

BUT Science des Données

SAE S2.01

Conception et Implémentation d'une Base de Données Relationnelles

1. Connaissance des deux base de données.....	1
2. Modélisation Entités-Associations et Relationnelle.....	2
2.1. Modélisation Entités-Associations.....	2
2.2. Schéma Relationnel.....	4
3. Script SQL - Création de la base de données.....	5
4. Script SQL - Peuplement de la base de données.....	6
4.1. Difficultés rencontrées.....	10
4.2. Vérification du Script.....	10
5. Interrogation de la base de données et visualisation des résultats.....	11
5.1. Tableau de Bord.....	12
5.2. Script SQL - Tableau de Bord.....	13
5.3. Analyse et Interprétation.....	14
5.3.1. Un pic exceptionnel en 2021 : un effet rebond.....	14
5.3.2. La Chine, pilier central du commerce vert.....	14
5.3.3. Le COVID a renforcé sa position dominante.....	14
5.3.4. Une croissance continue depuis 1994.....	14
5.3.5. Conclusion.....	14
5.3.6. Sources.....	15
6. Tableau de Répartition des Tâches.....	15

Ibrahim BENKHERFELLAH et Axel COULET - BUT1 SD - PRAXIS 2

1. Connaissance des deux base de données

Nous avons tout d'abord commencer par lire, comprendre et assimiler les deux sources de données afin d'avoir une vue globale de la situation et du contexte. Ainsi, nous avons répertorié les différentes colonnes des fichiers CSV et avons établi une liste des colonnes présentes dans les deux fichiers (l'intersection). Voici le résultat :

Bilateral Trade in Low Carbon ...	Trade in Low Carbon ...	Type
Country	Country	VARCHAR(50)
IS02	IS02	CHAR(2)
IS03	IS03	CHAR(3)
Counterpart Country		VARCHAR(50)
Counterpart IS02		CHAR(2)
Counterpart IS03		CHAR(3)
Indicator	Indicator	VARCHAR(80)
Unit	Unit	VARCHAR(50)
Source	Source	VARCHAR
CTS_Code	CTS_Code	VARCHAR(6)
CTS_Name	CTS_Name	VARCHAR(100)
CTS Full Descriptor	CTS Full Descriptor	VARCHAR(150)
Trade Flow	Trade Flow	VARCHAR(20)
Scale	Scale	VARCHAR(50)
F1994, ... , F2023	F1994, ... , F2023	DOUBLE PRECISION

On remarque donc que les 3 colonnes commençant par Counterpart ne figurent pas dans l'intersection des deux colonnes. Nous nous sommes posé la question suivante : quelles colonnes pouvons-nous regrouper ? Ainsi, nous savons que différentes colonnes permettent d'identifier un pays, exemple : country, ISO2, ISO3... D'autres permettent d'identifier les CTS (pour probablement Clean Technology Subsector) : CTS_Code, CTS_Name, etc...

Nous avons donc pu identifier différents groupes dans ces bases de données. Une fois tous les "groupes" identifiés, nous avons commencé à mettre en place notre schéma entité-association. On a été

Ibrahim BENKHERFELLAH et Axel COULET - BUT1 SD - PRAXIS 2

confronté à de nombreux problèmes et en particulier un qui est le suivant : Comment fusionner nos deux bases de données tout en gardant l'information qu'il y a deux pays, l'un import et l'autre export ?

Le problème le plus complexe à résoudre : éviter toute redondance avec les pays qui exportent et ceux qui importent.

2. Modélisation Entités-Associations et Relationnelle

2.1. Modélisation Entités-Associations

Nous avons tout d'abord voulu, faire deux tables : une pour le pays qui importe et une autre pour celui qui exporte. Or, directement, nous avons conclu que ce n'était pas le choix le plus judicieux.

Pourquoi ? Car en faisant cela, nous nous retrouverons avec deux tables presque identiques avec comme seule différence un petit attribut qui détermine l'origine de transport. Cela crée une redondance très importante et en stockant deux fois les mêmes informations, cela prend du stockage supplémentaire.

Après avoir demandé quelques conseils à Mme ZARGAYOUNA, nous avons appris qu'il existait une solution à ce problème. En effet, il est possible de faire une **Association Réflexive**.

Une association réflexive (ou unaire), dans le contexte d'un modèle conceptuel de données, **est un ensemble de liens de même nature entre éléments d'une même entité**. Autrement dit, il s'agit d'une association qui relie une entité à elle-même. Elle peut être de type $[n,n]$, $[1,n]$, ou $[1,1]$.

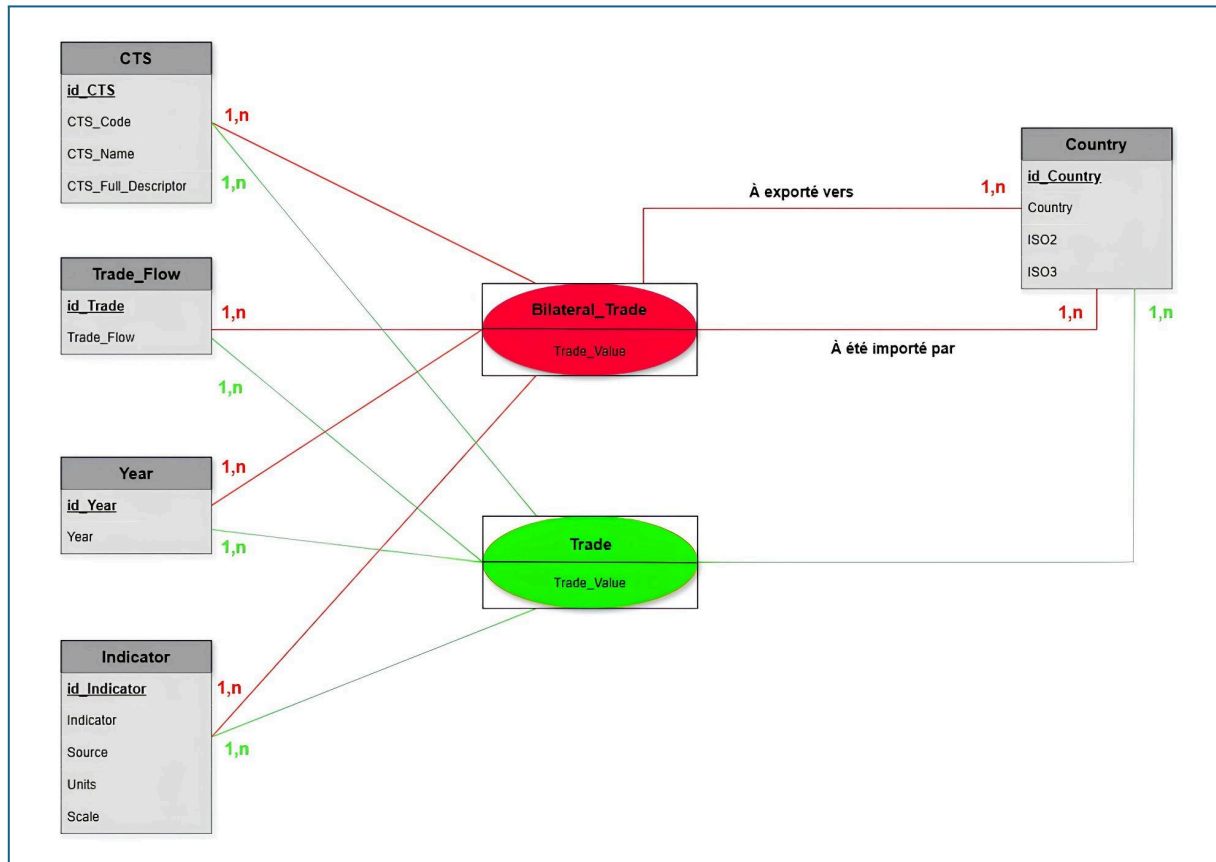
Dans notre cas, nous avons une association réflexive de types $[n,n]$ symétrique avec propriété. La propriété est dans notre situation, ce qu'on a décidé de nommer : *Trade_Value*.

Veuillez changer de page pour voir notre schéma Entité Association.

[Voir Schéma]

Le schéma Entité-Association terminé est supposé correct. Nous avons dans un classeur Excel (sur deux feuilles différentes) importé les deux bases de données. Nous avons ensuite essayé de déterminer la cohérence et le bon fonctionnement de notre modélisation. Après avoir effectué quelques tests, notre modèle EA semble être cohérent et fonctionnel.

Ibrahim BENKHERFELLAH et Axel COULET - BUT1 SD - PRAXIS 2



Le schéma comporte au total **5 types entités** ainsi que **2 types association**.

Types Entités :

- ❖ **Entité Country** : cette entité regroupe 3 attributs : Country, ISO2 et ISO3. Ces attributs ont été regroupés car ils ont des caractéristiques similaires. En effet, ces attributs permettent tous les trois d'identifier un pays. Il y a bien évidemment en plus, la clé associée à la table Country (id_Country)
- ❖ **Entité CTS** : par le même raisonnement qu'avec l'entité Country, les attributs liés au CTS ont été regroupés en une seule entité. avec en plus id_CTS.
- ❖ **Entité Trade Flow** : cette entité ne prend comme attribut seulement Trade_Flow ainsi que sa clé associée.
- ❖ **Entité Year** : nous avons créé une entité Year afin de conserver quelque part les années. Cela est particulièrement utile et primordial afin de faire des analyses temporelles. Car aucune colonne ne nous donne réellement l'année de l'échange.
- ❖ **Entité Indicator** : Enfin le dernier type entité, Indicator. Celui-ci regroupe d'un côté, les attributs permettant de déterminer la nature de l'échange (*Import, Export, etc.*) via Indicator mais également l'unité de celui-ci via l'attribut Units (*US Dollars, percent*). Nous avons mis Source ici car il permet de donner des indications sur la donnée (comment a-t-elle été collectée). L'attribut Scale, qui vaut toujours *Unit*, décrit l'échelle de mesure d'un indicateur, pas le flux commercial. C'est donc une propriété logique de Indicator.

Ibrahim BENKHERFELLAH et Axel COULET - BUT1 SD - PRAXIS 2**Types associations :**

- ❖ **Association Bilateral_Trade** : Cette association réflexive relie deux entités de type `Country`, ce qui signifie que l'échange se fait entre deux pays appartenant à la même entité. C'est une association réflexive de type $[n,n]$ symétrique avec propriétés. En effet, un pays A peut échanger avec un pays B et inversement.

Cette modélisation permet de représenter des échanges bilatéraux directionnels.

C'est-à-dire : → "Émet un flux vers" (Colonne Country) et "Reçoit un flux de" (Colonne Counterpart_Country).

- ❖ **Association Trade** : Cette association classique (non réflexive) relie l'entité `Country` à plusieurs autres dimensions : `Indicator`, `CTS`, `Trade_Flow` et `Year`. Elle permet de modéliser les données agrégées par pays : par exemple, le nombre d'importation fait par l'Allemagne en 2016. Chaque tuple de cette association correspond à une mesure pour un pays donné, selon un indicateur, une catégorie technologique, un flux commercial et une année.

Dans le schéma EA, certaines branches sont annotées de manière à exprimer les rôles de chaque entité dans la mesure.

2.2. Schéma Relationnel

Le schéma EA est fini et supposé correct. Nous avons fait le schéma relationnel de notre schéma EA. Pour ce faire nous avons simplement appliqué les formules du cours afin de le normaliser jusqu'en BOYCE CODD.

```
COUNTRY(id_Country, Country, IS02, IS03)
CTS(id_CTS, CTS_Code, CTS_Name, CTS_Full_Descriptor)
TRADE_FLOW(id_Trade_Flow, Trade_Flow)
INDICATOR(id_Indicator, Indicator, Source, Units, Scale)
YEAR(id_Year, Year)
BILATERAL_TRADE(id_Country, id_Counterpart_Country, id_Indicator, id_CTS, id_Trade_Flow, id_Year, trade_value)
→ où id_Country, id_Counterpart_Country, id_Indicator, id_CTS, id_Trade_Flow et id_Year font respectivement références à COUNTRY, COUNTRY, INDICATOR, CTS, YEAR, TRADE_FLOW.
TRADE(id_Country, id_Indicator, id_CTS, id_Trade_Flow, id_Year, trade_value)
→ où id_country, id_Indicator, id_CTS, id_Trade_Flow et id_Year font respectivement références à COUNTRY, INDICATOR, CTS, TRADE_FLOW et YEAR.
```

Le schéma relationnel étudié respecte d'abord la **première forme normale (1NF)**. En effet, tous les attributs contiennent des valeurs atomiques : il n'existe ni listes, ni champs composés, ni attributs multi-valués. Une transformation importante a été effectuée pour atteindre cette forme : les colonnes correspondant aux années (F1994 à F2023) ont été dépivotées en lignes, chaque enregistrement étant désormais caractérisé par une année et une valeur associée. Cette restructuration garantit que le schéma est bien en 1NF.

Ibrahim BENKHERFELLAH et Axel COULET - BUT1 SD - PRAXIS 2

Ensuite, le schéma respecte la **deuxième forme normale (2NF)**. Étant déjà en 1NF, il fallait pour cela éliminer les dépendances fonctionnelles partielles, c'est-à-dire les attributs dépendant uniquement d'une partie de la clé primaire. Par exemple, les informations comme le nom ou le descripteur d'un indicateur, qui dépendent uniquement du code CTS, ont été extraites dans une table distincte. Ainsi, toutes les dépendances partielles ont été supprimées, assurant que le schéma est conforme à la 2NF.

Le schéma atteint également la **troisième forme normale (3NF)**. Étant en 2NF, il fallait ensuite éliminer les dépendances transitives, c'est-à-dire celles où un attribut non-clé dépend d'un autre attribut non-clé. Un exemple notable est celui de l'unité de mesure d'un indicateur : elle dépend directement de l'indicateur et non indirectement du pays. De cette manière, chaque attribut non-clé dépend directement de la clé primaire, et uniquement de celle-ci, ce qui confirme la conformité à la 3NF.

Enfin, le schéma respecte la **forme normale de Boyce-Codd (BCNF)**. Cette forme exige que pour toute dépendance fonctionnelle non triviale, la partie gauche soit une super-clé. Cela signifie qu'il ne doit exister aucune dépendance où un attribut non-clé détermine un autre attribut non-clé. Par exemple, dans la table *Bilateral_Trade*, seule la clé complète permet de déterminer la valeur de l'échange commercial (*trade_value*), garantissant l'absence de violations de la BCNF.

3. Script SQL - Création de la base de données

Pour l'implémentation en SQL, nous avons créé les tables à partir du modèle relationnel précédemment établi. Chaque entité représentée dans le schéma relationnel correspond à une table distincte. Les attributs de ces entités ont été définis avec soin, chacun étant associé à un type de donnée approprié (par exemple *VARCHAR*, *INTEGER*, *DATE*, etc.), en accord avec la nature des informations qu'ils doivent contenir. Cette étape a permis de structurer la base de données de manière cohérente et conforme aux règles de modélisation relationnelle.

```
CREATE TABLE COUNTRY(  
    id_Country SMALLSERIAL PRIMARY KEY,  
    Country VARCHAR(50),  
    IS02 CHAR(2),  
    IS03 CHAR(3)  
);
```

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

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

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

Ibrahim BENKHERFELLAH et Axel COULET - BUT1 SD - PRAXIS 2

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

```
CREATE TABLE BILATERAL_TRADE (  
    id_Country SMALLINT REFERENCES Country(id_Country),  
    id_Counterpart_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_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)  
);
```

```
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_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)  
);
```

4. Script SQL - Peuplement de la base de données

Après avoir créé les différentes tables de notre base de données, nous avons entamé une réflexion sur la manière la plus efficace de les peupler. À l'issue de nos recherches, nous avons identifié une méthode à la fois simple et fiable : l'utilisation de tables temporaires pour importer les données issues des fichiers CSV. Ces tables temporaires servent de structure intermédiaire, facilitant le stockage initial des données brutes. Grâce à des requêtes SQL ciblées, nous pouvons ensuite transférer et intégrer ces données dans les tables définitives du schéma relationnel, en assurant leur conformité aux contraintes définies (clés primaires, étrangères, types, etc.).

Le peuplement se fait donc de la manière suivante:

CSV → TEMPORARY TABLE → TABLE

Ibrahim BENKHERFELLAH et Axel COULET - BUT1 SD - PRAXIS 2

Voici les deux tables temporaires que nous avons utilisées pour importer les données depuis nos fichiers CSV avant de les transférer dans les tables finales.

```
CREATE TEMPORARY TABLE
tmp_bilateral_trade_data (
  ObjectId INTEGER PRIMARY KEY,
  Country VARCHAR(50),
  IS02 CHAR(2),
  IS03 CHAR(3),
  Counterpart_Country VARCHAR(50),
  Counterpart_IS02 CHAR(2),
  Counterpart_IS03 CHAR(3),
  Indicator VARCHAR(80),
  Unit VARCHAR(50),
  Source VARCHAR,
  CTS_Code VARCHAR(6),
  CTS_Name VARCHAR(100),
  CTS_Full_Descriptor VARCHAR(150),
  Trade_Flow VARCHAR(20),
  Scale VARCHAR(5),
  F1994 DOUBLE PRECISION,
  ...
  F2023 DOUBLE PRECISION,
);
```

```
CREATE TEMPORARY TABLE tmp_trade_data (
  ObjectId INTEGER PRIMARY KEY,
  Country VARCHAR(50),
  IS02 CHAR(2),
  IS03 CHAR(3),
  Indicator VARCHAR(80),
  Unit VARCHAR(50),
  Source VARCHAR,
  CTS_Code VARCHAR(6),
  CTS_Name VARCHAR(100),
  CTS_Full_Descriptor VARCHAR(150),
  Trade_Flow VARCHAR(20),
  Scale VARCHAR(50),
  F1994 DOUBLE PRECISION,
  ...
  F2023 DOUBLE PRECISION
);
```

À la suite de la création des tables temporaires, nous avons utilisé la commande COPY afin d'y importer les données issues des fichiers CSV. Cette commande permet un chargement rapide et efficace des données dans les tables temporaires, facilitant ainsi leur traitement ultérieur avant insertion dans les tables finales.

```
\copy <temp_table> FROM 'chemin/vers/fichier.csv' DELIMITER ',' CSV HEADER;
```

Après avoir inséré les données dans les tables temporaires, nous les avons extraites puis insérées dans les tables finales à l'aide de scripts SQL adaptés. Ces scripts permettaient de transférer les informations pertinentes tout en respectant la structure et les contraintes de la base.

```
-- Country
INSERT INTO Country(Country, IS02, IS03)
SELECT Country, IS02, IS03
FROM (
  SELECT DISTINCT Country, IS02, IS03 FROM tmp_trade_data
  UNION
  SELECT DISTINCT Counterpart_Country, Counterpart_IS02, Counterpart_IS03
```


Ibrahim BENKHERFELLAH et Axel COULET - BUT1 SD - PRAXIS 2

```
FROM tmp_bilateral_trade_data
)
ORDER BY Country;
```

```
-- Indicator
INSERT INTO Indicator(Indicator, Source, Units, Scale)
SELECT DISTINCT ON (Indicator)
    Indicator, Source, Unit, Scale
FROM (
    SELECT Indicator, Source, Unit, Scale FROM tmp_trade_data
    UNION ALL
    SELECT Indicator, Source, Unit, Scale FROM tmp_bilateral_trade_data
)
ORDER BY Indicator, Source;
```

```
-- CTS
INSERT INTO CTS(CTS_Code, CTS_Name, CTS_Full_Descriptor)
SELECT DISTINCT ON (CTS_Code)
    CTS_Code, CTS_Name, CTS_Full_Descriptor
FROM (
    SELECT CTS_Code, CTS_Name, CTS_Full_Descriptor FROM tmp_trade_data
    UNION ALL
    SELECT CTS_Code, CTS_Name, CTS_Full_Descriptor FROM tmp_bilateral_trade_data
)
ORDER BY CTS_Code, CTS_Name;
```

```
-- Trade_Flow
INSERT INTO Trade_Flow(Trade_Flow)
SELECT Trade_Flow
FROM (
    SELECT Trade_Flow FROM tmp_trade_data
    UNION ALL
    SELECT Trade_Flow FROM tmp_bilateral_trade_data
)
GROUP BY Trade_Flow
ORDER BY Trade_Flow;
```

```
-- Year
INSERT INTO Year(id_Year, Year)
SELECT y, make_date(y, 1, 1)
FROM generate_series(1994, 2023) AS y;
```

Ibrahim BENKHERFELLAH et Axel COULET - BUT1 SD - PRAXIS 2

```
-- Trade
INSERT INTO Trade(
    id_Country, id_Indicator, id_CTS, id_Trade_Flow, id_Year, trade_value
)
SELECT c.id_Country, i.id_Indicator, cts.id_CTS, tf.id_Trade_Flow, y.id_Year,
flat.trade_value
FROM (
    SELECT td.Country, td.Indicator, td.CTS_Code, td.Trade_Flow,
    UNNEST(ARRAY[1994, 1995, ... , 2022, 2023]) AS year_year,
    UNNEST(ARRAY[F1994, F1995, ... , F2022, F2023]) AS trade_value
    FROM tmp_trade_data td
) AS flat
JOIN Country c ON c.Country = flat.Country
JOIN Indicator i ON i.Indicator = flat.Indicator
JOIN CTS cts ON cts.CTS_Code = flat.CTS_Code
JOIN Trade_Flow tf ON tf.Trade_Flow = flat.Trade_Flow
JOIN Year y ON y.id_Year = flat.year_year
WHERE flat.trade_value IS NOT NULL;
```

```
-- Bilateral_Trade
INSERT INTO Bilateral_Trade(
    id_Country, id_Counterpart_Country, id_Indicator,
    id_CTS, id_Trade_Flow, id_Year, trade_value
)
SELECT c1.id_Country, c2.id_Country, i.id_Indicator, cts.id_CTS,
tf.id_Trade_Flow, y.id_Year, flat.trade_value
FROM (
    SELECT tb.Country, tb.Counterpart_Country, tb.Indicator, tb.CTS_Code,
    tb.Trade_Flow,
    UNNEST(ARRAY[1994, 1995, ... , 2022, 2023]) AS year_year,
    UNNEST(ARRAY[F1994, F1995, ... , F2022, F2023]) AS trade_value
    FROM tmp_bilateral_trade_data tb
) AS flat
JOIN Country c1 ON c1.Country = flat.Country
JOIN Country c2 ON c2.Country = flat.Counterpart_Country
JOIN Indicator i ON i.Indicator = flat.Indicator
JOIN CTS cts ON cts.CTS_Code = flat.CTS_Code
JOIN Trade_Flow tf ON tf.Trade_Flow = flat.Trade_Flow
JOIN Year y ON y.id_Year = flat.year_year
WHERE flat.trade_value IS NOT NULL ;
```

Ibrahim BENKHERFELLAH et Axel COULET - BUT1 SD - PRAXIS 2

4.1. Difficultés rencontrées

L'un des premiers problèmes rencontrés lors du traitement des données concerne l'insertion dans les tables *Bilateral_Trade* et *Trade*. En effet, les données provenant des fichiers sources étaient structurées de manière dite « large », c'est-à-dire avec une colonne distincte pour chaque année. Cette organisation ne correspondait pas au format attendu par notre base de données finale, qui exigeait une structure normalisée, avec une seule ligne par année et par valeur de commerce.

Pour résoudre ce problème, nous avons dû dépivoter les colonnes représentant les années et leurs valeurs correspondantes. Cela nous a demandé de nombreuses recherches pour bien comprendre le fonctionnement des outils SQL adaptés à ce type de transformation.

Nous voulions utiliser la fonction [UNPIVOT](#). Cependant, celle-ci n'est pas disponible sur PostgreSQL. Nous avons finalement utilisé des requêtes basées sur la fonction [UNNEST](#), qui nous a permis de restructurer les données de manière efficace. Ces requêtes extraient les colonnes descriptives (comme le pays, le pays partenaire, le type d'indicateur, etc.), puis crée deux nouvelles colonnes : l'une pour les années (*year_year*), et l'autre pour les valeurs commerciales (*trade_value*), en transformant chaque ligne contenant plusieurs années en plusieurs lignes avec une seule année chacune.

Le premier tableau utilisé dans la fonction *UNNEST* contient la liste des années de 1994 à 2023, tandis que le second contient les valeurs des colonnes F1994 à F2023. Ces deux colonnes sont associées position par position, ce qui permet de générer des paires année/valeur cohérentes pour chaque ligne initiale.

Les jointures servent à remplacer les libellés par leurs identifiants afin d'insérer des clés étrangères conformes au modèle relationnel. Grâce à cette transformation, les données ont pu être insérées correctement dans la table, en respectant la structure relationnelle prévue et facilitant ainsi les traitements analytiques futurs.

NB : Nous avons créé une table *Year* contenant à la fois un identifiant (*id_Year*) et un attribut *year* au format *DATE*. L'identifiant permet des jointures avec les tables peuplées via *UNNEST*(*ARRAY*[...]), tandis que le champ *DATE* évite les problèmes d'interprétation dans Metabase, où les années en *INTEGER* sont mal reconnues quelquefois dans les analyses temporelles. Sinon, je vous l'accorde, l'année suffit amplement dans *Year*.

4.2. Vérification du Script

Une fois cette solution trouvée, nous avons construit notre script SQL autour de la fonction *UNNEST*, afin de transformer correctement les données avant leur insertion.

Pour vérifier que l'opération s'était déroulée correctement, nous avons mis en place un fichier de contrôle, généré à l'aide d'un script Python. Ce fichier indiquait le nombre total de lignes attendues après transformation.

Ibrahim BENKHERFELLAH et Axel COULET - BUT1 SD - PRAXIS 2

Lors de la première exécution de notre script utilisant UNNEST, nous avons constaté un écart important : la table générée contenait environ 1,4 million de lignes, tandis que le fichier de contrôle en prévoyait 1,7 million. Cela indiquait clairement un problème au niveau de l'insertion, qu'il a fallu identifier et corriger.

5. Interrogation de la base de données et visualisation des résultats

Après avoir effectué quelques requêtes exploratoires pour mieux appréhender la structure de notre base de données, nous nous sommes posés une question centrale : **quelles interrogations pertinentes pourraient émerger du croisement entre les tables Trade et Bilateral_Trade ?**

C'est à partir de cette réflexion que nous avons orienté nos analyses vers des axes comparatifs et géographiques, notamment en identifiant les flux dominants, les relations bilatérales significatives, et les évolutions dans le temps. L'année **2021** a naturellement retenu notre attention en raison de son pic d'activité, permettant de concentrer certaines visualisations sur cette période stratégique.

Plutôt que d'enchaîner plusieurs requêtes indépendantes, nous avons choisi de construire un **tableau de bord global** pour proposer une lecture cohérente et synthétique des données. Comme évoqué précédemment, l'année **2021** s'est imposée comme un point de bascule, avec un **pic significatif des exportations** de technologies bas carbone. Nous nous sommes donc interrogés : **pourquoi ce pic ?** Et surtout, **quel acteur en est le principal moteur ?**

Nous avons donc construit un tableau de bord synthétique visant à identifier les **acteurs majeurs de 2021**, comprendre l'**origine du pic d'exportation**, et mettre en évidence la **répartition des flux** entre échanges bilatéraux et nationaux. L'objectif était de répondre à une question centrale : **qui a porté ce pic et vers quels pays ?**

Veuillez changer de page pour voir notre Tableau de Bord.

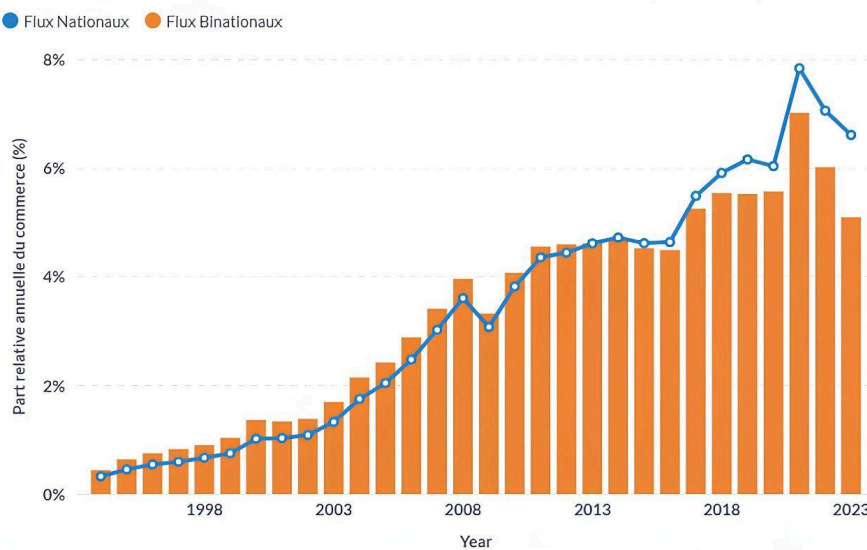
Ibrahim BENKHERFELLAH et Axel COULET - BUT1 SD - PRAXIS 2

5.1. Tableau de Bord

2021 : l'année de la domination chinoise dans le commerce vert

Comment les flux d'exportation de technologies bas carbone se sont-ils structurés à l'échelle mondiale en 2021, et quel a été le rôle stratégique de la Chine dans cette dynamique commerciale ?

Part annuelle des échanges bilatéraux vs nationaux dans le commerce mondial (1994 à 2023)

**399 Exports**

Nombre de flux d'exportation de la Chine enregistrés (2021)

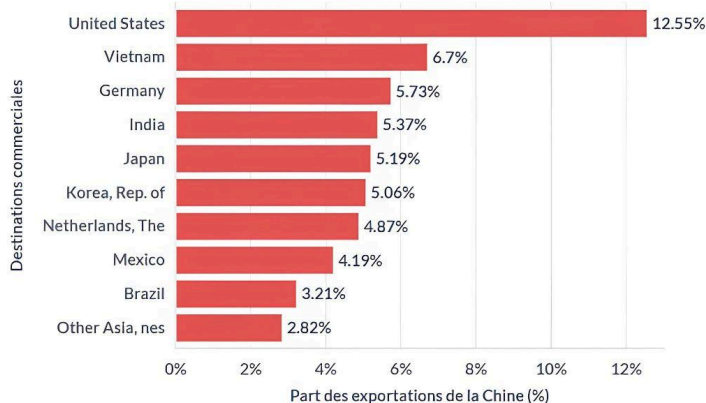
215 Pays

Nombre de pays importateurs depuis la Chine (2021)

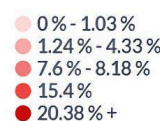
258,7B \$

Valeur totale des exportations de la Chine (2021)

Top 10 des pays importateurs de technologies en provenance de Chine (2021)



Répartition mondiale des exportations de technologies en 2021 (%)



Ce tableau de bord a été réalisé à 100% avec [Metabase](#) !

Fun fact : la requête du graphique en haut à gauche a pris 11 minutes et 42 secondes afin de générer le graphique, le PC en Position Latérale de Sécurité depuis...

Ibrahim BENKHERFELLAH et Axel COULET - BUT1 SD - PRAXIS 2

5.2. Script SQL - Tableau de Bord

```
-- Part annuelle des échanges bilatéraux vs nationaux dans le monde (1994 à 2023)
WITH totals AS (
  SELECT
    y.year,
    SUM(t.trade_value) AS total_trade,
    SUM(bt.trade_value) AS total_bilateral
  FROM year AS y
  LEFT JOIN trade AS t ON t.id_year = y.id_year
  LEFT JOIN country c_t ON t.id_country = c_t.id_country
  LEFT JOIN bilateral_trade AS bt ON bt.id_year = y.id_year
  WHERE c_t.iso2 NOT IN ('World')
    AND t.id_indicator IN (
      SELECT id_indicator FROM indicator WHERE units ILIKE '%dollar%'
    )
    AND bt.id_indicator IN (
      SELECT id_indicator FROM indicator WHERE units ILIKE '%dollar%'
    )
  GROUP BY y.year
)
SELECT
  year,
  total_trade / SUM(total_trade) OVER () AS pct_trade,
  total_bilateral / SUM(total_bilateral) OVER () AS pct_bilateral
FROM totals;
```

```
-- Nb d'exports → Chine (2021)
SELECT COUNT(*)
FROM bilateral_trade
JOIN country USING(id_country)
JOIN year USING(id_year)
JOIN trade_flow USING(id_trade_flow)
JOIN indicator USING(id_indicator)
WHERE country IN (
  SELECT DISTINCT country FROM country
  WHERE country ILIKE '%china%'
)
AND trade_flow = 'Exports'
AND units ~* '[$.]*dollar'
AND EXTRACT(YEAR FROM year) = 2021;
```

```
-- Nb pays Importateur → Chine (2021)
SELECT COUNT(DISTINCT id_counterpart_country)
FROM bilateral_trade
JOIN country USING(id_country)
JOIN year USING(id_year)
JOIN trade_flow USING(id_trade_flow)
JOIN indicator USING(id_indicator)
WHERE country IN (
  SELECT DISTINCT country FROM country
  WHERE country ILIKE '%china%'
)
AND trade_flow = 'Exports'
AND units ILIKE '%dollar%'
AND EXTRACT(YEAR FROM year) = 2021;
```

Malheureusement, les autres requêtes (dont les plus pertinentes) prennent trop de place. Vous les retrouverez donc dans un dossier en annexe fourni avec ce rapport (ici : `./scripts/script_dataviz/`).

Ibrahim BENKHERFELLAH et Axel COULET - BUT1 SD - PRAXIS 2

5.3. Analyse et Interprétation

5.3.1. Un pic exceptionnel en 2021 : un effet rebond

L'année 2021 marque un pic historique des exportations de technologies bas carbone. Ce bond s'explique par la reprise économique mondiale après le ralentissement dû au COVID-19. Les pays ont relancé massivement leurs investissements dans les infrastructures, notamment les énergies renouvelables. La demande mondiale a explosé, entraînant mécaniquement une hausse des échanges. Cette dynamique a profité aux pays capables de produire vite et à grande échelle.

5.3.2. La Chine, pilier central du commerce vert

Ce pic a mis en lumière la domination chinoise. Depuis plus d'une décennie, la Chine a investi lourdement dans l'industrie des technologies vertes. Résultat : elle est aujourd'hui le fournisseur principal de composants solaires, d'éoliennes et de batteries. En 2021, quasiment tous les grands importateurs (États-Unis, Vietnam, Inde...) s'approvisionnent massivement en Chine. Ce leadership vient autant d'une stratégie industrielle planifiée que de sa capacité de production hors norme.

5.3.3. Le COVID a renforcé sa position dominante

La pandémie, loin de freiner la Chine, l'a plutôt renforcée. Dès 2020, elle a relancé sa production plus vite que les autres. En 2021, alors que de nombreuses chaînes logistiques mondiales étaient encore perturbées, la Chine était déjà pleinement opérationnelle. Elle est ainsi devenue le fournisseur "de secours" pour de nombreux pays en manque de matériel, consolidant encore sa place sur le marché mondial.

5.3.4. Une croissance continue depuis 1994

Le pic de 2021 n'est pas un événement isolé. Depuis les années 1990, le commerce mondial des technologies vertes ne cesse de croître. Les accords climatiques, la baisse des coûts et la mondialisation des chaînes de valeur ont favorisé cette expansion. La Chine a su tirer parti de cette tendance pour se positionner comme l'acteur-clé. Le graphique du haut à gauche montre cette évolution progressive, avec un bond net depuis les années 2010, culminant en 2021.

5.3.5. Conclusion

En somme, les analyses réalisées mettent en évidence le rôle central de la Chine dans les échanges mondiaux de technologies bas carbone, notamment lors du pic observé en 2021. Cette dynamique résulte à la fois d'une croissance régulière sur le long terme amorcée depuis les années 1990 et d'un contexte particulier de reprise post-COVID, qui a accentué la dépendance aux capacités industrielles chinoises.

Ibrahim BENKHERFELLAH et Axel COULET - BUT1 SD - PRAXIS 2

5.3.6. Sources

- [IEA](#) (International Energy Agency)
- Les Echos (2021) : [Record d'exportations chinoises en 2021](#)
- Sino Shipping (2021) : [Les principales importations et exportations de la Chine](#)
- [Statista](#)

6. Tableau de Répartition des Tâches

Tâches	Réalisé par :
Connaissance des bases de données	Tous les deux
Modélisation Entité Association	55% Ibrahim et 45% Axel : <ul style="list-style-type: none">- Schéma Initial réalisés ensemble- Schéma Final refait par Ibrahim
Schéma Relationnel	Tous les deux
Script SQL - Création de la base de données	Tous les deux
Script SQL - Peuplement de la base de données	55% Ibrahim et 45% Axel : <ul style="list-style-type: none">- Script réalisés par Axel- Peuplement des tables : Bilateral_Trade et Trade ; par Ibrahim
Interrogation de la base de données : <ul style="list-style-type: none">- Scripts SQL - Tableau de bord- Tableau de Bord- Analyse et Interprétation	Ibrahim
Création du Diaporama (Oral)	Ibrahim → car Axel indisponible (justifié)
Rédaction du rapport final	Tous les deux

NB : L'ensemble des scripts créés sera mis à votre disposition dans le dossier Script.