

Entrepôt de données Ontario 511 — Rapport de projet

Realiser par :

- **Sabrina Dahmar**

- **Abdenmour Kerrouch**

1) Objectif & périmètre

Mettre en place un petit entrepôt de données fiable pour les flux opérationnels Ontario 511 (événements, travaux, caméras de circulation et conditions routières), avec une ingestion répétable, un typage strict, la déduplication, la traçabilité/audit et quelques requêtes de reporting rapide. La cible technique est Microsoft SQL Server.

2) Sources de données

Quatre jeux de données CSV encodés en UTF-8 sont ingérés :

Fichier	Description
evenements_ontario_511.csv	Événements de circulation (accidents, fermetures, incidents)
constructions_ontario_511.csv	Travaux planifiés/en cours
cameras_ontario_511.csv	Inventaire des caméras et vues
roadconditions_ontario_511.csv	Instantanés des conditions routières avec horodatage

Les chemins d'accès sont paramétrés via @RootPath et chargés avec des modèles BULK INSERT. Les fichiers bruts arrivent d'abord dans des tables de staging avant conversion et MERGE vers les tables curées.

3) Architecture & schéma



3.1 Tables cibles (curées)

Evenements, Constructions, Cameras, RoadConditions — chacune avec des clés et index adaptés.

Contraintes uniques exemples : UQ_Evenements(SourceId, ID), UQ_Cameras(BaseId, ViewId), UQ_RoadConditions(RoadwayName, LastUpdated, LocationDescription).

Dates au format DATETIME2(0). Règles d'intégrité : par ex., DateFin ≥ DateDebut lorsque pertinent.

3.2 Couche de staging

Pour chaque domaine, une table Staging_* conserve les colonnes en NVARCHAR pour absorber les variations des CSV avant la conversion typée. Cela isole les préoccupations d'ingestion des validations/contraintes.

3.3 Suivi des modifications & audit

Déclencheurs (triggers) qui mettent à jour DateModification lors des écritures sur les tables curées.

HistoriqueEvenements enregistre les instantanés avant mise à jour pour une traçabilité complète des événements.

4) Ingestion & transformation

4.1 Chargement BULK

BULK INSERT en UTF-8, terminators configurables (0x0d0a vs 0x0a), FIRSTROW=2.

MAXERRORS et ERRORFILE optionnels pour la résilience aux fichiers imparfaits.

Chargement uniquement dans les tables de staging.

4.2 MERGE vers la couche curée

Procédures stockées : Import_Evenements, Import_Constructions, Import_Cameras, Import_RoadConditions.

Conversions robustes avec TRY_CONVERT pour dates/décimaux/booleans.

Upsert basé sur des clés naturelles pour éviter les doublons et conserver la dernière version des attributs.

Mise à jour de DateModification lors des updates.

4.3 Correctifs qualité (observés)

Colonnes Description élargies en NVARCHAR(MAX) pour éliminer le risque de troncature.

Contrôles de parsing de dates et conversions alternatives (ex. DATETIMEOFFSET → DATETIME2) pour les timestamps hétérogènes dans RoadConditions.

5) Journalisation & orchestration

Un logger léger (LogsExecution) et la procédure Log_Exec capturent le type de dataset, le chemin du fichier, les volumes, le statut et les messages d'erreur. Le script d'orchestration encapsule chaque chargement dans TRY/CATCH, journalise succès/échec, puis appelle la procédure Import_* correspondante. Une requête finale affiche les dernières lignes du journal pour vérification rapide.

6) Utilitaires de reporting & exploration

Report_TopRoad_Cameras — route avec le plus de caméras.

Report_TopRoad_Constructions — routes les plus chargées en travaux (actuels/à venir).

Search_Evenements(@Road, @Type, @Date) — recherche filtrable rapide d'événements.

7) Guide d'exécution (démarrage rapide)

Créer la base & le schéma : exécuter ontario511.sql pour créer tables, triggers et procédures.

Renseigner les chemins : éditer @RootPath vers le dossier des CSV.

Charger le staging : utiliser les modèles BULK INSERT ; ajuster ROWTERMINATOR si les compteurs sont à zéro.

Importer vers la couche curée : exécuter Import_Evenements, Import_Constructions, Import_Cameras, Import_RoadConditions.

Vérifier : contrôler les volumes dans les tables curées et consulter LogsExecution pour les statuts/erreurs.

Rapports d'exemple : exécuter les trois procédures ci-dessus pour une validation rapide du contenu.

8) Points forts

Robustesse : stratégie staging-first avec parsing tolérant (TRY_CONVERT) et chargements récupérables.

Qualité & déduplication : contraintes uniques naturelles, clés de MERGE, contrôles de cohérence des dates.

Auditabilité : journalisation dédiée + trigger d'historisation des événements.

Performance : index ciblés ; TABLOCK lors des BULK INSERT lorsque pertinent.

9) Limites & pistes d'amélioration

Fuseaux horaires : normaliser en UTC dès l'ingestion pour RoadConditions si les formats sont mixtes.

Fraîcheur des données : planifier l'orchestration (SQL Agent ou scheduler externe) pour des mises à jour régulières.

Couche analytique : ajouter des vues pour des KPI fréquents (fermetures par corridor, durée des événements, recouvrement des travaux).

Géospatial : envisager geometry/geography pour EncodedPolyline et lat/long afin d'activer les requêtes cartographiques.