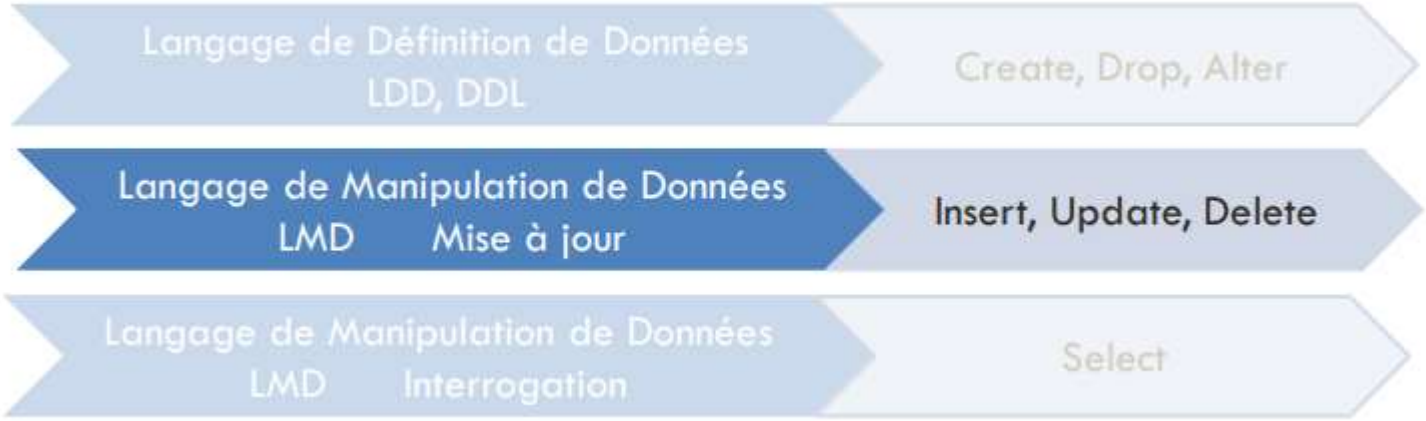
Université Abou Bakr Belkaid – Tlemcen
Faculté des Sciences de la Nature et de la Vie et Sciences de la Terre et de l'Univers
Département des Sciences de la Terre et de l'Univers

Le langage SQL Partie 2

GHENNANI Hind Selma
hindselma.ghennani@univ-tlemcen.dz

SQL - Langage de Manipulation de Données



SQL - Langage de Manipulation de Données

Insertion de données

INSERT INTO <Nom Table> [(Attribut1, Attribut2, Attribut3,….)]

VALUES (Valeur1, Valeur2, Valeur3,…), (Valeur1, Valeur2, Valeur3,…), … ;

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

- **Ex** : INSERT INTO Etudiant (Nom , Prénom, Age, Année_Insc) VALUES ('SARI', 'RIHAM', 20, 2016);
  INSERT INTO Etudiant VALUES ('BENSAYAH', 'FATIMA', 21, 2015);
  INSERT INTO Etudiant (Age, Prénom, Année_Insc, Nom) VALUES (20, 'AMAL', 2016, 'MAHI');
  INSERT INTO Etudiant (Prenom, Age) VALUES ('MERYEM', 20);
  INSERT INTO Etudiant VALUES ('BENSOUNA', Null, 20, Null);

SQL - Langage de Manipulation de Données

Insertion de données (Exercice)

Etudiant (Nom, Prénom, Age, Année_Insc) : Parmi ces 7 requêtes, quelles qui sont justes ou fausses ?

1. INSERT INTO Etudiant VALUES ('AMARBENSABEUR', 'NADJWA', '2015'); Fausse
2. INSERT INTO Etudiant VALUES ('ANITER', 'HICHAME', 21, 2016), ('ARICHI', '111', 20, 2022), ('ATTAR', '222', 19, 2010), ('BAGHLI', '333', 20, 1920); Juste
3. INSERT INTO Etudiant (Prénom, Age) VALUES ('AMINA', 20), ('ISLAM', 20), ('BAHAR', 'AYMEN', 20); Fausse
4. INSERT INTO Etudiant VALUES (19, 'ROMAISSA', 2016, 'BELDJILALI'); Fausse
5. INSERT INTO Etudiant (Age) VALUES (20); Juste
6. INSERT INTO Etudiant VALUES (Null, 'SOUFYANE', Null, Null); Juste
7. INSERT INTO Etudiant VALUES (Nom , Prénom, Age, Année_Insc) ('BENOMARI', Null, Null); Fausse

SQL - Langage de Manipulation de Données

Modification de données

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

- **SET** : Spécifie les colonnes à modifier
- **WHERE** : Clause facultative qui spécifie les lignes concernées par cette modification
- **Exp** : peut être une valeur, une fonction ou une formule
- **Ex : Etudiant (Nom, Prénom, Age, Ville, Note)**

```
UPDATE  Etudiant SET Ville = 'MAGHNIA' WHERE Nom='BENOSMAN' ;
UPDATE  Etudiant SET Age= MAX (Age) ;
UPDATE  Etudiant SET Note = Note+2 WHERE Note < 8 ;
```

SQL - Langage de Manipulation de Données

Modification de données (Exercice)

Etudiant (Nom, Prenom, Age, Ville, Note)

1. Remplacer la ville de tous les étudiants résidant à 'Relizane' par 'Mostaganem'

```
UPDATE Etudiant SET Ville = 'Mostaganem' WHERE Ville = 'Relizane' ;
```

2. Augmenter la note de tous les étudiants ayant un âge moins de 20 ans de 20 %

```
UPDATE Etudiant SET Note = Note * 1.2 WHERE Age < 20 ;
```

3. Initialiser tous les âges des étudiants

```
UPDATE Etudiant SET Age = 0 ;
```

4. Enlever tous les prénoms des étudiants habitant à 'Remchi'

```
UPDATE Etudiant SET Prenom = NULL WHERE Ville = 'Remchi' ;
```

SQL - Langage de Manipulation de Données

Suppression de données

DELETE FROM <Nom_Table>
[**WHERE** Condition] ;

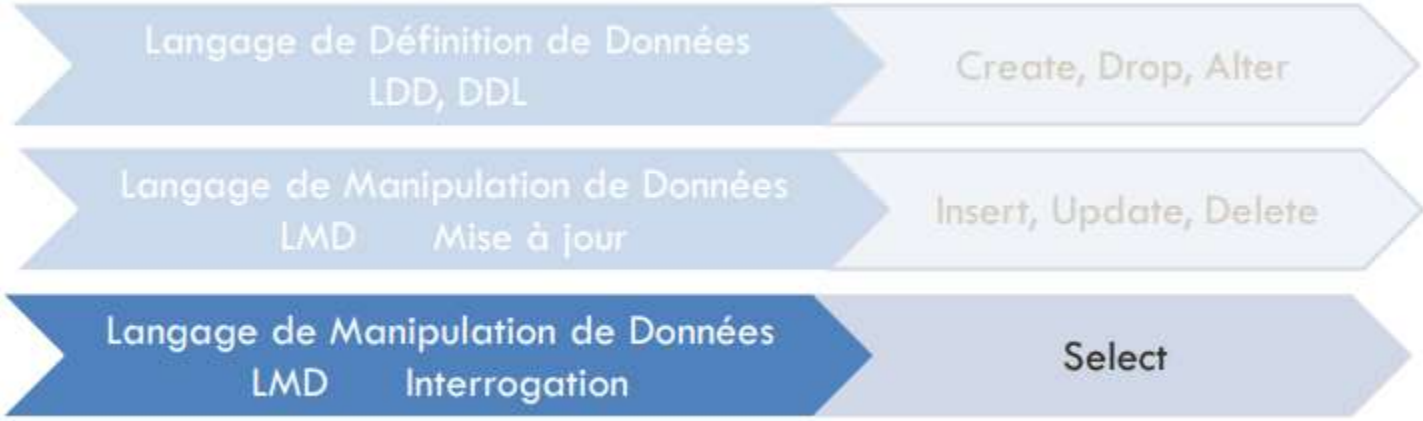
- Supprime une ou plusieurs lignes d'une table vérifiant une certaine condition
- La condition est optionnelle : Si elle n'est pas précisée, **toutes !!** les lignes de la table sont supprimées
- Cette opération n'affecte pas le schéma de la table relationnelle
- **Ex** : Supprimer les étudiants qui ont une note inférieure à 4

```
DELETE FROM Etudiant WHERE Note < 4 ;
```

Vider la table Etudiant

```
DELETE FROM Etudiant ;
```

SQL - Langage de Manipulation de Données



SQL - Langage d'Interrogation de Données

Schéma de la BD de Vente de Voitures d'Occasions

- **Voiture** (MatV, Marque, Type, Couleur, Km)
- **Client** (CodeC, Nom, Age, Ville, Sexe)
- **Vente** (Num, DateVente, Prix, MatV#, CodeC#)

VOITURE

MatV	Marque	Type	Couleur	Km(10^3)
11	PEUGEOT	307	Marron	150
22	CITROEN	C3	Noir	210
33	PEUGEOT	206	Gris	270
44	RENAULT	Clio		180
55	FIAT	Punto	Rouge	120
66	RENAULT	Megane	Noir	90

CLIENT

CodeC	Nom	Ville	Age	Sexe
1	BOUAYED	TLEMCEN	38	F
2	HEBRI	MAGHNIA	60	M
3	CHERIF	REMCHI	52	M
4	BOUDEHARI	SEBRA	45	F
5	DEMMOUCHE	TLEMCEN	75	M
6	MOUMENI	MAGHNIA	29	F

VENTE

Num	DateVente	Prix (*10^4)	MatV	CodeC
01	03/12/2015	165	11	1
02	30/03/2016	110	22	4
03	14/06/2014	75	44	1
04	02/04/2017	94	55	2
05	05/01/2018	138	66	5

SQL - Langage d’Interrogation de Données

SQL SELECT

L’utilisation la plus courante de SQL consiste à lire des données issues de la base de données. Cela s’effectue grâce à la commande **SELECT**, qui retourne des enregistrements dans un tableau de résultat. Cette commande peut sélectionner une ou plusieurs colonnes d’une table.

SQL - Langage d’Interrogation de Données

L’utilisation basique de cette commande s’effectue de la manière suivante:

```
SELECT nom_colonnes FROM nom_table
```

SQL - Langage d'Interrogation de Données

Projection sur toutes les colonnes de la table VENTE

- Etat des ventes

```
SELECT * FROM VENTE ;
```

VENTE

Num	DateVente	Prix (Million)	MatV	CodeC
01	03/12/2015	165	11	1
02	30/03/2016	110	22	4
03	14/06/2014	75	44	1
04	02/04/2017	94	55	2
05	05/01/2018	138	66	5

Projection sur quelques colonnes de la table CLIENT

- Liste des noms des clients avec leurs âges correspondants

```
SELECT Nom, Age FROM CLient ;     ou
SELECT Client.Nom, Client.Age FROM CLient ;
```

RESULTAT

Nom	Age
BOUAYED	38
HEBRI	60
CHERIF	52
BOUDEHARI	45
DEMMOUCHE	75
MOUMENI	29

SQL - Langage d'Interrogation de Données

DISTINCT : Elimine les éventuelles données dupliquées

- Afficher les marques des voitures

```
SELECT Marque FROM Voiture ;
```

RESULTAT

Marque
PEUGEOT
CITROEN
PEUGEOT
RENAULT
FIAT
RENAULT

- Afficher les différentes marques de voitures

```
SELECT DISTINCT Marque FROM Voiture ;    ou
SELECT DISTINCT (Marque) FROM Voiture ;
```

RESULTAT

Marque
PEUGEOT
CITROEN
RENAULT
FIAT

SQL - Langage d'Interrogation de Données

Fonctions statistiques de traitement des données d'une colonne (fonctions de groupe)

MAX	Maximum des valeurs d'une colonne
MIN	Minimum des valeurs d'une colonne
AVG	Moyenne des valeurs d'une colonne
SUM	Somme des valeurs d'une colonne
COUNT	Comptage du nombre de lignes de la table (Cardinalité)

SQL - Langage d'Interrogation de Données

• Calcul de la moyenne et la somme des prix de vente

SELECT AVG (Prix), SUM(Prix) FROM Vente ;

Prix Moyen	SUM(Prix)
116.4	582

• Donner le plus jeune et le plus âgé Client

SELECT MIN(Age), MAX(Age) FROM Client ;

MIN(Age)	MAX(Age)
29	75

• Nombre d'enregistrements de la table Client

SELECT COUNT (*) FROM Client

Nombre
6

• Nombre de valeurs renseignées de la colonne

SELECT COUNT (*) FROM Voiture

Nombre
5

SQL - Langage d’Interrogation de Données

SQL WHERE

La commande WHERE dans une requête SQL permet d’extraire les lignes d’une base de données qui respectent une condition. Cela permet d’obtenir uniquement les informations désirées.

```
SELECT nom_colonnes FROM nom_table WHERE condition
```

SQL - Langage d'Interrogation de Données

Opérateurs de comparaisons

Opérateur	Description
=	Égale
<>	Pas égale
!=	Pas égale
>	Supérieur à
<	Inférieur à
>=	Supérieur ou égale à
<=	Inférieur ou égale à
IN	Liste de plusieurs valeurs possibles
BETWEEN	Valeur comprise dans un intervalle donnée (utile pour les nombres ou dates)
LIKE	Recherche en spécifiant le début, milieu ou fin d'un mot.
IS NULL	Valeur est nulle
IS NOT NULL	Valeur n'est pas nulle

SQL - Langage d'Interrogation de Données

Extraction des ventes dont le prix est supérieur à 100

SELECT * FROM Vente WHERE Prix > 100 ;

VENTE

Num	DateVente	Prix (*10^4)	MatV	CodeC
01	03/12/2015	165	11	1
02	30/03/2016	110	22	4
05	05/01/2018	138	66	5

Extraction des voitures marron, grises ou rouges

SELECT * FROM Voiture WHERE Couleur IN ('Marron', 'Gris', 'Rouge')

VOITURE

MatV	Marque	Type	Couleur	Km(10^3)
11	PEUGEOT	307	Marron	150
33	PEUGEOT	206	Gris	270
55	FIAT	Punto	Rouge	120

Recherche des clients dont l'âge est compris entre 50 et 65 ans

SELECT * FROM Client WHERE Age BETWEEN 50 AND 65 ;

CLIENT

CodeC	Nom	Ville	Age	Sexe
2	HEBRI	MAGHNIA	60	M
3	CHERIF	REMCHI	52	M

Recherche des voitures dont la couleur est inconnue

SELECT * FROM Voiture WHERE Couleur IS NULL ;

VOITURE

MatV	Marque	Type	Couleur	Km(10^3)
44	RENAULT	Clio		180

SQL - Langage d'Interrogation de Données

Recherche des clients dont le nom contient la lettre « B »

SELECT * FROM Client WHERE Nom LIKE '%B%' ;

CLIENT

CodeC	Nom	Ville	Age	Sexe
1	BOUAYED	TLEMCEN	38	F
2	HEBRI	MAGHNIA	60	M
4	BOUDEHARI	SEBRA	45	F

Recherche des voitures de marque PEUGEOT dépassant les 200 (10^3) Km

SELECT * FROM Voiture WHERE Marque='PEUGEOT' AND Km > 200 ;

VOITURE

MatV	Marque	Type	Couleur	Km(10^3)
33	PEUGEOT	206	Gris	270

SQL - Langage d'Interrogation de Données

Recherche des clients résidants à Remchi et les clients résidants à Maghnia

```
SELECT * FROM Client WHERE (Ville='Remchi' OR Ville='Maghnia') ;
```

CLIENT

CodeC	Nom	Ville	Age	Sexe
2	HEBRI	MAGHNIA	60	M
3	CHERIF	REMCHI	52	M
6	MOUMENI	MAGHNIA	29	F

SQL - Langage d'Interrogation de Données

Opérateurs et connecteurs logiques

- Liste des voitures qui n'ont pas les couleurs marron, grises ou rouges

  SELECT * FROM Voiture WHERE Couleur NOT IN ('Marron', 'Gris', 'Rouge') ;

- Liste des clients dont l'âge n'est pas compris entre 50 et 65 ans

  SELECT * FROM Client WHERE Age NOT BETWEEN 50 AND 65 ;

- Liste des voitures dont la couleur est renseignée

  SELECT * FROM Voiture WHERE Couleur IS NOT NULL ;

- Recherche des clients dont le nom ne contient pas la lettre « B »

  SELECT * FROM Client WHERE Nom NOT LIKE '%B%' ;

SQL - Langage d’Interrogation de Données

Extraction de données de plusieurs tables (Jointures)

• Les jointures permettent d’extraire des données issues de plusieurs tables

• La majorité des requêtes utilisent les jointures nécessaires pour pouvoir extraire des données de tables distinctes

• Une jointure met en relation deux tables sur la base d’une clause de jointure (comparaison de colonnes)

SQL - Langage d'Interrogation de Données

Jointure

SELECT Col1, Col2, …

FROM Nom_Table1, Nom_Table2,…

WHERE Condition de jointure ;

- Afin d'éviter les ambiguïtés concernant les noms de colonnes, on utilise les alias pour suffixer les tables dans la clause FROM et préfixer les colonnes dans les clauses SELECT et WHERE

SELECT [Alias1.]Col1, [Alias2.]Col2, …

FROM Nom_Table1 [Alias1] , Nom_Table2 [Alias2],…

WHERE Condition de jointure ;

SQL - Langage d'Interrogation de Données

Jointure

En fonction de la nature de l'opérateur utilisé et les tables concernées, on distingue :

- **Thêta-jointure ou Inéqui-jointure** (<>, <, ≤, >, ≥, BETWEEN, LIKE, IN)
- **Equi-jointure** (=) (Equi Join)
- **Jointure naturelle** (Natural Join)
- **Auto-jointure** (Self Join)

SQL - Langage d'Interrogation de Données

Inéqui-jointure

Marques, Types de Voitures, Km et Prix de vente ou le Prix de vente est supérieur au Kilométrage

```
SELECT Marque, Type
FROM Vente, Voiture
WHERE Prix > Km ;
```

RESULTAT

Marque	Type	Km(10^3)	Prix
PEUGEOT	307	150	165
FIAT	Punto	120	165
FIAT	Punto	120	138
RENAULT	Megane	90	165
RENAULT	Megane	90	110
RENAULT	Megane	90	94
RENAULT	Megane	90	138

SQL - Langage d'Interrogation de Données

Equi-jointure

Noms, Ages de Clients, Dates de ventes ou le Matricule de la voiture est égal à l'âge Client

```
SELECT Nom, Age, DateVente
FROM Vente, Client
WHERE MatV = Age ;
```

RESULTAT

Nom	Age	DateVente
HEBRI	66	05/01/2018

SQL - Langage d'Interrogation de Données

Jointure Naturelle

- Est une équi-jointure sur les attributs équivalents suivie de la projection qui permet de conserver **un seul** de ces attributs égaux
- La comparaison fait intervenir la **clé étrangère** d'une table avec la **clé primaire** d'une autre table
- En pratique, c'est la jointure la plus utilisée

VOITURE

MatV	Marque	Type	Couleur	Km(10^3)
11	PEUGEOT	307	Marron	150
22	CITROEN	C3	Noir	210
33	PEUGEOT	206	Gris	270
44	RENAULT	Clio	Blanche	180
55	FIAT	Punto	Rouge	120
66	RENAULT	Megane	Noir	90

VENTE

Num	DateVente	Prix (*10^4)	MatV	CodeC
01	03/12/2015	165	11	1
02	30/03/2016	110	22	4
03	14/06/2014	75	44	1
04	02/04/2017	94	55	2
05	05/01/2018	138	66	5

CLIENT

CodeC	Nom	Ville	Age	Sexe
1	BOUAYED	TLEMCEN	38	F
2	HEBRI	MAGHNIA	66	M
3	CHERIF	REMCHI	52	M
4	BOUDEHARI	SEBRA	45	F
5	DEMMOUCHE	TLEMCEN	75	M
6	MOUMENI	MAGHNIA	29	F

SQL - Langage d'Interrogation de Données

Jointure Naturelle

- Noms, Villes, Ages des Clients avec les Matricules de Voitures achetées

```
SELECT Nom, Ville, Age, MatV
FROM Vente, Client
WHERE Vente.CodeC = Client.CodeC ;
```

ou

```
SELECT Client.Nom, Client.Ville, Client.Age, Vente.MatV
FROM Vente, Client
WHERE Vente.CodeC = Client.CodeC ;
```

RESULTAT

Nom	Ville	Age	MatV
BOUAYED	TLEMCEN	38	11
BOUAYED	TLEMCEN	38	44
HEBRI	MAGHNIA	66	55
BOUDEHARI	SEBRA	45	22
DEMMOUCHE	TLEMCEN	75	66

SQL - Langage d'Interrogation de Données

Jointure Naturelle

- Noms et Villes des Clients avec les Marques, Types et Couleurs des Voitures achetés

```
SELECT Nom, Ville, Marque, Type, Couleur
FROM Vente, Client, Voiture
WHERE Vente.CodeC = Client.CodeC AND Vente.MatV = Voiture.MatV ;
```

RESULTAT

Nom	Ville	Marque	Type	Couleur
BOUAYED	TLEMCEN	PEUGEOT	307	Marron
BOUAYED	TLEMCEN	RENAULT	Clio	Blanche
HEBRI	MAGHNIA	FIAT	Punto	Rouge
BOUDEHARI	SEBRA	CITROEN	C3	Noir
DEMMOUCHE	TLEMCEN	RENAULT	Megane	Noir

- Une jointure de trois tables est exprimée en deux jointures de deux tables

SQL - Langage d'Interrogation de Données

Auto-jointure (Cas particulier de jointure naturelle, reliant une table avec elle-même)

- Couples de Marques et Types de Voitures ayant la même Couleur
  - **SELECT** V1.Marque, V1.Type, V2.Marque, V2.Type
    FROM Voiture V1, Voiture V2
    WHERE V1.Couleur = V2.Couleur ;
- Pour éliminer les combinaisons inutiles
  - **SELECT** V1.Marque, V1.Type, V2.Marque, V2.Type
    FROM Voiture V1, Voiture V2
    WHERE V1.Couleur = V2.Couleur AND V1.MatV <> V2.MatV ;
- Améliorant davantage les résultats
  - **SELECT** V1.Marque, V1.Type, V2.Marque, V2.Type
    FROM Voiture V1, Voiture V2
    WHERE V1.Couleur = V2.Couleur AND V1.MatV > V2.MatV ;

RESULTAT

Marque	Type	Marque	Type
PEUGEOT	307	PEUGEOT	307
CITROEN	C3	CITROEN	C3
CITROEN	C3	RENAULT	Megane
PEUGEOT	206	PEUGEOT	206
RENAULT	Clio	RENAULT	Clio
FIAT	Punto	FIAT	Punto
RENAULT	Megane	RENAULT	Megane
RENAULT	Megane	CITROEN	C3

RESULTAT

Marque	Type	Marque	Type
CITROEN	C3	RENAULT	Megane
RENAULT	Megane	CITROEN	C3

RESULTAT

Marque	Type	Marque	Type
CITROEN	C3	RENAULT	Megane