

SQL

Plus d'informations sur <http://www.dawan.fr>
Contactez notre service commercial au [09.72.37.73.73](tel:09.72.37.73.73) (prix d'un appel local)

Objectifs



- ✓ Découvrir SQL
- ✓ Savoir modéliser une Base de Données
- ✓ Être capable de créer les requêtes standards de sélection, d'enregistrement, de modification ou de suppression de données
- ✓ Savoir relier des tables entre elles

Plan de l'intervention



- ✓ Introduction
- ✓ Commandes simples
- ✓ Présentation des SGBDR
- ✓ Utilisation avancée

Introduction

Introduction



Bases de données

✓ Une base de données ?

- 1) Une base de données est une collection de données organisées de façon à être facilement accessibles, administrées et mises à jour.
- 2) Les bases de données peuvent être classées par le type de contenu qu'elles renferment : **bibliographique, full text, images ou des nombres....**

Base de Données



- ✓ Un SGBD ?
- ✓ *Systeme de Gestion de Bases de Données*
- ✓ Le SQL ?
- ✓ *Structured Query Language*

Introduction



Historique, versions et normalisation

1970 : Invention du langage par IBM.

1982-1989 : SQL 1.

1992 : SQL 2 : Est développé à partir de SEQUEL de IBM .

1999 : SQL 3 : SQL3 supporte bien sûr la norme précédente SQL2 (compatibilité ascendante) mais propose également des extensions objets.

* 4 niveaux de complexité :Entry ,Transitional, Intermediate et Full

****Chaque implémentation de SQL doit maintenir au moins le niveau « Entry »**

SGBD



- ✓ Un système de gestion de base de données **(SGBD)** :-
- ✓ Est une application qui sert comme son nom l'indique à stocker et accéder à des données.
- ✓ **MySQL** : Sous licence GPL (gratuite) performante quoique légèrement incomplète
- ✓ **PostgreSQL** : (PSQL) Egalement Open Source performante et offrant de nombreuses fonctionnalités
- ✓ **Oracle** : La base de données professionnelle
- ✓ **DB2** (IBM): Une autre base de données professionnelle
- ✓ **SQL Server** (MS): La solution proposée par Microsoft

Introduction



Méthode MERISE est une méthode de conception, de développement et de réalisation de projets informatiques.

1. Modèle Conceptuel de Données
2. Modèle Logique de Données
3. Modèle Physique de Données

Méthode Merise

1. Modèle Conceptuel de Données (MCD)

- ✓ Le MCD a été créé pour normaliser l'écriture des données. C'est une représentation des données et de leurs interactions.
- ✓ Définir les entités et relations de notre BD
- a) **Entité** : ensemble de propriétés encore appelées attributs ou caractéristiques
- b) **Associations** : Une association définit un lien sémantique entre une ou plusieurs entités



Les requêtes de structure

Les Commandes



```
CREATE DATABASE NomDeLaBase;  
SHOW DATABASES;  
USE NomDeLaBase;  
SHOW TABLES;
```

Commandes simples



Les types de données

- ✓ **Numériques** : TINYINT (1 Octets) / SMALLINT (2 octets)
/ INTEGER (4 octets) / FLOAT / DOUBLE (UNSIGNED)

- ✓ **Textuels** : CHAR / VARCHAR / TEXT / LONGTEXT

- ✓ **Temporels** : DATE / TIME / DATETIME / TIMESTAMP

Commandes simples



Créations d'une table

```
CREATE TABLE [nomTable] (  
    [nomChamp] [type] [NULL] [options],  
    PRIMARY KEY ([nomChamp]) );
```

NULL : champ facultatif

NOT NULL : champ obligatoire

Options : AUTO_INCREMENT, DEFAULT, UNIQUE, CHECK

Commandes simples



Modifications de structure de la table

* **ALTER TABLE** [*nomTable*] ...

MODIFY [*nomChamp*] [type] [NULL] [*options*];

ADD COLUMN [*nomChamp*] [type] [NULL] [*options*];

DROP COLUMN [*nomChamp*];

* **DROP TABLE** [*nomTable*];

Commandes simples



Insertion de données

```
INSERT INTO [nomTable] (champX, ...)  
VALUES ( valeurX, ...);
```

Insertion sur plusieurs lignes possible.

Commandes simples



Insertion de données

Colonne muette :

- Colonne clé avec `auto_increment` > numéro automatique inséré
- Colonne avec valeur par défaut > valeur par défaut insérée
- Colonne *nullable* > valeur NULL insérée
- Colonne sans valeur par défaut et non nullable
 - > l'insertion de la ligne est refusée

Commandes simples



Modifier les données

Modifier :

UPDATE *nomTable*

SET *champX = valeurX*

WHERE *id = 1;*

Pas de condition : met à jour toute la table !

Commandes simples



Suppression de données

Supprimer :

```
DELETE FROM nomTable  
WHERE id = 1;
```

Pas de condition : met à jour toute la table !

Vider la table :

```
TRUNCATE TABLE nomTable ;
```

Commandes simples



Accéder aux données

Sélectionner :

```
SELECT * FROM nomTable;
```

Alias de colonne :

```
SELECT prix_ht, prix_ht*1.20 AS prix_ttc  
FROM nomTable ;
```

Commandes simples



Accéder aux données

Clause **WHERE** :

SELECT *champX, champY* **FROM** *nomTable*

WHERE *condition1*

AND *condition2...* ;

Comparaison : =, <, <=, >, >=, <> / !=, LIKE, BETWEEN

Commandes simples



Accéder aux données

Clause ORDER BY :

Organiser les résultats (alphabétique, numérique)

Clause LIMIT :

Limiter le nombre de résultats.

Commandes simples



L'agrégation

- **SUM** calcul la somme de la colonne
- **AVG** calcul la moyenne de la colonne
- **MAX** calcul le maximum de la colonne
- **MIN** calcul le minimum de la colonne
- **COUNT** dénombre les éléments

Uniquement dans le SELECT

Commandes simples



Le regroupement

Clause **GROUP BY** :

Regrouper les calculs par valeur d'une colonne

```
SELECT id_genre, COUNT(id_livre) AS nb_livre  
FROM livre  
GROUP BY id_genre ;
```


Commandes simples



L'ordre des mots clés SQL

SELECT...

FROM...

JOIN...

WHERE...

AND/OR...

GROUP BY...

HAVING...

ORDER BY...

LIMIT...

Commandes avancées

Les Jointures : Requêtes multi-tables

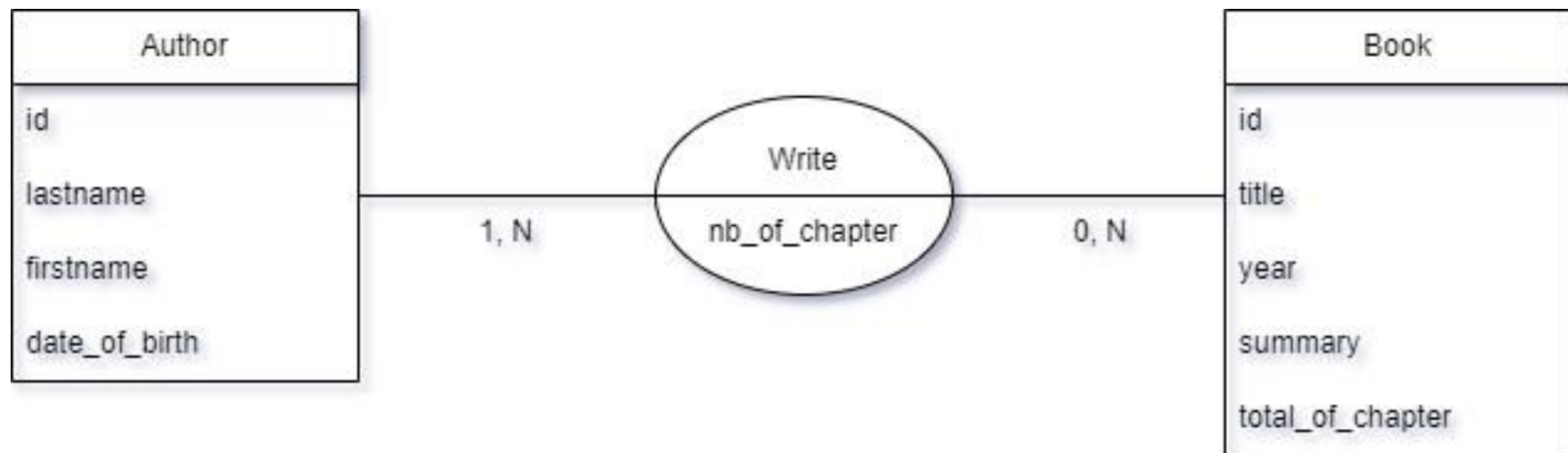
UPDATE avec jointures

Méthode Merise

Modèle Conceptuel de Données (MCD)

Exemple d'Entités et d'Association :

Remarquez bien les cardinalités 0,N/1,N de part et d'autre des entités



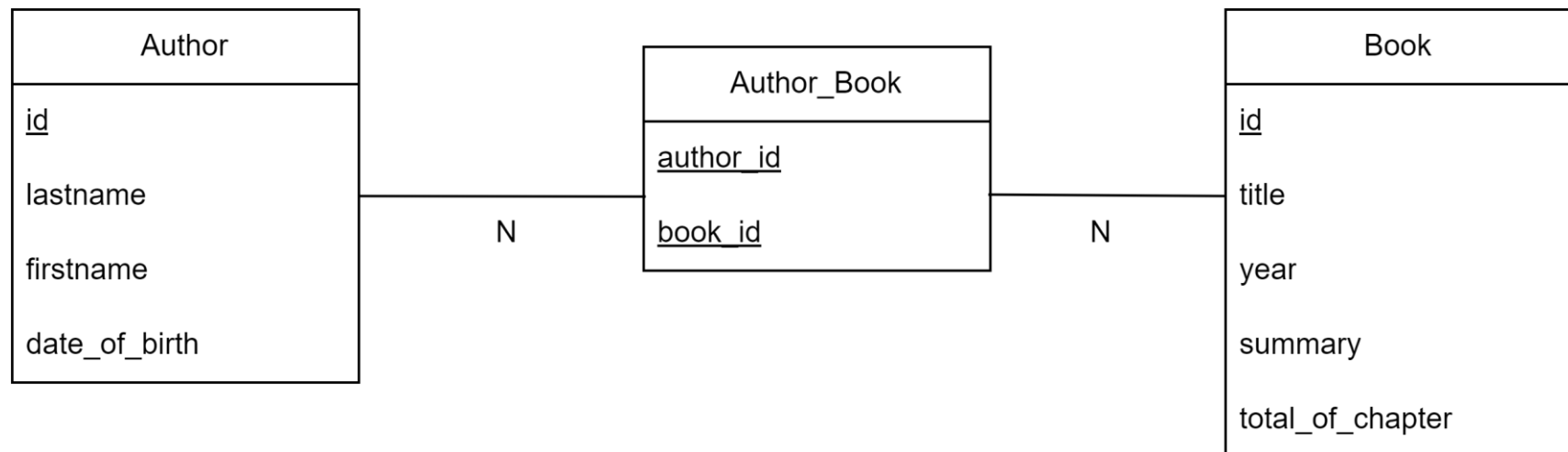
Une cardinalité 1.N signifie que chaque entité appartenant à une classe d'entité participe au moins une fois à la relation.

Une cardinalité 0.N signifie que chaque entité appartenant à une classe d'entité ne participe pas forcément à la relation.

Méthode Merise

2. Modèle Logique de Données (MLD)

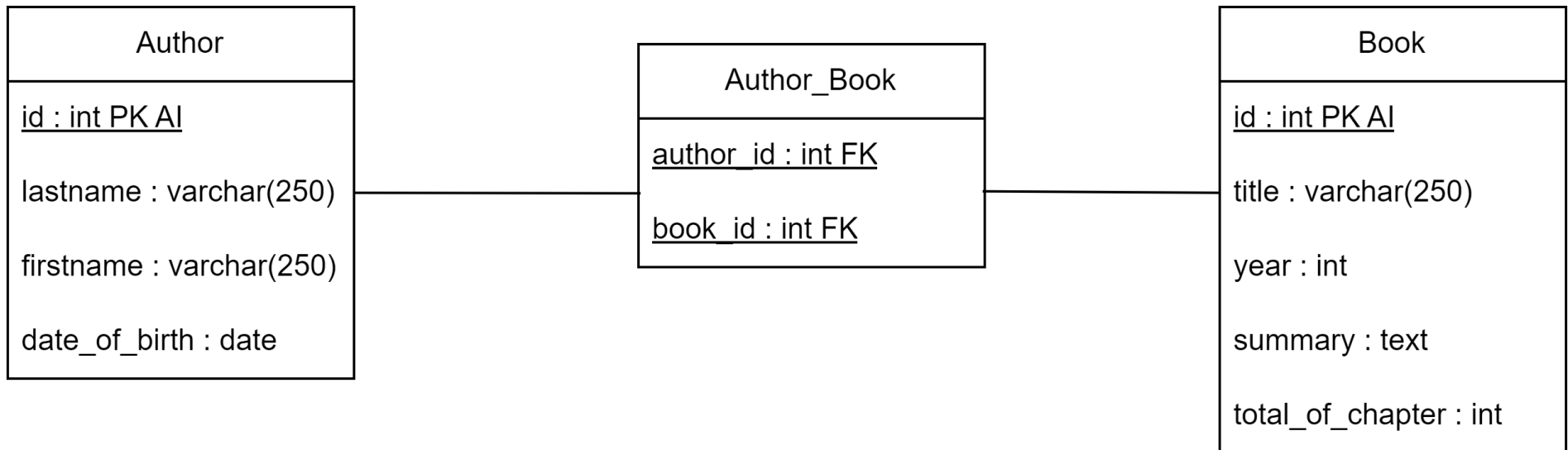
- ✓ Le **MLD** consiste à décrire la structure de données utilisée sans faire référence à un langage de programmation.
- ✓ Ainsi, le modèle logique est encore, à ce stade, indépendant du **type de base de données utilisé**.



Méthode Merise

3. Modèle Physique de Données (MPD)

- ✓ Il s'agit avec le **MPD** de préciser le type de données utilisées lors des traitements.
- ✓ Ainsi, le modèle logique est, cette fois, dépendant du **type de base de données utilisé**.



Les associations

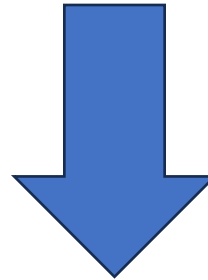
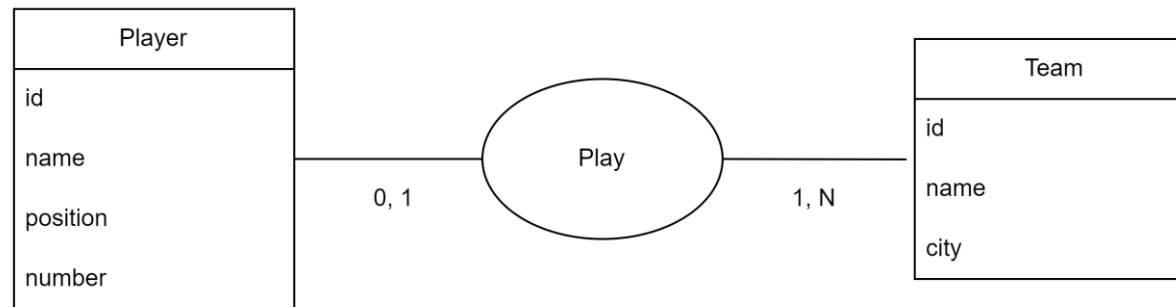


Entre les tables SQL – 3 types associations possibles

1. One – to – One
2. One – to – Many
3. Many – to – Many

One to many

MCD



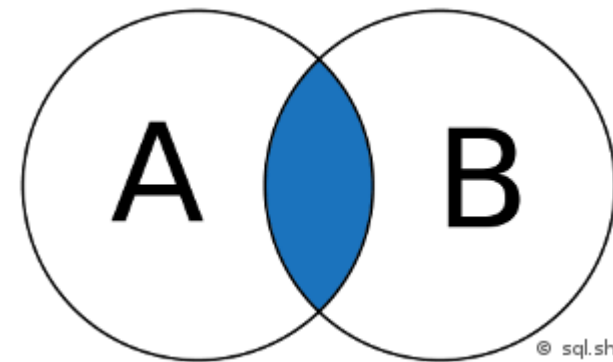
MLD



Jointures : Requêtes multi-tables

Les Jointures :

- **INNER JOIN**



```
SELECT * FROM A
```

```
INNER JOIN B ON A.key = B.key
```


Requête INNER JOIN

Sélection de données sur plusieurs tables

SELECT *

FROM tableA, tableB, tableC

WHERE tableA.id_tabB = tableB.id

AND tableA.id_tabC = tableC.id;

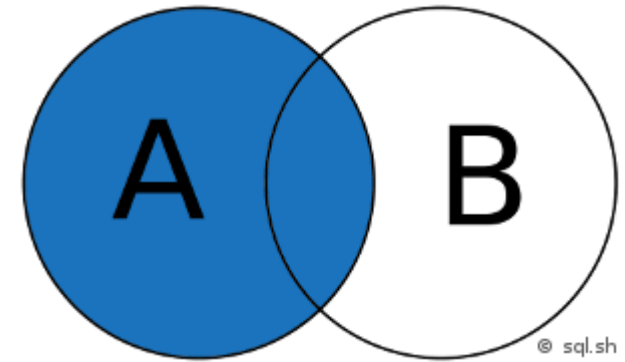
Commandes avancées

Jointures : Requêtes multi-tables

- LEFT JOIN**

```
SELECT * FROM A
```

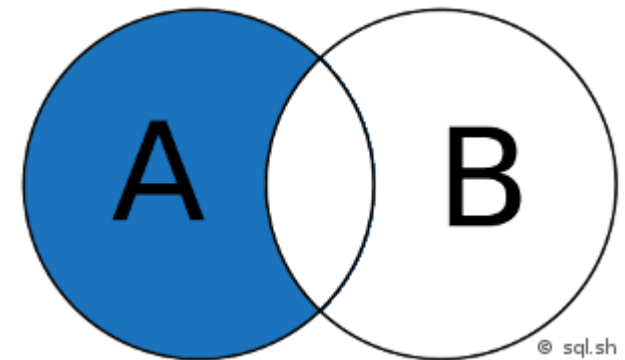
```
LEFT JOIN B ON A.key = B.key ;
```



```
SELECT * FROM A
```

```
LEFT JOIN B ON A.key = B.key
```

```
WHERE B.key IS NULL ;
```



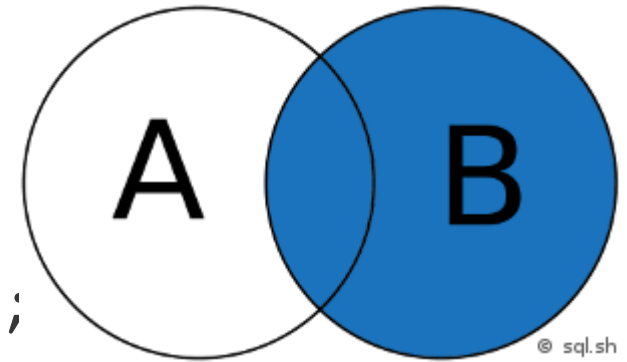
Commandes avancées

Jointures : Requêtes multi-tables

- RIGHT JOIN**

```
SELECT * FROM A
```

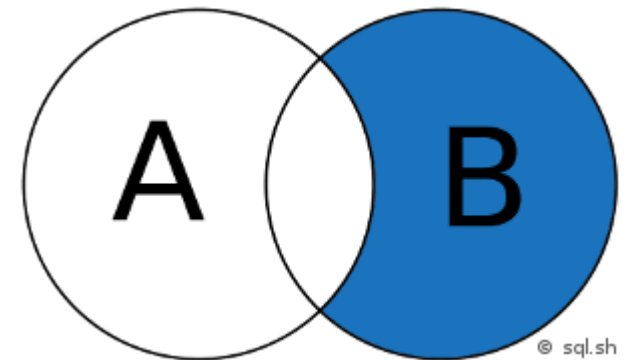
```
RIGHT JOIN B ON A.key = B.key ;
```



```
SELECT * FROM A
```

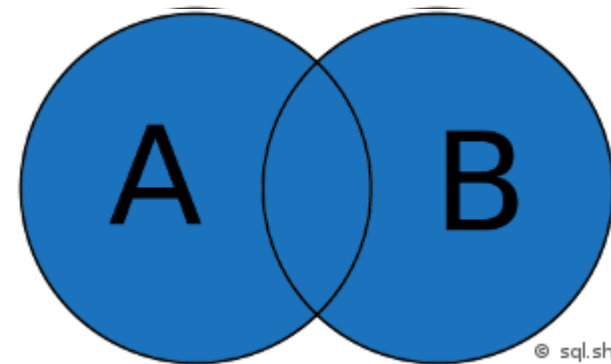
```
RIGHT JOIN B ON A.key = B.key
```

```
WHERE A.key IS NULL ;
```



Jointures : Requêtes multi-tables

- **FULL JOIN**



```
SELECT * FROM A
```

```
FULL JOIN B ON A.key = B.key
```

Présentation des SGBDR

Bases de Données Relationnelles

- **Base de Données avec relations :**
- *Une table est liée à une seconde table, ou plus, par une relation de **clé étrangère**, suivant des contraintes strictes dites « d'intégrité référentielle ».*

Définition de la clé étrangère

Définition des clés :

/ Uniquement MySQL */*

ALTER TABLE [*nomTable1*]

ADD CONSTRAINT *nomDeLaContrainte* (**ex**
FK_table_name_nomColonne)

FOREIGN KEY *nomTable1* (*id_table1*)

REFERENCES [*nomTable2*] (*id_table2*) ;

Contraintes d'intégrité référentielle

Définition des actions en chaîne :

... **ON UPDATE** [*value*] **ON DELETE** [*value*]

- **CASCADE** :- *DROP TABLE nom_table CASCADE CONSTRAINTS;*
- **SET NULL**
- **SET DEFAULT**
- **RESTRICT**

Diagram database



- ✓One-to-one relationships
- ✓One-to-many relationships
- ✓Many-to-many relationships
- ✓Mandatory or not?

Utilisation avancée

Les requêtes imbriquées

Traiter une requête en fonction d'une autre :

```
SELECT * FROM genre  
WHERE id_genre [NOT] IN (  
    SELECT DISTINCT id_genre  
    FROM livre  
) ;
```

Les vues : créer / utiliser

Une Vue est une table virtuelle issue d'une requête SQL

- ✓ d'éviter de taper une requête très longue : la vue sert à donner un nom à la requête pour l'utiliser souvent,.
- ✓ de masquer certaines données à certains utilisateurs .

Les vues : modifier / supprimer

• **CREATE** **VIEW** *view_nomDeLaVue* **AS**

SELECT * FROM *nomDeLaVue* ;

• **Modifier :**

ALTER VIEW *nomDeLaVue* **AS**

SELECT ;

• **Supprimer :**

DROP VIEW *nomDeLaVue* ;

Les vues : limites

- **Toujours une requête de sélection**
- **Une vue : une requête (possibilité d'imbriquer)**

Les procédures stockées (*approche*)

- Ensemble d'instruction de manipulation de données
- (**INSERT, UPDATE, DELETE**) qui peuvent être exécutés par un simple appel.
- *L'appel diffère d'un SGBD à l'autre :*
(*EXEC –SQL Server, CALL -MySQL, ...*)

Les procédures stockées (*approche*)

- **Exemple de procédure stockée :**
DELIMITER //
CREATE PROCEDURE nomProcedure()
BEGIN
 ...Requête SQL
END//

Les optimisations

- **Vérifier la charges des requêtes**
- **Sauvegarder les données ? Archiver ?**

Utilisation avancée

Les optimisations

L'outil SQL Explain :- *The EXPLAIN* statement provides information about how MySQL executes statements. ... EXPLAIN returns a **row of information** for each table used in the SELECT statement.

EXPLAIN select * from nom_table


id	select_type	table	type	possible_keys	key
1	SIMPLE	instructor	ALL	NULL	NULL
1	SIMPLE	grade	ALL	NULL	NULL

key_len	ref	rows	Extra
NULL	NULL	5	Using where; Using temporary; Using filesort
NULL	NULL	6	Using where; Using join buffer

Utilisation avancée

Les optimisations

- ✓Charge de la requête SQL
- ✓Interpréter les indicateurs Explain



key_len	ref	rows	Extra
NULL	NULL	5	Using where; Using temporary; Using filesort
NULL	NULL	6	Using where; Using join buffer

Sauvegarder les données

✓mysqldump

--user=mon_user

--password=mon_password

--all-databases > ALLBD_Backup.sql

✓mysqldump

✓--user=mon_user

✓--password=mon_password

✓--databases nom_de_la_base > BD_name_Backup.sql

Restaurez les données

✓Mysql

✓ --user=user_name

✓ --password=password < file_Backup.sql

✓Mysql

✓--user=user_name

✓--password=password database_name <
file_Backup.sql

Compresser la sauvegarde

```
✓mysqldump <commandes> | gzip >  
archive_file_name.sql.gz
```

SQL



Plus d'informations sur <http://www.dawan.fr>
Contactez notre service commercial au **0800.10.10.97** (prix d'un appel local)