tobiko

Evolving Data Pipelines at Scale

January 15, 2024

Copyright © 2024 Tobiko Data, Inc.

About Me

- Co-Founder & Chief Architect at Tobiko Data
- Leading the development of SQLMesh
- Over 10 years of experience in data / ML infra
- Previously Netflix, Apple

Why am I here?

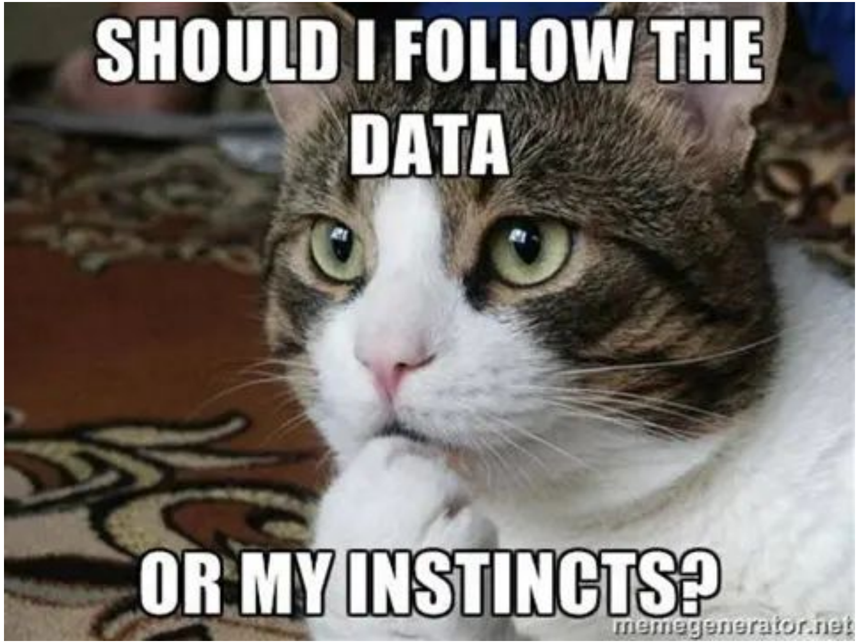
- Creating data environments for development is cumbersome

Why am I here?

- Populating dev environments with data is inefficient

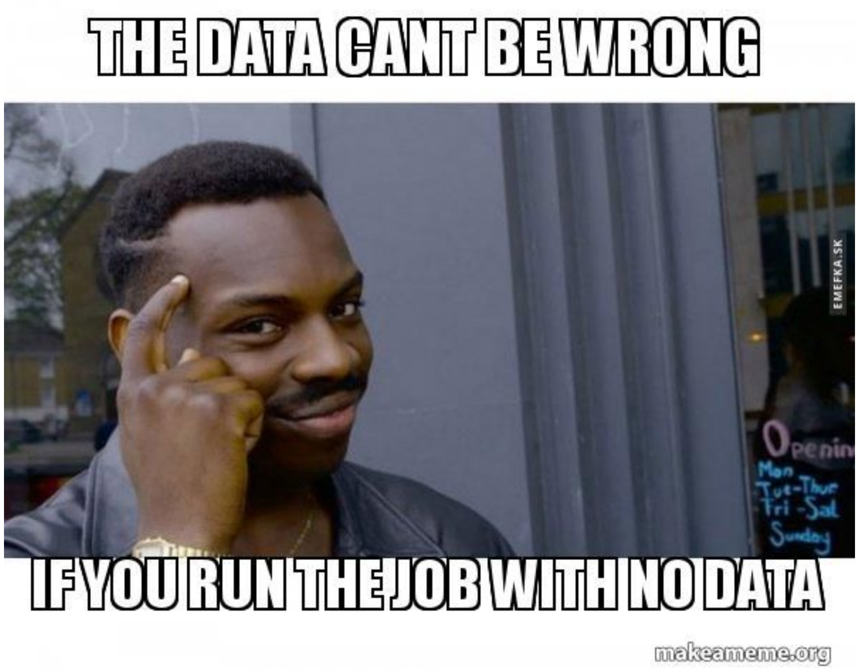
Why am I here?

- Development iterations are slow



Why am I here?

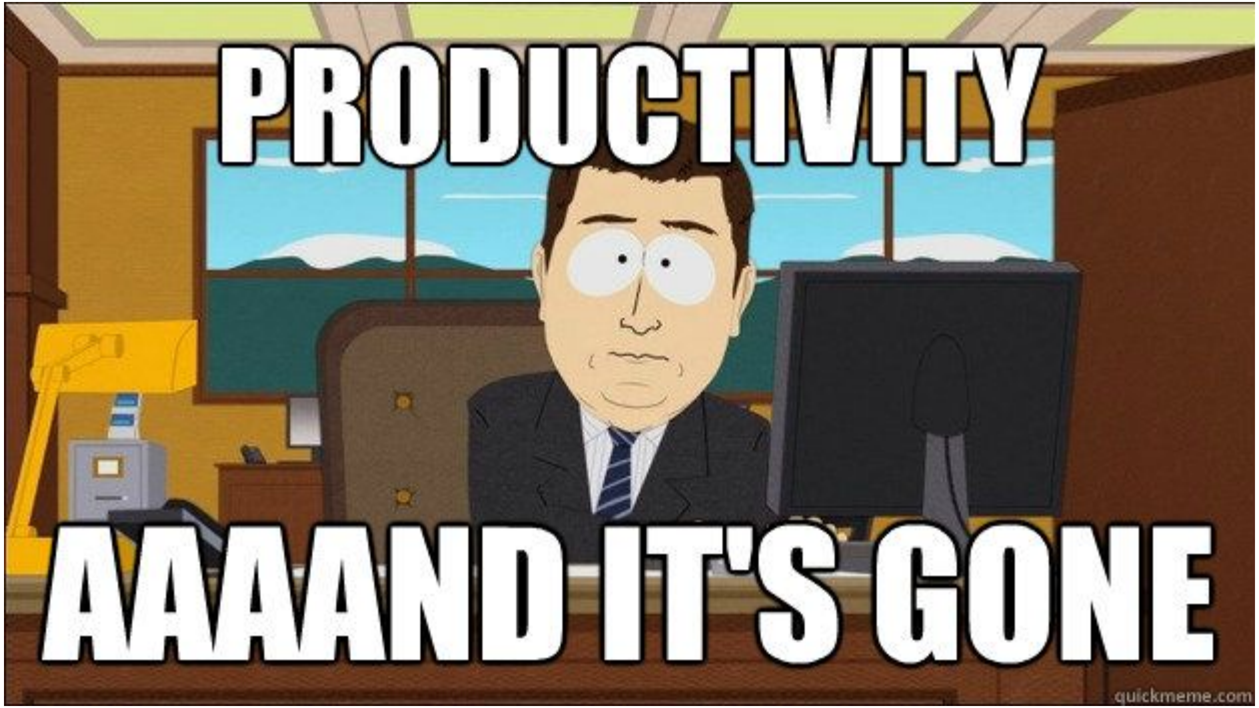
- Development iterations are slow



Why am I here?

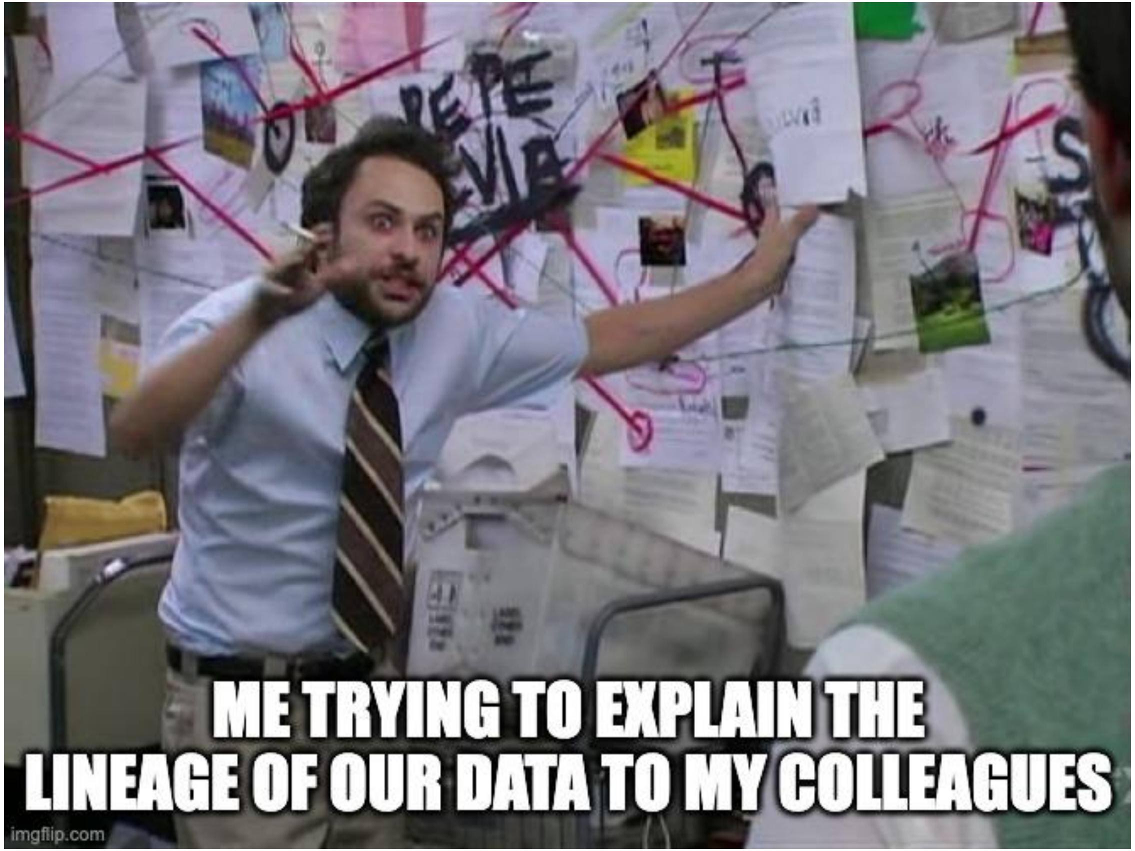
- Production deployments are ad-hoc and error-prone

Why am I here?

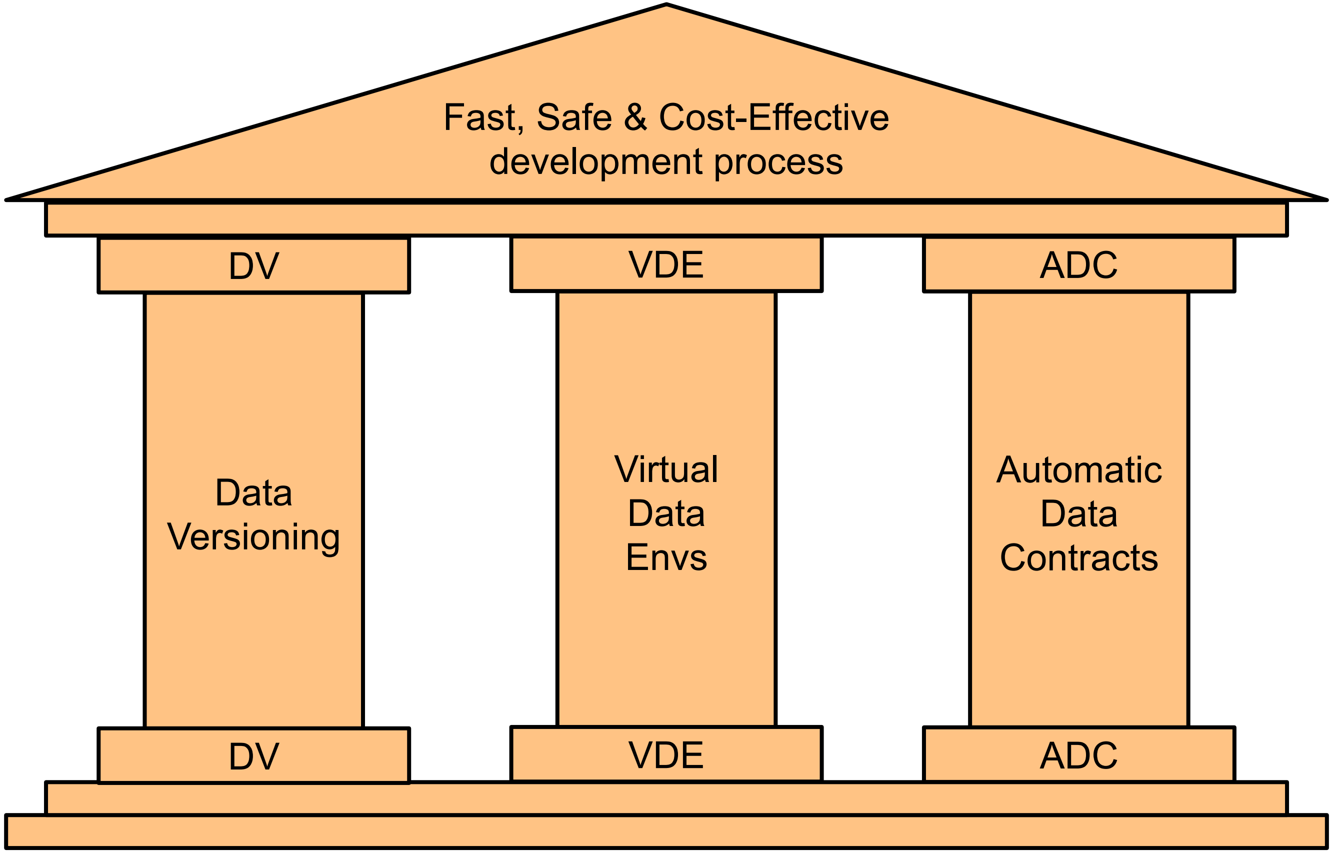


Scale

~~Scale~~ Organizational Complexity



Three Pillars of Data Productivity



Fast, Safe & Cost-Effective development process

DV

Data Versioning

DV

VDE

Virtual Data Envs

VDE

ADC

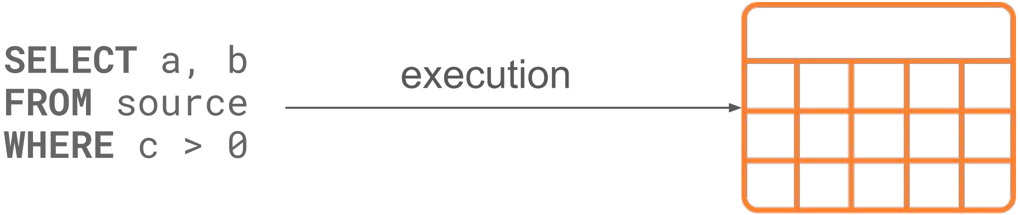
Automatic Data Contracts

ADC

Model: T in ETL

```sql
SELECT a, b FROM source WHERE c > 0
```

Model



```
SELECT a, b
FROM source
WHERE c > 0
```

Model Attributes

Impacts data

- Model query
- Type of data object
- Storage format
- Partitioning / clustering

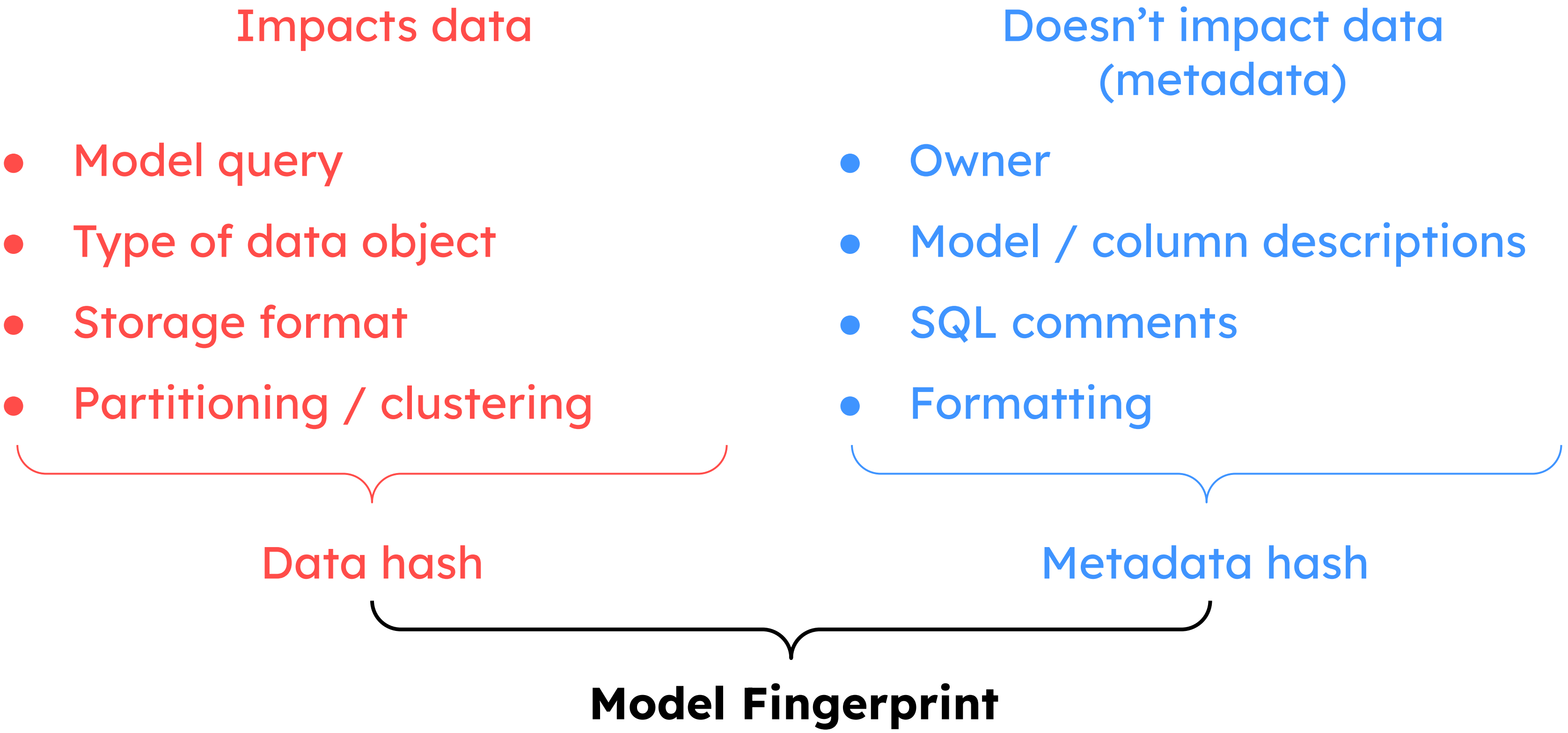
Data hash

Doesn’t impact data (metadata)

- Owner
- Model / column descriptions
- SQL comments
- Formatting

Metadata hash

Model Attributes



Impacts data

- Model query
- Type of data object
- Storage format
- Partitioning / clustering

Data hash

Doesn’t impact data (metadata)

- Owner
- Model / column descriptions
- SQL comments
- Formatting

Metadata hash

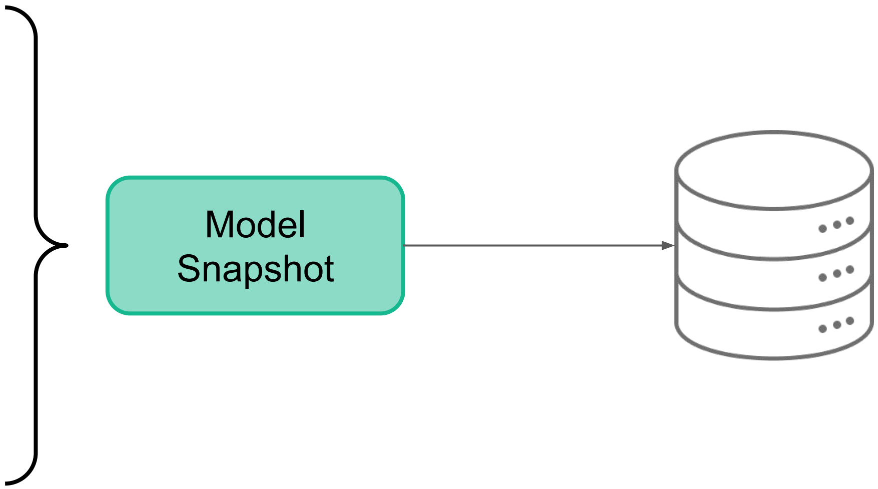
Model Fingerprint

Model Snapshot

```
SELECT a, b
FROM source
WHERE c > 0
```

Model Fingerprint

- Data hash
- Metadata hash

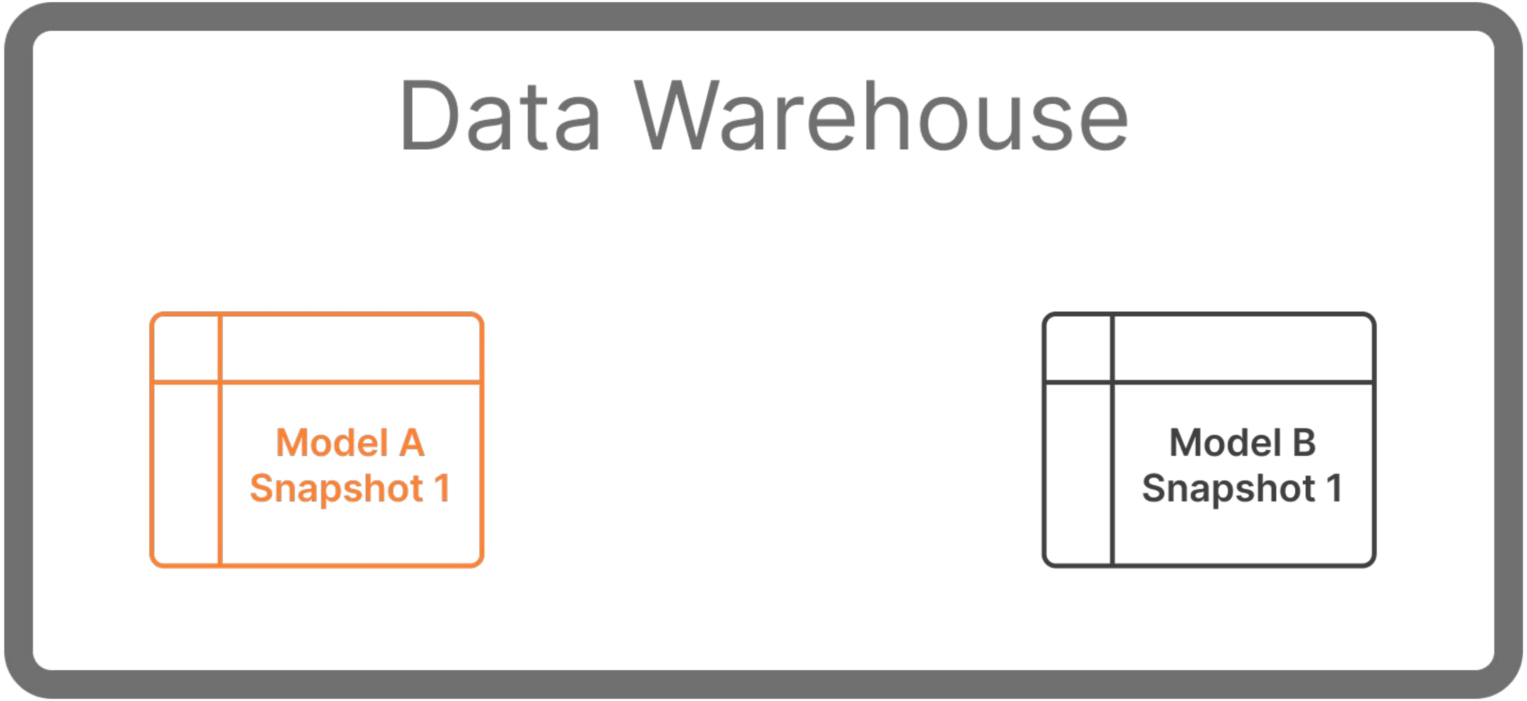


Model Snapshot

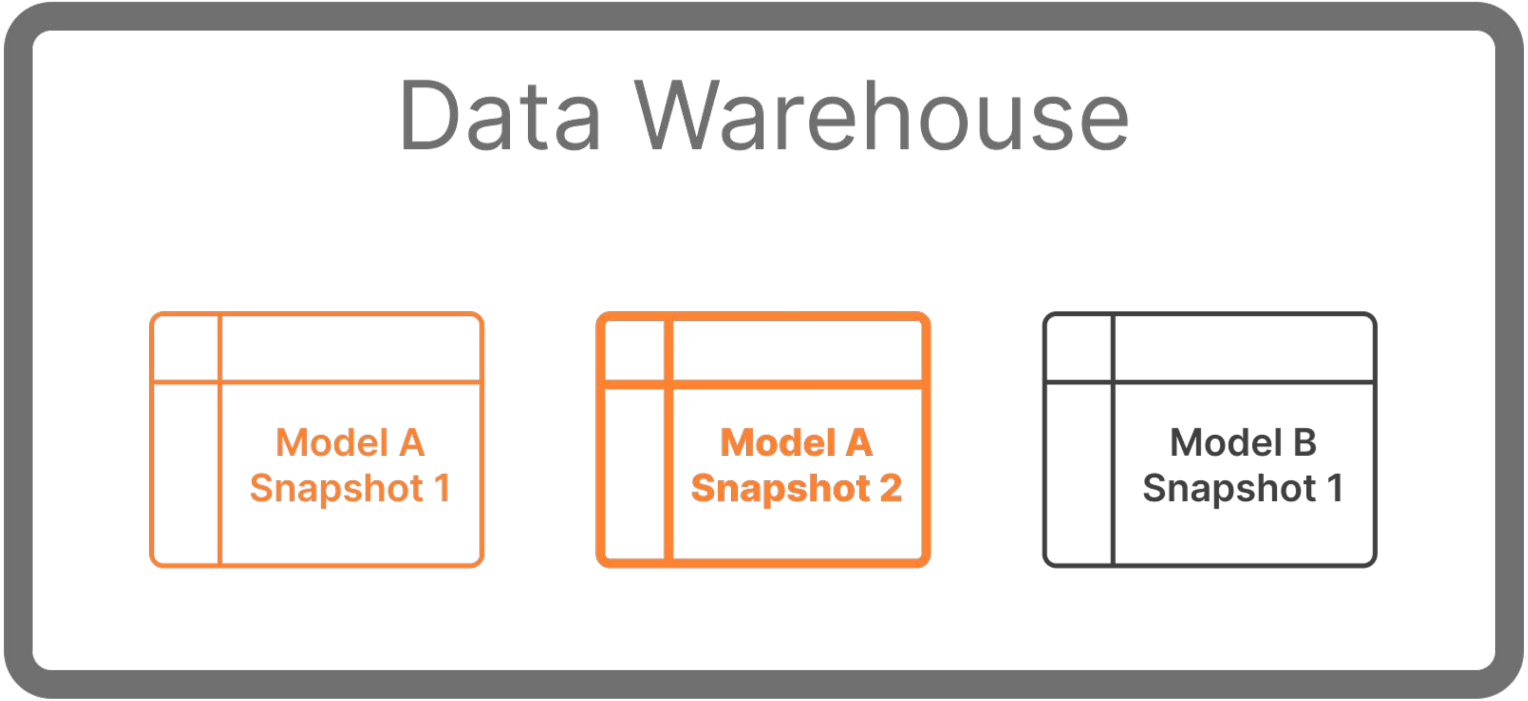
Snapshot Properties

- Model frozen in time
- Uniquely identified by the **fingerprint**
- Self-contained
- Immutable

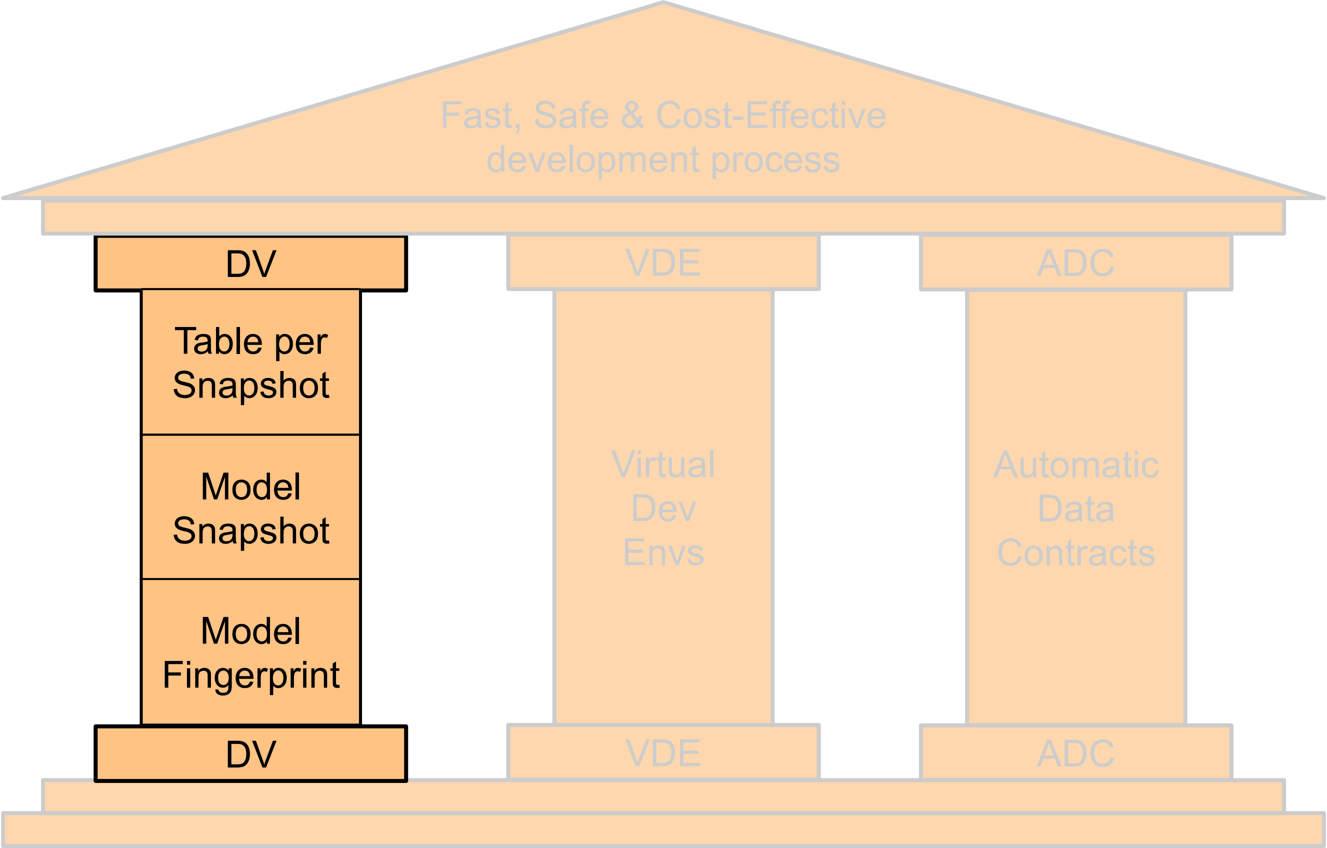
Snapshots in the Data Warehouse



Snapshots in the Data Warehouse



Pillar of Data Versioning

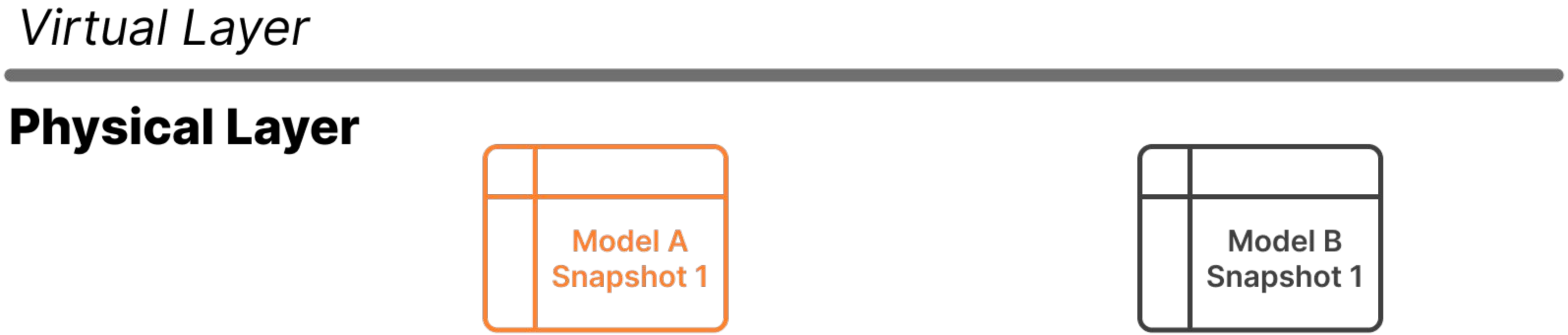


Virtual Data Environments

Virtual Layer

Physical Layer

Virtual Data Environments



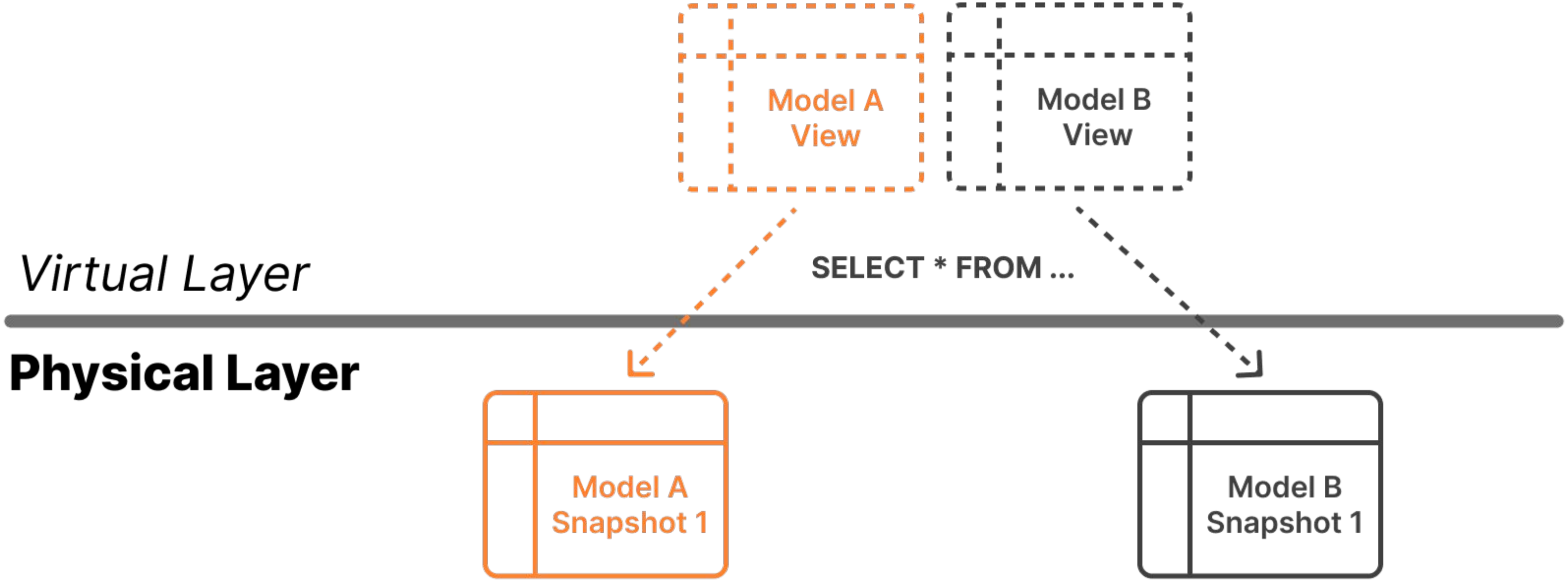
Virtual Layer

Physical Layer

Model A
Snapshot 1

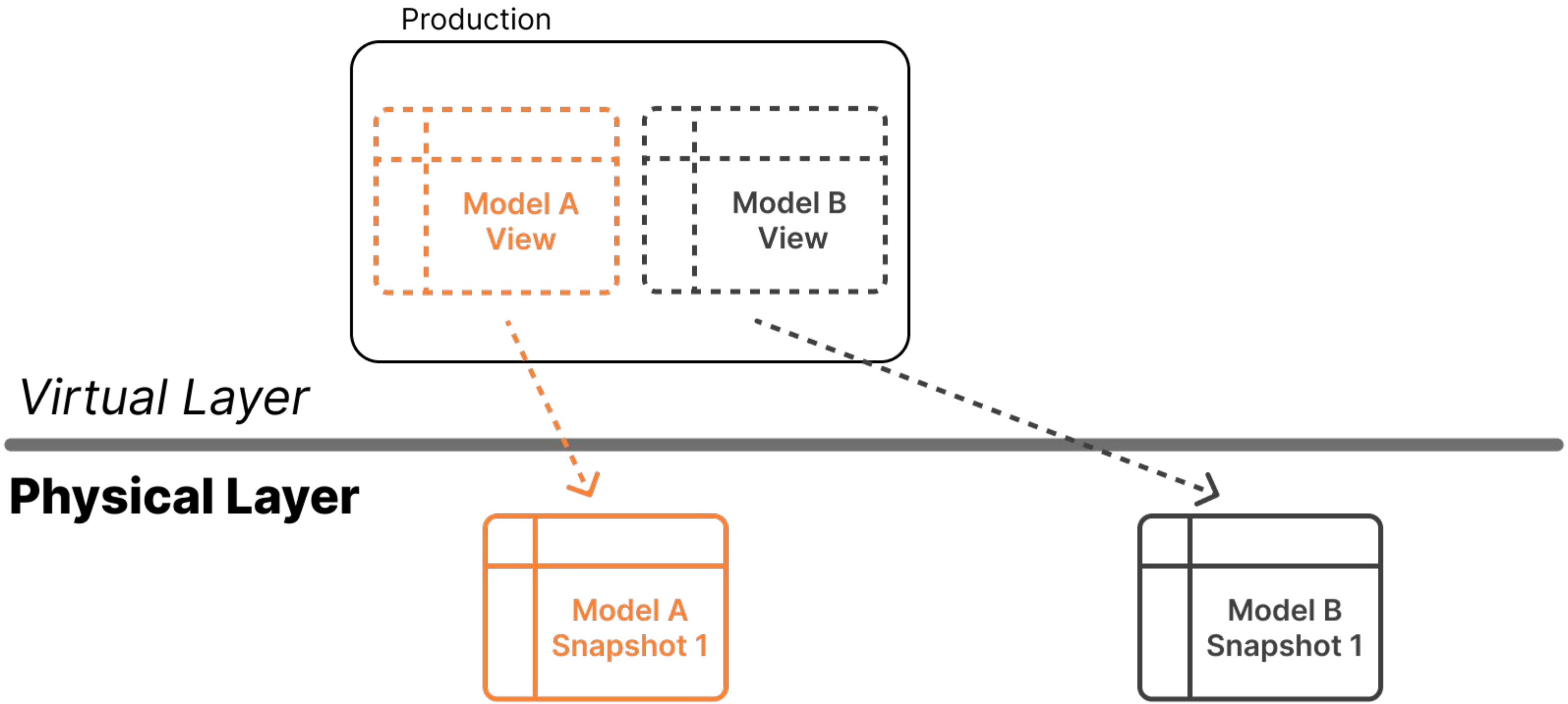
Model B
Snapshot 1

Virtual Data Environments



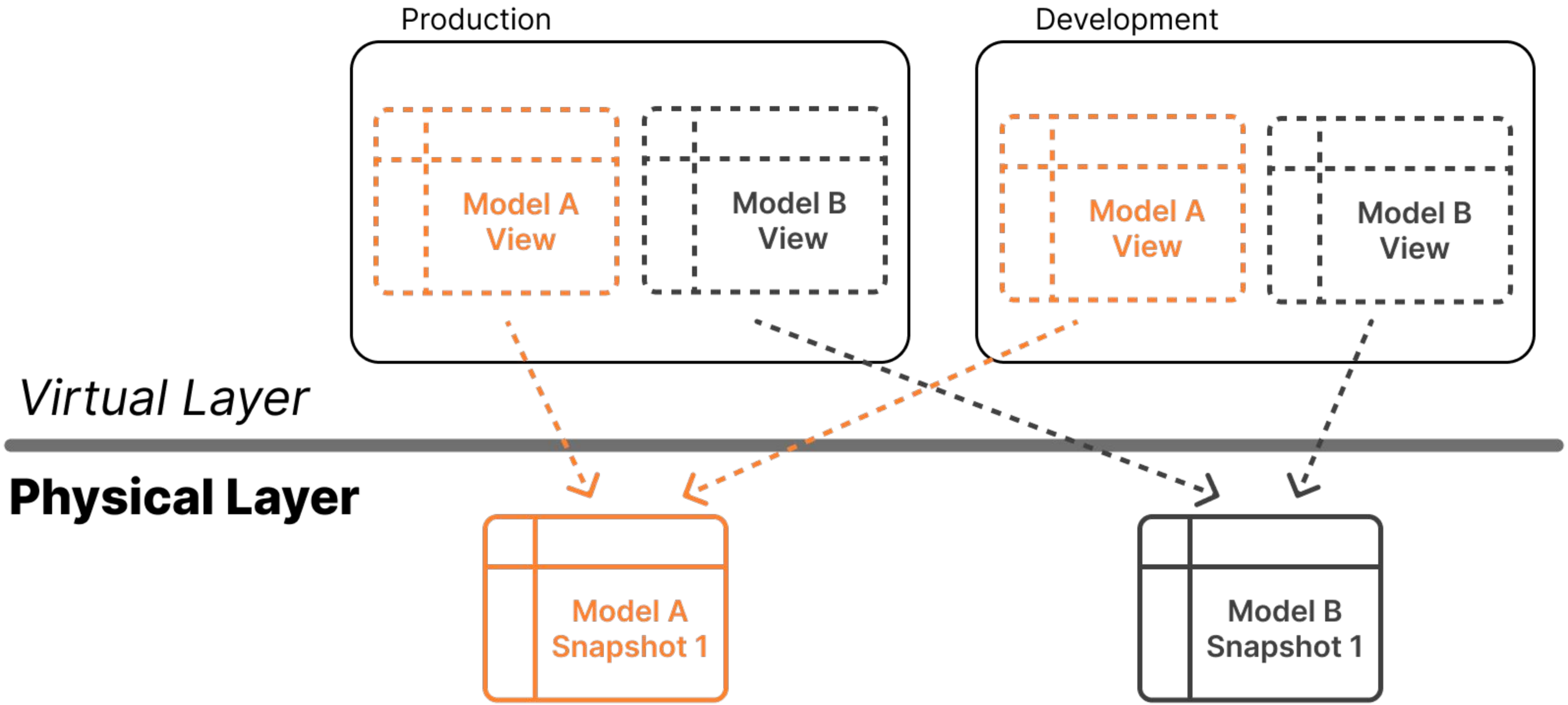
Virtual Data Environments: Flow

1. Starting production environment



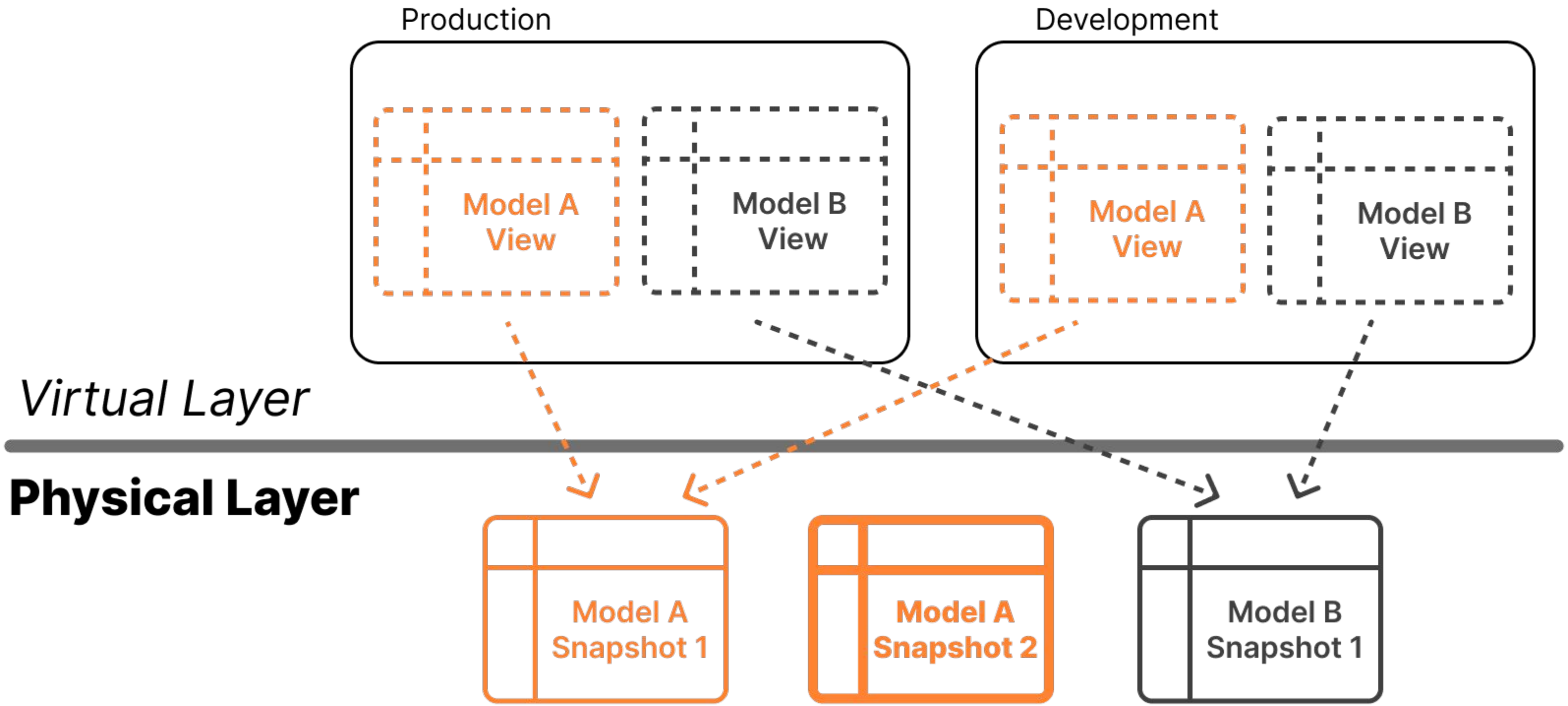
Virtual Data Environments: Flow

2. Create development as a mirror of production



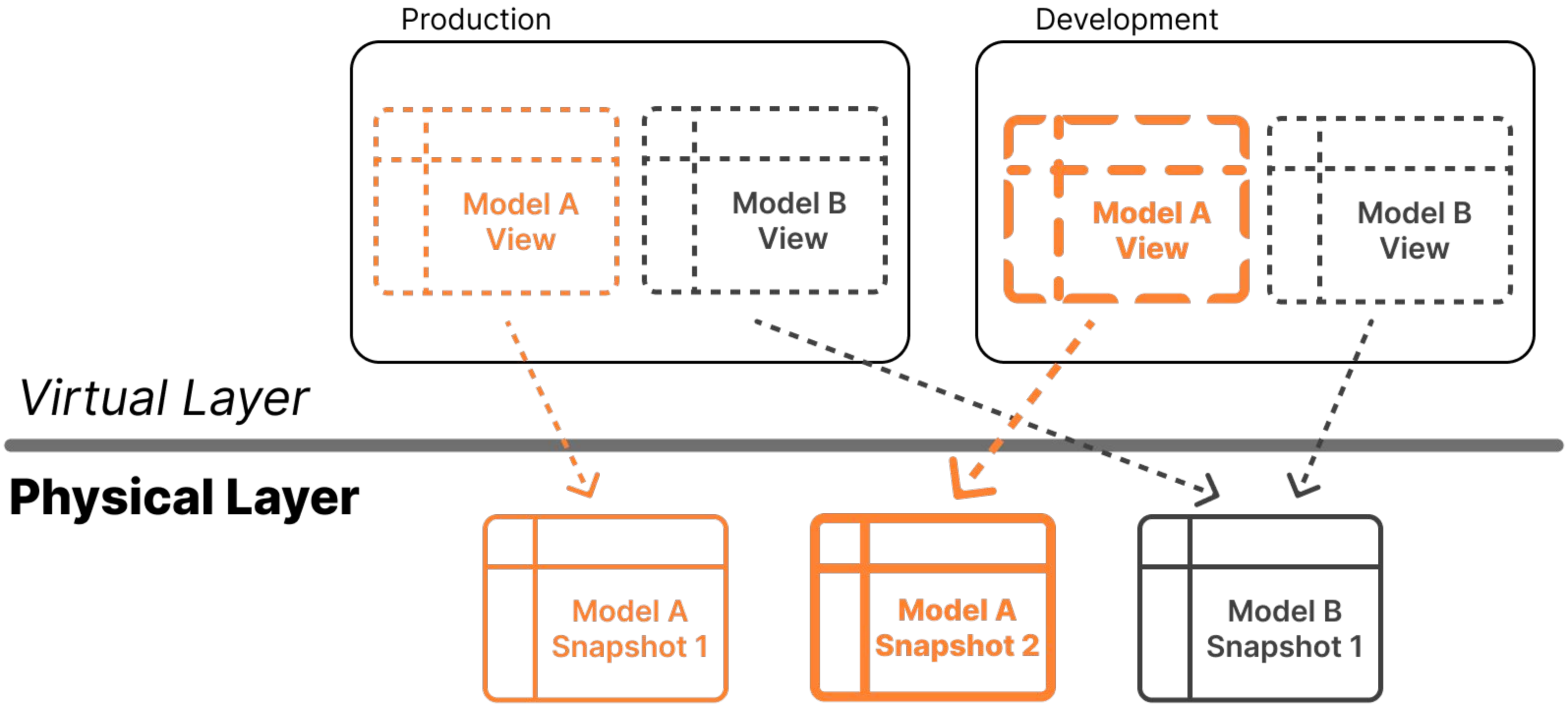
Virtual Data Environments: Flow

3. Make a change to Model A



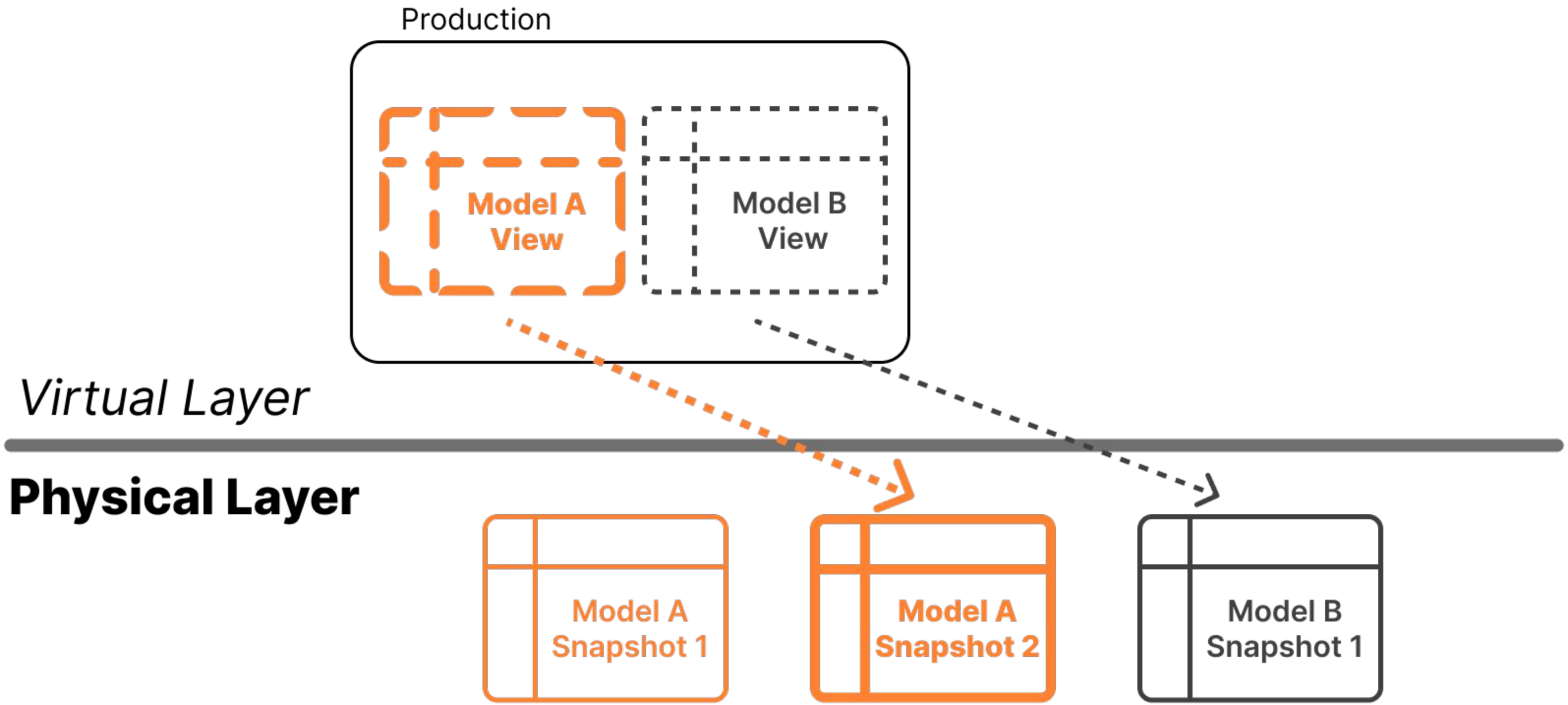
Virtual Data Environments: Flow

3. Make a change to Model A



Virtual Data Environments: Flow

4. Deploy model to production

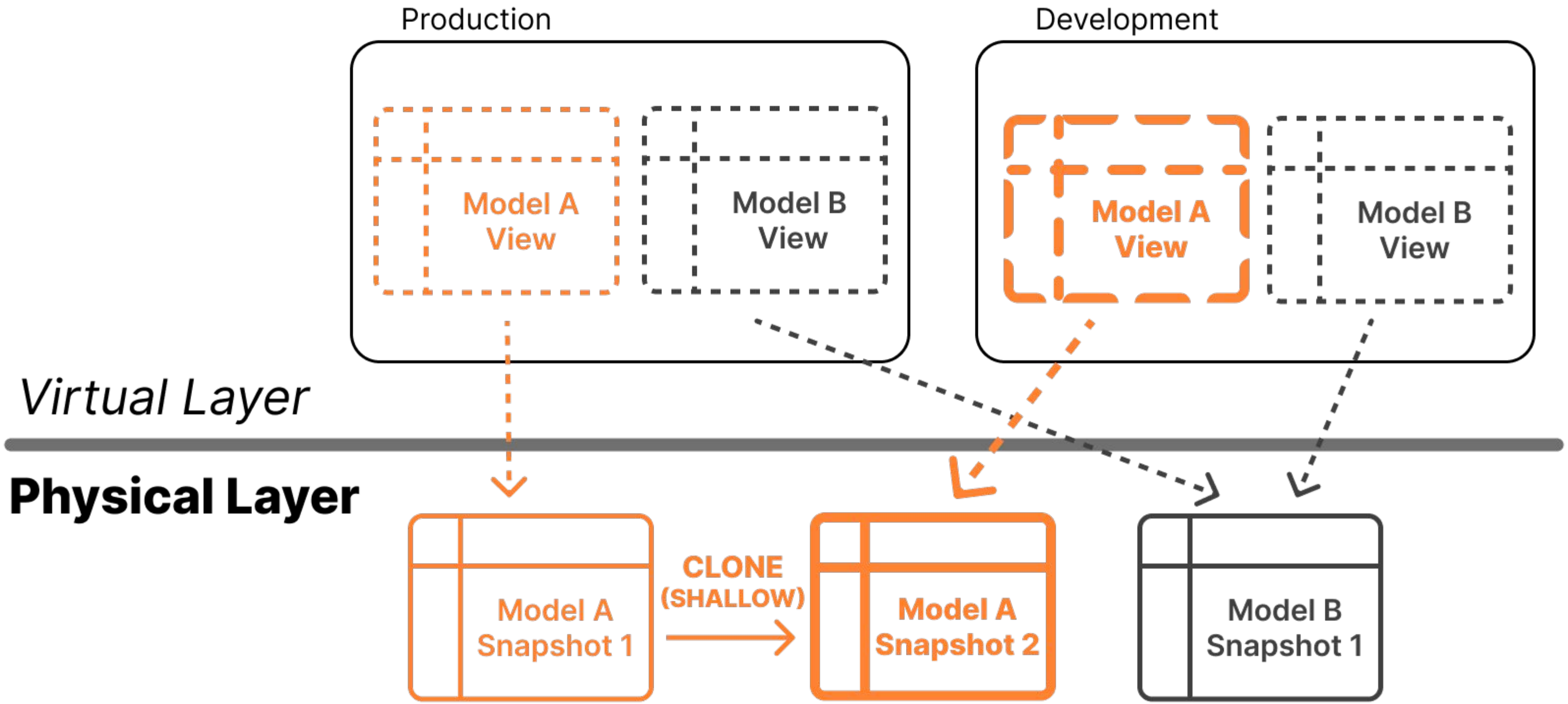


It's nice and all, but what if…

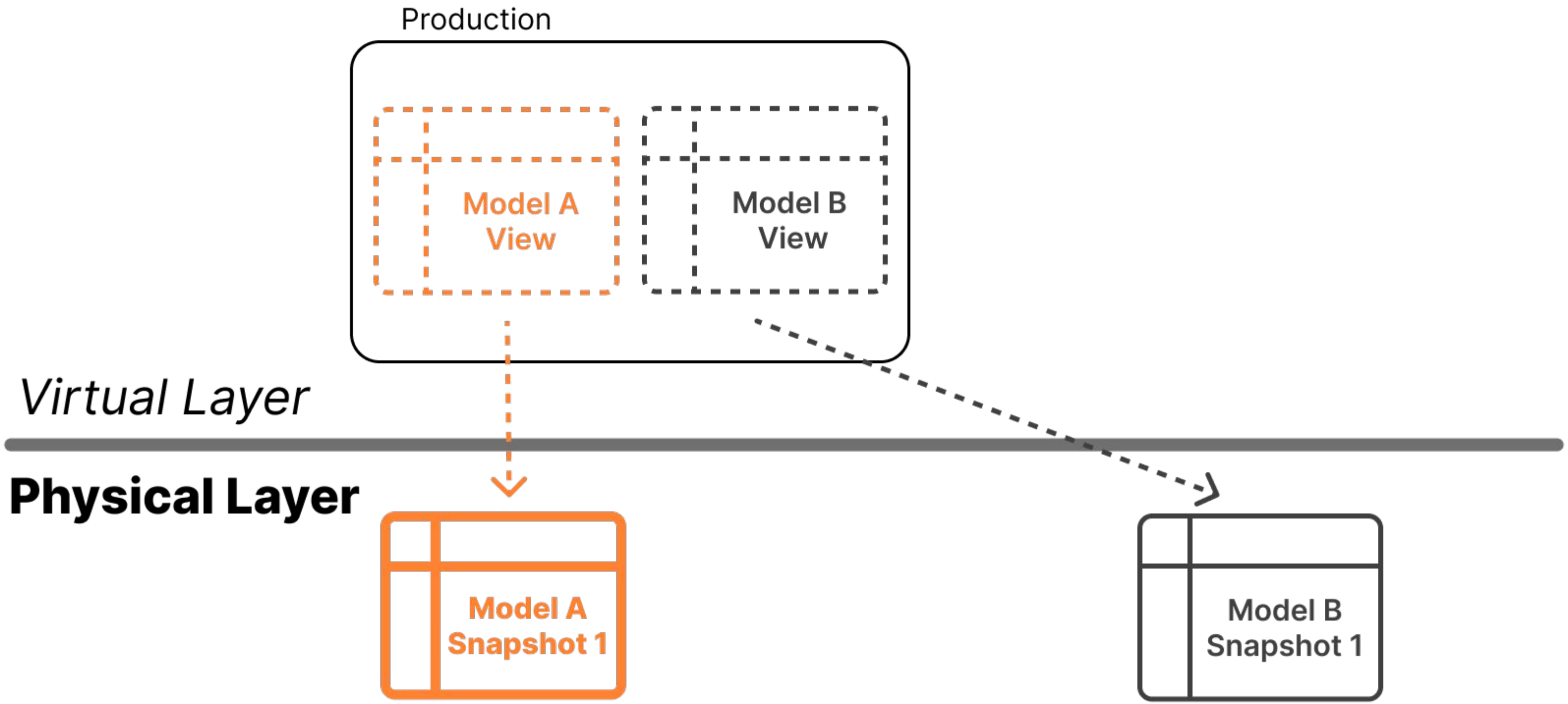
- The table is too large and costly to rebuild
- The model logic is not idempotent
- The historical data is not available upstream

It's nice and all, but what if…

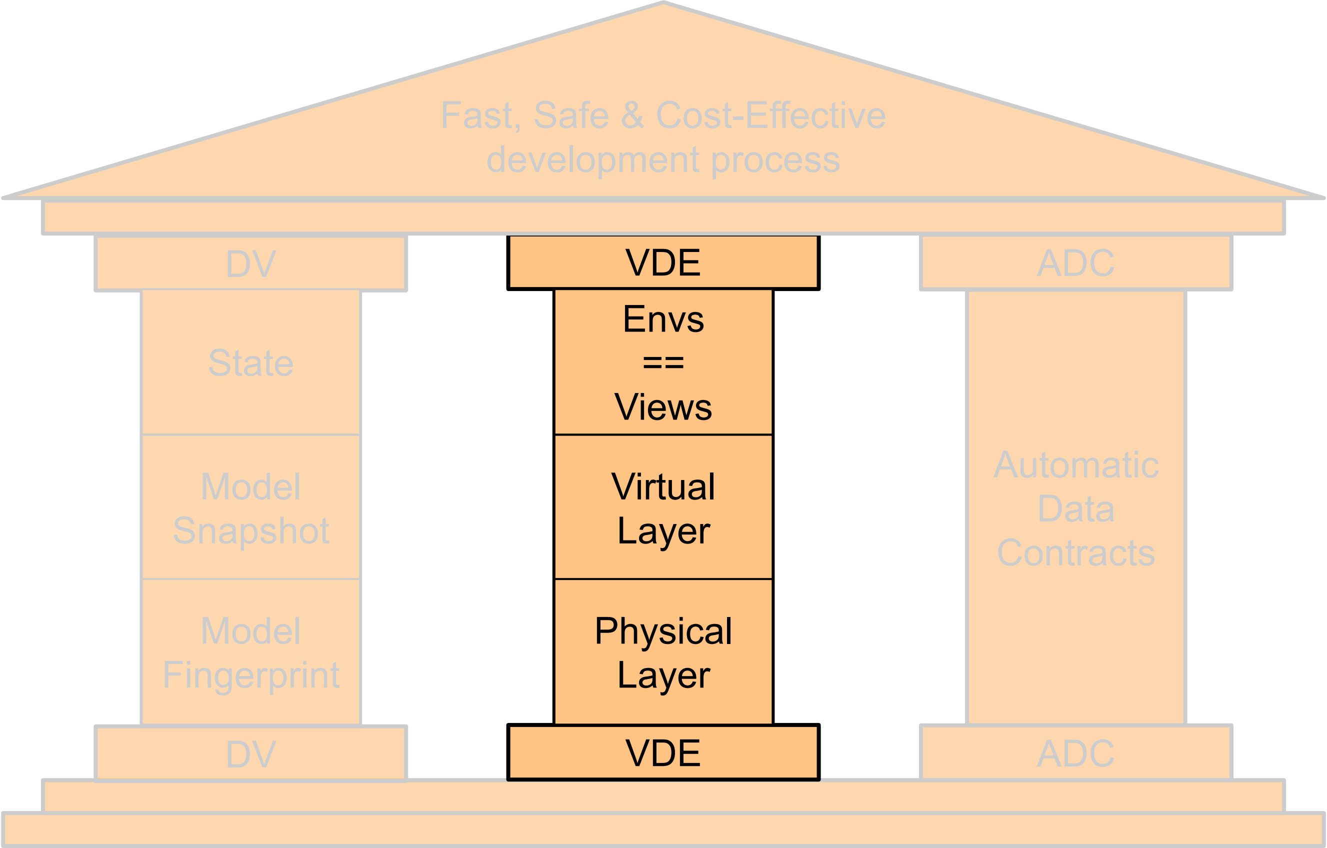
Forward-only Changes



Forward-only Changes



Pillar of Virtual Data Environments

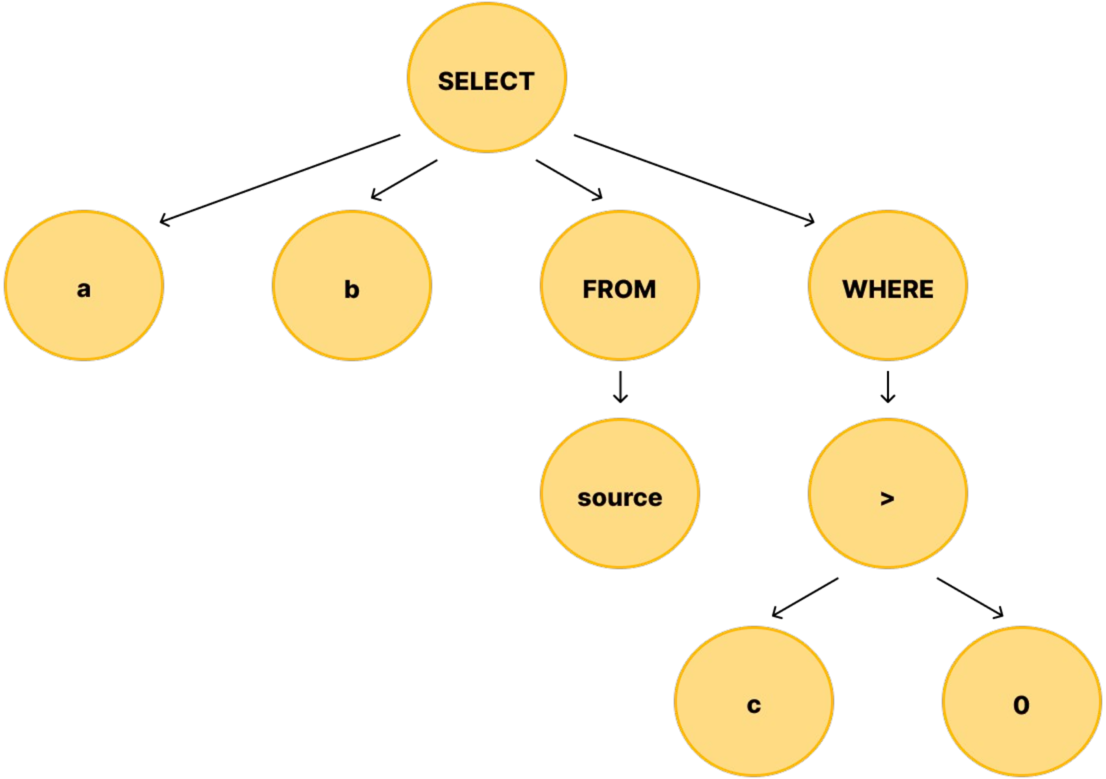


Semantic Understanding

```sql
SELECT a, b FROM source WHERE c > 0;
```

Semantic Understanding

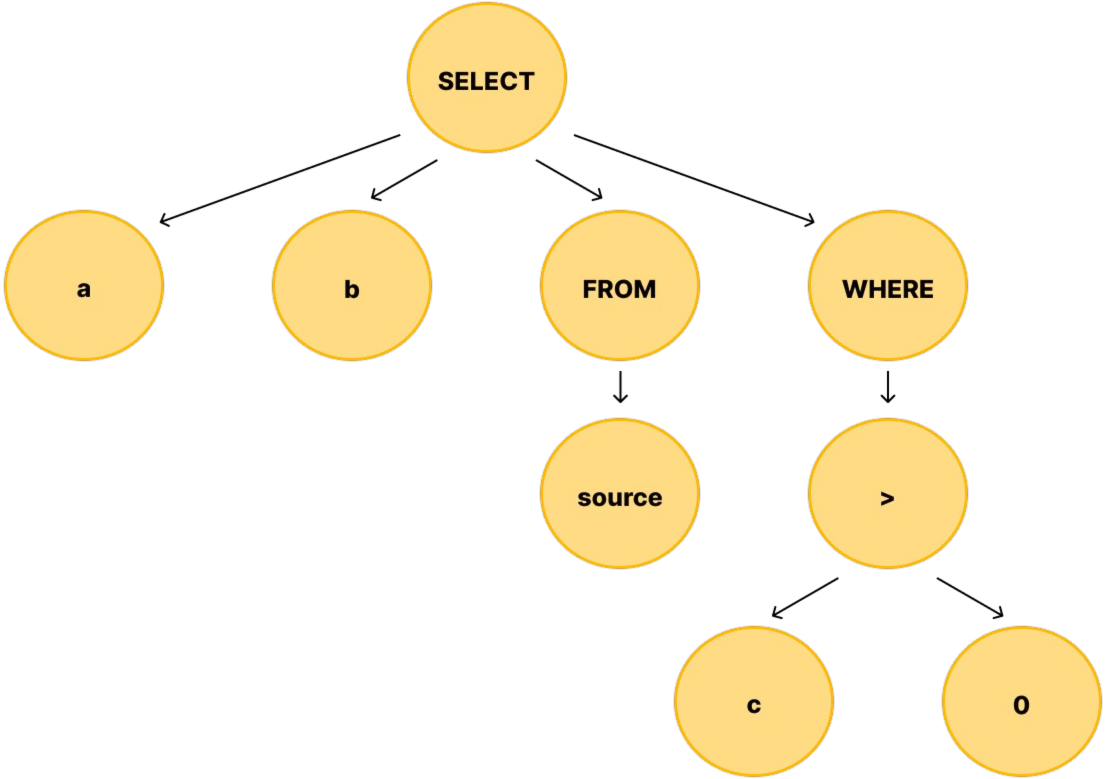
```
SELECT a, b FROM source WHERE c > 0;
```



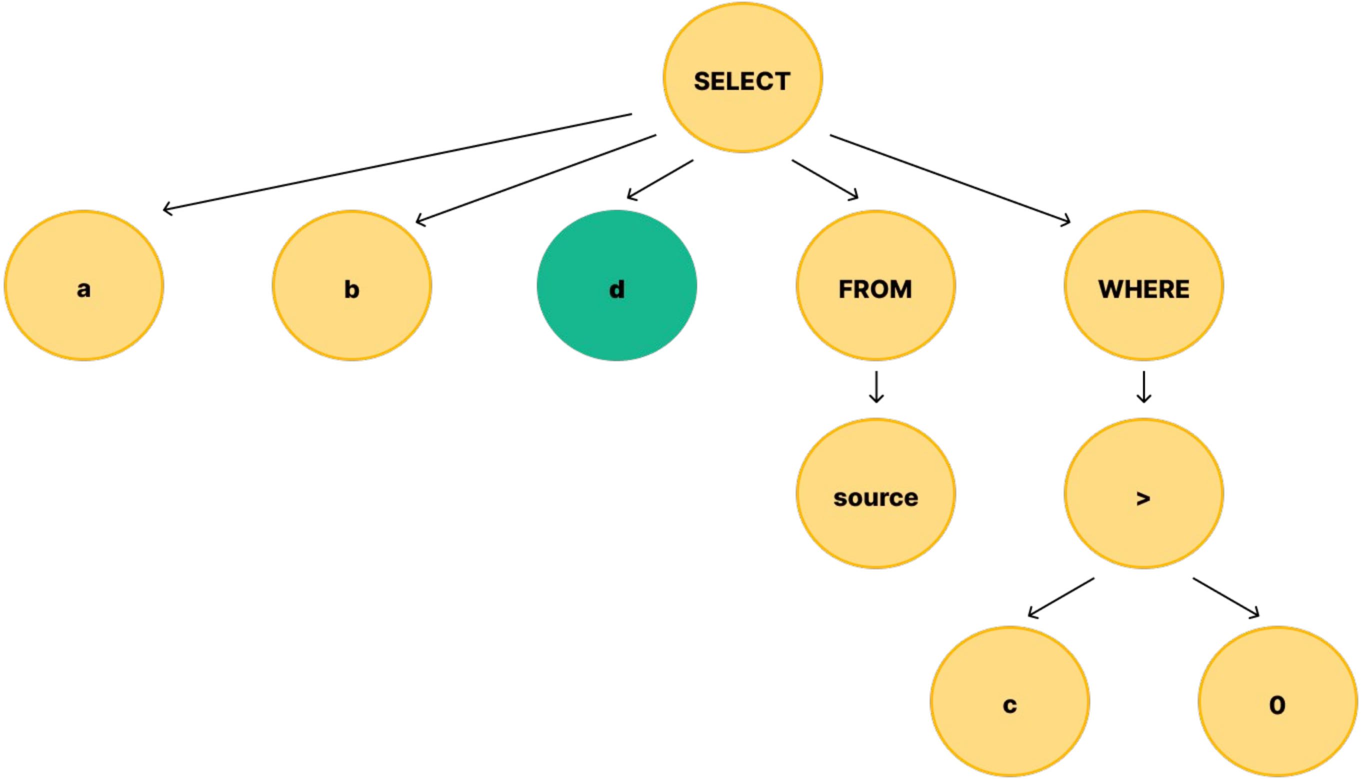
Semantic Understanding

```
SELECT a, b, d FROM source WHERE c > 0
```

Analyzing the Change

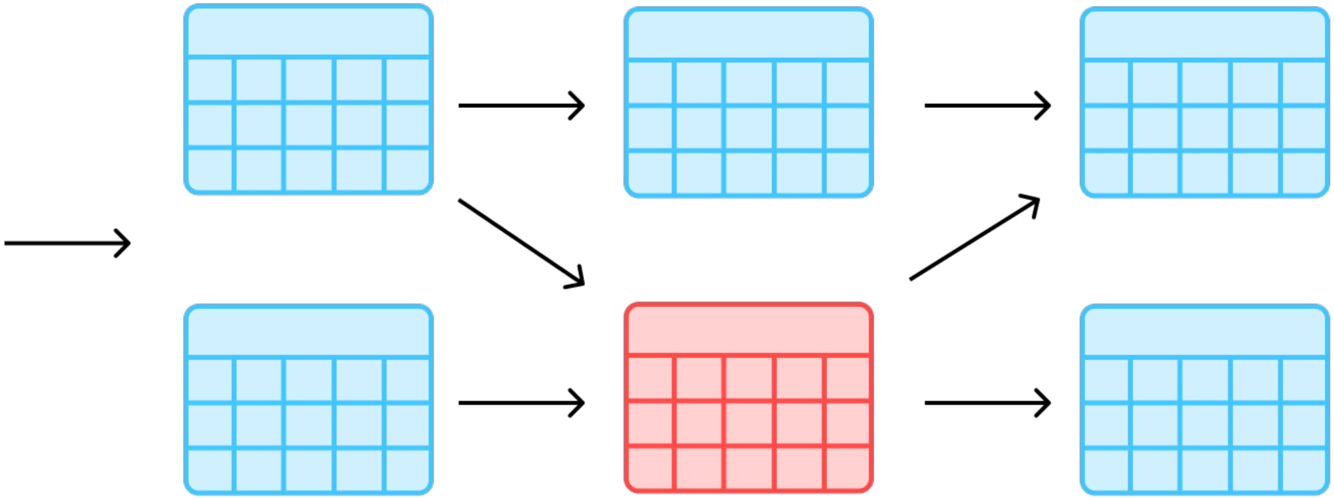


Analyzing the Change



Non-breaking Change

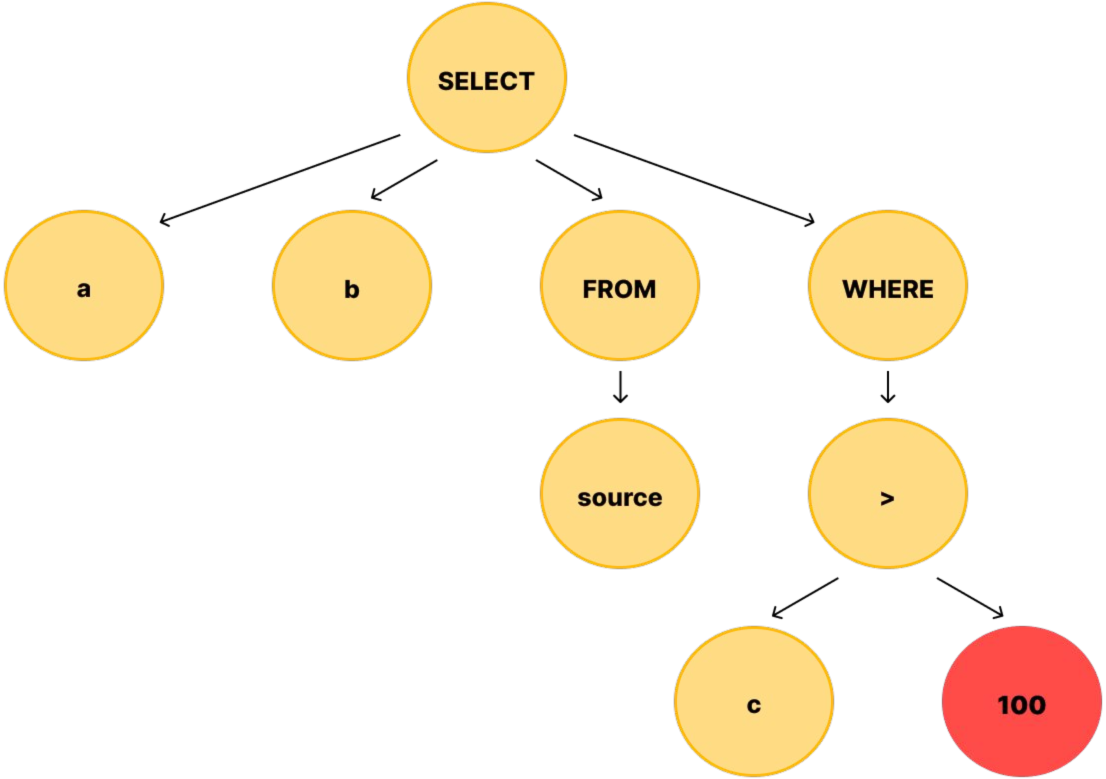
```
SELECT
     a,
     b,
     d
FROM source
WHERE c > 0
```



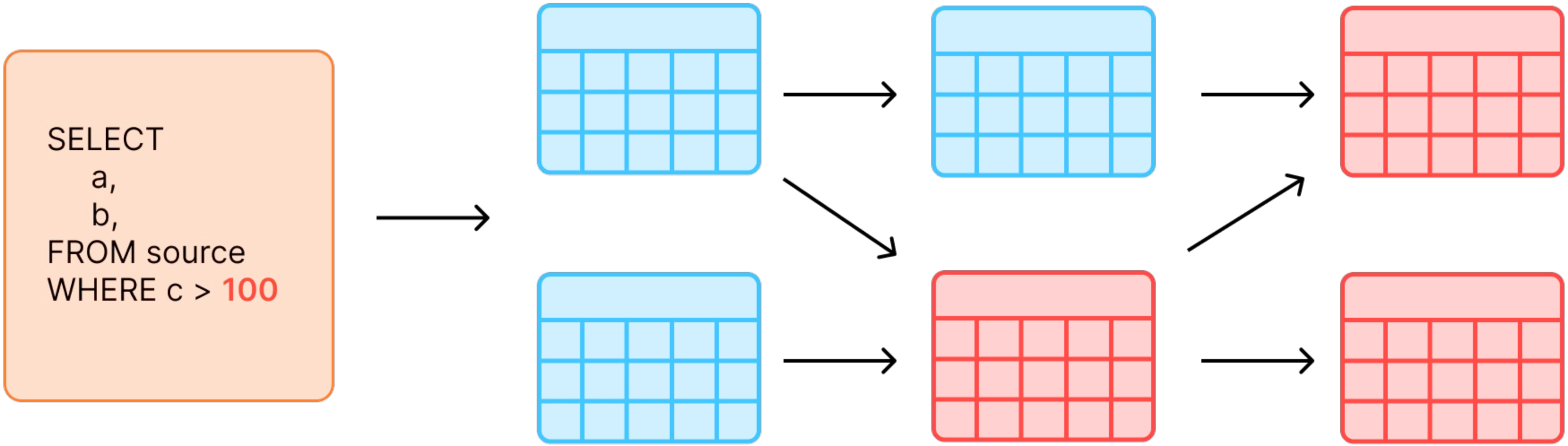
Breaking Change

```
SELECT a, b FROM source WHERE c > 100
```

Breaking Change

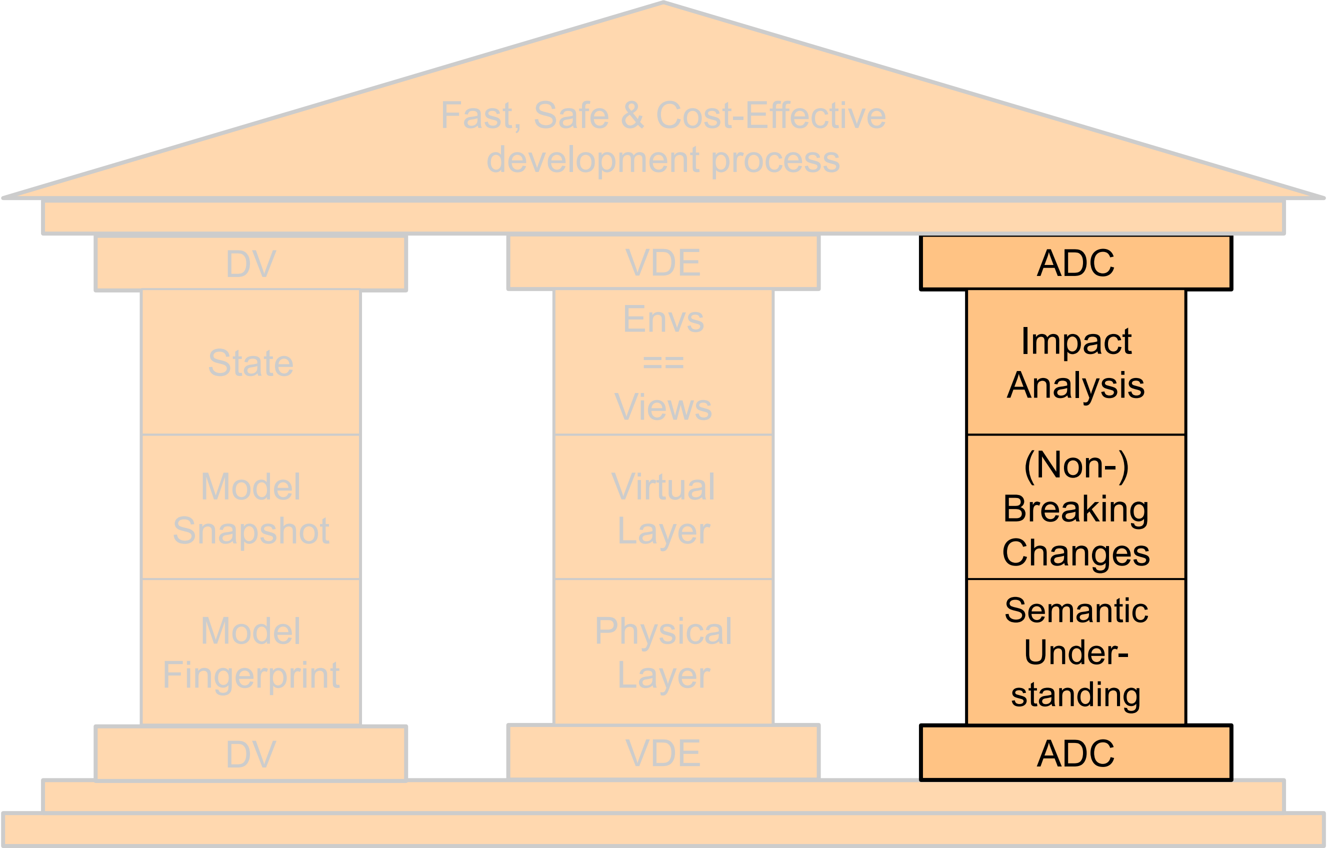


Breaking Change

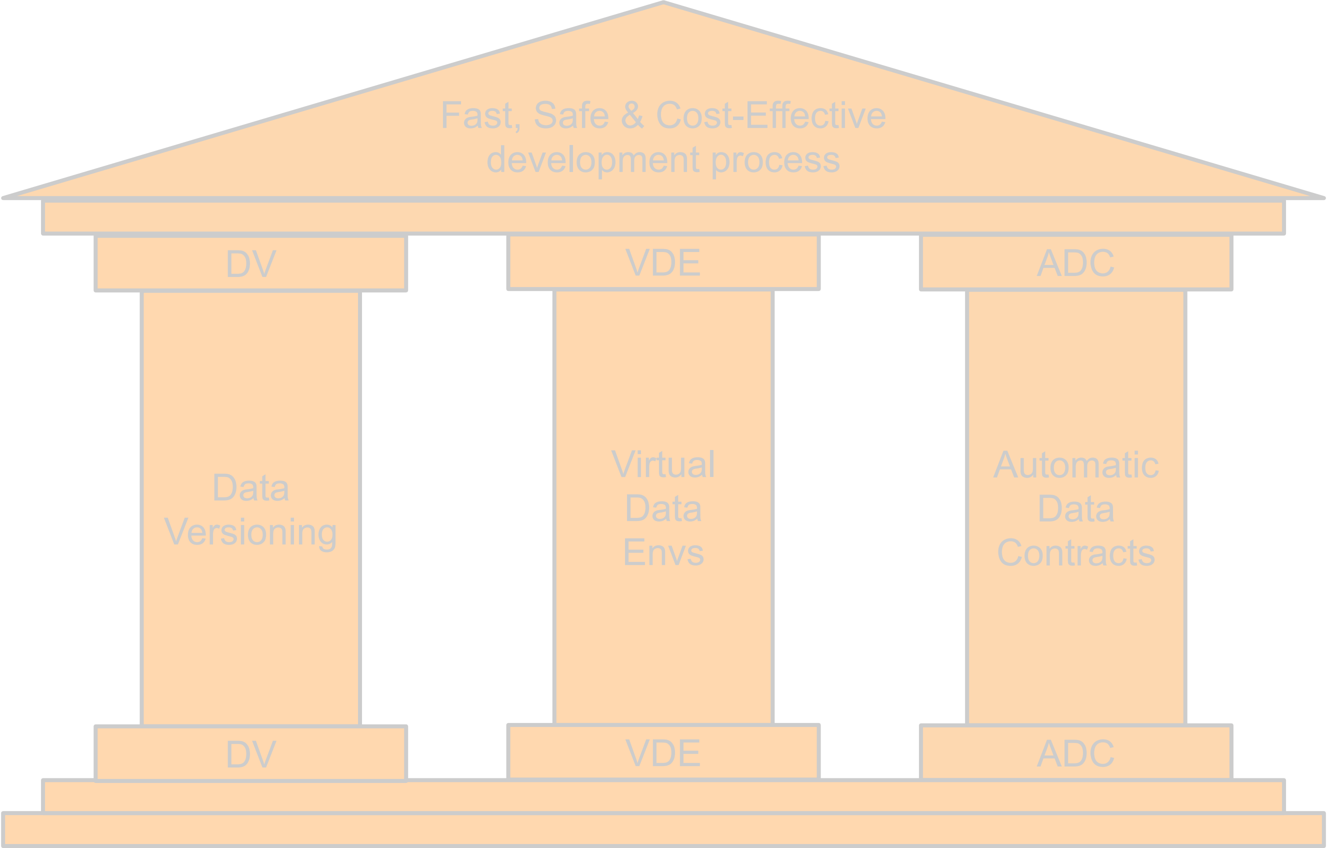


```
SELECT
    a,
    b,
FROM source
WHERE c > 100
```

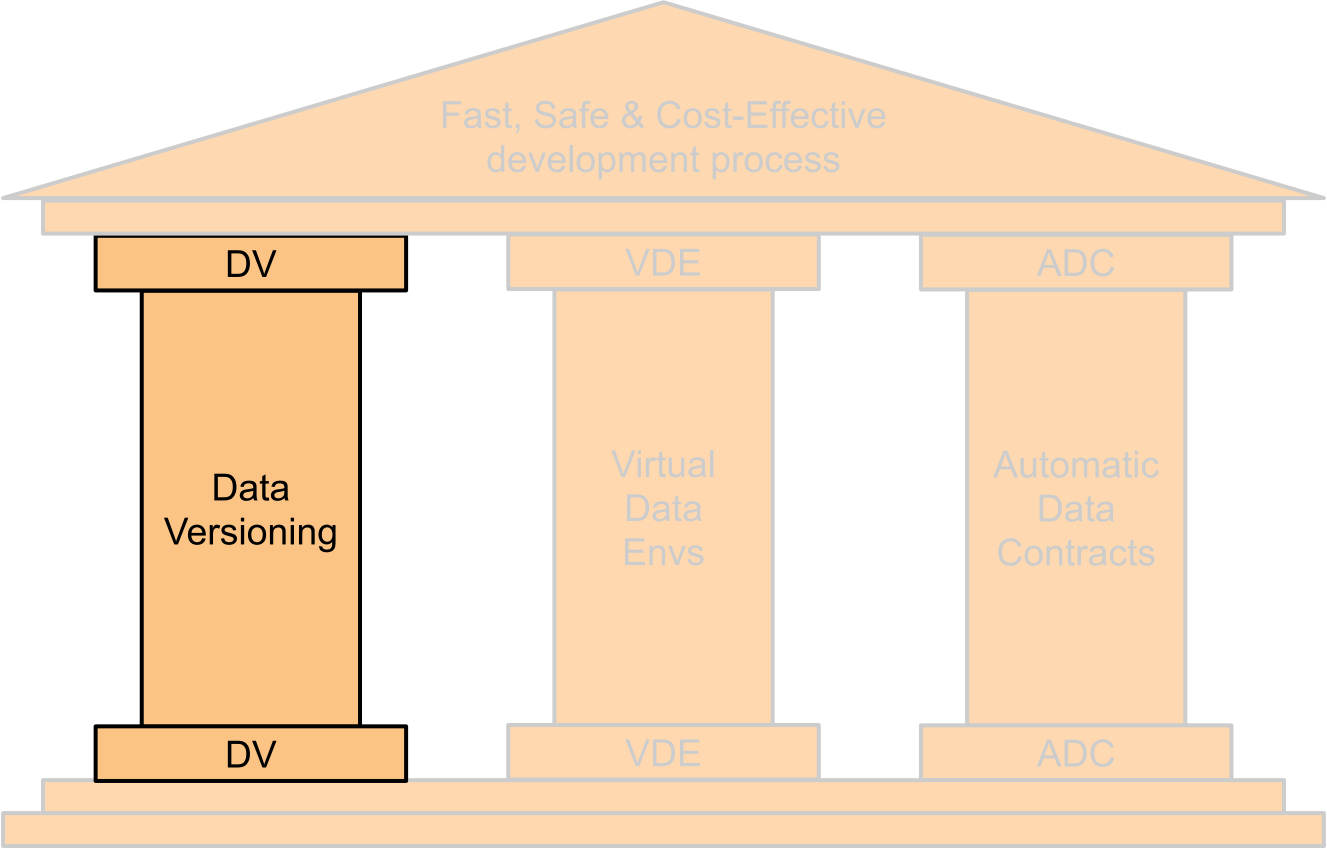
Pillar of Automatic Data Contracts



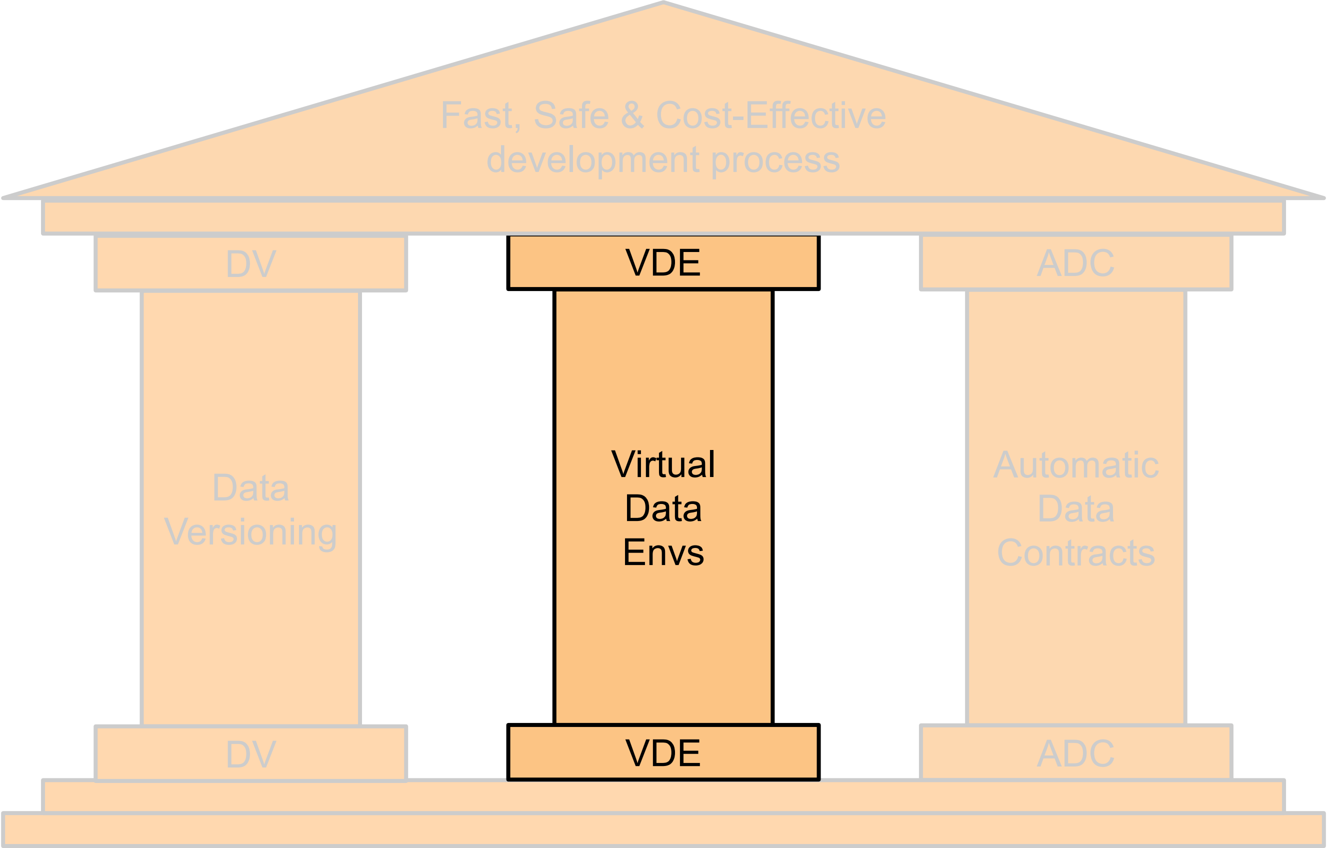
Three Pillars of Data Productivity



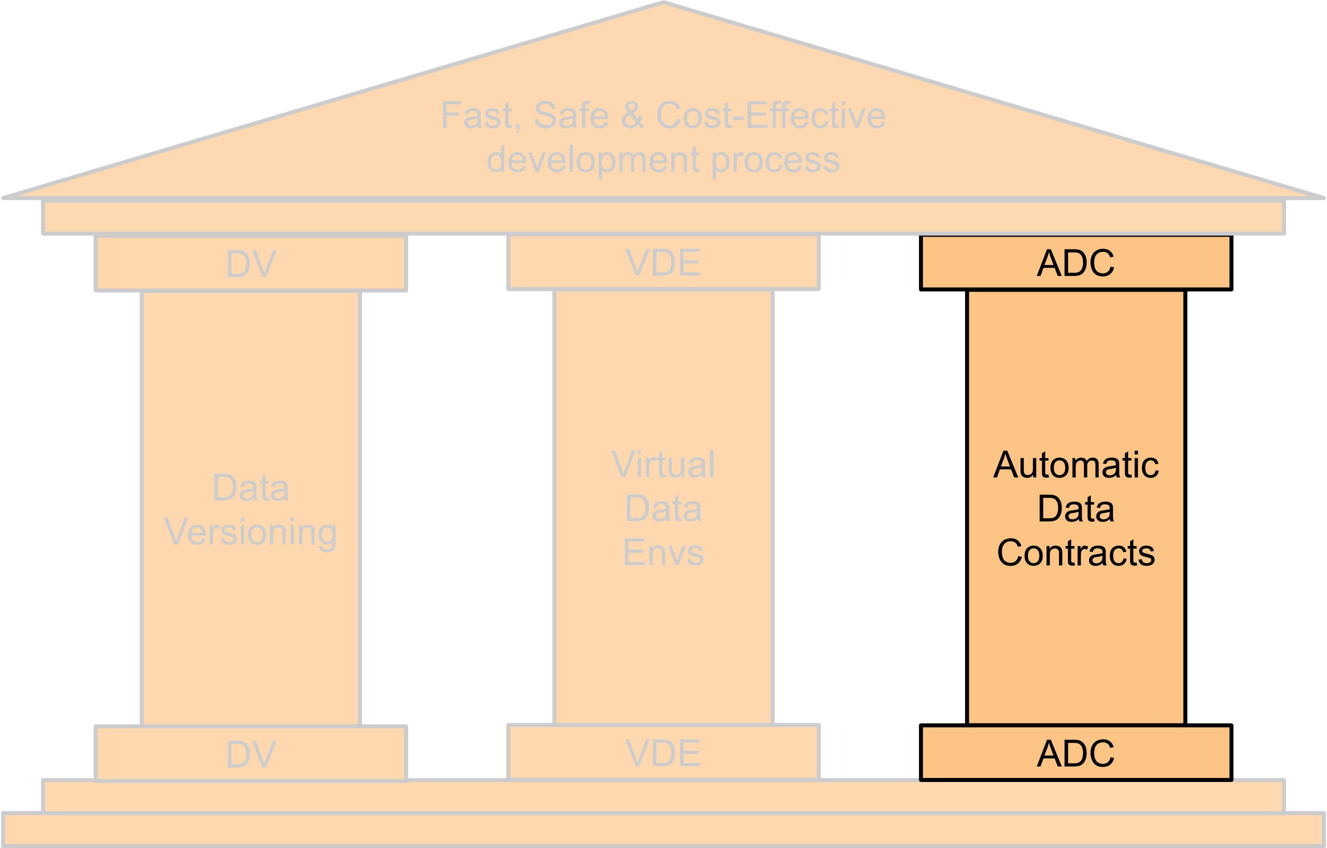
Three Pillars of Data Productivity



Three Pillars of Data Productivity



Three Pillars of Data Productivity



Fast, Safe & Cost-Effective development process

DV

Data Versioning

DV

VDE

Virtual Data Envs

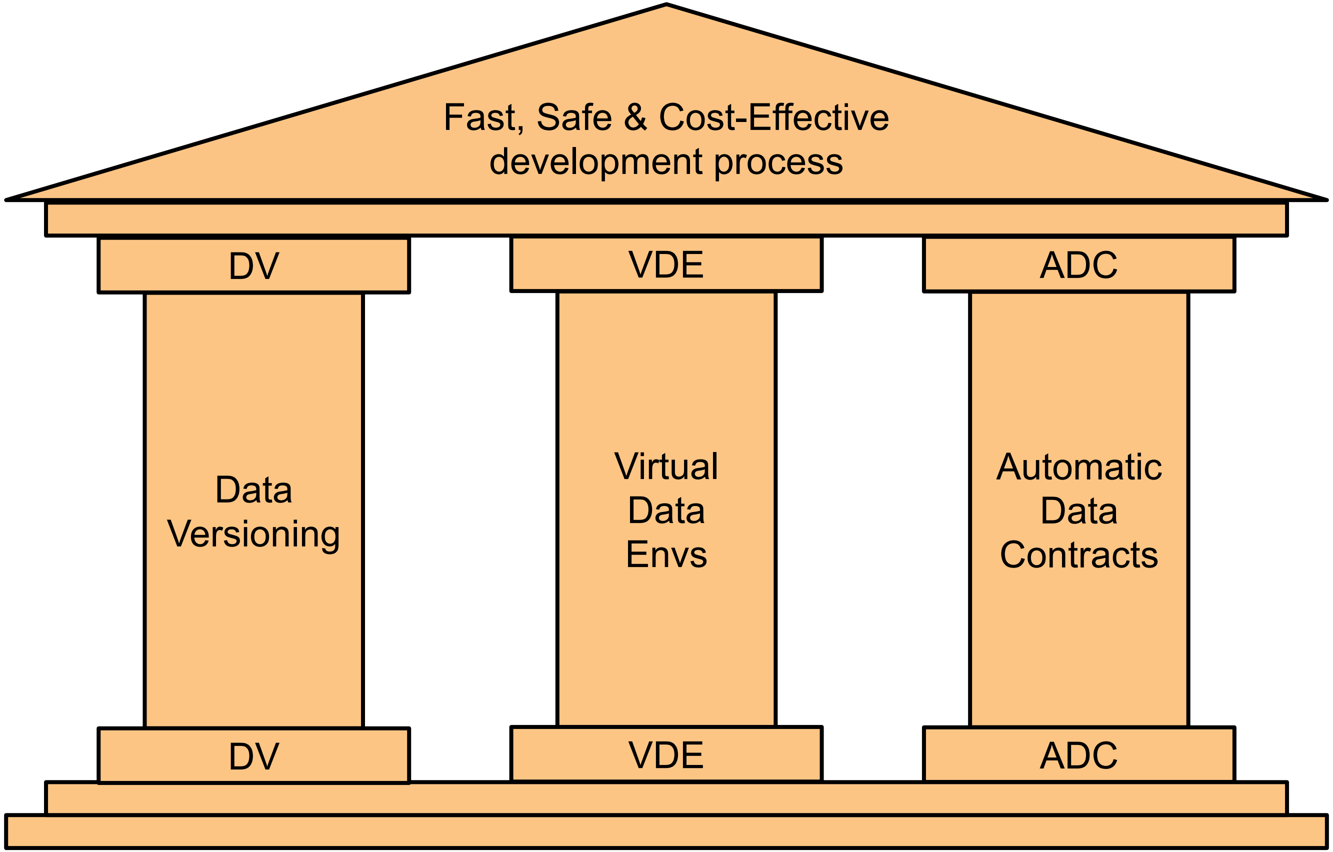
VDE

ADC

Automatic Data Contracts

ADC

Three Pillars of Data Productivity

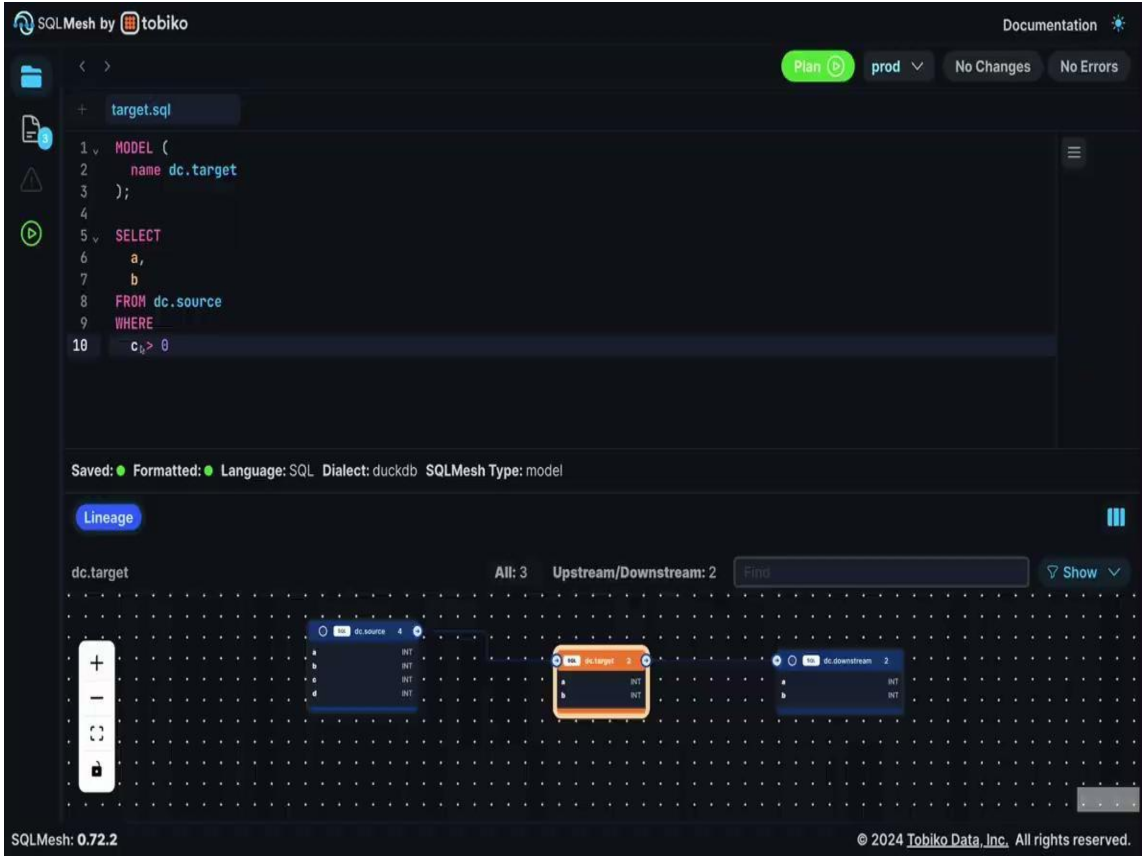


Putting Ideas into Practice with SQLMesh

About SQLMesh

- Open-source data transformation and modeling tool
- Pure SQL, no Jinja
- Batteries included: interval tracking, unit tests, CI/CD, etc
- Backwards compatible with dbt
- Brings **Three Pillars of Data Productivity** to life

Non-breaking Change



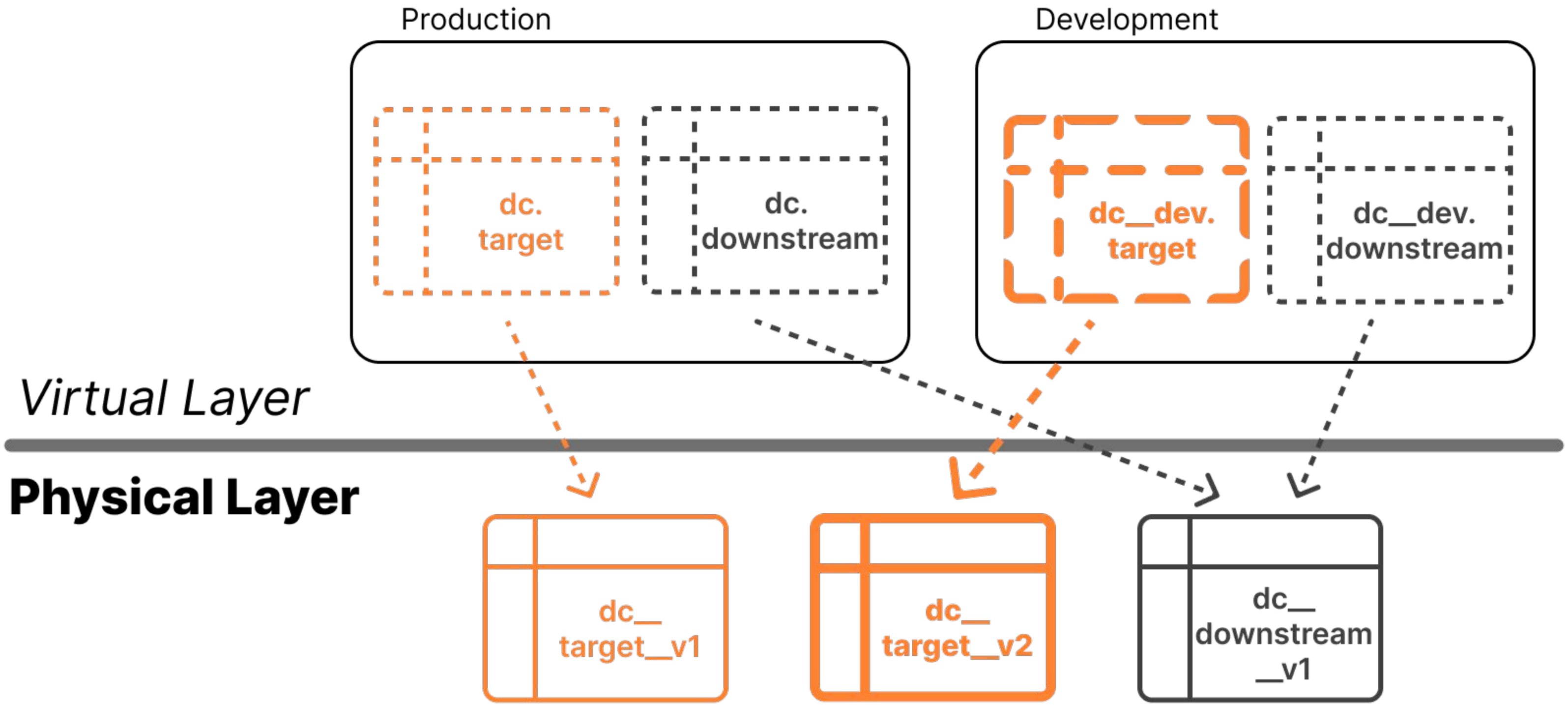
```sql
MODEL (
  name dc.target
);

SELECT
  a,
  b
FROM dc.source
WHERE
  c > 0
```

Saved: ● Formatted: ● Language: SQL Dialect: duckdb SQLMesh Type: model

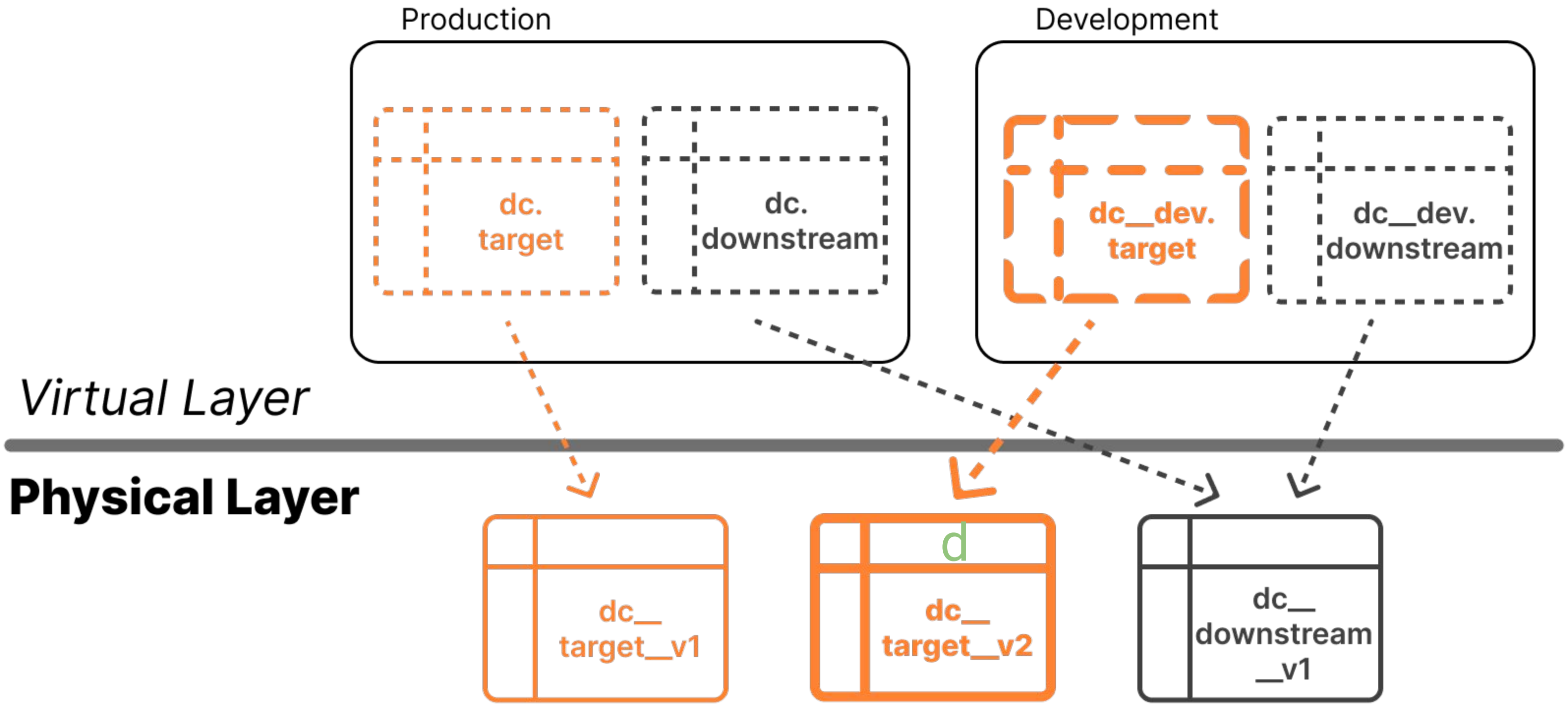
SQLMesh: 0.72.2

© 2024 Tobiko Data, Inc. All rights reserved.

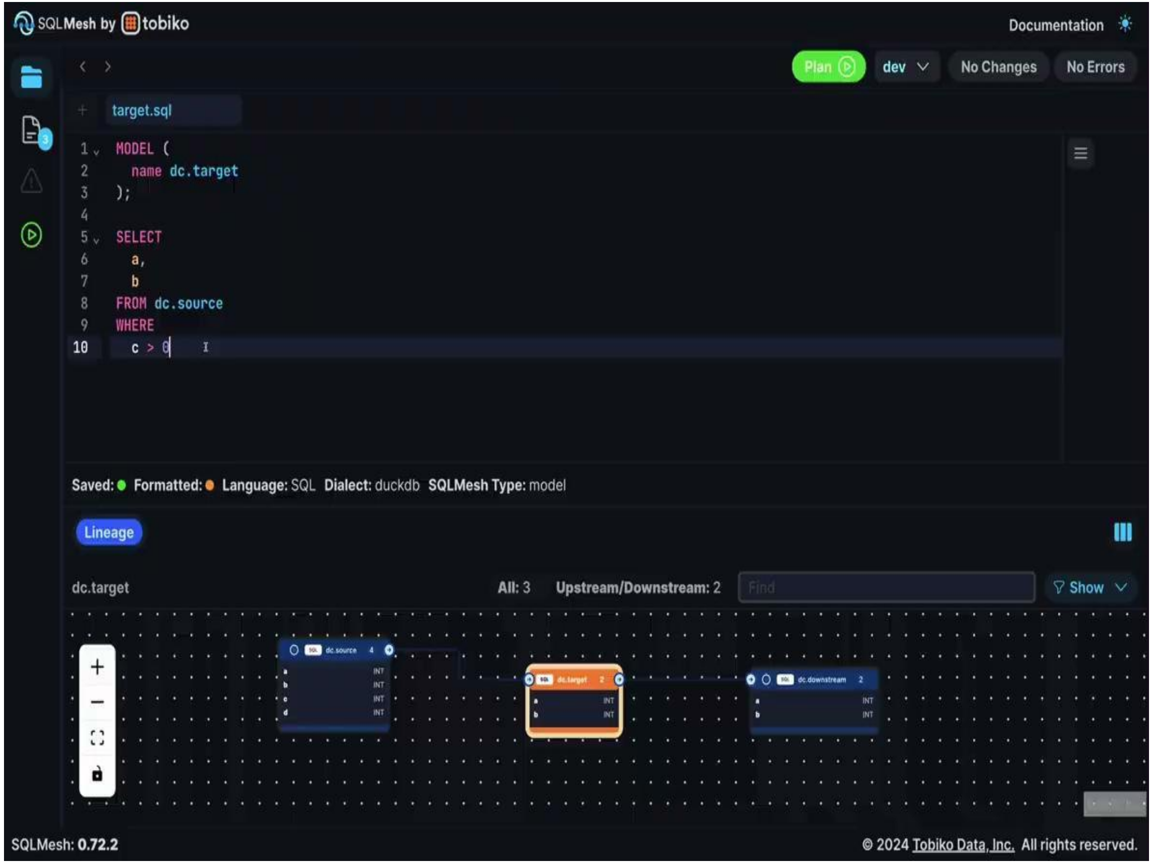
Non-breaking Change



Non-breaking Change



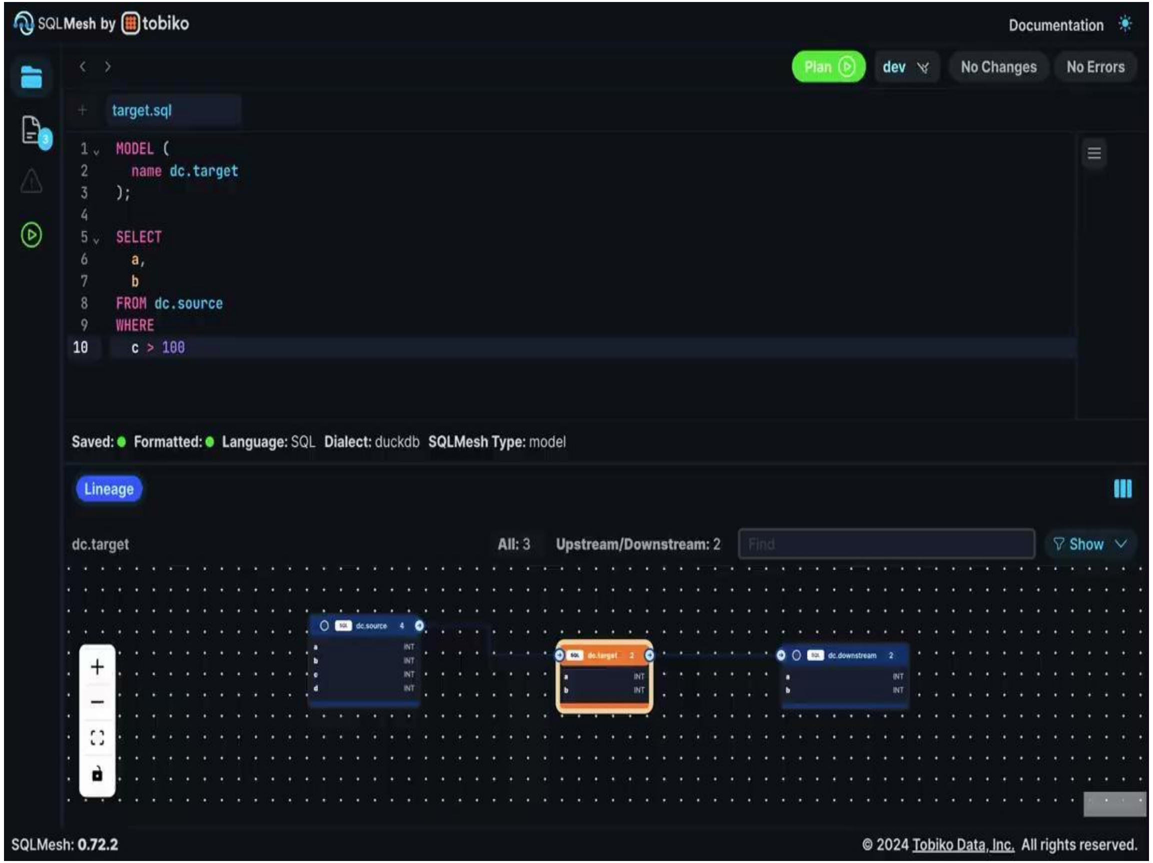
Breaking Change



```sql
MODEL (
  name dc.target
);

SELECT
  a,
  b
FROM dc.source
WHERE
  c > 0
```

Saved: ● Formatted: ● Language: SQL Dialect: duckdb SQLMesh Type: model

Deploy to Prod



```sql
MODEL (
  name dc.target
);

SELECT
  a,
  b
FROM dc.source
WHERE
  c > 100
```

Saved: ● Formatted: ● Language: SQL Dialect: duckdb SQLMesh Type: model

tobiko

Join us on Slack!

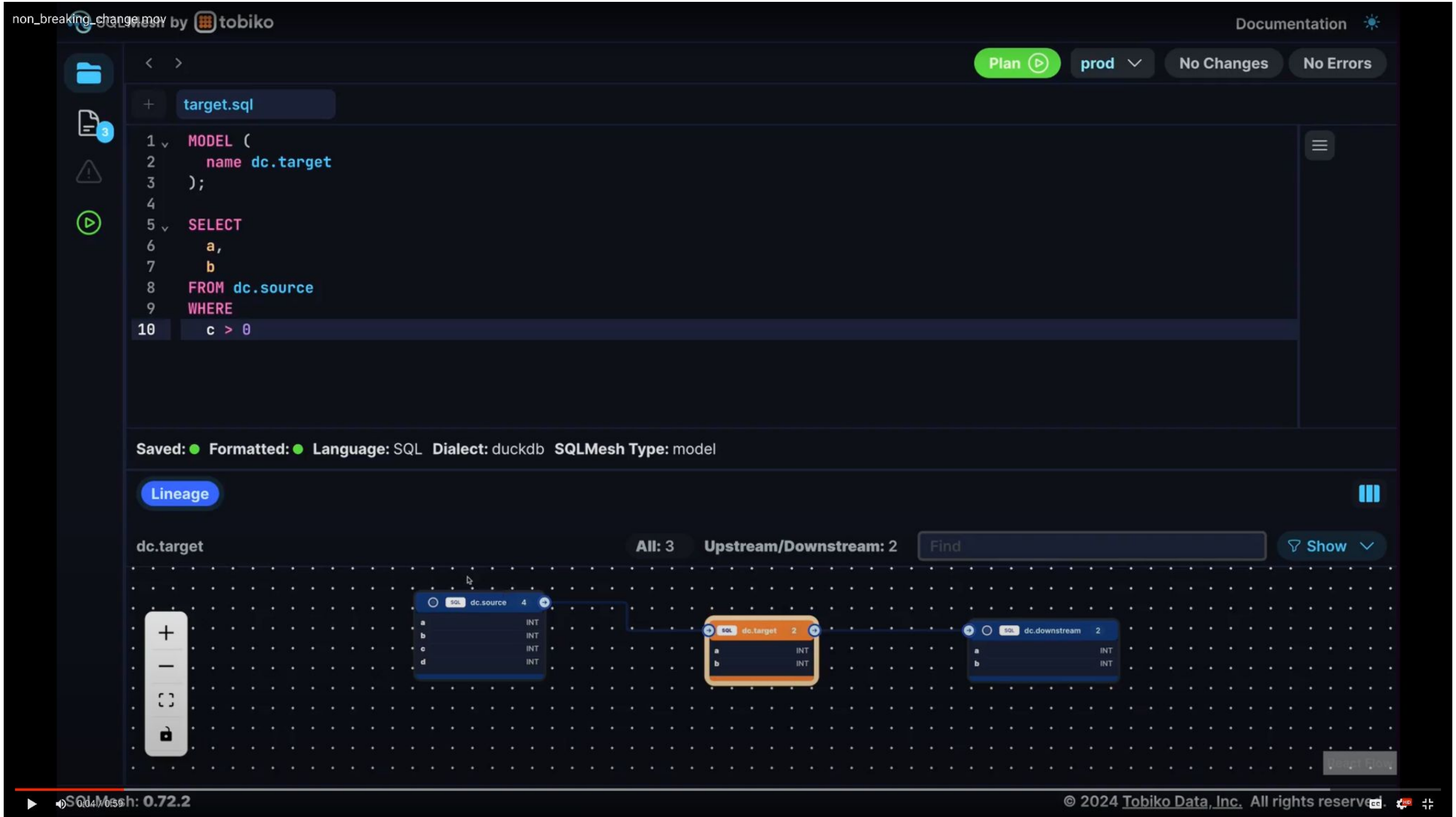
https://tobikodata.com/slack

Scan me!

Thank you!

Backup Slides

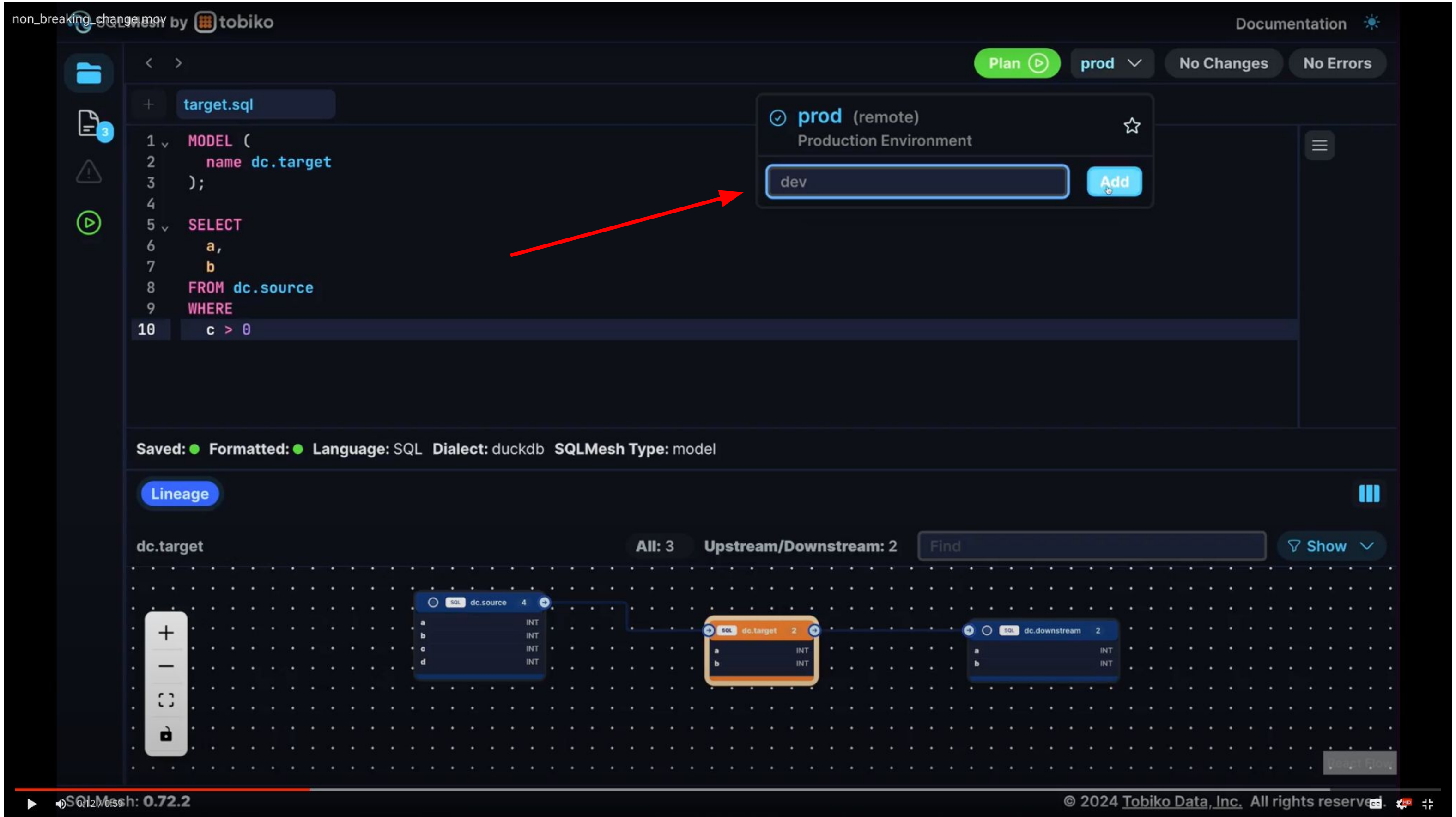
Non-breaking Change



```sql
MODEL (
  name dc.target
);

SELECT
  a,
  b
FROM dc.source
WHERE
  c > 0
```

Saved: ● **Formatted:** ● **Language:** SQL **Dialect:** duckdb **SQLMesh Type:** model

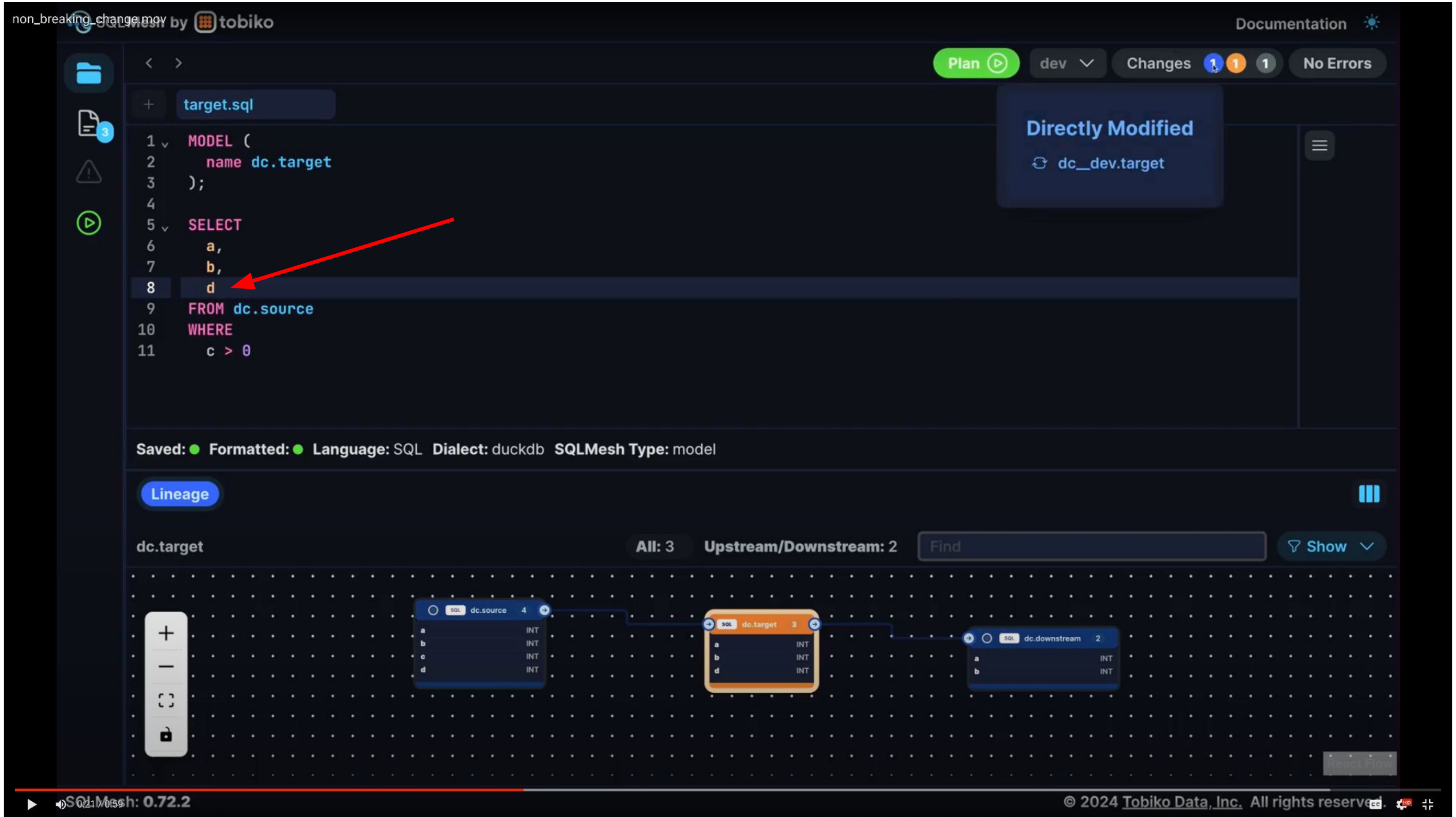
Non-breaking Change



```sql
MODEL (
  name dc.target
);

SELECT
  a,
  b
FROM dc.source
WHERE
  c > 0
```

Saved: ● **Formatted:** ● **Language:** SQL **Dialect:** duckdb **SQLMesh Type:** model

Non-breaking Change



```
MODEL (
  name dc.target
);

SELECT
  a,
  b,
  d
FROM dc.source
WHERE
  c > 0
```

Directly Modified

dc__dev.target

Saved: ● **Formatted:** ● **Language:** SQL **Dialect:** duckdb **SQLMesh Type:** model

dc.target All: 3 Upstream/Downstream: 2

dc.source	4
a	INT
b	INT
c	INT
d	INT

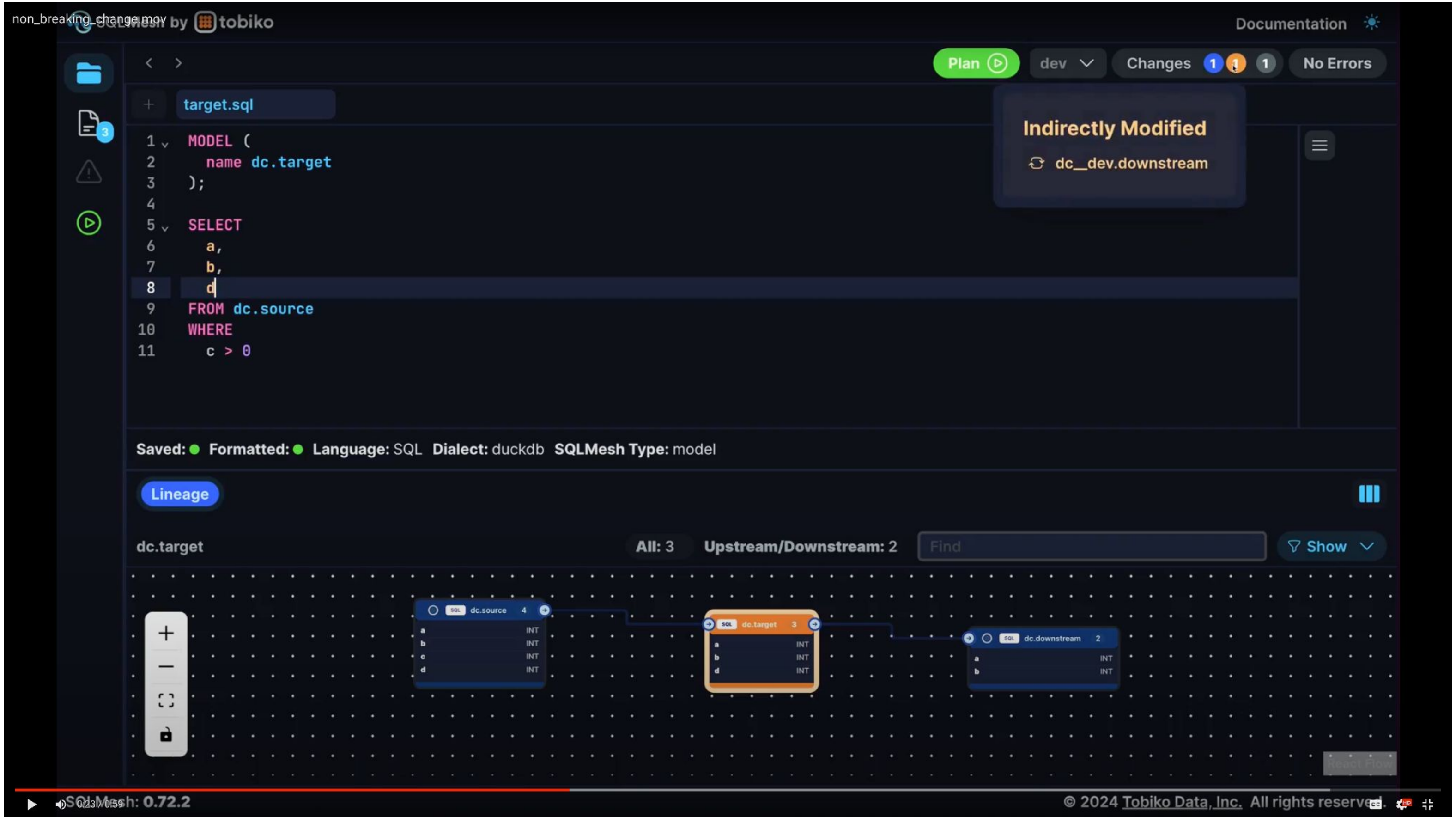
dc.target	3
a	INT
b	INT
d	INT

dc.downstream	2
a	INT
b	INT

SQLMesh: 0.72.2

© 2024 Tobiko Data, Inc. All rights reserved.

Non-breaking Change



non_breaking_change.mov

target.sql

```sql
MODEL (
  name dc.target
);

SELECT
  a,
  b,
  d
FROM dc.source
WHERE
  c > 0
```

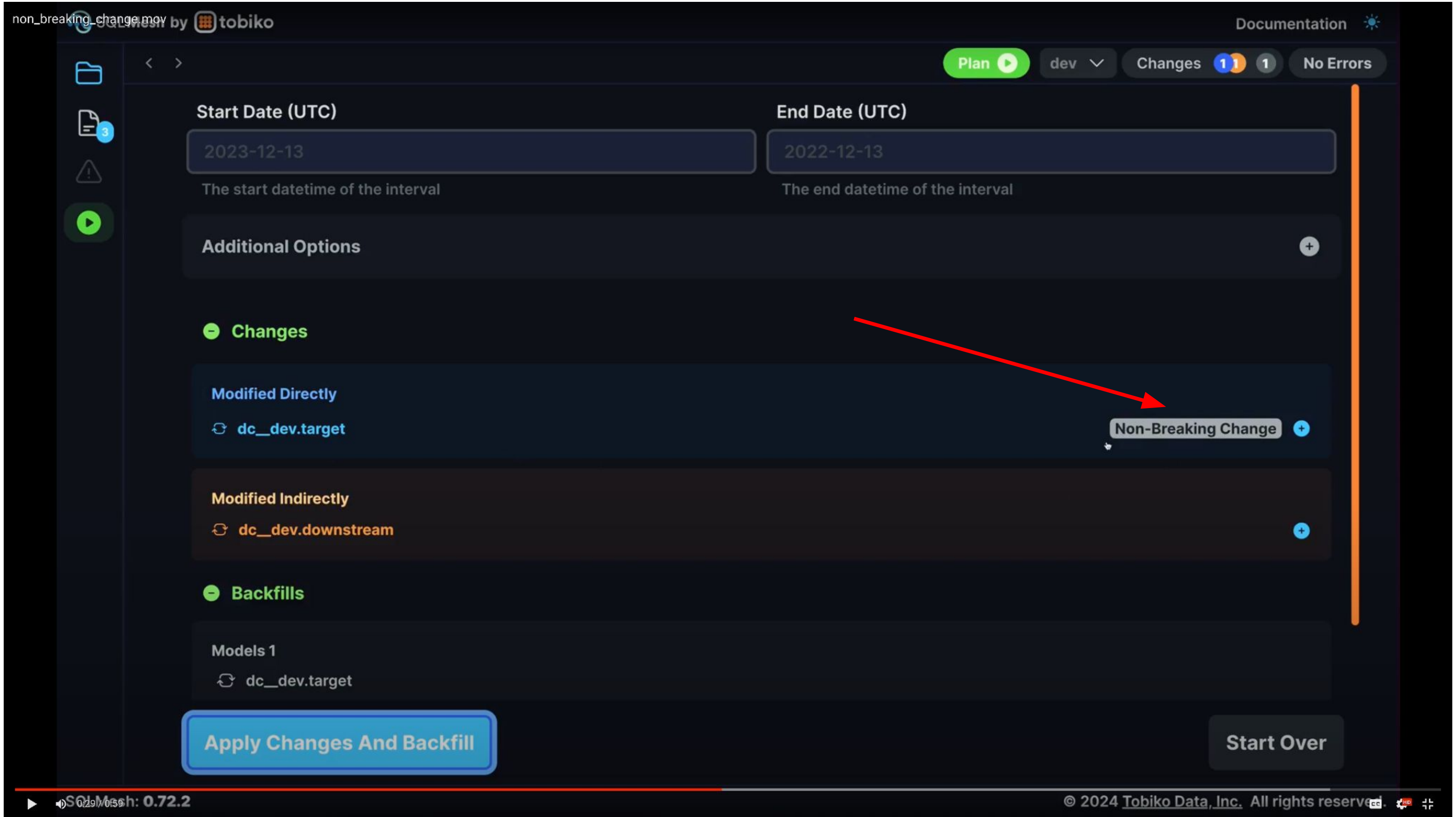
Indirectly Modified

dc__dev.downstream

Saved: ● Formatted: ● Language: SQL Dialect: duckdb SQLMesh Type: model

dc.target All: 3 Upstream/Downstream: 2

Non-breaking Change



Start Date (UTC)

2023-12-13

The start datetime of the interval

End Date (UTC)

2022-12-13

The end datetime of the interval

Additional Options

Changes

Modified Directly

dc__dev.target Non-Breaking Change

Modified Indirectly

dc__dev.downstream

Backfills

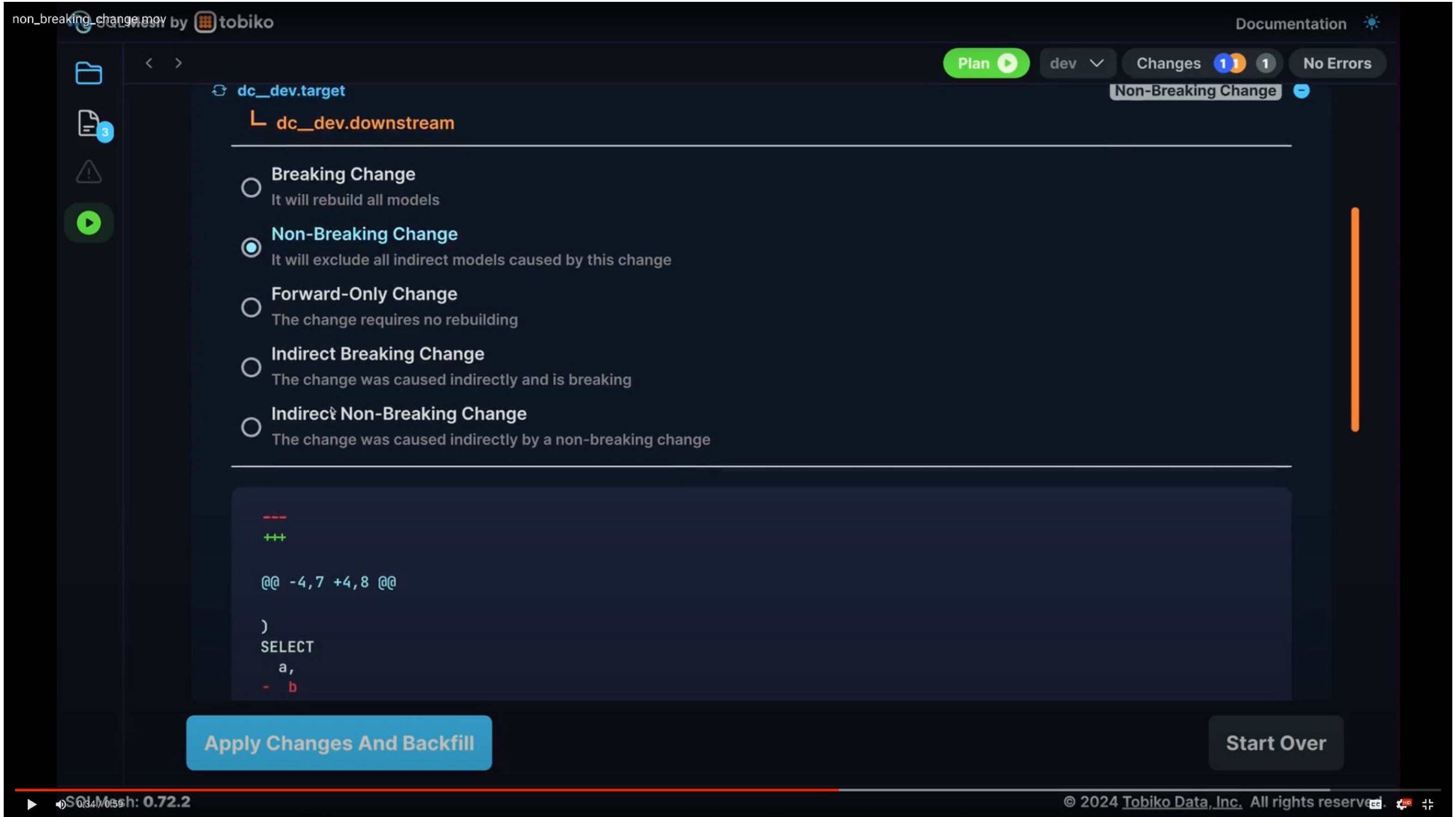
Models 1

dc__dev.target

Apply Changes And Backfill

Start Over

Non-breaking Change



dc__dev.target
└ dc__dev.downstream

Non-Breaking Change

○ **Breaking Change**
It will rebuild all models

● **Non-Breaking Change**
It will exclude all indirect models caused by this change

○ **Forward-Only Change**
The change requires no rebuilding

○ **Indirect Breaking Change**
The change was caused indirectly and is breaking

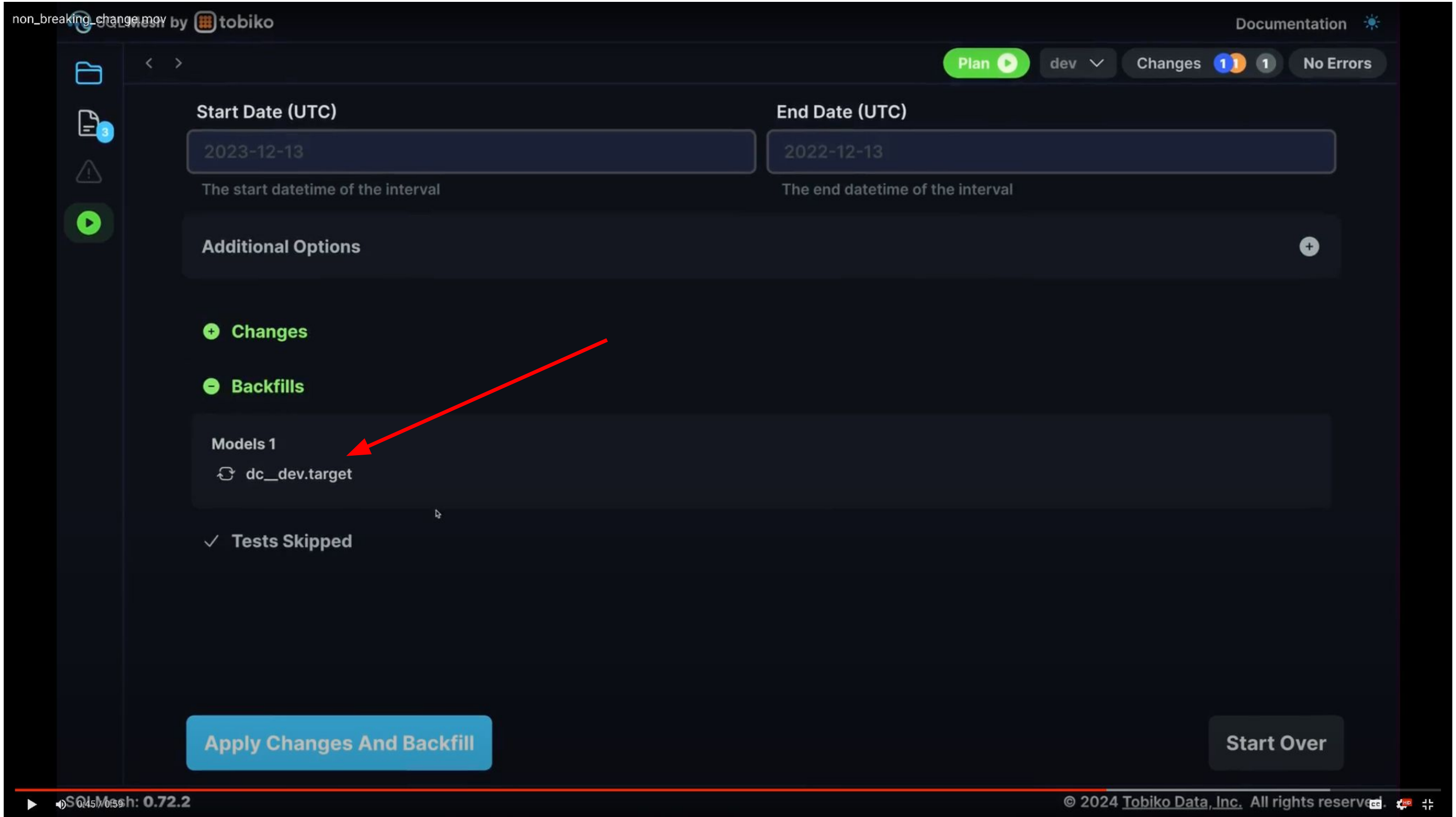
○ **Indirect Non-Breaking Change**
The change was caused indirectly by a non-breaking change

```
---
+++

@@ -4,7 +4,8 @@

)
SELECT
  a,
-  b
```

SQLMesh: 0.72.2

© 2024 Tobiko Data, Inc. All rights reserved.

Non-breaking Change



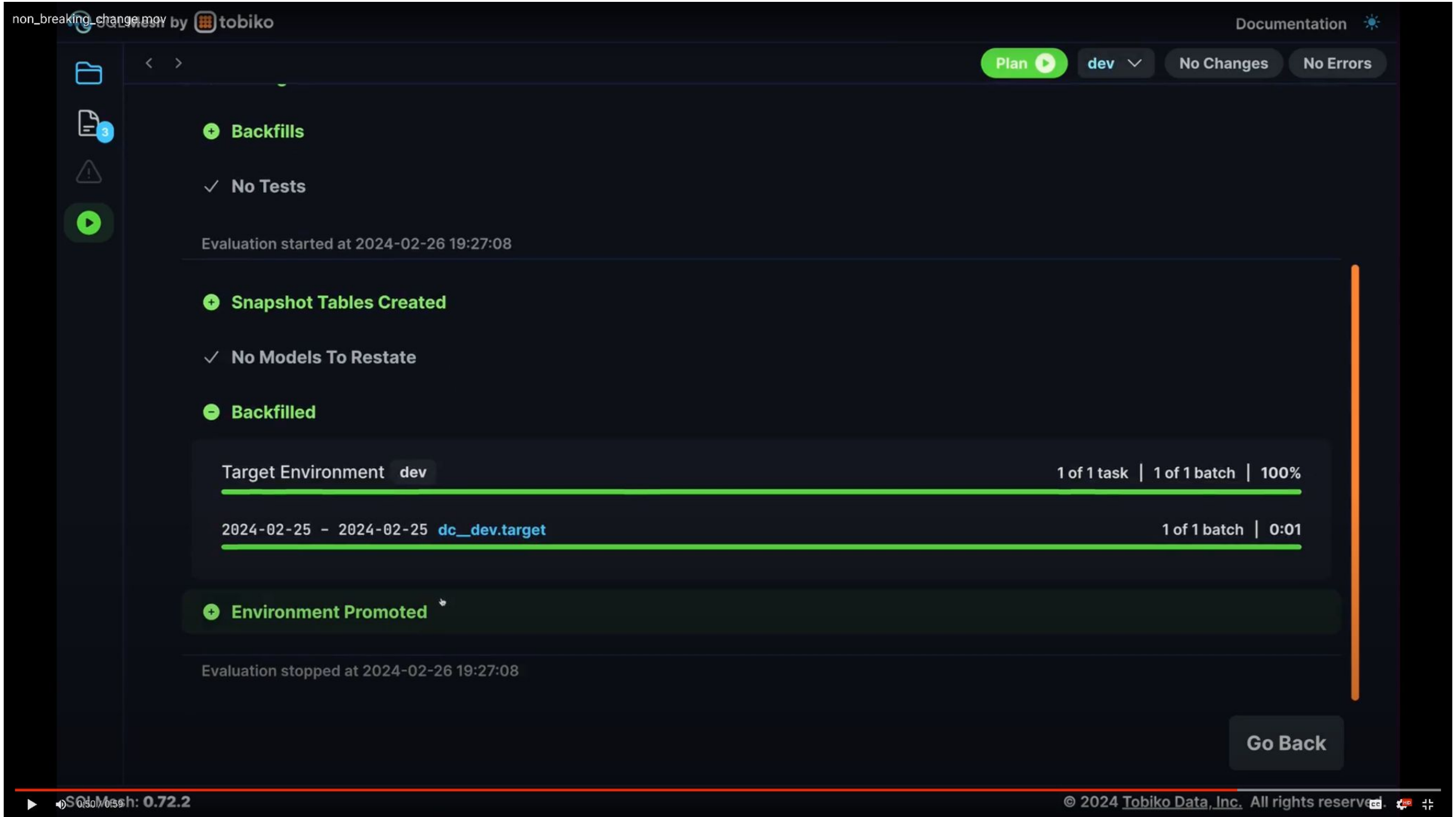
Start Date (UTC)
2023-12-13
The start datetime of the interval

End Date (UTC)
2022-12-13
The end datetime of the interval

Models 1
dc__dev.target

✓ Tests Skipped

Non-breaking Change



Backfills

✓ No Tests

Evaluation started at 2024-02-26 19:27:08

Snapshot Tables Created

✓ No Models To Restate

Backfilled

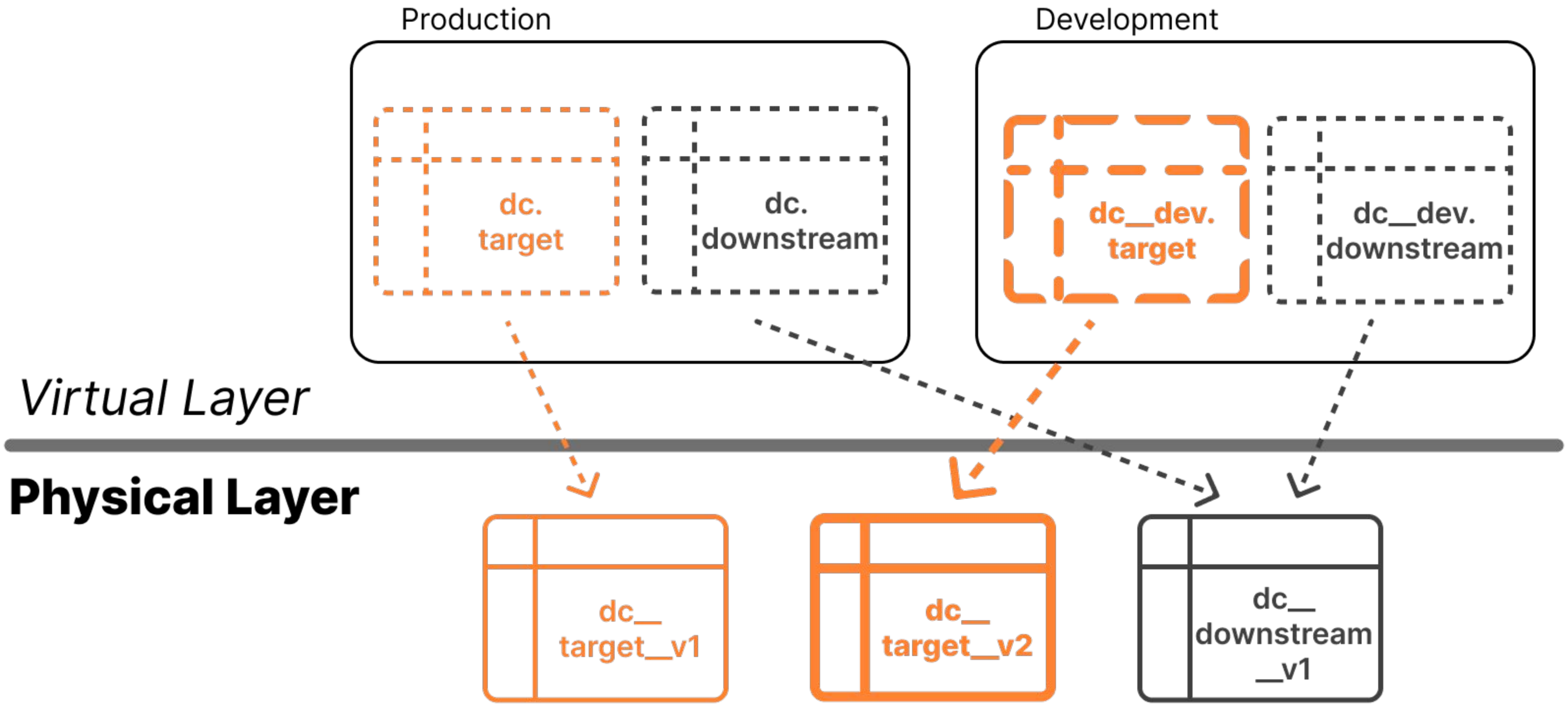
Target Environment dev 1 of 1 task | 1 of 1 batch | 100%

2024-02-25 - 2024-02-25 dc__dev.target 1 of 1 batch | 0:01

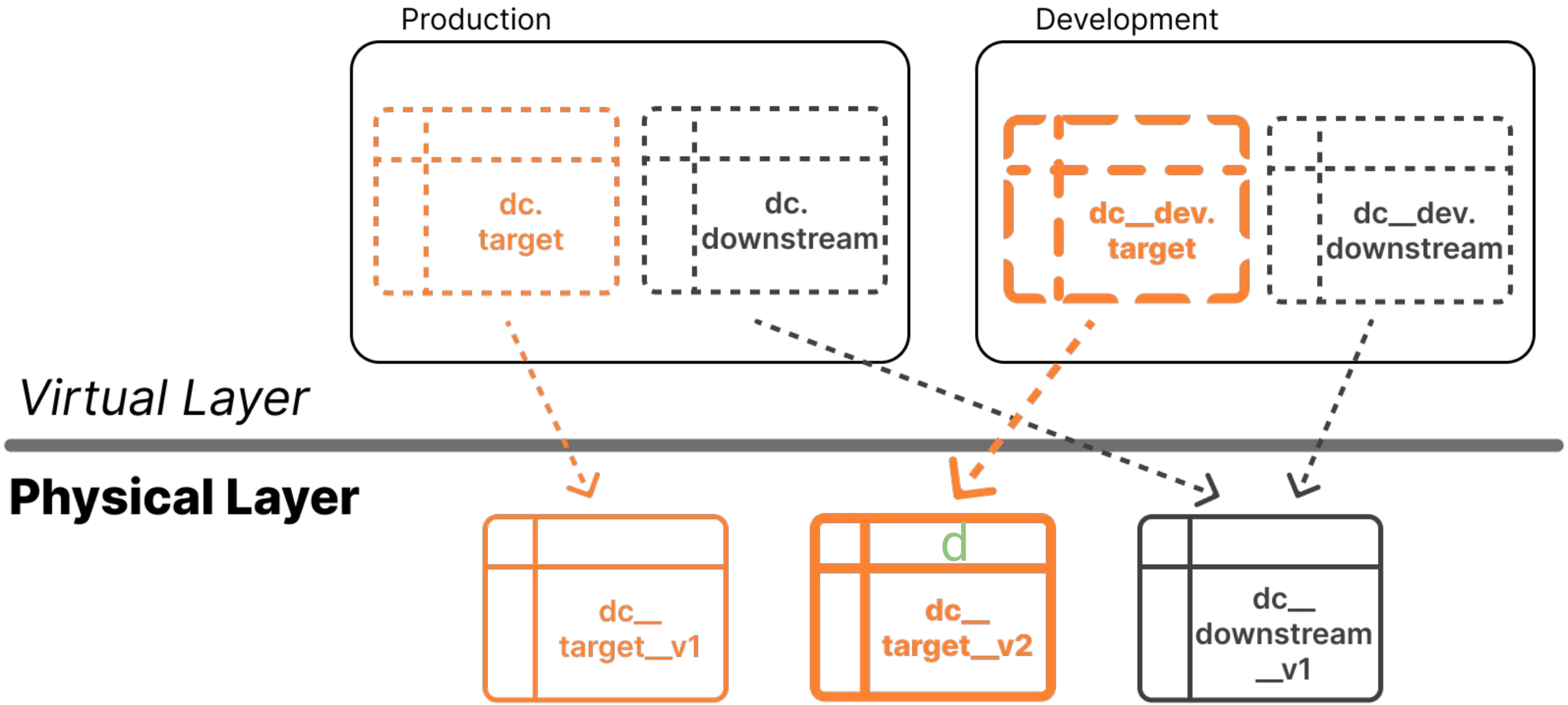
Environment Promoted

Evaluation stopped at 2024-02-26 19:27:08

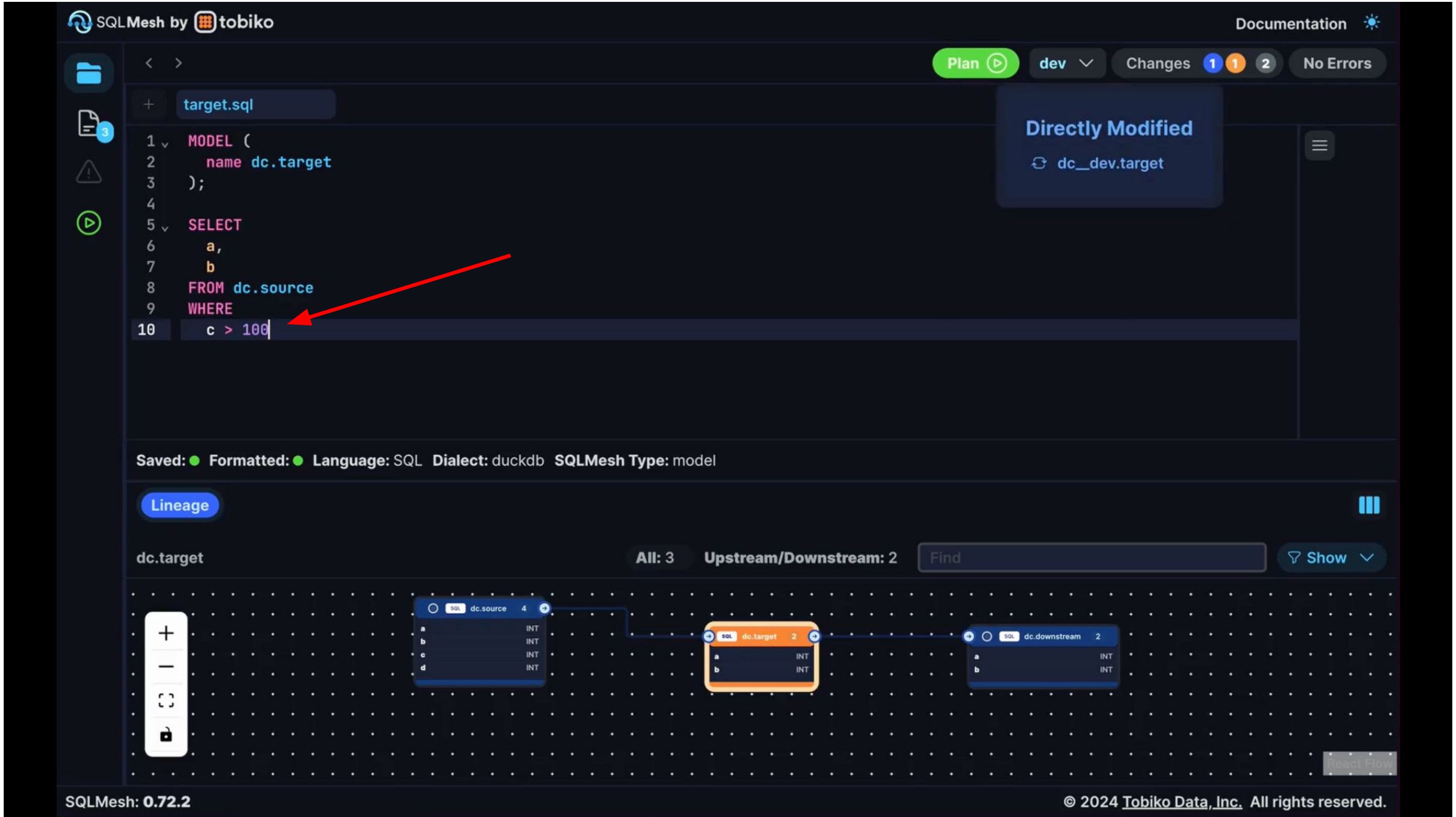
Non-breaking Change



Non-breaking Change



Breaking Change



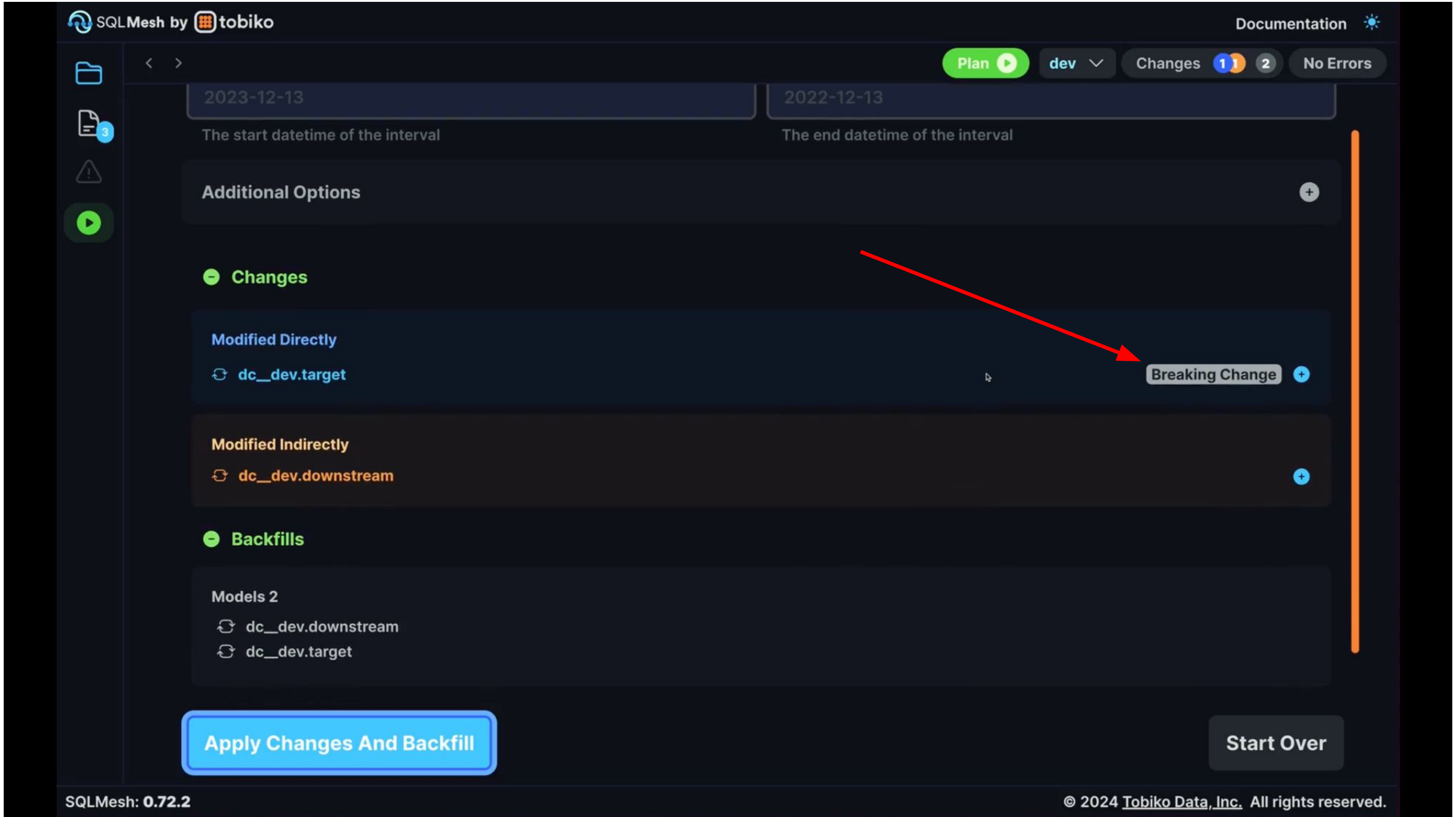
```
MODEL (
  name dc.target
);

SELECT
  a,
  b
FROM dc.source
WHERE
  c > 100
```

Saved: ● **Formatted:** ● **Language:** SQL **Dialect:** duckdb **SQLMesh Type:** model

SQLMesh: 0.72.2

© 2024 Tobiko Data, Inc. All rights reserved.

Breaking Change



SQLMesh by tobiko

Documentation

Plan

dev

Changes 1 1 2

No Errors

2023-12-13

The start datetime of the interval

2022-12-13

The end datetime of the interval

Additional Options

Changes

Modified Directly

dc__dev.target

Breaking Change

Modified Indirectly

dc__dev.downstream

Backfills

Models 2

dc__dev.downstream

dc__dev.target

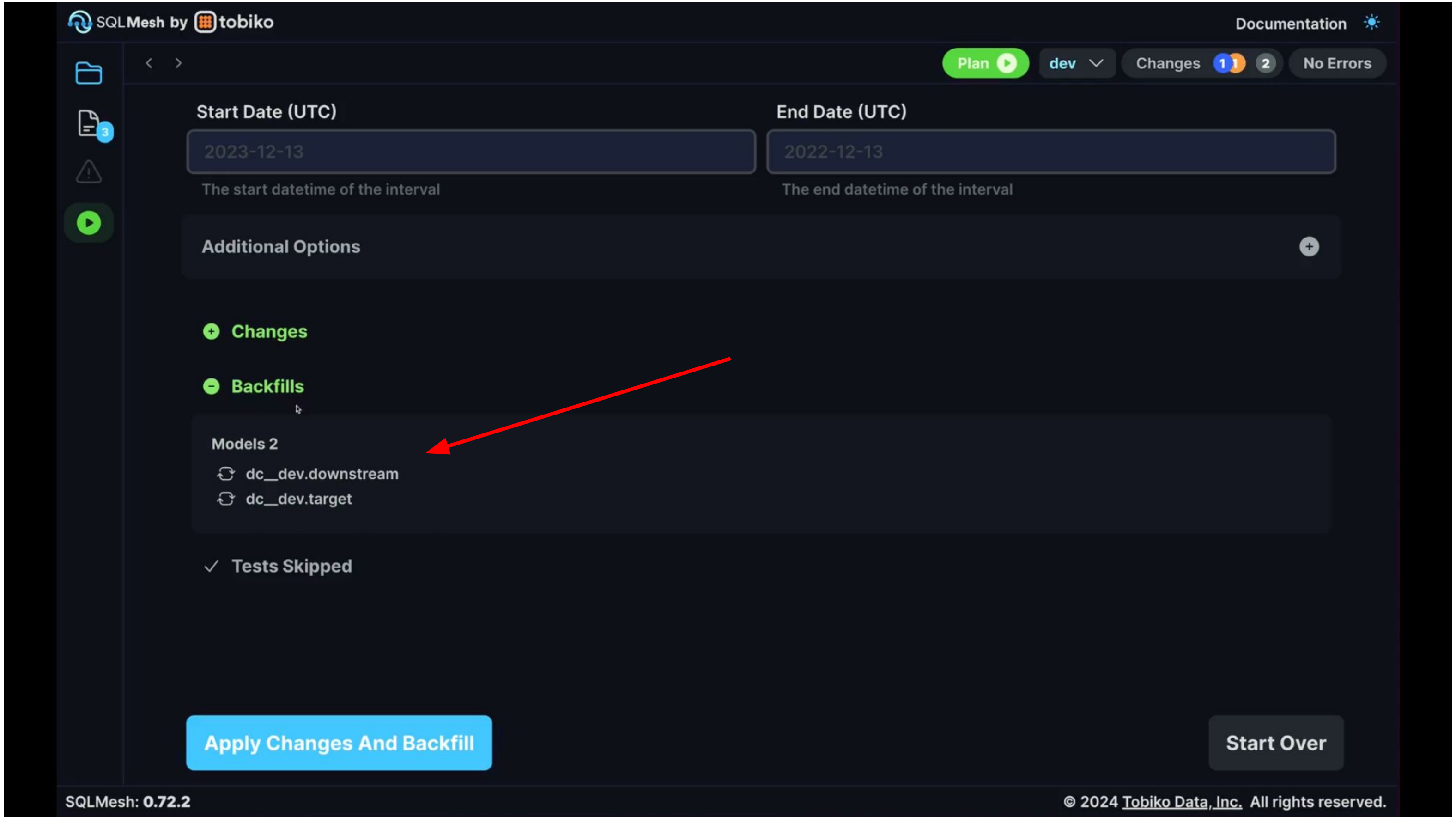
Apply Changes And Backfill

Start Over

SQLMesh: 0.72.2

© 2024 Tobiko Data, Inc. All rights reserved.

Breaking Change



Start Date (UTC)

2023-12-13

The start datetime of the interval

End Date (UTC)

2022-12-13

The end datetime of the interval

Additional Options

Changes

Backfills

Models 2

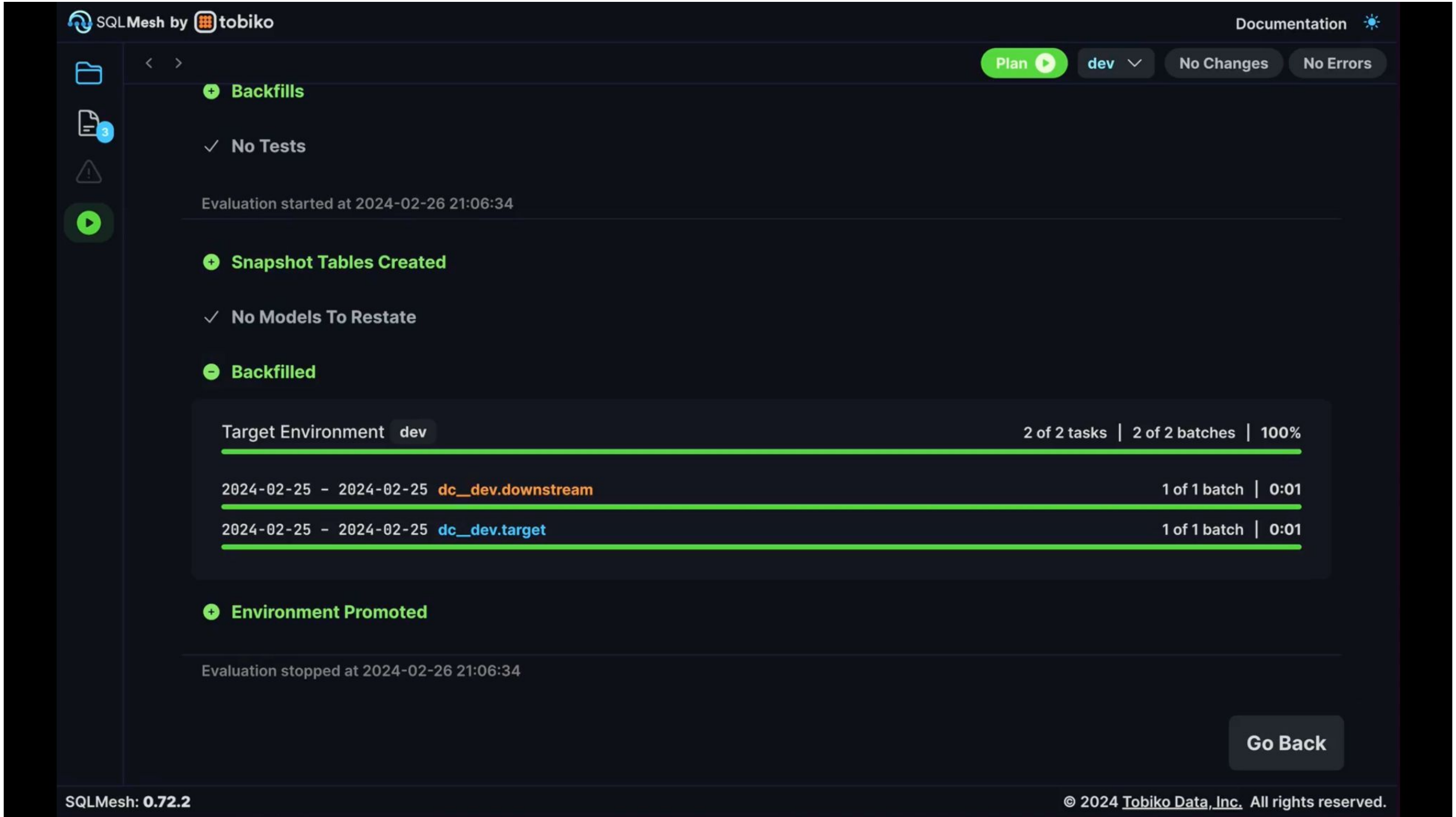
- dc__dev.downstream
- dc__dev.target

✓ Tests Skipped

SQLMesh: 0.72.2

© 2024 Tobiko Data, Inc. All rights reserved.

Breaking Change



Backfills

✓ No Tests

Evaluation started at 2024-02-26 21:06:34

Snapshot Tables Created

✓ No Models To Restate

Backfilled

Target Environment dev — 2 of 2 tasks | 2 of 2 batches | 100%

2024-02-25 - 2024-02-25 dc__dev.downstream — 1 of 1 batch | 0:01

2024-02-25 - 2024-02-25 dc__dev.target — 1 of 1 batch | 0:01

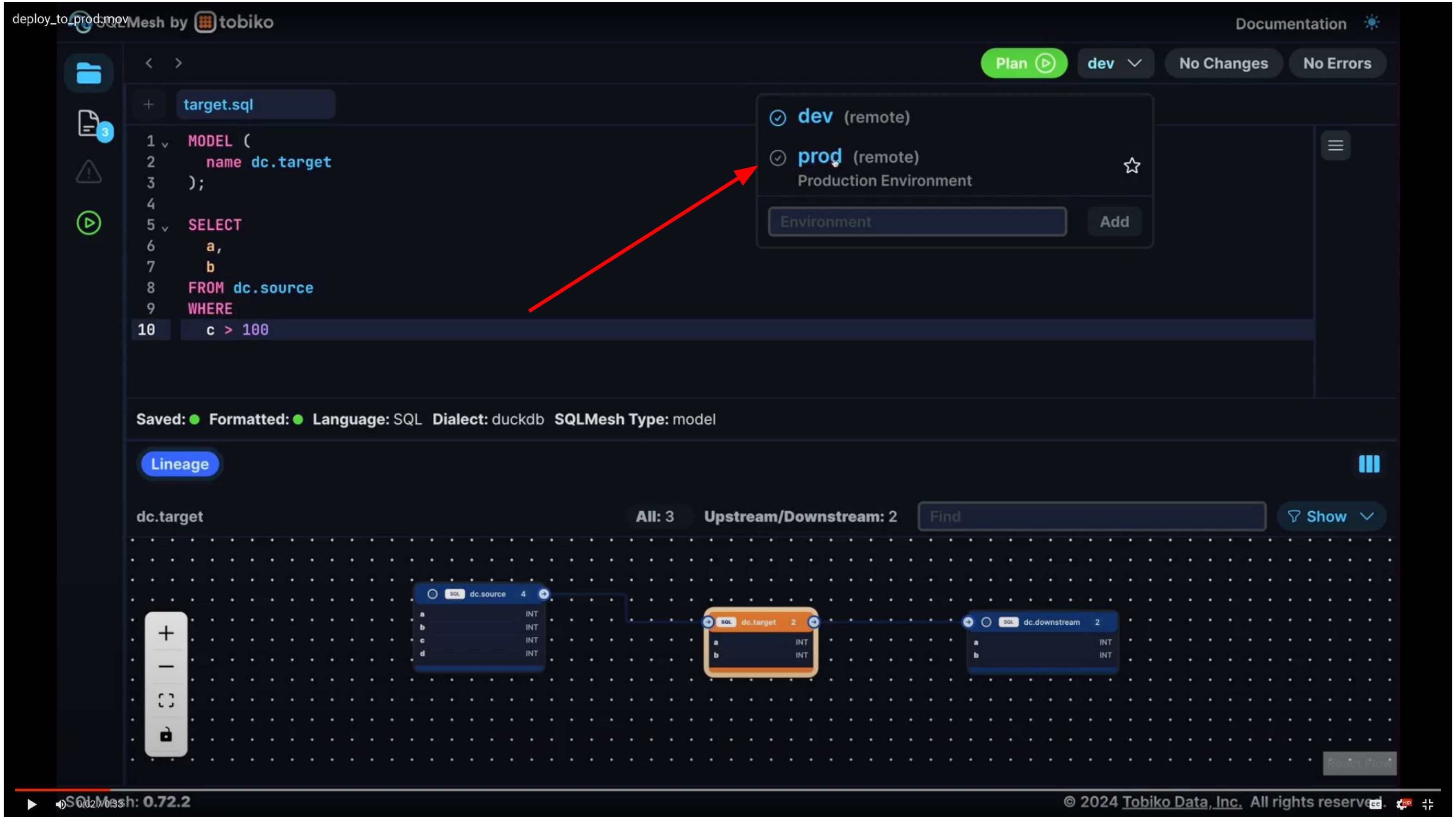
Environment Promoted

Evaluation stopped at 2024-02-26 21:06:34

SQLMesh: 0.72.2

© 2024 Tobiko Data, Inc. All rights reserved.

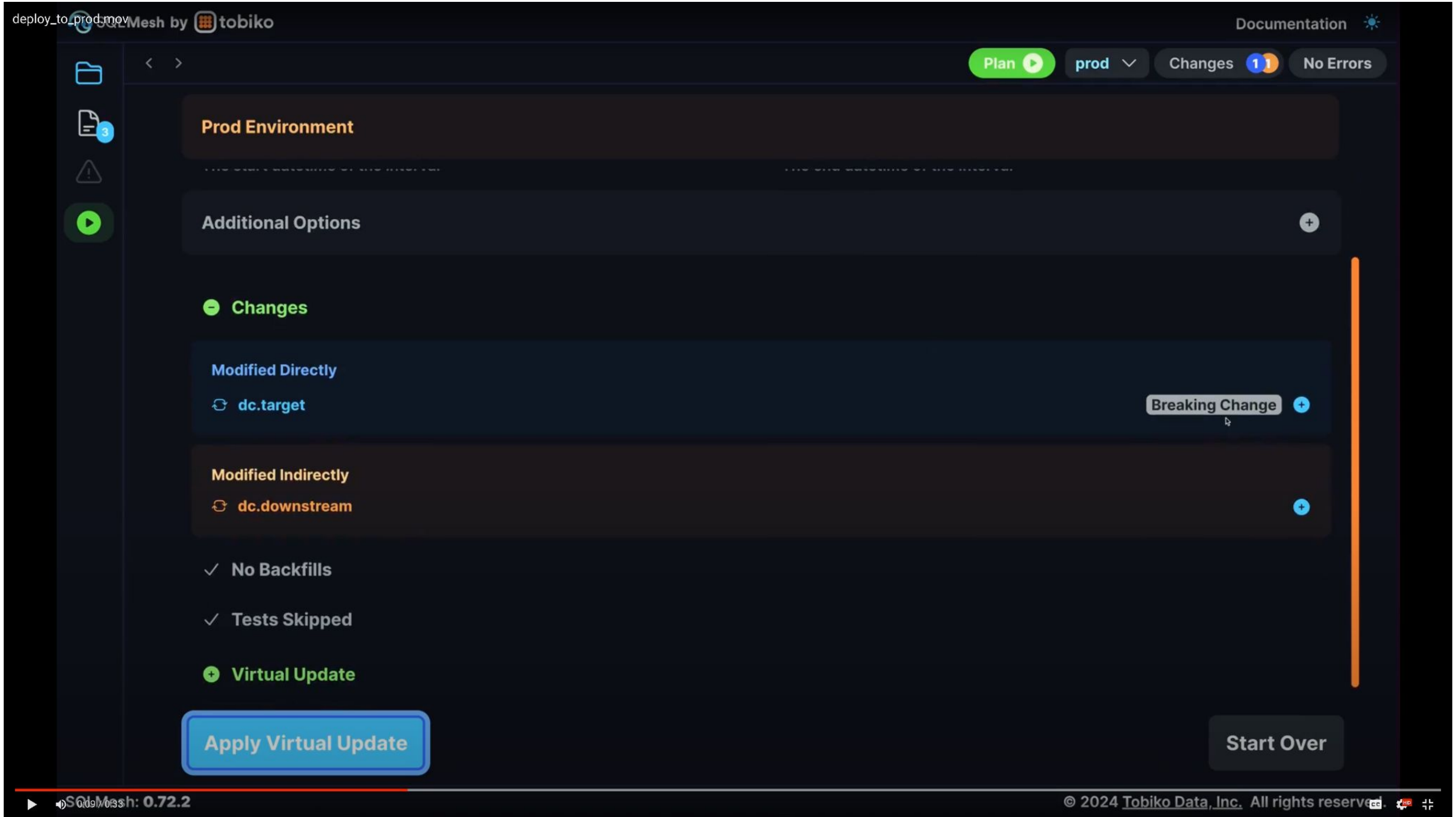
Deploy to Prod



```
MODEL (
  name dc.target
);

SELECT
  a,
  b
FROM dc.source
WHERE
  c > 100
```

Saved: ● Formatted: ● Language: SQL Dialect: duckdb SQLMesh Type: model

Deploy to Prod



Prod Environment

Additional Options

Changes

Modified Directly

dc.target — Breaking Change

Modified Indirectly

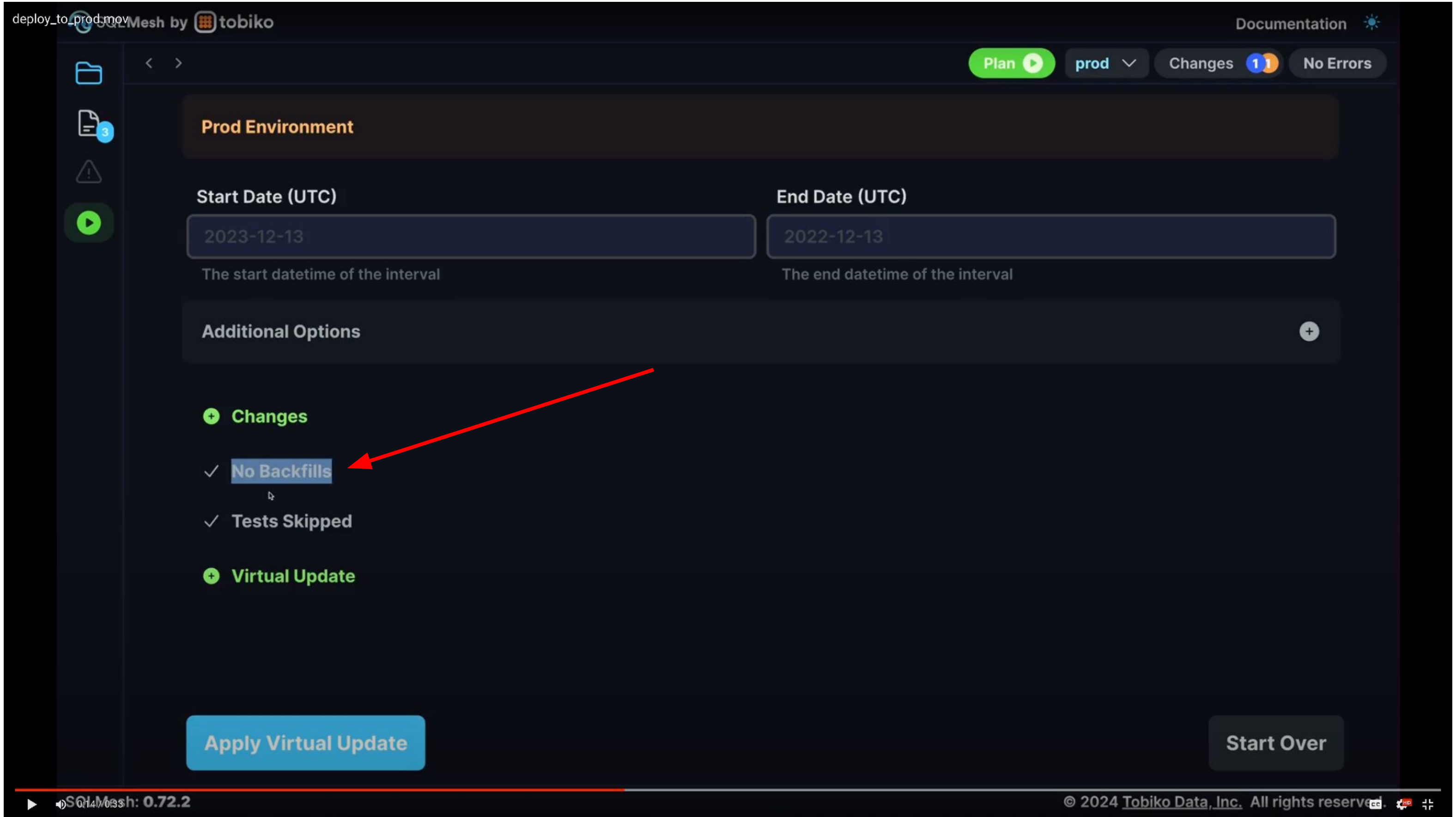
dc.downstream

✓ No Backfills

✓ Tests Skipped

Virtual Update

Deploy to Prod



Prod Environment

Start Date (UTC): 2023-12-13

The start datetime of the interval

End Date (UTC): 2022-12-13

The end datetime of the interval

Additional Options

Changes

✓ No Backfills

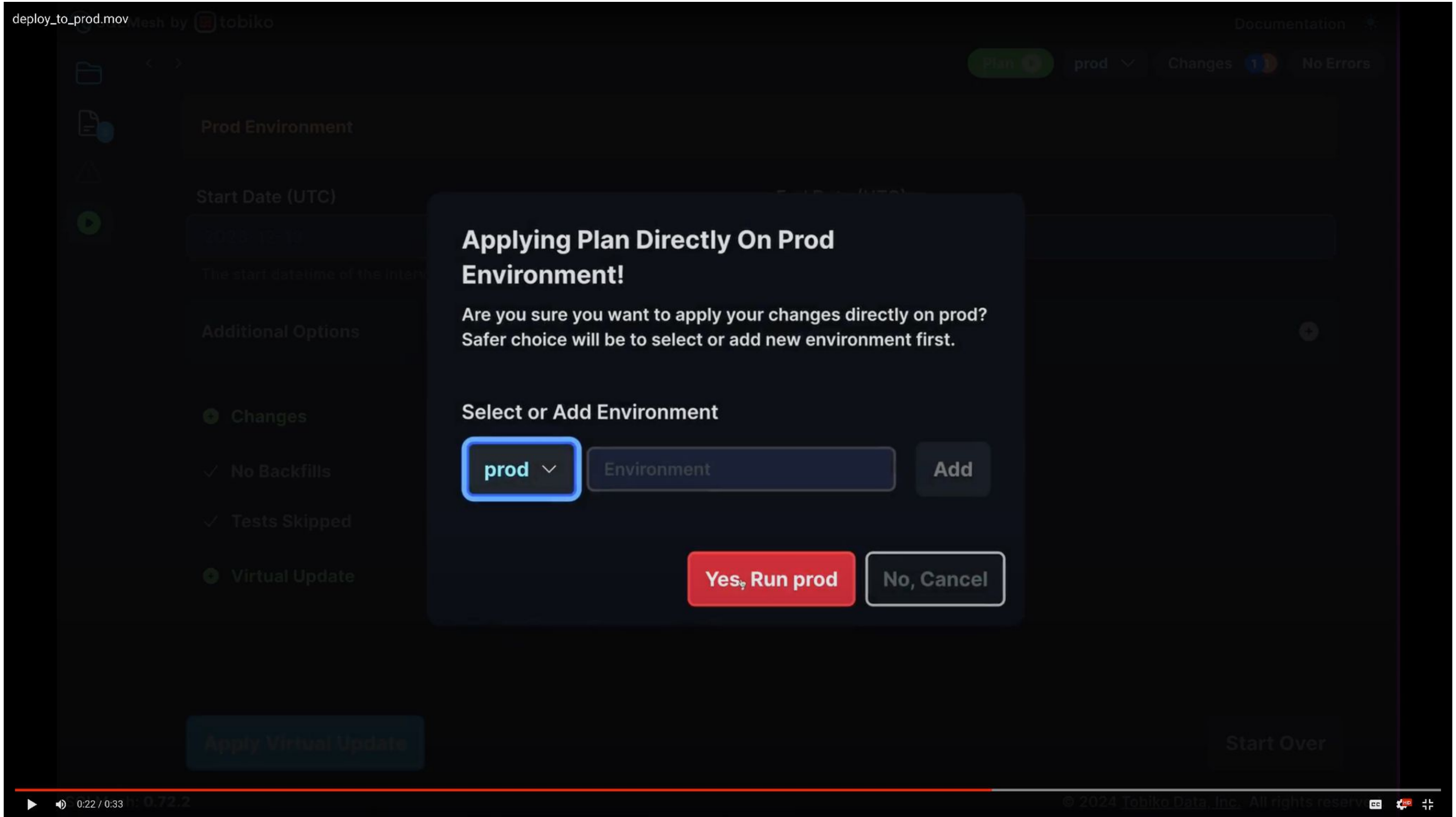
✓ Tests Skipped

Virtual Update

SQLMesh: 0.72.2

© 2024 Tobiko Data, Inc. All rights reserved.

Deploy to Prod



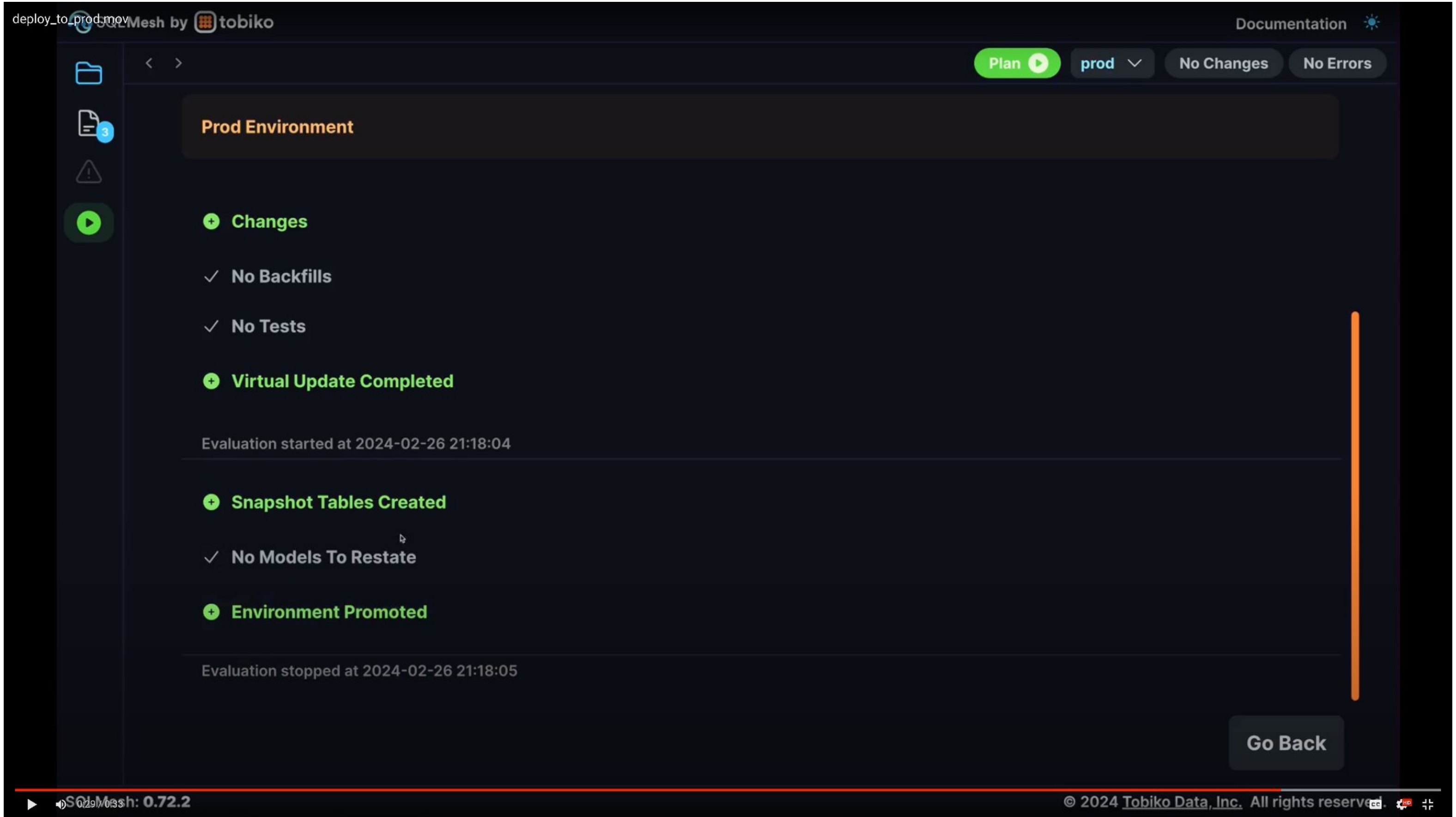
deploy_to_prod.mov

Applying Plan Directly On Prod Environment!

Are you sure you want to apply your changes directly on prod? Safer choice will be to select or add new environment first.

Select or Add Environment

Deploy to Prod



Prod Environment

- Changes
- ✓ No Backfills
- ✓ No Tests
- Virtual Update Completed

Evaluation started at 2024-02-26 21:18:04

- Snapshot Tables Created
- ✓ No Models To Restate
- Environment Promoted

Evaluation stopped at 2024-02-26 21:18:05

tobiko

Scan me!

Join us on Slack!

https://tobikodata.com/slack

Thank you!