Langage SQL – l’essentiel des ordres SQL (pour MySQL)

Les clauses facultatives sont entre crochets [ … ]; les élémént de "listes de" sont séparés par ","

# INTERROGER UNE BASE DE DONNEES – Data Query Language

|  |  |
| --- | --- |
| **Syntaxe générale** | **SELECT… FROM… WHERE…** **GROUP BY… HAVING…** **ORDER BY…;** |

|  |  |
| --- | --- |
| Projection | **SELECT** [DISTINCT] expr1 [AS alias1], expr2 [AS alias2],…**FROM** table1 [aliast1] |
| Sélection / Restriction lignesop. logiques | **WHERE condition**où **condition** utilise les opérateurs de comparaisons classiques et d'autres spécifiques : expr1 **operateur** expr2 **opérateur** parmi **: = ,** !**= , < , > , <= , >=**  |
|  expr1 **[NOT] BETWEEN** expr2 **AND** expr3 |
|  expr1 **[NOT] LIKE** chaîne1 chaine1 contient des caractères joker % et \_ |
|  expr1 **[NOT] IN** (expr2, expr3, …) |
|  expr1 **IS [NOT] NULL** |
|  condition1 **AND** condition2 / condition1 **OR** condition 2 / **NOT** condition |

*où « expr1 » est un nom de colonne, une expression calculée, une fonction appliquée à un colonne ou à une expression, ou encore '\*' qui signifie toutes les colonnes des tables concernées (= celles de la clause FROM)*

|  |  |
| --- | --- |
| **Fonctions liées aux dates et autres fonctions** | **YEAR**(une date) |
| **MONTH**(une date) |
| **DAY**(une date) |
| WEEKDAY(une date) |
| NOW(), CURRENT\_DATE() |
| DATE\_ADD(uneDate, INTERVAL n DAY / MONTH / YEAR) |
| ROUND(calcul, nbreDecimales) |

|  |  |
| --- | --- |
| **Jointure interne (d’égalité)****champ1 et champ2 : clef étrangère ou clef primaire** | SELECT liste d'expressions FROM table1 **INNER JOIN** table2 **ON table1.champ1 = table2.champ1 … ;** |
| SELECT liste d'expressions FROM table1 **INNER JOIN** table2 **USING (champ1)… ;** |
| SELECT liste d'expressions FROM table1, table2 **WHERE table1.champ1 = table2.champ1 … ;** |

|  |  |
| --- | --- |
| **Jointure externe****champ1 et champ2 : clef étrangère ou clef primaire** | SELECT liste d'expressions FROM table1 **LEFT / RIGHT / FULL** **OUTER JOIN** table2 **ON table1.champ1 = table2.champ1 … ;** |
| SELECT liste d'expressions FROM table1 **LEFT / RIGHT / FULL** **OUTER JOIN** table2 **USING (champ1)… ;** |

|  |  |
| --- | --- |
| **Produit cartésien** | SELECT liste d'expressions FROM table1 **CROSS JOIN** table2 … ; |
| SELECT liste d'expressions FROM table1, table2 … ; |

|  |  |
| --- | --- |
| Agrégat***Si expr1 🡺 utiliser GROUP BY*** | SELECT [liste d'expressions,] **SUM**(expr2) [AS alias1] … |
| SELECT [liste d'expressions,] **MAX**(expr2) [AS alias1] … |
| SELECT [liste d'expressions,] **MIN**(expr2) [AS alias1] … |
| SELECT [liste d'expressions,] **AVG**(expr2) [AS alias1] … |
| SELECT [liste d'expressions,] **COUNT**(\*) [AS alias1] … |
| SELECT [liste d'expressions,] **COUNT**(**DISTINCT** expr2) [AS alias1]) … |

 La clause **DISTINCT** est également applicable aux fonctions SUM et AVG.

|  |  |
| --- | --- |
| Regroupement et sélection après agrégat | **…GROUP BY** expr1 [, liste d'expressions] |
| **…HAVING** condition où condition est appliquée aux valeurs agrégées |

|  |  |
| --- | --- |
| Classement | **…ORDER BY** expr1 [ASC/DESC], expr2 [ASC/DESC],… |

|  |  |
| --- | --- |
| Intersection | **…WHERE** table1.champ1 **IN (**SELECT table2.champ1 FROM …**)** ; |
| Différence | **…WHERE** table1.champ1 **NOT IN** **(**SELECT table2.champ1 FROM …**)** ; |

|  |  |
| --- | --- |
| Union | Requête 1 **UNION** Requête 2 [ORDER BY …] ; |

# MODIFIER LES DONNEES D’UNE TABLE – Data Manipulation Language

|  |  |
| --- | --- |
| Suppression | **DELETE FROM** uneTable  [**WHERE** condition] ; |
| Insertion | **INSERT INTO** uneTable [(champ1, champ2,…)]  **VALUES** (val1, val2,…) ; |
| **INSERT INTO** table [(champ1, champ2,…)]  une requête Select |
| Mise à jour | **UPDATE** uneTable  **SET** champ1 = expr1, champ2 = expr2,…  [**WHERE** condition] |

# MODIFIER LA STRUCTURE DES TABLES (structure) – Data Definition Language

|  |  |
| --- | --- |
| Creation | **CREATE TABLE** uneTable ( liste des champs avec leur définition, contraintes de table) ; |
| nomChamp typeChamp  **[NOT NULL] [**contrainte de colonne**]** |
| *typeChamp parmi : INTEGER, VARCHAR(x), CHAR(n), DECIMAL(n,m), DATE* |
| **[ CONSTRAINT** nomContrainte ] **PRIMARY KEY (**liste decolonne-s**)** |
| **[ CONSTRAINT** nomContrainte ] **FOREIGN KEY (**liste decolonne-s**)** **REFERENCES** uneTable **(**liste decolonne-s**)** |
| Modification | **ALTER TABLE** uneTable  + type de mise à jour parmi :  |
|  **ADD COLUMN** definition d’un champ; |
|  **ADD** définition d’une contrainte |
|  **MODIFY** nouvelle définition d’un champ; |
|  **DROP** nom de champ/PRIMARY KEY/nom de contrainte |
| Suppression | **DROP**  uneTable ; |

# Autres objets de base de données (structure) – Data Definition Language

|  |  |
| --- | --- |
| Création d’une vue | **CREATE OR REPLACE VIEW** nomVue  **AS** une requête Select |
| Création d’une fonction | **CREATE FUNCTION** nomFonction (nom et type des paramètres) **RETURNS** type **DETERMINISTIC NO SQL** … RETURN … ;  |
| Utilisateurs et privilèges | **CREATE USER** nomUtilisateur **IDENTIFIED BY** motDePasse**;****GRANT** privilege [ON objetDeLaBase] TO utilisateur(s) [WITH GRANT OPTION];**REVOKE** privilege [ON objetDeLaBase] TO utilisateurs; |

où privilège est entre autres : SELECT, INSERT, UPDATE, DELETE, ALTER, ALL, etc.