FraudLens - dbt Documentation

Overview

FraudLens uses dbt (data build tool) to transform raw healthcare data into analytics-ready tables following the **Medallion Architecture** (Bronze → Silver → Gold).

Project Structure

```
dbt/
├── models/
|   ├── staging/       # Views on Bronze layer
|   ├── silver/        # Cleaned & enriched tables
|   └── gold/          # Business-ready aggregations
├── tests/             # Custom SQL tests
├── macros/            # Reusable SQL snippets
└── dbt_project.yml    # Project configuration
```

Data Layers

Staging Layer (Views)

Pass-through views on Bronze tables. No transformations.

Model	Source	Description
`stg_leie`	LEIE	Excluded providers list
`stg_medicare_hospital_spending`	MEDICARE_HOSPITAL_SPENDING	Hospital spending data
`stg_provider_information`	PROVIDER_INFORMATION	Nursing home providers
`stg_longterm_care_hospital`	LONGTERM_CARE_HOSPITAL	LTCH facilities
`stg_hospice`	HOSPICE	Hospice providers

Model	Source	Description
`stg_home_health_care`	HOME_HEALTH_CARE	Home health agencies
`stg_medicare_part_d_prescribers`	MEDICARE_PART_D_PRESCRIBERS	Part D prescriptions
`stg_open_payments_general`	OPEN_PAYMENTS_GENERAL	General pharma payments
`stg_open_payments_research`	OPEN_PAYMENTS_RESEARCH	Research payments
`stg_open_payments_ownership`	OPEN_PAYMENTS_OWNERSHIP	Ownership interests
`stg_nppes_provider`	DIM_PROVIDER (Marketplace)	NPI registry
`stg_nppes_provider_address`	DIM_PROVIDER_ADDRESS	Provider addresses
`stg_nppes_provider_taxonomy`	DIM_PROVIDER_TAXONOMY	Provider specialties

Silver Layer (Incremental Tables)

Cleaned, enriched, and deduplicated data with fraud indicators.

Model	Materialization	Description
`provider`	table	Master provider table with NPPES data
`excluded_providers`	table	LEIE exclusions enriched with NPPES
`payments`	table	Consolidated Open Payments
`prescriptions`	**incremental**	Medicare Part D with fraud flags
`facilities`	**incremental**	Unified facilities
`hospital_spending`	**incremental**	Hospital spending with benchmarks

Gold Layer (Tables)

Business-ready aggregations for analytics and dashboards.

Model	Description
`provider_360`	Complete provider profile (single source of truth)

Model	Description
`fraud_risk_score`	Composite fraud risk score (0-100 scale)
`high_risk_alerts`	Actionable alerts for investigation
`payments_summary`	Aggregated payment metrics by provider
`prescriptions_summary`	Aggregated prescription metrics by provider

Incremental Strategy

Silver models use `_LOAD_TIMESTAMP` to process only new records:

```
{% if is_incremental() %}
{% set max_loaded_at = run_query("select max(_loaded_at) from " ~
this).columns[0].values()[0] %}
{% endif %}

...
WHERE ...
{% if is_incremental() %}
  and _LOAD_TIMESTAMP > '{{ max_loaded_at }}'
{% endif %}
```

> **Note:** Snowflake doesn't support correlated subqueries with MAX() in WHERE clauses. We use `run_query` to fetch the timestamp before SQL execution.

Tests

Test Coverage: 80 tests

Layer	Test Types
Staging	Descriptions only (pass-through views)
Silver	`unique` , `not_null` , `accepted_values`

Layer	Test Types
Gold	`unique`, `not_null`, `accepted_values`, `relationships`, `accepted_range`

Schema Tests Example

```
- name: fraud_risk_score
  columns:
    - name: NPI
      tests:
        - unique
        - not_null
        - relationships:
            to: ref('provider_360')
            field: NPI
    - name: FRAUD_RISK_SCORE
      tests:
        - dbt_utils.accepted_range:
            min_value: 0
            max_value: 100
```

Custom SQL Tests

Test	Purpose
`assert_critical_risk_providers_are_excluded`	Validates risk scoring logic
`assert_no_negative_financial_amounts`	Data integrity check
`assert_all_alerts_have_valid_npi`	Referential integrity
`assert_financial_exposure_calculation`	Calculation validation

Risk Scoring Methodology

The `fraud_risk_score` model calculates a 0-100 score:

Risk Factor	Max Points
Exclusion risks (LEIE listed, still active)	40

Risk Factor	Max Points
Payment risks (high recipient tier, high-risk %)	20
Prescription risks (brand %, high-risk drugs)	20
Activity anomalies (inactive NPI with activity)	20

Risk Tiers

Score	Tier
70+	CRITICAL
50-69	HIGH
30-49	MEDIUM
10-29	LOW
0-9	MINIMAL

Alert Types

The `high_risk_alerts` model generates actionable alerts:

Alert Type	Priority	Description
`EXCLUDED_STILL_ACTIVE`	1	Provider excluded but still active in NPPES
`PRESCRIPTION_BY_EXCLUDED`	2	Excluded provider prescribed medications
`PAYMENT_TO_EXCLUDED`	3	Excluded provider received pharma payments
`HIGH_RISK_SCORE`	4	Provider has CRITICAL/HIGH risk score
`HIGH_BRAND_PRESCRIBER`	5	80%+ brand prescription rate

Running dbt

```
# Run all models
dbt run

# Run specific layer
dbt run --select silver.*

# Run tests
dbt test

# Full refresh (rebuild incremental)
dbt run --full-refresh --select prescriptions

# Generate documentation
dbt docs generate && dbt docs serve
```

CI/CD Integration

GitHub Actions workflow (`.github/workflows/dbt_deploy.yml`):

1. Triggered on push to `main` (dbt/ changes)
2. Runs `dbt run` to update models
3. Runs `dbt test` to validate data quality
4. Uses Snowflake private key authentication

Dependencies

- **dbt-snowflake**: 1.7.0
- **dbt-utils**: For `generate_surrogate_key` , `accepted_range`

Best Practices

1. **Incremental models** for large Silver tables (prescriptions, facilities, hospital_spending)
2. **Table models** for Gold aggregations (need full recalculation)
3. **Deduplication** with `qualify row_number() over (...) = 1`
4. **Referential integrity** via `relationships` tests

5. **Range validation** for calculated scores and amounts