

DOSSIER DE CONCEPTION

Base de données MySQL – WMS NordTransit Logistics (NTL)

Projet MSPR3 – NFL IT

<https://github.com/AnasElmaliki69/mspr-3>

1. Introduction

NFL IT mandate notre équipe afin de concevoir et sécuriser une nouvelle base de données destinée au **système de gestion d'entrepôt (WMS)** de **NordTransit Logistics (NTL)**.

Le WMS constitue un **élément critique** de la chaîne logistique : toute indisponibilité entraîne l'arrêt immédiat des opérations de réception, de préparation et d'expédition.

L'existant repose sur une **base MySQL centralisée**, présentant plusieurs faiblesses :

- absence de haute disponibilité,
- sauvegardes insuffisamment sécurisées,
- supervision limitée,
- dépendance à un seul datacenter.

La direction impose des objectifs stricts de continuité :

- **RTO : 1 heure**
- **RPO : 15 minutes**

Ce dossier de conception présente :

- la modélisation des données (MCD / MLD),
- le choix du SGBD,
- l'architecture de haute disponibilité et la réplication inter-sites,
- l'intégration du Plan de Reprise d'Activité (PRA),
- la stratégie de sauvegarde,
- la sécurité des accès,
- la supervision et les indicateurs critiques.

Objectif final : garantir un WMS **pérenne, performant, sécurisé et industrialisé**, conforme aux exigences opérationnelles de NTL.

Dossier de Conception

2. Contexte Général

NordTransit Logistics exploite plusieurs entrepôts (Lille, Lens, Valenciennes, Arras) et dépend entièrement de son WMS pour les opérations quotidiennes.

Contraintes majeures :

- activité continue de 5h30 à 18h30,
- fenêtres de maintenance très réduites,
- impact immédiat sur la production en cas d'arrêt.

Les limites actuelles identifiées :

- base de données unique (single point of failure),
- absence de réplication,
- dépendance à un seul site physique,
- PRA incomplet.

Le projet vise donc une refonte complète de la conception de la base, intégrant dès l'origine la haute disponibilité et la reprise après sinistre.

3. Modélisation des Données (MCD / MLD)

3.1 Objectifs du modèle

- Structurer les données essentielles au fonctionnement du WMS.
- Garantir l'intégrité, la cohérence et la traçabilité.
- Permettre une gestion fiable des stocks par article et localisation.
- Faciliter les requêtes opérationnelles (stocks, mouvements, historiques).

3.2 Modèle Conceptuel de Données (MCD)

Le MCD repose sur cinq entités principales :

- **CLIENT** : clients pour lesquels les marchandises sont stockées.
- **ARTICLE** : produits appartenant à un client.
- **LOCALISATION** : emplacements physiques dans l'entrepôt.
- **STOCK** : quantité d'un article dans une localisation donnée.
- **MOUVEMENT** : historique des entrées et sorties de stock.

Relations :

- CLIENT (1,N) ARTICLE
- ARTICLE (1,N) STOCK
- LOCALISATION (1,N) STOCK
- STOCK (1,N) MOUVEMENT

Ces relations respectent strictement les règles métier d'un WMS.

3.3 Modèle Logique de Données (MLD)

CLIENT

- id_client (PK)
- nom
- adresse

ARTICLE

- id_article (PK)
- id_client (FK)
- libelle
- poids
- dimensions

LOCALISATION

- id_localisation (PK)
- code_localisation
- type_localisation

STOCK

- id_stock (PK)
- id_article (FK)
- id_localisation (FK)
- quantite

MOUVEMENT

- id_mouvement (PK)
- id_stock (FK)
- type_mouvement
- quantite
- date_mouvement

3.4 Points Clés

- Séparation stricte des données par client.
- Gestion du stock au niveau article/localisation.
- Traçabilité complète via l'historique des mouvements.
- Modèle normalisé et optimisé pour un usage transactionnel intensif.

4. Choix du SGBD

4.1 Contraintes

- Base critique et transactionnelle.
- Besoin de cohérence stricte (ACID).
- Haute disponibilité et réplication.
- Respect RTO/RPO.
- Coût maîtrisé.

4.2 Analyse des solutions

- **NoSQL** : non adapté (absence d'intégrité relationnelle).
- **SGBD propriétaires** : performants mais coûteux.

- **SGBD SQL open source** : adaptés au contexte.

4.3 Justification du choix MySQL

MySQL a été retenu car :

- déjà utilisé dans l'existant,
- très performant en OLTP,
- réplication native fiable,
- binlogs compatibles avec le RPO,
- coût nul,
- administration maîtrisée par l'équipe IT.

5. Architecture de Haute Disponibilité & PRA

5.1 Objectifs

- Éliminer le point de défaillance unique.
- Garantir la continuité même en cas de perte de site.
- Intégrer le PRA dès la conception.

5.2 Architecture HA avec Réplication Inter-Datacenters

L'architecture repose sur trois instances MySQL :

- **PRIMARY – Datacenter de Lille**
 - Base active
 - Toutes les écritures WMS
- **REPLICA locale – Datacenter de Lille**
 - Réplication semi-synchrone
 - Haute disponibilité locale (PCA)
- **REPLICA distante – Datacenter de Lens**
 - Réplication asynchrone
 - Protection contre perte du site principal

Cette architecture garantit :

- continuité locale,
- résilience inter-sites,
- respect du RPO 15 minutes.

5.3 Intégration du PRA

Le PRA couvre :

- panne du PRIMARY,
- corruption de données,
- perte du datacenter principal,
- désastre majeur (perte totale).

Les procédures de bascule et de restauration sont alignées sur l'architecture HA et permettent :

- $RTO \leq 1$ heure
- $RPO \leq 15$ minutes

6. Stratégie de Sauvegarde

- Sauvegarde complète quotidienne.
- Archivage des binlogs toutes les 15 minutes.
- Stockage :
 - NAS local isolé,
 - Azure Blob Storage (externalisation).
- Application de la règle **3-2-1**.
- Support du **Point-In-Time Recovery (PITR)**.

7. Sécurité des Accès

- Comptes MySQL séparés par rôle.
- Principe du moindre privilège.
- Accès réseau strictement filtrés.
- Chiffrement TLS des flux.
- Chiffrement des sauvegardes.
- Journalisation et audit.

8. Supervision

Supervision en temps réel :

- réplication (lag),
- performances,
- stockage,
- sauvegardes.

Alertes automatiques en cas de dépassement de seuil critique.

9. Conclusion

La conception proposée :

- supprime les points de défaillance uniques,
- garantit la continuité du WMS,
- respecte strictement les objectifs RTO/RPO,
- s'adapte aux contraintes d'une PME logistique,
- reste simple, robuste et économiquement viable.