

# Projet Base de Données Distribuées - Instructions Détaillées

Ce document fournit les instructions détaillées pour réaliser le projet de base de données distribuées sur une seule machine. Les étudiants devront mettre en œuvre trois bases régionales (Dakar, Thiès, Saint-Louis) avec synchronisation automatique, en utilisant Java Spring Boot et PostgreSQL/MySQL.

## 1. Pré-requis

1. Java 21 installé
2. Maven installé
3. PostgreSQL ou MySQL installé et accessible
4. IDE (IntelliJ, Eclipse ou VS Code)

## 2. Création des bases de données

Exemple pour PostgreSQL :

```
CREATE DATABASE ventes_dakar;  
CREATE DATABASE ventes_thies;  
CREATE DATABASE ventes_stlouis;
```

Créer un utilisateur avec mot de passe et droits sur les trois bases :

```
CREATE USER dsms_user WITH PASSWORD 'dsms_pass';  
GRANT ALL PRIVILEGES ON DATABASE ventes_dakar TO dsms_user;  
GRANT ALL PRIVILEGES ON DATABASE ventes_thies TO dsms_user;  
GRANT ALL PRIVILEGES ON DATABASE ventes_stlouis TO dsms_user;
```

## 3. Structure du projet

- src/main/java/com/example/dsms/model : entité Vente
- src/main/java/com/example/dsms/dakar/repository : VenteRepositoryDakar
- src/main/java/com/example/dsms/thies/repository : VenteRepositoryThies
- src/main/java/com/example/dsms/stl/repository : VenteRepositoryStl
- src/main/java/com/example/dsms/service : MultiVenteService, SyncService
- src/main/java/com/example/dsms/config : DataSource Configs
- src/main/java/com/example/dsms/controller : REST et WebController
- src/main/resources/templates : Thymeleaf pages
- src/main/resources/application.yml : configuration des DataSources

## 4. Entité Vente

```
package com.example.dsms.model;
```

```
@Entity  
public class Vente {  
    @Id private UUID id;  
    private LocalDate dateVente;  
    private Double montant;  
    private String produit;  
    private String region;  
    private LocalDateTime updatedAt;  
    // @PrePersist et @PreUpdate pour updatedAt  
}
```

## 5. Configuration DataSources

Créer trois fichiers : DakarDataSourceConfig, ThiesDataSourceConfig, StlDataSourceConfig  
Chaque fichier définit : DataSource, EntityManagerFactory, TransactionManager  
Configurer spring.datasource.dakar, thies et stl dans application.yml

## 6. MultiVenteService

Service central pour les opérations CRUD :

- saveToRegion(Vente v, String region)
- findByRegion(String region)
- findAll() pour toutes les bases

## 7. SyncService

Scheduler pour synchronisation automatique :

- Comparer updatedAt pour résolution last-write-wins
- Copier les ventes manquantes dans les autres bases
- Intervalle configurable via sync.interval dans application.yml

## 8. Interface web Thymeleaf

Page principale :

- Liste des ventes consolidées
- Formulaire pour ajouter une vente
- Bouton pour synchronisation manuelle

## 9. Tests et scénarios

- Ajouter ventes dans chaque région et vérifier présence après sync
- Tester conflit et last-write-wins
- Simuler panne d'une DB et vérifier récupération après redémarrage

## 10. Critères d'évaluation

- Fonctionnement correct des 3 bases et synchronisation
- Code structuré et clair
- Interface web opérationnelle
- Rapport explicatif et démonstration des scénarios

## 11. Bonus possibles

- Synchronisation incrémentale
- Interface Angular
- Détection de conflits avancée
- Journaux ou monitoring de la synchronisation

Région	Base de données	Repository	Config Spring
Dakar	ventes_dakar	VenteRepositoryDakar	DakarDataSourceConfig
Thiès	ventes_thies	VenteRepositoryThies	ThiesDataSourceConfig
Saint-Louis	ventes_stlouis	VenteRepositoryStl	StlDataSourceConfig

Ce guide fournit un chemin clair pour réaliser le projet complet de base de données distribuées. Les étudiants doivent suivre les étapes pour configurer les bases, les DataSources, le service central et le scheduler de synchronisation. L'objectif pédagogique est de comprendre la cohérence éventuelle et la réplique entre bases multiples sur une seule machine.