Bonjour Monsieur, aujourd'hui on va vous présenter notre mini-projet GreenCity (BI).

On va organiser la présentation en trois parties :

1. d'abord les données et la conception de la base opérationnelle de facturation,
2. ensuite, chaque membre présentera l'ETL et le reporting de son Data Mart,
3. et on terminera par une synthèse rapide et l'automatisation.

On commence par la conception de la base de facturation `greencity_facturation`.
L'objectif de cette base est de représenter le fonctionnement opérationnel de GreenCity : gestion des régions et des bâtiments, installation des compteurs, gestion des clients, contrats et factures, puis suivi des paiements et des tarifs.

Concrètement, on a modélisé :

- `regions` pour la dimension géographique,
- `batiments` pour décrire les bâtiments (surface, type, adresse, région),
- `types_energie` et `compteurs` pour relier chaque compteur à un bâtiment et à un type d'énergie,
- `clients` et `contrats` pour représenter la relation client–compteur,
- `factures` et `paiements` pour l'aspect financier,
- `tarifs` pour l'historique des prix,
- et `temperatures` pour pouvoir faire plus tard des analyses de corrélation entre météo et consommation.

Enfin, pour le pilotage de l'ETL, on a ajouté une table technique `etl_control` qui stocke, pour chaque table source, la dernière date d'extraction et le nombre de lignes extraites. Ça nous permet de gérer l'incrémenta

Script de présentation (5 minutes) — DM1 Consommation Énergétique

Bonjour, je vais vous présenter mon travail sur le Data Mart 1 : Consommation Énergétique, et plus précisément le processus ETL complet, depuis les sources jusqu'au Data Warehouse et au dashboard Power BI.

1) Objectif et modèle du Data Mart (≈ 40 sec)

Le but de ce Data Mart est de suivre et analyser la consommation énergétique selon plusieurs axes : bâtiment, région, type d'énergie et temps.

Le modèle est en étoile : une table de faits `fait_consommation` et cinq dimensions : `dim_temps`, `dim_region`, `dim_batiment`, `dim_compteur`, `dim_type_energie`.
Le grain de la table de faits est une ligne par compteur et par jour, avec comme mesures la consommation et la température moyenne.

2) Sources de données (≈ 30 sec)

On utilise deux sources principales :

- Les fichiers JSON IoT qui contiennent les mesures horaires de consommation pour électricité, eau et gaz.
- La base MySQL `greencity_facturation` qui fournit les tables de référence : régions, bâtiments, compteurs, types d'énergie, ainsi que les températures.

3) Phase 1 — Extraction vers Staging (≈ 60 sec)

La phase d'extraction a pour objectif de copier les données brutes vers une zone staging sous forme de fichiers CSV.

Pour MySQL, j'ai créé 5 transformations d'extraction, toutes basées sur le même pattern :

- récupérer la dernière date d'extraction depuis la table `etl_control`,
- extraire uniquement ce qui a été modifié via `updated_at` (mode incrémental),
- générer un fichier staging `stg_*.csv`,
- compter le nombre de lignes extraites,
- mettre à jour `etl_control` avec le nouveau timestamp.

Pour les JSON IoT, il y a une transformation dédiée `tr_extract_json.ktr` :

- elle lit tous les fichiers JSON du dossier,
- elle parse le niveau racine puis le tableau `mesures`,
- elle aplatie les données en format tabulaire,
- elle unifie les champs `consommation_kWh` et `consommation_m3` dans une seule colonne consommation,
- elle sépare la date et l'heure,
- puis elle génère `stg_consommation_json.csv`.

4) Phase 2 — Transformation et nettoyage (≈ 90 sec)

Cette phase est le cœur de la qualité des données. Chaque fichier staging passe par une transformation de nettoyage.

- Nettoyage des régions : suppression des espaces parasites avec Trim, valeurs manquantes remplacées, dédoublonnage.
- Nettoyage des bâtiments : application d'une règle métier importante : surface ne peut pas être négative, donc j'utilise `ABS(surface_m2)`. J'ai aussi géré les valeurs nulles et normalisé le format de l'année de construction.
- Nettoyage des compteurs : standardisation des IDs en majuscules, statut en format propre, nettoyage de date d'installation : suppression de la partie horaire, remplacement des "/" en "-", conversion en type Date, tri puis dédoublonnage.
- Nettoyage des températures : c'est la transformation la plus complexe. J'ai géré plusieurs formats de date avec un step Modified JavaScript Value qui détecte un format

français et le convertit en yyyy-mm-dd. Ensuite j'arrondis les températures à 2 décimales, je trie par région et date, puis je dédoublonne pour garder une température par jour et par région.
- Nettoyage des consommations IoT : j'ai appliqué des règles de validation pour rejeter les consommations nulles, négatives ou aberrantes (ex : supérieures à 10 000). Les rejets sont envoyés vers un fichier `rejected_conso.csv` pour audit, et les données valides vont vers `clean_consommation.csv`.

5) Phase 3 — Chargement dans le Data Warehouse (≈ 60 sec)

Ensuite on charge dans `greencity_dw`.

- La dimension temps est pré-générée par procédure stockée, donc pas de chargement ETL pour `dim_temps`.
- Les autres dimensions (région, bâtiment, compteur, type énergie) sont chargées avec Insert/Update. L'idée est de gérer l'incrémental tout en créant des clés de substitution (surrogate keys).
- Pour la table de faits `fait_consommation`, j'ai fait :
    - lecture de la consommation + lecture des températures,
    - enrichissement via Stream Lookup sur `(id_region, date_mesure)` pour ajouter `temperature_moyenne`,
    - puis des lookups vers les dimensions afin de remplacer les clés métier par les clés techniques SK,
    - et enfin insertion dans `fait_consommation`.

6) Orchestration + Automatisation (≈ 40 sec)

Tout est orchestré dans un job Pentaho `job_etl_dm_consommation.kjb` en trois blocs : extraction, nettoyage, puis chargement.
Enfin, l'exécution est automatisée via un script batch `run_etl_dm1.bat`, planifié tous les jours à 02h00 avec Windows Task Scheduler.

7) Reporting Power BI (≈ 40 sec)

Pour le reporting DM1, j'ai construit un dashboard Power BI qui affiche :

- des KPI cards pour la consommation totale par type d'énergie,
- la répartition par quartier,
- un top 10 des bâtiments les plus énergivores,
- l'évolution mensuelle,
- une visualisation de corrélation consommation vs température,
- et la répartition des relevés IoT pour vérifier l'équilibre des données.

Conclusion : ce Data Mart fournit une vision fiable et exploitable de la consommation énergétique, avec un ETL incrémental, des contrôles qualité, et une restitution interactive via Power BI.

l et d’assurer la traçabilité.