



Dbt en action



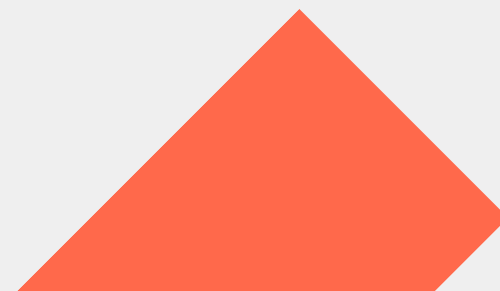
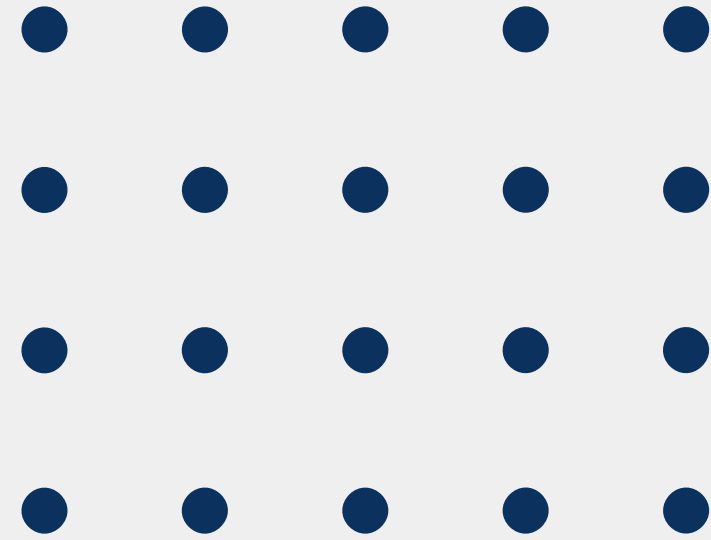
Réalisé Par:
OUMANOU Mohamed
ASMAOUI Larbi

Encadré Par:
Pr.NAJDI Lotfi



Plan:

- Introduction
- Qu'est-ce que dbt ?
- Fonctionnalités clés de DBT
- Environnement de DBT
- Les avantages de DBT
- Les limites de DBT
- Comment démarrer avec DBT
- Conclusion



Introduction:

Transformation des données : le moteur de la Business Intelligence

Dans un monde guidé par les données, la transformation des données joue un rôle essentiel en Business Intelligence (BI). Elle consiste à convertir des données brutes en un format exploitable pour l'analyse, le reporting et la prise de décision.

- **Nettoyage** : élimination des erreurs, incohérences et doublons.
- **Mise en forme** : standardisation des types de données (par exemple : dates, devises) pour faciliter l'analyse.
- **Création de champs dérivés** : génération de champs calculés à partir des données existantes (ex. : ventes totales par catégorie de produit).
- **Agrégation** : synthèse de grands ensembles de données pour une analyse plus rapide (ex. : chiffre d'affaires mensuel).

ETL vs ELT : quelle approche choisir ?:

ETL

Dans cette approche, les données sont d'abord extraites de différentes sources, puis transformées au format cible, avant d'être chargées dans l'entrepôt de données ou le système de BI. L'ETL garantit une qualité des données en amont, mais peut être plus lent pour les grands volumes de données.

ELT

Les données sont extraites puis chargées brutes dans l'entrepôt, où elles sont ensuite transformées. C'est une approche rapide pour les gros volumes, mais qui demande plus de ressources de calcul côté entrepôt.

Choisir entre ETL et ELT:

Le choix entre ETL et ELT dépend de plusieurs facteurs :

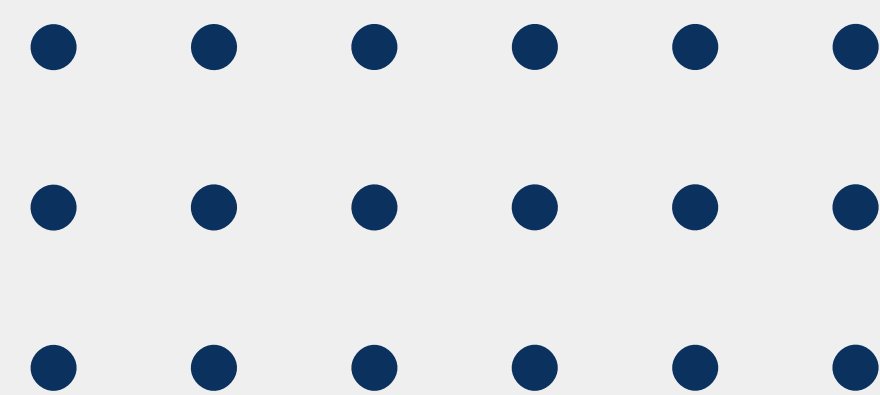
- **Taille et complexité des données :** pour de grands volumes complexes, l'ELT peut être plus rapide.
- **Exigences en matière de qualité des données :** si la qualité en amont est cruciale, l'ETL est souvent plus adapté.
- **Capacité de traitement de l'entrepôt de données :** il faut s'assurer que l'entrepôt peut supporter la charge de l'ELT.
- **Maturité de l'infrastructure BI :** les systèmes BI existants utilisent souvent déjà des processus ETL.

Contexte général:

Dans un paysage de l'analytique en constante évolution, les organisations font face à de nombreux défis pour exploiter pleinement leur données.

Entre la gestion de sources de données dispersées, l'assurance de la qualité des données, et la collaboration entre équipes, les environnements modernes peuvent freiner l'efficacité des analyses.

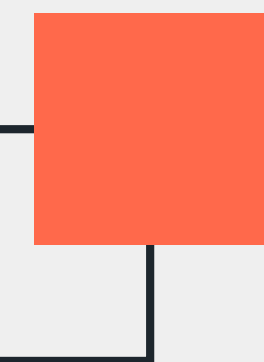
C'est dans ce contexte que des outils comme **DBT (Data Build Tool)** s'imposent comme des solutions innovantes, en intégrant **les meilleures pratiques de l'ingénierie logicielle** pour répondre à ces enjeux.



Qu'est-ce que dbt ?



dbt





DBT:

DBT (Data Build Tool) est un outil de transformation qui permet aux analystes et aux ingénieurs de transformer des données dans leur entrepôt de données en utilisant SQL. (Il effectue les transformations directement dans le système de destination.)

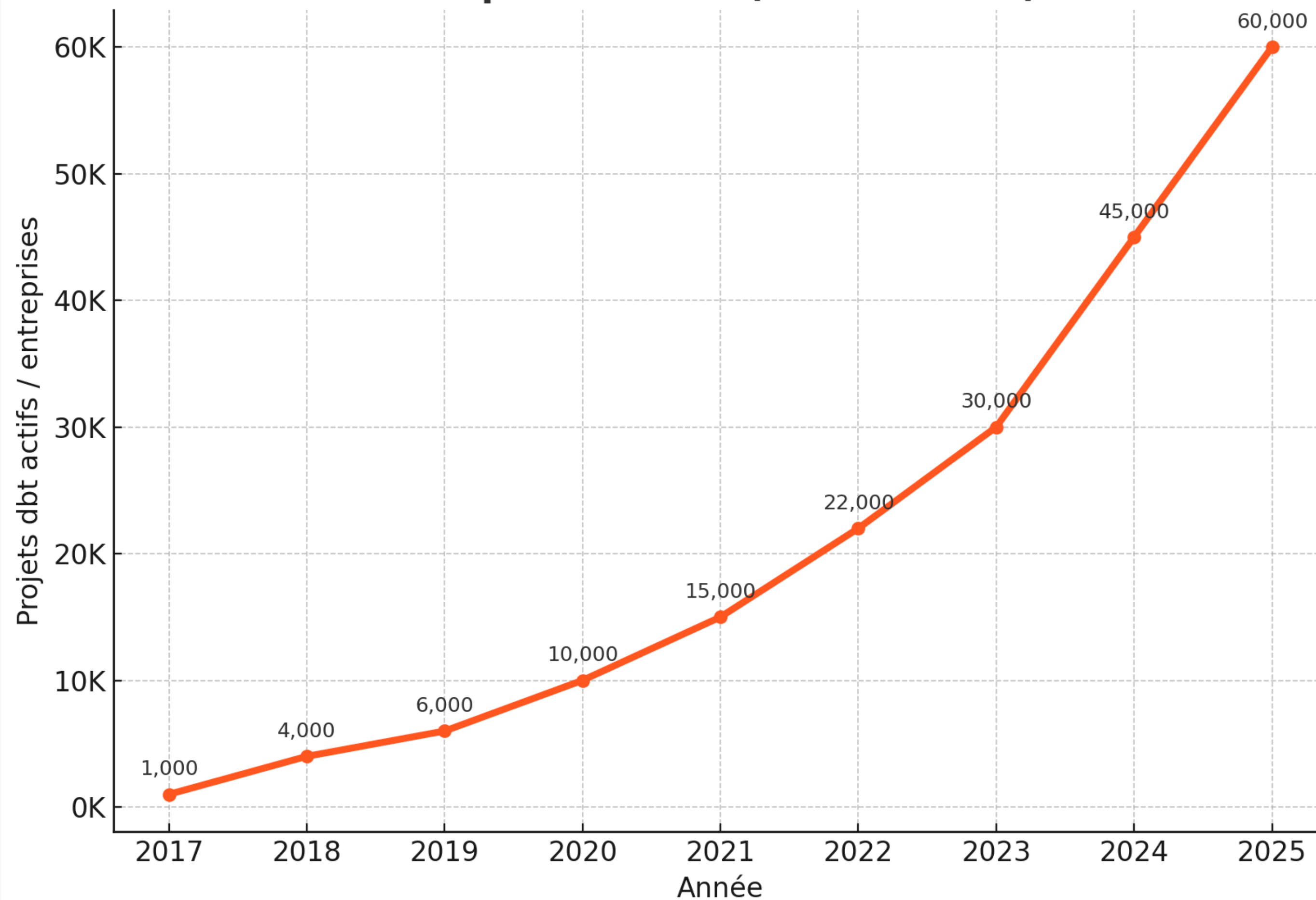
DBT prend en charge la phase de transformation (T) de l'ELT, mais ne propose pas de support pour les opérations d'extraction et de chargement. Il permet aux entreprises d'écrire des transformations sous forme de requêtes et de les orchestrer de manière plus efficace.

De nombreuses grandes entreprises comme **IBM, JetBlue, Dyson, Capgemini**, etc., utilisent cet outil. En **2025**, plus de **60 000** équipes à travers le monde utilisent DBT pour construire leur couche analytique sur les entrepôts de données modernes.



Adoption de dbt (2017 → 2025)


Source: dbt Labs + communauté dbt 2025






Pourquoi DBT est devenu un pilier de l'analytique des données

La popularité de DBT repose sur plusieurs facteurs clés :

- **Approche centrée sur SQL** : Les analystes de données, déjà familiers avec SQL, peuvent facilement écrire des transformations en SQL.
 - **Intégration avec DevOps** : DBT s'intègre aux systèmes de contrôle de version (comme Git) et prend en charge les tests automatisés et le déploiement, permettant aux équipes de tirer parti de leur expertise DevOps.
 - **Tests et documentation intégrés** : DBT garantit la qualité des données avec des tests unitaires et une documentation claire des modèles.
- 



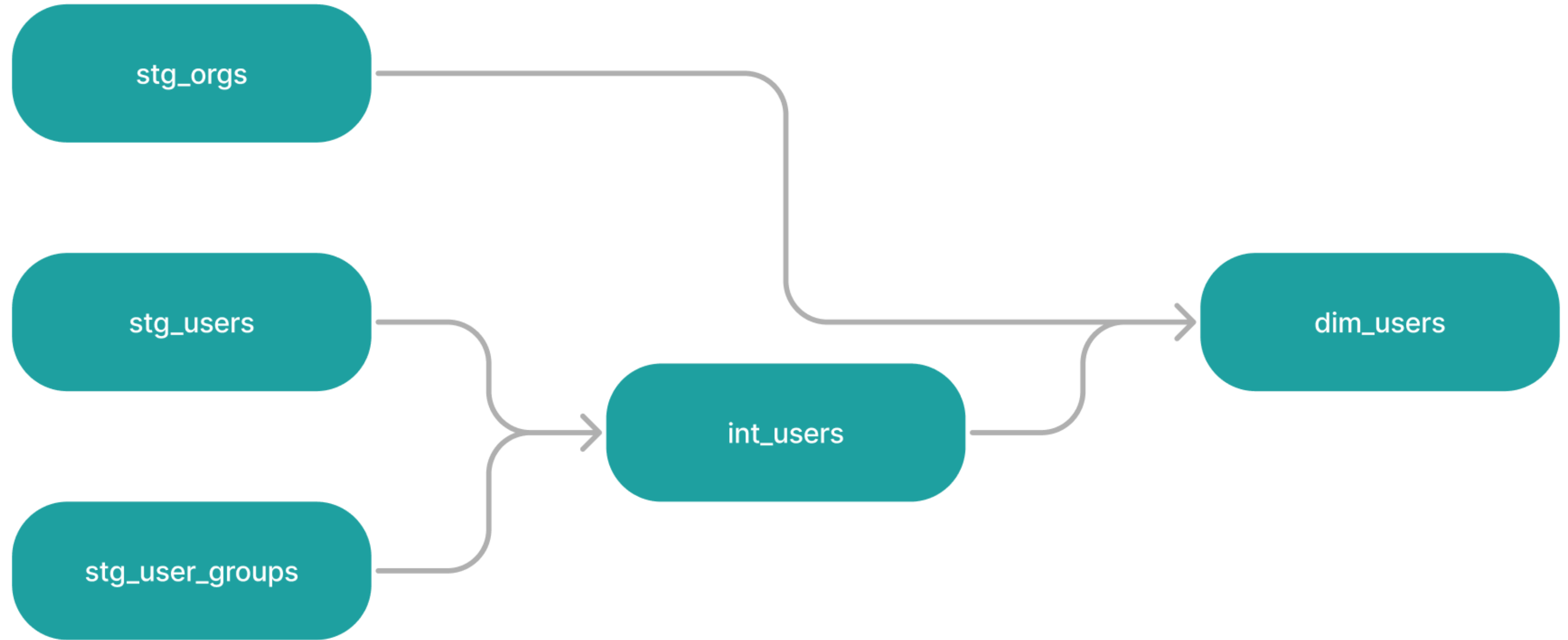
Fonctionnalités clés de DBT:

- **Transformations basées sur SQL:** DBT permet d'écrire des transformations en SQL, facilitant ainsi la collaboration entre analystes de données et ingénieurs tout en exploitant leurs compétences en SQL.
 - **Tests de données et documentation:** Avec DBT, vous pouvez écrire des tests pour valider la qualité de vos données et générer une documentation qui aide à comprendre et communiquer vos modèles de données.
 - **Code modulaire et réutilisable:** DBT soutient le code modulaire et réutilisable grâce à l'utilisation de macros et de modèles, permettant de construire et maintenir des transformations de données évolutives.
 - **Intégration avec le contrôle de version:** DBT s'intègre avec des systèmes de contrôle de version comme Git, vous permettant de suivre les modifications de vos modèles de données et de collaborer efficacement avec votre équipe.
- 

Fonctionnalités clés de DBT:

- **Data Lineage (DAG):** DBT génère automatiquement un graphe des dépendances (DAG) entre les modèles de données, ce qui permet une visualisation claire des relations entre les tables et les modèles. Grâce à ce Data Lineage, les équipes peuvent comprendre facilement les flux de transformation de données, **faciliter l'analyse d'impact** en cas de modification d'une source ou d'un modèle, et **améliorer la collaboration** entre data engineers et analysts. De plus, cette visualisation rend la maintenance des pipelines de données beaucoup **plus simple et efficace**.

Exemple de Data Lineage (DAG)



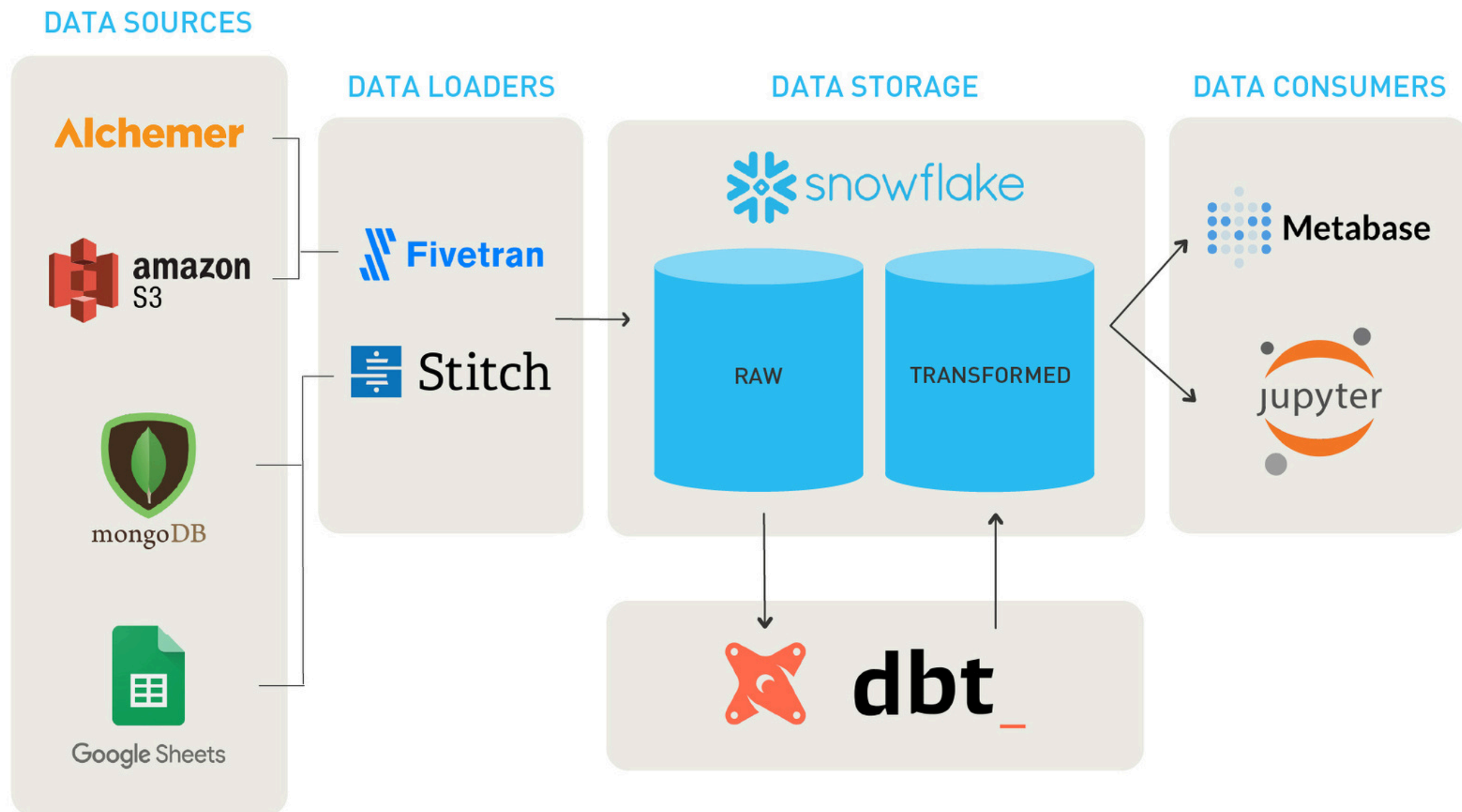
Fonctionnalités clés de DBT:

- **Materializations:** DBT permet de choisir comment chaque modèle SQL est matérialisé dans le data warehouse.

Types de Materializations :

- **view** : Vue logique, non persistée
- **table** : Table physique, requêtes plus rapides
- **incremental** : Table physique mise à jour partiellement (optimisé pour les gros volumes)
- **ephemeral** : Modèle temporaire (CTE), pas écrit dans le warehouse

Composants du Modern Data Stack



Environnement de DBT

Il existe essentiellement deux façons d'utiliser DBT : **DBT CLI** et **DBT Cloud**.

- **DBT Cloud (solution basée sur le cloud)** est une version hébergée qui simplifie le développement grâce à un environnement de développement intégré (IDE) en ligne et une interface permettant d'exécuter DBT selon un planning défini.
- **DBT Core** est un outil en **ligne de commande (CLI)** que l'on peut exécuter localement. Il s'agit de la version open source que vous pouvez installer et utiliser sur votre propre machine ou serveur.

DBT Cloud vs DBT Core

● DBT Core :

- Open source (licence Apache 2.0)
- Contient la logique principale de compilation SQL
- Utilise Jinja pour le templating
- Supporte des adaptateurs de base de données
- S'utilise via une interface en ligne de commande (CLI)
- La collaboration repose sur des outils externes (comme Git)
- L'utilisateur est responsable des mises à jour et de la maintenance

● DBT Cloud :

- SaaS (solution hébergée et gérée)
- IDE complet pour développer et tester votre code dbt
- Orchestration des jobs
- Journalisation (logs) et alertes
- Documentation intégrée
- Authentification des utilisateurs / SSO
- Fonctionnalités de collaboration intégrées
- Mises à jour et maintenance automatiques

Les avantages de DBT :

DBT se distingue par un ensemble d'avantages solides qui en font un outil de transformation de données très performant:

- **Approche centrée sur les modèles :** DBT permet de se concentrer sur la définition des transformations en SQL, sans avoir à concevoir des pipelines ETL complexes.
- **Environnement collaboratif :** Il facilite le travail en équipe grâce au partage et à la réutilisation du code, encourageant la collaboration entre les membres.
- **Intégration avec le contrôle de version :** DBT s'intègre parfaitement avec Git, ce qui facilite le développement collaboratif et le suivi des modifications.
- **Documentation et tests intégrés :** Les modèles DBT peuvent inclure une documentation claire directement dans le code, et DBT propose des tests intégrés pour garantir la qualité et la fiabilité des transformations.

Limites de l'outil DBT:

Même si DBT est puissant, il présente quelques inconvénients à connaître :

- **Pas de gestion de l'extraction et du chargement** : DBT ne prend en charge que la transformation (le "T" de ELT). Il faut donc utiliser d'autres outils pour extraire et charger les données dans l'entrepôt
- **Moins convivial pour les non-techniciens** : Étant basé sur SQL, DBT peut sembler moins accessible que des outils avec interface graphique (GUI).
- **Expertise technique nécessaire** : Modifier le code standard ou personnaliser DBT peut nécessiter un bon niveau technique.
- **Lisibilité à maintenir manuellement** : Pour garder les transformations compréhensibles dans l'interface, les ingénieurs doivent veiller à produire un code propre et bien structuré.

Comment démarrer avec DBT:

● Étape 1 : Installer DBT

- Créer un environnement virtuel : **python -m venv dbt_venv**
- Installer DBT avec ***pip*** (*le gestionnaire de paquets de Python*).
- Installer un adaptateur pour votre source de données (ex. Snowflake) : **pip install dbt-snowflake**.

● Étape 2 : Configurer votre projet

- Créer un dossier **.dbt** dans le répertoire utilisateur
- Initialiser le projet avec la commande : **dbt init nom_du_projet**

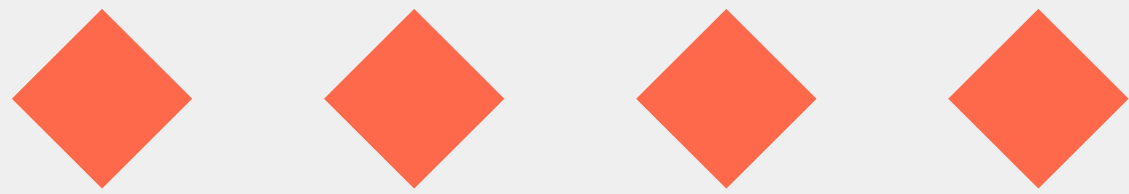
Comment démarrer avec DBT:

● Étape 3 : Définir vos modèles et transformations

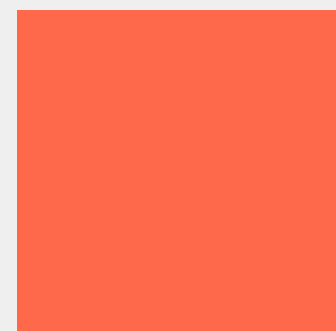
- Créez des fichiers SQL dans le dossier **models** pour chaque modèle ou transformation
- Écrivez la logique de transformation en **SQL**, avec la possibilité d'utiliser le **templating Jinja**

● Étape 4 : Exécuter les commandes dbt

- **dbt run** : exécute les transformations
- **dbt test** : lance les tests sur vos modèles
- **dbt docs generate** : génère la documentation du projet

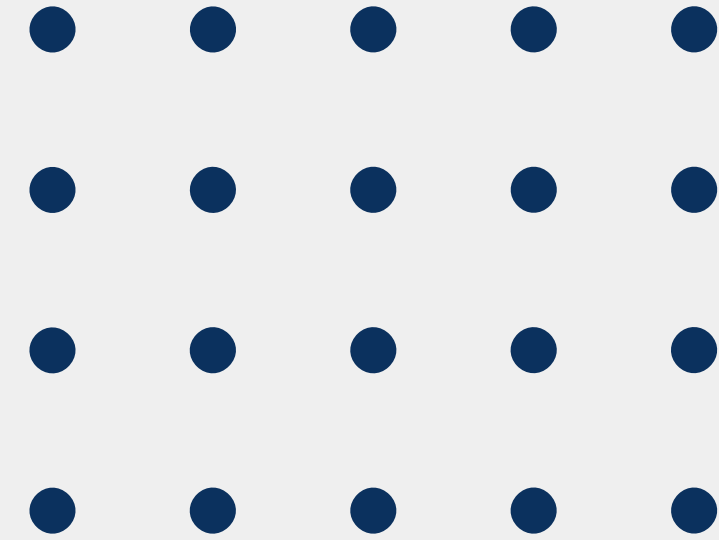


Implémentation de DBT





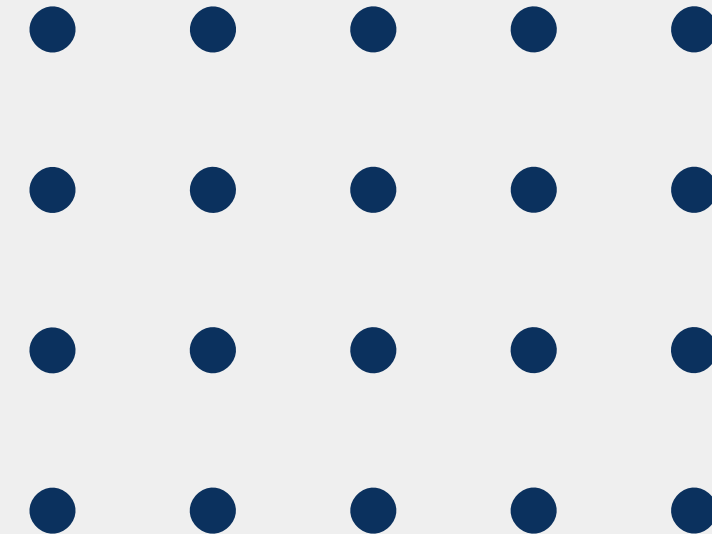
Conclusion



Dans un contexte où la donnée est au cœur de la prise de décision, **DBT** s'impose comme un outil moderne et efficace pour la transformation des données. Grâce à son approche basée sur SQL, son intégration avec les outils DevOps, et ses fonctionnalités de test, documentation et versioning, DBT permet aux équipes data de travailler de manière **plus collaborative, plus fiable** et **plus structurée**.

Bien qu'il nécessite une certaine expertise technique et s'appuie sur des outils tiers pour l'extraction et le chargement, ses avantages en matière de modularité, transparence et maintenabilité en font un incontournable dans l'univers de l'ELT moderne





Merci Pour
Votre
Attention !

