Fiche du cours

55 h

Titre :

BDD200 - Bases de données relationnelles (SQL & Modélisation)

Description :

Introduction pratique aux bases de données avec un focus PostgreSQL : modélisation conceptuelle (ER), passage au modèle relationnel, SQL (DDL/DML), intégrité (PK/FK/contraintes), normalisation (1NF→3NF), transactions & ACID, index & performance, sécurité/backup. Cours orienté cas réels, du besoin métier au schéma et aux requêtes exploitables.

Objectifs :

- Analyser un besoin et concevoir un schéma relationnel cohérent (ER → tables)*
- Écrire des requêtes SQL efficaces (sélection, jointures, agrégations, sous-requêtes).*
- Garantir l'intégrité (contraintes, clés) et appliquer la normalisation.*
- Comprendre transactions, isolation, verrous et éviter les anomalies.*
- Améliorer la performance (index, plan d'exécution) et gérer sécurité & sauvegardes.

Chapitres :

1. Fondamentaux & écosystème : SGBD, PostgreSQL, schémas, tables, vues*
2. Modélisation ER (MCD) : entités, attributs, relations, cardinalités*
3. Relationalisation : PK/FK, contraintes (UNIQUE, CHECK), règles de nommage*
4. SQL – DDL/DML : création de schéma, inserts/updates, sélection de base*
5. JOINS & agrégations : INNER/LEFT/RIGHT, GROUP BY/HAVING, sous-requêtes*
6. Qualité & normalisation : 1NF, 2NF, 3NF (BCNF aperçu), dénormalisation raisonnée*
7. Transactions & ACID : isolation (lecture sale/non répétable/phantom), verrous, deadlocks (notions)*
8. Index & perfs : B-Tree/Hash (aperçu), stratégies d'indexation, lecture de plan (EXPLAIN), vues matérialisées (aperçu)
9. Sécurité & opérations : rôles/permissions, backup/restore, import/export, migration de données
10. Mini-projet : du besoin métier au schéma (ER→tables), requêtes clés (JOINS/agrégats), contraintes & index, jeu de données et rapport de résultats

À la fin :

Vous saurez concevoir un schéma relationnel, écrire des requêtes SQL robustes, garantir l’intégrité des données, gérer transactions & index pour des performances correctes, et assurer sécurité & sauvegardes — livrable portfolio-ready. Vous pourrez aussi normaliser (et, si besoin, dénormaliser) avec discernement et lire un plan d’exécution (EXPLAIN) pour optimiser vos requêtes. Enfin, vous produirez une documentation claire (diagramme ER + scripts SQL) et saurez mettre en œuvre un plan de sauvegarde/restauration testé.