9. Testing

Types of Tests

DBT supports two main types of tests:

1. **Generic tests**: Reusable tests that can be applied to multiple models
2. **Singular tests**: Custom SQL queries that test specific conditions

Generic Tests

Generic tests are reusable assertions about your models. DBT comes with four built-in generic tests:

1. **Unique**: Tests that values in a column are unique
2. **Not Null**: Tests that a column contains no null values
3. **Relationships**: Tests referential integrity between tables
4. **Accepted Values**: Tests that values in a column are from a defined set

Configuring Generic Tests

Generic tests are typically defined in YAML files alongside your models:

yaml

```yaml
version: 2

models:
  - name: stg_customers
    description: Staged customer data
    columns:
      - name: customer_id
        description: Primary key
        tests:
          - unique
          - not_null
      - name: email
        description: Customer email address
```

```
        tests:
          - unique
          - not_null
      - name: status
        description: Customer status
        tests:
          - accepted_values:
              values: ['active', 'inactive', 'pending']

  - name: stg_orders
    description: Staged order data
    columns:
      - name: order_id
        description: Primary key
        tests:
          - unique
          - not_null
      - name: customer_id
        description: Foreign key to customers
        tests:
          - not_null
          - relationships:
              to: ref('stg_customers')
              field: customer_id
```

Singular Tests

Singular tests are custom SQL queries that return rows that fail the test. An empty result set means the test passes.

Creating a Singular Test

1. Create a file in the `tests` directory, e.g., `tests/order_amount_is_positive.sql`:

sql

```
-- This test ensures that all order amounts are positive
SELECT
    order_id,
    amount
FROM {{ ref('stg_orders') }}
WHERE amount <= 0
```

2. Any records returned by this query represent test failures.

Test Selection and Configuration

Running Tests

To run all tests:

bash

```
dbt test
```

To run specific tests:

bash

```
# Run tests for a specific model
dbt test --models stg_customers

# Run only the unique tests
dbt test --select test_type:unique

# Run specific test
dbt test --select test_name:positive_amounts
```

Test Configuration

You can configure test behavior in several ways:

1. **Severity**: Set tests to either `error` (default) or `warn`:

yaml

```
columns:
  - name: status
    tests:
      - accepted_values:
          values: ['active', 'inactive', 'pending']
          severity: warn
```

2. **Thresholds**: Allow a certain number or percentage of failures:

yaml

```yaml
columns:
  - name: email
    tests:
      - unique:
          config:
            error_if: ">10"  # Fail if more than 10 duplicates
            warn_if: ">5"    # Warn if more than 5 duplicates
```

3. **Where Clause**: Add filters to your tests:

yaml

```yaml
columns:
  - name: customer_id
    tests:
      - unique:
          config:
            where: "status = 'active'"  # Only test active customers
```