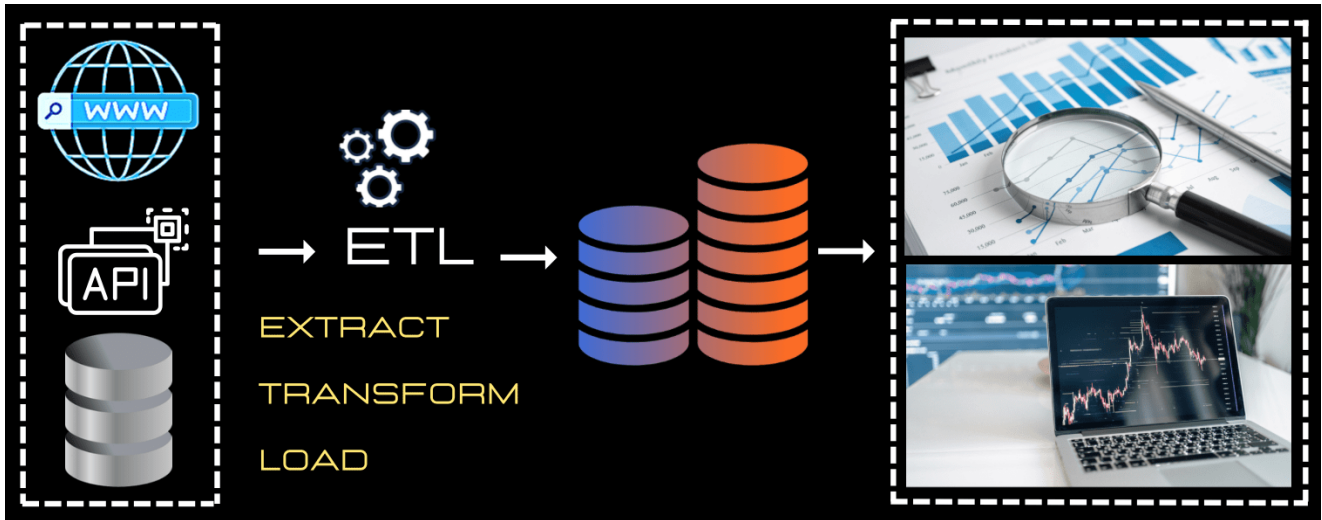


## DIT Master I - IA - 2023 / 2024

---

### Introduction à la mise en place de pipelines ETL (Extract Transform Load)

- Fiche de TP - Création de PIPELINES ETL, prétraitement et visualisation de données



1. Récupérez les 12 fichiers à partir de l'URL suivante:  
[https://github.com/ousmanhamit/sales\\_datasets.git](https://github.com/ousmanhamit/sales_datasets.git)
  2. Création d'un environnement virtuel et installation des dépendances en utilisant les commandes ci-dessous:
    - `python3.12 -m venv .venv && source ~/.venv/bin/activate`
    - `pip install -r requirements.txt`
- 

### TRANSFORMATION DE DONNEES

1. Écrire une fonction qui permet de fusionner les 12 fichiers en un seul document
2. Écrivez une fonction qui supprime tous les caractères non alphabétiques des libellés des variables
3. Identifier et supprimer les données dupliquées

4. Transformer en entier (int) ou en nombre décimal (float) les colonnes numériques actuellement exprimées sous forme de chaînes de caractères.
5. Convertir la colonne 'OrderDate' to datetime
6. Ajoutez les colonnes 'Day', 'Month' et 'Year' à partir de la colonne 'OrderDate', puis insérez-les immédiatement à la suite de celle-ci. Ensuite, convertissez-les en entiers tout en remplaçant respectivement les valeurs manquantes.
7. Identifier et traiter les valeurs manquantes, c'est-à-dire les remplacer ou les supprimer.
8. Identifier et traiter les valeurs aberrantes, c'est-à-dire les remplacer ou les supprimer.
9. Déterminer la distribution moyenne de prix selon chaque catégorie de produit
10. Visualiser le total de ventes par mois
11. Afficher la liste des 10 produits les plus vendus
12. Afficher la liste des 10 produits ayant enregistré le moins de ventes

---

## LOAD DATA

- Se connecter à une base de données Postgres (il faut l'avoir installée au préalable sur sa machine, créer une base de données nommée "mydb" et un utilisateur "myuser" avec ses identifiants de connexion), utilisez des variables d'environnement pour protéger vos informations d'identification, comme illustré ci-dessous

```
load_dotenv(find_dotenv())

connection = psycopg2.connect(
    user = os.getenv("DATABASE_USERNAME"),
    password = os.getenv("DATABASE_PASSWORD"),
    host = os.getenv("DATABASE_HOST"),
    port = os.getenv("DATABASE_PORT"),
    database = os.getenv("DATABASE_NAME")
)

cursor = connection.cursor()
```

The image shows a code editor with Python code for connecting to a PostgreSQL database. Annotations include:

- A yellow box pointing to `load_dotenv(find_dotenv())` with the text "identifiants de connexion retrouvés grace aux variables d'environnements".
- A red box around the connection parameters with a blue arrow pointing to it from the text "protection de données sensibles".
- A red box around the same parameters with a red arrow pointing to it from the text "informations cachées dans .env".
- A red box around `cursor = connection.cursor()` with a red arrow pointing to it from the text "Permet de parcourir les résultats d'une requête SQL".

Python

13. Écrire une fonction qui permet de charger (load) les données que vous venez de nettoyer à la base de données mydb dans une table nommée "dit\_etl\_pipeline"