

**INSIA**  
**Bases de données**  
**ING 1**  
**MySQL – Cours et TP 01**

**LA BIBLE : MySQL 5.0 Reference Manual**  
**<http://dev.mysql.com/doc/refman/5.0/fr/index.html>**

Site officiel MySql : <http://www-fr.mysql.com/>  
Documentation MySQL : <http://mysql.org/>  
*La Base de Données Open Source la plus Populaire au Monde*  
**Bertrand LIAUDET**

## SOMMAIRE

<b>SOMMAIRE</b>	<b>1</b>
<b>MODÉLISATION : MODELE RELATIONNEL</b>	<b>3</b>
<b>1. La modélisation</b>	<b>3</b>
<b>2. Le modèle relationnel</b>	<b>3</b>
Présentation	3
Table, tuple, attribut, clé primaire	4
Schéma de la BD	5
Définition de la BD	5
<b>SQL : CREATION DE LA BASE DE DONNÉES</b>	<b>6</b>
<b>1. Manipulation des bases de données</b>	<b>6</b>
Consultation des BD du SGBD	6
Les bases de données pré-installées	6
Création d'une BD	6
Destruction d'une BD	7
Utilisation d'une BD	7
Connaître la BD actuellement utilisée	7
Consultation des tables d'une BD	7
Consultation du code de création des tables enregistrées par le SGBD	8
Consultation des attributs d'une table	8

Consultation des tuples d'une table	8
<b>2. Création des tables</b>	<b>8</b>
<b>3. Création des tuples</b>	<b>10</b>
<b>5. Moteurs MyIsam et InnoDB</b>	<b>11</b>
<b>6. Le fichier de configuration : my.ini</b>	<b>13</b>
my.ini	13
Quelques variables paramétrable du fichier my.ini	13
Quelques paramétrages du fichier my.ini	13
<b>7. Consultation des données sur le disque</b>	<b>14</b>
Moteur MyISAM	14
Moteur InnoDB	14
<b>8. Sauvegarde</b>	<b>16</b>
C:\mysql dump	16
<b>9. Les utilisateurs</b>	<b>17</b>
Utilisateur courant	17
Tous les utilisateurs	17
<b>10. Messages du serveur</b>	<b>17</b>
<b>TP N°1 : PRISE EN MAIN DU CLC-MYSQL</b>	<b>18</b>
<b>Exercice 01 : Exercices post installation - suite</b>	<b>18</b>
<b>Exercice 02 : les utilisateurs</b>	<b>18</b>
<b>Exercice 03 : Créer la base employés</b>	<b>18</b>

Première édition : septembre 2007

Deuxième : septembre 2008

Troisième : septembre 2009

# MODÉLISATION : MODELE RELATIONNEL

## PRINCIPALES NOTIONS

Modèle relationnel	Clé primaire
Table	Clé significative
Tuple	Schéma de la BD
Attribut	NULL

## 1. La modélisation

**La modélisation** est l'activité qui consiste à produire un modèle.

**Un modèle** est ce qui sert ou doit servir d'objet d'imitation pour faire ou reproduire quelque chose.

On s'intéresse ici à la modélisation des données.

Un modèle des données est une représentation de l'ensemble des données. Cette représentation prend en compte un outil de représentation (un langage) et un niveau de précision (des contraintes méthodologiques).

Il existe plusieurs modèles de représentation des données : hiérarchique, relationnel, entité-association, objet, ensembliste, etc.

Les deux modèles dominant actuellement sont : le **modèle relationnel : MR** (qui correspond aux SGBD-R) et le **modèle entité-association : MEA** (qui est indépendant du type de SGBD utilisé). Ces deux modèles correspondent à 2 langages différents.

Les schémas entité-relation et les diagrammes de classe UML peuvent être utilisés comme autres langages à peu près équivalents au MEA.

La méthode MERISE, utilisée quasi-exclusivement en France, distingue entre 3 types de modèles selon des critères méthodologique : le **modèle conceptuel des données : MCD**, le **modèle logique des données : MLD** et le **modèle physique des données : MPD**. L'usage tend à rendre équivalents **MCD et MEA**, **MLD et MR**, **MPD et SQL**.

La « jungle » des modèles !			
<b>Méthode</b>	MCD	MLD	MPD
<b>Langage</b>	MEA, schema E-R, UML	MR	SQL

## 2. Le modèle relationnel

### Présentation

Le modèle relationnel a été inventé par Codd à IBM-San Jose en 1970.

C'est un modèle mathématique rigoureux basé sur un concept simple : celui de relation (ou table, ou tableau).

Ce modèle, c'est celui qui est implanté dans les SGBR-R.

Il permet à la fois de fabriquer la BD et de l'interroger.

## Table, tuple, attribut, clé primaire

### Exemple traité

Un service de ressource humaine dans une entreprise veut gérer le personnel. Dans un premier temps, on veut pouvoir connaître le nom, la fonction, la date d'entrée, le salaire et la commission (part de salaire variable) de chaque employé.

Chaque employé a donc les caractéristiques suivantes :

Nom, fonction, date d'entrée, salaire, commission

### Table, tuples et attributs

Pour ranger ces données, on peut faire un tableau à 5 colonnes :

RELATION 5 attributs :

Employé	Nom	Fonction	Date d'entrée	Salaire	Commission
4 tuples :	TURNER	SALESMAN	8-SEP-81	1500	0
	JAMES	CLERK	3-DEC-81	950	NULL
	WARD	SALESMAN	22-FEB-81	1250	500
	TURNER	ANALYST	3-DEC-81	3000	NULL

### Vocabulaire

**Relation** = tableau = table = classe = ensemble = collection

**Tuple** = ligne du tableau = élément = enregistrement = individu = objet = donnée

**Attribut** = colonne du tableau = caractéristique = propriété = champ

**BD** = toutes les lignes de toutes les tables

### NULL

NULL est la seule information codée qu'on rentre dans une table : elle signifie « non renseignée ». La valeur « 0 », par contre, ne signifie pas du tout « non renseignée », mais bien « valeur = 0 », comme on dirait « valeur = 500 ».

### Clé primaire

On souhaite pouvoir distinguer facilement chaque ligne d'une autre ligne. Or, certains employés ont le même nom.

Pour distinguer chaque ligne, on introduit la notion de clé primaire.

La clé primaire est un attribut qui détermine tous les autres.

Une clé primaire est toujours renseignée.

Exemple type de clé primaire : le numéro de sécurité sociale dans un tableau de personne. Quand on connaît le numéro de sécurité sociale, on sait de qui on parle, donc tous les attributs sont déterminés (même si on ne connaît pas leur valeur à un instant donné).

Dans le tableau des employés, la clé primaire pourrait être un numéro de référence choisi par l'entreprise. On le nomme NE (pour Numéro d'Employé).

## RELATION

6 attributs :

Employés	NE	Nom	Fonction	Date d'entrée	Salaire	Commission
4 tuples :	1	TURNER	SALESMAN	8-SEP-81	3000	0
	2	JAMES	CLERK	3-DEC-81	1800	NULL
	3	WARD	SALESMAN	22-FEB-81	2500	500
	4	TURNER	ANALYST	3-DEC-81	5000	NULL

### Clé secondaire

Une clé secondaire est un attribut qui pourrait être clé primaire, mais qui ne l'est pas.

Par exemple, dans le tableau des employés, on pourrait avoir le numéro de sécurité social. Cet attribut détermine tous les autres. Si on garde le numéro d'employé comme clé primaire, le numéro de sécurité sociale est alors clé secondaire.

En l'occurrence, on a tout intérêt à ne pas faire du numéro de sécurité sociale la clé primaire car on peut imaginer que l'employé existe sans que cette information soit renseignée.

Une clé secondaire peut ne pas être renseignée.

### Clé significative

La clé significative, c'est l'attribut qui sert de clé dans le langage ordinaire. Dans le cas des employés, c'est leur nom. Toutefois, il peut y avoir des homonymes : la clé significative est utile dans le langage ordinaire pour savoir de qui on parle, mais elle est insuffisante dans le langage mathématique pour garantir l'identification de l'individu.

## Schéma de la BD

### Schéma des tables et schéma de la BD

Le schéma d'une table consiste à écrire la table sur une ligne avec les noms de code des attributs:

**EMPLOYES**(NE, nom, fonction, dateEmb, sal, comm)

L'ensemble des schémas des tables forme le schéma de la BD.

### Formalisme

- La clé primaire est notée en premier et est soulignée.
- La table des employés représente une réalité physique. On l'appelle « **table-nom** ».
- Le nom donné à une table-nom est un nom commun, au pluriel.
- La clé primaire d'une table-nom est N (pour numéro) suivi de la première lettre du nom de la table.

## Définition de la BD

Une BD c'est un ensemble de tables avec leurs tuples.

Un SGBD gère plusieurs BD distinctes.

# SQL : CREATION DE LA BASE DE DONNÉES

## PRINCIPALES NOTIONS

Show databases	Use
Create database	Desc
Drop database	CREATE TABLE
Show tables from	INSERT INTO

## 1. Manipulation des bases de données

### Consultation des BD du SGBD

La BD est un ensemble de tables. C'est une sorte de dossier contenant des tables.

Le SGBD permet d'accéder à la liste de toutes les BD qu'il contient :

```
Show databases1;
```

### Les bases de données pré-installées

#### 3 bases de données pré-installées :

- information\_schema
- mysql
- test

#### Dictionnaire des données

information\_schema et mysql sont des bases systèmes qui correspondent à ce qu'on appelle le dictionnaire des données.

Ces deux bases contiennent des méta-données : données sur les données.

#### La base de test

MySQL installe une base de test vide. On peut l'utiliser pour des tests. On peut aussi la supprimer.

### Création d'une BD

La création d'une base de donnée consiste seulement à créer une sorte de dossier qui contiendra les futures tables de la BD :

---

1 Remarques sur le métalangage utilisé : il ne prétend pas être parfaitement formel! Son objectif est d'associer pédagogie et rigueur formelle. Les mots clés du langage SQL sont en gras (**Show database**). Les expressions générales et variables sont en italiques (*NomBD*). Si nécessaire, les explications concernant ces expressions générales sont données soit en note, soit dans le texte, soit à travers des exemples. Les cas particuliers des exemples sont au format standard.

```
Create database NomBD ;
```

*NomBD* est le nom de la BD qu'on veut créer.

On peut aussi vérifier la non-existence de la BD avant de la créer :

```
Create database if not exists NomBD ;
```

Ca évite un signal d'erreur si la BD existe.

### Destruction d'une BD

La destruction d'une base de donnée consiste à détruire le dossier et toutes les tables de la BD. C'est à manier avec prudence :

```
Drop database NomBD ;
```

On peut aussi vérifier l'existence de la BD avant de la détruire :

```
Drop database if exists NomBD ;
```

Ca évite un signal d'erreur si la BD n'existe pas.

### Utilisation d'une BD

Pour pouvoir manipuler les tables d'une BD, il faut d'abord signaler qu'on l'utilise :

```
Use NomBD ;
```

#### ➤ *Remarque* :

On peut accéder à toutes les tables en préfixant le nom de leur BD : *nomBD.nomTable*

### Connaître la BD actuellement utilisée

Pour savoir dans quelle BD on se trouve :

```
Select database() ;
```

### Consultation des tables d'une BD

On peut afficher la liste des tables contenues dans une BD :

```
Show tables from NomBD ;
```

Ou, si la BD est utilisée :

```
Show tables ;
```

## Consultation du code de création des tables enregistrées par le SGBD

### ➤ *Principe*

Quand on envoie une commande de création de table, le SGBD enregistre la structure de la table en mémoire.

Il enregistre aussi le code de création de la table.

On peut afficher ce code.

### ➤ *Syntaxe MySQL*

```
SHOW CREATE TABLE NomTable ;
```

## Consultation des attributs d'une table

On peut afficher la liste des attributs d'une table :

```
Desc NomTable ;
```

utilise.

## Consultation des tuples d'une table

### Syntaxe SQL

```
SELECT liste d'attributs FROM NomTable;
```

### Exemples : tous les attributs

```
SELECT * FROM employes ;
```

### Exemples : certains les attributs

```
SELECT NE, nom FROM employes ;
```

## 2. Création des tables

### Syntaxe SQL

```
CREATE TABLE NomTable (  
    attribut_1 type [contrainte d'intégrité],  
    attribut_2 type [contrainte d'intégrité],  
    ... ,  
    attribut_n type [contrainte d'intégrité]  
);
```

la contrainte est facultative.

La table est une coquille vide : c'est une structure. Elle permettra ensuite de créer des tuples.

### **Exemple**

Création de la table des employés :

```
CREATE TABLE EMPLOYES (  
  NE          integer primary key auto_increment,  
  NOM         varchar(10),  
  JOB         varchar(9),  
  DATEMB     date,  
  SAL         float(7,2),  
  COMM       float(7,2),  
  ND         integer not null  
);
```

### **Les types des données**

Les types des données correspondent pour la plupart aux types habituels des langages de programmation.

Pour avoir la liste et la signification de tous les types :

<http://dev.mysql.com/doc/refman/5.0/fr/column-types.html>

### 3. Création des tuples

#### Syntaxe SQL

```
INSERT INTO NomTable VALUES (val_att_1, ... , val_att_n)
```

On passe la liste des valeurs pour chaque attribut du tuple, dans l'ordre de la déclaration dans la table.

#### Exemples

```
INSERT INTO EMPLOYES VALUES (7839, 'KING', 'PRESIDENT', '1981-11-17', 5000, NULL, 10);  
INSERT INTO EMPLOYES VALUES (7698, 'BLAKE', 'MANAGER', '1981-05-1', 2850, NULL, 30);  
INSERT INTO EMPLOYES VALUES (7782, 'CLARK', 'MANAGER', '1981-06-9', 2450, NULL, 10);
```

## 5. Moteurs MyIsam et InnoDB

### Notion de « moteur »

MySQL offre plusieurs « moteurs » pour gérer les tables. Les deux principaux sont : MyISAM et InnoDB.

Un moteur peut être vu comme un SGBD particulier pour gérer les tables.

### Moteur par défaut : « storage engine » et « table type »

```
Show variable like '%engine%';
```

Ou bien

```
Show variable like '%type%';
```

Les variables « storage\_engine » et « table\_type » sont équivalentes.

### Moteur d'une table

#### ➤ *Par défaut*

Quand on fait un CREATE TABLE, le moteur associé est celui par défaut

#### ➤ *Explicitement*

A la création de la table, on peut préciser le moteur :

```
CREATE TABLE NomTable (  
    ...  
) ENGINE MyISAM;
```

On peut écrire ENGINE ou TYPE.

### Modification du moteur par défaut

#### ➤ *Par modification du fichier de configuration*

On peut préciser dans le fichier « my.ini » :

```
[mysqld]
```

```
storage_engine = 'MyISAM' ;
```

Il faut alors relancer le serveur pour que la modification soit prise en compte.

#### ➤ *Par modification dynamique des variables du serveur*

```
Set @@table_type = 'myisam';
```

Est équivalent à :

```
Set @@local.table_type = 'myisam';
```

Est équivalent à :

```
Set @@session.table_type = 'myisam';
```

Est équivalent à :

```
Set local table_type = 'myisam';
```

La modification est prise en compte par le client qui a lancé la commande.

En remplaçant « local » par « global » :

```
Set global table_type = 'myisam';
```

Est équivalent à:

```
Set @@global.table_type = 'myisam';
```

La modification sera prise en compte par tout nouveau client : elle n'est donc pas prise en compte par le client qui a envoyé la commande.

### **Choix d'un moteur**

Le moteur MyISAM est le moteur d'origine. Il est très permissif : il ne gère pas les contraintes d'intégrité référentielle.

Le moteur InnoDB gère l'intégrité référentielle. On n'utilisera toujours ce moteur pendant ce cours.

## 6. Le fichier de configuration : my.ini

### my.ini

Le fichier my.ini permet de configurer le serveur.

Il permet d'affecter des variables du serveur.

L'installation propose d'autres exemples de fichiers de configuration.

### Quelques variables paramétrables du fichier my.ini

#### Langue

```
Mysql > show variables like '%lang%' ;
```

#### Moteur par défaut

```
Mysql > show variables like '%type%' ;
```

#### Toutes les variables

```
Mysql > show variables;
```

### Quelques paramètres du fichier my.ini

#### Passer en français

```
[mysqld]  
language = French
```

#### Fonctionnement normalisé du « group by »

```
[mysqld]  
sql_mode=ONLY_FULL_GROUP_BY
```

#### Moteur InnoDB par défaut

```
[mysqld]  
default-table-type=InnoDB
```

## 7. Consultation des données sur le disque

Selon le moteur choisi, MySQL enregistre différemment les données sur le disque.

### Moteur MyISAM

#### Répertoire de stockage : le « datadir »

La commande :

```
mysql > show variables like 'datadir';
```

ou

```
mysql > Show variable like '%dir%';
```

Permet de consulter la valeur du « datadir » ;

#### Organisation des données

Chaque BD est rangée dans le DATADIR et dans un répertoire du nom de la BD.

Chaque table d'une BD MyISAM est stockée sur disque dans trois fichiers.

Les fichiers portent le nom de la table et ont une extension qui spécifie le type de fichier.

Il faut éviter de manipuler directement ces fichiers.

#### Structure de la base : db.opt

La commande :

```
Create database nomBase ;
```

Crée le répertoire « *nomBase* » dans le datadir et un fichier « db.opt » dans ce répertoire.

#### Fichier de description des tables : nomTable.FRM

La commande :

```
Create table nomTable (...);
```

Crée dans un fichier .frm qui décrit la structure de la table.

La commande crée aussi deux fichiers pour les données : un fichier « .myd » pour les tuples et un fichiers « .myi » pour les index.

Au départ, ces fichiers sont vides.

#### Fichier de données et d'index : nomTable.MYD et nomTable.MYI

La commande

```
Insert into table nomTable values (...);
```

Remplit les fichiers MYD et MYI correspondants à la table.

### Moteur InnoDB

#### Répertoire de stockage : « InnoDB data home dir » ou « datadir »

La commande :

```
Mysql> Show variable like 'InnoDB_data_home_dir';
```

ou

```
Mysql> Show variable like '%dir%';
```

Permet de consulter la valeur du « InnoDB\_data\_home\_dir ».

Si la valeur n'est pas renseignée, le répertoire de stockage correspond à la valeur du DATADIR

### **Fichier de description des tables : nomTable.frm**

La commande :

```
Create table nomTable (...);
```

Crée dans un fichier .frm qui décrit la structure de chaque table dans le répertoire « InnoDB\_data\_home\_dir ».

La commande met à jour le fichier « ibdata1 »

### **Fichier de données et d'index : ibdata1**

La commande :

```
Insert into table nomTable values (...);
```

Met à jour le fichier « ibdata1 » du répertoire « InnoDB\_data\_home\_dir ».

### **Fichier de log : ib\_logfile0**

Un fichier de log est aussi mis à jour.

## 8. Sauvegarde

### C:\ mysql dump

<http://dev.mysql.com/doc/refman/5.0/fr/mysqldump.html>

Mysqldump est une commande qui permet de produire le code SQL permettant de recréer entièrement la BD.

### Trois usages de mysqldump

➤ *Sauvegarder des BD*

```
C:\ mysqldump [options] --databases DB1 [DB2 DB3...]
```

➤ *Sauvegarder toutes les BD*

```
C:\ mysqldump [options] --all-databases
```

### Affichage à l'écran ou redirection dans un fichier

```
C:\ mysqldump -uroot -p nomBD
```

La commande affiche le code SQL de la BD nomBD.

```
C:\ mysqldump -uroot -p nomBD > nomFichier
```

La commande écrit le code SQL de la BD nomBD dans le fichier nomFichier.

### Usage courant de MYSQLDUMP pour sauvegarder et recharger une BD

Commande usuelle de sauvegarde d'une BD :

```
C:\ mysqldump --opt nomBD > backupFile.sql
```

L'option --opt inclut l'option --quick et l'option --lock-tables, entre autres.

Pour recharger le fichier de sauvegarde :

```
C:\ mysql nomBD < backupFile.sql
```

### Usage courant de MYSQLDUMP pour sauvegarder toutes les BDs

Commande usuelle de sauvegarde de toutes les BD :

```
C:\ mysqldump --opt --all-databases > allBD.sql
```

Pour recharger le fichier de sauvegarde de toutes les BD :

```
C:\ mysql < allBDsql
```

### Usage spécial de MYSQLDUMP : ne sauvegarder que le schéma de la BD

-- no-data

## 9. Les utilisateurs

### Utilisateur courant

Quand on se connecte en tant que client, on est utilisateur de la BD. Un utilisateur à un nom et une machine hôte de son application cliente.

Pour connaître le nom de l'utilisateur connecté :

```
mysql > select user();
```

### Tous les utilisateurs

Les utilisateurs de la BD sont listés dans la table « user » de la BD « mysql ».

```
mysql > use mysql  
mysql > Select host, user from user
```

ou

```
mysql > Select host, user from mysql.user
```

## 10. Messages du serveur

Le serveur envoie des messages de validation et des messages d'erreur.

**Il faut analyser les messages du serveur!!!**

En cas d'erreur, le serveur renvoie un n° de ligne : il faut analyser ce numéro.

Exemples de message :

Query ok

Database changed

etc

On passe la liste des valeurs pour chaque attribut du tuple, dans l'ordre de la déclaration dans la table.

# TP N°1 : PRISE EN MAIN DU CLC-MYSQL

## Exercice 01 : Exercices post installation - suite

1. Ouvrir une calculette et afficher le DATADIR et le répertoire des données INNODB
2. Mettre dans le répertoire « bd ing1 » (ou « mysql ») créé sur le bureau au TP 0, ajouter :
  - Un raccourci permettant d'ouvrir le répertoire des données (DATADIR) : Consulter les données.
3. Regarder les variables « language », « sql\_mode », « table\_type ». Utiliser les « % » dans la commande : like '%lan%'
4. Configurer la langue, le full group by et le moteur par défaut dans le fichier my.ini comme précisé dans le cours.
5. Arrêter et redémarrer le serveur.
6. Vérifier la prise en compte des variables.

## Exercice 02 : les utilisateurs

7. Ouvrir une calculette à partir d'une fenêtre DOS ( ...-u... -p..)
8. Afficher l'aide : « help ». Analyser les résultats.

La commande « help » permet de voir quelques commandes utilisables.

On retrouve les résultats sur le site :

<http://dev.mysql.com/doc/refman/5.0/fr/mysql-commands.html>
9. Lister toute les BD du SGBD
10. Aller dans la BD mysql
11. Lister toutes les tables de la BD « mysql »
12. Lister les attributs de la table « user » de la BD « mysql »
13. Afficher tous les utilisateurs : on n'affiche que les attributs « host » et « user ».

## Exercice 03 : Créer la base employés

14. Télécharger le fichier « EmployesTP01.txt » sur le site : <http://bliaudet.free.fr>. Aller dans : INSIA / ING1 / INSIA-ING1-BD-Piscine 01
15. Analyser les principales étapes du script du fichier « EmployesTP01.txt »
16. Chargez la BD du fichier « EmployesTP01.txt » avec un « source ». **Source** *chemin/monScript.sql* ; Le chemin relatif part du répertoire de lancement de la calculette SQL.
17. Chargez la BD du fichier « EmployesTP01.txt » en faisant un « copier-coller ».

18. Un certain nombre d'erreurs apparaissent à l'exécution de « EmployesTP01.txt ». Analyser les messages d'erreur. Trouver les erreurs. Les corriger.
19. Une fois les erreurs corrigées, Afficher le code de création de la table pris en compte par le SGBD. (Show create table...)
20. Quel est le moteur utilisé ?
21. Quel sera la valeur par défaut pour le prochain numéro d'employé ?
22. Regarder où et comment ont été stockées les données du script du TP.
23. Passer le moteur par défaut en MyISAM. avec un Set table\_type...
24. Relancer le script du fichier « EmployesTP01.txt ».
25. Regarder où et comment ont été stockées les données du script du TP.
26. Arrêter le serveur. Arrêtez la calculette SQL. Relancer le serveur, ouvrez une calculette, afficher le moteur par défaut : il doit être « innodb ».
27. Afficher le code de création de la table « employes » : quel est le moteur utilisé ?
28. Recharger le fichier « EmployesTP01.txt » en version corrigé.
29. Faire une sauvegarde (mysqldump) de la base avec les données.
30. Faire une sauvegarde (mysqldump) du schéma de la base uniquement.