

Attention !

La commande suivante supprime **toutes les lignes** d'une table :

`DELETE FROM nom_table ;`

[Voir site SQL.sh pour + d'informations.](#)

SQL : fiche pense-bête



SQL est l'abréviation de **Structured Query Language**.

C'est un SGBD, **Système de Gestion de Base de Données**.

Un SGBD est un logiciel qui permet de stocker des informations dans une base de données.

Catégories de SGBD :

SGBD Relationnel : C'est le premier SGBD créé dans les années 1970.

Les données y sont représentées sous forme de tables à 2 dimensions (lignes et colonnes de valeurs).

La manipulation est automatique, l'insertion ou l'interrogation des données s'effectue sans mention de leur représentation ni de leur stockage sur disque.

Ex : (Microsoft SQL Server, MySQL, Oracle, PostgreSQL, Informix

...

De nouvelles catégories de SGBD commencent à prendre de l'ampleur : les **SGBD In-memory** qui utilisent principalement la mémoire pour stocker les données au lieu de les stocker sur disques (AeroSpike, VoltDB ...).

Les **SGBD NoSQL** autorisent un schéma flexible dans lequel il n'est pas nécessaire de définir chaque donnée pour chaque entité. Donc elle facilite la manipulation de grandes quantités de données qui ne sont pas toutes renseignées. Cette souplesse est au détriment de la souplesse d'indexation et de la facilité d'interrogation (Cassandra, Redis, MongoDB...).

SQL est un langage de programmation déclaratif : Elle consiste à créer des applications sur la base de composants logiciels indépendants du contexte et ne comportant aucun état interne. Autrement dit, l'appel d'un de ces composants avec les mêmes arguments produit exactement le même résultat, quel que soit le moment et le contexte de l'appel. Contrairement aux langages de programmation impératif (C ou Java) qui décrivent les opérations en séquences d'instructions exécutées par l'ordinateur pour modifier l'état du programme. Ce type de programmation est le plus répandu parmi l'ensemble des langages de programmation existants, et se différencie de la programmation déclarative (dont la programmation logique ou encore la programmation fonctionnelle sont des sous-ensembles).

Vocabulaire :

Table ou relation :

Ensemble d'informations de même nature regroupées dans un tableau à 2 dimensions.

Ligne de tables ou tuple ou enregistrement :

Ensemble d'informations appartenant à un élément d'une relation.

Colonne ou attribut ou rubrique :

Nom de la partie élémentaire d'une relation.

Degré :

Nombre d'attributs d'une relation (nombre de colonnes d'une table).

Cardinalité :

Nombre de lignes dans une table.

Clé primaire ou Primary Key (PK).

Colonnes ou ensemble de colonnes permettant de reconnaître de manière unique 1 ligne.

Clé candidate :

Colonnes ou ensemble de colonnes permettant de reconnaître de manière unique 1 ligne mais qui n'a pas été retenu comme clé identifiée.

Composant :

Colonne faisant partie d'une clé.

Domaine :

Ensemble de valeurs que peut prendre un type de colonne (Data Types ou type de données).

Vue :

Présentation différente des informations appartenant à une ou plusieurs tables. La vue peut être le résultat de n'importe quelle opération relationnelle (projection, sélection, jointure ...).

Clé étrangère ou Foreign Key (FK) :

Colonnes qui appartiennent à 2 ou plusieurs tables et qui pourront servir à établir un lien entre ces différentes tables.

L'acronyme CRUD :

Il désigne la majorité des opérations implémentées dans les bases de données relationnelles.

Create Read Update Delete.

Operation	SQL	HTTP
Create	INSERT	POST (en)
Read (Retrieve)	SELECT	GET (en)
Update (Modify)	UPDATE	PUT (en)
Delete (Destroy)	DELETE	DELETE (en)

Utilisation du langage SQL:

En interactif : avec un outil permettant de transmettre une requête SQL via une invite de commande. Ces outils utilisent des interfaces logicielles telles qu'ODBC, JDBC ou autre.

Par programmation : avec des requêtes dites embedded SQL : les requêtes SQL sont incorporées dans le code source d'un programme écrit dans un autre langage (COBOL, C FORTRAN...).

Par procédures stockées : des procédures écrites en langage SQL sont enregistrées dans la base de données en vue d'être exécutée par le SGBD. Cette technique est utilisée pour les trigger, procédures stockées déclenchées automatiquement sur modification du contenu de la base de données.

Groupe de commandes SQL:

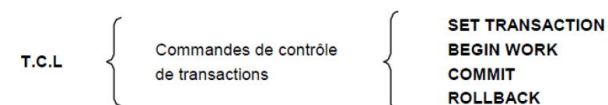
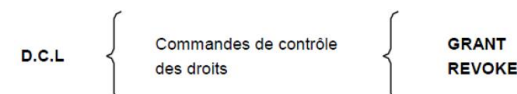
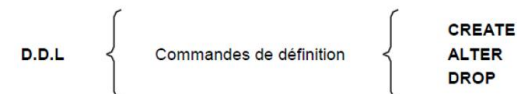
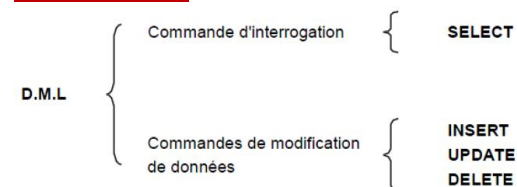
Le **DML** (Data Manipulation Language), pour le traitement des données.

Le **DDL** (Data Description Language), pour la description des données.

Le **DCL** (Data Control Language), pour le contrôle d'accès aux données.

Le **TCL** (Transaction Control Language), pour la gestion des transactions.

Les ordres SQL :



Commandes SQL :

Extraction de toutes les lignes d'une table : **SELECT**

SELECT * FROM nom_table ;

Extraction de plusieurs colonnes depuis toutes les lignes d'une table :

SELECT colonne1, colonne2 FROM nom_table ;

ALL (implicite) affiche toutes les lignes y compris les doublons.

DISTINCT permet de supprimer les doublons.

SELECT DISTINCT job FROM employee ;

WHERE d'ajouter une condition de recherche.

SELECT * FROM employee WHERE salaire > 3000 ;

NOT permet d'inverser une condition.

AND et **OR** permettent de chaîner les conditions de recherche.

IN permet d'ajouter une condition de recherche comprise dans une suite de valeurs

SELECT * FROM employee WHERE manager IN (7289,7564) ;

BETWEEN recherche comprise entre 2 valeurs

SELECT * FROM employee
WHERE salaire BETWEEN 1500 AND 2500 ;

LIKE recherche d'une valeur qui ressemble à une chaîne de caractères.

SELECT * FROM employee WHERE name LIKE '_I%' ;

SELECT * FROM employee WHERE name LIKE 'A%' ;

'_I%' comporte un I en 2nde position

'A%' commence par un A

IS NULL : n'est pas renseigné.

IS NOT NULL : est renseigné.

AS : alias d'une colonne ou résultat d'un calcul

SELECT name AS nom, salaire+prime AS revenu FROM emp ;

ORDER BY : permet de trier le résultat.

SELECT * FROM employee ORDER BY name ;

ASC : sens de tri ascendant

DESC : sens de tri descendant

Insertion d'une ligne : **INSERT**

INSERT INTO employee VALUES ('BROWN', 'JOHN', 2500) ;

Ou

INSERT INTO employee (fname, lname, sal)
VALUES ('BROWN', 'JOHN', 2500) ;

Mise à jour d'une ligne : **UPDATE**

UPDATE nom_table SET colonne = 'valeur' ;

Suppression d'une ligne : **DELETE**

DELETE FROM nom_table WHERE condition