MySQL

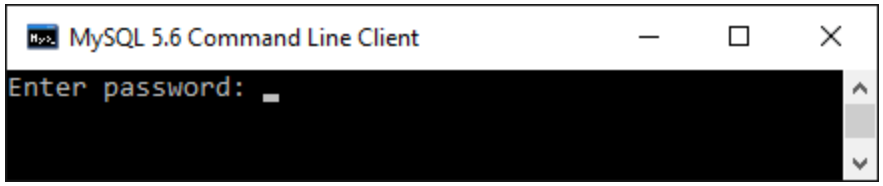
Présentation

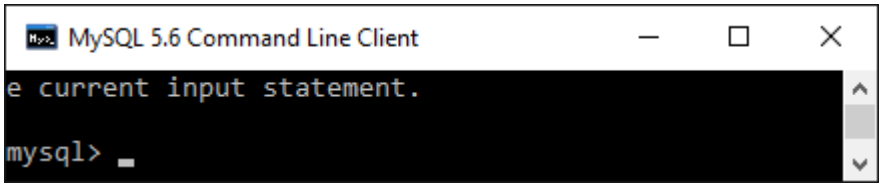
- C'est un SGBDR (Système de Gestion de Bases de Données Relationnelles) libre et gratuit
- Rapide, multithread, robuste et multi-utilisateurs
- Conçu par une société suédoise MySQL AB
- Fonctionne sur pratiquement toutes les plateformes
- Les tables peuvent être de différents types : MyISAM, MERGE, HEAP, ISAM, InnoDB, Berkley DB, MEMORY
- Accessible en utilisant la plupart des langages de programmation du marché

Administration

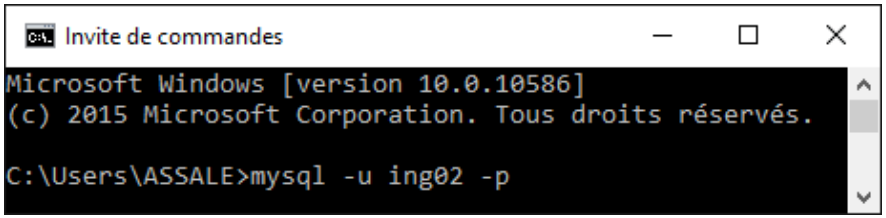
Connexion

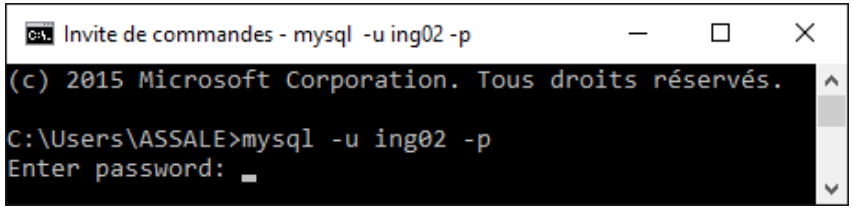
- En tant que root
  - ✓ On lance MySQL command Line et on rentre le mot de passe





- En tant qu'utilisateur
  - ✓ On lance une fenêtre shell ou DOS
  - ✓ On tape : ***mysql –u <nom_util> -p [–D <nom_base>]***





Administration

Création d'une base de données

- Syntaxe
  - ✓ create database <nom_base> / drop database <nom_base>
- Utilisation d'une base de données
  - ✓ use <nom_base>
- Visualisation des bases de données
  - ✓ show databases;
- Exemples :
  - ✓ *create database mabase;*
  - ✓ *use mabase;*
  - ✓ *create table …*

Administration

Création d'utilisateurs

- Syntaxe de création
  - ✓ create user <nom_utilisateur> [identified by [password] 'mot_de_passe'] [, <nom_utilisateur> [identified by [password] 'mot_de_passe']]…
  - ✓ ***<nom_utilisateur>*** de la forme ***<'nom'>[@'hôte']***
    - hôte adresse à partir de laquelle utilisateur va se connecter:
      - localhost : à partir de la machine du serveur
      - @IP : à partir de la machine dont l'adresse IP est spécifiée
      - %.nom_domaine : n'importe quelle machine du domaine
      - % : n'importe quel hôte valeur par défaut

Administration

Création d'utilisateurs

- Suppression d'un utilisateur
  - ✓ drop user <nom_utilisateur>
- Renommer un utilisateur :
  - ✓ rename user <ancien_nom> to <nouveau_nom> [, <ancien_nom> to <nouveau_nom>]…
- Exemples :
  - ✓ *create user assale identified by 'louis09', koffi identified by 'alexis',kone@'localhost';*
  - ✓ *rename user assale to adje;*
  - ✓ *drop user adje;*

Administration

Création d'utilisateurs

- Assignation/Changement de mot de passe
  - ✓ set password = password('mot_de_passe')
  - ✓ set password for <nom_utilisateur>= password('mot_de_passe')

  1ère syntaxe modifie le mot de passe de l'un utilisateur courant
  2nde syntaxe modifie le mot de passe pour un utilisateur donné
- Afficher les utilisateurs :
  - ✓ select user from mysql.user;
- Exemples :
  - ✓ *set password for kone = password('hamed');*
  - ✓ *set password = password('toto');*

Administration

Privilèges

- Utilisateurs et leurs privilèges stockés dans base mysql
  - ✓ Les utilisateurs sont stockés dans la table user
  - ✓ Les privilèges sont stockés dans 4 tables à différent niveau :
    - db: privilèges au niveau des bases de données
    - tables_priv : privilèges au niveau des tables
    - columns_priv : privilèges au niveau des colonnes
    - proc_priv : privilèges au niveau des routines (procédures et fonctions)
  - ✓ On peut gérer les utilisateurs et leur droits en utilisant des requêtes INSERT, UPDATE et DELETE directement sur les tables user, db, tables_priv, columns_priv et proc_priv

Administration

Privilèges

- Quelques privilèges :
  - ✓ **SELECT**, **INSERT**, **UPDATE** et **DELETE** sur une table avec possibilités de restrictions au niveau colonne
  - ✓ **CREATE DATABASE**, **CREATE TABLE**, **CREATE TEMPORARY TABLE**, **CREATE VIEW**, **ALTER** (tables) , **DROP** (tables, vues et bases)
  - ✓ **CREATE ROUTINE** (procédures et fonctions stockées), **ALTER ROUTINE**, **EXECUTE**, **INDEX** (création d'index de tables), **TRIGGERS** (déclencheurs), **LOCK TABLES** (verrouillage de tables), **CREATE USER**

Administration

Privilèges

- Accord de privilèges
  - ✓ *GRANT privilege [(liste_colonnes)] [, privilege [(liste_colonnes)], ...] ON [type_objet] niveau_privilege TO utilisateur [IDENTIFIED BY mot_de_passe][, ...];*
    - privilège : le privilège à accorder (SELECT, EXECUTE, CREATE…)
    - (liste_colonnes) : listes colonnes auxquelles le privilège s'applique
    - niveau_privilege : niveau auquel le privilège s'applique
      . : niveau global à tous les objets de toutes les bases
      *: si aucune base sélectionnée avec (use nom_base) ⇔ à *.*
      nom_base.* : tous les objets de la base
      nom_base.nom_table : la table de la base de données spécifiée
      nom_table : la table de la base de données courante
      nom_base.nom_routine : la routine de la base spécifiée

Administration

Privilèges

- Accord de privilèges
  - type_objet : on peut préciser le type de l'objet
  - Si utilisateur n'existe pas il est créé avec ou sans mot de passe
  - Si utilisateur existe et clause *IDENTIFIED BY* *mot_de_passe* est spécifiée le mot de passe est modifié

✓ Exemples :

- *GRANT SELECT, UPDATE (nom, sexe), DELETE, INSERT ON mabase.etudiant TO 'kone'@'localhost' IDENTIFIED BY 'salif';*
- *GRANT SELECT ON TABLE mabase.Etudiant TO 'assale';*
- *GRANT CREATE ROUTINE, EXECUTE ON mabase.* TO 'koffi';*

Administration

Privilèges

- Revocation de privilèges
  - ✓ REVOKE privilege [, privilege, ...]
    ON niveau_privilege FROM utilisateur [, …];
  - ✓ Exemples :
    - *REVOKE DELETE ON mabase.Etudiant FROM 'kone'@'localhost';*
- Les privilèges ***ALL***, ***USAGE*** et ***GRANT OPTION***
  - ✓ ***ALL*** ou ***ALL PRIVILEGES*** : tous les droits sauf ***GRANT OPTION***
  - ✓ ***USAGE*** : aucun droit
  - ✓ ***GRANT OPTION*** : permet d'utiliser la commande ***GRANT***

Administration

Travaux Pratiques

- Créer une base de données mabase
- Créer les utilisateurs : ing01à connexion local avec mot de passe ing01 et ing02 à connexion à partir de n'importe quel poste avec pour mot de passe ing02
- Donner les droits de création, modification et suppression de tables à ing01 e ing02 sur la base de données mabase. Donner les droits de création et exécution de programmes à ing02.

Langage SQL : L.D.D.

Les types de données

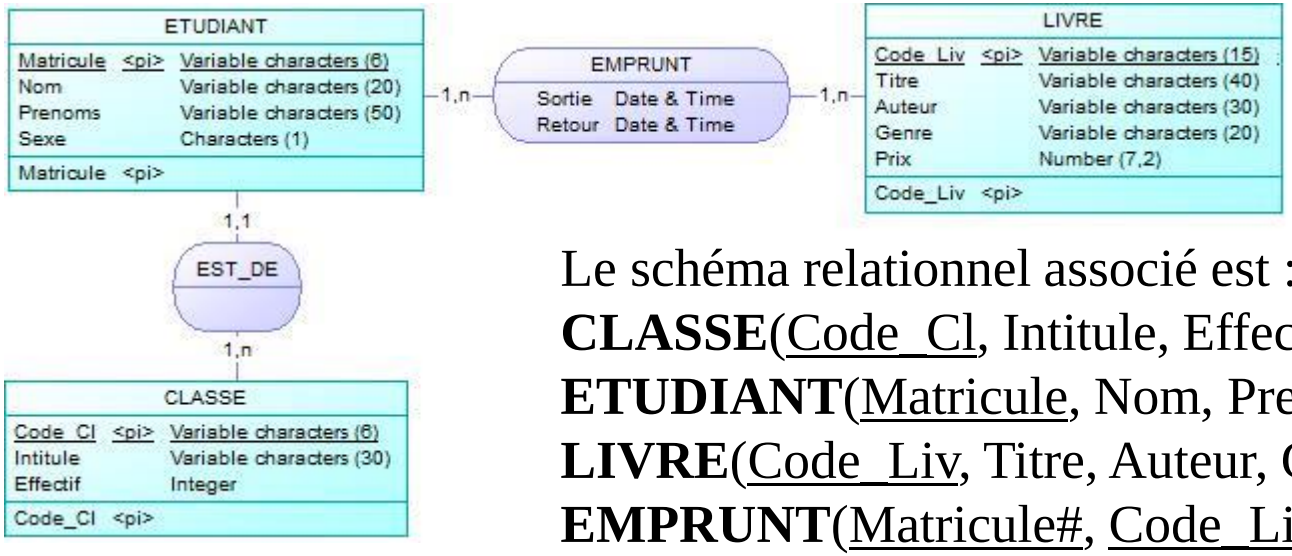
- Chaine de caractères
  - ✓ ***CHAR(n)*** : longueur fixe n
  - ✓ ***VARCHAR(n)*** : longueur variable maximale n
  - ✓ TEXT : texte de longueur quelconque
- Numériques
  - ✓ ***INTEGER***, ***INT***, ***UNSIGNED*** : entier
  - ✓ ***DECIMAL(e, p)***, ***FLOAT(e,p)***, ***DOUBLE(e,p)*** : réels
- Dates
  - ✓ ***DATE*** : date au format AAAA-MM-JJ
  - ✓ ***TIME*** : heure au format HH:MM:SS
  - ✓ ***DATETIME*** : format AAAA-MM-JJ HH:MM:SS

Langage SQL: L.D.D.

Les types de données

- Énumérations
  - ✓ ENUM(valeur1, valeur2, …) : choix d'1 seule valeur dans liste
  - ✓ SET(valeur1, valeur2, …) : choix plusieurs valeurs dans la liste

Base de travail



Le schéma relationnel associé est :

CLASSE(Code_Cl, Intitule, Effectif)
ETUDIANT(Matricule, Nom, Prenoms, Sexe, Code_Cl#)
LIVRE(Code_Liv, Titre, Auteur, Genre, Prix)
EMPRUNT(Matricule#, Code_Liv#, Sortie, Retour)

Langage SQL: L.D.D.

Création d'objets

- Création de tables
  - ✓ Create [temporary] Table [if not exists] <nom_table> (<attribut1><type attribut1>[<contrainte1>],
    …
    <attributN><type attributN>[<contrainte>]) [ENGINE = nom_engine];
- Exemple :
  - ✓ *Create table etudiant (Matricule varchar(6) primary key, Nom varchar(20) not null, Prenoms char(40));*

Langage SQL: L.D.D.

Création d'objets

- Création de vue
  - ✓ Create view <nom_vue> [(liste de colonnes)]
    AS <requête de sélection>;
- Exemple :
  - ✓ *Create vue roman AS Select Code_Liv, Titre, Auteur, Prix From Livre Where genre = 'roman';*
- Création de table à partir d'une autre table
  - ✓ Create table <nom_table> As <requête de sélection>;

Langage SQL: L.D.D.

Création d'objets

- Création d'index
  - ✓ Create [unique] Index <nom_index> on <nom_table> (<col1><ordre>[, <col2><ordre> …]);
    - Unique → chaque valeur de l'index est unique
    - <ordre> ASC ou DESC par défaut ASC
- Exemple
  - ✓ *Create Index Ind_Etud on Etudiant (nom, Prenoms Desc);*
- Suppression d'objets
  - ✓ Drop Table [if Exists] <nom_table>;
  - ✓ Drop View <nom_vue>;
  - ✓ Drop Index <nom_index>;

Langage SQL: L.D.D.

Modification de schéma de table

- Commandes Alter Table
  - ✓ Alter Table <nom_table> Add (<col><type>[<contrainte>], …)
  - ✓ Alter Table <nom_table> Modify <col><type_col>;
  - ✓ Alter Table <nom_table> Drop [column] <col>;
  - ✓ Alter Table <nom_table> Change [column] <col> <nouv_nom> [<nouv_type_col];
  - ✓ Alter Table <nom_table> rename [to|as] <nouveau_nom>;
- Exemples
  - ✓ *Alter table Etudiant Add Sexe char(1);*
  - ✓ *Alter table Etudiant Modify Prenoms varchar(40);*

Langage SQL: L.D.D.

Spécification de contraintes de table

- Contrainte de domaine Default
  - ✓ Create Table <nom_table> (…, <col><type> default <valeur>, …);
- Contrainte d'application Check
  - ✓ Create table <nom_table>(…, <col><type> check (<col> <op> <valeur>), …);
  - ✓ Contrainte acceptée mais pas prise en compte par MySQL on utilise plutôt ENUM ou SET
  - ✓ Create table <nom_table>(…, <col><type> <col> Enum (<valeur1>, …, <valeur>N), …);

Langage SQL: L.D.D.

Spécification de contraintes de table

- Contraintes d'entité :
  - ✓ Create Table <nom_table> (…, <col><type> {not null | unique | primary key}, …);
- Contraintes d'entité : Unique primary key sur +sieurs col
  - ✓ Create table <nom_table>(…, <col><type>, {unique | primary key}(<col1> [,…]), …);
- Contrainte de création d'index
  - ✓ Create table <nom_table>(…, <col><type>, Index [unique] [<nom_index>](<col1> [,…]), …);

Langage SQL: L.D.D.

Spécification de contraintes de table

- Contrainte de référence : references et/ou foreign key
  - ✓ Valable seulement sur le moteur InnoDB de mySQL
  - ✓ Create Table <nom_table> (…, <col><type> references <nom_table2>[(<col>)] {[on Delete [restrict | cascade | set null] | [on Update [restrict | cascade | set null]}, …);
  - ✓ Create table <nom_table>(…, <colN><type>, …, foreign key (<col1> [,…]) references <nom_table2>[(<col1> [,…])] {[on Delete [restrict | cascade | set null] | [on Update [restrict | cascade | set null]}, …);
    - Restrict : rejette la modification avec un message d'erreur
    - Cascade : répercute la modification sur la table référencée
    - Set null : effectue l'action et annule la clé sur la table référençante

Langage SQL: L.D.D.

modification de contraintes de table

- Ajout de contraintes
  - ✓ Alter Table <nom_table> Add([constraint <nom_contrainte>][,…]);
    - Alter Table <nom_table> Add primary key(<col1>[, …]);
    - Alter Table <nom_table> Add index [<nom_index>](<col1>[, …]);
    - Alter Table <nom_table> Add unique [<nom_index>](<col1>[, …]);
- Suppression de contraintes
  - ✓ Alter table <nom_table> Drop primary key;
  - ✓ Alter table <nom_table> Drop foreign key [<nom_contrainte>];
  - ✓ Alter table <nom_table> Drop index [<nom_index>];

Langage SQL: L.D.D.

Travaux Pratiques

- Créer la table Classe avec les contraintes de clé primaire, et default 0 et valeur>=0 sur effectif
- Compléter les attributs manquants de la table Etudiant et spécifier les contrainte de foreign key, et changer le type de sexe pour n'autoriser que 'M' et 'F' comme valeurs
- Créer la table Livre avec les contraintes de clé primaire, et default 1000 pour le prix
- Créer la table Emprunt avec les contraintes de clés primaire et étrangères.

Langage SQL: L.M.D.

Insertion de données

- Une ligne à la fois
  - ✓ Insert Into <nom_table> [(<liste de colonnes>)]
    values (<liste de valeurs>);
- Plusieurs lignes à la fois
  - ✓ Insert Into <nom_table> [(liste de colonnes>)]
    <requête de sélection>;
- N.B. :
  - ✓ Autant de colonnes dans liste de colonnes que de valeurs
  - ✓ désirer ne pas renseigner une colonne spécifier valeur NULL

Langage SQL: L.M.D.

Travaux Pratiques

- Insérer 2 classes
- Insérer 5 étudiants dont 2 de nom Koffi, Zié et un 3ème de nom Jules et de prénoms Kanga
- Insérer 5 livres
- Insérer 10 emprunts avec 4 sans date de retour et 6 avec date de retour – 2 mêmes livres empruntés par Koffi et Zié – le livre numéro 4 est emprunté 2 fois par koffi, la 2ème fois sans date de retour

Langage SQL: L.M.D.

Modification/suppression de données

- Mise à jour de données
  - ✓ Update <nom_table> set <col1>=<expr1> [, <col2>=<expr2>…] where <condition> [limit <val>];
  - ✓ Condition : expressions combinées de la forme *<col><op><expr>*, *(<col1>, <col2>, ...)<op> (<expr1>, <expr2>, ...)* ou *(<col1>, <col2>, ...) <op> (Select...)* à l'aide des opérateurs ***And*** – ***Or*** et ***Not***.
- Suppression de données
  - ✓ Delete from <nom_table> where <condition> [limit <val>];
  - ✓ Delete t1, t2 [,…] from t1 join t2 join t3 [join…] where …;
  - ✓ Truncate Table <nom_table>; : supprime tout le contenu

Langage SQL: L.M.D.

Travaux Pratiques

- Le nom et prénoms de Kanga Jules on été inversés, comment y remédier?
- Koffi vient de rendre le livre numéro 4

Langage SQL: L.M.D.

Sélection de données

- Syntaxe générale
  - ✓ Select [Distinct | All] <liste de colonnes>
    From <liste de tables>
    [where <condition>]
    [Group by <liste de colonnes>]
    [Having <condition>]
    [Order By <critère d'ordre>]
    [Limit <valeur>];
  - ✓ Distinct affiche un seul exemplaire des lignes en double – All est la valeur par défaut

Langage SQL: L.M.D.

Sélection de données

- Clause Select (obligatoire)
  - ✓ Liste de colonnes (opérateur projection) peut être
    - * pour toutes les colonnes (pas de projection)
    - Noms de colonnes (précédé de nom ou alias de table) séparé par (,) : nom_table.col – alias_table.col
    - Colonne peut être calculée : Count(col), …
- Clause From (obligatoire)
  - ✓ Liste de tables (opérateur produit ou jointure)
    - Tables séparées par (,) – table peut être suivi d'alias : Etudiant e, Livre l
    - Ou table peut être requête Select nommée : (Select …) As nom_requête
    - Si plus de 2 tables dans from penser à une jointure condition à spécifier dans le where sinon système effectue produit cartésien

Langage SQL: L.M.D.

Sélection de données

- Clause Where (facultative)
  - ✓ Condition (opérateur sélection) combinaison de
    - Condition de jointure : <alias1.col> <op> <alias2.col>
    - Condition de sélection mormale <alias.col> <op> <expr>
    - <op> peut être tout opérateur arithmétique/logique et les suivants: ***BETWEEN – [NOT]IN – IS[NOT]NULL – LIKE***
      - <col> between <val1> and <val2> - <col> Is Null
      - <col> In (<liste valeurs>) - <col> Like <valeur> : valeur doit contenir '%' (+sieurs caractères) ou '_' (1 seul caractère)
    - Condition sous requête : <col> <op> (Select …)
      - Si <op> ∈{=,<,>,…} Select doit renvoyer 1 et 1 seule ligne
      - Si <op> ∈{=,<,>,…}{any|all} ou <op> ∈{in, not in} elle peut renvoyer plusieurs lignes

Langage SQL: L.M.D.

Sélection de données

- Exemples :
  - ✓ Select numéro, nom from clients;
  - ✓ Select * from articles where prix between 20 and 100;
  - ✓ Select * from clients where nom like '%y%';
  - ✓ Select * from clients where nom not in ('Assale', 'Koffi', 'Zie');
  - ✓ Select * from clients where code_postal is null;
  - ✓ Select designation, prix*1.20 as "Prix TTC" from articles;

Langage SQL: L.M.D.

Sélection de données

- Clause Where (facultative)
  - ✓ Condition (opérateur sélection) combinaison de
    - Condition sous requête corrélée : Where [not] exists (Select …)
      - Select peut contenir plusieurs colonnes et peut renvoyer plusieurs lignes
    - Condition peut être : *(<col1> [, <col2>, ...]) <op> (<expr1>[,<expr2>,...])*
    - Ou *(<col1> [, <col2>, ...]) <op> (Select,...)*
- Clause Group By (facultative)
  - ✓ Fait des partitions et s'utilise avec des fonctions d'agrégats : Count() – Sum() – Avg() – Min() – Max() …
  - ✓ Liste de colonnes : colonnes du Select sauf colonnes calculées

Langage SQL: L.M.D.

Sélection de données

- Clause Having (facultative)
  - ✓ Ne peut s'utiliser que quand il y a Group By
  - ✓ Condition (opérateur de sélection pour les agrégats | partitions)
    - Autorise fonctions d'agrégats sur colonnes dans la condition
- Clause Order By (facultative)
  - ✓ Trie le résultats selon des colonnes
  - ✓ Critère d'ordre:
    - nom ou numéro d'ordre de colonne du Select suivi de ASC ou DESC
- Clause Limit (facultative)
  - ✓ Valeur : spécifie le nombre de lignes à afficher dans le résultat

Langage SQL: L.M.D.

Sélection de données

- Exemples :
  - ✓ Select nom_client, count(*) from clients cl, commandes co where cl.num_cl=co.num_cl group by nom_client;
  - ✓ Select co.num_cmde, count(num_art) from commandes co, ligne_cmde li where co.num_cmde=li.num_cmde group by co.num_cmde having count(num_art)>10;
  - ✓ Select num_art, désignation from articles where prix> (Select avg(prix) from articles);
  - ✓ Select num_art, désignation from articles where prix> any (Select prix from articles where couleur = 'rouge');
  - ✓ Select num_art, désignation from articles where num_art not in (Select num_art from ligne_cmde);

Langage SQL: L.M.D.

Sélection de données: jointures

- Jointure simple
  - ✓ *SELECT liste_colonne FROM table1 t1, table2 t2 [,...] WHERE t1.nomcolonne opérateur t2.nomcolonne [...]*
- Jointure simple normalisée
  - ✓ *SELECT liste_colonne FROM table1 t1 [INNER] JOIN table2 t2 ON t1.nomcolonne opérateur t2.nomcolonne WHERE...*
  - ✓ *SELECT liste_colonne FROM table1 t1 [INNER] JOIN table2 t2 USING(nomcolonne) WHERE ...*

Langage SQL: L.M.D.

Sélection de données: jointures

- Jointures externes normalisées
  - ✓ *FROM nomtable1 LEFT [OUTER] JOIN nomtable2 ON Nomtable1.nomcolonne opérateur nomtable2.nomcolonne WHERE …*
  - ✓ *FROM nomtable1 RIGHT [OUTER] JOIN nomtable2 ON Nomtable1.nomcolonne opérateur nomtable2.nomcolonne WHERE…*
  - ✓ *FROM nomtable1 FULL [OUTER] JOIN nomtable2 ON Nomtable1.nomcolonne opérateur nomtable2.nomcolonne WHERE…*

Langage SQL: L.M.D.

Sélection de données

- Exemples
  - ✓ SELECT a.nom FROM articles a JOIN categories c USING (idcategorie) WHERE c.nom = 'squash';
  - ✓ SELECT a.nom FROM articles AS a JOIN categories AS c ON a.idcategorie=c.idcategorie WHERE c.nom LIKE '%col%';
  - ✓ SELECT nom_client, prenom_client, num_cmde FROM clients LEFT JOIN commandes USING (num_cl) WHERE num_cmde IS NULL ;

Langage SQL: L.M.D.

Sélection de données : opérateurs ensemblistes

- Union : effectue l'union de 2 requêtes
  - ✓ <requête_selection1>
    Union
    <requête_selection2>;
    - Les requêtes doivent avoir même nombre de colonnes avec correspondance de type
- Intersect – Except|Minus
  - ✓ Non implémentés par MySQL
  - ✓ Remplacés par Where [not] Exists (Select…)

Langage SQL: L.M.D.

Sélection de données : opérateurs ensemblistes

- Exemples
  - ✓ Select num_art, designation from ligne_cmde li, articles art where li.num_art= art.num_art and num_cmde='005' union Select num_art, designation from ligne_cmde li, articles art where li.num_art= art.num_art and num_cmde='006' ;
  - ✓ Select num_art from ligne_cmde l, commandes c where l.num_cmde= c.num_cmde and num_cl='CL02' except Select num_art from ligne_cmde l, commandes c where l.num_cmde= c.num_cmde and num_cl='CL10';

Langage SQL: L.M.D.

Travaux Pratiques

- Les étudiants avec les titres de livres empruntés
- Les étudiants avec les titres de livres en cours d'emprunt
- Combien de livres chaque étudiant a emprunté?
- Combien chaque auteur a gagné avec la bibliothèque?
- Quels sont les mêmes livres empruntés par Zié et Koffi?

Langage SQL: L.M.D.

Verrouillage de tables

- Commandes
  - ✓ Lock Tables <nom_table> [[AS] <alias>] <type_verrou> [, <nom_table> [[AS] <alias>] <type_verrou>] …
  - ✓ <type_verrou>
    - Read : empêche l'écriture tout le monde peut lire
    - Write : empêche tous les autres accès – low_Priority Write
  - ✓ Unlock Tables : déverrouillage
  - ✓ Verrous implicites posés en lecture
    - Select … Lock In Share mode : verrou partagé (S) permet à tous de lire la ligne et de poser (S)
    - Select … For Update : verrou exclusif (X) interdire aux autres de poser (X) et (S) sur la ligne

Langage SQL: L.M.D.

Transactions

- Validation/annulation
  - ✓ Commit; : pour valider les modifications dans la base
  - ✓ Rollback; : annule les modification dans la base
- Autocommit
  - ✓ Set autocommit = [0|1]; : passe ou non en mode autocomit
- Points de transaction
  - ✓ Savepoint <nom_point>; : définit un point de validation
  - ✓ Rollback to <nom_point>; : annulation jusqu'à ce point
  - ✓ Release Savepoint <nom_point>; : détruit le point de validation