

# Challenge Stagiaire Data Engineer\_ Artefact CI

## À Propos d'Artefact

Artefact est un cabinet de conseil nouvelle génération, entièrement dédié à la **data** et à la **transformation AI**. Avec plus de **2000 talents présents dans 24 pays**, nous aidons les organisations à activer pleinement la puissance de leurs données pour accélérer leur performance.

Notre métier ? Transformer la donnée en impact business mesurable.

Nous intervenons sur toute la chaîne de valeur data dont:

- Automatisation des processus critiques grâce à l'IA
- Optimisation et personnalisation des expériences clients
- Stratégie data & marketing digital
- Pilotage média, measurement & analytics avancés
- Construction et déploiement de solutions data robustes et scalables

## Contexte

Dans le cadre du processus de recrutement pour un poste de **Data Engineer**, ce mini-projet vise à évaluer vos compétences techniques et méthodologiques autour des thématiques suivantes :

- Analyse de données métier
- Modélisation relationnelle et normalisation
- SQL (PostgreSQL)
- Conteneurisation avec Docker
- Développement de scripts d'ingestion de données en Python


Le projet repose sur un jeu de données représentant les **ventes d'un site e-commerce**, fourni en pièce jointe.

## Objectifs du test

Ce test a pour objectif d'évaluer votre capacité à :

- Analyser un jeu de données réel
- Concevoir un **modèle de données normalisé (3<sup>e</sup> forme normale)**
- Implémenter ce modèle dans PostgreSQL
- Déployer une base de données et un s3 storage via Docker
- Développer un script d'ingestion de données robuste et maintenable
- Déployer un Dag airflow
- Structurer un projet de manière professionnelle

## Durée estimée

 **14 Jours**

Vous êtes libre de gérer votre temps. La qualité, la clarté et la cohérence des livrables sont privilégiées par rapport à l'exhaustivité.

---

## Données en entrée

- Un fichier de données (fourni par la responsable People)
  - Ce fichier contient les ventes d'un site e-commerce
  - Le champ **sale\_date** permet d'identifier la date de vente et sera utilisé pour l'ingestion ciblée
- 

## Travaux demandés

### 1. Analyse exploratoire

- Examiner la structure du fichier fourni (colonnes, types, cardinalité)
  - Identifier les entités métier principales
  - Mettre en évidence les redondances ou anomalies potentielles
  - 💡 Une analyse formelle (documentée) est attendue (pdf ou notebook)
- 

### 2. Modélisation & normalisation

- Normaliser les données jusqu'à la **troisième forme normale (3FN)**
  - Identifier clairement :
    - o les tables
    - o les clés primaires
    - o les clés étrangères
  - Justifier brièvement vos choix de modélisation
  - Normaliser les données jusqu'à la **DKNF (Domain-Key Normal Form)**
  - Identifier clairement :
    - o les tables
    - o les clés primaires
    - o les clés étrangères
  - Justifier brièvement vos choix de modélisation
  - Dans notre cas, dire quel est l'intérêt d'aller jusqu'à la normalisation DKNF ?
-

### 3. Implémentation SQL (PostgreSQL)

- Écrire les requêtes CREATE TABLE correspondant aux tables issues de la DKNF
  - Les scripts doivent être **compatibles PostgreSQL**
  - Le choix du schéma est libre (ex : public, sales, e-commerce, etc.)
  - Ecrire les requêtes permettant de créer une vue qui fait une modélisation en Étoile des données à partir des tables issues de la **DKNF**
- 

### 4. Déploiement PostgreSQL & Minio avec Docker

#### ♦ *Instructions*

- Déployer une base **PostgreSQL** à l'aide de **Docker**
- Déployer une instance **Minio** à l'aide de **Docker**
- L'utilisation de **Docker Compose** est fortement recommandée
- Uploader le jeu de données dans un bucket Minio que vous nommerez **"folder\_source"**

#### ♦ *Contraintes techniques*

- Les tables issues de la **DKNF** ainsi que la vue doivent automatiquement se créer au déploiement de la base PostgreSQL dans Docker
- 

### 5. Script Python d'ingestion

Mettre en place un script Python répondant aux exigences suivantes :

#### ♦ *Fonctionnalités attendues*

- Le script prend en **argument une date** au format YYYYMMDD
- Il lit le fichier source de ventes se trouvant dans Minio
- Il filtre les lignes dont le champ **sale\_date** correspond à la date fournie en paramètre
- Il alimente les tables PostgreSQL normalisées

#### ♦ *Contraintes techniques*

- Le script doit donner le même résultat pour la même entrée (idempotence)
- Gestion minimale des erreurs :
  - o format de date invalide
  - o connexion à la base
  - o insertion des données
- Le script doit être relançable sans provoquer d'incohérences majeures
- Logging dans le script Python

#### ♦ *Exemple d'exécution*

```
python main.py 20250616
```

---

## 6. Dag airflow

- Déployer une instance de Airflow 3.x avec Docker
  - Mettre en place un Dag qui fait la même chose que le script d'ingestion
  - ♦ *Contraintes techniques*
    - Utiliser la fonctionnalité **Connexion** de Airflow pour la configuration des différentes connexions à Postgres et à Minio
    - Utiliser la fonctionnalité **Variable** de Airflow (si applicable)
- 

## Technologies attendues

- **Stockage** : PostgreSQL, Minio
- **Conteneurisation** : Docker / Docker Compose
- **Langage** : Python3, SQL
- **Orchestration** : Airflow

## Livrables attendus

- Un repo github contenant :
    - Le notebook ou le pdf de l'analyse exploratoire
    - les scripts SQL
    - le fichier docker-compose.yml
    - les scripts Python d'ingestion
    - un fichier README.md
  - Le projet doit être exécutable localement via des instructions claires
- 

## Critères d'évaluation

Le candidat sera évalué sur :

- Qualité de l'analyse des données (forme et fond)
  - Pertinence et clarté de la modélisation (3FN)
  - Qualité du SQL (types, contraintes, relations)
  - Maîtrise de Docker et Docker Compose
  - Qualité du code Python (lisibilité, robustesse, structure)
  - Organisation globale du repo
  - Capacité à justifier ses choix techniques
  - Clarté et facilité de compréhension du README.md
-

## Bonus (optionnel)

- Tests simples ou données de validation
- 

## Remarques finales

- Les hypothèses peuvent être formulées si certaines informations sont manquantes
- La clarté et la justification des choix priment sur la complexité
- Tout élément permettant de démontrer une démarche professionnelle sera apprécié