Air France'
INSTR(<i>str,soustr</i>)	INSTR('compagnie Air France','Air')	11
LOWER(<i>str</i>)	LOWER('Air France')	'air france'
LOCATE(<i>str1,str2,[pos]</i>)	LOCATE('Air','compagnie Air France')	11
LENGTH(<i>str</i>)	LENGTH('compagnie Air France')	20
LEFT(<i>str,n</i>)	LEFT('compagnie Air France',9)	'compagnie'
LPAD(<i>str1,n,str2</i>)	LPAD('rien',10,'.-')	'.-.-.-rien' longueur 10 au total
REPLACE(<i>str1,str2,str3</i>)	REPLACE('compagnie Air France', 'Air France','Luxair')	'compagnie Luxair'
REVERSE(<i>str</i>)	REVERSE('rien')	'neir'
RIGHT(<i>str,n</i>)	RIGHT('compagnie Air France',6)	'France'
RPAD(<i>str1,n,str2</i>)	RPAD('Rien',10,'.-')	'Rien.-.-.-'
SOUNDEX(<i>str</i>)	extrait la phonétique anglaise d'une expression	
SUBSTR(<i>str,n,[t]</i>)	SUBSTR('compagnie Air France', 11,3)	'Air'
TRIM(<i>str</i>)	TRIM(' rien ')	'rien'
LTRIM(<i>str</i>)	LTRIM(' rien ')	'rien '
RTRIM(<i>str</i>)	RTRIM(' rien ')	' rien'

1.2. Les fonctions numériques

En plus des opérateurs arithmétiques : +, -, *, /, DIV et des fonctions de groupes : MIN, MAX, SUM, COUNT, AVG étudiées à la période précédente, d'autres fonctions sur les valeurs numériques existent.

ABS(<i>n</i>)	valeur absolue de <i>n</i>	
CEIL(<i>n</i>)	valeur entière supérieure de <i>n</i> : CEIL(15.3)	16
EXP(<i>n</i>)	e(2.71828163) à la puissance <i>n</i>	
LN(<i>n</i>), LOG(<i>n</i>)	Logarithme népérien de <i>n</i> , logarithme décimal de <i>n</i>	
MOD(<i>m</i>,<i>n</i>)	MOD(14,3)	2
POW(<i>m</i>,<i>n</i>)	POW(2,5)	32
RAND()	Flottant aléatoire (à 14 décimales) entre 0 et 1	
ROUND(<i>m</i>,<i>n</i>)	ROUND(15.3473,1) ROUND(15.3473,2)	15.3 15.35
SIGN(<i>n</i>)	Retourne le signe du nombre (-1, 0 ou 1)	
SQRT(<i>n</i>)	SQRT(16.6)	4.0743097574926725
TRUNCATE(<i>n</i>,<i>m</i>)	TRUNCATE(15.3473,2)	15.34

1.3. Les fonctions pour les dates

L'affichage et le traitement de données de type « date », « datetime », « time » en SQL imposent souvent d'utiliser des fonctions SQL appropriées.

Le format de stockage d'une date en SQL est défini par un ou plusieurs facteurs : la région d'utilisation des données, le système d'exploitation du serveur, les contraintes de l'administrateur de bases de données.

Exemple de format de stockage/affichage d'une date :

jj/mm/aaaa 23/03/2020	mm-jj-aa 03-23-20	jj-mm-aaaa 23-03-2020	aaaa-mm-jj 2020-03-23
-----------------------	-------------------	-----------------------	-----------------------

En conclusion, vous devez toujours contraindre vos requêtes à traiter une donnée au format date en utilisant les fonctions présentées ci-après.

ADDDATE(<i>date</i>,<i>n</i>)	ajoute <i>n</i> jours à une date
ADDTIME(<i>date1</i>, <i>date</i>)	ajoute les deux dates
DATE(<i>datetime</i>)	extraie une date à partir d'une date au format datetime
DATEDIFF(<i>date1</i>,<i>date2</i>)	nombre entier de jours entre deux dates
DATE_ADD (<i>date</i>, INTERVAL <i>expr type</i>)	ajoute un intervalle à une date et type indique comment interpréter le format de l'expression date_add(NOW(), INTERVAL 10 DAY) : ajoute 10 jours à la date courante
DATE_FORMAT (<i>date</i>, <i>format</i>)	présente la date en chaîne de caractères selon un format : DATE_FORMAT(NOW(), '%H:%i:%S')

DATE_SUB <i>(date, INTERVAL n type)</i>	soustrait un intervalle à une date et type indique comment interpréter le format de l'expression. date_sub(NOW(), INTERVAL 10 DAY) : soustrait 10 jours à la date courante
DAYNAME(date)	nom du jour en anglais
DAY(date)	numéro du jour dans le mois
DAYOFYEAR(date)	numéro du jour dans l'année
HOUR(time)	extraît l'heure
LAST_DAY(date)	dernier jour du mois d'une date
NOW()	date et heure systèmes (date et heure courantes)
MAKEDATE(année, njour)	construit une date à partir d'une année et d'un nombre de jours (> 0). Si le nombre de jours > 365 alors l'année s'incrémente automatiquement.
MAKETIME(hour, min, sec)	construit une heure
MINUTE(time)	extraît les minutes
MONTH(date)	retourne le numéro du mois (1 à 12)
MONTHNAME(date)	retourne le nom du mois en anglais
PERIOD_DIFF <i>(date1, date2)</i>	nombre de mois séparant deux dates au format 'yy-mm' ou 'yyyy-mm'
SECOND(time)	extraît les secondes
SEC_TO_TIME(secondes)	construit une heure au format hh:mm:ss
STR_TO_DATE(str, format)	convertit une chaîne de caractères avec l'expression de formatage en donnée SQL format datetime STR_TO_DATE('06:08:07','%H:%i:%s') <i>signifie que dans la chaîne '06:08:07', les deux premiers caractères représentent des heures, suivi de ":", suivi de deux caractères pour les minutes, suivi de ":", suivi des deux derniers caractères pour les secondes</i>
SUBDATE(date, n)	retranche n jours à une date
SUBTIME(date, time)	retranche un temps à une date
TIME(datetime)	extraît l'heure d'une date-heure
TIMEDIFF <i>(datetime1, datetime2)</i>	temps entre deux dates-heure
TIMESTAMP(date)	construit une estampille à partir d'une date
TIME_TO_SEC(time)	retourne le nombre de seconde d'un temps
TO_DAYS(date)	retourne le nombre de jours à partir d'une date
WEEKDAY(date)	numéro du jour (0 : lundi à 6 : dimanche) d'une date
WEEKOFYEAR(date)	numéro de la semaine dans l'année (1 à 53)

Principaux formats pour les fonctions DATE_FORMAT et STR_TO_DATE

%a	nom du jour de la semaine, en abrégé et en anglais (Sun..Sat)
%b	nom du mois, en abrégé et en anglais (Jan..Dec)
%c	mois, au format numérique (1..12)
%d	jour du mois, au format numérique (00..31)
%e	jour du mois, au format numérique (0..31)
%H	heure (00..23)
%i	minutes, au format numérique (00..59)
%j	jour de l'année (001..366)
%k	heure (0..23)
%m	mois, au format numérique (01..12)
%M	nom du mois (January..December)
%s	Secondes (00..59)
%T	Heures, au format 24 heures (hh:mm:ss)
%U	Numéro de la semaine (00..53), où Dimanche est le premier jour de la semaine
%u	Numéro de la semaine (00..53), où Lundi est le premier jour de la semaine
%W	Nom du jour de la semaine (Sunday..Saturday)
%w	Numéro du jour de la semaine (0=Sunday..6=Saturday)
%y	Année, au format numérique, sur 2 chiffres
%Y	Année, au format numérique, sur 4 chiffres

Paramètres d'intervalle pour les fonctions DATE_ADD et DATE_SUB

Type	SECOND	MINUTE	HOURL	DAY	WEEK	MONTH	YEAR
------	--------	--------	-------	-----	------	-------	------

Support TD et TP :

Base de données simplifiée pour la gestion d'un parc de stationnement de

NIVEAU (niv_id, niv_num, niv_lib, niv_nbplace)
TARIF_HORAIRE (trfh_id, trfh_duree, trfh_tarif)
TARIF_ABONNE (trfa_id, trfa_periode, trfa_tarif)
ABONNÉ (abo_id, abo_nom, abo_prn, abo_tel, abo_mel)
ABONNEMENT (abo_num, abo_dt, abo_debut, pers_id, niv_id, trfa_id)
ENTRÉE (ent_id, jour_in, heure_in, jour_out, heure_out, niv_id, abo_num)

Gestion des données pour l'année civile uniquement.

Renseignements sur les données :

- **niv_num** : ENTIER $\in [-10..10]$
- **niv_nbplace** : ENTIER > 0
- **trfh_duree** : ENTIER (tranche de stationnement) $\in [0..1440]$ [minutes
- **trfh_tarif** : 2 décimales (tarif en euros)
- **abo_dt** : DATE (date de signature de l'abonnement)
- **abo_debut** : ENTIER (numéro de mois du début de l'abonnement) : $\in [1..12]$
- **trfa_periode** : ENTIER (nombre de mois) $\in [1..12]$
- **trfa_tarif** : ENTIER (tarif en euros)
- **jour_in** et **jour_out** : ENTIER $\in [1..366]$ tel que $\text{jour_in} \leq \text{jour_out}$
- **heure_in** et **heure_out** $\in [0..1440]$ [minutes

Les abonnements sont valables uniquement pour l'année en cours et débutent le 1^{er} jour du mois « **abo_debut** ». La période choisie pour un abonnement ne peut pas dépasser l'année en cours : $\text{abo_debut} + \text{période choisie} \leq 13$

Contenu des tables

NIVEAU

tel que niv_num : ENTIER $\in [-10..10]$

<u>niv_id</u>	<u>niv_num</u>	<u>niv_lib</u>	<u>niv_nbplace</u>
1	-2	vert	350
2	-1	rouge	350
3	0	jaune	350

TARIF_HORAIRE

*trfh_duree : ENTIER (tranche de stationnement)
 $\in [1..1440]$ minutes*

trfh_tarif : 2 décimales maxi (tarif en euros)

<u>trfh_id</u>	<u>trfh_duree</u>	<u>trfh_tarif</u>
1	20	0
2	40	0,8
3	60	1,5
4	90	2,5
5	120	3
6	180	4,3
7	240	5,5
8	300	6,6
9	360	7,4
10	420	8,1
11	480	8,6
12	1440	9

TARIF_ABONNE

*trfa_pperiode : ENTIER (nombre de mois)
 $\in [1..12]$*

trfa_tarif : ENTIER (tarif en euros)

<u>trfa_id</u>	<u>trfa_pperiode</u>	<u>trfa_tarif</u>
1	1	130
2	2	250
3	3	360
4	4	460
5	5	560
6	6	650
7	7	750
8	8	850
9	9	940
10	10	1030
11	11	1120
12	12	1200

ABONNÉ

<u>abo_id</u>	<u>abo_nom</u>	<u>abo_prn</u>	<u>abo_tel</u>	<u>abo_mel</u>
1	DUPOND	Paul	0383452565	dupond.paul@gmail.com
2	DITRA	Fernando	0645897122	fergmail.ditra@free.fr
3	STERN	Sylviane	0387112543	
4	BOZZO	Casimir	0387256895	bozcasi@gmail.com
5	BOUZOUBAH	Sydney	0752364771	sydney.bouzoubah@univ-lorraine.fr

ENTRÉE

jour_in et jour_out : ENTIER $\in [1..366]$ tel que jour_in \leq jour_out

heure_in et heure_out : ENTIER $\in [0..1440]$ [minutes]

216437 lignes

<u>ent_id</u>	<u>jour_in</u>	<u>heure_in</u>	<u>jour_out</u>	<u>heure_out</u>	<u>niv_id</u>	<u>abo_num</u>
1	1	534	1	770	2	
2	1	760	1	1280	3	
3	1	1039	1	1401	3	
4	1	897	1	1312	1	
5	1	1219	1	1287	3	
6	1	1285	1	1348	1	
7	1	486	1	1376	1	
8	1	970	1	1186	3	
9	1	798	1	1110	1	
10	1	221	1	951	3	
11	1	1378	1	1432	3	
12	1	1243	1	1371	2	
13	1	652	1	1313	3	
14	1	1183	1	1222	1	
...	

ABONNEMENT

abo_dt : DATE (date de signature de l'abonnement)

abo_debut : ENTIER (numéro de mois du début de l'abonnement) : $\in [1..12]$

<u>abo_num</u>	<u>abo_dt</u>	<u>abo_debut</u>	<u>abo_id</u>	<u>niv_id</u>	<u>trfa_id</u>
BOUZSY01	2013-01-05	1	5	1	6
BOZZCA01	2013-02-12	4	4	1	6
BOZZCA02	2013-09-16	10	4	1	3
DITRFE01	2013-01-01	1	2	2	12
DUPOPA01	2013-01-10	1	1	2	5
DUPOPA02	2013-03-25	4	1	2	9
DUPOPA03	2013-04-07	5	1	2	2
DUPOPA04	2013-07-01	7	1	2	6
STERSY01	2013-03-11	3	3	3	10

Les abonnements

- valables uniquement pour l'année en cours
- débutent le 1^{er} jour du mois « abo_debut ».
- abo_debut + période choisie ≤ 13