6. Seeds and Sources

Working with Seeds

Seeds in DBT are CSV files that you can load into your data warehouse. They're useful for static reference data like country codes, ZIP code mappings, or product categories.

Creating a Seed

1. Place a CSV file in the `seeds` directory:

`seeds/country_codes.csv`:

```
country_code,country_name
US,United States
CA,Canada
UK,United Kingdom
FR,France
DE,Germany
```

2. Load the seed into your data warehouse:

bash

```bash
dbt seed
```

3. Reference the seed in your models:

sql

```sql
SELECT
    orders.*,
    countries.country_name
FROM {{ ref('stg_orders') }} orders
JOIN {{ ref('country_codes') }} countries ON orders.country_code =
countries.country_code
```

Configuring Seeds

You can configure seeds in your `dbt_project.yml`:

yaml

```yaml
seeds:
  my_project:
    country_codes:
      +column_types:
        country_code: varchar(2)
        country_name: varchar(50)
    zip_codes:
      +quote_columns: true
```

Configuring Sources

Sources in DBT represent the raw data loaded into your data warehouse. Configuring sources provides several benefits:

- Documentation of where your data comes from
- Testing capabilities for raw data
- The ability to check if sources are "fresh"

Defining Sources

Sources are defined in YAML files. Typically, you'd create a file like `models/sources.yml`:

yaml

```yaml
version: 2

sources:
  - name: raw_data
    database: production_db
    schema: raw_data
    tables:
      - name: customers
        description: Raw customer data from the CRM
        columns:
          - name: customer_id
            description: Primary key for customers
            tests:
              - unique
              - not_null
        loaded_at_field: updated_at
        freshness:
          warn_after: {count: 12, period: hour}
          error_after: {count: 24, period: hour}

      - name: orders
        description: Raw order data from the ordering system
        columns:
          - name: order_id
            description: Primary key for orders
            tests:
              - unique
              - not_null
          - name: customer_id
            description: Foreign key to customers
            tests:
              - relationships:
                  to: source('raw_data', 'customers')
                  field: customer_id
```

Using Sources in Models

You can reference sources in your models using the `source()` function:

sql

```sql
SELECT
    customer_id,
    first_name,
    last_name,
    email
FROM {{ source('raw_data', 'customers') }}
WHERE created_at > '2021-01-01'
```

Checking Source Freshness

You can check if your sources are up-to-date:

bash

```
dbt source freshness
```

This will check all sources with freshness configurations and report whether they're fresh, stale (warning), or old (error).