# Vue générale du modèle

Dans le cadre du projet « **Baromètre** », notre objectif est de créer des tableaux contenant plusieurs indicateurs et statistiques clés. Pour cela, nous utiliserons un moteur de calcul **en Python** afin de gérer le nombre immense d'opérations nécessaires pour remplir, en fin de compte, la base de données finale avec tous ces indicateurs.

Ce moteur de calcul contiendra 5 parties :

1. **Mise en œuvre de la BDD :** Un *script 00-Creation\_BDD.sql* doit créer la base de données avec tous les tables utilisées et leurs contraintes, il doit aussi respecter précisément les consignes fournies pour chaque table afin de préserver la cohérence du model
2. **Import des données** : Un programme *01-ImportCsvBdd.py* extraira les données sources des fichiers CSV et les insérera ensuite dans leurs tables respectives.
3. **Synthèse des requêtes SQL** : Un programme *02-Remplir\_Refsql.py* construira les requêtes SQL en tenant compte des filtres et des règles appliquées à chaque ligne et colonne.
4. **Exécution des requêtes SQL (CALCUL SOURCE)** : Un programme

*03-Remplir\_Result\_Source.py* exécutera ces requêtes SQL.

1. **Calcul des indicateurs composés** : Un dernier programme

*04-Remplir\_Result\_Calcul.py* agira sur les colonnes contenant des calculs dépendant d'autres colonnes sources.

Chaque partie doit être capable d'être exécutée indépendamment. De plus, chacune sera associée à son propre journal de logs qui permettra de tracer les opérations

# Mise en œuvre de la BDD

## Présentation

The first of step of our program is to establish the necessary database architecture

Using an SQL script. this script has to create a Mysql database containing all the following 13 tables :

2 tables pour les données source

**Vw\_cube\_dplt\_icare\_bpce\_encr**

**Vw\_cube\_vbpce\_apc\_faits**

9 tables pour le croisement entre ligne et colonne

**prm\_cols :**

**prm\_cols\_calcul**

**prm\_cols\_composant**

**prm\_cols\_filtre**

**prm\_rows**

**prm\_rows\_filtre**

**prm\_ra\_liens**

**prm\_tdb\_objets**

**prm\_sql\_model**

1 table for the SQL queries

**prm\_ref\_sql**

1 table for the final result (Source and Calcul)

**prm\_ref\_result**

the relationship between all these tables is further shown in the drawing below :

## Schéma de la base de données

Une image contenant texte, capture d’écran, diagramme, Parallèle

Description générée automatiquement

# Import des Données

## Presentation

Once the database created come the next step: data upload

As said before the data is originally written in multiple csv files by the users

Afterwards Its uploaded using a python script into the MySQL database.

Une image contenant ligne, diagramme, cercle

Description générée automatiquement

## Constraints and order of upload

The csv should have the exact same name as the tables they are supposed to fill

It is also very important to respect the correct order of upload since most of the tables have foreign keys.

The correct order is the following:

"prm\_tdb\_objets"

        "prm\_cols\_calcul"

        "prm\_sql\_model"

        "vw\_cube\_dplt\_icare\_bpce\_encr"

        "vw\_cube\_vbpce\_apc\_faits"

        "prm\_rows"

        "prm\_cols"

        "prm\_cols\_filtre"

        "prm\_ra\_liens"

        "prm\_rows\_filtre"

        "prm\_cols\_composant"

## Logging and monitoring

As for every part of the program this part has its individual log file named *CsvUpload.log* that saves all attempts to upload data. this functionality has 2 main goals:

1. To help debug by saving all the successful uploads to the log file, this way we can easily monitor the state of the database
2. To save errors into the file to make the problem solving easier for the users

Naturally the log files need to track the timestamp for each operation as well as the result (successful or error)

It should also have a sorting system that can classified entries according to their importance

For example, it’s possible to use the following classifications:

INFO – DEBUG – WARNING – ERROR – CRITICAL

# Synthesis of SQL queries

## Presentation

Now that the data has been uploaded to the from the CSVs successfully its time to create the sql queries . these sql queries will then be stored in a separate table named

Prm\_ref\_sql that will house the sql aswell as additional data like the row’s id the columns name etc..