Exploration et Extraction de Données



Un Aperçu Complet du Data Scraping

« Analyse et Visualisation des Accidents de la Circulation Routière »

Encadrant Académique

* Hamdi Braiek

Équipe (Groupe 5)

* Ayoub Bouslimi
* Achraf Fakhfakh
* Temim Abidi
* Khalil Ahmed Abaza
* Maher Omrani
* Houssem Hammami

# Table des matières

Introduction générale

1. Contextualisation du Scrapping

1.1 Fonctionnement

1.2 Limites

1.3 Performances

1.4 Politesse et Bonnes Pratiques

2. Architecture et Technologie du Scrapping

2.1 Structure générale

2.2 Technologies employées

3. Crawling vs Scrapping

3.1 Différences et complémentarités

5. Choix de la Méthode de Scrapping

5.1 Critères de sélection

5.2 Impact sur les résultats

6. Logiciels & bibliothèque prêts à l’Emploi

7.Cas d’Utilisation et Techniques

8.Annexe

# Introduction générale :

L'extraction de données web, communément appelée scrapping, s'est érigée en un domaine technologique captivant et indispensable dans le paysage numérique contemporain. Ce processus automatisé, permettant d'extraire des informations spécifiques à partir de sites web, a connu une ascension fulgurante en raison de son potentiel à fournir des données cruciales pour une pléthore d'applications allant de l'analyse commerciale à la recherche académique. Toutefois, son utilisation englobe un ensemble complexe de nuances, de défis et de considérations éthiques.

Le fonctionnement du scrapping repose sur une alchimie subtile entre la programmation informatique et l'analyse des pages web. Des scripts automatisés simulent le comportement d'un utilisateur, naviguant à travers les dédales du code source HTML pour extraire des données structurées. Cette pratique peut varier entre le scrapping statique, qui se concentre sur des pages web fixes, et le scrapping dynamique, adapté à des contenus générés en temps réel.

Cependant, cette quête d'informations n'est pas dépourvue de limites. Des sites web peuvent explicitement interdire le scrapping dans leurs conditions d'utilisation, ce qui soulève des questions juridiques et éthiques. La qualité des données extraites dépend fortement de la structure du site, ajoutant une couche supplémentaire de complexité au processus. C'est dans cet équilibre délicat entre exploration des données et respect des règles édictées par les sites web que réside le cœur des défis du scrapping.

Les performances du scrapping, quant à elles, sont tributaires de plusieurs facteurs. La vitesse d'accès au site cible, la complexité de la structure HTML et la capacité à contourner les mécanismes de protection mis en place par les sites sont autant de paramètres qui impactent la réussite de cette entreprise. L'optimisation des performances implique souvent des ajustements minutieux dans la conception des scripts et une vigilance constante face aux évolutions des sites web.

Cependant, audelà des compétences techniques, le scrapping exige également une approche empreinte de politesse et de bonnes pratiques. Le nonrespect des conditions d'utilisation des sites peut entraîner des conséquences néfastes, allant de mesures de blocage à des poursuites légales. Ainsi, l'éthique dans le scrapping est inextricablement liée à la politesse numérique et à la responsabilité dans l'utilisation des données extraites.

La protection des données est une préoccupation centrale dans l'univers du scrapping. La confidentialité des utilisateurs et la sécurité des informations collectées doivent être des priorités absolues. Les praticiens doivent s'engager à éviter la collecte de données sensibles et à respecter les normes de protection de la vie privée.

L'architecture physique et logicielle du scrapping constitue un autre aspect crucial. Des serveurs dédiés, des clusters informatiques et des services cloud sont souvent mobilisés pour assurer une exécution efficace des scripts. Les librairies et les logiciels prêts à l'emploi, tels que BeautifulSoup BS, selenium et Scrapy, jouent un rôle déterminant dans la mise en œuvre de ces processus d'extraction.

Audelà des nuances techniques, le scrapping se révèle être une discipline aux utilisations vastes et diversifiées. De la collecte de données pour la recherche scientifique à la surveillance concurrentielle dans le domaine commercial, ses applications sont aussi variées que ses défis.

# Scraping (ou Data Scraping) :

Le scraping, également appelé extraction de données, est le processus d'automatisation de l'extraction d'informations spécifiques depuis des pages web. Les outils de scraping analysent le code source HTML des pages pour extraire des données structurées, telles que des tableaux, des listes, des prix, des descriptions, etc. Ces données peuvent ensuite être collectées, traitées et utilisées à diverses fins, telles que l'analyse, la visualisation, la comparaison de prix, ou l'alimentation de bases de données. Le scraping peut être effectué à l'aide de bibliothèques de programmation ou d'outils dédiés, automatisant ainsi le processus d'extraction. Il est important de noter que le scraping doit être effectué conformément aux lois et aux politiques d'utilisation éthique des données pour éviter tout problème légal ou éthique.

# Fonctionnement du Data Scraping :

Le fonctionnement du data scraping repose sur l'identification des éléments clés sur une page web, tels que les balises HTML, les classes CSS ou les identifiants uniques, qui contiennent les données recherchées. Une fois ces éléments identifiés, le robot peut automatiser le processus d'extraction, collectant ainsi des données de manière rapide et efficace. Cette technique est largement utilisée dans divers domaines, tels que le commerce électronique, la veille concurrentielle, la recherche académique et la collecte de données pour l'analyse et la prise de décision.

# Limites du Data Scraping :

Bien que le data scraping puisse être une méthode puissante pour collecter des données à partir de sources en ligne, il présente certaines limites et défis qu'il est important de prendre en compte :

1)Légalité et Éthique : L'une des principales limites du data scraping concerne les questions juridiques et éthiques. Certains sites web interdisent explicitement le scraping dans leurs conditions d'utilisation, et l'utilisation non autorisée peut entraîner des poursuites judiciaires. Il est crucial de respecter les politiques des sites web ciblés et de ne pas enfreindre les lois sur la protection des données.

2)Structure changeante des sites web : Les sites web sont souvent mis à jour et modifiés, ce qui peut entraîner des changements dans leur structure. Si un scraper est conçu pour extraire des données en se basant sur la structure actuelle d'un site, il peut devenir obsolète lorsque cette structure est modifiée. Cela nécessite une maintenance continue du scraper pour s'adapter aux évolutions des sites web.

3)Qualité des données : Le data scraping repose sur la qualité des balises et des modèles de recherche utilisés pour extraire les données. Si la structure d'une page web change, cela peut entraîner des erreurs d'extraction ou la collecte de données incorrectes. La qualité des données dépend donc de la robustesse du scraper.

4)Fréquence des requêtes : L'envoi fréquent de requêtes depuis un même serveur vers un site web peut entraîner des blocages ou des restrictions d'accès. Certains sites limitent le nombre de requêtes qu'un utilisateur peut effectuer sur une période donnée pour éviter le scraping intensif.

En résumé, bien que le data scraping soit une méthode puissante, son utilisation doit être soigneusement planifiée et réalisée de manière éthique pour éviter des problèmes légaux et s'adapter aux éventuels changements dans la structure des sites web ciblés.

# Performance du Data Scraping :

# 1)Vitesse d'extraction : Les performances du data scraping sont souvent mesurées par la vitesse à laquelle les données peuvent être extraites. Un scraper efficace doit être capable de collecter rapidement les informations nécessaires sans compromettre la qualité des données.

# 2)Robustesse : La robustesse d'un scraper fait référence à sa capacité à fonctionner de manière fiable même face à des changements dans la structure des sites web ciblés. Un scraper robuste peut s'adapter aux modifications sans nécessiter une intervention constante.

# 3)Éviter les blocages et restrictions : Un bon scraper doit être capable d'éviter les blocages ou les restrictions imposées par les sites web. Cela peut impliquer la gestion appropriée de la fréquence des requêtes, l'utilisation de proxies, ou l'émulation d'un comportement humain pour ne pas être détecté comme un bot.

# 4)Précision : Les performances du data scraping incluent également la précision dans l'extraction des données. Un scraper précis doit être capable de collecter des informations exactes et complètes conformément aux besoins spécifiques de l'utilisateur.

# 5)Gestion des erreurs : La capacité à gérer les erreurs est cruciale pour maintenir des performances stables. Un scraper bien conçu doit être capable de gérer les éventuelles erreurs de connexion, les changements de structure de page, ou d'autres problèmes rencontrés pendant l'extraction.

# 6)Éviter la détection : Certains sites web mettent en place des mécanismes de détection pour identifier et bloquer les activités de scraping. Un scraper performant doit être capable de contourner ces mécanismes de détection pour maintenir sa discrétion.

# Il est important de noter que les performances du data scraping dépendent également de la manière dont il est utilisé et des politiques du site web ciblé. Un usage éthique et conforme aux conditions d'utilisation des sites est essentiel pour éviter des problèmes juridiques et maintenir des performances durables.

# Politesse et bonnes pratiques

Lorsque vous pratiquez le data scraping, il est essentiel de suivre des politesses et bonnes pratiques pour assurer un usage éthique et respectueux de cette technique. Voici quelques conseils à considérer :

1)Respectez les conditions d'utilisation : Avant de scraper un site web, assurez-vous de lire et de comprendre les conditions d'utilisation. Certains sites interdisent explicitement le scraping dans leurs termes, tandis que d'autres peuvent avoir des restrictions spécifiques. Le non-respect de ces conditions peut entraîner des conséquences légales.

2)Utilisez des headers HTTP appropriés : Configurez les entêtes HTTP de vos requêtes de manière à imiter le comportement d'un navigateur standard. Cela peut aider à éviter d'être détecté comme un bot et à réduire le risque de blocage par le site web.

3)Gérez la fréquence des requêtes : Limitez la fréquence de vos requêtes pour éviter de surcharger le serveur du site web cible. L'utilisation excessive peut entraîner des blocages ou des restrictions d'accès. Respectez les directives du fichier robots.txt du site s'il en existe un.

4)Évitez le scraping intensif : Évitez de scraper de grandes quantités de données en une seule fois. Un scraping intensif peut affecter les performances du site web cible et est généralement mal vu.

En suivant ces politesses et bonnes pratiques, vous contribuez à maintenir une utilisation éthique du data scraping, évitez les problèmes juridiques potentiels, et favorisez une relation respectueuse avec les propriétaires des sites web ciblés.

# Architecture du scrapping

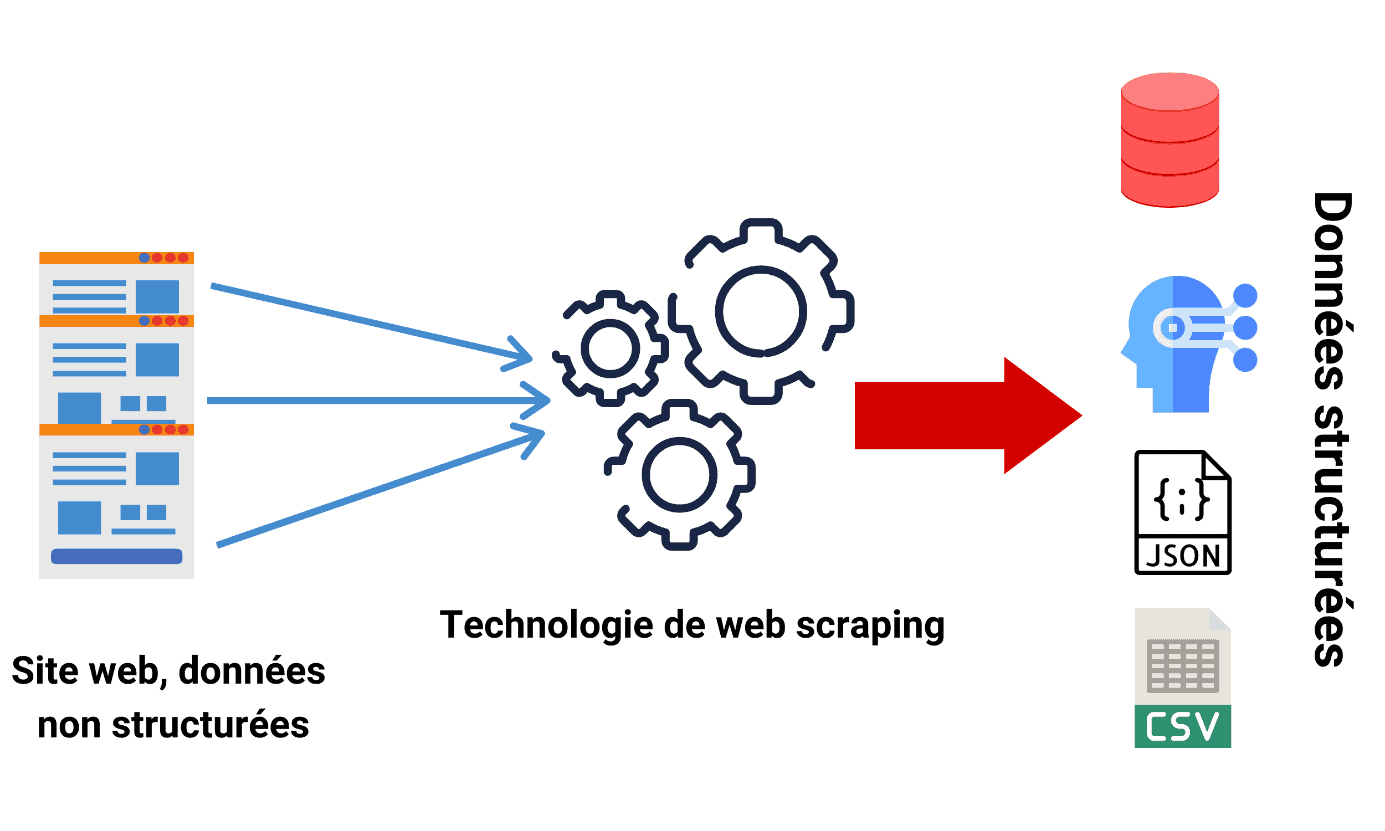


Image 1 : Architecture de data scraping

Le data scraping est une technique qui permet d'extraire des données d'un site web de manière automatisée. Cette technique est utilisée dans de nombreux domaines, tels que l'e-commerce, le marketing, la finance ou la recherche.

L'architecture de base d'un système de data scraping est composée de trois éléments principaux :

* Le client : il s'agit de l'application qui lance le processus de scraping. Il peut s'agir d'une application web, d'une application mobile ou d'un script.
* La technologie de scraping : il s'agit du logiciel qui extrait les données du site web. Cette technologie peut être basée sur des bibliothèques ou des Framework open source, ou sur des solutions propriétaires.
* Le serveur de données : il s'agit du système qui stocke les données extraites. Le serveur de données peut être un système relationnel, un système NoSQL ou un système de stockage en masse.

Le client

Le client est l'élément qui gère les interactions entre l'utilisateur et le système de scraping. Il est responsable de spécifier les données à extraire, les pages web à scraper et la fréquence des scraping.

# La technologie de scraping

La technologie de scraping est responsable de l'extraction des données du site web. Elle utilise des techniques telles que l'analyse XPath ou l'analyse CSS pour identifier les données à extraire.

Le serveur de données est responsable du stockage des données extraites. Il peut être utilisé pour indexer les données afin de faciliter leur recherche et leur analyse.

Les données extraites peuvent être de deux types :

* Les données structurées sont des données organisées dans un format prédéfini, comme un tableau ou un objet JSON.
* Les données non structurées sont des données non organisées, comme du texte ou des images.

Les données structurées sont généralement plus faciles à traiter que les données non structurées. Elles peuvent être importées directement dans un système de données relationnel. Les données non structurées nécessitent souvent un traitement supplémentaire avant de pouvoir être utilisées.

# Crawling vs scrapping

Le crawling (exploration web) et le scraping (extraction de données) sont deux concepts distincts, bien que souvent utilisés conjointement dans le contexte de l'analyse de données web.

1. Crawling (Exploration Web) :

Objectif : Le crawling consiste à parcourir systématiquement le contenu du web, en suivant les liens d'une page à une autre.

Fonctionnement : Un robot d'exploration web (crawler) démarre à partir d'une URL de départ, puis suit les liens vers d'autres pages, créant ainsi une vaste indexation du contenu du web.

Exemple d'outil : Google bot est un exemple de crawler utilisé par les moteurs de recherche pour explorer et indexer des pages web.

2. Scraping (Extraction de Données) :

Objectif : Le scraping consiste à extraire des données spécifiques à partir des pages web, souvent pour des besoins d'analyse ou de stockage.

Fonctionnement : Les outils de scraping utilisent des techniques automatisées pour extraire des informations structurées à partir du code HTML des pages web.

Exemple d'outil : Beautiful Soup, Scrapy, et Selenium sont des exemples d'outils de scraping couramment utilisés.

En résumé, le crawling se concentre sur la navigation à travers le web pour indexer son contenu, tandis que le scraping se concentre sur l'extraction ciblée de données à partir des pages web. Les deux processus sont souvent utilisés ensemble pour construire des moteurs de recherche, collecter des données pour l'analyse, ou automatiser des tâches liées à la récupération d'informations en ligne.

Choix de la méthode :  
Dans le cadre de notre projet, nous avons choisi d'utiliser le data scraping pour collecter des données sur les accidents routiers à partir de divers sites web spécialisés. Cette décision découle de la nécessité d'obtenir des informations spécifiques, telles que la localisation, les circonstances, et les statistiques associées aux incidents, afin d'analyser et de comprendre les tendances en matière de sécurité routière.

Le data scraping s'avère être l'approche idéale pour notre cas, car elle offre une flexibilité dans la collecte de données structurées à partir du code HTML des pages dédiées aux rapports d'accidents. Cette méthode automatisée nous permettra d'extraire sélectivement les détails pertinents, tout en nous donnant la capacité d'ajuster rapidement nos scripts en cas de modifications dans la structure des pages.

En outre, le data scraping facilite l'enrichissement des données en nous permettant d'agréger des informations provenant de différentes sources, élargissant ainsi la portée de notre analyse. Cette approche nous offre la possibilité d'obtenir des données actualisées en temps réel, ce qui est essentiel pour une compréhension dynamique des modèles d'accidents routiers.

Cependant, nous sommes pleinement conscients des responsabilités éthiques qui accompagnent le data scraping. Nous nous engageons à respecter les conditions d'utilisation des sites web, à éviter toute collecte de données sensibles, et à minimiser l'impact sur les performances des serveurs sources. En adoptant cette approche, notre projet vise à fournir des données précises et pertinentes tout en préservant l'intégrité et la légitimité de notre démarche.

# Logiciel prêt à l’emploi

Selenium, Beautiful Soup, et Scrapy sont trois bibliothèques Python populaires utilisées dans le domaine du web scraping. Chacune de ces bibliothèques a ses propres caractéristiques et avantages, et elles sont souvent utilisées en combinaison pour des projets de scraping complets.

1. Selenium :

Objectif : Selenium est principalement utilisé pour l'automatisation des navigateurs. Il permet de contrôler un navigateur web (comme Chrome, Firefox, etc.) via un script Python.

Fonctionnalités : Selenium est utile lorsque le site web utilise des technologies telles que JavaScript pour générer du contenu de manière dynamique. Il peut simuler l'interaction humaine avec le navigateur, remplir des formulaires, cliquer sur des boutons, et charger du contenu après le chargement initial de la page.

Utilisation typique : Selenium est souvent utilisé pour les sites web interactifs où le contenu est généré dynamiquement après le chargement initial de la page.



Image 2: selenium logo

2. Beautiful Soup:

Objectif : Beautiful Soup est une bibliothèque Python conçue pour extraire des informations à partir de documents HTML et XML.

Fonctionnalités : Beautiful Soup analyse le code HTML et XML, crée un arbre de parseurs, puis permet aux utilisateurs de naviguer facilement dans la structure du document pour extraire des données.

 Utilisation typique : Beautiful Soup est souvent utilisé lorsque le HTML est statique et que le site web n'utilise pas beaucoup de JavaScript pour générer son contenu.

Image 3 : Beautiful Soup logo

3. Scrapy :

Objectif : Scrapy est un Framework open source dédié au web scraping. Il est utilisé pour créer des spiders, des scripts qui parcourent les sites web et extraient des données.

Fonctionnalités : Scrapy fournit un ensemble d'outils pour gérer le processus de scraping, gérer les requêtes HTTP, stocker les données extraites, et suivre les liens entre les pages. Il peut être utilisé de manière efficace pour le scraping de sites web complexes et de grande envergure.

Utilisation typique : Scrapy est souvent utilisé pour des projets de scraping plus complexes où la structure des données à extraire est bien définie, et où la navigation à travers plusieurs pages est nécessaire.

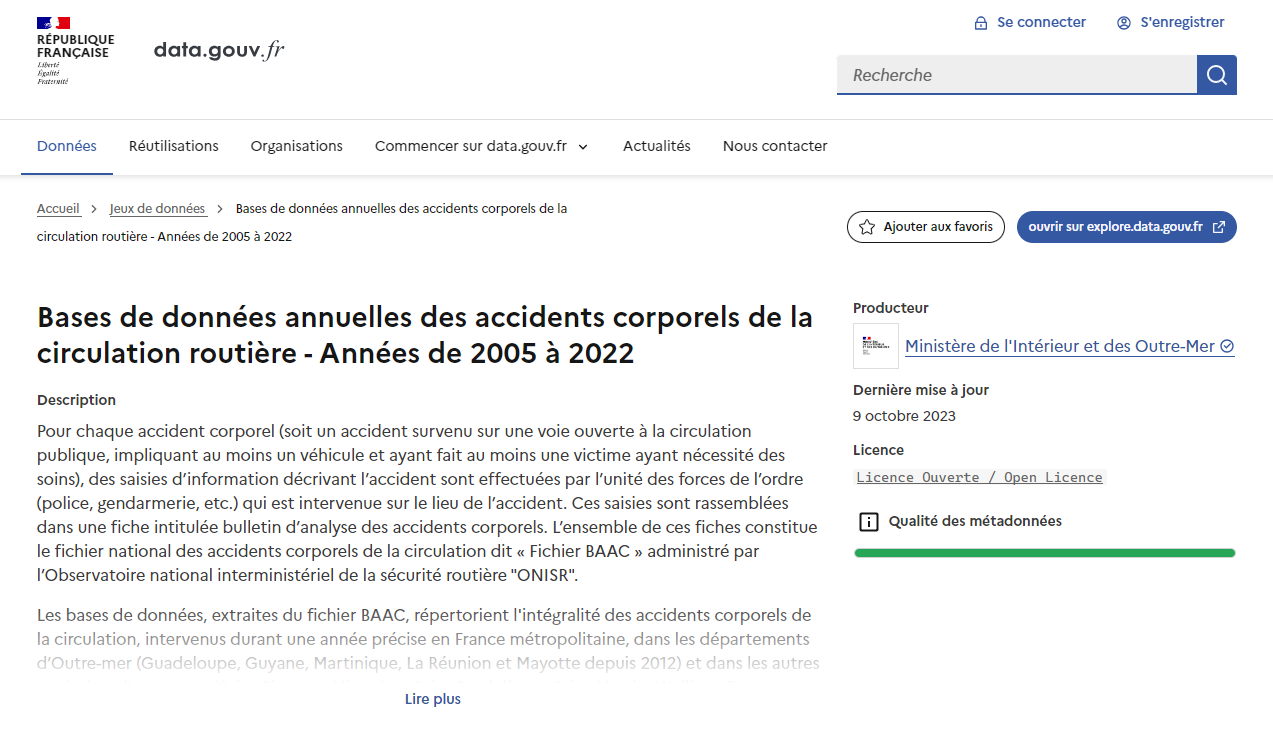
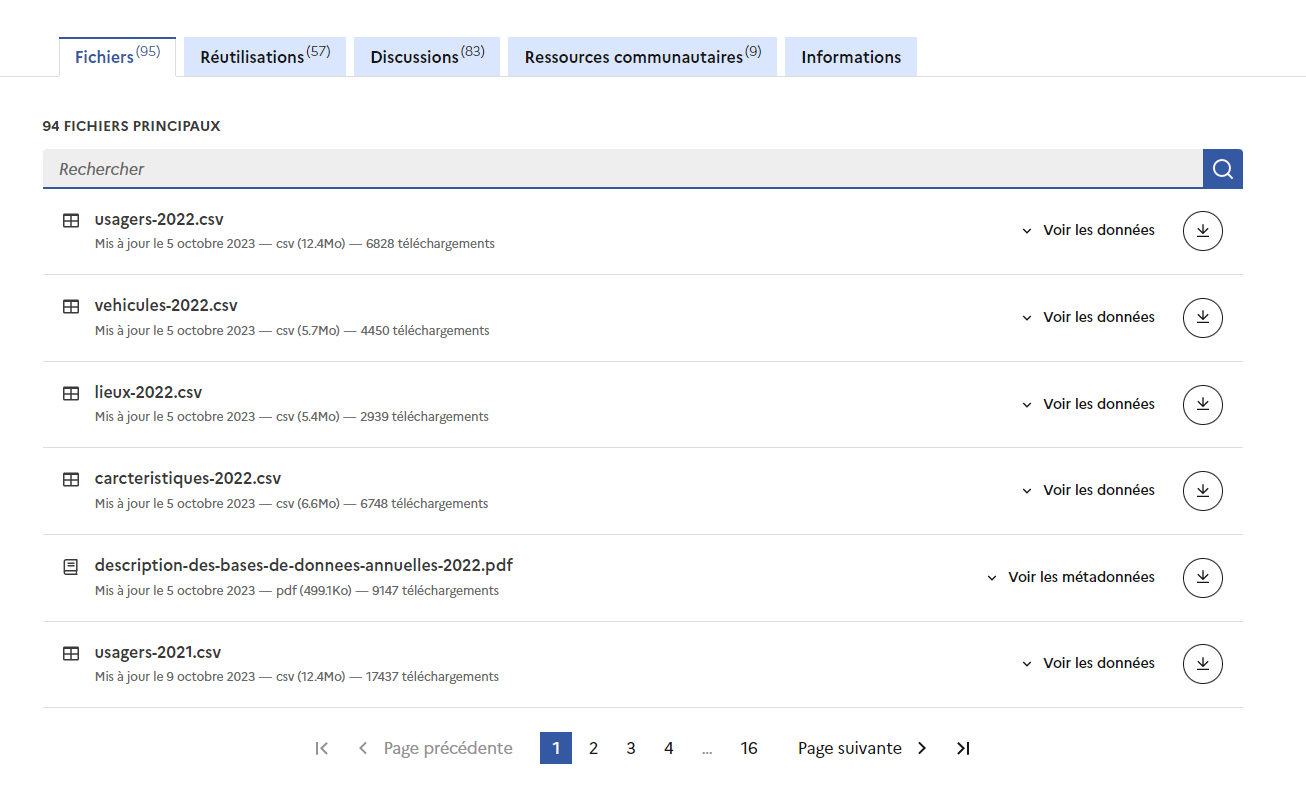


Image 4 : logo Scrapy

En résumé, ces bibliothèques offrent des fonctionnalités complémentaires et peuvent être combinées de manière judicieuse en fonction des besoins spécifiques d'un projet

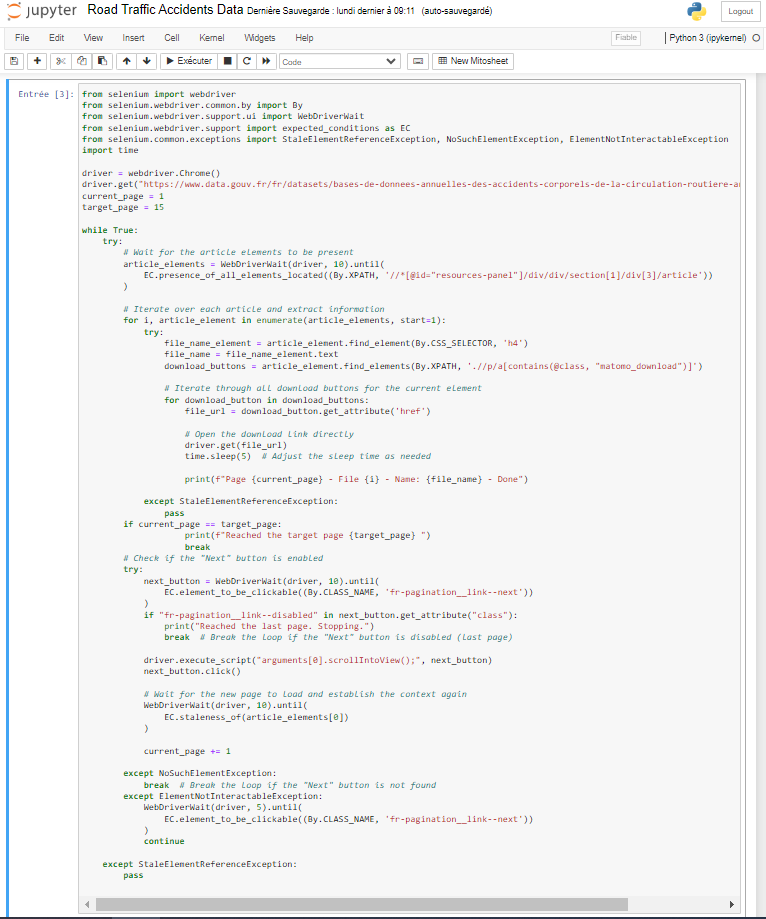
# Cas d’utilisation et techniques

Cas : <https://www.data.gouv.fr/> par **Ayoub Bouslimi**



Description du site :  
Le site data.gouv.fr est une plateforme gouvernementale française qui offre un accès transparent et ouvert à une multitude de données publiques. En ce qui concerne les statistiques sur les accidents routiers, le portail propose des ensembles de données détaillés, généralement au format CSV, fournissant des informations cruciales telles que la localisation, la date, les types d'accidents, et d'autres paramètres liés à la sécurité routière. La convivialité de l'interface facilite la recherche, le téléchargement et l'analyse de ces données, offrant ainsi aux chercheurs, aux professionnels de la sécurité routière et au grand public un accès précieux pour mieux comprendre et améliorer la sécurité sur les routes en France.

### Code source :

Partie de la code source, vous trouverez la suite sur a page annexe.

### 

### Explication du code :

Notre code a pour but d’automatiser le processus d'extraction de fichiers à partir d'une page web spécifique sur le site data.gouv.fr. Voici une explication détaillée de son fonctionnement :

1. Initialisation et Navigation : Le script démarre un navigateur Chrome via Selenium et se rend sur la page spécifiée contenant les bases de données annuelles des accidents corporels de la circulation routière.

2. Boucle Principale : Le script utilise une boucle while infinie pour itérer à travers les pages de la liste de fichiers à extraire. La variable `target\_page` détermine jusqu'à quelle page le scraping doit être effectué.

3. Extraction des Informations : À l'intérieur de la boucle, le script utilise des attentes explicites pour s'assurer que les éléments d'article sont présents sur la page. Il extrait ensuite les noms de fichiers et les liens de téléchargement associés.

4. Téléchargement des Fichiers : Pour chaque fichier, le script ouvre directement le lien de téléchargement pour obtenir le fichier. Il utilise une pause de 5 secondes entre chaque téléchargement pour éviter tout problème potentiel lié aux limitations du serveur.

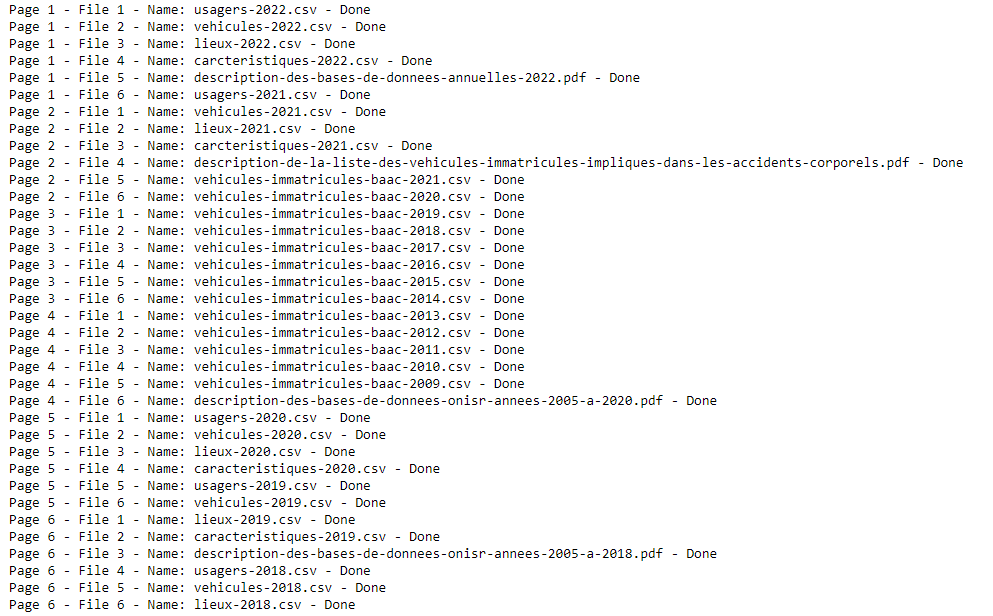
5. Gestion des Exceptions : Le script gère les exceptions telles que les éléments stalles, les éléments non interactifs, et les éléments introuvables. Ces gestionnaires d'exceptions permettent au script de continuer à fonctionner même en présence de problèmes mineurs.

6. Pagination : Le script clique sur le bouton "Next" pour passer à la page suivante de la liste. Il vérifie également si le bouton "Next" est désactivé, indiquant que la dernière page a été atteinte, et arrête la boucle en conséquence.

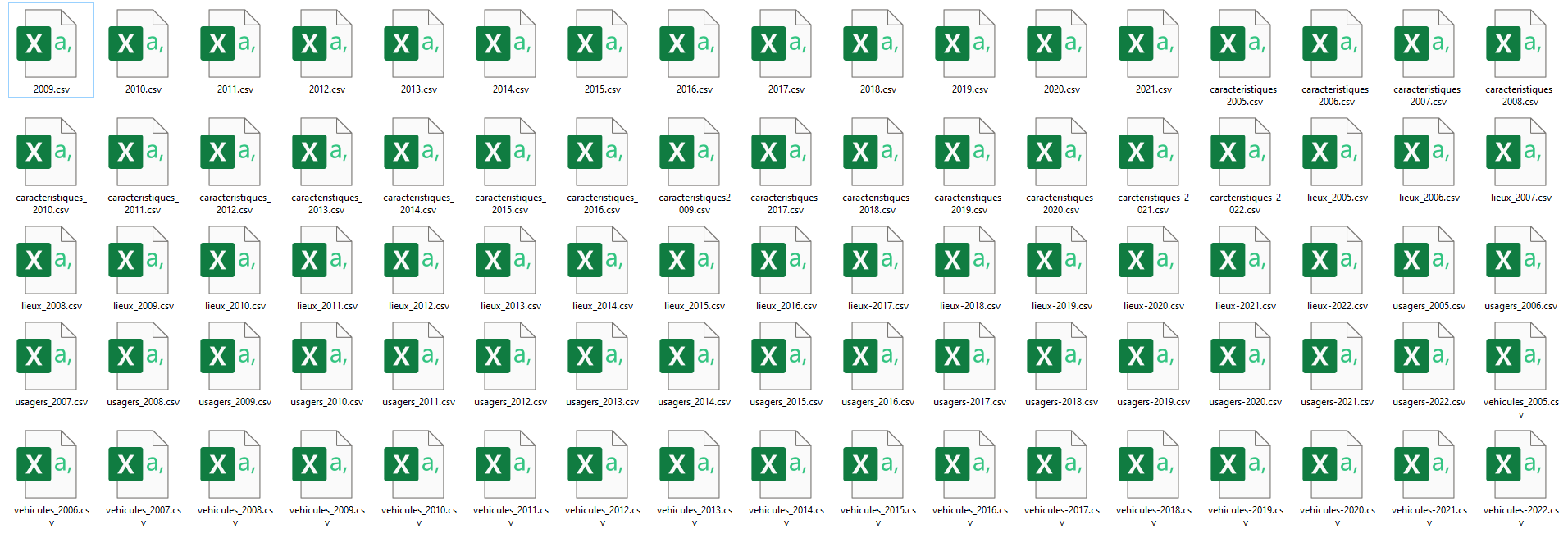
En résumé, ce script automatisé utilise Selenium pour extraire des données sur les accidents routiers du site data.gouv.fr. Il parcourt les pages, extrait les informations nécessaires, puis télécharge les fichiers associés. Ce processus est répété jusqu'à ce que la page cible spécifiée soit atteinte.

### Résultat d’exécution :

Suite à l'exécution du script de scraping sur le site data.gouv.fr, les résultats obtenus sont significatifs. Au total, 87 fichiers CSV ont été extraits, fournissant des données détaillées sur les accidents corporels de la circulation routière couvrant les années de 2005 à 2022. Ces fichiers contiennent des informations cruciales telles que la localisation des accidents, les types d'accidents, les conditions météorologiques, et d'autres paramètres essentiels à l'analyse de la sécurité routière en France. En plus des fichiers CSV, des fichiers PDF ont également été fournis, offrant un contexte et des explications supplémentaires pour comprendre les données extraites. Ces documents peuvent servir de référence précieuse pour interpréter les résultats du scraping et fournir des informations contextuelles sur les différentes catégories de données. Par ailleurs, pour garantir un suivi efficace de l'exécution du code, un fichier journal (log) a été généré. Ce log enregistre en détail les étapes du script, fournissant des informations sur les pages visitées, les fichiers extraits, ainsi que les éventuelles erreurs rencontrées pendant le processus de scraping. Ce journal s'avère essentiel pour assurer la transparence, la traçabilité et la résolution rapide de problèmes éventuels lors de futures analyses ou ajustements du script.

Voilà un exemple de Log :

Vous trouverez le reste du journal (log) sur la page annexe fournie avec son code.

Fichiers scrapper :

# Compréhension et description des données

### Définitions du fichier national des données BAAC Bulletins d’Analyse des Accidents Corporels de la circulation

**Un accident corporel** (mortel et non mortel) de la circulation routière relevé par les forces de l’ordre :

* Implique au moins une victime,
* Survient sur une voie publique ou privée, ouverte à la circulation publique,
* Implique au moins un véhicule.

Un accident corporel implique un certain nombre d’usagers. Parmi ceux-ci, on distingue :

Les personnes **indemnes** : impliquées non décédées et dont l’état ne nécessite aucun soin médical du fait de l’accident,

Les **victimes** : impliquées non indemnes.

Les personnes **tuées** : personnes qui décèdent du fait de l’accident, sur le coup ou dans les trente jours qui suivent l’accident,

Les personnes **blessées** : victimes non tuées.

Les blessés dits « **hospitalisés** » : victimes hospitalisées plus de 24 heures,

Les blessés **légers** : victimes ayant fait l'objet de soins médicaux mais n'ayant pas été admises comme patients à l'hôpital plus de 24 heures.

### Spécifications de la base

La base Etalab de données des accidents corporels de la circulation d'une année donnée, est répartie en 4 rubriques sous la forme pour chacune d'elles d'un fichier au format csv.

1. La rubrique **CARACTERISTIQUES** qui décrit les circonstances générales de l’accident
2. La rubrique **LIEUX** qui décrit le lieu principal de l’accident même si celui-ci s’est déroulé à une intersection
3. La rubrique **VEHICULES** impliqués
4. La rubrique **USAGERS** impliqués

Chacune des variables contenues dans une rubrique doit pouvoir être reliée aux variables des autres rubriques. Le n° d'identifiant de l’accident ("Num\_Acc") présent dans ces 4 rubriques permet d'établir un lien entre toutes les variables qui décrivent un accident. Quand un accident comporte plusieurs véhicules, il faut aussi pouvoir relier chaque véhicule à ses occupants. Ce lien est fait par la variable id\_vehicule.

La plupart des variables contenues dans les quatre fichiers précédemment énumérés peuvent contenir des cellules vides ou un zéro ou un point. Il s’agit, dans ces trois cas, d’une cellule non renseignée par les forces de l'ordre ou sans objet.

Liste complète des champs avec le détail de leur contenu pour chaque fichier

En 2019, la base de données des accidents a évolué, dans le descriptif cidessous, en vert les nouvelles modalités de certaines variables et les nouvelles variables ajoutées.

**La rubrique CARACTERISTIQUES**

Num\_Acc : Numéro d'identifiant de l’accident.

Jour : Jour de l'accident.

Mois : Mois de l'accident.

An : Année de l'accident.

Hrmn Heure et minutes de l'accident.

Lum : Lumière : conditions d’éclairage dans lesquelles l'accident s'est produit :

1 Plein jour / 2 Crépuscule ou aube / 3 Nuit sans éclairage public /4 Nuit avec éclairage public

Dep : Département : Code INSEE (Institut National de la Statistique et des Etudes Economiques)

Com : Commune : Le numéro de commune est un code donné par l‘INSEE. Le code est composé du code INSEE du département suivi par 3 chiffres.

Agg : Localisation : 1 Hors agglomération / 2 En agglomération

Int : Intersection :

1 Hors intersection / 2 Intersection en X / 3 Intersection en T / 4 Intersection en Y

5 Intersection à plus de 4 branches / 6 Giratoire / 7 Place / 8 Passage à niveau

9 Autre intersection

Atm : Conditions atmosphériques :

1 Non renseigné / 1 Normale / 2 Pluie légère / 3 Pluie forte / 4 Neige / 5 Brouillard fumée

6 Vent fort tempête / 7 Temps éblouissant / 8 Temps couvert / 9 Autre

Col : Type de collision : 1 Non renseigné / 1 Deux véhicules frontale / 2 Deux véhicules par l’arrière /3 Deux véhicules par le coté /4 Trois véhicules et plus en chaîne

/5 Trois véhicules et plus collisions multiples / 6 Autre collision /7 Sans collision

adr : Adresse postale : variable renseignée pour les accidents survenus en agglomération.

Lat : Latitude

Long : Longitude

**La rubrique LIEUX**

Num\_Acc : Identifiant de l’accident identique à celui du fichier "rubrique CARACTERISTIQUES" repris dans l’accident.

Catr : Catégorie de route : 1 Autoroute /2 Route nationale /3 Route Départementale /4 Voie Communales /5 Hors réseau public /6 Parc de stationnement ouvert à la circulation publique

/7 Routes de métropole urbaine /9 autre voie

V1 : Indice numérique du numéro de route (exemple : 2 bis, 3 ter etc.).

V2 : Lettre indice alphanumérique de la route.

Circ : Régime de circulation : 1 Non renseigné / 1 A sens unique /2 Bidirectionnelle   
/3 A chaussées séparées /4 Avec voies d’affectation variable

Nbv : Nombre total de voies de circulation.

Vosp : Signale l’existence d’une voie réservée, indépendamment du fait que l’accident ait lieu ou non sur cette voie.

1 Non renseigné / 0 Sans objet /1 Piste cyclable /2 Bande cyclable /3 Voie réservée

Prof : Profil en long décrit la déclivité de la route à l'endroit de l'accident :

1 Non renseigné / 1 Plat /2 Pente /3 Sommet de côte /4 Bas de côte

Pr : Numéro du PR de rattachement (numéro de la borne amont). La valeur 1 signifie que le PR n’est pas renseigné.

Pr1 : Distance en mètres au PR (par rapport à la borne amont). La valeur 1 signifie que le PR n’est pas renseigné.

Plan : Tracé en plan : 1 Non renseigné / 1 Partie rectiligne /2 En courbe à gauche /3 En courbe à droite / 4 En « S »

Lartpc : Largeur du terreplein central (TPC) s'il existe (en m).

Larrout : Largeur de la chaussée affectée à la circulation des véhicules ne sont pas compris les bandes d'arrêt d'urgence, les TPC et les places de stationnement (en m).

Surf : Etat de la surface : 1 Non renseigné / 1 Normale /2 Mouillée /3 Flaques /4 Inondée

/5 Enneigée /6 Boue /7 Verglacée /8 Corps gras huile /9 Autre

Infra : Aménagement Infrastructure : -1 Non renseigné /0 Aucun / 1 Souterrain tunnel   
/2 Pont autopont /3 Bretelle d’échangeur ou de raccordement /4 Voie ferrée

5 Carrefour aménagé /6 Zone piétonne /7 Zone de péage /8 Chantier /9 Autres

Situ : Situation de l’accident : -1 Non renseigné /0 Aucun/1 Sur chaussée

/2 Sur bande d’arrêt d’urgence /3 Sur accotement/4 Sur trottoir /5 Sur piste cyclable

/6 Sur autre voie spéciale /7 Autres

Vma : Vitesse maximale autorisée sur le lieu et au moment de l’accident.

**La rubrique VÉHICULES**

Num\_Acc

Identifiant de l’accident identique à celui du fichier "rubrique CARACTERISTIQUES" repris pour chacun des véhicules décrits impliqués dans l’accident.

id\_vehicule

Identifiant unique du véhicule repris pour chacun des usagers occupant ce véhicule (y compris les piétons qui sont rattachés aux véhicules qui les ont heurtés) Code numérique.

Num\_Veh

Identifiant du véhicule repris pour chacun des usagers occupant ce véhicule (y compris les piétons qui sont rattachés aux véhicules qui les ont heurtés) Code alphanumérique.

Senc : Sens de circulation : -1 Non renseigné /0 Inconnu

/1 PK ou PR ou numéro d’adresse postale croissant

/2PK ou PR ou numéro d’adresse postale décroissant /3 Absence de repère

Catv : Catégorie du véhicule :

00 Indéterminable

01 Bicyclette

02 Cyclomoteur <50cm3

03 Voiturette (Quadricycle à moteur carrossé) (anciennement "voiturette ou tricycle à moteur")  
04 Référence inutilisée depuis 2006 (scooter immatriculé)

05 Référence inutilisée depuis 2006 (motocyclette)  
06 Référence inutilisée depuis 2006 (sidecar)

07 VL seul

08 Référence inutilisée depuis 2006 (VL + caravane)   
09 Référence inutilisée depuis 2006 (VL + remorque)

10 VU seul 1,5T <= PTAC <= 3,5T avec ou sans remorque (anciennement VU seul 1,5T <= PTAC

<= 3,5T)

11 Référence inutilisée depuis 2006 (VU (10) + caravane)   
12 Référence inutilisée depuis 2006 (VU (10) + remorque)   
13 PL seul 3,5T <PTCA <= 7,5T

14 PL seul > 7,5T

15 PL > 3,5T + remorque   
16 Tracteur routier seul

17 Tracteur routier + semiremorque

18 Référence inutilisée depuis 2006 (transport en commun)   
19 Référence inutilisée depuis 2006 (tramway)

20 Engin spécial

21Tracteur agricole 30 Scooter < 50 cm3

31 Motocyclette > 50 cm3 et <= 125 cm3 32 Scooter > 50 cm3 et <= 125 cm3

33 Motocyclette > 125 cm3

34 Scooter > 125 cm3

35 Quad léger <= 50 cm3 (Quadricycle à moteur non carrossé) 36 Quad lourd > 50 cm3 (Quadricycle à moteur non carrossé) 37 Autobus

38 Autocar

39 Train

40 Tramway/41 3RM <= 50 cm3 /42 3RM > 50 cm3 <= 125 cm3 /43 3RM > 125 cm3 /50 EDP à moteur /60 EDP sans moteur 80 VAE /99 Autre véhicule

Obs : Obstacle fixe heurté : 1 Non renseigné /0 Sans objet /1 Véhicule en stationnement

/2 Arbre /3 Glissière métallique /4 Glissière béton /5 Autre glissière

/6Bâtiment, mur, pile de pont

/7Support de signalisation verticale ou poste d’appel d’urgence /8Poteau

/9 Mobilier urbain /10 Parapet

/11 Ilot, refuge, borne haute /12 Bordure de trottoir /13 Fossé, talus, paroi rocheuse

/14 Autre obstacle fixe sur chaussée /15 Autre obstacle fixe sur trottoir ou accotement

/16 Sortie de chaussée sans obstacle /17 Buse tête d’aqueduc

Obsm Obstacle mobile heurté : 1 Non renseigné /0 Aucun /1 Piéton /2 Véhicule /4 Véhicule sur rail /5Animal domestique /6 Animal sauvage/9 Autre

Choc Point de choc initial :

1 Non renseigné /0 Aucun /1 Avant /2 Avant droit /3 Avant gauche /4 Arrière /5 Arrière droit

/6 Arrière gauche /7 Côté droit /8 Côté gauche /9 Chocs multiples (tonneaux)

Manv Manoeuvre principale avant l’accident :

1 Non renseigné /0 Inconnue /1 Sans changement de direction/ 2Même sens, même file

3 Entre 2 files/4 En marche arrière/5 A contresens/6 En franchissant le terreplein central

7 Dans le couloir bus, dans le même sens /8 Dans le couloir bus, dans le sens inverse

9 En s’insérant /10 En faisant demitour sur la chaussée Changeant de file

11 A gauche /12 A droite Déporté /13 A gauche /14 A droite Tournant

15 A gauche /16 A droite Dépassant/17 A gauche /18 A droite Divers

19Traversant la chaussée/20 Manœuvre de stationnement /21 Manœuvre d’évitement

22Ouverture de porte /23Arrêté (hors stationnement)

24En stationnement (avec occupants /25 Circulant sur trottoir /26 Autres manœuvres

Motor Type de motorisation du véhicule :

1 Non renseigné 0 Inconnue 1 Hydrocarbures 2 Hybride électrique 3 Electrique

4 Hydrogène 5 Humaine 6 Autre

Occutc : Nombre d’occupants dans le transport en commun.

**La rubrique USAGERS**

Num\_Acc

Identifiant de l’accident identique à celui du fichier "rubrique CARACTERISTIQUES" repris pour chacun des usagers décrits impliqués dans l’accident.

id\_usager

Identifiant unique de l’usager (y compris les piétons qui sont rattachés aux véhicules qui les ont heurtés) Code numérique.

id\_vehicule

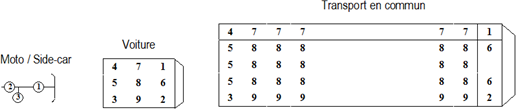
Identifiant unique du véhicule repris pour chacun des usagers occupant ce véhicule (y compris les piétons qui sont rattachés aux véhicules qui les ont heurtés) Code numérique.

num\_Veh

Identifiant du véhicule repris pour chacun des usagers occupant ce véhicule (y compris les piétons qui sont rattachés aux véhicules qui les ont heurtés) Code alphanumérique.

Place

Permet de situer la place occupée dans le véhicule par l'usager au moment de l'accident. Le détail est donné par l’illustration cidessous :



10 Piéton (non applicable)

catu

Catégorie d'usager :

1 Conducteur

2 Passager

3 Piéton

grav

Gravité de blessure de l'usager, les usagers accidentés sont classés en trois catégories de victimes plus les indemnes :

1 Indemne

2 Tué

3 Blessé hospitalisé 4 Blessé léger

sexe

Sexe de l'usager :

1 Masculin

2 Féminin

An\_nais

Année de naissance de l'usager.

trajet

Motif du déplacement au moment de l’accident :

1 Non renseigné 0 Non renseigné

1 Domicile travail 2 Domicile école 3 Courses achats

4 Utilisation professionnelle 5 Promenade loisirs

9 Autre

Les équipements de sécurité jusqu’en 2018 étaient en 2 variables : existence et utilisation.

A partir de 2019, il s’agit de l’utilisation avec jusqu’à 3 équipements possibles pour un même usager (notamment pour les motocyclistes dont le port du casque et des gants est obligatoire).

secu1

Le renseignement du caractère indique la présence et l’utilisation de l’équipement de sécurité :

1 Non renseigné

0 Aucun équipement 1 Ceinture

2 Casque

3 Dispositif enfants 4 Gilet réfléchissant 5 Airbag (2RM/3RM)

6 Gants (2RM/3RM)

7 Gants + Airbag (2RM/3RM) 8 Non déterminable

9 Autre

locp

Localisation du piéton :

1 Non renseigné 0 Sans objet

Sur chaussée :

1 A + 50 m du passage piéton 2 A 50 m du passage piéton Sur passage piéton :

3 Sans signalisation lumineuse 4 Avec signalisation lumineuse

Divers :

5 Sur trottoir

6 Sur accotement

7 Sur refuge ou BAU

8 Sur contre allée

9 Inconnue

Actp : Action du piéton : 1 Non renseigné / 0 Non renseigné ou sans objet / 1 Sens véhicule heurtant /2 Sens inverse du véhicule Divers

3 Traversant

4 Masqué

5 Jouant courant

6 Avec animal

9 Autre

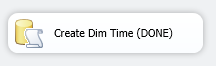
A Monte/descend du véhicule B Inconnue

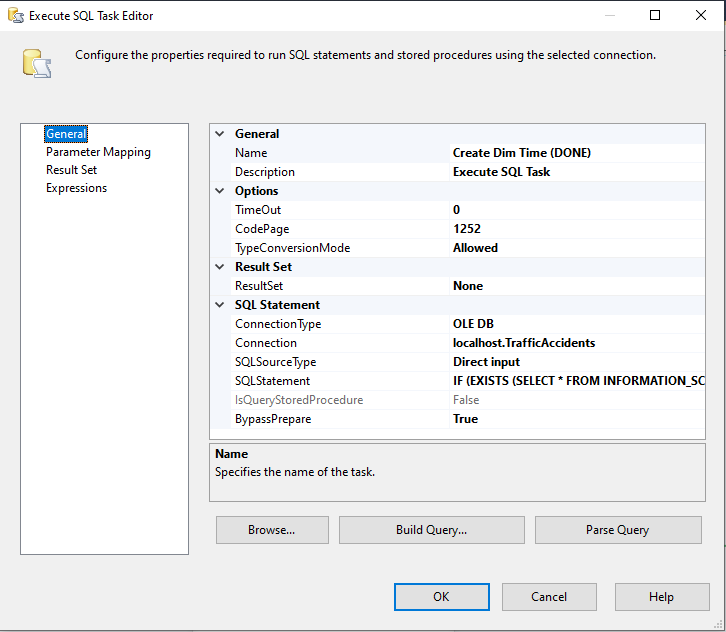
Etatp : Cette variable permet de préciser si le piéton accidenté était seul ou non :

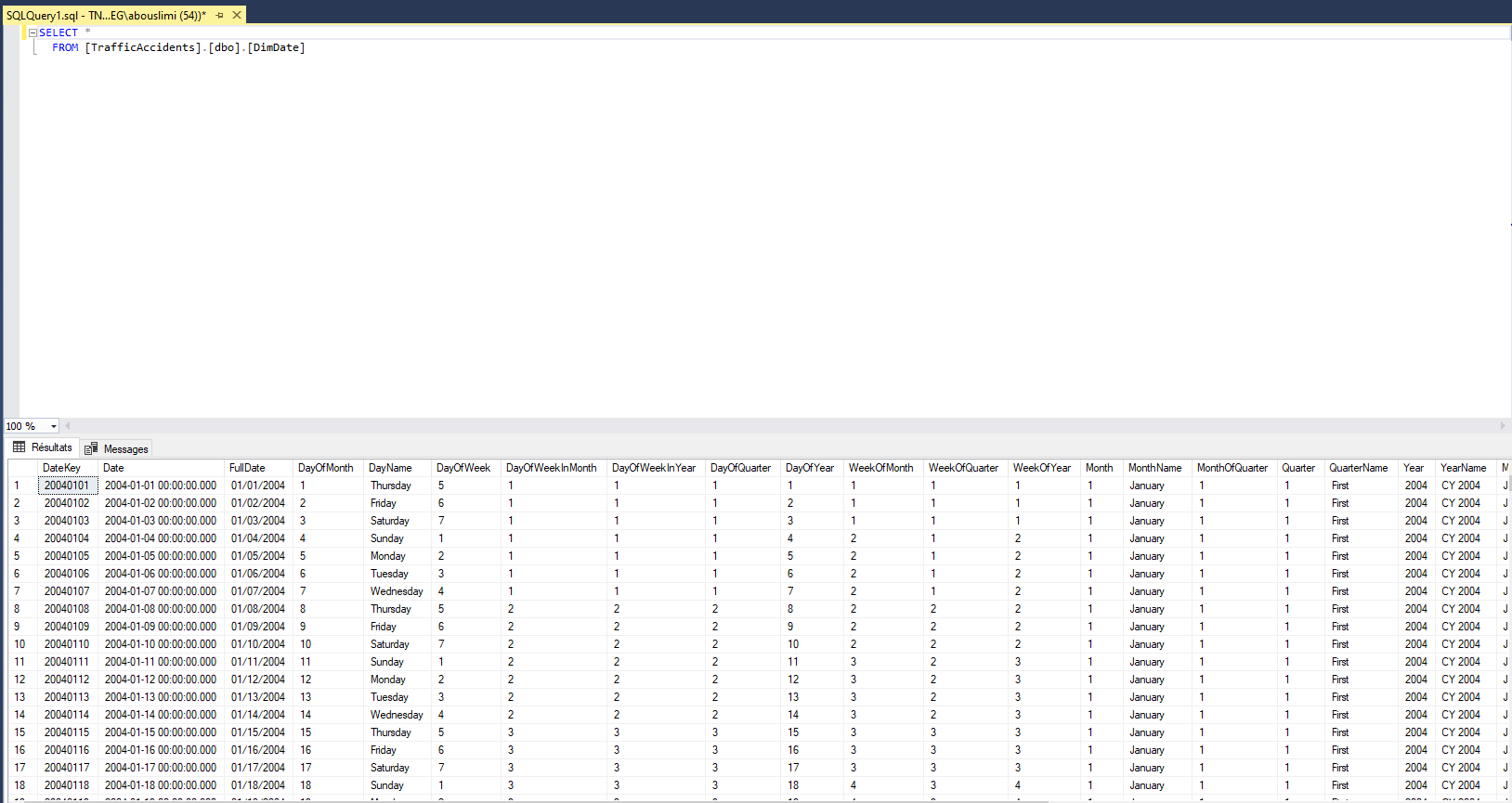
1 Non renseigné / 1 Seul / 2 Accompagné / 3 En groupe

# Traitement des données (Data preprocessing)

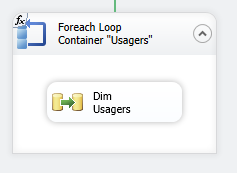
La préparation des données, parfois appelée « prétraitement », est l'étape pendant laquelle les données brutes sont nettoyées et structurées en vue de l'étape suivante du traitement des données. Pendant cette phase de préparation, les données brutes sont vérifiées avec soin afin de déceler d'éventuelles erreurs.  
commençant par la création du dimension temps :   
cette dimension a été créé à travers un script SQL



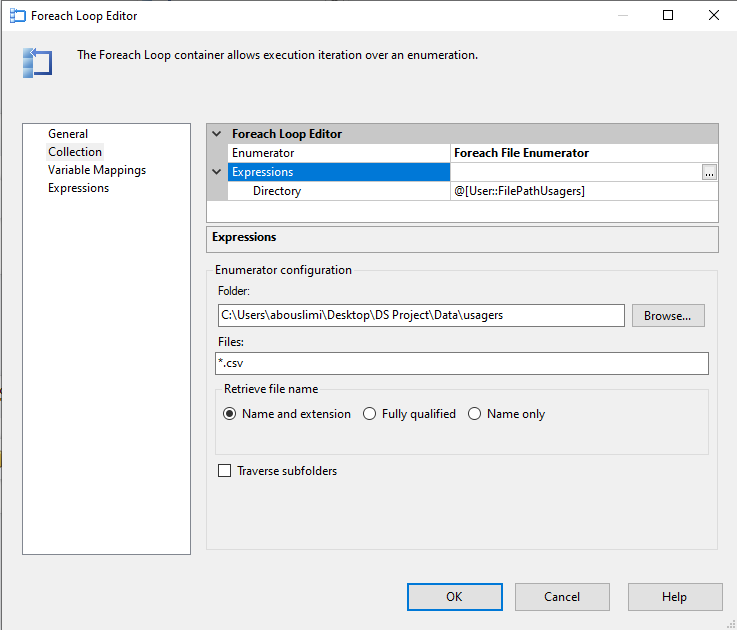


Voici nos données soigneusement intégrées dans notre entrepôt de données.

Dimension usagers :



Puisque nous disposons d'un grand nombre de fichiers à scraper, nous avons décidé d'utiliser le composant 'For Each Loop Container' après avoir configuré la tâche 'Data Flow' pour les dimensions des utilisateurs. Voici le flux de tâches pour la dimension des utilisateurs



A diagram of a data flow

Description automatically generated

Nous avons commencé par utiliser le composant 'Data Flow Task' en insérant notre fichier source. Nous avons choisi l'un des fichiers car nous allons itérer sur les autres fichiers.

A screenshot of a computer

Description automatically generated

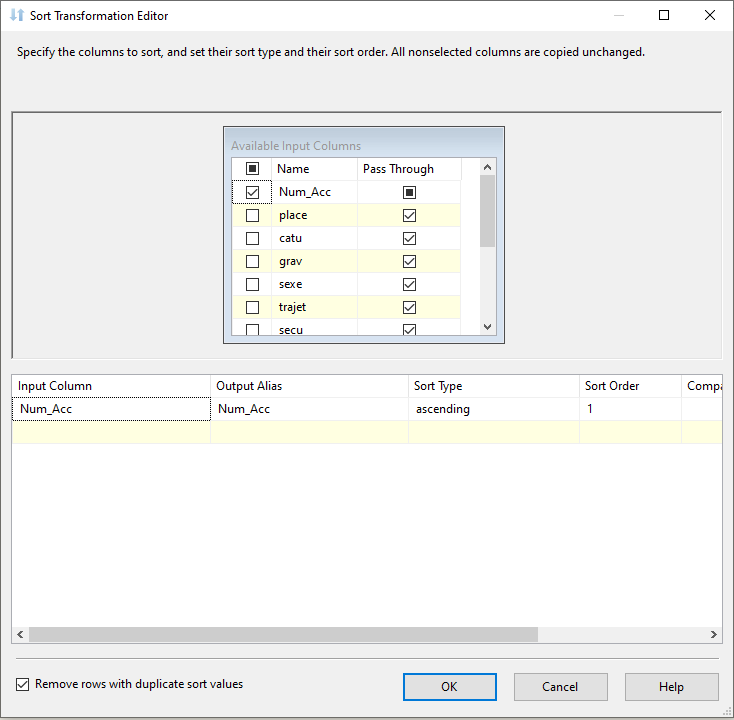
Nous avons établi la connexion entre SSIS et notre fichier source.  
A screenshot of a computer

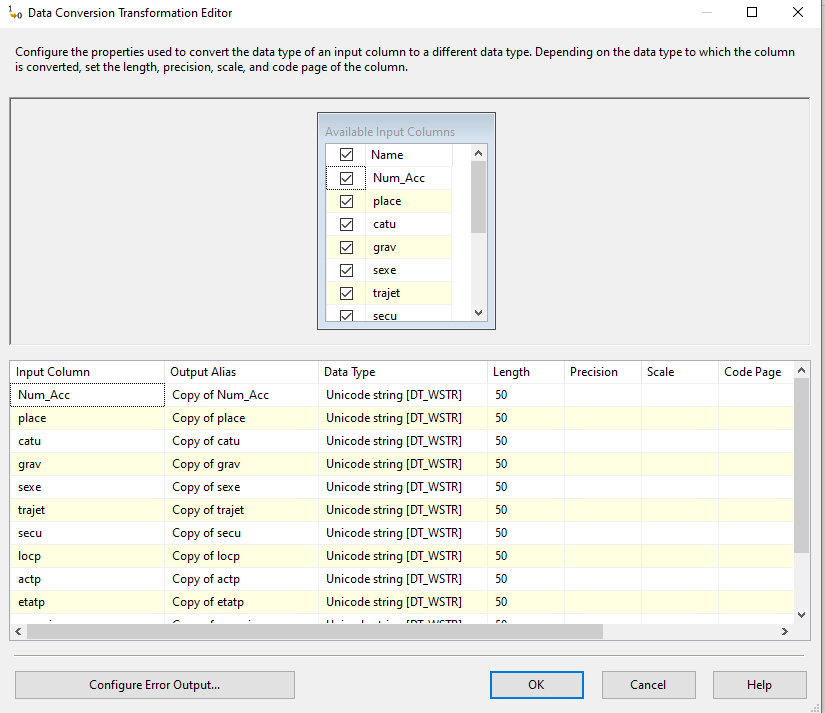
Description automatically generated

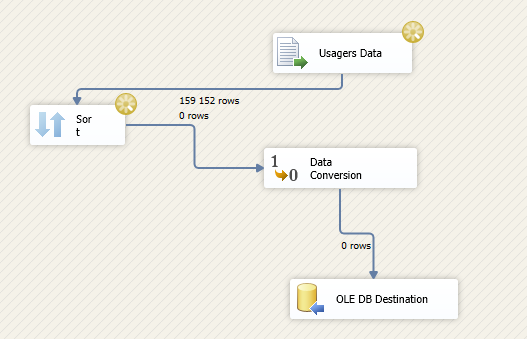
Ensuite, nous avons inséré le composant 'Sort' pour mieux aligner les données et les comprendre tout en supprimant les doublons simultanément.

A screenshot of a computer

Description automatically generated



Ensuite, nous avons utilisé le composant 'Data Conversion' pour convertir les données. Il est essentiel de modifier le type de données pour qu'il corresponde au type utilisé dans SQL Server Management Studio.  


Finalement, nous allons insérer nos données nettoyées dans notre entrepôt de données (data Warehouse).  


A screenshot of a computer

Description automatically generated

Fichier par fichier

A screenshot of a computer screen

Description automatically generated

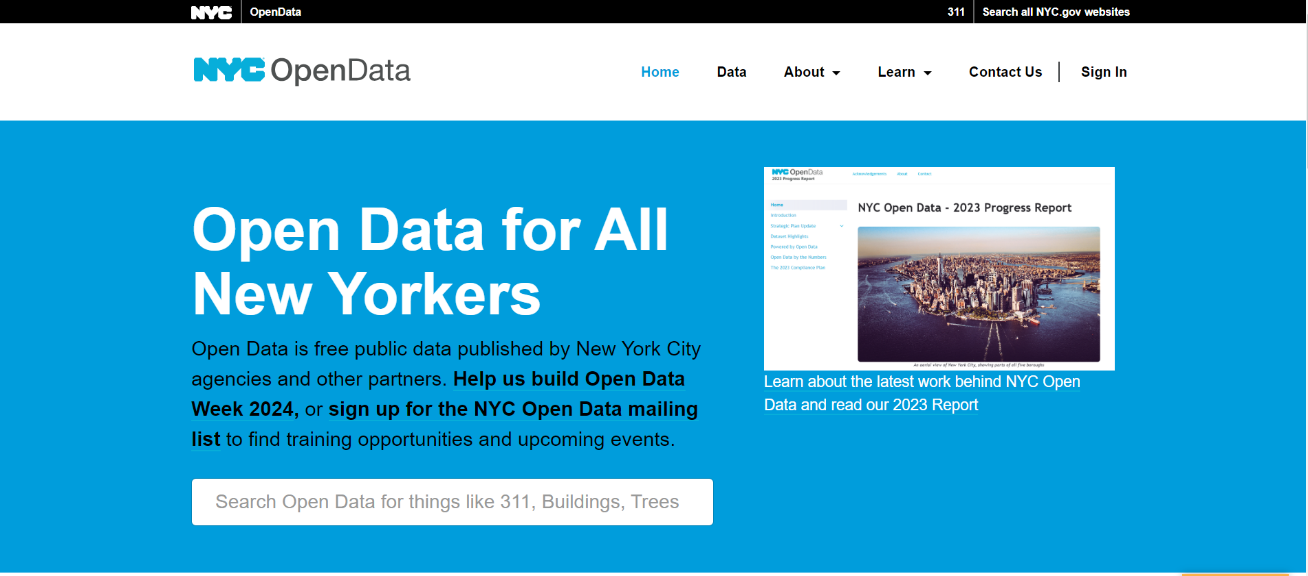
Le même travaille pour les autres dimensions

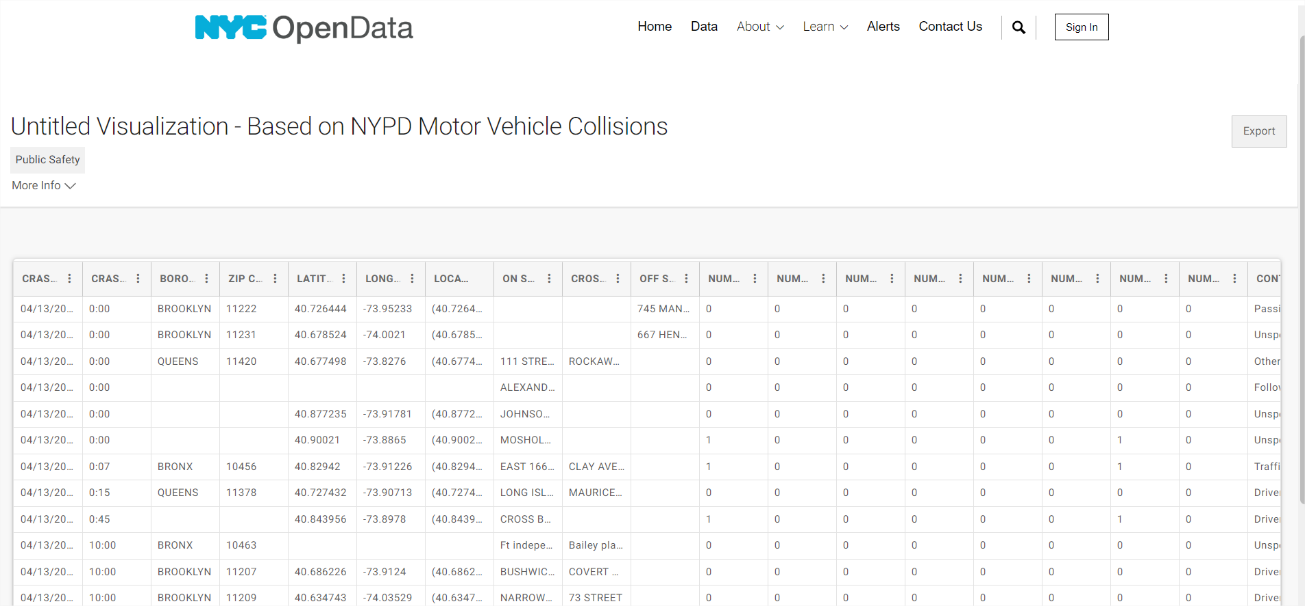
Pour finalement trouver un schéma comme ca  
A screenshot of a computer

Description automatically generated

# Cas d’utilisation et techniques

Cas : <https://data.cityofnewyork.us/Public-Safety/Untitled-Visualization-Based-on-NYPD-Motor-Vehicle/6625-xkgg> par **Maher Omrani**





Description du site :  
Le site https://data.cityofnewyork.us/Public-Safety/Untitled-Visualization-Based-on-NYPD-Motor-Vehicle offre un accès précieux à des données sur les collisions de véhicules à moteur dans la ville de New York. Cette ressource fournit des informations détaillées sur les accidents de la circulation, y compris leur emplacement, la date et l'heure de l'incident, les types de véhicules impliqués, les blessures subies et d'autres facteurs contributifs. Les utilisateurs peuvent explorer et analyser ces données pour mieux comprendre les schémas de collisions, identifier les zones à risque et informer les initiatives de sécurité routière. Cette base de données constitue une ressource essentielle pour les chercheurs, les professionnels de la sécurité publique et les décideurs souhaitant améliorer la sécurité sur les routes de la ville de New York.

### Code source :

Partie de la code source, vous trouverez la suite sur a page annexe.

### 

### Explication du code :

Notre code a pour but d’automatiser le processus d'extraction de fichiers à partir d'une page web spécifique sur le site data.cityofnewyork.us. Voici une explication détaillée de son fonctionnement :

1. **From selenium import webdriver**: Importation du module webdriver de Selenium pour contrôler le navigateur.

2. **From selenium.webdriver.common.by import By**: Importation de la classe By qui permet de rechercher des éléments dans la page web.

3. **From selenium.webdriver.support.ui import WebDriverWait** : Importation de la classe WebDriverWait qui permet d'attendre qu'un certain état de la page soit atteint avant de continuer l'exécution du script.

4. **From selenium.webdriver.support import expected\_conditions as EC** : Importation de la classe expected\_conditions qui fournit des conditions attendues (telles que la visibilité, la présence, etc.) pour le webdriver.

5. **From selenium.common.exceptions import TimeoutException, NoSuchElementException** : Importation des exceptions que Selenium peut rencontrer, telles que le délai d'attente dépassé ou l'élément non trouvé.

6. **Import time** : Importation du module time pour introduire des délais dans le script.

7. **Driver = webdriver.Chrome() :** Initialisation d'une instance du navigateur Chrome.

8. **Driver.get("https://data.cityofnewyork.us/Public-Safety/Untitled-Visualization-Based-on-NYPD-Motor-Vehicle/6625-xkgg") :** Ouverture de l'URL spécifiée dans le navigateur.

9. **Export\_button = WebDriverWait(driver, 10).until(...):** Attend que le bouton d'exportation soit cliquable pendant au plus 10 secondes.

10. **EC.element\_to\_be\_clickable((By.XPATH, '...')):** Utilisation d'une condition pour trouver l'élément exportable par son chemin d'accès XPath.

11. **Export\_button.click():** Clique sur le bouton d'exportation une fois qu'il est disponible.

12. **Download\_button = WebDriverWait(driver, 10).until(...):** Attend que le bouton de téléchargement soit cliquable pendant au plus 10 secondes.

13**. EC.element\_to\_be\_clickable((By.XPATH, '...')):** Utilisation d'une condition pour trouver l'élément téléchargeable par son chemin d'accès XPath.

14. **Download\_button.click():** Clique sur le bouton de téléchargement une fois qu'il est disponible.

### En résumé, ce script automatisé utilise la bibliothèque Selenium pour récupérer des données sur les accidents routiers à partir du site data.cityofnewyork.us. Il navigue à travers les pages, extrait les données requises, et télécharge les fichiers correspondants. Cette opération se répète jusqu'à ce que la page cible spécifiée soit atteinte.

### Résultat d’exécution :

Après avoir exécuté le script de scraping sur le site data.gouv.fr, les résultats sont remarquables. Des fichiers CSV ont été extraits, fournissant des données détaillées sur les accidents corporels de la circulation routière, couvrant la période de 2014 à 2024. Ces fichiers renferment des informations essentielles comme la localisation des accidents, les types d'accidents, les conditions météorologiques, et d'autres paramètres cruciaux pour l'analyse de la sécurité routière à New York.

Fichier scrapper :

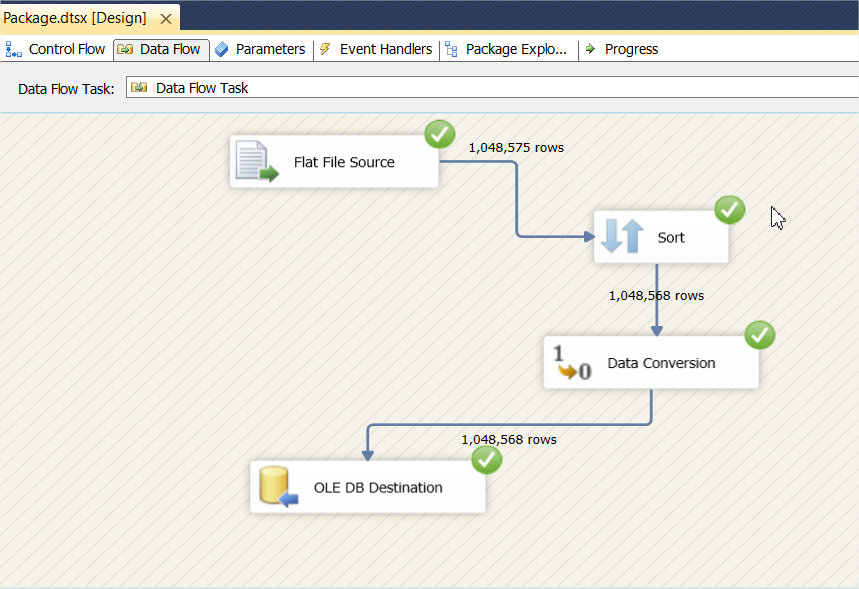


# Compréhension et description des données

|  |  |  |  |
| --- | --- | --- | --- |
| **CRASH DATE** | Occurrence date of collision | Date & Time |  |
| **CRASH TIME** | Occurrence time of collision | Plain Text |  |
| **BOROUGH** | Borough where collision occurred | Plain Text |  |
| **ZIP CODE** | Postal code of incident occurrence | Plain Text |  |
| **LATITUDE** | Latitude coordinate for Global Coordinate System, WGS 1984, decimal degrees (EPSG 4326) | Number |  |
| **LONGITUDE** | Longitude coordinate for Global Coordinate System, WGS 1984, decimal degrees (EPSG 4326) | Number |  |
| **LOCATION** | Latitude , Longitude pair | Location |  |
| **ON STREET NAME** | Street on which the collision occurred | Plain Text |  |
| **CROSS STREET NAME** | Nearest cross street to the collision | Plain Text |  |
| **OFF STREET NAME** | Street address if known | Plain Text |  |
| **NUMBER OF PERSONS INJURED** | Number of persons injured | Number |  |
| **NUMBER OF PERSONS KILLED** | Number of persons killed | Number |  |
| **NUMBER OF PEDESTRIANS INJURED** | Number of pedestrians injured | Number |  |
| **NUMBER OF PEDESTRIANS KILLED** | Number of pedestrians killed | Number |  |
| **NUMBER OF CYCLIST INJURED** | Number of cyclists injured | Number |  |
| **NUMBER OF CYCLIST KILLED** | Number of cyclists killed | Number |  |
| **NUMBER OF MOTORIST INJURED** | Number of vehicle occupants injured | Number |  |
| **NUMBER OF MOTORIST KILLED** | Number of vehicle occupants killed | Number |  |
| **CONTRIBUTING FACTOR VEHICLE 1** | Factors contributing to the collision for designated vehicle | Plain Text |  |
| **CONTRIBUTING FACTOR VEHICLE 2** | Factors contributing to the collision for designated vehicle | Plain Text |  |
| **CONTRIBUTING FACTOR VEHICLE 3** | Factors contributing to the collision for designated vehicle | Plain Text |  |
| **CONTRIBUTING FACTOR VEHICLE 4** | Factors contributing to the collision for designated vehicle | Plain Text |  |
| **CONTRIBUTING FACTOR VEHICLE 5** | Factors contributing to the collision for designated vehicle | Plain Text |  |
| **COLLISION\_ID** | Unique record code generated by system. Primary Key for Crash table. | Number |  |
| **VEHICLE TYPE CODE 1** | Type of vehicle based on the selected vehicle category (ATV, bicycle, car/suv, ebike, escooter, truck/bus, motorcycle, other) | Plain Text |  |
| **VEHICLE TYPE CODE 2** | Type of vehicle based on the selected vehicle category (ATV, bicycle, car/suv, ebike, escooter, truck/bus, motorcycle, other) | Plain Text |  |
| **VEHICLE TYPE CODE 3** | Type of vehicle based on the selected vehicle category (ATV, bicycle, car/suv, ebike, escooter, truck/bus, motorcycle, other) | Plain Text |  |
| **VEHICLE TYPE CODE 4** | Type of vehicle based on the selected vehicle category (ATV, bicycle, car/suv, ebike, escooter, truck/bus, motorcycle, other) | Plain Text |  |
| **VEHICLE TYPE CODE 5** | Type of vehicle based on the selected vehicle category (ATV, bicycle, car/suv, ebike, escooter, truck/bus, motorcycle, other) | Plain Text |  |

# Traitement des données (Data preprocessing)

La préparation des données, parfois appelée « prétraitement », est l'étape pendant laquelle les données brutes sont nettoyées et structurées en vue de l'étape suivante du traitement des données. Pendant cette phase de préparation, les données brutes sont vérifiées avec soin afin de déceler d'éventuelles erreurs.



1. **Flat File Source :**

C'est une tâche dans SSIS qui permet de lire des données à partir d'un fichier plat, tel qu'un fichier texte ou un fichier CSV. Vous configurez cette tâche en spécifiant le chemin du fichier, le type de délimiteur (s'il s'agit d'un fichier CSV), les colonnes, etc.

1. **Sort :**

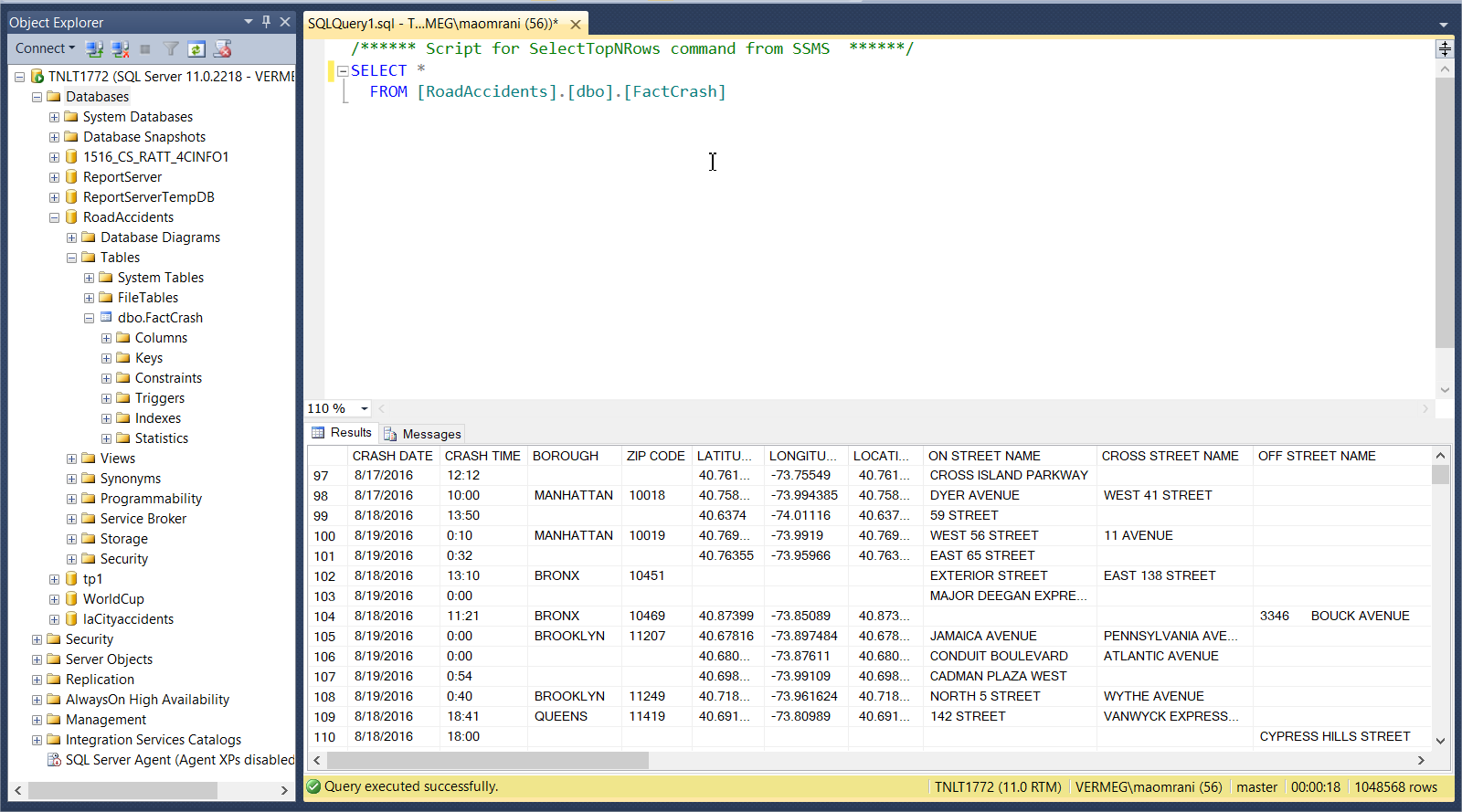
C'est une transformation dans SSIS qui trie les données en fonction d'une ou plusieurs colonnes spécifiées. Cela peut être utile lorsqu'il est nécessaire de garantir un ordre particulier des lignes de données avant de les traiter davantage.

1. **Data Conversion :**

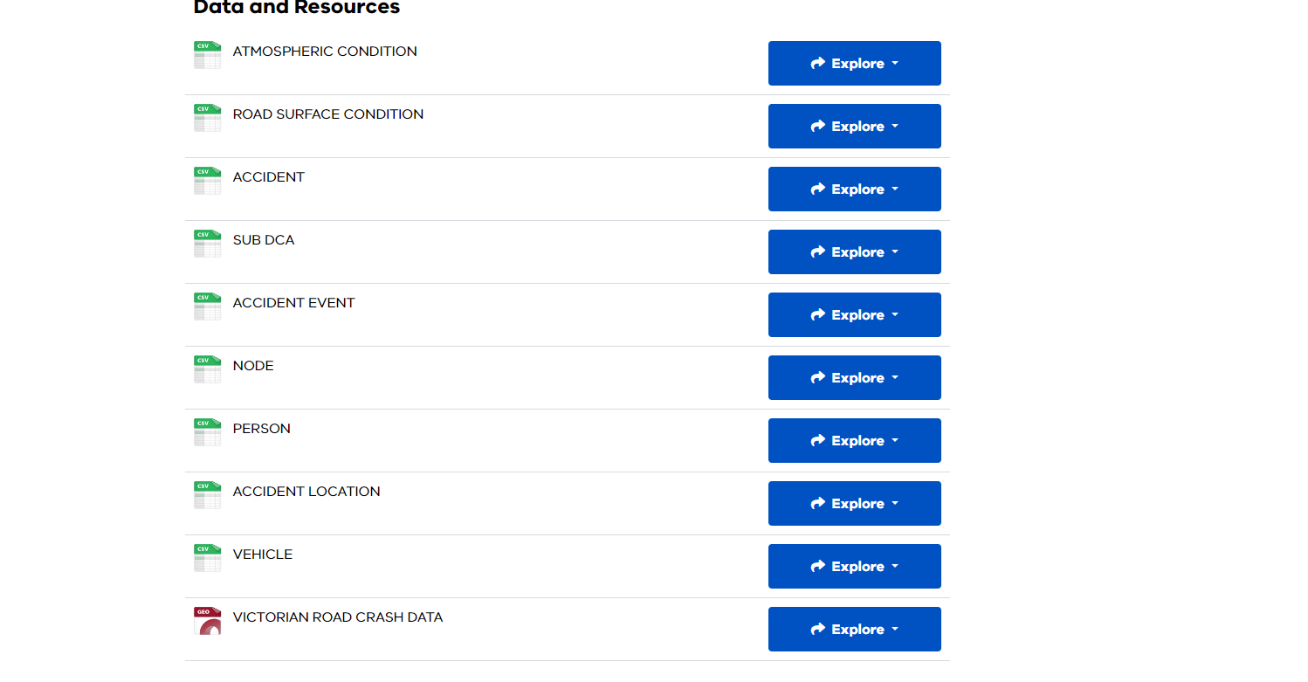
C'est une transformation dans SSIS qui permet de convertir le type de données d'une colonne d'un type à un autre. Par exemple, vous pouvez convertir une chaîne de caractères en entier, une chaîne Unicode en chaîne ANSI, etc. Cela est souvent nécessaire pour assurer la compatibilité des types de données entre les différentes étapes du flux de travail SSIS.

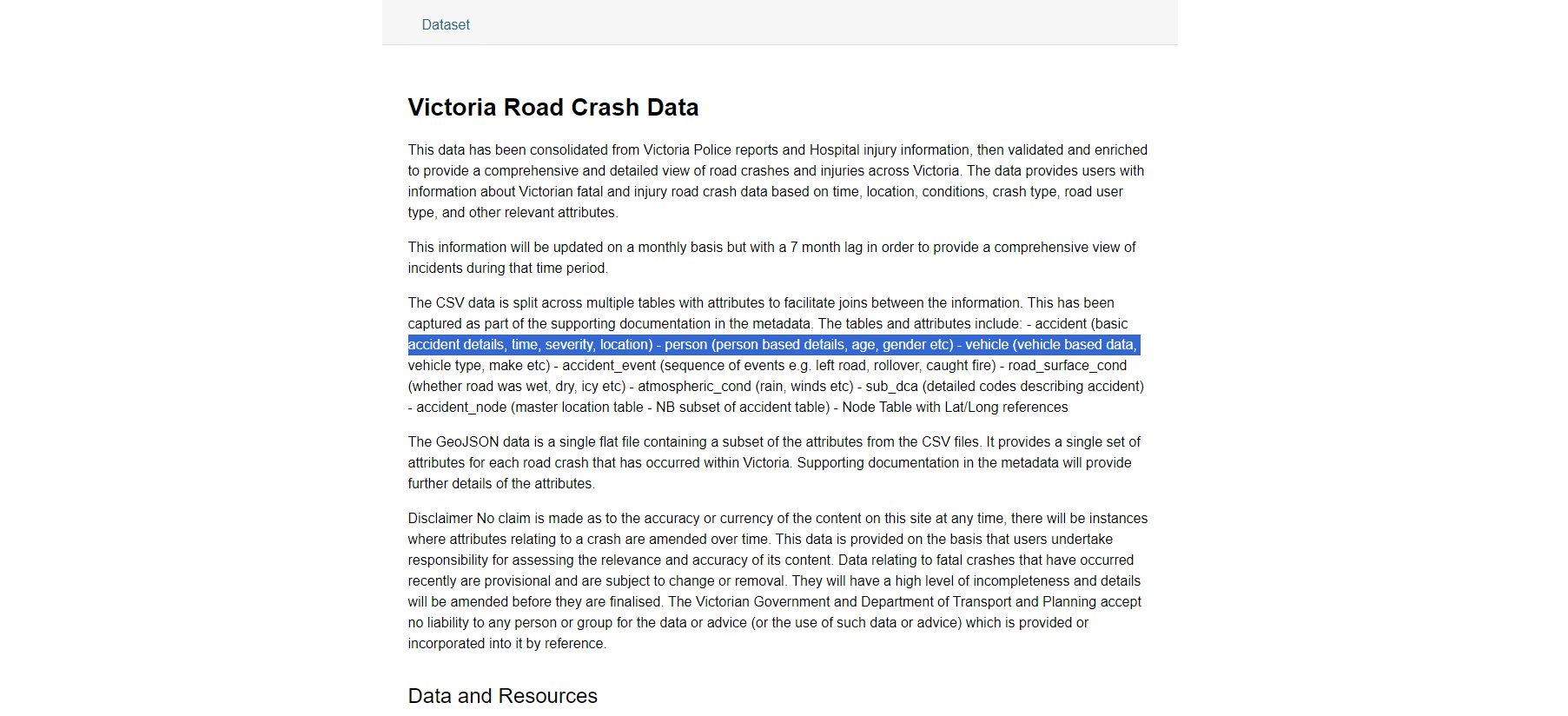
1. **OLE DB Destination :**

C'est une tâche dans SSIS qui permet d'écrire des données dans une destination OLE DB, telle qu'une table dans une base de données SQL Server ou une autre base de données prenant en charge le fournisseur OLE DB. Vous configurez cette tâche en spécifiant la connexion à la base de données, la table de destination, et en mappant les colonnes des données source aux colonnes de la table de destination.



Cas : [https://discover.data.vic.gov.au/dataset/victoria-road-crash-data#](https://discover.data.vic.gov.au/dataset/victoria-road-crash-data) par **Khalil Ahmed ABAZA**





Description du site :  
Ce site web, intitulé "Découvrir les données du Victoria", est une ressource qui fournit des données sur les accidents de la route dans l'État de Victoria, en Australie. Il offre un accès à une base de données contenant des informations détaillées sur les accidents de la route, y compris des données sur les emplacements des accidents, les types de véhicules impliqués, les blessures subies et d'autres facteurs pertinents. Les utilisateurs peuvent explorer ces données pour analyser les tendances des accidents de la route, identifier les zones à risque et élaborer des stratégies de prévention des accidents.

Code source :

Partie de la code source, vous trouverez la suite sur a page annexe.

### Explication du code :

Notre code a pour but d’automatiser le processus d'extraction de fichiers à partir d'une page web spécifique sur le site https://discover.data.vic.gov.au/dataset/victoria-road-crash-data#. Voici une explication détaillée de son fonctionnement :

1. Importation des modules : Les bibliothèques nécessaires sont importées, notamment **os**, **requests**, **BeautifulSoup** et **urljoin** pour manipuler les fichiers, effectuer des requêtes HTTP, et analyser le contenu HTML.

2. Définition de la fonction **download\_csv\_files(url, download\_path)** : Une fonction est créée pour télécharger les fichiers CSV à partir d'une URL donnée et les sauvegarder dans un répertoire spécifié.

3. Requête HTTP : Une requête GET est envoyée à l'URL fournie pour récupérer le contenu de la page Web.

4. Analyse HTML : Le contenu HTML de la réponse est analysé à l'aide de BeautifulSoup pour extraire les liens vers les fichiers CSV.

5. Sélection des liens CSV : Seuls les liens se terminant par '.csv' sont sélectionnés à l'aide de la méthode **find\_all**.

6. Téléchargement des fichiers : Pour chaque lien CSV trouvé, le fichier est téléchargé à l'aide de la bibliothèque **requests**. Le fichier est sauvegardé dans le répertoire de téléchargement spécifié.

6. Téléchargement des fichiers : Pour chaque lien CSV trouvé, le fichier est téléchargé à l'aide de la bibliothèque **requests**. Le fichier est sauvegardé dans le répertoire de téléchargement spécifié.

En résumé, ce script automatisé utilise Selenium pour extraire des données sur les accidents routiers du site data.gouv.fr. Il parcourt les pages, extrait les informations nécessaires, puis télécharge les fichiers associés. Ce processus est répété jusqu'à ce que la page cible spécifiée soit atteinte.

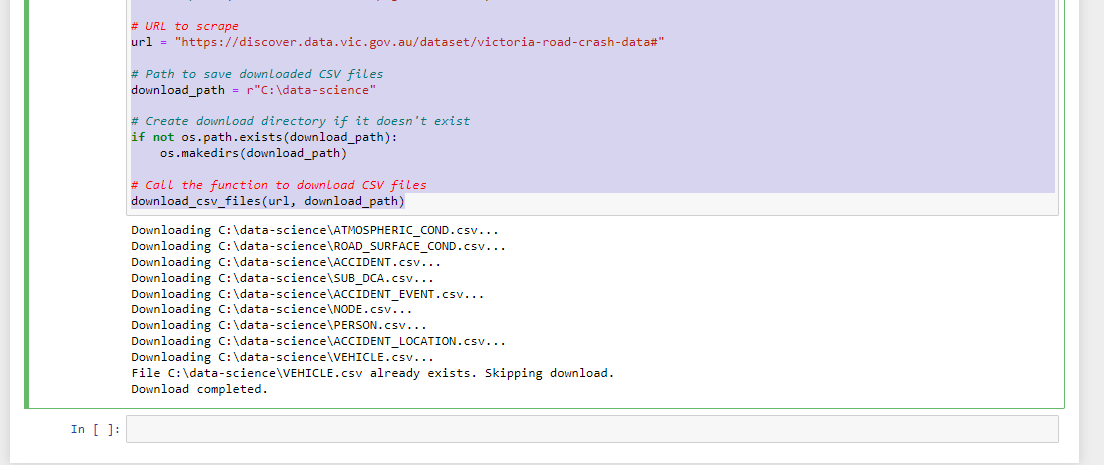
### Résultat d’exécution :

Suite à l'exécution du script de scraping sur le site [https://discover.data.vic.gov.au/dataset/victoria-road-crash-data#](https://discover.data.vic.gov.au/dataset/victoria-road-crash-data) , les résultats obtenus sont significatifs. Au total, 9 fichiers CSV ont été extraits, fournissant des données détaillées sur les accidents corporels de la circulation routière couvrant les années de 2005 à 2022. Ces données ont été consolidées à partir de rapports de la police de Victoria et d'informations sur les blessures hospitalières, puis validées et enrichies pour fournir une vue globale et détaillée des accidents de la route et des blessures à travers Victoria. Les données fournissent aux utilisateurs des informations sur les accidents mortels et les blessures sur les routes victoriennes, basées sur le temps, l'emplacement, les conditions, le type d'accident, le type d'usager de la route et d'autres attributs pertinents.

Ces informations seront mises à jour sur une base mensuelle, mais avec un décalage de 7 mois afin de fournir une vue globale des incidents au cours de cette période.

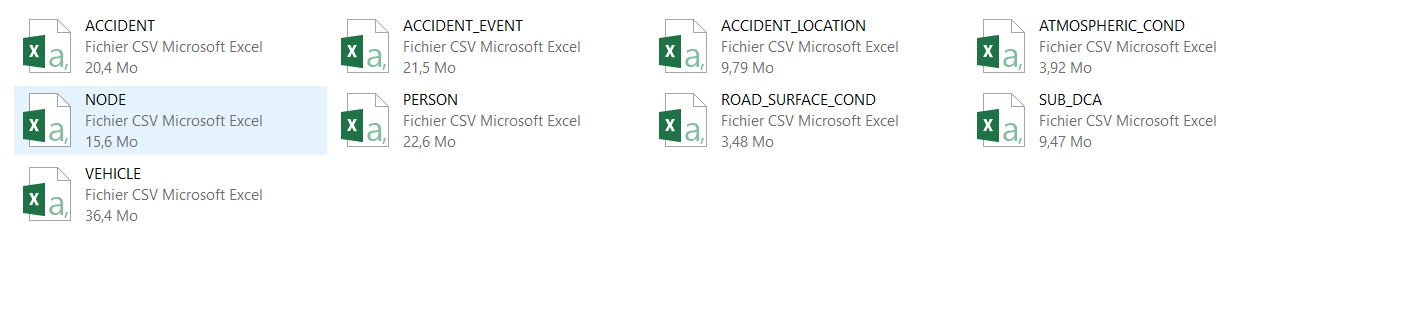
Les données CSV sont réparties dans plusieurs tables avec des attributs pour faciliter les jointures entre les informations.

Voilà un exemple de Log :



Vous trouverez le reste du journal (log) sur la page annexe fournie avec son code.

Fichiers scrapper :



# Compréhension et description des données

### Définitions du fichier national des données BAAC Bulletins d’Analyse des Accidents Corporels de la circulation

Cela a été capturé dans la documentation d'accompagnement dans les métadonnées. Les tables et attributs comprennent :

* **accident (détails de base sur l'accident, heure, gravité, emplacement)**
* **person (détails basés sur la personne, âge, sexe, etc.)**
* **vehicle (données basées sur le véhicule, type de véhicule, marque, etc.)**
* **accident\_event (séquence d'événements, par exemple, sortie de route, tonneau, incendie)**
* **road\_surface\_cond (si la route était mouillée, sèche, verglacée, etc.)**
* **atmospheric\_cond (pluie, vents, etc.)**
* **sub\_dca (codes détaillés décrivant l'accident)**
* **accident\_node (table de localisation principale - NB sous-ensemble de la table d'accident)**
* **Table de nœuds avec des références Lat/Long**

**Accident**

1. **ACCIDENT\_NO**: Identifiant unique pour chaque accident de la route, comme expliqué précédemment.
2. **ACCIDENT\_DATE**: Date à laquelle l'accident s'est produit.
3. **ACCIDENT\_TIME**: Heure à laquelle l'accident s'est produit.
4. **ACCIDENT\_TYPE**: Type d'accident, généralement un code numérique ou une abréviation.
5. **ACCIDENT\_TYPE\_DESC**: Description du type d'accident.
6. **DAY\_OF\_WEEK**: Jour de la semaine où l'accident s'est produit (par exemple, lundi, mardi, etc.).
7. **DAY\_WEEK\_DESC**: Description du jour de la semaine.
8. **DCA\_CODE**: Code de la cause directe de l'accident.
9. **DCA\_DESC**: Description de la cause directe de l'accident.
10. **LIGHT\_CONDITION**: Conditions d'éclairage au moment de l'accident.
11. **NODE\_ID**: Identifiant de nœud, comme expliqué précédemment.
12. **NO\_OF\_VEHICLES**: Nombre de véhicules impliqués dans l'accident.
13. **NO\_PERSONS\_KILLED**: Nombre de personnes tuées dans l'accident.
14. **NO\_PERSONS\_INJ\_2**: Nombre de personnes gravement blessées dans l'accident.
15. **NO\_PERSONS\_INJ\_3**: Nombre de personnes légèrement blessées dans l'accident.
16. **NO\_PERSONS\_NOT\_INJ**: Nombre de personnes non blessées dans l'accident.
17. **NO\_PERSONS**: Nombre total de personnes impliquées dans l'accident.
18. **POLICE\_ATTEND**: Indicateur de présence de la police sur les lieux de l'accident.
19. **ROAD\_GEOMETRY**: Géométrie de la route au moment de l'accident, généralement sous forme de code numérique ou d'abréviation.
20. **ROAD\_GEOMETRY\_DESC**: Description de la géométrie de la route.
21. **SEVERITY**: Gravité de l'accident.
22. **SPEED\_ZONE**: Zone de vitesse où l'accident s'est produit.

**ACCIDENT\_LOCATION**

1. **ACCIDENT\_NO**: C'est un identifiant unique pour chaque accident de la route. Chaque accident est assigné un numéro d'accident unique pour le distinguer des autres incidents.
2. **NODE\_ID**: C'est un identifiant de nœud qui fait référence à une table de localisation. Il identifie l'emplacement où l'accident s'est produit sur le réseau routier. Le NODE\_ID peut être utilisé pour lier cet enregistrement à des détails de localisation spécifiques dans une table de nœuds.
3. **ROAD\_ROUTE\_1**: Il s'agit du numéro de route principal associé à l'emplacement de l'accident. Cela peut inclure des informations sur l'autoroute, la route principale ou toute autre désignation de route importante.
4. **ROAD\_NAME**: Il s'agit du nom de la route où l'accident s'est produit. Cela indique le nom spécifique de la route sur laquelle l'incident s'est produit.
5. **ROAD\_TYPE**: Ceci décrit le type de route où l'accident s'est produit, par exemple, autoroute, route principale, route secondaire, rue, etc.
6. **ROAD\_NAME\_INT**: Si l'accident s'est produit à un intersection, il s'agit du nom de la route intersectée.
7. **ROAD\_TYPE\_INT**: Ceci décrit le type de route intersectée.
8. **DISTANCE\_LOCATION**: C'est la distance de l'accident depuis un point de référence spécifique sur la route, comme un kilomètreage.
9. **DIRECTION\_LOCATION**: Cela indique la direction de l'accident par rapport au point de référence spécifique, par exemple nord, sud, est, ouest, etc.

**VEHICLE**

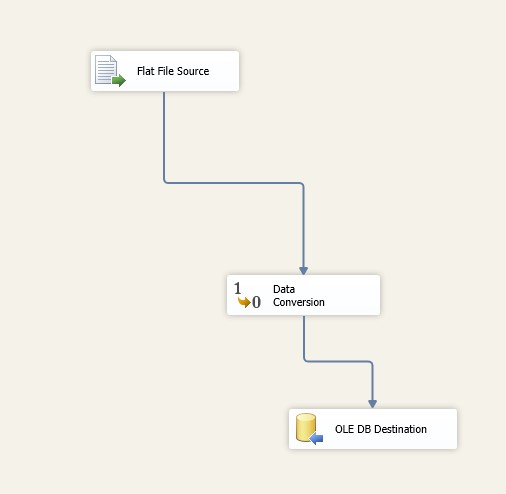
1. **ACCIDENT\_NO**: Identifiant unique pour chaque accident de la route, comme expliqué précédemment.
2. **VEHICLE\_ID**: Identifiant unique pour chaque véhicule impliqué dans l'accident.
3. **VEHICLE\_YEAR\_MANUF**: Année de fabrication du véhicule.
4. **VEHICLE\_DCA\_CODE**: Code de la cause directe de l'accident spécifique au véhicule.
5. **INITIAL\_DIRECTION**: Direction initiale du véhicule avant l'accident.
6. **ROAD\_SURFACE\_TYPE**: Type de surface de la route au moment de l'accident, généralement sous forme de code numérique ou d'abréviation.
7. **ROAD\_SURFACE\_TYPE\_DESC**: Description du type de surface de la route.
8. **REG\_STATE**: État d'immatriculation du véhicule.
9. **VEHICLE\_BODY\_STYLE**: Style de carrosserie du véhicule.
10. **VEHICLE\_MAKE**: Marque du véhicule.
11. **VEHICLE\_MODEL**: Modèle du véhicule.
12. **VEHICLE\_POWER**: Puissance du véhicule.
13. **VEHICLE\_TYPE**: Type de véhicule, généralement sous forme de code numérique ou d'abréviation.
14. **VEHICLE\_TYPE\_DESC**: Description du type de véhicule.
15. **VEHICLE\_WEIGHT**: Poids du véhicule.
16. **CONSTRUCTION\_TYPE**: Type de construction du véhicule.
17. **FUEL\_TYPE**: Type de carburant utilisé par le véhicule.
18. **NO\_OF\_WHEELS**: Nombre de roues du véhicule.
19. **NO\_OF\_CYLINDERS**: Nombre de cylindres du moteur du véhicule.
20. **SEATING\_CAPACITY**: Capacité d'assise du véhicule.
21. **TARE\_WEIGHT**: Poids à vide du véhicule.
22. **TOTAL\_NO\_OCCUPANTS**: Nombre total d'occupants dans le véhicule.
23. **CARRY\_CAPACITY**: Capacité de charge du véhicule.
24. **CUBIC\_CAPACITY**: Capacité cubique du véhicule.
25. **FINAL\_DIRECTION**: Direction finale du véhicule après l'accident.
26. **DRIVER\_INTENT**: Intention du conducteur au moment de l'accident.
27. **VEHICLE\_MOVEMENT**: Mouvement du véhicule au moment de l'accident.
28. **TRAILER\_TYPE**: Type de remorque, le cas échéant.
29. **VEHICLE\_COLOUR\_1**: Couleur principale du véhicule.
30. **VEHICLE\_COLOUR\_2**: Couleur secondaire du véhicule.
31. **CAUGHT\_FIRE**: Indicateur si le véhicule a pris feu pendant l'accident.
32. **INITIAL\_IMPACT**: Impact initial subi par le véhicule.
33. **LAMPS**: État des lampes du véhicule.
34. **LEVEL\_OF\_DAMAGE**: Niveau de dommage subi par le véhicule.
35. **TOWED\_AWAY\_FLAG**: Indicateur si le véhicule a été remorqué après l'accident.
36. **TRAFFIC\_CONTROL**: Contrôle de la circulation au moment de l'accident, généralement sous forme de code numérique ou d'abréviation.
37. **TRAFFIC\_CONTROL\_DESC**: Description du contrôle de la circulation.

**ACCIDENT\_EVENT**

1. **ACCIDENT\_NO**: Identifiant unique pour chaque accident de la route, comme expliqué précédemment.
2. **EVENT\_SEQ\_NO**: Numéro de séquence de l'événement, indiquant l'ordre dans lequel les événements se sont produits pendant l'accident.
3. **EVENT\_TYPE**: Type d'événement, généralement sous forme de code numérique ou d'abréviation.
4. **EVENT\_TYPE\_DESC**: Description du type d'événement.
5. **VEHICLE\_1\_ID**: Identifiant du premier véhicule impliqué dans l'événement.
6. **VEHICLE\_1\_COLL\_PT**: Point de collision du premier véhicule, généralement sous forme de code numérique ou d'abréviation.
7. **VEHICLE 1 COLL PT DESC**: Description du point de collision du premier véhicule.
8. **VEHICLE\_2\_ID**: Identifiant du deuxième véhicule impliqué dans l'événement.
9. **VEHICLE\_2\_COLL\_PT**: Point de collision du deuxième véhicule, généralement sous forme de code numérique ou d'abréviation.
10. **VEHICLE 2 COLL PT DESC**: Description du point de collision du deuxième véhicule.
11. **PERSON\_ID**: Identifiant de la personne impliquée dans l'événement.
12. **OBJECT\_TYPE**: Type d'objet impliqué dans l'événement, généralement sous forme de code numérique ou d'abréviation.
13. **OBJECT\_TYPE\_DESC**: Description du type d'objet impliqué dans l'événement.

# Traitement des données (Data preprocessing)

La préparation des données, parfois appelée « prétraitement », est l'étape pendant laquelle les données brutes sont nettoyées et structurées en vue de l'étape suivante du traitement des données. Pendant cette phase de préparation, les données brutes sont vérifiées avec soin afin de déceler d'éventuelles erreurs.

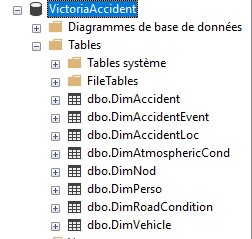


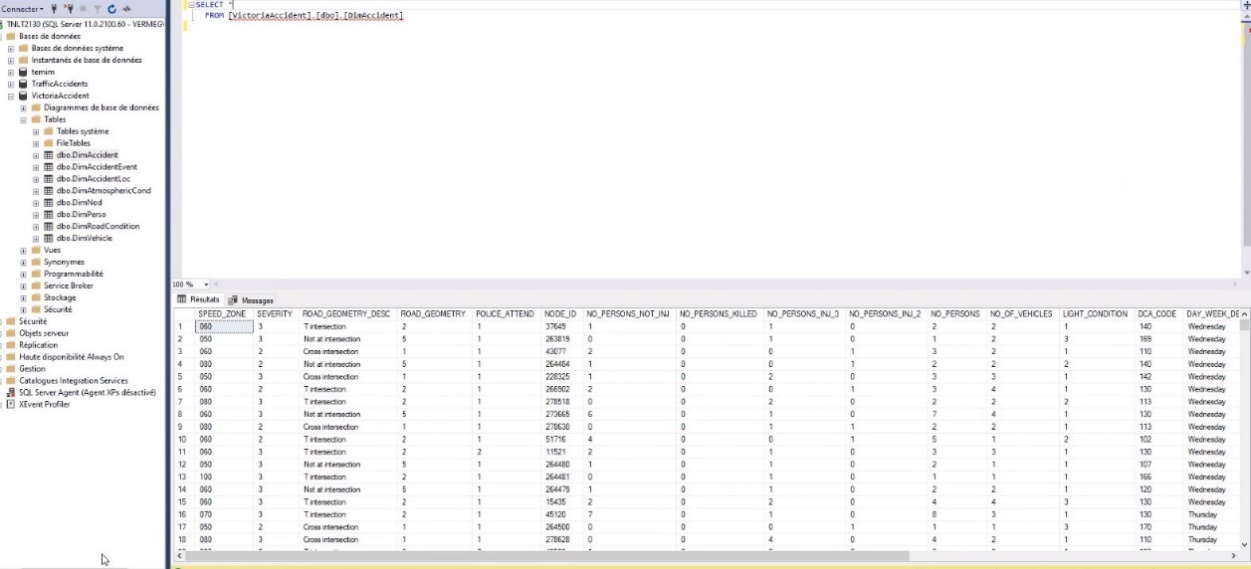
**Un "Flat File Source"** (source de fichier plat) dans SSIS (SQL Server Integration Services) est un composant utilisé pour extraire des données à partir de fichiers plats, tels que des fichiers texte, CSV ou des fichiers avec un format personnalisé. Ce composant permet de spécifier le chemin du fichier à partir duquel les données seront lues, ainsi que le format des données à lire. Il peut également effectuer des opérations de transformation de base sur les données lors de leur extraction, comme le filtrage ou le reformatage.

**Le composant "Data Conversion"** dans SSIS est utilisé pour convertir les données d'un type de données à un autre. Par exemple, il peut être utilisé pour convertir une chaîne de caractères en entier, une date en chaîne de caractères, etc.

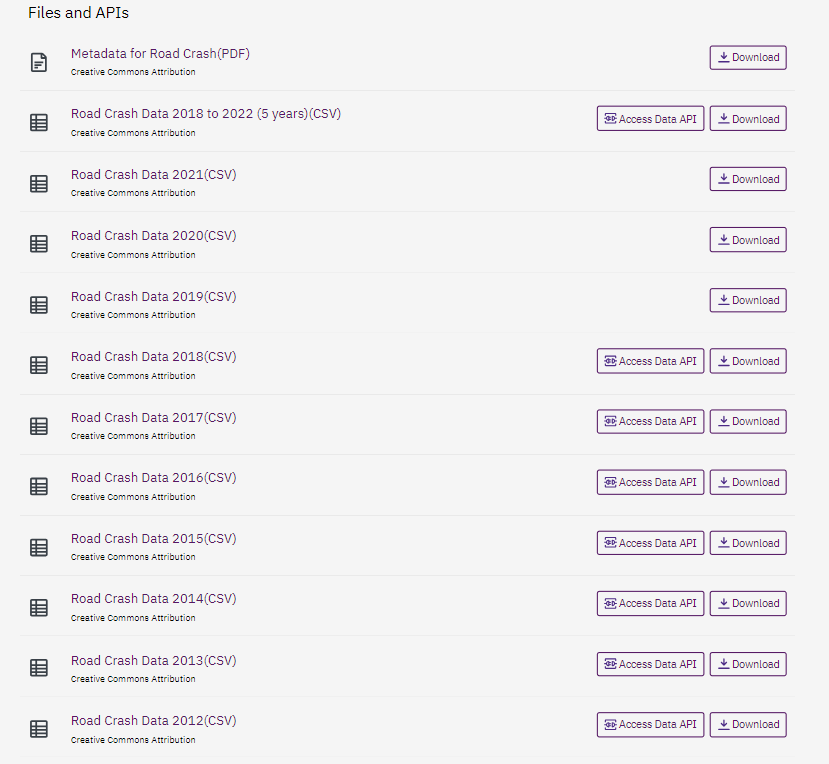
Ce composant est souvent utilisé lorsqu'il y a une incompatibilité entre les types de données dans différentes étapes du flux de données SSIS. Par exemple, si vous extrayez des données d'un fichier plat où toutes les valeurs sont des chaînes de caractères, mais que vous devez les charger dans une base de données où certaines colonnes sont de type entier ou date, vous pouvez utiliser le composant "Data Conversion" pour convertir les valeurs au format approprié avant de les charger dans la base de données.

Le composant "OLE DB Destination" dans SSIS (SQL Server Integration Services) est utilisé pour charger des données à partir d'un flux de données SSIS dans une destination OLE DB, telle qu'une table SQL Server, une vue, ou même une procédure stockée. Ensuite, dans votre package SSIS, vous configurez le composant "OLE DB Destination" pour qu'il pointe vers la table ou l'objet de destination que vous avez créé dans SSMS. Dans notre cas la table VictoriaAccident





Cas : <https://data.gov.au/> par **Temim ABIDI**



Description du site :

[data.gov.au](https://data.gov.au/) est la plateforme centralisée du gouvernement australien qui offre un accès à une large gamme de données publiques et de jeux de données gouvernementaux. Cette plateforme conviviale permet de rechercher et de télécharger des ensembles de données sur divers sujets, tels que l'environnement, l'économie, la santé, l'éducation, etc. Les ensembles de données sont généralement disponibles au format CSV, ce qui facilite leur exploration et leur analyse. En ce qui concerne les statistiques sur la sécurité routière, data.gov.au propose des ensembles de données détaillés comprenant des informations telles que les emplacements des accidents, les dates, les types d'accidents, et d'autres paramètres liés à la sécurité routière. Ces données sont précieuses pour les chercheurs, les décideurs, les professionnels de la sécurité routière et le grand public afin de comprendre les tendances en matière de sécurité routière et d'informer les stratégies visant à améliorer la sécurité sur les routes en Australie.

### Code source :



### Explication du code :

1. **Importation des modules Selenium nécessaires pour l'automatisation des tests** : - `webdriver` : Module principal de Selenium pour contrôler le navigateur. - `By` : Module pour identifier les éléments de la page. - `WebDriverWait` : Classe permettant d'attendre un certain temps jusqu'à ce qu'une condition soit remplie. - `expected\_conditions` : Module contenant des conditions attendues (par exemple, un élément est cliquable). - `TimeoutException`, `NoSuchElementException` : Exceptions pouvant être levées lors de l'utilisation de Selenium.

2. **Initialisation du navigateur Chrome via WebDriver :** - `webdriver.Chrome()` : Crée une instance du navigateur Google Chrome.

3. **Ouverture de l'URL de la page** [**https://data.gov.au/**](https://data.gov.au/) **contenant des données à télécharger**.

4. **Attente du chargement de la page et du rendu du lien "Download" cliquable** : - `WebDriverWait(driver, 10)` : Attend jusqu'à 10 secondes au maximum. - `EC.element\_to\_be\_clickable((By.LINK\_TEXT, "Download"))` : Condition pour vérifier si l'élément avec le texte "Download" est cliquable.

**5. Clic sur le lien "Download"** : - `download\_link.click()` : Clic sur le lien "Download" une fois qu'il est cliquable.

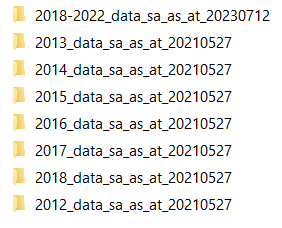
**6. Recherche de tous les éléments** avec la classe "download-button" : - `WebDriverWait(driver, 30)` : Attend jusqu'à 30 secondes au maximum. - `EC.presence\_of\_all\_elements\_located((By.CLASS\_NAME, "download-button"))` : Condition pour vérifier si tous les éléments avec la classe "download-button" sont présents.

**7. Itération à travers chaque bouton de téléchargement et exécution des actions :** - `enumerate(download\_buttons, start=1)` : Itère à travers la liste des boutons de téléchargement en commençant à l'index 1. - `download\_button.click()` : Clic sur le bouton de téléchargement actuel. - `time.sleep(5)` : Pause de 5 secondes pour permettre le téléchargement avant de passer au bouton suivant.

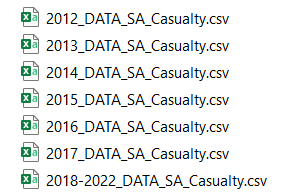
**8. Gestion des exceptions :** - `try-except` : Capture et gère les exceptions éventuellement levées lors du clic sur le bouton de téléchargement. Ce code utilise Selenium pour automatiser le téléchargement de fichiers à partir du site <https://data.gov.au/>. Il attend que la page soit chargée, recherche le lien de téléchargement, clique dessus, puis récupère tous les boutons de téléchargement disponibles sur la page et les clique un par un. En cas d'échec, il imprime un message indiquant le bouton de téléchargement concerné.

### Résultat d’exécution :

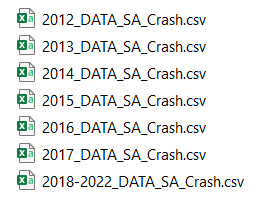
Suite à l'exécution du script de scraping sur le site data.gouv.au, les résultats obtenus sont significatifs. Au total, 21 fichiers CSV ont été extraits, fournissant des données détaillées sur les accidents corporels de la circulation routière couvrant les années de 2012 à 2022. Ces fichiers contiennent des informations cruciales telles que la localisation des accidents, les types d'accidents, les conditions météorologiques, et d'autres paramètres essentiels à l'analyse de la sécurité routière en Australie.



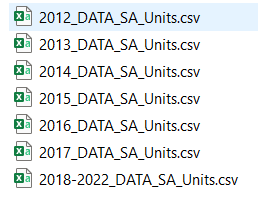
Dossiers Scrappés



Fichiers Casualty

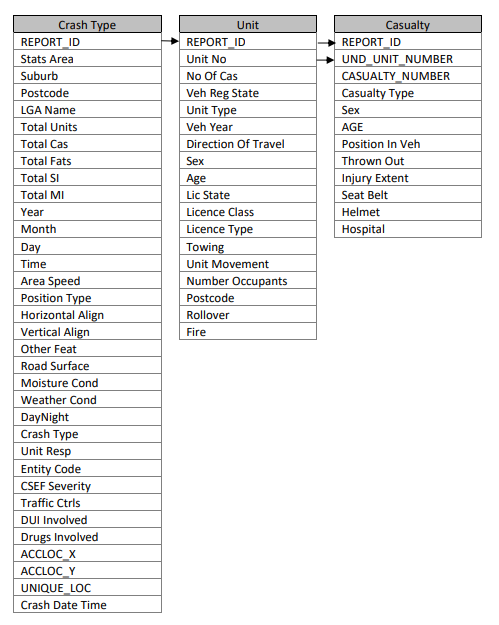


Fichiers Crash



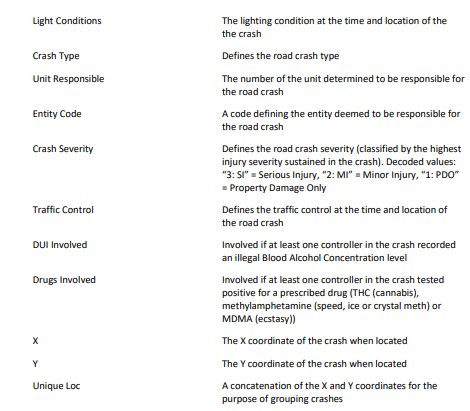
Fichiers Units

# Compréhension et description des données

Colonnes pour chaques fichiers :   


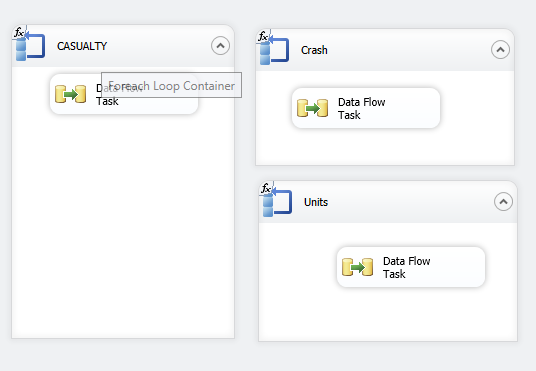
Explication des colonnes du fichier Crash





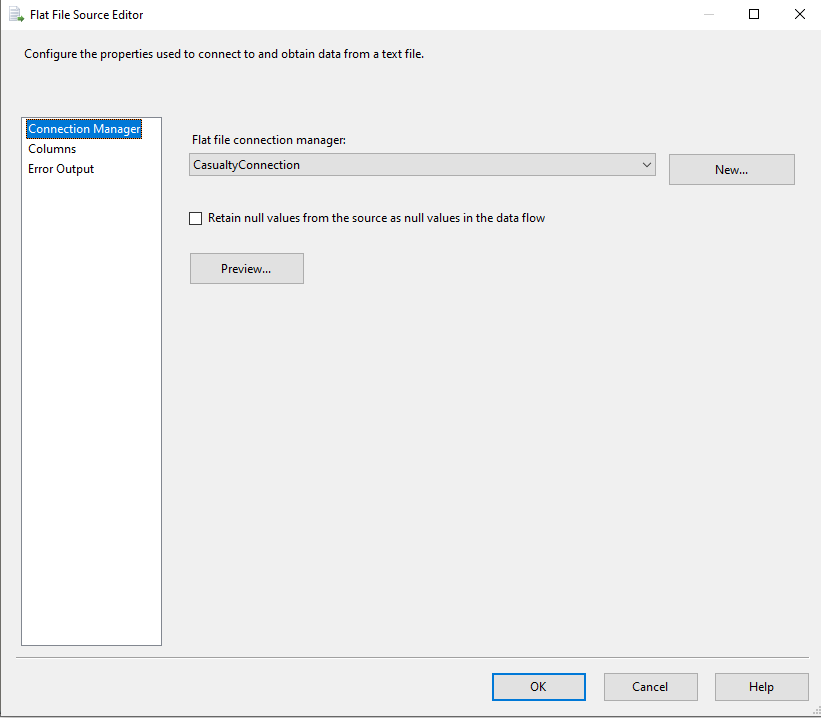
# Traitement des données (Data preprocessing)

La préparation des données, parfois appelée « prétraitement », est l'étape pendant laquelle les données brutes sont nettoyées et structurées en vue de l'étape suivante du traitement des données. Pendant cette phase de préparation, les données brutes sont vérifiées avec soin afin de déceler d'éventuelles erreurs.



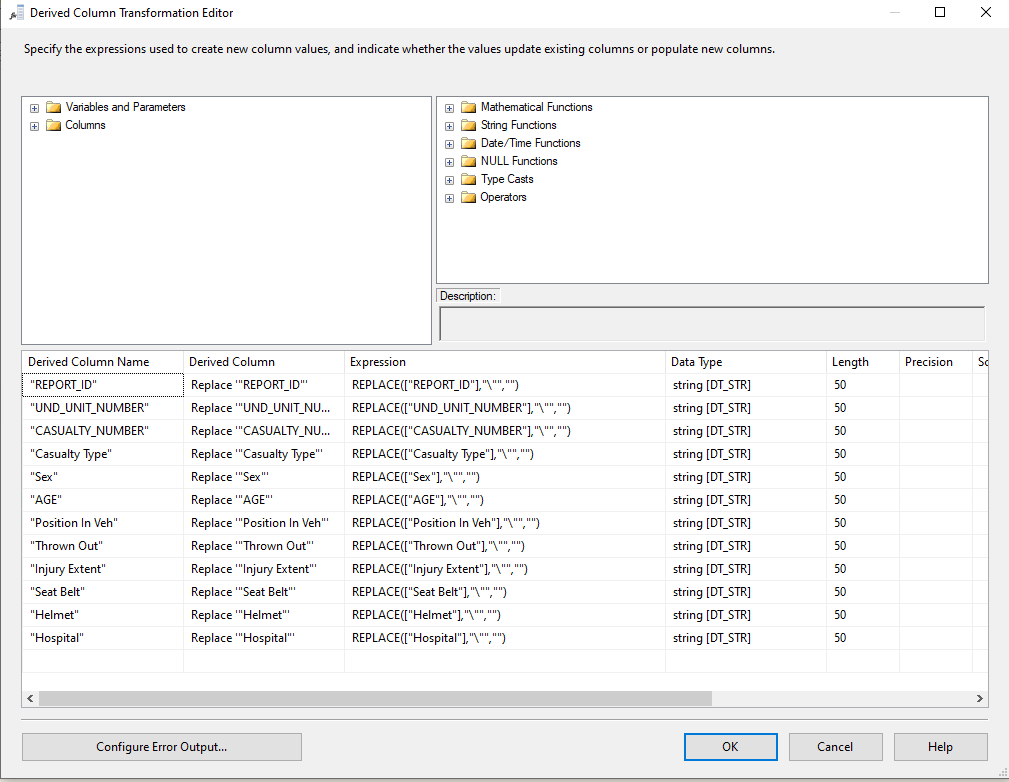
Data Flows des 3 types de fichiers : Casualty, Crash et Units

**Un "Flat File Source"** (source de fichier plat) dans SSIS (SQL Server Integration Services) est un composant utilisé pour extraire des données à partir de fichiers plats, tels que des fichiers texte, CSV ou des fichiers avec un format personnalisé. Ce composant permet de spécifier le chemin du fichier à partir duquel les données seront lues, ainsi que le format des données à lire. Il peut également effectuer des opérations de transformation de base sur les données lors de leur extraction, comme le filtrage ou le reformatage.



Le composant "**Derived Column**" (Colonnes dérivées) est un outil dans les environnements de transformation de données, comme SSIS (SQL Server Integration Services), qui permet de créer de nouvelles colonnes ou de transformer des colonnes existantes à partir des données d'entrée. En utilisant des expressions ou des fonctions, vous pouvez effectuer des opérations arithmétiques, de manipulation de chaînes, de conversion de données, ou d'autres calculs pour dériver de nouvelles valeurs à partir des données sources. C'est un outil essentiel pour effectuer des transformations complexes sur les données dans le processus d'intégration ou de migration de données.

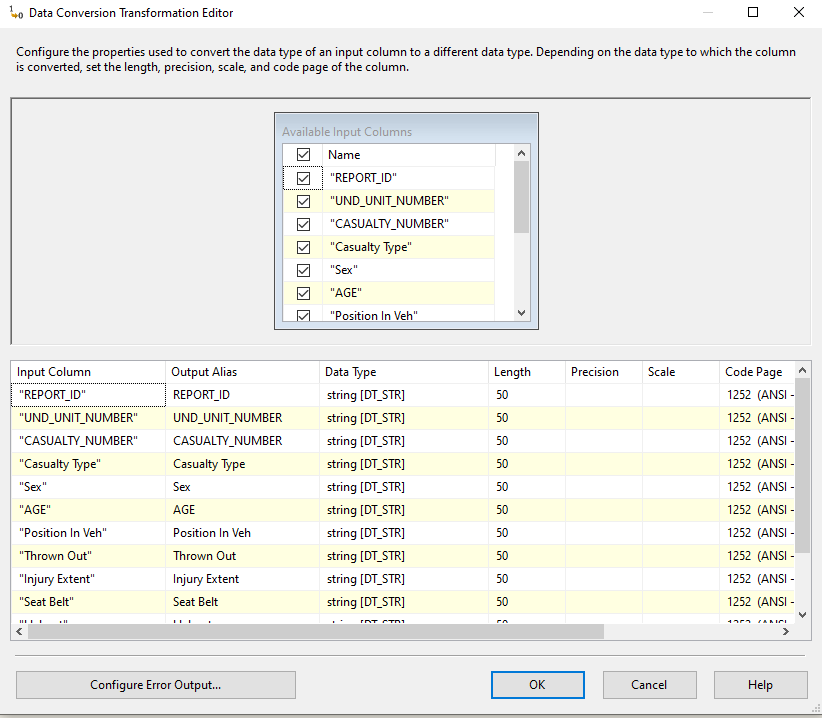
Ce composant est dédié dans ce cas pour enlever les “” avant de insérer les données dans notre base de données.



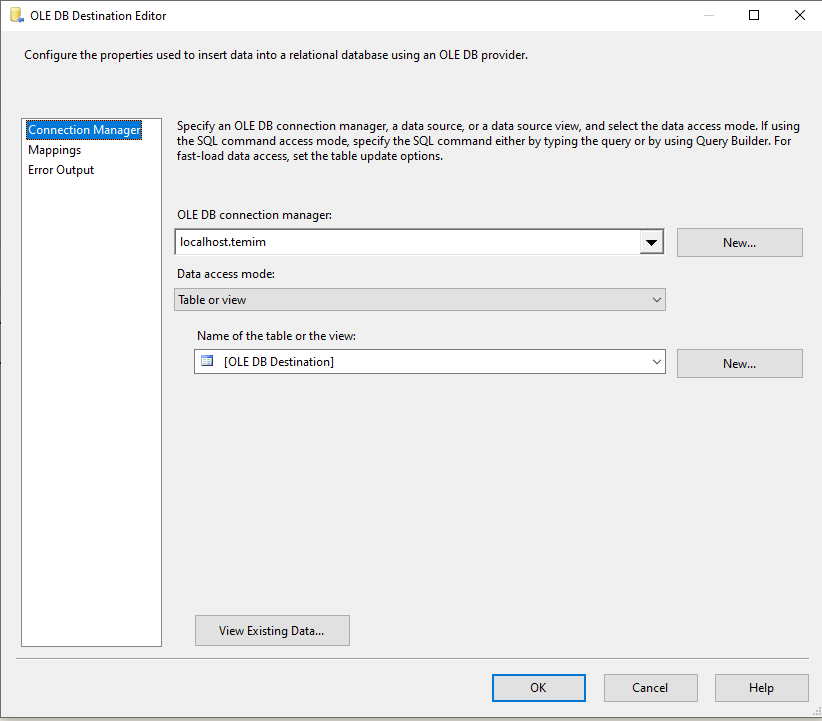
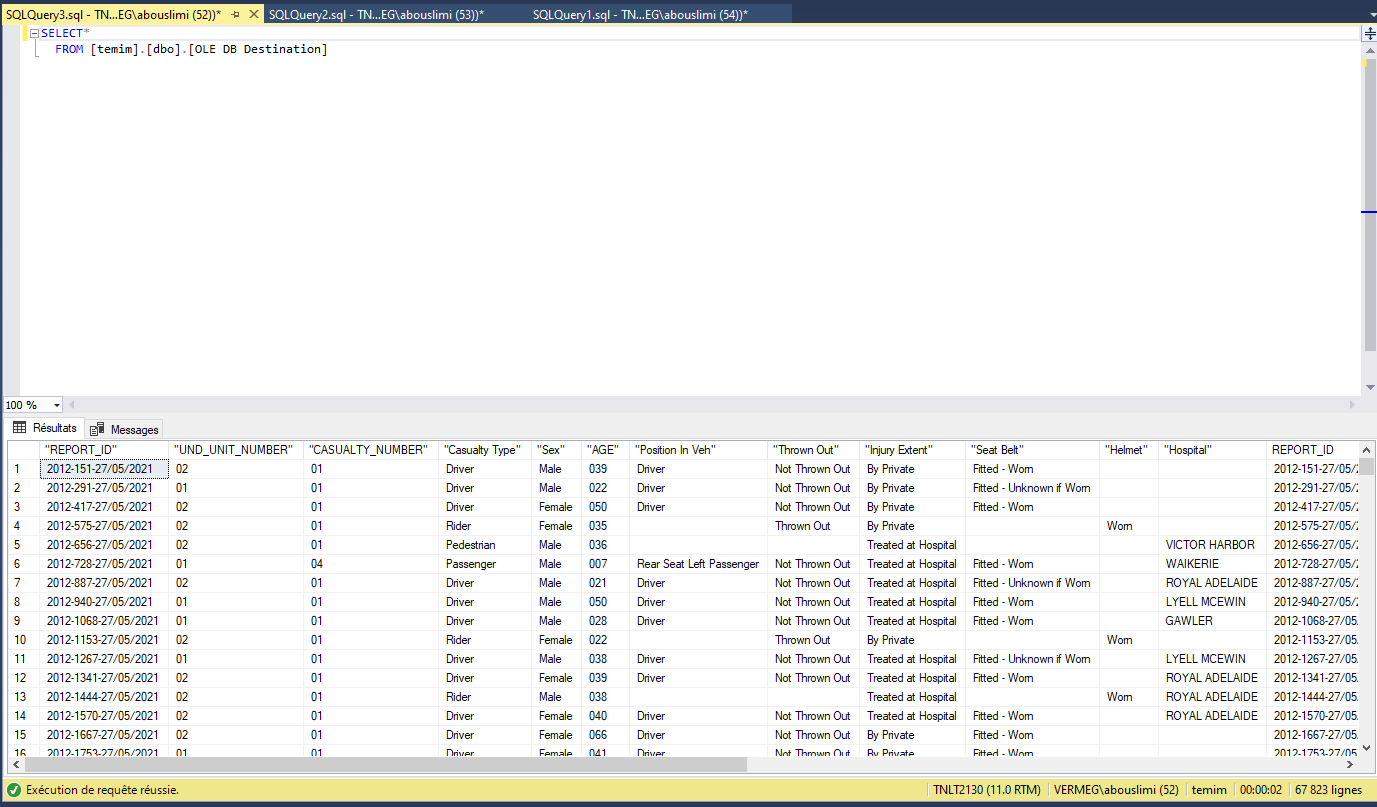
**Le composant "Data Conversion"** dans SSIS est utilisé pour convertir les données d'un type de données à un autre. Par exemple, il peut être utilisé pour convertir une chaîne de caractères en entier, une date en chaîne de caractères, etc.

Ce composant est souvent utilisé lorsqu'il y a une incompatibilité entre les types de données dans différentes étapes du flux de données SSIS. Par exemple, si vous extrayez des données d'un fichier plat où toutes les valeurs sont des chaînes de caractères, mais que vous devez les charger dans une base de données où certaines colonnes sont de type entier ou date, vous pouvez utiliser le composant "Data Conversion" pour convertir les valeurs au format approprié avant de les charger dans la base de données.

On va mapper les colonnes de chacun des fichiers Casualty, Crash et Units :



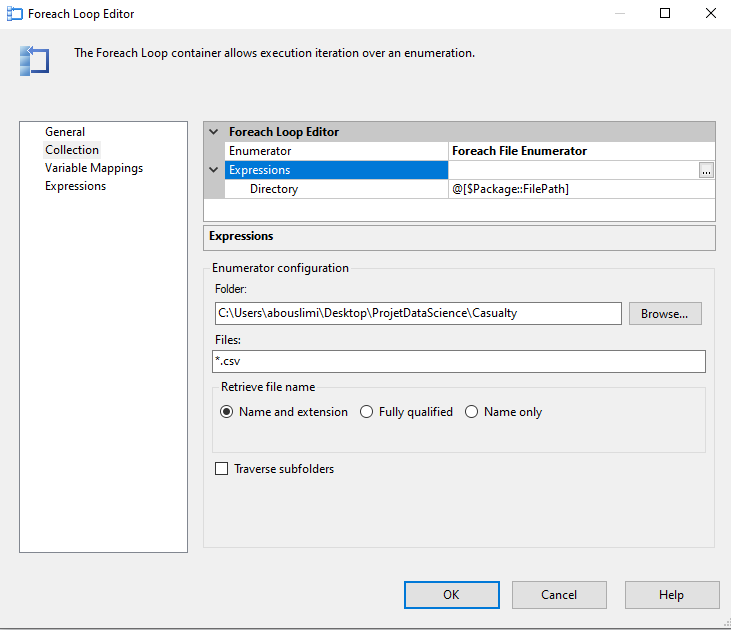
Le composant "**OLE DB Destination**" (Destination OLE DB) est un outil utilisé dans les processus d'intégration de données, tels que SSIS (SQL Server Integration Services), pour charger des données à partir de sources de données externes dans une base de données ou un entrepôt de données. Il permet de définir la connexion à la base de données de destination et de spécifier la table ou la vue cible où les données doivent être chargées. Avec ce composant, vous pouvez mapper les colonnes des données sources aux colonnes de la table cible, gérer les erreurs de chargement et spécifier des options de chargement telles que le mode de chargement (Insert, Update, Delete) et les options de validation. En utilisant le composant "OLE DB Destination", vous pouvez effectuer des opérations de chargement de données de manière efficace et fiable dans votre flux de travail d'intégration de données.



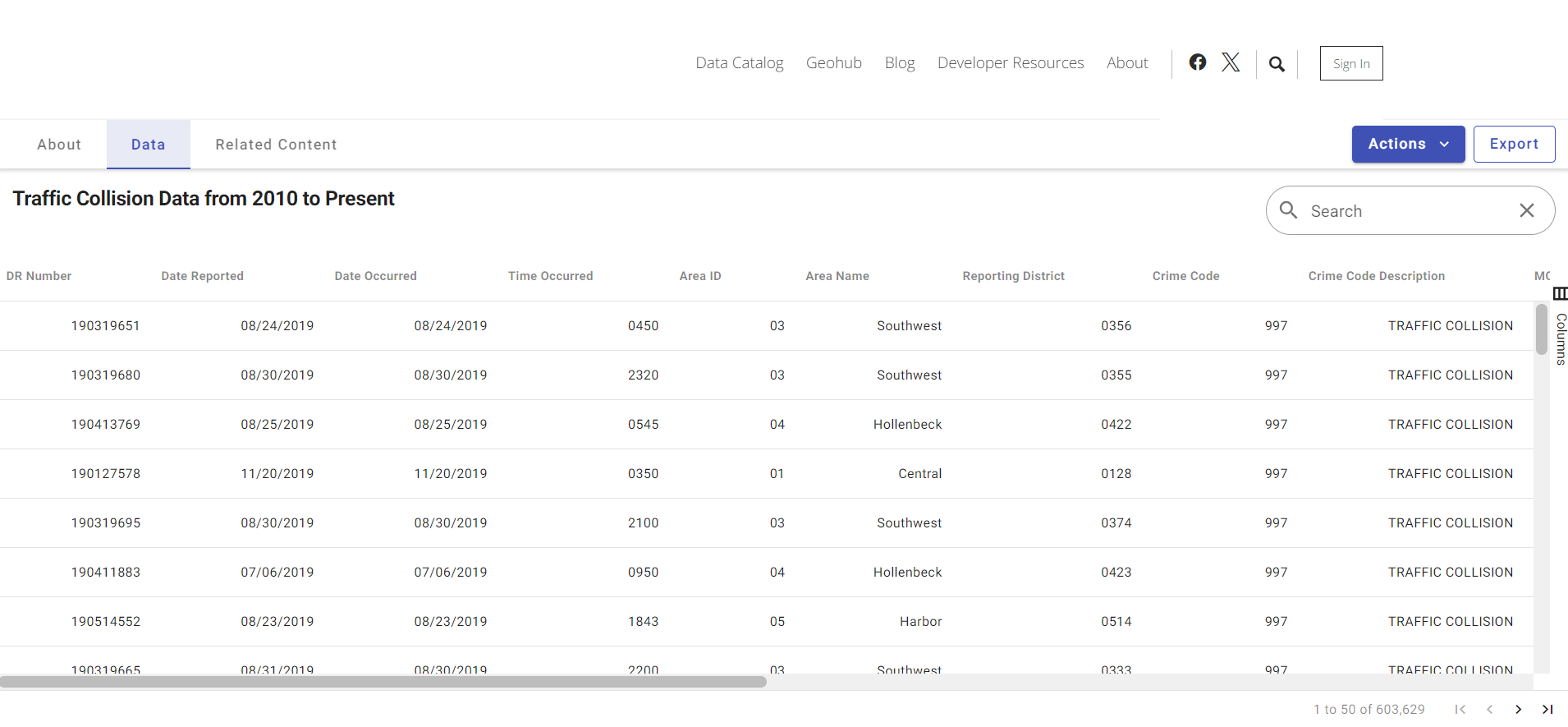
Le composant "**Loop**" (Boucle) est un outil utilisé dans les processus d'intégration de données pour itérer à travers une série d'éléments ou de fichiers afin d'effectuer des opérations répétitives. Dans votre cas, vous avez utilisé ce composant pour parcourir plusieurs fichiers datés de 2012 à 2022.

Dans un contexte de SSIS (SQL Server Integration Services), cela peut être réalisé en utilisant une boucle "Foreach Loop Container" (Conteneur de boucle Pour chaque) qui vous permet de spécifier un ensemble de fichiers ou de répertoires à parcourir. Vous pouvez ensuite définir des opérations à effectuer pour chaque fichier trouvé dans la boucle, comme l'extraction de données, la transformation ou le chargement dans une base de données.

En résumé, en utilisant le composant "Loop" dans votre flux SSIS, vous avez pu automatiser le processus de traitement de multiples fichiers sur une période de temps étendue, facilitant ainsi l'intégration et la gestion de données sur une grande échelle.



Cas : [https://discover.data.vic.gov.au/dataset/victoria-road-crash-data#](https://discover.data.vic.gov.au/dataset/victoria-road-crash-data) par **Houssem Hammami**



Description du site :  
Ce site web, intitulé "Découvrir les données du Victoria", est une ressource qui fournit des données reflète les accidents de la route survenus dans la ville de Los Angeles depuis 2010. Ces données sont transcrites à partir des rapports de circulation originaux qui sont tapés sur papier et il peut donc y avoir quelques inexactitudes dans les données. Certains champs de localisation pour lesquels des données sont manquantes sont notés (0°, 0°). Les champs d'adresse ne sont fournis que jusqu'à la centaine de pâtés de maisons la plus proche afin de préserver la confidentialité. Ces données sont aussi précises que les données contenues dans la base de données. Veuillez faire part de vos questions ou de vos préoccupations dans les commentaires.

Code source :





Partie de la code source, vous trouverez la suite sur a page annexe.

### Explication du code :

Notre code a pour but d’automatiser le processus d'extraction de fichiers à partir d'une page web spécifique sur le <https://data.lacity.org/Public-Safety/Traffic-Collision-Data-from-2010-to-Present/d5tf-ez2w/about_data>

Voici une explication détaillée de son fonctionnement :

* Importation des modules :
  + import os`: Manipulation de fichiers et de répertoires.
  + `import time`: Manipulation du temps.
  + `from selenium import webdriver`: Contrôle des navigateurs web.
  + `from selenium.webdriver.common.by import By`: Localisation des éléments sur une page web.
  + `from selenium.webdriver.support.ui import WebDriverWait`: Attente d'éléments de page web.
  + `from selenium.webdriver.support import expected\_conditions as EC`: Conditions attendues pour WebDriver.
  + `from webdriver\_manager.chrome import ChromeDriverManager`: Gestion automatique du pilote Chrome.
  + `from selenium.webdriver.chrome.options import Options`: Options de configuration du navigateur Chrome
* Fonction `is\_file\_downloaded(download\_dir, file\_name)`:
  + Vérifie si un fichier donné existe dans un répertoire spécifique.
* Fonction `download\_csv\_with\_selenium()`:
  + Automatise le téléchargement d'un fichier CSV depuis une page web.
  + Configure le répertoire de téléchargement et ouvre un navigateur Chrome.
  + Accède à une URL spécifique.
  + Attend que le bouton d'exportation soit cliquable.
  + Clique sur le bouton d'exportation et télécharge le fichier.
  + Attend que le fichier soit téléchargé.
* Bloc `try`, `except`, `finally` :
  + Exécute le code principal, capture les exceptions, et ferme le navigateur à la fin.
* Condition `if \_\_name\_\_ == "\_\_main\_\_":`
  + Vérifie si le script est exécuté directement.
  + Démarre le processus de téléchargement si c'est le cas.

### Résultat d’exécution :

Suite à l'exécution du script de scraping sur le site <https://data.lacity.org/Public-Safety/Traffic-Collision-Data-from-2010-to-Present/d5tf-ez2w/about_data> , les résultats obtenus sont significatifs. 1 fichier CSV a été extraits, fournit des données détaillées sur les accidents corporels de la circulation routière couvrant les années de 2010 à 2024.

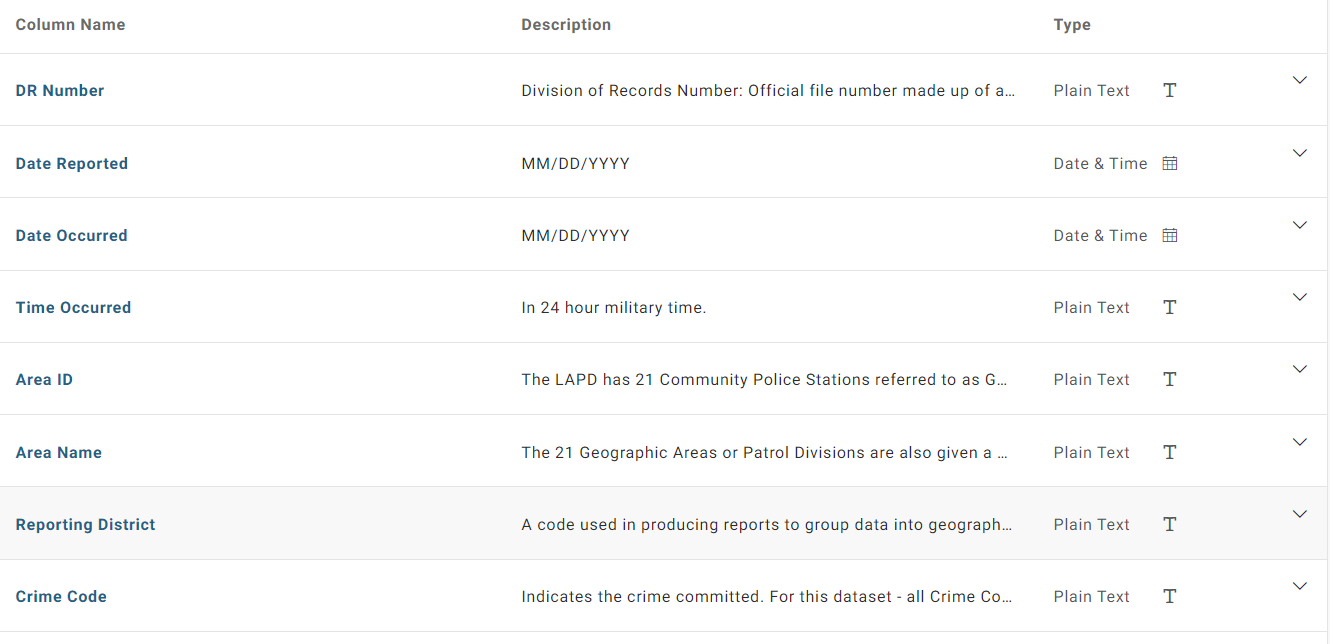
Fichier scrapper :

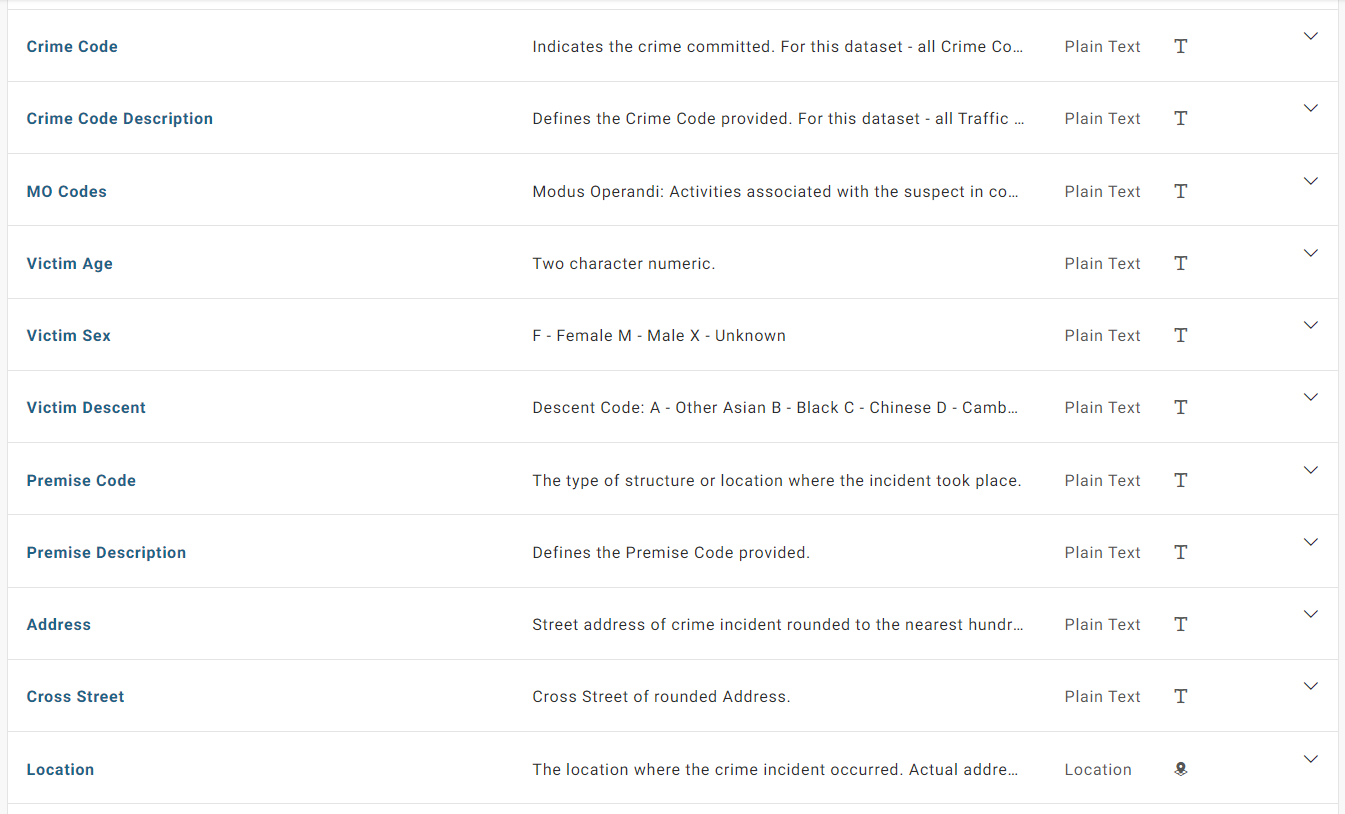


# Compréhension et description des données

### Définitions des colonnes :

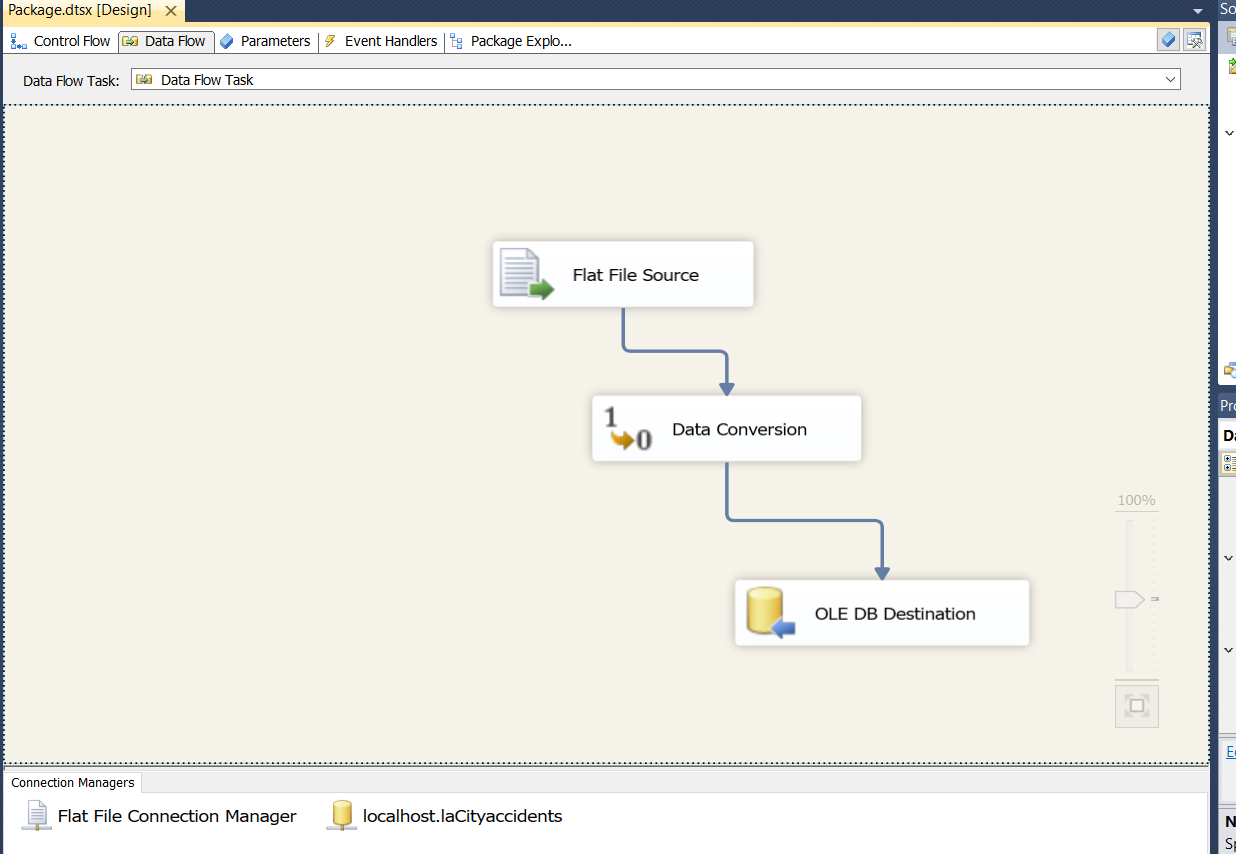
Cela a été capturé dans la documentation d'accompagnement dans les métadonnées. Les tables et attributs comprennent :





# Traitement des données (Data preprocessing)

La préparation des données, parfois appelée « prétraitement », est l'étape pendant laquelle les données brutes sont nettoyées et structurées en vue de l'étape suivante du traitement des données. Pendant cette phase de préparation, les données brutes sont vérifiées avec soin afin de déceler d'éventuelles erreurs.



Le "Flat File Source" dans SSIS est un outil pour extraire des données de fichiers plats comme des fichiers texte ou CSV. Il permet de choisir le fichier à lire et son format. Il peut aussi filtrer ou reformater les données.

Le composant "Data Conversion" dans SSIS convertit les données d'un type à un autre, par exemple de chaînes de caractères à entiers ou de dates à chaînes de caractères. On l'utilise quand les types de données ne correspondent pas entre différentes étapes du flux de données SSIS.

L' "OLE DB Destination" dans SSIS sert à charger des données d'un flux SSIS vers une destination OLE DB comme une table SQL Server. On configure ce composant pour qu'il envoie les données vers la table ou l'objet créé dans SSMS.

