

AI Factory - dbt Silver Layer Documentation

AI Factory - Documentation dbt Silver Layer
Guide de Transformation et Detection de Fraude
Table des Matieres

1. Introduction
 - 1.1 Objectif
 - 1.2 Volumes de Donnees
2. Architecture dbt
 - 2.1 Structure du Projet
 - 2.2 Flux de Donnees (DAG)
 - 2.3 Materialisation
3. Macros et Packages
 - 3.1 Qu'est-ce qu'une Macro dbt ?
 - 3.2 Syntaxe des Macros
 - 3.3 Package dbt_utils
 - 3.4 Macros Utilisees dans le Projet
 - 3.5 Macros dbt Natives
 - 3.6 Exemple Complet
 - 3.7 Creer ses Propres Macros
 - 3.8 Resume des Avantages des Macros
4. Couche Staging
 - 3.1 Role
 - 3.2 Sources Declarees
 - 3.3 Exemple de Modele Staging
5. Couche Silver
 - 5.1 Transformations Appliquees
 - 5.2 Cles de Jointure
6. Flags de Detection de Fraude
 - 6.1 Vue d'Ensemble
 - 6.2 Flags par Modele
7. Modeles Silver en Detail
 - 7.1 silver_provider
 - 7.2 silver_excluded_providers
 - 7.3 silver_payments
 - 7.4 silver_prescriptions
 - 7.5 silver_facilities
 - 7.6 silver_hospital_spending
8. Tests et Qualite
 - 8.1 Tests dbt Implementes
 - 8.2 Execution des Tests
 - 8.3 Freshness des Sources
9. Utilisation

- 9.1 Commandes dbt
 - 9.2 Requetes d'Analyse de Fraude
 - 9.3 Documentation Web
- Annexe: Glossaire des Flags

AI Factory - Documentation dbt Silver Layer

Guide de Transformation et Detection de Fraude

Version: 1.0.0 **Date:** Fevrier 2026

Table des Matieres

1. [Introduction](#)
 2. [Architecture dbt](#)
 3. [Macros et Packages](#)
 4. [Couche Staging](#)
 5. [Couche Silver](#)
 6. [Flags de Detection de Fraude](#)
 7. [Modeles Silver en Detail](#)
 8. [Tests et Qualite](#)
 9. [Utilisation](#)
-

1. Introduction

1.1 Objectif

La couche Silver de dbt transforme les donnees brutes (Bronze) en donnees propres, typees et enrichies, pretes pour l'analyse de fraude medicale. Elle ajoute des indicateurs de risque et des metriques calculees.

1.2 Volumes de Donnees

Table Silver	Lignes	Description
silver_provider	17,897,600	Provider master enrichi
silver_excluded_providers	90,316	Providers exclus LEIE
silver_payments	16,158,446	Paielements Open Payments
silver_prescriptions	1,332,309	Prescriptions Part D
silver_facilities	19,221	Etablissements de sante

Table Silver	Lignes	Description
silver_hospital_spending	64,658	Depenses hospitalieres
TOTAL	35,562,550	

2. Architecture dbt

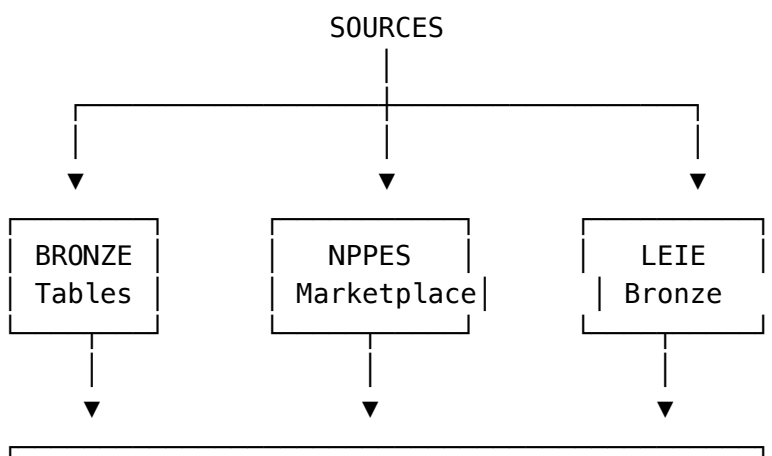
2.1 Structure du Projet

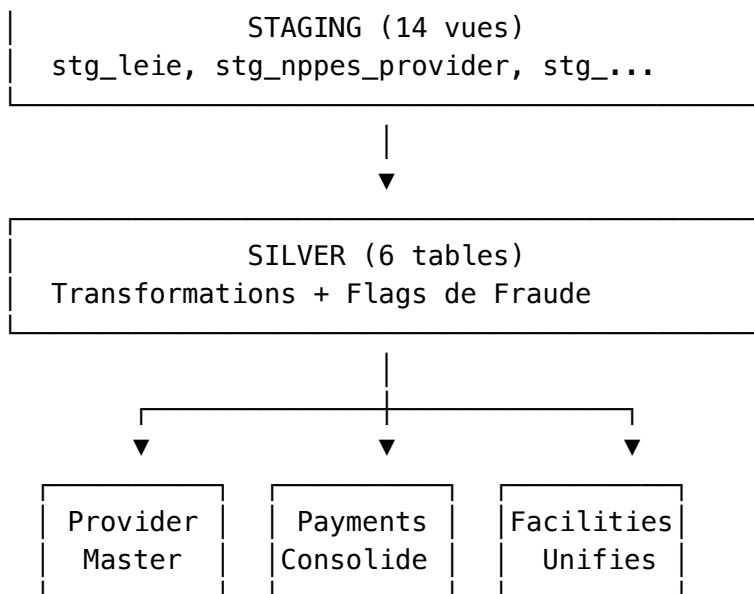
```

dbt/
├── dbt_project.yml          # Configuration projet
├── profiles.yml            # Connexion Snowflake
├── packages.yml            # Dependencies (dbt_utils)
├── models/
│   ├── staging/            # Vues 1:1 sur Bronze + NPPES
│   │   ├── sources.yml     # Declaration des sources
│   │   ├── schema.yml     # Documentation staging
│   │   └── stg_*.sql       # 14 modeles staging
│   └── silver/             # Tables transformees
│       ├── schema.yml     # Tests et documentation
│       ├── silver_provider.sql
│       ├── silver_excluded_providers.sql
│       ├── silver_payments.sql
│       ├── silver_prescriptions.sql
│       ├── silver_facilities.sql
│       └── silver_hospital_spending.sql
└── target/                # Fichiers generes
    ├── manifest.json
    ├── catalog.json
    └── index.html          # Documentation web

```

2.2 Flux de Donnees (DAG)





2.3 Materialisation

Couche	Type	Raison
Staging	VIEW	Pas de stockage, lecture directe
Silver	TABLE	Performance, donnees transformees
Gold	TABLE	Agregations pour dashboards

3. Macros et Packages

3.1 Qu'est-ce qu'une Macro dbt ?

Une **macro** dbt est une fonction réutilisable écrite en Jinja2 (langage de template Python). Elle permet de :

- **Eviter la répétition** de code SQL
- **Standardiser** les transformations
- **Paramétrer** des opérations complexes
- **Abstraire** la logique métier

Les macros sont l'équivalent des fonctions dans un langage de programmation.

3.2 Syntaxe des Macros

Definition d'une macro:

```
{% macro nom_macro(parametre1, parametre2='valeur_defaut') %}
  -- Code SQL genere par la macro
  SELECT {{ parametre1 }} FROM {{ parametre2 }}
{% endmacro %}
```

Appel d'une macro:

```
{{ nom_macro('colonne', 'table') }}
```

3.3 Package dbt_utils

Nous utilisons le package **dbt_utils** (maintenu par dbt Labs) qui fournit des macros pre-construites et testées.

Installation (packages.yml):

```
packages:
  - package: dbt-labs/dbt_utils
    version: ">=1.0.0"
```

Commande d'installation:

```
dbt deps --profiles-dir .
```

3.4 Macros Utilisees dans le Projet

generate_surrogate_key

Objectif: Generer une cle primaire unique et deterministe a partir de plusieurs colonnes.

Pourquoi l'utiliser: - Les tables source n'ont pas toujours de cle primaire - On veut une cle stable qui ne change pas entre les executions - Evite les doublons lors des unions de tables

Syntaxe:

```
{{ dbt_utils.generate_surrogate_key(['colonne1', 'colonne2',
                                     'colonne3']) }}
```

Exemple concret dans silver_payments:

```
-- Genere un hash MD5 unique pour chaque paiement
{{ dbt_utils.generate_surrogate_key(['PAYMENT_CATEGORY', 'RECORD_ID'])
  }} as PAYMENT_ID
```

Ce que ca genere:

```
MD5(CAST(COALESCE(CAST(PAYMENT_CATEGORY AS VARCHAR),
                  '_dbt_utils_surrogate_key_null_')
  || '-' || COALESCE(CAST(RECORD_ID AS VARCHAR),
                  '_dbt_utils_surrogate_key_null_') AS VARCHAR))
```

Utilisations dans le projet:

Modele	Colonnes	Resultat
silver_payments	PAYMENT_CATEGORY, RECORD_ID	PAYMENT_ID

Modele	Colonnes	Resultat
silver_prescriptions	NPI, BRND_NAME, GNRC_NAME	PRESCRIPTION_ID
silver_excluded_providers	NPI, EXCLDATE, LASTNAME, FIRSTNAME	EXCLUSION_ID
silver_facilities	FACILITY_TYPE, CCN	FACILITY_ID
silver_hospital_spending	FACILITY_ID, CLAIM_TYPE, PERIOD	SPENDING_ID

3.5 Macros dbt Natives

dbt fournit aussi des macros natives (sans package):

ref()

Objectif: Referencier un autre modele dbt.

```
-- Reference le modele stg_leie
SELECT * FROM {{ ref('stg_leie') }}
```

Avantages: - Cree automatiquement les dependances (DAG) - Adapte le nom du schema selon l'environnement - Permet de visualiser le lineage

source()

Objectif: Referencier une table source externe.

```
-- Reference la table LEIE dans le schema BRONZE
SELECT * FROM {{ source('bronze', 'LEIE') }}
```

Avantages: - Documente les sources externes - Permet de tester la freshness des donnees - Isole les modeles des changements de schema

config()

Objectif: Configurer le comportement d'un modele.

```
{{
  config(
    materialized='table',      -- table ou view
    unique_key='payment_id',   -- cle pour incremental
    schema='silver'           -- schema cible
  )
}}
```

3.6 Exemple Complet

Voici comment les macros sont utilisees dans `silver_payments.sql`:

```

{{
    config(
        materialized='table',          -- Macro config()
        unique_key='payment_id'
    )
}}

with general_payments as (
    select * from {{ ref('stg_open_payments_general') }} -- Macro
        ref()
),

final as (
    select
        -- Macro generate_surrogate_key()
        {{ dbt_utils.generate_surrogate_key(['PAYMENT_CATEGORY',
        'RECORD_ID']) }} as PAYMENT_ID,
        ...
    from all_payments
)

select * from final

```

3.7 Créer ses Propres Macros

Pour des transformations répétitives, on peut créer des macros personnalisées dans dbt/macros/.

Exemple - Macro de nettoyage de texte:

```

-- macros/clean_text.sql
{% macro clean_text(column_name) %}
    UPPER(TRIM(COALESCE({{ column_name }}, '')))
{% endmacro %}

```

Utilisation:

```

SELECT
    {{ clean_text('LASTNAME') }} as LAST_NAME,
    {{ clean_text('FIRSTNAME') }} as FIRST_NAME
FROM source

```

3.8 Résumé des Avantages des Macros

Avantage	Description
DRY	Don't Repeat Yourself - code réutilisable
Maintenabilité	Un seul endroit à modifier
Lisibilité	Code SQL plus clair
Tests	Macros testées par la communauté
Abstraction	Cache la complexité SQL

4. Couche Staging

3.1 Role

Les modeles staging sont des **vues 1:1** sur les tables sources. Ils servent a : - Isoler les modeles Silver des changements de schema Bronze - Ajouter un timestamp de chargement (`_loaded_at`) - Documenter les sources

3.2 Sources Declarees

Bronze (AI_FACTORY_DB.BRONZE): - LEIE -
 MEDICARE_HOSPITAL_SPENDING - OPEN_PAYMENTS_GENERAL -
 OPEN_PAYMENTS_RESEARCH - OPEN_PAYMENTS_OWNERSHIP -
 PROVIDER_INFORMATION - LONGTERM_CARE_HOSPITAL - HOSPICE -
 HOME_HEALTH_CARE - MEDICARE_PART_D_PRESCRIBERS

NPPES Marketplace (AFFINE_NPPES_PROVIDER_DATA.REF_DW): -
 DIM_PROVIDER - DIM_PROVIDER_ADDRESS - DIM_PROVIDER_TAXONOMY -
 REF_TAXONOMY_CODE

3.3 Exemple de Modele Staging

```
-- models/staging/stg_leie.sql
with source as (
    select * from {{ source('bronze', 'LEIE') }}
)

select
    *,
    current_timestamp() as _loaded_at
from source
```

5. Couche Silver

5.1 Transformations Appliquees

Transformation	Description
Typage	VARCHAR vers DATE, NUMBER, BOOLEAN
Nettoyage	Trim, Upper, gestion des NULLs
Deduplication	ROW_NUMBER() pour garder la derniere version
Enrichissement	Jointures NPI, calculs de metriques
Standardisation	Noms de colonnes en snake_case
Flags de fraude	Indicateurs de risque booleens

5.2 Cles de Jointure

Le **NPI (National Provider Identifier)** est la cle principale pour joindre :

NPPES.DIM_PROVIDER.NPI

- LEIE.NPI
- OPEN_PAYMENTS.Covered_Recipient_NPI
- MEDICARE_PART_D.PRSCRBR_NPI
- PROVIDER_INFORMATION (via CCN)

6. Flags de Detection de Fraude

6.1 Vue d'Ensemble

Les modeles Silver incluent des **indicateurs booléens** et des **niveaux de risque** pour faciliter la detection de fraude.

6.2 Flags par Modele

silver_provider

Flag	Type	Description	Logique
IS_EXCLUDED	BOOLEAN	Provider sur liste d'exclusion LEIE	Jointure NPI avec LEIE
IS_NPI_ACTIVE	BOOLEAN	NPI toujours actif dans NPPES	Champ IS_ACTIVE de NPPES

Cas de fraude potentielle: Un provider avec IS_EXCLUDED = true et IS_NPI_ACTIVE = true pourrait tenter de facturer malgre son exclusion.

silver_excluded_providers

Flag	Type	Description	Logique
IS_CURRENTLY_EXCLUDED	BOOLEAN	Exclusion toujours active	REINDATE est NULL
IS_HIGH_RISK	BOOLEAN	Exclu mais NPI encore actif	IS_CURRENTLY_EXCLUDED AND NPPES_IS_ACTIVE
MATCH_STATUS	STRING	Qualite du match NPI	NPI_MATCH, NO_NPI_IN_LEIE, NPI_NOT_FOUND

Interpretation de IS_HIGH_RISK:

IS_HIGH_RISK = TRUE signifie:

- Le provider est actuellement exclu de Medicare/Medicaid
- MAIS son NPI est toujours actif dans le registre NPDES
- Risque: peut potentiellement facturer sous ce NPI

silver_payments

Flag	Type	Description	Logique
IS_RECIPIENT_EXCLUDED	BOOLEAN	Destinataire du paiement est exclu	Jointure NPI avec LEIE
PAYMENT_RISK_TIER	STRING	Niveau de risque du paiement	Base sur le montant

Niveaux de risque des paiements:

PAYMENT_RISK_TIER	Montant	Description
VERY_HIGH	>= \$100,000	Paiements tres eleves, audit prioritaire
HIGH	>= \$10,000	Paiements significatifs
MEDIUM	>= \$1,000	Paiements moderees
LOW	>= \$100	Paiements faibles
MINIMAL	< \$100	Paiements minimales

Cas de fraude potentielle: Un provider avec IS_RECIPIENT_EXCLUDED = true recoit des paiements de l'industrie pharmaceutique alors qu'il est exclu.

silver_prescriptions

Flag	Type	Description	Logique
IS_PRESCRIBER_EXCLUDED	BOOLEAN	Prescripteur est exclu	Jointure NPI avec LEIE
IS_HIGH_VOLUME_PRESCRIBER	BOOLEAN	Volume anormal	TOTAL_CLAIMS > 100
IS_HIGH_COST_PRESCRIBER	BOOLEAN	Cout moyen eleve	AVG_COST_PER_CLA
IS_HIGH_CLAIMS_PER_BENE	BOOLEAN	Trop de claims par patient	AVG_CLAIMS_PER_BI > 10
RISK_TIER	STRING	Niveau de risque global	Combine plusieurs facteu

Niveaux de risque des prescripteurs:

RISK_TIER	Condition	Description
CRITICAL	IS_PRESCRIBER_EXCLUDED = true	Prescripteur exclu mais prescrit
VERY_HIGH	TOTAL_DRUG_COST > \$1M	Volume de prescription tres eleve
HIGH	TOTAL_DRUG_COST > \$100K	Volume significatif
MEDIUM	TOTAL_DRUG_COST > \$10K	Volume modere
LOW	Autre	Risque faible

Metriques de fraude calculees:

Metrique	Formule	Seuil suspect
AVG_COST_PER_CLAIM	TOTAL_DRUG_COST / TOTAL_CLAIMS	> \$500
AVG_COST_PER_BENEFICIARY	TOTAL_DRUG_COST / TOTAL_BENEFICIARIES	> \$1000
AVG_CLAIMS_PER_BENEFICIARY	TOTAL_CLAIMS / TOTAL_BENEFICIARIES	> 10

silver_facilities

Flag	Type	Description	Logique
IS_LOW_QUALITY	BOOLEAN	Etablissement de mauvaise qualite	OVERALL_RATING = 1
OCCUPANCY_RATE	NUMBER	Taux d'occupation	RESIDENTS / BEDS * 100
OWNERSHIP_CATEGORY	STRING	Type de propriete	FOR_PROFIT, NON_PROFIT, GOVERNMENT

Correlation fraude/qualite: Les etablissements a but lucratif avec une note de 1 etoile sont statistiquement plus a risque de fraude.

silver_hospital_spending

Flag	Type	Description	Logique
IS_HIGH_SPENDER	BOOLEAN	Depenses anormalement elevees	PCT_VARIANCE_FR > 50%
SPENDING_RISK_TIER	STRING	Niveau de risque depenses	Base sur variance natio

Flag	Type	Description	Logique
STATE_BENCHMARK_STATUS	STRING	Position vs moyenne etat	BELOW, AT, ABOVE_
NATIONAL_BENCHMARK_STATUS	STRING	Position vs moyenne nationale	BELOW, AT, ABOVE_

Niveaux de risque des depenses:

SPENDING_RISK_TIER	Variance	Description
HIGH	> +50%	Tres au-dessus de la moyenne nationale
MEDIUM	+20% a +50%	Au-dessus de la moyenne
NORMAL	-20% a +20%	Dans la moyenne
LOW	< -20%	En dessous de la moyenne

7. Modeles Silver en Detail

7.1 silver_provider

Objectif: Table de reference principale des providers, enrichie avec NPPES et status d'exclusion.

Sources: - stg_nppes_provider (principal) - stg_nppes_provider_address (adresse primaire) - stg_nppes_provider_taxonomy (specialite primaire) - stg_leie (status exclusion)

Transformations:

```
-- Jointure avec adresses (primaires uniquement)
left join nppes_address a on p.NPI = a.NPI
where a.IS_PRIMARY = true

-- Jointure avec taxonomie (primaire uniquement)
left join nppes_taxonomy t on p.NPI = t.NPI
where t.IS_PRIMARY_TAXONOMY = true

-- Flag d'exclusion
case when e.NPI is not null then true else false end as IS_EXCLUDED
```

Colonnes cles: - NPI (Primary Key) - ENTITY_TYPE (Individual/Organization) - FULL_NAME, ORGANIZATION_NAME - STATE, CITY, ZIP_CODE - SPECIALTY, PROVIDER_TYPE - IS_NPI_ACTIVE, IS_EXCLUDED

7.2 silver_excluded_providers

Objectif: Liste enrichie des providers exclus, avec indicateurs de risque.

Sources: - stg_leie (principal) - stg_nppes_provider (enrichissement) - stg_nppes_provider_address (adresse)

Transformations:

```
-- Parsing des dates d'exclusion
try_to_date(EXCLDATE, 'YYYYMMDD') as EXCLUSION_DATE

-- Mapping des codes d'exclusion
case EXCLTYPE
  when '1128a1' then 'Conviction of program-related crimes'
  when '1128a2' then 'Conviction relating to patient abuse'
  when '1128a3' then 'Felony conviction relating to health care
    fraud'
  ...
end as EXCLUSION_REASON

-- Flag de risque eleve
case
  when IS_CURRENTLY_EXCLUDED = true AND NPPES_IS_ACTIVE = true
  then true else false
end as IS_HIGH_RISK
```

Types d'exclusion (EXCLTYPE):

Code	Description
1128a1	Conviction pour crimes lies au programme
1128a2	Conviction pour maltraitance de patient
1128a3	Crime lie a la fraude medicale
1128a4	Crime lie aux substances controlees
1128b4	Revocation/suspension de licence
1128b7	Fraude, pots-de-vin, activites interdites

7.3 silver_payments

Objectif: Consolider les 3 types de paiements Open Payments en une seule table.

Sources: - stg_open_payments_general (paiements generaux) - stg_open_payments_research (paiements recherche) - stg_open_payments_ownership (participations) - stg_leie (status exclusion)

Transformations:

```
-- Union des 3 sources
select 'GENERAL' as PAYMENT_CATEGORY, ... from
  stg_open_payments_general
union all
select 'RESEARCH' as PAYMENT_CATEGORY, ... from
  stg_open_payments_research
union all
```

```

select 'OWNERSHIP' as PAYMENT_CATEGORY, ... from
    stg_open_payments_ownership

-- Typage des montants
try_to_number("Total_Amount_of_Payment_USDollars", 18, 2) as
    PAYMENT_AMOUNT

-- Parsing des dates
try_to_date("Date_of_Payment", 'MM/DD/YYYY') as PAYMENT_DATE

-- Niveau de risque
case
    when PAYMENT_AMOUNT >= 100000 then 'VERY_HIGH'
    when PAYMENT_AMOUNT >= 10000 then 'HIGH'
    ...
end as PAYMENT_RISK_TIER

```

Note technique: Les colonnes Open Payments sont case-sensitive (creees via INFER_SCHEMA). Utiliser les guillemets doubles: "Record_ID".

7.4 silver_prescriptions

Objectif: Nettoyer les donnees Part D et ajouter des metriques de detection de fraude.

Sources: - stg_medicare_part_d_prescribers - stg_leie (status exclusion)

Metriques calculees:

```

-- Cout moyen par prescription
TOTAL_DRUG_COST / TOTAL_CLAIMS as AVG_COST_PER_CLAIM

-- Cout moyen par beneficiaire
TOTAL_DRUG_COST / TOTAL_BENEFICIARIES as AVG_COST_PER_BENEFICIARY

-- Prescriptions par beneficiaire
TOTAL_CLAIMS / TOTAL_BENEFICIARIES as AVG_CLAIMS_PER_BENEFICIARY

-- Duree moyenne par renouvellement
TOTAL_DAYS_SUPPLY / TOTAL_30DAY_FILLS as AVG_DAYS_PER_FILL

```

Flags de fraude:

```

-- Volume anormal
case when TOTAL_CLAIMS > 1000 then true else false end as
    IS_HIGH_VOLUME_PRESCRIBER

-- Cout anormal
case when AVG_COST_PER_CLAIM > 500 then true else false end as
    IS_HIGH_COST_PRESCRIBER

-- Trop de prescriptions par patient
case when AVG_CLAIMS_PER_BENEFICIARY > 10 then true else false end as
    IS_HIGH_CLAIMS_PER_BENE

```

7.5 silver_facilities

Objectif: Unifier les 4 types d'établissements en une seule table.

Sources: - stg_hospice - stg_home_health_care - stg_longterm_care_hospital - stg_provider_information (nursing homes)

Transformations:

```
-- Union des sources
select 'HOSPICE' as FACILITY_TYPE, ... from stg_hospice
union all
select 'HOME_HEALTH' as FACILITY_TYPE, ... from stg_home_health_care
union all
select 'LTCH' as FACILITY_TYPE, ... from stg_longterm_care_hospital
union all
select 'NURSING_HOME' as FACILITY_TYPE, ... from
    stg_provider_information

-- Deduplication par CCN
row_number() over (partition by CCN order by _LOAD_TIMESTAMP desc) as
    rn
where rn = 1

-- Categorisation ownership
case
    when OWNERSHIP_TYPE ilike '%profit%' and not ilike '%non%' then
        'FOR_PROFIT'
    when OWNERSHIP_TYPE ilike '%non%profit%' then 'NON_PROFIT'
    when OWNERSHIP_TYPE ilike '%government%' then 'GOVERNMENT'
    else 'OTHER'
end as OWNERSHIP_CATEGORY

-- Taux d'occupation
NUMBER_OF_RESIDENTS / NUMBER_OF_BEDS * 100 as OCCUPANCY_RATE
```

7.6 silver_hospital_spending

Objectif: Nettoyer les données de dépenses et calculer les écarts aux benchmarks.

Sources: - stg_medicare_hospital_spending

Transformations:

```
-- Typage des montants
try_to_number(AVG_SPENDING_PER_EPISODE_HOSPITAL, 18, 2) as
    AVG_SPENDING_HOSPITAL

-- Variance vs benchmark national
(AVG_SPENDING_HOSPITAL - AVG_SPENDING_NATION) / AVG_SPENDING_NATION *
    100
as PCT_VARIANCE_FROM_NATION
```

```
-- Status benchmark
case
  when AVG_SPENDING_HOSPITAL > AVG_SPENDING_NATION * 1.2 then
    'ABOVE_NATIONAL_AVG_20PCT'
  when AVG_SPENDING_HOSPITAL < AVG_SPENDING_NATION then
    'BELOW_NATIONAL_AVG'
  else 'AT_NATIONAL_AVG'
end as NATIONAL_BENCHMARK_STATUS
```

8. Tests et Qualite

8.1 Tests dbt Implementes

```
# models/silver/schema.yml
```

```
models:
  - name: silver_provider
    columns:
      - name: NPI
        tests:
          - unique
          - not_null
      - name: ENTITY_TYPE
        tests:
          - accepted_values:
              values: ['Individual', 'Organization', 'Unknown']

  - name: silver_payments
    columns:
      - name: PAYMENT_ID
        tests:
          - unique
          - not_null
      - name: PAYMENT_CATEGORY
        tests:
          - accepted_values:
              values: ['GENERAL', 'RESEARCH', 'OWNERSHIP']
```

8.2 Execution des Tests

```
# Executer tous les tests
```

```
dbt test --profiles-dir .
```

```
# Tester un modele specifique
```

```
dbt test --profiles-dir . --select silver_provider
```


8.3 Freshness des Sources

```
# sources.yml
sources:
  - name: bronze
    freshness:
      warn_after: {count: 24, period: hour}
      error_after: {count: 48, period: hour}
```

9. Utilisation

9.1 Commandes dbt

```
# Depuis le container Docker
docker exec ai_factory-airflow-worker-1 bash -c "cd /opt/airflow/dbt
&& dbt <command> --profiles-dir ."
```

```
# Installer les packages
dbt deps
```

```
# Compiler les modeles
dbt compile
```

```
# Executer tous les modeles
dbt run
```

```
# Executer Silver uniquement
dbt run --select silver
```

```
# Executer un modele specifique
dbt run --select silver_provider
```

```
# Generer la documentation
dbt docs generate
```

```
# Servir la documentation
dbt docs serve --port 8085 --host 0.0.0.0
```

9.2 Requetes d'Analyse de Fraude

Providers exclus a haut risque:

```
SELECT
  NPI,
  LAST_NAME,
  FIRST_NAME,
  EXCLUSION_REASON,
  EXCLUSION_DATE
```

```

FROM AI_FACTORY_DB.SILVER_silver.SILVER_EXCLUDED_PROVIDERS
WHERE IS_HIGH_RISK = TRUE
ORDER BY EXCLUSION_DATE DESC;

```

Paielements vers providers exclus:

```

SELECT
    p.NPI,
    p.FULL_NAME,
    p.PAYER_NAME,
    SUM(p.PAYMENT_AMOUNT) as TOTAL_PAYMENTS,
    COUNT(*) as NUM_PAYMENTS
FROM AI_FACTORY_DB.SILVER_silver.SILVER_PAYMENTS p
WHERE p.IS_RECIPIENT_EXCLUDED = TRUE
GROUP BY p.NPI, p.FULL_NAME, p.PAYER_NAME
ORDER BY TOTAL_PAYMENTS DESC;

```

Prescripteurs a risque critique:

```

SELECT
    NPI,
    PRESCRIBER_FULL_NAME,
    TOTAL_DRUG_COST,
    TOTAL_CLAIMS,
    AVG_COST_PER_CLAIM,
    RISK_TIER
FROM AI_FACTORY_DB.SILVER_silver.SILVER_PRESCRIPTIONS
WHERE RISK_TIER = 'CRITICAL'
ORDER BY TOTAL_DRUG_COST DESC;

```

Hopitaux avec depenses anormales:

```

SELECT
    HOSPITAL_NAME,
    STATE,
    AVG_SPENDING_HOSPITAL,
    AVG_SPENDING_NATION,
    PCT_VARIANCE_FROM_NATION,
    SPENDING_RISK_TIER
FROM AI_FACTORY_DB.SILVER_silver.SILVER_HOSPITAL_SPENDING
WHERE IS_HIGH_SPENDER = TRUE
ORDER BY PCT_VARIANCE_FROM_NATION DESC;

```

9.3 Documentation Web

La documentation interactive est disponible a l'adresse: **<http://localhost:8085>**

Elle inclut: - Graphe de lineage interactif - Description de tous les modeles -
Documentation des colonnes - Tests definis

Annexe: Glossaire des Flags

Flag	Modele	Type	Descripti
IS_EXCLUDED	silver_provider	BOOL	Provide liste LE
IS_NPI_ACTIVE	silver_provider	BOOL	NPI acti dans NF
IS_CURRENTLY_EXCLUDED	silver_excluded_providers	BOOL	Exclusi non lev
IS_HIGH_RISK	silver_excluded_providers	BOOL	Exclu n NPI acti
IS_RECIPIENT_EXCLUDED	silver_payments	BOOL	Destina exclu
PAYMENT_RISK_TIER	silver_payments	STRING	Risque l sur mon
IS_PRESCRIBER_EXCLUDED	silver_prescriptions	BOOL	Prescrip exclu
IS_HIGH_VOLUME_PRESCRIBER	silver_prescriptions	BOOL	Volume 1000 cl
IS_HIGH_COST_PRESCRIBER	silver_prescriptions	BOOL	Cout > \$500/cl
IS_HIGH_CLAIMS_PER_BENE	silver_prescriptions	BOOL	> 10 claims/p
RISK_TIER	silver_prescriptions	STRING	Risque l prescrip
IS_LOW_QUALITY	silver_facilities	BOOL	Rating = etoile
IS_HIGH_SPENDER	silver_hospital_spending	BOOL	> 50% d dessus moyenn
SPENDING_RISK_TIER	silver_hospital_spending	STRING	Risque depense

Document genere pour AI Factory - dbt Silver Layer v1.0.0