



BASES DE DONNÉES : SQL ET NOSQL

# SQL et bases de données relationnelles avec MySQL

Maîtriser SQL pour interroger, manipuler et concevoir des bases de données relationnelles. Indispensable pour tout développeur backend.

DURÉE	TARIF HT	NIVEAU	LANGUE	GROUPE	FORMAT
4j (28h)	<del>1000.00</del> € <b>600.00 €</b>	<b>Débutant</b>	<b>Français</b>	<b>3-12</b>	<b>Formation</b>

## 1 PUBLIC VISÉ

- Débutants souhaitant apprendre à travailler avec des bases de données relationnelles sans aucune expérience préalable.
- Développeurs front-end ou débutants en backend voulant comprendre et manipuler les données côté serveur.
- Personnes en reconversion vers le développement web ayant besoin de maîtriser SQL pour aborder PHP, Java ou Python.
- Chefs de projet, product managers ou analystes amenés à lire et comprendre des données sans passer par un développeur.

## 2 PRÉREQUIS

Aucun prérequis technique. Savoir utiliser un ordinateur et un navigateur web est suffisant pour suivre cette formation. Une curiosité pour la logique et l'organisation des données est un atout.

## 3 OBJECTIFS PÉDAGOGIQUES

- Comprendre le modèle relationnel et le rôle d'un système de gestion de base de données
- Créer des tables avec les types de données adaptés et les contraintes d'intégrité
- Écrire des requêtes SQL de lecture, insertion, modification et suppression
- Filtrer, trier et regrouper des données avec WHERE, ORDER BY et GROUP BY
- Maîtriser les jointures pour interroger plusieurs tables simultanément
- Utiliser les sous-requêtes et les fonctions d'agrégation
- Concevoir un schéma de base de données à partir d'un besoin métier : MCD et MLD
- Optimiser des requêtes simples avec les index
- Utiliser MySQL et phpMyAdmin comme environnement de travail principal



## 4 PROGRAMME DÉTAILLÉ

Jour 1 (7h) – Le modèle relationnel et les bases du langage

Module 1 – Comprendre les bases de données relationnelles (2h)

### Le modèle relationnel (1h)

- Qu'est-ce qu'une base de données ? Pourquoi stocker les données de manière structurée
- Le modèle relationnel : tables, lignes, colonnes, relations
- Les SGBD les plus utilisés : MySQL, PostgreSQL, SQLite, Oracle, SQL Server
- Différence entre SQL et NoSQL : quand choisir l'un ou l'autre
- SQL dans l'écosystème développeur : indispensable pour PHP, Java, Python, Node.js

### Mise en place de l'environnement (1h)

- Installer MySQL avec XAMPP ou Laragon et accéder à phpMyAdmin
- Naviguer dans phpMyAdmin : créer une base de données, explorer les tables
- Se connecter à MySQL en ligne de commande
- Exécuter sa première requête SQL : `SELECT 1+1`
- *Cas pratique : créer une base de données et explorer son interface*

Module 2 – Créer et structurer des tables (2h)

### Types de données et création de tables (1h)

- Les types de données MySQL : INT, BIGINT, VARCHAR, TEXT, BOOLEAN, DATE, DATETIME, DECIMAL
- Choisir le bon type selon les données : taille, précision, performances
- CREATE TABLE : syntaxe complète avec types et contraintes
- Les contraintes : NOT NULL, UNIQUE, DEFAULT, CHECK
- Clés primaires : PRIMARY KEY et AUTO\_INCREMENT
- *Cas pratique : créer les tables d'une application de gestion de contacts*

### Modifier et supprimer des tables (1h)

- ALTER TABLE : ajouter, modifier et supprimer une colonne
- RENAME TABLE : renommer une table
- DROP TABLE : supprimer une table et ses données
- TRUNCATE TABLE : vider une table sans la supprimer
- Les commentaires SQL : documenter son schéma
- *Cas pratique : faire évoluer le schéma de l'application de contacts*

Module 3 – CRUD : lire, insérer, modifier, supprimer (3h)

### Insérer et lire des données (1h30)

- INSERT INTO : insérer un ou plusieurs enregistrements
- INSERT ... SELECT : insérer des données issues d'une autre table
- SELECT : lire toutes les colonnes ou une sélection
- DISTINCT : éliminer les doublons dans les résultats



Les alias : AS pour renommer colonnes et tables dans les résultats

- LIMIT et OFFSET : paginer les résultats
- *Cas pratique : peupler et interroger une table de produits*

### Modifier et supprimer des données (1h30)

- UPDATE : modifier un ou plusieurs enregistrements avec WHERE
- DELETE : supprimer des enregistrements ciblés avec WHERE
- La règle d'or : toujours tester avec SELECT avant UPDATE ou DELETE
- ON UPDATE et ON DELETE : comportement des clés étrangères
- Transactions : BEGIN, COMMIT, ROLLBACK pour sécuriser les opérations critiques
- *Cas pratique : mettre à jour des prix et supprimer des produits expirés*

Jour 2 (7h) - Filtres, agrégations et jointures fondamentales

Module 4 - Filtrer et trier les données (2h30)

### La clause WHERE (1h15)

- Opérateurs de comparaison : =, !=, >, <, >=, <=
- Opérateurs logiques : AND, OR, NOT
- BETWEEN : filtrer dans un intervalle
- IN et NOT IN : filtrer selon une liste de valeurs
- LIKE et NOT LIKE : recherche par motif avec % et \_
- IS NULL et IS NOT NULL : gérer les valeurs manquantes
- *Cas pratique : requêtes de filtrage avancées sur un catalogue de produits*

### Trier et limiter les résultats (1h15)

- ORDER BY : trier par une ou plusieurs colonnes, ASC et DESC
- LIMIT et OFFSET : implémenter une pagination
- CASE WHEN : créer des colonnes calculées conditionnelles
- COALESCE : remplacer les valeurs NULL par une valeur par défaut
- *Cas pratique : top 10 des produits les plus vendus avec pagination*

Module 5 - Fonctions d'agrégation et GROUP BY (2h30)

### Fonctions d'agrégation (1h15)

- COUNT : compter les enregistrements
- SUM : calculer une somme
- AVG : calculer une moyenne
- MIN et MAX : trouver le minimum et le maximum
- Combiner plusieurs fonctions d'agrégation dans une même requête
- *Cas pratique : statistiques de ventes par période*

### GROUP BY et HAVING (1h15)

- GROUP BY : regrouper les résultats par valeur
- HAVING : filtrer les groupes après agrégation



Différence entre WHERE et HAVING : quand utiliser l'un ou l'autre

- GROUP BY avec plusieurs colonnes
- *Cas pratique : chiffre d'affaires par catégorie avec filtre sur le minimum*

Module 6 – Jointures fondamentales (2h)

### Les jointures essentielles (2h)

- Pourquoi les jointures : interroger plusieurs tables liées
- INNER JOIN : retourner uniquement les enregistrements qui correspondent dans les deux tables
- LEFT JOIN : retourner tous les enregistrements de la table gauche même sans correspondance
- RIGHT JOIN : retourner tous les enregistrements de la table droite
- Les clés étrangères : FOREIGN KEY et contraintes référentielles
- Jointures sur plusieurs tables : enchaîner plusieurs JOIN
- *Cas pratique : commandes avec leurs clients, produits et catégories en une seule requête*

Jour 3 (7h) – Jointures avancées, sous-requêtes et vues

Module 7 – Jointures avancées (2h)

### Au-delà des jointures classiques (2h)

- FULL OUTER JOIN : tous les enregistrements des deux tables
- SELF JOIN : joindre une table avec elle-même
- CROSS JOIN : produit cartésien
- Optimiser les jointures : index sur les clés étrangères
- *Cas pratique : arborescence de catégories avec self join*
- Atelier : enchaîner jointures classiques et avancées sur un schéma complet

Module 8 – Sous-requêtes et vues (3h)

### Sous-requêtes (1h30)

- Sous-requête dans WHERE : filtrer selon le résultat d'une autre requête
- Sous-requête dans SELECT : calculer une valeur par ligne
- Sous-requête dans FROM : utiliser un résultat comme table temporaire
- EXISTS et NOT EXISTS : tester l'existence d'enregistrements liés
- Sous-requête corrélée : référencer la table parente
- *Cas pratique : trouver les clients n'ayant passé aucune commande*

### Vues et requêtes communes (1h30)

- CREATE VIEW : encapsuler une requête complexe dans une vue réutilisable
- Avantages des vues : simplification, sécurité, maintenabilité
- CTE (Common Table Expression) avec WITH : structurer les requêtes complexes
- *Cas pratique : vue des commandes en cours avec CTE pour le calcul du total*

Module 9 – Index et optimisation (2h)

### Comprendre et appliquer les index (2h)

Pourquoi les index : accélérer les recherches sur les grandes tables

- Créer un index simple et un index composite
- EXPLAIN : analyser le plan d'exécution d'une requête
- Quand ne pas indexer : coût en écriture et en stockage
- Les index sur les clés étrangères : la règle de base
- *Cas pratique : analyser et optimiser une requête lente avec EXPLAIN*
- Atelier : comparer les temps d'exécution avant et après indexation sur un jeu de données volumineux

Jour 4 (7h) - Conception et projet de synthèse

Module 10 - Concevoir un schéma de base de données (3h)

### **Modélisation conceptuelle et logique (2h)**

- Le MCD (Modèle Conceptuel des Données) : entités, attributs, associations
- Les cardinalités : 1-1, 1-N, N-N et leur traduction en tables
- Le MLD (Modèle Logique des Données) : transformer le MCD en schéma SQL
- Les formes normales : 1NF, 2NF, 3NF — comprendre et appliquer
- Les tables de jointure pour les relations N-N

### **Atelier de conception guidé (1h)**

- Analyser un cahier des charges et identifier les entités
- Dessiner le MCD puis le MLD
- Générer le SQL de création à partir du MLD
- *Cas pratique : concevoir le schéma complet d'une plateforme e-commerce*

Module 11 - Projet de synthèse (4h)

### **Réalisation (2h30)**

- Cahier des charges : base de données d'une plateforme de réservation ou d'un réseau social simplifié
- Étape 1 : modélisation MCD et MLD à partir du cahier des charges
- Étape 2 : création du schéma SQL avec toutes les contraintes
- Étape 3 : insertion de jeux de données réalistes
- Étape 4 : écriture de requêtes métier complexes avec jointures, agrégations et sous-requêtes

### **Revue collective et bilan (1h30)**

- Présentation des choix de modélisation par chaque participant
- Revue collective : qualité du schéma, index manquants, requêtes optimisables
- Retour formateur individualisé sur le projet rendu
- Ouverture : SQL dans les frameworks — Doctrine (Symfony), Eloquent (Laravel), Hibernate (Spring Boot), SQLAlchemy (Python)



## 5 COMPÉTENCES VISÉES

---

- Écrire des requêtes SQL complexes avec jointures, sous-requêtes et agrégations
- Modéliser une base de données relationnelle à partir d'un cahier des charges
- Créer et maintenir un schéma de base de données avec les contraintes d'intégrité
- Identifier et corriger les problèmes de performance sur des requêtes lentes
- Utiliser MySQL en ligne de commande et via une interface graphique

## 6 MODALITÉS PÉDAGOGIQUES

---

Formation délivrée en présentiel ou distanciel (visioconférence). Le formateur alterne entre méthode démonstrative (écriture et exécution de requêtes commentées en direct), méthode interrogative (analyse de schémas existants et identification des problèmes de modélisation) et méthode active (exercices de requêtage et ateliers de conception). L'accent est mis sur la pratique immédiate : chaque notion est illustrée par une requête exécutée en direct sur des données réelles.

## 7 MOYENS PÉDAGOGIQUES ET TECHNIQUES

---

- Support de cours numérique mis à disposition des apprenants
- Scripts SQL de démonstration avec jeux de données réalistes par module
- Environnement de développement : MySQL + phpMyAdmin via XAMPP ou Laragon
- Pour le distanciel : visioconférence (Zoom ou équivalent), partage d'écran, chat en direct
- Accès à la plateforme pédagogique LaPolaris (supports, ressources, émargement)

## 8 MODALITÉS D'ÉVALUATION

---

- En cours de formation : exercices de requêtage corrigés à chaque module
- En fin de formation : conception d'un schéma complet et écriture de requêtes métier sur un cas réel
- Questionnaire d'auto-évaluation des acquis en fin de parcours

## 9 CRITÈRES D'ÉVALUATION

---

- Écriture de requêtes SQL syntaxiquement correctes retournant les résultats attendus
- Utilisation adaptée des jointures selon le type de relation entre les tables
- Modélisation correcte d'un schéma avec clés primaires, clés étrangères et contraintes d'intégrité
- Application des formes normales pour éviter la redondance des données
- Utilisation pertinente des index sur les colonnes fréquemment interrogées

## 10 MODALITÉS DE VALIDATION

---

Attestation de fin de formation délivrée à l'issue du parcours, conditionnée à une assiduité d'au moins 80 % et à la réalisation du projet de synthèse. L'attestation précise les objectifs atteints et les compétences acquises.



## 11 SUIVI ET ACCOMPAGNEMENT

- Feuilles d'émargement signées par demi-journée (présentiel) ou émargement numérique (distanciel)
- Traçabilité des activités pédagogiques réalisées
- Attestation d'assiduité délivrée en fin de formation
- Suivi individuel via les exercices de requêtage corrigés et le projet de synthèse

## 12 CONDITIONS D'ACCÈS

Formation accessible sur inscription directe, sans prérequis administratif particulier. Le financement peut être pris en charge par l'employeur dans le cadre d'un plan de développement des compétences, ou en autofinancement avec possibilité de paiement en plusieurs fois.

## 13 DÉLAIS D'ACCÈS

Inscription possible jusqu'à **5 jours ouvrés** avant le début de la session. Pour toute demande urgente, nous contacter directement.

## S PROCHAINES SESSIONS

**01/06/2026 → 04/06/2026**

À distance

12 places restantes

**08/06/2026 → 11/06/2026**

À distance

12 places restantes

**15/06/2026 → 18/06/2026**

À distance

12 places restantes

**22/06/2026 → 25/06/2026**

À distance

12 places restantes

### ACCESSIBILITÉ · HANDICAP

Nos formations sont accessibles aux personnes en situation de handicap. Pour toute situation nécessitant un aménagement (matériel, temporel ou pédagogique), nous vous invitons à nous contacter avant l'inscription afin d'étudier les adaptations possibles.

Référent handicap : [contact@lapolaris.fr](mailto:contact@lapolaris.fr)

### LaPolaris

TÉL. +33762584798

EMAIL [contact@lapolaris.fr](mailto:contact@lapolaris.fr)

WEB [lapolaris.fr](http://lapolaris.fr)