Rapport de Projet SAÉ 302 : Mise en œuvre d'un Système Décisionnel (Data Warehouse)

Participants :

MARRE Ewann
MAURIN Antoine
SANZ Rafaël
ZAVAGNO Quentin

Année universitaire : 2025-2026

Établissement : IUT Lumière Lyon 2 – Département SD

Table des matières

1. Introduction

Dans le cadre de la SAÉ 302 "Intégration de données dans un datawarehouse", notre équipe a eu pour mission de concevoir et de développer une chaîne décisionnelle complète (BI Pipeline). L'objectif était de partir de données brutes, de les nettoyer, de les stocker dans un entrepôt de données (Data Warehouse) modélisé spécifiquement pour l'analyse, et enfin de produire des tableaux de bord interactifs.

Pour ce projet, nous avons choisi le domaine de la **Formule 1**. Ce sport, riche en statistiques historiques et en données techniques (télémétrie, temps au tour, arrêts aux stands), constitue un terrain idéal pour l'analyse décisionnelle. Il permet de traiter des volumétries intéressantes et de croiser des axes d'analyse variés (pilotes, écuries, circuits, temps).

Ce rapport détaille les étapes techniques et fonctionnelles de notre démarche, depuis la récupération des fichiers sources CSV jusqu'à la visualisation sous Power BI, en passant par l'ETL réalisé avec Pentaho Data Integration et le stockage sous PostgreSQL.

2. Contexte Métier et Objectifs Décisionnels

2.1. Le contexte

Le championnat du monde de Formule 1 existe depuis 1950. Chaque saison est composée d'une série de courses (Grands Prix) disputées sur des circuits à travers le monde. Les pilotes et les constructeurs accumulent des points en fonction de leur classement à l'arrivée.

2.2. Problématique et Questions Métier

L'objectif de notre système décisionnel est de permettre à un analyste sportif ou à une écurie de comprendre les facteurs de performance et l'évolution historique du sport. Nous avons identifié plusieurs axes d'analyse prioritaires :

- **Performance des Pilotes :** Qui sont les pilotes les plus titrés ? Quel est leur taux de victoire par rapport au nombre de courses disputées ? Comment évolue leur performance avec l'âge ?
- **Domination des Constructeurs :** Quelle écurie a dominé quelle ère (ex: Ferrari années 2000, Mercedes années 2010) ? Quelle est la fiabilité des voitures (analysée via les statuts d'abandon) ?
- **Analyse des Circuits :** Quels circuits favorisent les vitesses les plus élevées ? Où y a-t-il le plus d'accidents ?
- **Analyse Temporelle :** Comparaison des temps au tour et des vitesses moyennes au fil des décennies.

3. Sources de Données

Pour alimenter notre entrepôt, nous avons utilisé un jeu de données public (Open Data) provenant de la plateforme Kaggle, réputé pour son exhaustivité sur l'histoire de la F1.

3.1. Description des fichiers

Les données sont fournies sous forme de fichiers plats (CSV). Voici les principaux fichiers utilisés :

- drivers.csv : Informations sur les pilotes (Nom, Prénom, Nationalité, Date de naissance, URL Wikipedia).
- constructors.csv : Informations sur les écuries (Nom, Nationalité).
- circuits.csv : Détails géographiques des circuits (Nom, Lieu, Pays, Latitude, Longitude, Altitude).
- races.csv : Calendrier des courses (Date, Heure, Saison, Circuit associé).
- results.csv : Table centrale contenant le classement final de chaque pilote pour chaque course, les points, le temps de course, le rang, etc.
- lap_times.csv : Données volumineuses détaillant le temps de chaque tour pour chaque pilote.
- pit_stops.csv : Durée et moment des arrêts aux stands.
- qualifying.csv : Résultats des séances de qualification.
- status.csv : Référentiel expliquant les codes de statut de fin de course (ex: 1 = "Finished", 3 = "Accident", 5 = "Engine").

3.2. Qualité des données brutes

Les données brutes présentaient plusieurs défis que nous avons dû gérer lors de l'ETL :

- La valeur \N était utilisée pour signifier NULL dans les fichiers CSV, nécessitant un traitement spécifique pour ne pas être interprétée comme une chaîne de caractères "N".
- Les formats de temps (ex: "1:23.456") devaient être harmonisés ou convertis en millisecondes pour permettre des calculs d'agrégation (moyennes, écarts).

4. Architecture Technique

Nous avons mis en œuvre une architecture en couches classiques en Business Intelligence :

1. **Zone de Staging (RAW) :**
    - Chargement des fichiers CSV "tels quels" dans une base de données PostgreSQL (schéma raw). Aucun nettoyage n'est effectué ici, l'objectif est l'acquisition rapide.
    - Scripts utilisés : Insert_drivers_raw_F1.ktr, Insert_races_raw_F1.ktr, etc.
2. **Zone de Stockage et d'Assainissement (SAS - Storage Area System) :**
    - Les données sont nettoyées, typées (conversion des chaînes en entiers/dates) et normalisées dans le schéma sas.
    - C'est ici que nous gérons les valeurs nulles et les formats de données.
    - Tables créées : sas.drivers, sas.results, sas.circuits, etc.
3. **Entrepôt de Données (DWH - Data Warehouse) :**
    - Les données sont restructurées selon un modèle en étoile (Star Schema) optimisé pour les requêtes analytiques.
    - Schéma dwh dans PostgreSQL.

Outils utilisés :

- **SGBD :** PostgreSQL (pour les schémas raw, sas, dwh).
- **ETL :** Pentaho Data Integration (Spoon) pour la création des flux (.ktr) et l'orchestration (.kjb).
- **Dataviz :** Microsoft Power BI (PB_F1.pbix).

5. Modélisation du Data Warehouse

Pour répondre aux questions métier, nous avons conçu un **Schéma en Étoile**. Ce choix se justifie par sa simplicité de navigation pour l'outil de reporting et ses performances de lecture.

5.1. Tables de Dimensions (dim_)

Les dimensions fournissent le contexte de l'analyse (Qui ? Où ? Quand ?).

- **dwh.dim_drivers** : Contient une clé artificielle (key_driver), l'ID source, le nom, prénom, nationalité et date de naissance.
- **dwh.dim_constructeurs** : Répertorie les écuries (Ferrari, McLaren, Red Bull...) avec leur nationalité.
- **dwh.dim_circuits** : Géolocalisation des circuits (Latitude, Longitude, Altitude, Pays).
- **dwh.dim_races** : Dimension spécifique à l'événement "Course", incluant le nom du Grand Prix et l'année.
- **dwh.dim_calendrier** : Dimension temporelle générée (Date, Année, Mois, Jour, Trimestre, Indicateur Week-end). Elle est cruciale pour les agrégations temporelles.
- **dwh.dim_status** : Permet d'analyser les causes d'abandons (libellé du statut).

5.2. Tables de Faits (fact_)

Les faits contiennent les métriques numériques (Combien ?).

- **dwh.fact_results** : Table de faits principale (granularité : un pilote par course).
    - *Métriques :* Points gagnés (nb_points), Rang à l'arrivée (rang_arrivee), Temps de course en ms, Vitesse du meilleur tour.
    - *Clés étrangères :* Vers dim_drivers, dim_constructeurs, dim_races, dim_status.
- **dwh.fact_lap_times** : Table de faits détaillée (granularité : un tour par pilote par course).
    - *Métriques :* Temps du tour, position à ce tour.
    - Permet des analyses fines sur la régularité des pilotes.

6. Processus ETL (Extract, Transform, Load)

Le peuplement du Data Warehouse est automatisé via Pentaho (PDI). L'orchestration est gérée par des "Jobs" (.kjb) qui appellent séquentiellement les "Transformations" (.ktr).

6.1. Phase RAW et SAS (job_raw.kjb et job_sas.kjb)

Nous importons d'abord les CSV. Une transformation typique (ex: T_SAS_F1_DRIVERS.ktr) lit la table brute, filtre les colonnes inutiles, remplace les chaînes "\N" par des NULLs database, et convertit les dates (ex: birth date) au format SQL correct avant d'insérer dans le schéma SAS.

6.2. Phase DWH (Chargement des Dimensions et Faits)

C'est l'étape la plus complexe, impliquant la gestion des clés de substitution (Surrogate Keys).

- **Chargement des Dimensions (ex: T_DIM_DRIVERS.ktr) :**

Nous utilisons des composants "Combination Lookup/Update" ou des séquences PostgreSQL pour générer des clés techniques (key_driver) uniques, indépendantes des IDs opérationnels. Cela permet de gérer l'historisation si nécessaire (SCD Type 1 ou 2).

- **Chargement des Faits (T_FACT_RESULTS.ktr) :**

La transformation récupère les données de sas.results. Pour chaque ligne, elle effectue des "Database Lookups" vers les tables de dimensions (dim_drivers, dim_races, etc.) pour récupérer les clés techniques (key_*) correspondantes.

  - *Calculs :* Nous avons calculé le temps total en millisecondes pour faciliter les moyennes.
  - *Nettoyage final :* Les positions non numériques sont traitées.

6.3. Ordonnancement

L'ordre d'exécution est strict :

1. Chargement RAW (CSV -> DB).
2. Chargement SAS (Typage).
3. Chargement Dimensions (Création des référentiels).
4. Chargement Faits (Nécessite que les dimensions soient peuplées pour les clés étrangères).

7. Analyse et Visualisation (Power BI)

Le fichier PB_F1.pbix contient notre tableau de bord final. Grâce au modèle en étoile, les relations entre les tables sont gérées automatiquement (relations 1-to-Many entre Dimensions et Faits).

7.1. Tableau de Bord Général

Une vue d'ensemble permet de filtrer par **Saison** (via dim_calendrier ou dim_races) et par **Constructeur**.

- **KPIs :** Nombre total de points, Nombre de victoires, Nombre de podiums.
- **Carte Géographique :** Utilisation des lat/long de dim_circuits pour visualiser la répartition mondiale des Grands Prix.

7.2. Analyse Pilotes & Écuries

- **Histogrammes :** Top 10 des pilotes par nombre de victoires historiques.
- **Courbes d'évolution :** Cumul des points d'un constructeur sur une saison donnée (ex: Duel Mercedes vs Red Bull en 2021).
- **Radar Chart (si utilisé) :** Comparaison des caractéristiques techniques ou des performances sur différents types de circuits.

7.3. Analyse de la Fiabilité

En utilisant la table de faits fact_results croisée avec dim_status, nous avons pu créer des graphiques (Camemberts / Treemaps) montrant la répartition des causes d'abandon (Moteur, Accident, Panne hydraulique) par écurie.

8. Conclusion

Ce projet SAÉ 302 nous a permis de mettre en pratique l'ensemble de la chaîne de valeur de la Business Intelligence. Nous sommes passés de fichiers CSV bruts et dispersés à un système d'information décisionnel structuré et exploitable.

Les principaux acquis techniques sont :

- La maîtrise de Pentaho pour la création de flux de données complexes.
- La compréhension de l'importance de la modélisation en étoile pour la performance des requêtes.
- La capacité à nettoyer des données réelles (gestion des NULLs, formats hétérogènes).

Sur le plan fonctionnel, l'entrepôt de données créé est robuste et évolutif. Il pourrait être enrichi par l'ajout de nouvelles saisons ou par l'intégration de données météorologiques pour affiner l'analyse des performances. Le tableau de bord Power BI offre une interface intuitive répondant aux besoins d'analyse identifiés en début de projet.