Université Abou Bakr Belkaid Tlemcen
Faculté des Sciences
Département d'informatique

Administration des Bases de Données

L3 2018-2019

Mr H.MATALLAH

Administration des Bases de Données

1. Notions fondamentales
2. SQL Avancé
3. Gestion d'intégrité et de cohérence
4. Vues et Index
5. Optimisation des requêtes
6. Gestion des transactions : Gestion des accès concurrents

CHAPITRE 2

SQL Avancé

SQL Avancé

Actions sur une BD (Rappel)

- **Définition de la structure de données** (Description du **Contenant**)
- **Interrogation des données** (Opérations de lecture du **Contenu**)
- **Mise à jour des données** (Opérations d'écriture du **Contenu**)
  - Insertion
  - Modification
  - Suppression
- **Contrôle des données** (Cohérence, Sécurité, Concurrence,..)

SQL Avancé

SQL « Structured Query Language » (Rappel)

- **S**tructured **Q**uery **L**anguage est un langage complet de gestion de BD relationnelles
- **Langage d'Interrogation Structuré** qui consiste à **définir , manipuler, contrôler** les BDs
- Un **standard** des **BD R**elationnelles (ISO et ANSI)
- Chaque éditeur tend de développer son propre **dialecte** : rajouter des éléments hors normes qui sont fonctionnellement identiques mais de syntaxes différentes
- Fonctionne sur différents systèmes d'exploitation **(Multi-Plateforme)**

SQL Avancé

SQL « Structured Query Language » (Rappel)

- SQL est un langage de type « **Déclaratif** » : on spécifie ce qu'on veut et c'est la machine qui décide comment elle doit l'exécuter
- Une commande s'appelle un **ordre,** une instruction s'appelle une **requête**
- Une requête peut être utilisée de manière **interactive** ou incluse dans un **programme**
- Les systèmes relationnels sont dénommés dans plusieurs travaux récents par les **systèmes SQL**

SQL Avancé

SQL « Structured Query Language » (Rappel)

- Composé de 3 Sous langages :

Langage de Définition de Données LDD, DDL	Create, Drop, Alter
Langage de Manipulation de Données LMD, DML	Insert, Update, Delete Select
Langages de Contrôle de Données LCD, DCL	Grant, Revoke Commit, Rollback

SQL Avancé

Langage de Définition de données

- **Création de table**
  - **CREATE TABLE <NomTable>** (Attribut1 Domaine [Contrainte_integrité],
    Attribut2 Domaine [Contrainte_integrité],
    ……
    ……
    [Contrainte_integrité], …
    [AS SELECT]) ;
  - **AS requête-SQL** : Création de table avec insertion de données issues d'une autre table (**Importation de données**)
  - Le schéma de la table (Nbre d'attributs, Domaine) correspond à celui retourné par SELECT
  - La définition de la clé primaire peut être reportée

SQL Avancé

Langage de Définition de Données

- **Modification de la structure de la table**
  - **ALTER TABLE** Nom_Table
    RENAME TO Nouveau_Nom_Table |
    RENAME COLUMN Old_Name TO New_Name |
    ADD Nom_Col Domaine|
    MODIFY Nom_Col Domaine|
    DROP COLUMN Nom-Col ;
  - **COLUMN** : Facultatif dans plusieurs SGBD
  - **MODIFY** ou **CHANGE** dans quelques SGBD permet aussi de renommer les colonnes
  - En général, on peut ajouter plusieurs colonnes en une seule requête mais pas pour la modification ou le renommage (une colonne par requête)
- **Suppression de la table**
  - **DROP TABLE** Nom_Table ;

Langage de Manipulation de Données

- **Ajout de données**

1. **INSERT INTO** Nom_Table (Col1, Col2, Col3, ….)
   VALUES (Val1, Val2, Val3,……) ;

   - Les attributs non spécifiés seront NULL ou à la valeur par défaut
   - Les noms de colonnes sont facultatifs si on respecte l'ordre de définition et toutes les valeurs de colonnes sont fournies

Langage de Manipulation de Données

- **Ajout de données**

2. **INSERT INTO** Nom_Table (Col1, Col2, Col3, ….)

   SELECT……..….;

- Insertion de données issues d'une autre table (**Migration de données**)
- Le schéma de la table (Nbre d'attributs, Domaine) correspond à celui retourné par SELECT
- On peut pas ajouter des données dans une table issues d'une sous-sélection de la même table
- Une constante placée dans SELECT permet d'insérer des lignes avec une valeur par défaut
- Toutes les clauses du SELECT sont acceptées mise à part ORDER BY

SQL Avancé

Langage de Manipulation de Données

- **Modification des données**

1. **UPDATE** Nom_Table
   SET Col1=Exp1, Col2=Exp2, …..
   [WHERE Condition] ;

   - « Exp » peut être une valeur, une fonction ou une formule (250, Max(Qte), Moyenne+2)

2. **UPDATE** Nom_Table
   SET (Col1, Col2, …..) = (SELECT ……)
   [WHERE Condition] ;

   - « Exp » peut être un SELECT renvoyant des valeurs attribuées aux colonnes (pour modifier indirectement le contenu d'une table en utilisant un sous-ensemble de données obtenues à partir d'une instruction de requête secondaire)
   - La condition peut contenir des sous interrogations
   - Il n'est pas conseillé de faire des UPDATE sur des colonnes utilisées dans 1 clé primaire

Langage de Manipulation de Données

- **Suppression des données**

1. **DELETE FROM** Nom_Table

   [WHERE Condition] ;

   - Elle est appliquée pour supprimer une ou plusieurs lignes « complétement »
   - Suppression d'une valeur d'une colonne c'est un UPDATE (SET Col=NULL)
   - Si la clause WHERE n'est pas précisée, toutes les lignes de la table sont supprimées
   - La condition peut être plus complexe comme le test d'existence dans une autre table [WHERE Col IN (SELECT Col FROM Table)]

2. **Conseils**

   - Tester d'abord, la clause WHERE dans un simple SELECT
   - Sauvegarder les tuples dans une table temporaire (CREATE …. AS SELECT……)

SQL Avancé

Langage de Manipulation de Données

- **Interrogation**

1. **SELECT** [DISTINCT] {* | Liste_Exp} **Quoi ?** Noms des colonnes à afficher
   FROM {Liste_Tables | Requête AS Nom} **Où ?** Noms des tables ou se trouvent les col
   [WHERE Condition] **Quelle condition ?** Conditions à remplir par les lignes
   [GROUP BY Liste_Exp] Regroupements des lignes en sous tables
   [HAVING Condition] Condition à remplir par le groupe
   [ORDER BY Liste Exp] (ASC, DESC) ; **Comment ?** Ordre d'affichage

- « Exp » est une ou toutes les **colonnes (*),** sinon une **constante** ou une **fonction** ou même une **formule**
- La condition peut employer le résultat d'une autre requête (Sous requête)
- Le nom de la colonne peut ou doit être préfixé par le nom de la table (Table.Col)
- Seules les colonnes du Group By ou les fonctions sont acceptées dans le Select de la requête

Langage de Manipulation de Données

Tri des résultats

1. **SELECT** .....

   [ORDER BY { NomColonne | PositionColonne | Expression }

   [ASC | DESC]

   [NULLS FIRST | NULLS LAST]] ;

- Une colonne de tri peut être spécifiée sous la forme d'un nom ou d'un alias de colonne, ou sous la forme d'un N° de la position de la colonne dans la liste de sélection (1,2,..), ou une expression
- La séquence des colonnes de tri définit l'organisation du jeu de résultats trié (Tri par la première colonne, puis par la deuxième colonne, et ainsi de suite)
- Le nombre de colonnes dans la clause ORDER BY n'est pas limité. Toutefois, la taille totale des colonnes spécifiées généralement est limitée *(SQL Server : 8 060 octets)*

SQL Avancé

Langage de Manipulation de Données

- **Tri des résultats**

1. **SELECT** .....

   [ORDER BY { NomColonne | PositionColonne | Expression }
   [ASC | DESC]
   [NULLS FIRST | NULLS LAST]] ;

- NULLS FIRST : Spécifie que les valeurs NULL doivent être renvoyées avant les valeurs non NULL
- NULLS LAST : Spécifie que les valeurs NULL doivent être renvoyées après les valeurs non NULL
- Si le classement NULL n'est pas spécifié, le traitement des valeurs NULL est le suivant:
  NULLS LAST si le tri est ASC et NULLS FIRST si le tri est DESC
- le tri par défaut des NULL dépend du SGBD *(Oracle : Si ni l'ordre ASC, DESC n'est spécifié, ni l'ordre NULL non plus spécifié, les deux valeurs par défaut sont utilisées et l'ordre sera donc Croissant avec NULLS LAST)*

Langage de Manipulation de Données

Tri des résultats

1. **SELECT** .....

   [ORDER BY { NomColonne | PositionColonne | Expression }

   [ASC | DESC]

   [NULLS FIRST | NULLS LAST]] ;

- Les colonnes de type Ntext , Text , Image , Geography , Geometry et Xml ne peuvent pas être utilisées dans une clause ORDER BY
- Le tri est interne : On ne peut utiliser dans ORDER BY que les col présentes dans le SELECT
- Le tri peut être externe : On peut utiliser dans ORDER BY, une colonne définie dans une table spécifiée dans le FROM qui n'est pas spécifiée dans le SELECT *(le plus fréquent mais avec conditions)*
- Le fait de placer DISTINCT suffit, en général, à établir un tri puisque le moteur doit se livrer à une comparaison des lignes

SQL Avancé

Langage de Manipulation de Données

- **Alias ou Synonymes**

1. Permettent de renommer des colonnes à l'affichage (SELECT Col AS Alias…..)
2. Permettent de renommer des tables dans la requête (SELECT Col,.. FROM Table AS Alias…..)

Utilisation des alias dans Oracle

- Les alias de la clause FROM sont reconnus dans toutes les autres clauses du SELECT
- Les alias du SELECT ne sont reconnus que dans le ORDER BY
- Ils ne le sont pas dans les clauses FROM,WHERE, GROUP BY et HAVING
  (l'ordre d'analyse d'un Select est le suivant FROM, WHERE, GROUP BY/HAVING, SELECT, ORDER BY)
- Un alias défini dans la clause SELECT n'est pas réutilisable dans cette même clause SELECT dans quelques SGBD *(SELECT Sal AS "Salaire", Salaire*12 FROM Emp : Erreur)*

SQL Avancé

Langage de Manipulation de Données

- **Fonctions d'agrégation**

1. Moyenne : **AVG** (Col)
2. Somme : **SUM** (Col)
3. Minimum : **MIN** (Col)
4. Maximum : **MAX** (Col)
5. Cardinalité : **COUNT**

COUNT (*)	Nombre d'enregistrements de la table
COUNT (PK)	Nombre de valeurs de la clé (= COUNT(*))
COUNT (Col)	Nombre de valeurs renseignées de la colonne
COUNT (DISTINCT Col)	Nombre de valeurs distinctes et renseignées

SQL Avancé

Langage de Manipulation de Données

- **Sous requêtes**

La sous-requête dans WHERE de la requête peut retourner une constante ou une relation :

- SELECT…... WHERE Prix = (SELECT Prix FROM……..); *(Retourne une seule valeur)*
- SELECT…... WHERE Prix IN (SELECT Prix FROM……); *(Retourne une ou plusieurs valeurs)*
- SELECT…... WHERE Prix NOT IN (SELECT Prix FROM……); *(Retourne une ou plusieurs valeurs)*
- SELECT…..... WHERE Prix < ALL (SELECT Prix FROM……..); *(Retourne une ou plusieurs valeurs)*
- SELECT……. WHERE Prix < ANY (SELECT Prix FROM……); *(Retourne une ou plusieurs valeurs)*
- SELECT…..... WHERE EXISTS (SELECT... WHERE Prix=5); *(Teste si l'ensemble n'est pas vide)*
- SELECT…..... WHERE NOT EXISTS (SELECT... WHERE Prix=5); *(Teste si l'ensemble est vide)*

SQL Avancé

Langage de Manipulation de Données

- **Jointures**

1. **Produit catésien** (Jointure sans condition)
2. **Thêta-jointure** (Jointure avec une condition quelconque de comparaison entre les attributs)
3. **Equi-jointure** (Jointure avec la condition d'égalité entre les attributs des deux tables)
4. **Jointure naturelle** (Equi-jointure avec test d'égalité entre 2 attributs équivalents en gardant un seul)
5. **Auto-jointure** (Jointure naturelle d'une table avec elle même)
6. **Jointure externe** (Jointure des lignes vérifiant et non vérifiant la condition de jointure)

Langage de Manipulation de Données

- **Jointures internes et externes**

1. **INNER JOIN** : Jointure interne ou fermée, les données doivent être à la fois dans les 2 tables (*Choix par défaut*)

2. **OUTER JOIN** : Jointure externe ou ouverte, on lit les données d'une table en y associant éventuellement celle de l'autre table (*Remplir avec des valeurs NULL si la condition n'est pas respectée*)

Langage de Manipulation de Données

Jointures externes

3 types d'associations :

1. **LEFT [OUTER] JOIN** : On lit les données de la table de gauche en y associant éventuellement celle de la table de droite

2. **RIGHT [OUTER] JOIN** : On lit les données de la table de droite en y associant éventuellement celle de la table de gauche

3. **FULL [OUTER] JOIN** : On lit les données de la table gauche en y associant celle de la table droite ainsi que les données la table droite en y associant celle de la table gauche

Ex : Liste de tous les étudiants avec leurs adresses éventuelles

- Permet d'avoir tous les étudiants ayant d'adresses et n'ayant pas d'adresses
- Ces derniers auront les champs de la table « Adresse » à NULL