Système d'information et base de données

CM2 : Design d'une base de données MySQL

Mickaël Martin Nevot

V2.3.2

Cette œuvre est mise à disposition selon les termes de la licence Creative Commons Attribution – Pas d'Utilisation Commerciale – Partage à l'Identique 3.0 non transposé.

BY-NC-SA

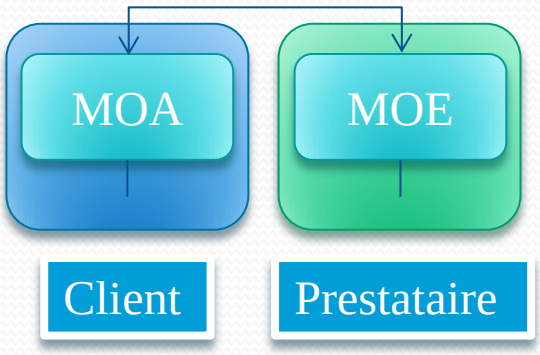
Système d'information et base de données

I. Présentation du cours
II. SI
III. SGBD
IV. Design
V. Droits
VI. Maintenance
VII. Réplication/Sécurité
VIII. Optimisation

BY-NC-SA

Analyse préalable / ressources

- Besoins (rappels) :
  - Expression par le MOA (maître d'ouvrage)
  - Prise en compte par le MOE (maître d'œuvre)
  - Types de besoins :
    - **Fonctionnels** : services offerts
    - **Non fonctionnels** : efficacité, sécurité, utilisation, portabilité, etc.
- Ressources :
  - **Volumétrie**
  - **Accès concurrentiels**



BY-NC-SA

Optimisations à prévoir

- Langage de manipulation de données (LMD):
  - Tables
  - Champs
  - Dépendances
  - Clefs (primaires, étrangères)

Déjà vu

- Langage de description des schémas physiques (LDSP)
  - **Indexation**
  - **Dénormalisation**
  - **Vue matérialisée**
  - Choix de **moteur SQL**
  - Etc.

Optimisations à prévoir dès la phase d'analyse

BY-NC-SA

Index

- Structures de données, physiquement et logiquement indépendantes des données stockées dans la base
- Permet un accès direct (**rapide**) aux enregistrements
- Permet l'**optimisation de requêtes**
- Peut être composite (multi-champs)
- Bonne utilisation :
  - Trouver le meilleur compromis entre :
    - Efficacité des requêtes
    - Coût d'exécution des mises à jour
    - Espace de stockage nécessaire

BY-NC-SA

Cas d'utilisation d'index

- Champs utilisés dans des **conditions de sélection** simples (c'est-à-dire sans `!=`, `IS NULL`, `NOT IN`, `LIKE`, `||`, une fonction de calcul horizontal ou vertical)
- Champs utilisés pour des **jointures** (**clefs étrangères**, etc.)
- **Index composites** plutôt que plusieurs index simples
- Tous les champs ayant une **forte cardinalité**
- Pas pour un champ **volatile**
  (avec une fréquence de mise à jour des données élevée)
- Taille des données indexées importante :
  privilégier les champs de type entier
- Attention à l'ordre de spécification des champs indexés !

BY-NC-SA

Forme normale (normalisation)

- Avantages :
  - Éviter les anomalies transactionnelles
  - Éviter les anomalies de lecture
  - Éviter les incohérences de données
  - Éviter la redondance des données
  - Éviter la contre performance (mises à jour inutiles)
  - **Vérifier la robustesse de la conception**
- Inconvénients :
  - **Lectures souvent trop lentes**
  - Fragilité des données (puisque non redondance)
  - Manque de flexibilité dans l'utilisation de l'espace disque

BY-NC-SA

Dénormalisation

- Dénormalisation ≠ anormalisation
- Pour éviter les inconvénients de la normalisation
- Requêtes utilisant des **jointures** fréquemment
- Les **bases de données en lecture seule**

Car cela permet de s'affranchir d'incohérence de données

User	
Id	Name
1	Toto
2	Tata

Purchase	
User_id	Product_id
1	2
2	1

Product	
Id	Name
1	TV
2	HDD

User	
Id	Name
1	Toto
2	Tata

Purchase	
User	Product
Toto	HDD
Tata	TV

Product	
Id	Name
1	TV
2	HDD

BY-NC-SA

Vue matérialisée

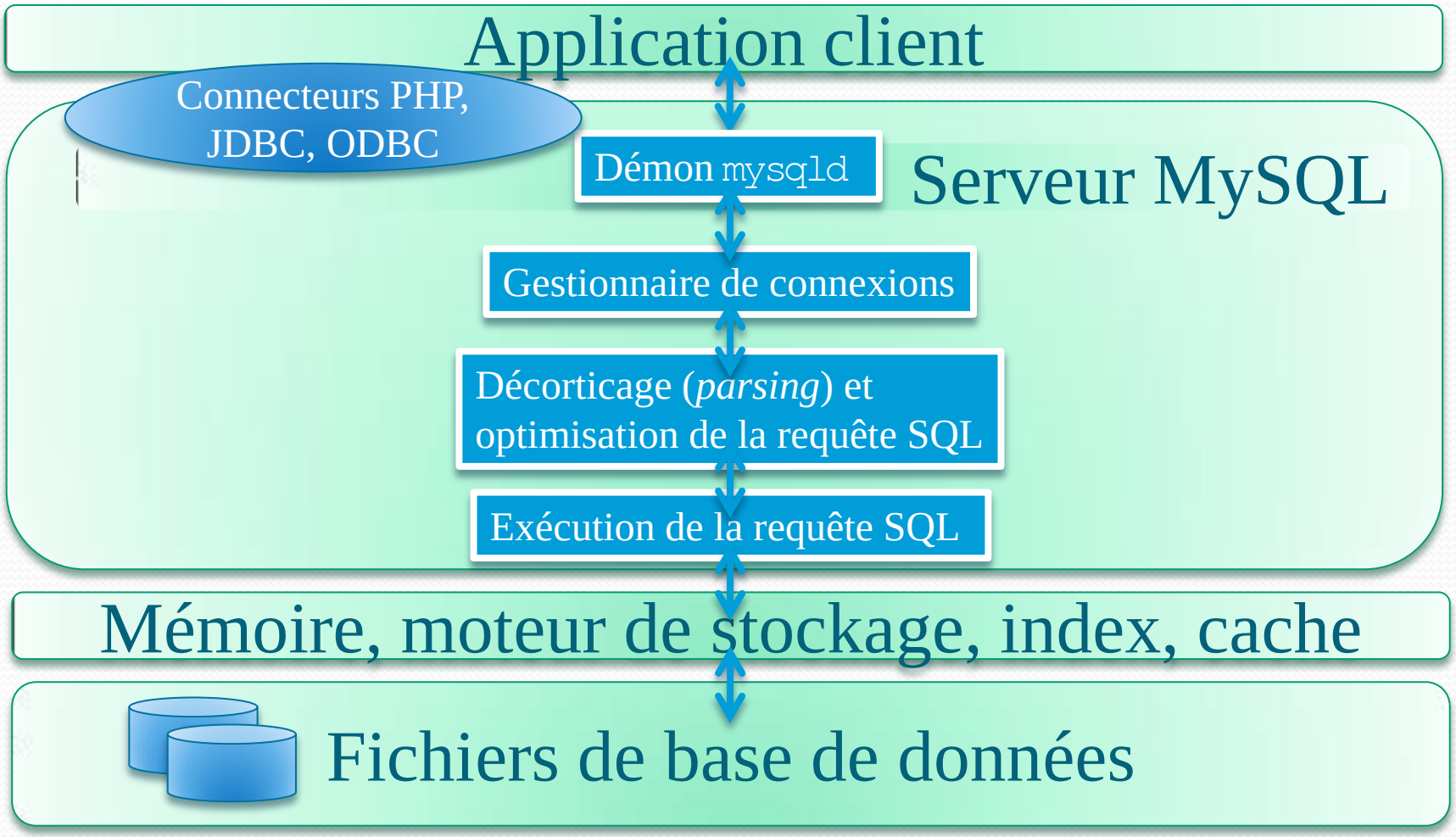
- Vue :
  - Dynamique : table virtuelle définie par le résultat d'une requête
  - Matérialisée : **données dupliquées** physiquement
- Temps de latence : fréquence de duplication

Il est important de prendre ce facteur en compte !

N'existe pas de manière native dans MySQL !

(cc) BY-NC-SA

Architecture MySQL



BY-NC-SA

Moteur/type de tables MySQL

MyISAM

Le plus utilisé

- Type : Non Transactionnel
- Application :
  - Recherche **plein texte**
  - Table en **lecture seule**
  - Table de *« log »*
- Stockage (répertoire `/Data/`) :
  - `myTable.frm` : fichier de définition de la table
  - `myTable.MYD` : fichier contenant les données de la table
  - `myTable.MYI` : fichier d'index

InnoDB

Par défaut à partir de MySQL 5.5

- Type : **Transactionnel**
- Application :
  - Fiabilité de l'information
  - Gestion des transactions
- Stockage :
  - Toutes les données de toutes les tables de toutes les bases sont stockées dans un espace de tables commun spécifique à InnoDB (rigide)

BY-NC-SA

Moteur/type de tables MySQL

MyISAM

Le plus utilisé

- Avantages :
  - Moteur rapide
  - Possibilité d'écrire et lire en même temps sans risque de verrouillage de table
  - Verrouillage de table manuel
  - La mise en cache des clefs
  - Gain de place sur disque
  - Gain de mémoire lors des mises à jour
  - Gestion de la recherche **plein texte**
  - Couramment disponible chez les hébergements mutualisés
- Inconvénients :
  - Pas de gestion des contraintes de clefs étrangères
  - Pas de gestion de transactions (pas de `COMMIT`/`ROLLBACK` possibles)

InnoDB

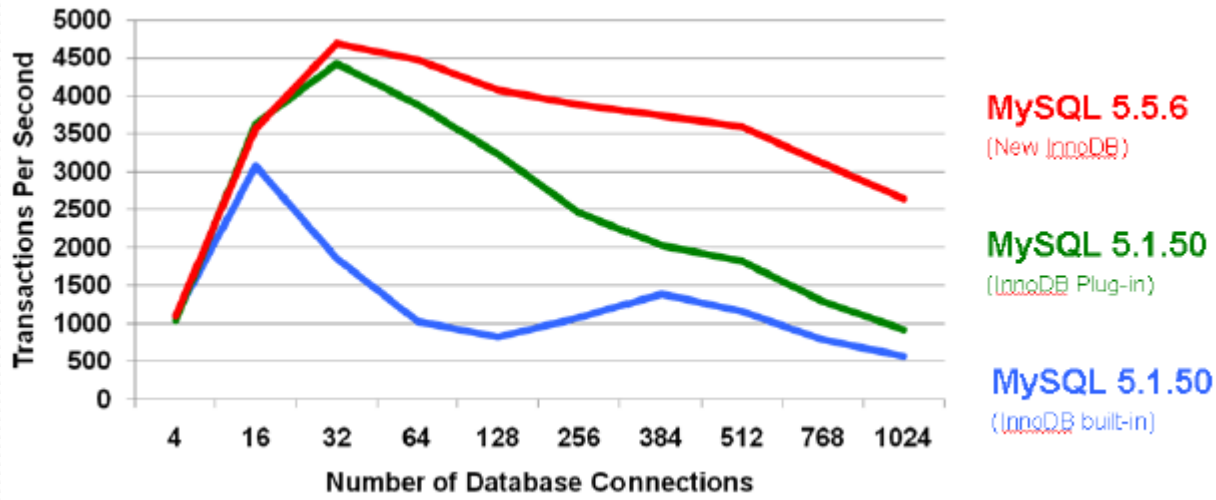
Par défaut à partir de MySQL 5.5

- Avantages :
  - Verrouillage de ligne
  - Gestion de transactions
  - Gère les gros volumes de données
  - Gestion des clefs étrangères
  - Grande panoplie d'éléments de configuration du moteur
  - Gestion de *backups* sans bloquer une base en production
  - Couramment disponible chez les hébergements mutualisés
- Inconvénients :
  - Lenteur de certaines opérations telles que `SELECT COUNT(*) FROM myTable`
  - `TRUNCATE` est synonyme de `DELETE`
  - Les statistiques sont des estimations

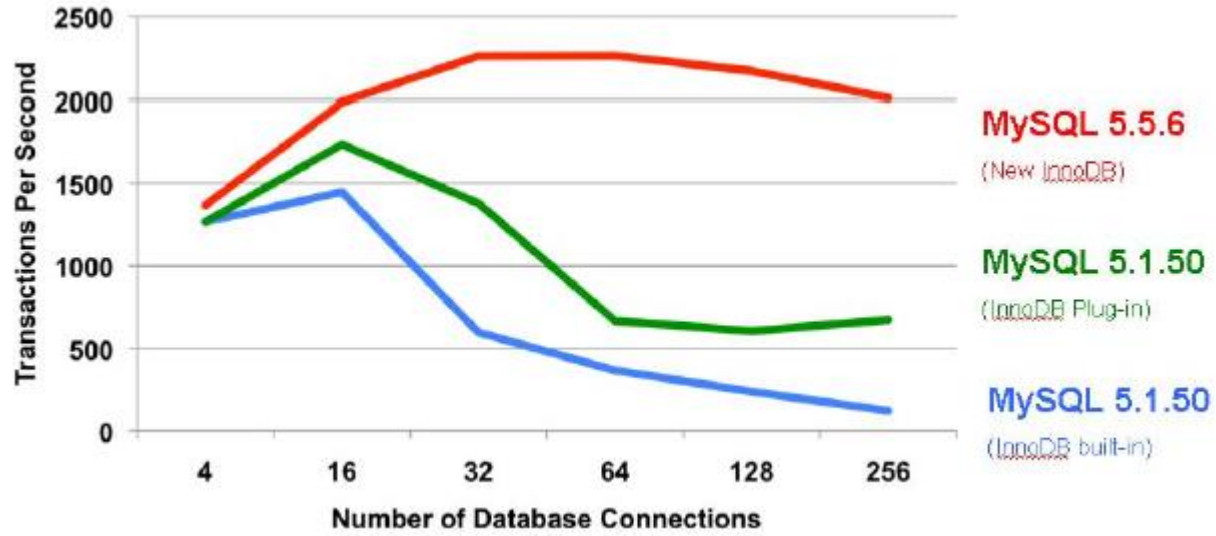
BY-NC-SA

InnoDB et MySQL 5.5

- Linux



- Windows



BY-NC-SA

Recherche plein texte

Spécifique à MyISAM

- Un des plus grands avantages de MyISAM
- Approche d'un vrai moteur de recherche (comme Google)
- Possibilité d'utiliser les expressions régulières
- Permet une « recherche en aveugle »
- Remplace `LIKE` avantageusement
- Principe : examiner tous les **mots** (complets, pas une simple série de caractères) de chaque document enregistré dans un **index *ad hoc*** (de type `FULLTEXT`) et essayer de les faire correspondre à ceux fournis

Aussi appelée recherche en texte intégral, de texte libre ou *full text* (en anglais)

BY-NC-SA

Recherche plein texte : exemple

```
CREATE TABLE article (
    id INT(10) UNSIGNED NOT NULL AUTO_INCREMENT,
    titre VARCHAR(250) CHARACTER SET latin1 DEFAULT NULL,
    article TEXT CHARACTER SET latin1,
    PRIMARY KEY (id),
    FULLTEXT KEY titre (titre, article)
) ENGINE=MyISAM AUTO_INCREMENT=0 DEFAULT CHARSET=utf8;

INSERT INTO article (titre, article) VALUES ('MyISAM', 'Ce moteur est une version évoluée
de ISAM avec des extensions en plus....');
INSERT INTO article (titre, article) VALUES ('Memory', 'Les tables de type Memory stockent
les enregistrements dans la mémoire physique...');
INSERT INTO article (titre, article) VALUES ('CSV', 'Ce type de format facilite le
transport entre différentes sources...');
...
```

BY-NC-SA

Recherche plein texte : exemple

```
SELECT *, MATCH (titre, article) AGAINST ('base de données') AS score FROM article ORDER BY score DESC;
```

```
Id | Titre     | Article                                  | Score
7  | Federated | Le moteur Federated permet de déporter...  | 0,21985759722453
6  | Exemple   | Ce type de table est assez particulier...  | 0,15944281721118
8  | InnoDB    | Le moteur transactionnel le plus utilisé... | 0,14023887676722
1  | MyIsam    | Version évoluée d'ISAM avec...             | 0,0686859973869
2  | Memory    | Les tables de type Memory stockent les...  | 0
...
```

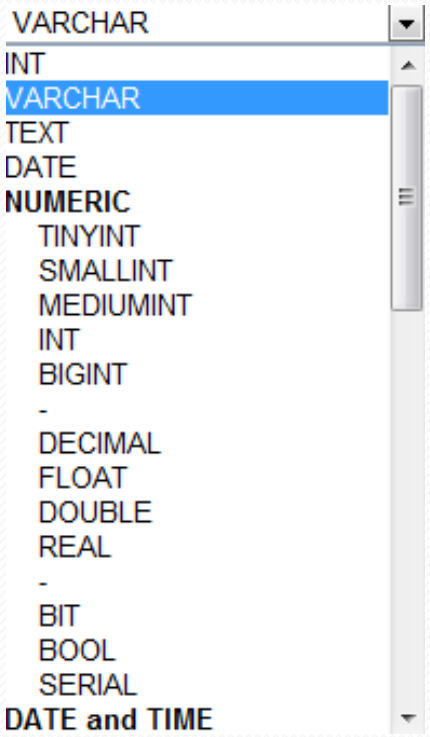
Index `FULLTEXT` : uniquement depuis des champs de types `CHAR`, `VARCHAR` ou `TEXT`

Attention : `MATCH` considère que les mots présents dans au moins la moitié des lignes sont trop fréquents pour être pertinents (donc, par exemple, la recherche plein texte ne renverra aucun résultat si la table n'a pas au moins trois lignes)

BY-NC-SA

Les types de données MySQL

- **Numériques**
- **Caractères**
- **Dates**
- **Binaires**
- **Énuméré**
- **Ensemble**



BY-NC-SA

Les types numériques

Type	Taille mémoire	À la norme (ANSI)
TINYINT	1 octet	✗
SMALLINT	2 octets	✓
MEDIUMINT	3 octets	✗
INTEGER(M) ZEROFILL	4 octets	✓
BIGINT	8 octets	✓
FLOAT(M, D) ZEROFILL	4 octets	✓
DOUBLE {PRECISION}	8 octets	✓
REAL	8 octets	✓
NUMERIC(M, D)	M, (D + 2 si M < D)	✓
DECIMAL(M, D)	M, (D + 2 si M < D)	✓

M,D signifie : numérique de taille maximal M avec D décimales

BY-NC-SA

Les types caractères

- CHAR :
  - Chaîne de taille fixe : 255
  - Complétée/tronquée en fonction de la taille réelle
- VARCHAR(...) :
  - Chaîne de longueur variable
- BLOB (*binary large object*) / TEXT :
  - TINYBLOB/TINYTEXT (taille maximale : 256)
  - BLOB/TEXT (taille maximale : 65536)
  - MEDIUMBLOB/MEDIUMTEXT (taille maximale : 16777216)
  - LONGBLOB/LONGTEXT (taille maximale : 4294967296)

BINARY : sensible à la casse (valable pour CHAR et VARCHAR)

Taille fixe

Taille maximale spécifiée en paramètre

Avec l'attribut BINARY

Longues chaînes

Taille variable

Taille maximale : en nombre de caractères

BY-NC-SA

Quel type de caractères ?

Taille fixe

- Avantages :
  - Accès direct facile (rapide)
  - Pas de fragmentation
  - Réparation facile de table
- Inconvénient :
  - Pas d'économie de place

CHAR

Taille variable

- Avantage :
  - Gain de place
- Inconvénients :
  - Insertion plus longue (car calcul de la taille exacte)
  - Fragmentation inévitable
  - Temps d'accès : moindre performance

VARCHAR/BLOB/TEXT

BY-NC-SA

Les types de date

- `DATE` :
  - Entre `1000-01-01` et le `9999-12-31`
- `TIME` :
  - Entre `00:00:00` et `23:59:59`
- `DATETIME` :
  - Entre `1000-01-01 00:00:00` et `9999-12-31 23:59:59`
- `TIMESTAMP(...)` : ← Longueur de l'affichage, `14` par défaut
  - Nombre de secondes écoulées depuis le 01/01/1970
- `YEAR` :
  - Format 2 : entre `70` et `69` (1970 et 2069)
  - Format 4 : entre `1901` et `2155`

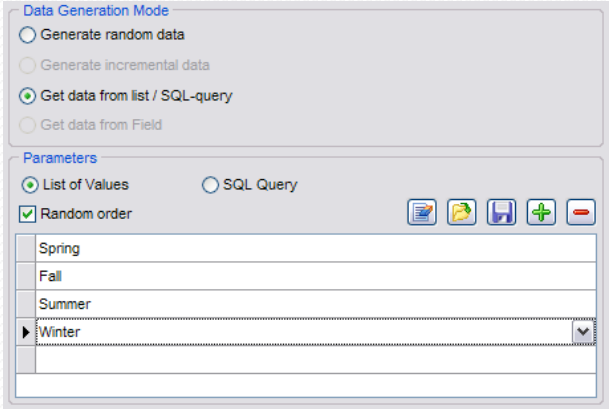
BY-NC-SA

Les types énuméré/ensemble

- ENUM :
  - Type énuméré dont les instances prennent leur unique valeur parmi un ensemble explicitement spécifié :

```
ENUM('blue', 'white', 'red')
```

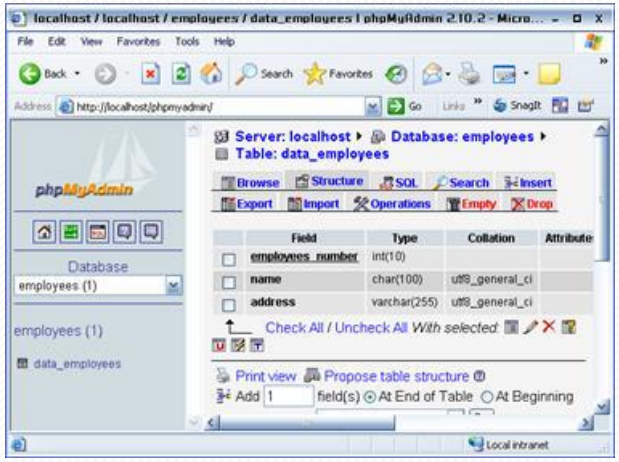
- SET :
  - Similaire à ENUM
  - Peut prendre plusieurs valeurs parmi l'ensemble énuméré



BY-NC-SA

Les commandes MySQL

- Création de BD
- Destruction de BD
- Création des tables
- Destruction des tables
- Modifications de la structure des tables
- Création de clefs étrangères
- Création d'index



BY-NC-SA

Création/destruction de BD

- `CREATE DATABASE` :
  - Crée un répertoire vide dans `/var/lib/mysql` (par défaut)

Syntaxe :

```
CREATE DATABASE [IF NOT EXISTS] db_name [create_specification [,
create_specification] ...]

create_specification: [DEFAULT] CHARACTER SET charset_name | [DEFAULT] COLLATE
collation_name
```

- `DROP DATABASE` :
  - Efface tous les fichiers des tables
  - S'interrompt si la base n'existe pas, sauf si l'option `IF EXISTS` est spécifiée

Syntaxe :

```
DROP DATABASE [IF EXISTS] db_name
```

BY-NC-SA

Création/destruction de tables

- CREATE :

Syntaxe :

```
CREATE [TEMPORARY] TABLE [IF NOT EXISTS] nom_table
[(create_definition,...)] [table_options] [select_statement]
```

- DROP :
  - Efface définitivement le fichier d'une table

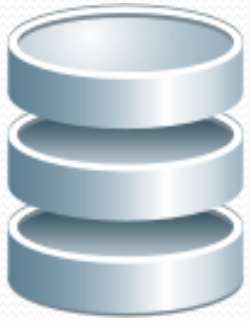
Syntaxe :

```
DROP [TEMPORARY] TABLE [IF EXISTS] tbl_name [, tbl_name] ... [RESTRICT | CASCADE]
```

BY-NC-SA

Modification des tables

- ALTER TABLE : mysql> ALTER TABLE table ACTION;
  - ACTION :
    - ADD COLUMN col CREATION_CLAUSE
    - ADD INDEX index (col1, col2)
    - ALTER col SET DEFAULT value
    - MODIFY col CREATION_CLAUSE
    - DROP INDEX index
    - DROP COLUMN col
    - RENAME tablename
    - Etc.



BY-NC-SA

Clefs étrangères

- **Clef primaire d'une autre table de la BD**
- Avec le moteur InnoDB (essentiellement)
- Contraintes d'intégrités vérifiées par le serveur MySQL :
  - Occasionne une baisse de performance
  - Certaines stratégies placent une partie ou la totalité de la logique de vérification au niveau de l'applicatif (déconseillé)
- `FOREIGN KEY : FOREIGN KEY (fk) REFERENCES table(col)`

BY-NC-SA

Création d'index

- Types d'index :
  - `PRIMARY KEY` : **index à valeur unique non nulle**
  - `KEY` : synonyme de `INDEX`
  - `INDEX` (simple ou composite) :
    - `INDEX index (name, firstname)`
  - `UNIQUE` : **index à valeur unique**

Un seul par table

Pouvant être nulle

- Création/destruction :
  - `CREATE INDEX` :

```
mysql> CREATE INDEX index ON table (col1, col2);
```

  - `DROP INDEX` :

```
mysql> DROP INDEX index ON table;
mysql> ALTER TABLE table DROP INDEX index;
```

CC BY-NC-SA

Aller plus loin

- Moteur Aria (MARIA) : évolution ACID de MyISAM

BY-NC-SA

Liens

- Documents électroniques :
  - http://www.elliptic.fr/doc/mysql
  - http://dev.mysql.com/doc/refman/5.5/en
- Documents classiques :
  - Cours :
    - Maurice Libes. *Administration et exploitation du SGBDR MySQL*.
    - Cyril Gruau. *Conception d'une base de données*.
    - Jean-Marc Petit. *Administration des bases de données*.

BY-NC-SA

Crédits

Auteur

Mickaël Martin Nevot
mmartin.nevot@gmail.com

Carte de visite électronique

Relecteurs

Cours en ligne sur : www.mickael-martin-nevot.com