8. Snapshots

What are Snapshots?

Snapshots in DBT are a built-in implementation of Type 2 Slowly Changing Dimensions. They record the state of a mutable table over time by:

- Tracking changes to specified columns
- Adding effective dates (`dbt_valid_from` and `dbt_valid_to`)
- Maintaining a complete history of changes

Snapshot Strategies

DBT offers two main strategies for snapshots:

1. Timestamp Strategy

This strategy uses a timestamp column to identify changes:

sql

```
{% snapshot customers_snapshot %}

{{
    config(
      target_schema='snapshots',
      unique_key='customer_id',
      strategy='timestamp',
      updated_at='updated_at',
    )
}}

SELECT * FROM {{ source('raw', 'customers') }}

{% endsnapshot %}
```

Running Snapshots

To run all snapshots in your project:

bash

```
dbt snapshot
```

This command will:

1. Check for changes since the last snapshot run
2. Create new records for any changed data
3. Update the `dbt_valid_to` field for outdated records

Managing Snapshot Data

When snapshots are run, DBT creates or updates a table with the following structure:

```
customer_id | first_name | last_name | email | ... | dbt_valid_from |
dbt_valid_to | dbt_scd_id | dbt_updated_at
-----------+------------+-----------+-------+-----+----------------+------
-------+-----------+---------------
123         | John       | Doe       | j@d.co| ... | 2023-01-01     | 2023-
02-15  | abc123    | 2023-01-01
123         | John       | Smith     | j@d.co| ... | 2023-02-15     | null
| def456    | 2023-02-15
```

The special columns added by DBT:

- **dbt_valid_from**: When this version became active
- **dbt_valid_to**: When this version was superseded (null for current)
- **dbt_scd_id**: Unique identifier for each version
- **dbt_updated_at**: When this snapshot was taken

Querying Snapshot Data

To get the current state of all customers:

sql

```
SELECT * FROM {{ ref('customers_snapshot') }}
```

```
WHERE dbt_valid_to IS NULL
```

To get the state at a specific point in time:

sql

```
SELECT * FROM {{ ref('customers_snapshot') }}
WHERE '2023-03-15' BETWEEN dbt_valid_from AND COALESCE(dbt_valid_to,
'9999-12-31')
```

To see the history of changes for a specific customer:

sql

```
SELECT * FROM {{ ref('customers_snapshot') }}
WHERE customer_id = 123
ORDER BY dbt_valid_from
```