

# Avancement SAE

## S2.01 Conception et implémentation d'une base de données

Ibrahim BENKHERFELLAH   Axel COULET

Université Sorbonne Paris Nord  
BUT1 SD Semestre 2

May 21, 2025

# Table des matières

- 1 Introduction
- 2 Exploration et compréhension des données
- 3 Modélisation de la base de données
- 4 Script de création et peuplement SQL
- 5 Interrogation et visualisation des données
- 6 Conclusion

# Objectif du projet

- Comprendre et modéliser le commerce des technologies à faible émission de carbone.
- Concevoir une base de données normalisée à partir de deux jeux de données CSV.
- Interroger et visualiser les données pour produire des analyses pertinentes.

# Objectif du projet

- Comprendre et modéliser le commerce des technologies à faible émission de carbone.
- Concevoir une base de données normalisée à partir de deux jeux de données CSV.
- Interroger et visualiser les données pour produire des analyses pertinentes.

# Objectif du projet

- Comprendre et modéliser le commerce des technologies à faible émission de carbone.
- Concevoir une base de données normalisée à partir de deux jeux de données CSV.
- Interroger et visualiser les données pour produire des analyses pertinentes.

- Fichiers CSV étudiés :
  - `Trade_in_Low_Carbon_Technology_Products.csv`
  - `Bilateral_Trade_in_Low_Carbon_Technology_Products.csv`
- Exploration initiale avec Python (pandas) :  
`.head()`, `.info()`, etc.
- Identification de colonnes clés : `CTS_Code`, `Indicator`, `Trade_Flow`, etc.
- Difficultés rencontrées : [à compléter]

- Fichiers CSV étudiés :
  - `Trade_in_Low_Carbon_Technology_Products.csv`
  - `Bilateral_Trade_in_Low_Carbon_Technology_Products.csv`
- Exploration initiale avec Python (pandas) :
  - `.head()`, `.info()`, etc.
- Identification de colonnes clés : `CTS_Code`, `Indicator`, `Trade_Flow`, etc.
- Difficultés rencontrées : [à compléter]

- Fichiers CSV étudiés :
  - `Trade_in_Low_Carbon_Technology_Products.csv`
  - `Bilateral_Trade_in_Low_Carbon_Technology_Products.csv`
- Exploration initiale avec Python (pandas) :
  - `.head()`, `.info()`, etc.
- Identification de colonnes clés : `CTS_Code`, `Indicator`, `Trade_Flow`, etc.
- Difficultés rencontrées : [à compléter]



- Fichiers CSV étudiés :
  - `Trade_in_Low_Carbon_Technology_Products.csv`
  - `Bilateral_Trade_in_Low_Carbon_Technology_Products.csv`
- Exploration initiale avec Python (pandas) :
  - `.head()`, `.info()`, etc.
- Identification de colonnes clés : `CTS_Code`, `Indicator`, `Trade_Flow`, etc.
- Difficultés rencontrées : [à compléter]

- Données structurées en colonnes d'années : F1994 à F2023.
- Présence de colonnes de description redondantes (CTS\_Code, CTS\_Name, CTS\_Full\_Descriptor).
- Première étape : lecture manuelle pour repérage des redondances et structures tabulaires.

- Données structurées en colonnes d'années : F1994 à F2023.
- Présence de colonnes de description redondantes (CTS\_Code, CTS\_Name, CTS\_Full\_Descriptor).
- Première étape : lecture manuelle pour repérage des redondances et structures tabulaires.

- Données structurées en colonnes d'années : F1994 à F2023.
- Présence de colonnes de description redondantes (CTS\_Code, CTS\_Name, CTS\_Full\_Descriptor).
- Première étape : lecture manuelle pour repérage des redondances et structures tabulaires.

- Conversion des colonnes FXXXX en format court (année, valeur).
- Normalisation des années (F1994 à F2023) :
  - Chaque colonne FXXXX devient une ligne avec une valeur d'année.
  - Transformation effectuée via `UNION ALL` dans SQL pour correspondre à la logique de `stack()` en Python.
- Problèmes rencontrés :
  - Ambiguïté sur le pays origine.
  - Duplication de lignes si filtrage incorrect.

- Conversion des colonnes FXXXX en format court (année, valeur).
- Normalisation des années (F1994 à F2023) :
  - Chaque colonne FXXXX devient une ligne avec une valeur d'année.
  - Transformation effectuée via `UNION ALL` dans SQL pour correspondre à la logique de `stack()` en Python.
- Problèmes rencontrés :
  - Ambiguïté sur le pays origine.
  - Duplication de lignes si filtrage incorrect.

- Conversion des colonnes FXXXX en format court (année, valeur).
- Normalisation des années (F1994 à F2023) :
  - Chaque colonne FXXXX devient une ligne avec une valeur d'année.
  - Transformation effectuée via `UNION ALL` dans SQL pour correspondre à la logique de `stack()` en Python.
- Problèmes rencontrés :
  - Ambiguïté sur le pays origine.
  - Duplication de lignes si filtrage incorrect.

- Création d'une table `Bilateral_Trade` pour les données bilatérales :
  - Relation réflexive entre deux pays (deux clés étrangères vers `country`).
- Table `Trade` pour les données nationales.
- Lien systématique aux dimensions : pays, indicateur, catégorie, année.



- Création d'une table `Bilateral_Trade` pour les données bilatérales :
  - Relation réflexive entre deux pays (deux clés étrangères vers `country`).
- Table `Trade` pour les données nationales.
- Lien systématique aux dimensions : pays, indicateur, catégorie, année.

- Création d'une table `Bilateral_Trade` pour les données bilatérales :
  - Relation réflexive entre deux pays (deux clés étrangères vers `country`).
- Table `Trade` pour les données nationales.
- Lien systématique aux dimensions : pays, indicateur, catégorie, année.

# Schéma Entité-Association

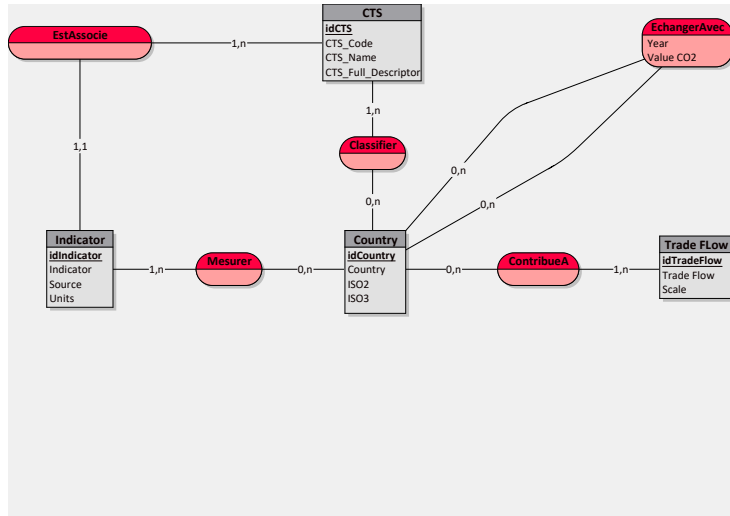


Figure: Schéma Entité-Association Initial

# Schéma Entité-Association

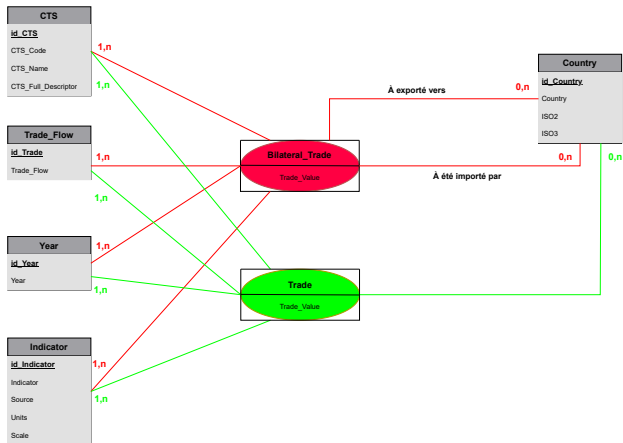


Figure: Schéma Entité-Association Final

- Type Entités : Country, Indicator, CTS, Trade\_Flow, Year
- Type Associations :
  - Bilateral\_Trade (réflexive sur Country)
  - Trade (relation simple avec agrégation)
- Justification des cardinalités et des relations
- Illustration du schéma :

- Type Entités : Country, Indicator, CTS, Trade\_Flow, Year
- Type Associations :
  - Bilateral\_Trade (réflexive sur Country)
  - Trade (relation simple avec agrégation)
- Justification des cardinalités et des relations
- Illustration du schéma :

- Type Entités : Country, Indicator, CTS, Trade\_Flow, Year
- Type Associations :
  - Bilateral\_Trade (réflexive sur Country)
  - Trade (relation simple avec agrégation)
- Justification des cardinalités et des relations
- Illustration du schéma :

- Type Entités : Country, Indicator, CTS, Trade\_Flow, Year
- Type Associations :
  - Bilateral\_Trade (réflexive sur Country)
  - Trade (relation simple avec agrégation)
- Justification des cardinalités et des relations
- Illustration du schéma :



- Schéma relationnel résultant de la modélisation EA :
  - Country(idCountry, Country, ISO2, ISO3)
  - CTS(idCTS, CTS\_Code, CTS\_Name, CTS\_Full\_Descriptor)
  - Trade\_Flow(id\_Trade\_Flow, Trade\_Flow, Scale)
  - Indicator(idIndicator, Indicator\_, Source, Units)
  - Year(id\_Year, Year)
  - Bilateral\_Trade(id\_country, id\_counterpart, id\_indicator, id\_cts, id\_Trade\_Flow, id\_year, trade\_value)
  - Trade(id\_country, id\_indicator, id\_cts, id\_Trade\_Flow, id\_year, trade\_value)

# Forme normale : 1NF

- Toutes les valeurs sont atomiques : aucun champ multivalué ou composé.
- Transformation des colonnes F1994 à F2023 en une colonne `id_year` via pivot SQL.
- Tables relationnelles sans redondance horizontale.
- **Conclusion** : le schéma respecte la **première forme normale (1NF)**.

- Le schéma est en 1NF.
- Aucune dépendance fonctionnelle partielle dans les tables à clés composites.
- Exemple : les descripteurs CTS ont été extraits dans une table distincte, reliée par CTS\_Code.
- **Conclusion** : toutes les dépendances fonctionnelles concernent la clé entière → **2NF validée**.

- Le schéma est en 2NF.
- Suppression des dépendances transitives.
- Exemple : l'unité d'un indicateur dépend uniquement de l'indicateur, pas d'un pays ou autre entité.
- **Conclusion** : toutes les colonnes non-clés dépendent uniquement de la clé primaire → **schéma en 3NF**.

- Toutes les dépendances fonctionnelles ont un antécédent qui est une super-clé.
- Exemple : dans `Bilateral_Trade`, seule la combinaison complète des identifiants détermine `trade_value`.
- Il n'existe pas de dépendance fonctionnelle violant cette condition.
- **Conclusion** : le schéma respecte la forme normale de **Boyce-Codd (BCNF)**.

# Création de la table Country

```
CREATE TABLE Country(  
    id_Country SMALLINT PRIMARY KEY,  
    Country VARCHAR(50),  
    ISO2 CHAR(2),  
    ISO3 CHAR(3)  
);
```

# Création de la table Indicator

```
CREATE TABLE Indicator(  
    id_Indicator SMALLINT PRIMARY KEY,  
    Indicator VARCHAR(80),  
    Source VARCHAR,  
    Units VARCHAR(50),  
    Scale VARCHAR(5)  
);
```

# Création de la table CTS

```
CREATE TABLE CTS(  
    id_CTS SMALLINT PRIMARY KEY,  
    CTS_Code VARCHAR(6),  
    CTS_Name VARCHAR(100),  
    CTS_Full_Descriptor VARCHAR(150)  
);
```



# Création de la table Trade\_Flow

```
CREATE TABLE Trade_Flow(  
    id_Trade_Flow SMALLINT PRIMARY KEY,  
    Trade_Flow VARCHAR(20)  
);
```

# Création de la table Year

```
CREATE TABLE Year(  
    id_Year SMALLINT PRIMARY KEY,  
    Year DATE  
);
```

# Création de la table Echanger\_Avec

```
CREATE TABLE Bilateral\_Trade (  
    id_Country SMALLINT REFERENCES Country(id_Country),  
    id_Counterpart SMALLINT REFERENCES Country(id_Country),  
    id_Indicator SMALLINT REFERENCES Indicator(id_Indicator),  
    id_CTS SMALLINT REFERENCES CTS(id_CTS),  
    id_Trade_Flow SMALLINT REFERENCES Trade_Flow(...),  
    id_Year SMALLINT REFERENCES Year(id_Year),  
    trade_value DOUBLE PRECISION,  
    PRIMARY KEY(id_Country, id_Counterpart_Country,  
                id_Indicator, id_CTS, id_Trade_Flow, id_Year)  
);
```

# Création de la table Donnee\_Pays

```
CREATE TABLE Trade (  
    id_Country SMALLINT REFERENCES Country(id_Country),  
    id_Indicator SMALLINT REFERENCES Indicator(id_Indicator),  
    id_CTS SMALLINT REFERENCES CTS(id_CTS),  
    id_Trade_Flow SMALLINT REFERENCES Trade_Flow(...),  
    id_Year SMALLINT REFERENCES Year(id_Year),  
    trade_value DOUBLE PRECISION,  
    PRIMARY KEY(id_Country, id_Indicator, id_CTS,  
    id_Trade_Flow, id_Year)  
);
```

- Utilisation de COPY, INSERT INTO avec jointures.
- Alignement avec les traitements Python (Avec `.stack()`).
- Cas particuliers traités :
  - Échanges d'un pays avec lui-même
  - Lignes NULL / Trade Flow = Not Applicable
- Difficultés rencontrées :
  - Contraintes techniques : NULL, unicité, jointures complexes.
  - Problèmes de données : fusion sans doublons, nettoyage, volume élevé.

- Utilisation de COPY, INSERT INTO avec jointures.
- Alignement avec les traitements Python (Avec `.stack()`).
- Cas particuliers traités :
  - Échanges d'un pays avec lui-même
  - Lignes NULL / Trade Flow = Not Applicable
- Difficultés rencontrées :
  - Contraintes techniques : NULL, unicité, jointures complexes.
  - Problèmes de données : fusion sans doublons, nettoyage, volume élevé.

- Évolution annuelle des importations mondiales de technologies bas-carbone.
- Top 5 technologies les plus échangées par type de flux
- Top pays exportateurs par valeur totale échangée (3 dernières années)
- Comparaison des échanges bilatéraux vs nationaux sur une année donnée

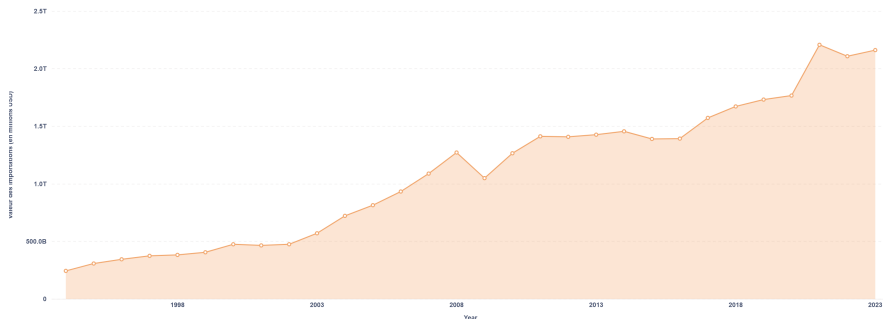
- Utilisation de Metabase pour interroger la BD
- Utilisation de Tableau pour valider les valeurs issues du CSV
- Comparaison SQL / Python → vérification de l'intégrité
- Exemples visuels :



# Requête SQL – Importations bas carbone

```
SELECT year, SUM(trade_value) AS "Valeur des
importations (en millions USD)"
FROM Trade
JOIN year y USING(id_year)
JOIN indicator USING(id_indicator)
JOIN trade_flow USING(id_trade_flow)
WHERE trade_flow = 'Imports'
      AND indicator = 'Imports of low carbon technology
products'
GROUP BY year
ORDER BY year;
```

Listing 1: Requête SQL pour les importations de technologies bas carbone



**Figure:** Évolution annuelle des importations mondiales de technologies bas-carbone

# Relation Tableau – Top 5 technologies

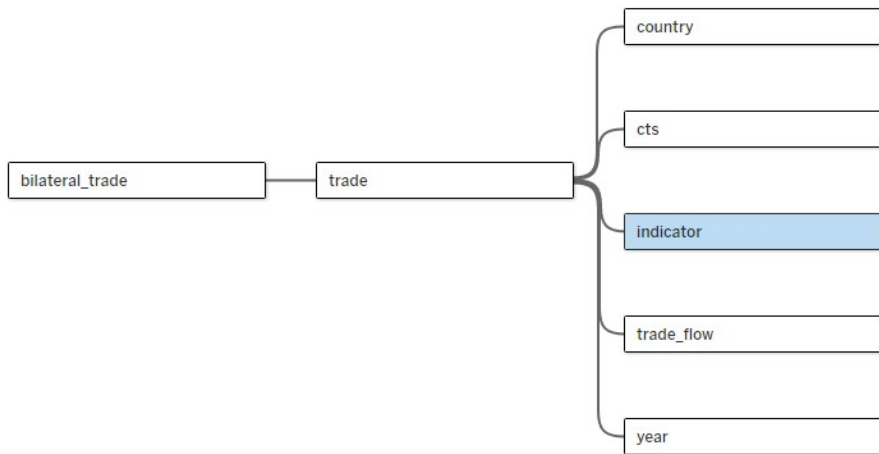


Figure: Relations établie sur Tableau Desktop

## Top 5 technologies les plus échangées par type de flux

Trade Flow	Cts Name	
Not Applicable	Trade in Low Carbon Technology Products	66 540 287 104 001
	Trade in Low Carbon Technology Products; Trade Balance	802 826 776 438
	Trade in Low Carbon Technology Products; Comparative Advantage	1 680
Exports	Trade in Low Carbon Technology Products; Exports	33 671 741 958 940
Imports	Trade in Low Carbon Technology Products; Imports	32 886 089 437 153

**Figure:** Évolution annuelle des importations mondiales de technologies bas-carbone

- Base de données fonctionnelle et fidèle au CSV (dans nos rêves)
- Respect des étapes du processus de modélisation
- Difficultés sur l'import (pivot, valeurs nulles, sens du flux)
- Vérification croisée avec Python
- Améliorations possibles :

Merci pour votre attention

**Questions ?**