

BASE DE DONNEES ET SYSTEMES DE GESTION DE BASE DE DONNEES (SGBD)

CHAPITRE V : FONCTIONS STATISTIQUES, CHAINE, DATES ET DE CONVERSION

SOMMAIRE

LES OPERATEURS ARITHMETIQUES : +, -, *, /	3
LES CONVERSIONS	6

FONCTIONS STATISTIQUES (PREDEFINIES)

Certaines fonctions arithmétiques calculent une valeur unique au départ des diverses valeurs d'une colonne. Ces fonctions contribuent à augmenter la puissance des requêtes en SQL et elles sont reprises dans le tableau ci-dessous, associées à leur définition :

FONCTION	ROLE
AVG	Produit la moyenne des valeurs de son argument.
MAX	Détermine le maximum des valeurs de son argument, qui est le nom d'une colonne.
MIN	Détermine le minimum des valeurs de son argument.
SUM	Calcule la somme des valeurs d'une colonne, son argument.
COUNT	Compte le nombre d'éléments d'une colonne.

Dans les instructions SQL, ces fonctions apparaissent sous la forme suivante :

AVG ... MAX ([DISTINCT] nom_col expression ALL expression) ... MIN SUM ... COUNT (*) ... COUNT ([DISTINCT] nom_col) ...

Les fonctions statistiques peuvent être appliquées aux valeurs d'une table complète ou à un sous-ensemble des valeurs de celui-ci.

Le mot DISTINCT élimine les doublets. Si ALL est spécifié, les doublets ne sont pas éliminés ; il en est de même si ALL et DISTINCT ne sont ni l'un ni l'autre précisés.

Il est inutile de préciser que les arguments de SUM et AVG doivent être des nombres. Les colonnes de type LONG VARCHAR ne peuvent apparaître explicitement comme arguments d'une fonction statistique, sauf pour COUNT (*).

D'autre part, les valeurs NULL sont ignorées dans le calcul des fonctions, excepté pour COUNT (*) où chaque ligne est comptée, y compris celles qui ont des valeurs NULL.

Exemples :

```
SELECT AVG(auto) AS Moyenne FROM avions ;
```

Ou

```
SELECT SUM(auto)/COUNT(auto) AS Moyenne FROM avions ;
```

```

SELECT MIN(nbre_passag) AS MINI,
        MAX(nbre_passag) AS MAXI FROM avions ;

SELECT COUNT(DISTINCT passagers.localite) FROM passagers ;

SELECT SUM(nbre_passag) AS Total_Passagers FROM avions ;

```

LES OPERATEURS ARITHMETIQUES : +, -, *, /

Ces opérateurs contribuent à la formation des EXPRESSIONS qui apparaissent dans les clauses SELECT et WHERE, et dont la syntaxe est :

```

... [ + | - ] < fonction | (expression) | constante | nom_col >
[ < + | - | * | / > < fonction | (expression) |
constante | nom_col > ... ]

```

Exemple

Cherchons la vitesse de croisière des avions en m/s ainsi que le temps maximum possible de vol :

```

SELECT type, marque, modele,
        (vit_croi * 1000 / 3600) AS VC,
        (auto / vit_croi) AS TM FROM avions ;

```

OPERATIONS SUR CHAINES DE CARACTERES

De la même façon que les fonctions arithmétiques agissent sur des nombres, un certain nombre de fonctions chaînes de caractères agissent sur des données alphanumériques (de types CHAR ou VARCHAR).

Remarque

Les fonctions arithmétiques, chaînes de caractères et dates varient d'une version à l'autre de SQL, tout comme le nombre de ces fonctions. Il n'est cependant pas nécessaire de prendre en considération toutes les variantes pour comprendre la philosophie du langage. Si certaines de ces fonctions sont utilisées dans l'un ou l'autre exemple, une définition succincte se trouvera annexée.

- La concaténation de deux chaînes de caractères est réalisée par le symbole " || " placé entre les deux chaînes :

Chaîne 1 || chaîne 2.

Exemple

Si la colonne LOCALITE contient la valeur 'Namur', l'opération :

'Habite à' || localité

retournera la chaîne :

'Habite à Namur'.

En MySQL :

Exemple 1 :

```
SELECT CONCAT('Localité : ', localite), nom FROM passagers ;
```

Exemple 2 :

```
SELECT CONCAT(nom,CONCAT(' ',prenom))
```

```
AS NOM_PRENOMS
```

```
FROM passagers ;
```

Ou

```
SELECT CONCAT(nom,' ',prenom) AS NOM_PRENOMS
```

```
FROM passagers ;
```

- Il est parfois intéressant de connaître la longueur d'une chaîne de caractères. C'est possible grâce à la fonction LENGTH (chaîne) qui retourne toujours un nombre entier.

Ainsi, LENGTH (localite) donnera le nombre 5 (pour 'Namur').

En MySQL :

```
SELECT (LENGTH (localite)) AS LongLocal, localite, nom FROM passagers;
```

- La fonction SUBSTRING (string, départ, long.), notée SUBSTR dans certaines versions, fournit une portion de la chaîne "string" ; cette portion démarre à la position "départ" de la chaîne initiale et sa longueur est "long".

Ainsi, SUBSTRING('Monsieur X',4,5) retourne 'sieur' et
SUBSTRING('Dr WHO',4,20) donnera 'WHO'.

En MySQL :

```
SELECT SUBSTRING(localite,2,4) AS localCoupe,  
localite, nom FROM passagers;
```

OPERATIONS RELATIVES AUX DATES ET OPERATIONS DE CONVERSION

Les fonctions « date » soit fournissent une information sur une colonne de type DATETIME, soit génèrent un résultat de ce type. Ainsi, DAY(date) retourne un nombre compris entre 1 et 31 qui représente le jour du mois. Si, par exemple, date_dep contient '02/27/1989', alors DAY (date_dep) donnera 27.

Exemple :

```
SELECT DAY(date_dep), date_dep from vols;
```

De même, YEARNUM (date) fournit un nombre de quatre chiffres représentant l'année de "date". Si donc date_dep contient '12/15/1988', alors YEARNUM (date_dep) retourne 1988.

En MySQL :

```
SELECT YEAR(date_dep), date_dep from vols;
```

En MySQL :

```
SELECT MONTH(date_dep), date_dep from vols;
```

LES CONVERSIONS

Des opérations dans SQL permettent d'effectuer des conversions entre les divers types de données :

La conversion d'un caractère en une date se fait par la fonction TO_DATE (char). Par exemple, TO_DATE('12/15/1972') sortira la date : 15-dec-1972.

La conversion inverse, d'une date en une chaîne de caractères, s'effectue par la fonction TO_CHAR (date). Ainsi, TO_CHAR(25-jan/1985) donnera la chaîne : '25 jan-1985'. Il est ici possible de préciser sous quelle forme on souhaite voir s'afficher la date : si l'on écrit TO_CHAR(date_dep, 'mm/jj/aa'), la chaîne résultante sera : '01/25/1985'.

En MySQL : On utilise la fonction CAST(expression, Type)

Exemple :

```
SELECT nom, vols.no_vol, vols.date_dep
FROM passagers, reservations, vols
WHERE passagers.no_pass = reservations.no_pass
AND vols.date_dep > CAST('1988-04-01' AS DATE);
```

La conversion entre valeurs numériques et caractères est réalisée par les fonctions STRING et VALUE :

- STRING (nombre, dec) convertit "nombre" en une chaîne, en ne gardant que "dec" positions décimales, le système se chargeant d'arrondir si nécessaire.

Exemple

STRING (215.268,2) renvoie la chaîne '215.27'.

En MySQL :

Exemple 1 :

```
SELECT CONCAT ('MySQL CAST Exemple # ', CAST(2 AS CHAR));
```

Donne comme résultat :

```
MySQL CAST Exemple # 2
```

Exemple 2 :

SELECT type, modele, marque,

CONCAT('VITESSE = ',CAST(vit_croi AS CHAR)) AS VIT_CROI

FROM avions ;

- VALUE (string) convertit en un nombre la chaîne “string” obligatoirement composée de digits (0-9) et éventuellement du point décimal.

Exemple

VALUE ('215217') retourne 215217 qui sera par la suite interprété comme un nombre.

En MySQL :

Exemple :

SELECT (4 * CAST('2' AS UNSIGNED)) ;

Donne comme résultat : 8

L’annexion aux fonctions de recherche en SQL d’opérations sur les divers types de données conduit à accroître la puissance de ce langage. En outre, la possibilité de convertir certains types en d’autres élargit considérablement la gamme des manipulations réalisables sur les données.