CoursInfo

MySQL — Bases de Données

Apprenez à concevoir, interroger et sécuriser des bases de données relationnelles avec MySQL et le langage SQL.

Niveau : Intermédiaire • Durée : ~14h • 7 chapitres • Certificat inclus

---

1. Introduction aux bases de données relationnelles

MySQL est un Système de Gestion de Base de Données Relationnelle (SGBDR) open source. Les données sont organisées en tables (comme des feuilles de tableur), chaque table ayant des colonnes (attributs) et des lignes (enregistrements).

Terme	Définition
Table	Ensemble structuré de données sur un même sujet (ex : utilisateurs)
Colonne	Un attribut de la table (ex : email, nom)
Ligne (enregistrement)	Une entrée précise dans la table
Clé primaire	Colonne qui identifie de façon unique chaque ligne

■ SQL : MySQL utilise le langage SQL (Structured Query Language), un standard commun à la plupart des bases relationnelles (PostgreSQL, SQLite, SQL Server...).

2. Créer une base de données et des tables (DDL)

```
CREATE DATABASE
cours_info;

USE
cours_info;


CREATE TABLE
utilisateurs (
id
INT

AUTO_INCREMENT PRIMARY KEY
,
nom
VARCHAR
(
100
)
NOT NULL
,
email
VARCHAR
(
150
)
UNIQUE NOT NULL
,
mot_de_passe
VARCHAR
(
255
)
NOT NULL
,
date_creation
TIMESTAMP

DEFAULT
CURRENT_TIMESTAMP
);
```

■■ Mots de passe : ne stockez jamais un mot de passe en clair. Utilisez password_hash() en PHP avant l'insertion, et VARCHAR(255) pour laisser assez de place au hash.

3. Types de données MySQL

Type	Usage
INT	Nombre entier
DECIMAL(8,2)	Nombre décimal précis (ex : prix, montants)
VARCHAR(n)	Chaîne de texte de longueur variable, max n caractères
TEXT	Long texte (article, message)
DATE / DATETIME	Date seule / date + heure
BOOLEAN	Vrai/faux (stocké comme TINYINT 0/1)

■ Choisir le bon type : un type trop large gaspille de l'espace disque, un type trop étroit tronque vos données. VARCHAR(150) suffit largement pour un email.

4. CRUD : Insérer, lire, modifier, supprimer

```
-- CREATE : ajouter une ligne


INSERT INTO
utilisateurs (nom, email, mot_de_passe)

VALUES
(
'Fatou'
,
'fatou@example.com'
,
'$2y$10...'
);


-- READ : lire des données


SELECT
*
FROM
utilisateurs;

SELECT
nom, email
FROM
utilisateurs
WHERE
id =
1
;


-- UPDATE : modifier une ligne existante


UPDATE
utilisateurs
SET
nom =
'Fatou Diop'

WHERE
id =
1
;


-- DELETE : supprimer une ligne


DELETE FROM
utilisateurs
WHERE
id =
1
```

```
;
```

■■ Toujours un WHERE : un UPDATE ou DELETE sans clause WHERE s'applique à toutes les lignes de la table — vérifiez toujours avant d'exécuter.

5. WHERE, ORDER BY, JOIN et fonctions d'agrégation

```
-- Filtrer et trier


SELECT
*
FROM
messages

WHERE
date_submitted >=
'2026-01-01'


ORDER BY
date_submitted
DESC


LIMIT

10
;


-- Jointure entre deux tables liées


SELECT
utilisateurs.nom, inscriptions.cours_id

FROM
utilisateurs

JOIN
inscriptions
ON
utilisateurs.id = inscriptions.utilisateur_id;


-- Fonctions d'agrégation


SELECT COUNT
(*)
FROM
utilisateurs;

SELECT AVG
(note)
FROM
resultats;
```

Type de JOIN	Retourne
INNER JOIN	Uniquement les lignes qui correspondent dans les deux tables
LEFT JOIN	Toutes les lignes de la table de gauche, correspondance ou non

6. Relations entre tables et clés étrangères

Une base bien conçue évite de dupliquer l'information : on la répartit dans plusieurs tables reliées par des clés étrangères (foreign key).

```
CREATE TABLE
messages (
id
INT

AUTO_INCREMENT PRIMARY KEY
,
utilisateur_id
INT
,
contenu
TEXT

NOT NULL
,

FOREIGN KEY
(utilisateur_id)
REFERENCES
utilisateurs(id)

ON DELETE CASCADE

);
```

ON DELETE CASCADE supprime automatiquement les messages liés si l'utilisateur correspondant est supprimé — évite les données orphelines.

■■ Relations courantes : 1-N (un utilisateur a plusieurs messages) se modélise avec une clé étrangère simple ; N-N (étudiants ↔ cours) nécessite une table de liaison intermédiaire.

7. Sécurité et bonnes pratiques

La faille la plus fréquente sur les sites mal sécurisés est l'injection SQL : insérer du code SQL dans un champ de formulaire pour manipuler la base.

```
-- Champ email rempli avec : ' OR '1'='1


SELECT
*
FROM
utilisateurs
WHERE
email =
''

OR

'1'='1'
;

-- Cette requête retourne TOUS les utilisateurs !
```

- Toujours utiliser des requêtes préparées (voir cours PHP) plutôt que de concaténer des chaînes SQL
- Créer un utilisateur MySQL dédié par application, avec des droits limités (pas root)
- Sauvegarder régulièrement la base avec mysqldump
- Ne jamais exposer les identifiants de connexion dans un fichier accessible publiquement

■ Exercice : Créez une table cours (id, titre, niveau, duree) et une table inscriptions (id, utilisateur_id, cours_id, date_inscription) avec les clés étrangères adéquates. Écrivez la requête qui liste le nom de chaque utilisateur avec le titre de chaque cours auquel il est inscrit.

Exercices & Quiz de vérification

Testez vos connaissances avant de passer au cours suivant. Chaque question a une correction détaillée à dérouler.

1. Quelle instruction permet d'ajouter une nouvelle ligne dans une table ?

- A. UPDATE
- B. INSERT INTO
- C. ADD ROW

Correction : Réponse : B. INSERT INTO table (...) VALUES (...) ajoute un nouvel enregistrement.

2. Que se passe-t-il si on exécute un DELETE sans clause WHERE ?

- A. Rien, MySQL bloque l'opération
- B. Toutes les lignes de la table sont supprimées
- C. Seule la première ligne est supprimée

Correction : Réponse : B. Sans condition, l'instruction s'applique à l'intégralité de la table — d'où l'importance de toujours vérifier son WHERE avant d'exécuter.

3. Quel type de clé garantit qu'une valeur de colonne est unique et identifie une ligne ?

- A. Clé primaire
- B. Clé étrangère
- C. Index secondaire

Correction : Réponse : A. La clé primaire (PRIMARY KEY) identifie de manière unique chaque enregistrement de la table.

4. Qu'est-ce que l'injection SQL ?

- A. Une méthode d'optimisation de requêtes
- B. Une attaque exploitant une entrée utilisateur non filtrée pour manipuler la base
- C. Un type de sauvegarde

Correction : Réponse : B. Elle se prévient avec des requêtes préparées et une validation stricte des entrées utilisateur.

Ressources supplémentaires

- Documentation officielle MySQL (dev.mysql.com)
- MySQL — Prévenir l'injection SQL
- MySQL Tutorial — Exercices pratiques (EN)