FraudLens - Snowflake

Documentation de l'infrastructure Data Warehouse

Vue d'ensemble

Snowflake est le Data Warehouse central du projet FraudLens. Il stocke et traite les donnees de sante Medicare/Medicaid selon une architecture Medallion (Bronze → Silver → Gold).

Architecture Snowflake

```
FRAUDLENS_DB (Database)
    │
    ├── RAW_DATA (Schema)
    │       └── Données brutes directement depuis S3
    │
    ├── BRONZE (Schema)
    │       ├── LEIE
    │       ├── MEDICARE_HOSPITAL_SPENDING
    │       ├── PROVIDER_INFORMATION
    │       ├── LONGTERM_CARE_HOSPITAL
    │       ├── HOSPICE
    │       ├── HOME_HEALTH_CARE
    │       ├── MEDICARE_PART_D_PRESCRIBERS
    │       ├── OPEN_PAYMENTS_GENERAL
    │       ├── OPEN_PAYMENTS_RESEARCH
    │       └── OPEN_PAYMENTS_OWNERSHIP
    │
    ├── STAGING (Schema)
    │       └── Vues dbt sur les tables Bronze
    │
    ├── SILVER (Schema)
    │       └── Tables transformées et enrichies
    │
    └── GOLD (Schema)
            ├── PROVIDER_360
```

```
├── PAYMENTS_SUMMARY
├── PRESCRIPTIONS_SUMMARY
├── FRAUD_RISK_SCORE
└── HIGH_RISK_ALERTS
```

Objets Snowflake

Warehouse

Parametre	Valeur
Nom	`FRAUDLENS_WH`
Taille	XSMALL
Type	STANDARD
Auto Suspend	300 secondes (5 min)
Auto Resume	TRUE
Initially Suspended	TRUE

```
CREATE OR REPLACE WAREHOUSE FRAUDLENS_WH
  WAREHOUSE_SIZE = 'XSMALL'
  WAREHOUSE_TYPE = 'STANDARD'
  AUTO_SUSPEND = 300
  AUTO_RESUME = TRUE
  INITIALLY_SUSPENDED = TRUE;
```

Database & Schemas

Schema	Description	Usage

RAW_DATA	Donnees brutes depuis S3	Landing zone initiale
BRONZE	Donnees apres ingestion	Tables Parquet chargees
STAGING	Staging area dbt	Vues sur Bronze
SILVER	Donnees transformees	Tables nettoyees, jointes
GOLD	Donnees finales	Tables pour BI et analyse

Integration S3

> Les donnees sont stockees sur Amazon S3 et chargees dans Snowflake via un Stage externe.

Storage Integration

Une Storage Integration est configuree pour permettre a Snowflake d'acceder au bucket S3 :

```
CREATE OR REPLACE STORAGE INTEGRATION S3_INTEGRATION
  TYPE = EXTERNAL_STAGE
  STORAGE_PROVIDER = 'S3'
  ENABLED = TRUE
  STORAGE_AWS_ROLE_ARN = 'arn:aws:iam::XXXX:role/snowflake-role'
  STORAGE_ALLOWED_LOCATIONS = ('s3://ai-factory-bckt/');
```

External Stage

```
CREATE OR REPLACE STAGE BRONZE_S3_STAGE
  STORAGE_INTEGRATION = S3_INTEGRATION
  URL = 's3://ai-factory-bckt/bronze/'
  FILE_FORMAT = PARQUET_FORMAT;
```

File Format

```
CREATE OR REPLACE FILE FORMAT PARQUET_FORMAT
  TYPE = PARQUET
  COMPRESSION = AUTO;
```

Tables Bronze

Les tables Bronze contiennent les donnees brutes chargees depuis les fichiers Parquet sur S3.

1. LEIE - Excluded Individuals/Entities

Colonne	Type	Description
NPI	VARCHAR	National Provider Identifier
LASTNAME, FIRSTNAME	VARCHAR	Nom du provider
BUSNAME	VARCHAR	Nom de l'entreprise
SPECIALTY	VARCHAR	Specialite medicale
EXCLTYPE	VARCHAR	Type d'exclusion
EXCLDATE	VARCHAR	Date d'exclusion
STATE	VARCHAR	Etat
_LOAD_TIMESTAMP	TIMESTAMP_NTZ	Timestamp de chargement

2. MEDICARE_HOSPITAL_SPENDING

Colonne	Type	Description
FACILITY_ID	VARCHAR	Identifiant de l'etablissement

FACILITY_NAME	VARCHAR	Nom de l'etablissement
CLAIM_TYPE	VARCHAR	Type de reclamation
AVG_SPENDING_PER_EPISODE_HOSPITAL	VARCHAR	Depense moyenne par episode
STATE	VARCHAR	Etat

3. MEDICARE_PART_D_PRESCRIBERS

Colonne	Type	Description
PRSCRBR_NPI	VARCHAR	NPI du prescripteur
PRSCRBR_LAST_ORG_NAME	VARCHAR	Nom du prescripteur
BRND_NAME	VARCHAR	Nom de marque du medicament
GNRC_NAME	VARCHAR	Nom generique
TOT_CLMS	VARCHAR	Total des reclamations
TOT_DRUG_CST	VARCHAR	Cout total des medicaments

4. OPEN_PAYMENTS (3 tables)

- `OPEN_PAYMENTS_GENERAL` - Paiements generaux
- `OPEN_PAYMENTS_RESEARCH` - Paiements de recherche
- `OPEN_PAYMENTS_OWNERSHIP` - Interets de propriete

5. Autres tables

- `PROVIDER_INFORMATION` - Informations sur les nursing homes
- `LONGTERM_CARE_HOSPITAL` - Hopitaux de soins prolonges
- `HOSPICE` - Etablissements de soins palliatifs

- `HOME_HEALTH_CARE` - Soins a domicile

Chargement des Donnees

> Le chargement est orchestre par Apache Airflow via le DAG `load_bronze_tables`.

Commande COPY INTO

```
COPY INTO FRAUDLENS_DB.BRONZE.LEIE
FROM @BRONZE_S3_STAGE/leie/
FILE_FORMAT = PARQUET_FORMAT
MATCH_BY_COLUMN_NAME = CASE_INSENSITIVE
ON_ERROR = CONTINUE;
```

Options de chargement

Option	Valeur	Description
MATCH_BY_COLUMN_NAME	CASE_INSENSITIVE	Mapping automatique des colonnes
ON_ERROR	CONTINUE	Continuer en cas d'erreur
FILE_FORMAT	PARQUET_FORMAT	Format Parquet avec compression auto

Donnees Externes (Marketplace)

> Le referentiel NPPES est disponible via le Snowflake Marketplace (Affine).

Source	Database	Description
Affine NPPES	`AFFINE__NPPES__PROVIDER_DATA`	Registre national de tous les NPI aux USA

Authentification

Methode: Cle Privee RSA

L'authentification utilise une cle privee RSA (fichier `.p8`) pour une securite renforcee.

> **Important:** Les fichiers de cle privee (`*.p8`) ne doivent jamais etre commites dans le repository Git.

Configuration

1. Generer une paire de cles RSA
2. Configurer la cle publique dans Snowflake
3. Stocker la cle privee dans `snowflake/keys/`
4. Configurer le chemin dans `profiles.yml`

```
# ~/.dbt/profiles.yml
fraudlens:
  target: prod
  outputs:
    prod:
      type: snowflake
      account: rdsnbdu-gbc86569
      user: FISHER
      private_key_path: ~/.snowflake/rsa_key.p8
      role: ACCOUNTADMIN
      database: FRAUDLENS_DB
      warehouse: FRAUDLENS_WH
      schema: STAGING
      threads: 4
```

Scripts SQL

Les scripts SQL d'initialisation sont dans le dossier `snowflake/` :

Fichier	Description

`init_warehouse.sql`	Creation du warehouse FRAUDLENS_WH
`init_schemas.sql`	Creation de la database et des schemas
`init_s3_stage.sql`	Creation du stage S3 et file format
`bronze/create_tables.sql`	DDL des tables Bronze
`bronze/load_tables.sql`	Commandes COPY INTO
`bronze/truncate_tables.sql`	Truncate des tables Bronze

Bonnes Pratiques

1. Gestion des Couts

- Utiliser `AUTO_SUSPEND = 300` pour suspendre le warehouse apres 5 min d'inactivite
- Utiliser la taille `XSMALL` pour les workloads legers
- Monitorer les credits consommes regulierement

2. Securite

- Utiliser l'authentification par cle privee (pas de mot de passe)
- Ne jamais commiter les fichiers `.p8`
- Utiliser des roles avec le minimum de privileges necessaires

3. Performance

- Utiliser le format Parquet pour les fichiers sources
- Activer le clustering sur les grandes tables si necessaire
- Utiliser `MATCH_BY_COLUMN_NAME` pour simplifier le chargement

Commandes Utiles

Verifier le statut du warehouse

```
SHOW WAREHOUSES LIKE 'FRAUDLENS_WH';
```

Lister les tables Bronze

```
SHOW TABLES IN FRAUDLENS_DB.BRONZE;
```

Verifier les fichiers dans le stage

```
LIST @BRONZE_S3_STAGE;
```

Verifier l'historique de chargement

```
SELECT * FROM TABLE(INFORMATION_SCHEMA.COPY_HISTORY(
  TABLE_NAME => 'LEIE',
  START_TIME => DATEADD(hours, -24, CURRENT_TIMESTAMP())
));
```

FraudLens - Snowflake Documentation

MooM | 2026