

Confluence

Search

+ Create

Upgrade

5

For you

Recent

Starred

Spaces

Apps

Souad Tahri

Shortcuts

Content + ...

Search by title

Getting started...

2026-0... DRAFT

Docum... DRAFT

Create

Invite people

Documentation du Projet : Transit Ridership Analysis (Chicago & Philadelphia) Saved STS Publish... Close Share ...

Write Tt Normal text B A Emojis Status Header image A

Documentation du Projet : Transit Ridership Analysis (Chicago & Philadelphia)

By Souad Tahri 11 min

1. Objectif du Projet

L'objectif principal est de fournir une solution de Business Intelligence centralisée pour analyser la fréquentation des transports urbains de Chicago et Philadelphie. Le dashboard permet de piloter la performance par mode de transport et par ligne (Route) pour optimiser la gestion du trafic.

2. Architecture & Traitement des Données (ETL)

Le pipeline de données a été construit de manière modulaire pour garantir la flexibilité de l'analyse.

A. Extraction & Nettoyage (Python Script)

- Traitement RDF (Chicago)** : Utilisation de `rdflib` pour extraire les triplets des fichiers RDF. Les données ont été converties en DataFrame, nettoyées (types de données), filtrées (après 2018), puis exportées vers `chicago_rdf_combined.csv`.
- Traitement Excel (Chicago)** : Chargement du fichier Excel des totaux de boarding. Filtrage pour ne garder que les données après 2018 et exportation vers `chicago_excel_cleaned.csv`.
- Analyse de Philadelphie** : Chargement des fichiers CSV `By_Route` et `By_Mode`. Réalisation d'un EDA (Taille, Qualité, Doublons) pour chaque fichier avant l'exportation finale.

B. Transformation Avancée (Power Query - M)

C'est ici que la véritable consolidation a eu lieu :

- Unpivoting (Chicago)** : Transformation des colonnes `bus` et `rail` du fichier Excel Chicago en lignes pour créer un format "Mode" compatible avec Philadelphie.
- Normalisation (Philadelphie)** : Fusion des catégories `Heavy Rail` et `Regional Rail` en un seul mode `Rail`.
- Consolidation Finale** : Utilisation de `Table.Combine` pour créer les tables de faits globales :
 - `table_de_fait_mode` : Union de Chicago (Excel) et Philadelphie (Mode).
 - `table_de_fait_Route` : Union de Chicago (RDF) et Philadelphie (Route).

3. Transformation et Modélisation dans Power BI

Après l'étape de préparation sur Python, les données ont été importées dans Power BI pour subir des transformations avancées via **Power Query**.

A. Tables Sources (Staging Area)

Dans cette partie, nous avons importé les fichiers CSV générés par Python. Chaque table a subi des transformations pour garantir la cohérence des formats.

chicago_route

Cette table traite les données granulaires issues du fichier RDF converti. L'objectif ici était de passer d'une donnée brute journalière à une donnée agrégée mensuelle pour fluidifier le modèle.

Étapes clés de transformation :

- Extraction Temporelle : À partir de la colonne `date` originale, deux nouvelles colonnes ont été créées : `Year` (Année) et `Month` (Mois) pour permettre l'analyse chronologique.
- Nettoyage et Renommage : Suppression de la colonne `daytype` (non nécessaire pour l'analyse globale) et renommage des colonnes pour correspondre aux standards du projet (ex: `route` devient `Route`, `ridership` devient `Ridership`)

© 2023 Microsoft Corporation. Tous droits réservés.

- Identification de la Source : Ajout d'une colonne personnalisée `City` avec la valeur "Chicago" pour distinguer ces données lors de la future consolidation.
- Agrégation de Performance (Grouping) : C'est l'étape cruciale où nous avons utilisé `Table.Group` pour calculer la **moyenne du Ridership** (`List.Average`) par année, par mois et par ligne. Cela permet de réduire considérablement le nombre de lignes tout en conservant la précision statistique nécessaire.
- Réorganisation : Les colonnes ont été permutées pour suivre l'ordre logique : `Year`, `Month`, `Route`, `Ridership`, `City`.

philadelphie_route

Cette table contient les données de fréquentation par ligne pour la ville de Philadelphie. L'objectif des transformations était d'aligner parfaitement sa structure avec celle de Chicago pour permettre une consolidation ultérieure.

Étapes clés de transformation :

- Nettoyage des données : Suppression des colonnes techniques non analytiques, à savoir `Source` et `ObjectId`, afin d'alléger le modèle de données.
- Standardisation des en-têtes : Pour assurer la compatibilité entre les sources, les colonnes temporelles `Calendar_Year` et `Calendar_Month` ont été renommées respectivement en `Year` et `Month`.
- Harmonisation de la métrique : La colonne `Average_Daily_Ridership` a été renommée en `Ridership` pour maintenir une nomenclature uniforme à travers tout le projet.
- Identification Géographique : Ajout d'une colonne personnalisée `City` avec la valeur fixe "philadelphie". Cette étape est cruciale pour distinguer l'origine des données lors de l'union dans la table de faits.
- Typage rigoureux : Application de types de données stricts (Entiers pour les années/mois et Texte pour la ville) pour garantir la précision des calculs DAX.

chicago_Mode

Cette table traite les données de fréquentation globale par mode (Bus et Rail) pour Chicago. Elle a nécessité des transformations structurelles avancées pour passer d'un format "Large" (colonnes par mode) à un format "Long" compatible avec le reste du modèle.

Étapes clés de transformation :

- Ingénierie Temporelle : À partir de la colonne `service_date`, extraction de l'année (`Year`) et du mois (`Month`) pour permettre une analyse granulaire du temps.
- Dépivotage des colonnes (Unpivot) : C'est l'étape la plus critique. Les colonnes originales `bus` et `rail_boardings` ont été transformées en lignes. Cela a permis de créer une colonne `Mode` (contenant les étiquettes) et une colonne `Ridership` (contenant les valeurs), rendant la table dynamique et facile à agréger.
- Normalisation sémantique : Utilisation de la fonction `Table.ReplaceValue` pour uniformiser les noms des modes. Les valeurs techniques comme `rail_boardings` ont été remplacées par "Rail" et `bus` par "Bus" pour correspondre exactement aux données de Philadelphie.
- Filtrage Stratégique : Application d'un filtre sur la colonne `Year` pour restreindre l'analyse à la période cible (**2019 à 2025**).
- Attribution Géographique : Ajout de la colonne `City` avec la valeur "Chicago" pour assurer la traçabilité de la source lors de la phase de consolidation.

philadelphie_Mode

Cette table centralise la fréquentation par mode de transport pour Philadelphie. Elle a subi des transformations de normalisation importantes pour s'aligner sur la structure de données de Chicago, notamment au niveau de la classification des modes de transport.

Étapes clés de transformation :

- Sélection et Nettoyage : Suppression des métadonnées sources (`Source` et `ObjectId`) pour ne conserver que les dimensions analytiques. Les colonnes temporelles ont été renommées en `Year` et `Month`.
- Filtrage des Modes : Restriction des données aux modes principaux : `Bus`, `Heavy Rail` et `Regional Rail`. Cette étape élimine les modes secondaires non comparables.
- Normalisation et Fusion (Mapping) : Pour permettre une comparaison directe avec Chicago, les

catégories `Heavy Rail` et `Regional Rail` ont été fusionnées sous l'étiquette unique "`Rail`" via la fonction `Table.ReplaceValue`.

- Agrégation par Moyenne (Grouping) : Utilisation de `Table.Group` pour calculer la moyenne du `Ridership` (`List.Average`) par année, mois et mode. Cela garantit une cohérence statistique lors de la consolidation avec les données de Chicago.
- Finalisation de la Structure : Réorganisation des colonnes (`Year`, `Month`, `Mode`, `Ridership`, `City`) et typage des données pour assurer l'intégrité du modèle relationnel.

B. Tables de Faits (Fact Tables)

Les tables de faits constituent le pivot central de notre modèle de données. Elles stockent les mesures quantitatives (`Ridership`) associées aux différentes clés étrangères qui les relient aux dimensions.

1. **table_de_fait_mode** : Cette table est le résultat de la consolidation des données de haut niveau (par mode de transport) pour les deux métropoles. Elle permet d'effectuer des analyses comparatives macro-économiques entre Chicago et Philadelphie.

Détails de l'implémentation :

- Consolidation (Union) : Utilisation de la fonction `Table.Combine`. Contrairement à une jointure (`Merge`), cette opération de type "Append" empile les enregistrements de `philadelphie_Mode` et `chicago_Mode`.
- Alignement Structurel : Grâce au travail de normalisation effectué en amont dans la Staging Area (renommage des colonnes et alignement des noms de modes "Bus" et "Rail"), Power Query a pu fusionner les deux sources de manière transparente, les colonnes portant des noms identiques étant automatiquement alignées.
- Centralisation des Flux : Cette table permet d'agrégier le volume massif de passagers (plus de 2 milliards de trajets) dans un référentiel unique, facilitant ainsi la création de mesures DAX globales comme le *Total Ridership*.

2. **table_de_fait_Route** : Cette table représente le niveau de granularité le plus fin de notre modèle. Elle regroupe l'ensemble des données de fréquentation par ligne spécifique (Route) pour les deux réseaux de transport.

Détails de l'implémentation:

- Fusion Granulaire (Append) : À l'instar de la table des modes, nous avons utilisé `Table.Combine` pour unir `chicago_route` (issu du traitement RDF) et `philadelphie_route` (issu du CSV).
- Uniformisation des Routes : Cette consolidation permet de traiter les **194 lignes** de transport identifiées comme une seule entité analytique. Cela facilite l'identification des lignes les plus performantes (Top Routes) indépendamment de la ville.
- Optimisation du Filtrage : L'étape `Table.SelectRows(Source, each true)` garantit l'intégrité des lignes importées tout en préparant le terrain pour d'éventuels filtres de sécurité ou de qualité de données.
- Polyvalence Analytique : C'est sur cette table que reposent les indicateurs de performance clés (KPI) tels que le `Max Ridership` par ligne et l'identification dynamique de la route leader.

C. Tables de Dimensions (Star Schema)

Pour structurer notre Schéma en Étoile, nous avons créé des tables de dimensions uniques qui servent de filtres pour les tables de faits. L'objectif de cette section est de transformer les colonnes descriptives en tables de référence indépendantes. Cela optimise les performances de filtrage et permet une gestion centralisée des attributs.

table_Dim_Year

Cette dimension est fondamentale pour activer les analyses temporelles (Time Intelligence). Elle permet de comparer les performances d'une année sur l'autre de manière cohérente.

Détails de l'implémentation :

- Extraction de la Source Globale : Utilisation de `Table.Combine` sur les sources de Chicago et Philadelphie pour s'assurer que l'intégralité de la plage temporelle du projet est couverte.
- Déduplication (Unicité) : Application de la fonction `Table.Distinct` après avoir isolé la colonne `Year`. Cette étape garantit que chaque année n'apparaît qu'une seule fois, ce qui est obligatoire

pour une table de dimension (clé primaire).

- Création de la Clé Primaire (ID) : Ajout d'une colonne d'index nommée `ID_Year`. Cette clé numérique est utilisée pour créer une relation stable et performante avec les tables de faits.
- Rôle dans le Modèle : Elle alimente les segments (Slicers) d'année sur le dashboard, permettant un filtrage instantané de l'ensemble des graphiques de fréquentation.

?

table_Dim_Month :

Cette dimension permet d'affiner l'analyse à l'échelle mensuelle. Elle est indispensable pour identifier les cycles saisonniers (comme la baisse de fréquentation observée au 12ème mois) sur l'ensemble du réseau.

Détails de l'implémentation :

- Extraction Multi-Sources : Comme pour les années, nous avons consolidé les colonnes `Month` de Chicago et Philadelphie pour obtenir une liste exhaustive des périodes traitées.
- Nettoyage et Unicité : Utilisation de `Table.Distinct` pour supprimer les doublons et ne conserver que les 12 mois de l'année. Cette structure garantit l'intégrité de la relation de type **1:N** vers les tables de faits.
- Indexation Technique : Création de la colonne `ID_Month` via une fonction d'indexation. Cet identifiant numérique optimise les jointures internes du moteur Power BI.
- Finalité Analytique : Cette table sert de base aux axes X des graphiques linéaires (Line Charts), permettant de visualiser l'évolution du "Ridership" mois par mois.

?

table_Dim_City :

La dimension `Dim_City` agit comme le pivot géographique du modèle. Elle permet de filtrer dynamiquement les données pour comparer les performances de Chicago et Philadelphie ou pour obtenir une vue consolidée des deux métropoles.

Détails de l'implémentation :

?

- Consolidation des sources : Le code récupère les valeurs de la colonne `City` depuis les deux tables sources normalisées.
- Normalisation et Unicité : L'utilisation de `Table.Distinct` permet d'isoler les deux entités uniques ("Chicago" et "philadelphie"), créant ainsi une table de référence propre sans redondance.
- Clé de jointure (Index) : Création d'une colonne `ID_City`. Cette clé primaire numérique est indispensable pour établir une relation de type **1:N** avec les tables de faits, garantissant ainsi que le filtrage par ville se propage correctement à tous les indicateurs du dashboard.
- Intégrité des données : L'étape finale de typage (`type text`) assure que les noms des villes sont traités correctement par les visuels de Power BI (comme les Slicers et les titres dynamiques).

table_Dim_Mode :

?

La dimension `Dim_Mode` est le référentiel des types de transport analysés. Elle permet de segmenter le trafic global pour comprendre la répartition entre les services de bus et les services ferroviaires (Rail).

Détails de l'implémentation :

?

- Consolidation des Attributs : Le code fusionne les colonnes `Mode` issues des tables de Chicago et Philadelphie. Grâce au travail préalable de normalisation (où "Heavy Rail" et "Regional Rail" ont été convertis en "Rail"), nous obtenons un référentiel propre.
- Unicité des Catégories : L'application de `Table.Distinct` permet d'obtenir les deux valeurs uniques : **Bus** et **Rail**. Cette structure simplifiée est essentielle pour la clarté des visuels (comme le Donut Chart sur le dashboard).
- Clé Primaire (`ID_Mode`) : Un index numérique a été ajouté pour servir de clé de jointure. Dans un Schéma en Étoile, l'utilisation d'ID numériques pour les relations améliore la vitesse de calcul du moteur Power BI.
- Impact sur l'Analyse : Cette table alimente les filtres de catégorie, permettant de découvrir que le Bus représente **57,45%** du trafic total.

?

table_Dim_Route :

La dimension `Dim_Route` est la table la plus volumineuse de notre schéma en étoile. Elle sert de référentiel unique pour l'ensemble des lignes de transport (Bus et Rail) identifiées dans les réseaux de Chicago et Philadelphie.

Détails de l'implémentation :

- Consolidation Granulaire : Contrairement aux autres dimensions, cette table est extraite à partir de `Table.Combine({chicago_route, philadelphie_route})`. Cela garantit que chaque ligne spécifique (Route) est répertoriée.
- Extraction et Unicité : Après avoir isolé la colonne `Route`, la fonction `Table.Distinct` a été appliquée pour supprimer les occurrences multiples. Cela a permis d'identifier précisément les **194 routes uniques** qui composent notre jeu de données.
- Clé de Liaison (`ID_Route`) : L'ajout d'un index numérique `ID_Route` est crucial ici. Compte tenu du nombre élevé de lignes, l'utilisation d'une clé primaire numérique optimise les performances des filtres croisés et des recherches dans le dashboard.
- Capacité de Filtrage : Cette dimension alimente les visuels de détail (Top Routes, Bar Charts), permettant à l'utilisateur final de sélectionner une ligne précise pour voir son évolution historique et sa performance par rapport à la moyenne.

4. Schéma Relationnel et Modélisation (Star Schema)

Une fois les tables préparées dans Power Query, nous avons structuré les données selon un **Schéma en Étoile**. Bien que nous ayons générée des index lors de l'ETL, nous avons privilégié une modélisation basée sur les **Clés Naturelles** pour garantir une lisibilité directe des relations entre les dimensions et les faits.

A. Architecture des Relations

Le modèle repose sur des relations de type **1:N (One-to-Many)**. Cela signifie qu'une valeur unique dans une table de dimension (ex: une ville) peut être associée à plusieurs lignes de données dans les tables de faits (ex: plusieurs mois de fréquentation).

- **Direction du filtre** : Unidirectionnelle (La dimension filtre les faits).
- **Type de Clés** : Clés Naturelles (Textes et Années).

B. Détails du Mapping des Relations

Voici comment les tables communiquent entre elles :

Table de Dimension	Table de Fait (Mode / Route)	Colonne de Liaison
Dim_City	table_de_fait_mode & Route	City
Dim_Year	table_de_fait_mode & Route	Year
Dim_Month	table_de_fait_mode & Route	Month
Dim_Mode	table_de_fait_mode	Mode
Dim_Route	table_de_fait_Route	Route

C. Intégrité et Performance

- **Simplicité et Transparence** : L'utilisation des noms (Chicago, Bus, 2023) comme clés de liaison facilite le débogage et permet de vérifier rapidement l'intégrité des données sans passer par des tables de correspondance complexes.
- **Flexibilité** : Ce système permet d'ajouter de nouvelles données simplement en respectant la nomenclature des noms, assurant ainsi une mise à jour fluide du Dashboard.

5. Analyse et Intelligence DAX

Les mesures ont été centralisées dans une table technique nommée `_Mesures` pour faciliter la maintenance du modèle. Nous avons utilisé le langage DAX pour créer des indicateurs de performance clés (KPIs) dynamiques.

A. Indicateurs de Volume (Base)

Ces mesures permettent de quantifier l'activité globale sur les réseaux de transport.

- **Total Ridership** : La somme globale de tous les passagers.

$$\text{Total Ridership} = \text{SUM('table_de_fait_mode'[Ridership])}$$

- **Max Ridership** : Identifie le pic de fréquentation (ex: 136K), utile pour le Benchmarking des lignes.

Max Ridership = MAX('table_de_fait_route'[Ridership])

B. Analyses Temporelles & Croissance

(?)

Pour mesurer l'évolution de la performance, nous avons mis en place des calculs de comparaison :

- **Growth % (Croissance)** : Calcule la variation du trafic d'une période à l'autre.
- **Moyenne par Ville** : Permet de comparer l'efficacité des réseaux entre Chicago et Philadelphie.

C. Visualisation des Insights (Dashboard)

Grâce à ces mesures, le dashboard affiche des informations stratégiques instantanées :

1. **Répartition par Mode** : Le Bus domine avec **57,45%** de parts de marché.
2. **Top Routes** : Identification automatique de la ligne leader (ex: **L1**).
3. **Saisonnalité** : Mise en évidence d'une chute d'activité systématique au mois **12** pour les deux villes, suggérant des facteurs climatiques ou de vacances.

(?)

6. Conclusion

Ce projet de Business Intelligence a permis de transformer des données brutes et hétérogènes (RDF, Excel, CSV) en un outil d'aide à la décision puissant pour le transport urbain.

Résultats Clés :

- **Centralisation** : Réunir deux grandes métropoles (Chicago et Philadelphie) dans un modèle unique.
- **Performance** : Analyse de plus de **2 milliards de trajets** avec une fluidité optimale grâce au schéma en étoile.
- **Insight Majeur** : Identification d'une corrélation entre la saisonnalité (Mois 12) et la baisse de fréquentation, ainsi qu'une domination du mode Bus (57%).

Perspectives d'évolution :

- **Intégration Temps Réel** : Connecter le dashboard à une API pour suivre le trafic en direct.
- **Analyse Prédictive** : Utiliser le Machine Learning pour prévoir la fréquentation des mois à venir en fonction des événements urbains.
- **Géolocalisation** : Intégrer des cartes SIG (GIS) pour visualiser les routes avec leurs coordonnées géographiques exactes.

(?)