



SQL

MANIPULATION DE DONNÉES

1

MANIPULATION DE DONNÉES : MODIFICATION DE LA TABLE

- Suppression d'éléments : DROP (VIEW, INDEX, TABLE)
- DROP TABLE NomTable
(supprime les données et la structure de la table)
- Suppression de données uniquement : TRUNCATE
- TRUNCATE TABLE NomTable
- Renommer une table : RENAME
RENAME TABLE AncienNom TO NouveauNom

MANIPULATION DE DONNÉES : MODIFICATIONS DE SCHÉMAS

Avec ALTER on peut modifier les colonnes d'une table :

- ✓ Modifier le type d'une colonne
ALTER TABLE NomTable
MODIFY NomColonne TypeDonnees

- ✓ Ajouter de nouvelles colonnes
ALTER TABLE NomTable
ADD NomColonne TypeDonnees

- ✓ Supprimer des colonnes
ALTER TABLE NomTable
DROP COLUMN NomCcolonne

(possible que si la colonne ne fait pas partie d'une vue, ne fait pas l'objet d'une contrainte d'intégrité)

MANIPULATION DE DONNÉES : MODIFICATIONS DE SCHÉMAS

- ✓ Ajouter de nouvelles contraintes

```
ALTER TABLE NomTable  
ADD CONSTRAINT NomContrainte
```

- ✓ Supprimer des contraintes

```
ALTER TABLE NomTable  
DROP CONSTRAINT NomContrainte
```

- ✓ Activer ou désactiver toutes les contraintes

```
ALTER TABLE NomTable  
CHECK CONSTRAINT NomContrainte
```

```
ALTER TABLE NomTable  
NOCHECK CONSTRAINT ALL
```

MANIPULATION DE DONNÉES : CRÉATION DE VUES

Une vue est une table virtuelle évaluée à chaque consultation (les données ne sont pas stockées dans une table de la BD). Une vue est définie par une clause SELECT.

- ❖ Obtenir des informations synthétiques
- ❖ Éviter de divulguer certaines informations
- ❖ Assurer l'indépendance du schéma externe

MANIPULATION DE DONNÉES : CRÉATION DE VUES

La syntaxe pour définir une vue est la suivante :

```
CREATE VIEW NomVue(NomColonne1,...) AS SELECT  
NomColonne1,.. FROM NomTable WHERE Condition
```

Exemple :

```
CREATE VIEW EtudiantsSrc(Nom,Prenom) AS SELECT  
Nom,Prenom FROM Etudiants WHERE n_formation=12
```

MANIPULATION DE DONNÉES : INTERROGATION D'UN BD

La commande SELECT permet d'interroger une BD. La syntaxe est la suivante :

```
SELECT [ALL | DISTINCT] NomColonne1,... | *  
FROM NomTable1,...  
WHERE Condition
```

- ALL : option par défaut, sélectionne toutes les lignes qui satisfont la condition
- DISTINCT : permet d'éliminer les doublons

MANIPULATION DE DONNÉES : LA CLAUSE AS

La clause AS permet de renommer les champs dans une requête définie par SELECT.

Exemple :

```
SELECT Compteur AS Ctp FROM Vehicule
```


MANIPULATION DE DONNÉES : RESTRICTION

Les conditions peuvent faire appel aux opérateurs suivants :

- Opérateurs logiques : AND, OR, NOT
- Comparateurs de chaînes : IN, BETWEEN, LIKE
- Opérateurs arithmétiques : +, -, /, %
- Comparateurs arithmétiques : =, ≠, ≤, ≥, <>

Exemple :

```
SELECT * FROM Vehicules  
WHERE (Compteur>10000) AND (Compteur<=30000)
```

MANIPULATION DE DONNÉES : RESTRICTION SUR UNE COMPARAISON DE CHAÎNES

Le prédicat LIKE permet de faire des restrictions sur des chaînes grâce à des caractères appelés caractères joker :

- Le caractère % remplace une séquence (éventuellement vide) de caractères
- le caractère _ remplace un caractère
- les caractères [-] définissent un intervalle de caractères (par exemple [A - E])

Exemple :

```
SELECT * FROM Vehicules WHERE Marque LIKE "_e%"
```

MANIPULATION DE DONNÉES : RESTRICTION SUR UN ENSEMBLE

Les prédicats BETWEEN et IN vérifient

- ❑ Qu'une valeur se trouve dans un intervalle
- ❑ Qu'une valeur appartient à une liste de valeurs

Exemple :

```
SELECT * FROM Vehicules WHERE Compteur BETWEEN  
10000 AND 30000
```

```
SELECT * FROM Vehicules WHERE Marque IN  
("Peugeot","Citroën")
```

MANIPULATION DE DONNÉES : TRI ET AUTRES OPÉRATIONS

- ✓ le tri : ORDER BY...DESC ou ORDER BY...ASC

```
SELECT *  
FROM Vehicules ORDER BY Marque ASC, Compteur DESC
```

- ✓ La moyenne d'une colonne : AVG
- ✓ Le nombre de lignes d'une table : COUNT
- ✓ La valeur maximale d'une colonne : MAX
- ✓ La valeur minimale d'une colonne : MIN
- ✓ La somme des valeurs d'une colonne :
SUM
- ✓ ...

MANIPULATION DE DONNÉES : OPÉRATIONS ENSEMBLISTES

Le produit cartésien :

```
SELECT *  
FROM NomTable1,NomTable2  
WHERE ...
```

Les deux tables sur lesquelles on travaille doivent avoir le même schéma !

- l'union : UNION (pour conserver les doublons : UNION ALL)
SELECT Nom,Prenom FROM Table_Employes UNION SELECT
Nom,Prenom FROM Table_Clients
- l'intersection : INTERSECT
- La différence ensembliste : MINUS