



SQL

UE15

Informatique appliquée

Frédéric Pluquet

pluquetf@helha.be



v1 - slides originaux : Nathalie Buseyne

2018-2019

Le langage SQL

- 1. Qu'est-ce que le langage SQL ?
- 2. Sélection de données
- 3. Modification des données
- 4. Gestion de tables
- 5. Gestion de base de données
- 6. Procédures stockées
- 7. Les vues
- 8. Les fonctions

1. Qu'est-ce que le langage SQL ?

- Structured Query Language
- Langage de requêtes structuré
- Langage de définition (création , modification et suppression de table, de base de données) et de contrôle (sélection, insertion, modification, suppression de données) de données pour les bases de données relationnelles

2. Sélection des données

- 2.1. Expression des projections
- 2.2. Expression des sélections
- 2.3. Regroupement
- 2.4. Tri par ordre croissant ou décroissant
- 2.5. Jointures
- 2.6. Sous- requêtes
- 2.7. Opérations ensemblistes

2. Sélection des données

2.1. Expression des projections

- Une projection est une instruction permettant de sélectionner un ensemble de colonnes dans une table.
- Projection : restriction sur les colonnes
- Syntaxe :

SELECT [prédicats] <expressions des valeurs> FROM <nom de tables>

2. Sélection des données

2.2. Expression des sélections

- Une restriction consiste à sélectionner les lignes satisfaisant à une condition logique effectuée sur leurs attributs.
- Elles s'expriment à l'aide de la clause WHERE suivie de la condition.
- Sélection : restriction sur les lignes
- Cette condition utilise :
 - des opérateurs logiques : AND, OR, NOT
 - de comparaison : =, <>, <, >, <=, >=
 - d'intervalle : BETWEEN AND
 - de comparaison de texte : LIKE
 - de test de nullité : NULL
 - d'appartenance: IN
 - de non appartenance : NOT IN

2. Sélection des données

2.3. Regroupement

- La clause GROUP BY combine en un seul enregistrement les enregistrements comportant des valeurs identiques dans la liste des champs spécifiés
- La clause HAVING permet d'appliquer une restriction sur les groupes créés grâce à la clause GROUP BY

2. Sélection des données

2.4. Tri par ordre croissant ou décroissant

- Il est possible d'organiser les données en les triant dans l'ordre croissant ou dans l'ordre décroissant.
- Syntaxe :
ORDER BY champ1 [ASC|DESC], champ2 [ASC|DESC], ...

2. Sélection des données

2.5. Jointures

- La jointure est une opération d'association de tables, d'extraction d'informations de plusieurs tables afin de les présenter sous la forme d'une seule table
- Différents types de jointures:
 - Produit cartésien
 - Equijointures
 - Avec conditions
 - Jointures internes
 - Jointures externes

2.5. Jointures

Produit cartésien

- = Relation regroupant toutes les possibilités de combinaison des occurrences des tables ($R1 \times R2$)
- Jointure : produit cartésien suivie d'une ou plusieurs conditions
- Equijointure: jointure qui utilise l'opérateur =
- Thétajointure: jointure qui utilise les opérateurs :
 $<, >, <=, >=, <>$ ou \neq

2.5. Jointures

Jointures internes

- Elles permettent de fusionner les enregistrements de 2 tables lorsqu'un champ commun contient des valeurs identiques
- L'opération utilisée est **INNER JOIN**

2.5. Jointures

Jointures externes

- Elles permettent de retourner des lignes répondant à la condition mais aussi toutes celles de la première table ou toutes celles de la deuxième table ou des 2 qui ne vérifient pas cette condition
- Les opérations utilisées sont :
 - LEFT JOIN (jointure externe gauche)
 - RIGHT JOIN (jointure externe droite)
 - FULL OUTER JOIN (jointure externe complète)

2. Sélection des données

2.6. Les sous-requêtes

- Une sous-requête consiste à effectuer une requête à l'intérieur d'une autre requête
- Une sous-requête se compose d'une instruction SELECT imbriquée dans une instruction SELECT ou dans une autre sous-requête
- Les variantes de syntaxe sont:
 - a. <comparaison> (sous-requête)
 - b. <comparaison> [ANY|ALL|SOME] (sous-requête)
 - <expression> [NOT] IN (sous-requête)
 - [NOT] EXISTS (sous-requête)

2. Sélection des données

2.7. Les opérations ensemblistes

- Une opération ensembliste est une opération définie dans l'algèbre relationnelle: union, intersection et différence
- Elle est réalisée grâce aux opérateurs:
 - **UNION**
 - **INTERSECT**
 - **EXCEPT**
- Ces opérateurs s'utilisent entre 2 clauses SELECT

3. Modification des données

- 3.1. Insertion de données
- 3.2. Mise à jour de données
- 3.3. Suppression des données

3. Modification de données

3.1. Insertion de données

- L'instruction INSERT permet d'ajouter un ou plusieurs enregistrements dans une table.
- Syntaxe:
 1. Ajout de plusieurs enregistrements :
**INSERT INTO tabledestination (champ1,champ2,...) SELECT
tablesource.champ1,... FROM expression**
 2. Ajout d'un seul enregistrement:
**INSERT INTO tabledestination(champ1,champ2,...) VALUES
(valeur1,valeur2,...)**

3. Modification de données

3.2. Mise à jour de données

- L'instruction UPDATE permet de modifier les valeurs des champs d'une table spécifiée, selon des critères déterminés.
- Syntaxe: **UPDATE table SET expression WHERE condition**

3. Modification de données

3.3. Suppression des données

- L'instruction DELETE permet de supprimer des enregistrements dans une table.
- Syntaxe:

DELETE FROM table WHERE condition

4. Gestion de tables

- 4.1. Syntaxe
- 4.2. Déclaration d'attributs
- 4.3. Contraintes de tables
- 4.4. Modification de table
- 4.5. Renommer une table
- 4.6. Suppression de table

4. Gestion de tables

4.1. Syntaxe

- Pour créer une table on utilise la commande **CREATE TABLE**.

Sa syntaxe :

Create table < nom de table > ({ < element de table > }) ;

- Définition des éléments de table :
< élément de table > : = < définition attribut> / < contrainte de table >

4. Gestion de tables

4.2. Déclaration d'attributs

- Syntaxe de la déclaration d'un attribut :

< nom attribut > < type_données > [default <valeur>] [contrainte_attribut]

- Types de données: CHAR, VARCHAR, NUMERIC, INT, SMALLINT, FLOAT, DATE, TIME, TIMESTAMP
- Définir une valeur par défaut lorsqu'un champ de la base n'est pas renseigné grâce à la clause **DEFAULT**
- Liste des contraintes d'attributs: NOT NULL, UNIQUE, PRIMARY KEY, REFERENCES, CHECK

4. Gestion de tables

4.3. Contraintes de tables

- Liste des contraintes :
 - L'unicité : sa syntaxe est **UNIQUE / PRIMARY KEY**
(<nom_attribut1>, < nom_attribut2 >,...)
 - Les clés étrangères (CONSTRAINTS REFERENTIELLES) :
FOREIGN KEY (< nom attribut référençant >) REFERENCES
< nom de table référencée >
 - Contrainte générale: **CHECK (condition)**

4. Gestion de tables

4.4. Modification de table

- Syntaxe: **ALTER TABLE <table> <action>**
- Action:
 - * **ADD <nom_colonne> <type_colonne>**
 - * **DROP [COLUMN] <nom_colonne>**
 - * **ADD CONSTRAINT <contrainte>**
 - * **DROP CONSTRAINT <nom_contrainte>**

4. Gestion de tables

4.5. Renommer une table

- Syntaxe:

RENAME TABLE ancien_nom TO nouveau_nom

ALTER TABLE nom_ancien RENAME TO nouveau_nom

- Pas implémenté dans tous les systèmes de gestion de base de données relationnelles
- Pour SQL SERVER, il faut utiliser la procédure stockée système sp_rename: **exec sp_rename 'ancien_nom ','nouveaunom'**

4. Gestion de tables

4.6. Suppression de table

- Syntaxe:

Drop Table < nom de table >

- Syntaxe:

Truncate Table <nom table>

5. Gestion de base de données

- 5.1. Création de base de données
- 5.2. Modification de base de données
- 5.3. Suppression de base de données

5. Gestion de base de données

5.1. Création de base de données

- La création d'une base de données comprend l'affectation d'un nom à celle-ci, la déclaration de sa taille et de l'emplacement des fichiers de base de données
- **CREATE DATABASE nom_base [ON { [PRIMARY] (NAME = nom_fichier_logique, FILENAME = 'nom_fichier_os' [, SIZE = taille] [, MAXSIZE = taille_max] [, FILEGROWTH = incrément_croissance]) }] [LOG ON { (NAME = nom_fichier_logique, FILENAME = 'nom_fichier_os' [, SIZE = taille]) }]**

5. Gestion de base de données

5.2. Modification de base de données

- Fermer tous les accès à la base de données
- **ALTER DATABASE nom_base_données MODIFY
NAME=nouveau_nom**

5. Gestion de base de données

5.3. Suppression de base de données

- Fermer tous les accès à la base de données
- **DROP DATABASE nom_base_données**