

Le langage SQL

Structured Query Language

Langage de requête structurée

T. Alran – N. Bouafia – Y. Bouiche – H. Fouah – J. Pons

Présentation du langage SQL (1)

- Langage complet de gestion de bases de données relationnelles
- Aujourd'hui SQL est devenu un standard (normalisé par l'ANSI depuis 1986) disponible sur presque tous les SGBD relationnels (ORACLE, INFORMIX, Access, SQL Server, etc.).

Présentation du langage SQL (2)

Le SQL est à la fois:

- ✓ Un langage d'Interrogation de bases de données (LID)
- ✓ Un langage de mise à jour - Manipulation (LMD)
- ✓ Un langage de Définition des données (LDD)
- ✓ Un langage de Contrôle d'accès aux données (LCD)

Le Langage d'interrogation de données (LID)

La clause **SELECT** permet de réaliser : la projection et la sélection.

Syntaxe générale :

SELECT [DISTINCT] < noms des champs / * >

FROM < noms des tables >

[WHERE < conditions de sélection >]

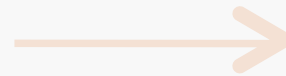
[GROUP BY < noms des champs >]

[HAVING < conditions de having >]

[ORDER BY < noms des champs > [ASC / DESC]

Tables et données

Structure de la table Client



client
<u>idclient</u>
nomclient
prenomclient
villeclient
ageclient

Données des différents clients



idclient	nomclient	prenomclient	villeclient	ageclient
011	Dubois	Patrick	Lyon	46
012	Durand	Stéphanie	Toulon	61
013	Girault	Philippe	Marseille	37
014	Dupont	Antoine	Lyon	29
015	Fabian	Alain	Nice	54

LID : La Projection

La Projection (SELECT ... FROM ...)

L'opération de projection consiste à choisir le nom des attributs de la (ou des) table(s) que l'on souhaite voir apparaître dans la réponse.

Si l'on veut afficher toutes les colonnes, il faut utiliser « * ». Les noms des attributs sont données dans la clause **SELECT**.

Si l'on veut éliminer les doublons, il faut utiliser « DISTINCT ». Il est utilisé pour chaque attribut non clé.

Si l'on souhaite créer un alias pour une meilleure lisibilité (attributs et tables), il faut utiliser « AS ».

Exemple : **SELECT** idclient **AS** identifiant **FROM** client **AS** c

LID : La Projection

Exemples :

Requête 1 : **SELECT** villeclient **FROM** client

Requête 2 : **SELECT * FROM** client

Requête 3 : **SELECT DISTINCT** villeclient **FROM** client

Résultat de la requête 1 :

villeclient
Lyon
Toulon
Marseille
Lyon
Nice

Résultat de la requête 2 :

idclient	nomclient	prenomclient	villeclient	ageclient
011	Dubois	Patrick	Lyon	46
012	Durand	Stéphanie	Toulon	61
013	Girault	Philippe	Marseille	37
014	Dupont	Antoine	Lyon	29
015	Fabian	Alain	Nice	54

Résultat de la requête 3 :

villeclient
Lyon
Toulon
Marseille
Nice

LID : La Sélection

La Sélection (SELECT ...FROM...WHERE...)

L'opération de sélection consiste à sélectionner des lignes d'une (ou plusieurs) table(s) qui satisfont certaines conditions.

Les conditions sont exprimées dans la clause WHERE.

La sélection peut se faire sur 1..N critères, à l'aide :

- ✓ Des opérateurs de comparaison (<, >, =, <=, >=, <>)
- ✓ Des opérateurs logiques (AND, OR, NOT)
- ✓ Des opérateurs arithmétiques (+, -, *, /)
- ✓ Des opérateurs IN (Appartenance), BETWEEN (Intervalle, bornes incluses), LIKE (Comparaison de chaînes), NULL.

LID : La Sélection

Exemple:

Donnez les caractéristiques du client ayant comme identifiant 012.

```
SELECT * FROM client WHERE idclient=012
```

Résultat de la requête:

idclient	nomclient	prenomclient	villeclient	ageclient
012	Durand	Stéphanie	Toulon	61

LID : La Projection et la Sélection

Projection et Sélection (SELECT ...FROM...WHERE...)

Les opérations de projection et de sélection peuvent évidemment être utilisées dans la même requête SQL.

Exemple:

Donnez les noms et prénoms des clients client âgés entre 40 et 60 ans.

```
SELECT nomclient FROM client WHERE ageclient BETWEEN 40 AND 60
```

Résultat de la requête:

nomclient
Dubois
Fabian

Les opérateurs

✓ Les opérateurs et fonctions arithmétiques (dans les clauses SELECT et WHERE)

Les opérateurs les plus utilisés sont:

ROUND : (nbre décimal, nbre de décimales)

ABS (nombre): Valeur absolue d'un nombre

SQR (nombre): Racine carrée d'un nombre (doit être >0)

✓ Les fonctions d'agrégation (dans les clauses SELECT et HAVING)

AVG : calcule la moyenne des valeurs d'un champ numérique

COUNT : dénombre les occurrences d'un champ

MAX, MIN : donne la valeur max, min d'un champ

SUM : calcule la somme des valeurs d'un champ numérique

La clause GROUP BY

La clause GROUP BY est nécessaire dès que l'on utilise des fonctions de calculs statistiques avec des données brutes.

Cette clause regroupe les occurrences ayant même valeur pour l'attribut(s) spécifié(s) dans la clause Group By, et renvoie une seule ligne par groupe.

Tous les attributs du SELECT et du GROUP BY doivent être identiques (aux fonctions agrégats près).

GROUP BY trie par ordre croissant de ses attributs

Exemple:

Donner, pour chaque ville de client, le nombre de client y résidant.

```
SELECT villeclient, count(idclient)
FROM client
GROUP BY villeclient
```

La clause HAVING

Elle ne s'utilise qu'avec la clause GROUP BY, pour faire des sélections sur les calculs de regroupement.

Exemple:

Donner, pour chaque ville de client, le nombre de client y résidant si celui-ci est supérieur à 10.

```
SELECT villeclient, count(idclient)
FROM client
GROUP BY villeclient HAVING count(idclient)>10
```

La clause ORDER BY

Elle permet de trier les données de la table résultat (sans aucune altération de la table physique).

Les attributs du ORDER BY sont inclus dans ceux du SELECT.

Les valeurs NULL se retrouvent en tête du tri.

Exemple:

```
SELECT DISTINCT villeclient  
FROM client  
ORDER BY villeclient ASC/DESC
```

LDD : Définitions des données

Le LDD du langage SQL englobe les instructions CREATE, ALTER, DROP, permettant d'agir sur les objets de la BD, à savoir les tables, index, vues, synonymes.

Toutes ces actions peuvent se faire en mode interactif, ou intégrées dans une application.

Elles sont toutefois toujours soumises aux autorisations dont bénéficie leur utilisateur.

CREATE

La clause CREATE permet de créer une table. Il faudra alors fournir les noms et types des attributs de cette table.

Syntaxe :

CREATE TABLE <nom de la table>
(<Nom attribut> <type de l'attribut> ...)

Les types d'attributs sous Access :

Nombre entiers : *SHORT/LONG*

Nombres décimaux : *FLOAT*

Chaines de caractères : *CHAR(x)* (x lettres), *VARCHAR(x)* (jusqu'à x lettres), *TEXT* (255 lettres)

Bouléens : *LOGICAL*

Données temporelles : *DATE, TIME*

LDD : Définitions des données

DROP

La clause DROP supprime

- Tous les tuples de la table (son contenu)
- La définition de la structure de la table (le contenant) qui est stockée dans le dictionnaire des objets de la BD. De plus, elle libère l'espace mémoire.

Syntaxe :

DROP TABLE <nom de la table>

ALTER

Les modifications possibles sont les suivantes :

- Ajout de nouveaux attributs (en fin de table),
- Suppression d'attributs,
- Modification d'attributs (nom, type, propriété).

Syntaxe :

- ALTER TABLE <nom de la table>
ADD <définition attribut>
- ALTER TABLE <nom de la table>
DROP COLUMN <nom attribut>
- ALTER TABLE <nom de la table>
MODIFY <définition attribut>
↳ Impossible sous Access

MAINTENANT...

*...PLACE AUX
EXERCICES*