



Evolving Data Pipelines at Scale

January 15, 2024

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



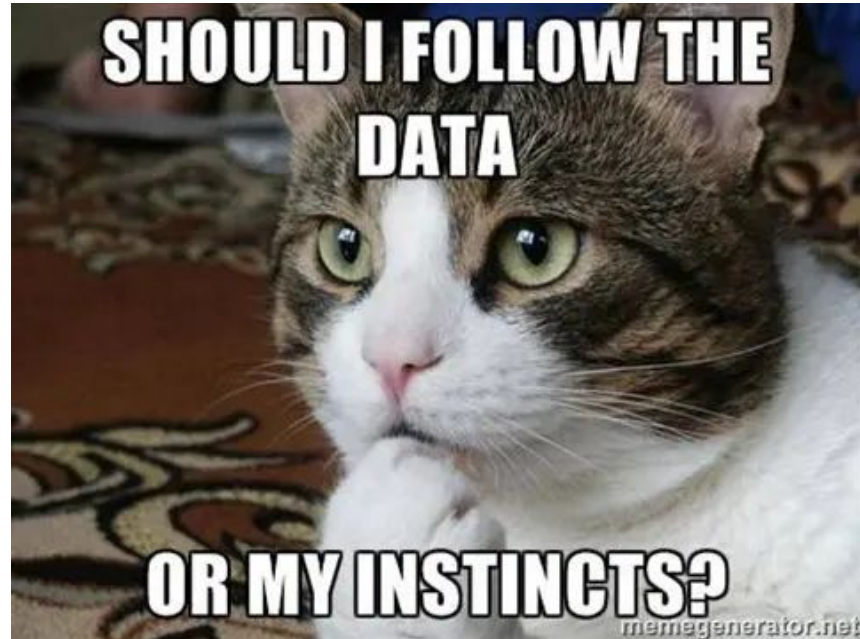
Why am I here?

- Populating dev environments with data is inefficient



Why am I here?

- Development iterations are slow



Why am I here?

- Development iterations are slow

THE DATA CANT BE WRONG



IF YOU RUN THE JOB WITH NO DATA

Why am I here?

- Production deployments are ad-hoc and error-prone



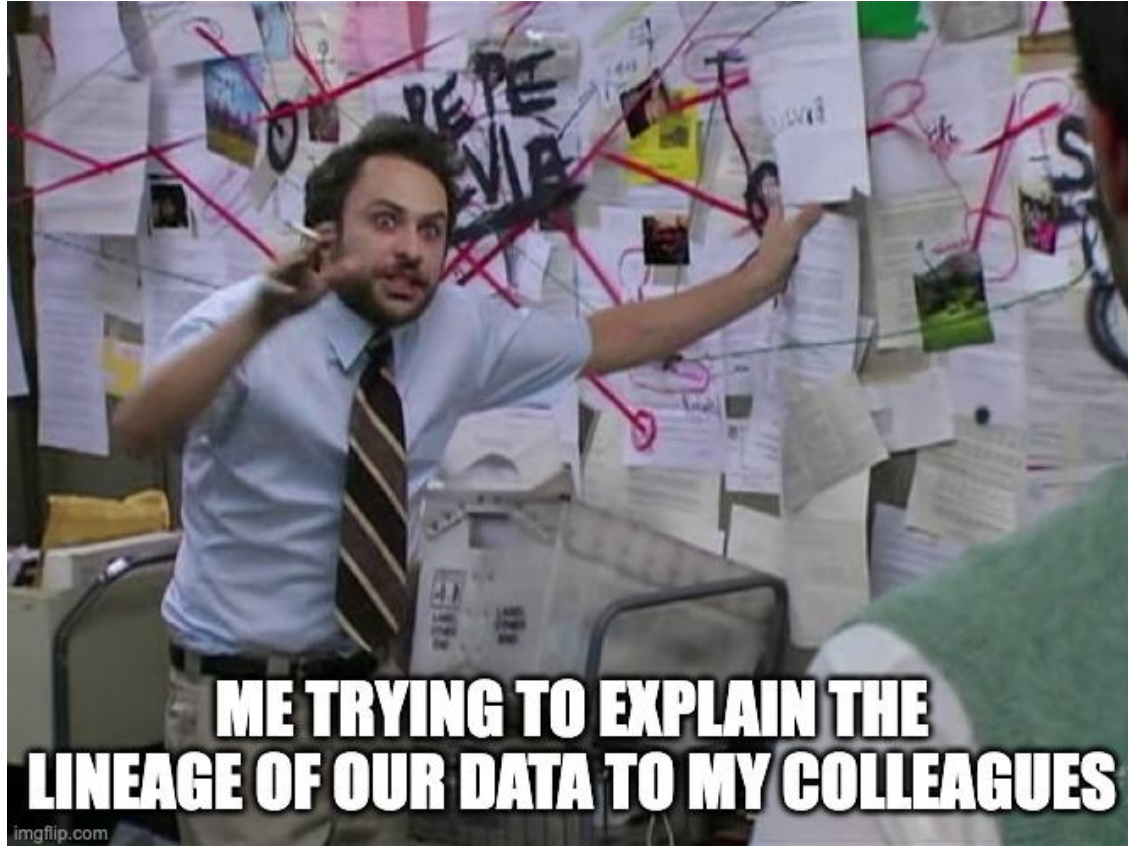
Why am I here?



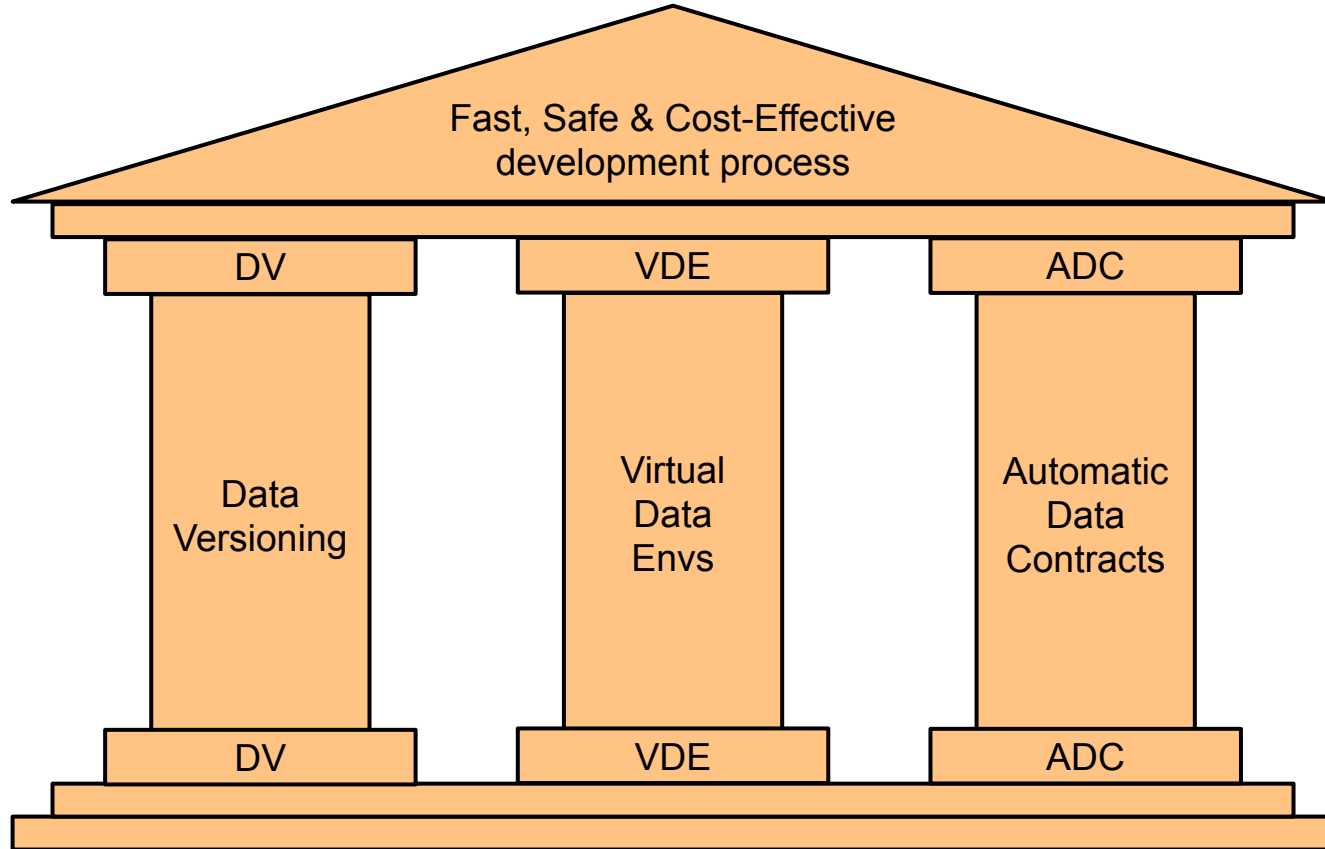
Scale



Scale Organizational Complexity



Three Pillars of Data Productivity



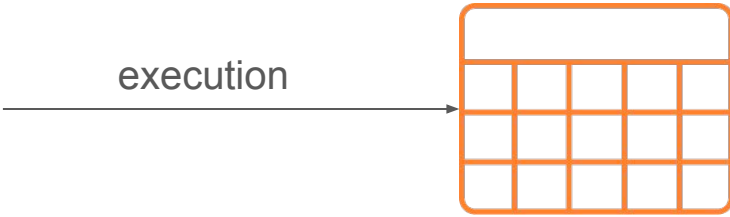
Model: T in ETL

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

Model

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

execution



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

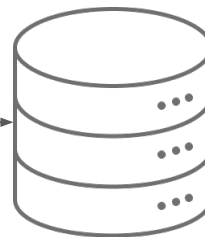
Model Snapshot

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

Model Fingerprint

- Data hash
- Metadata hash

Model
Snapshot

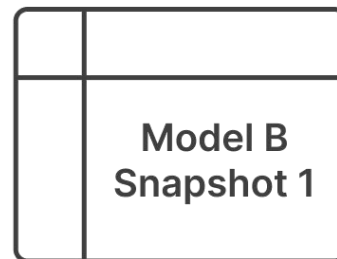
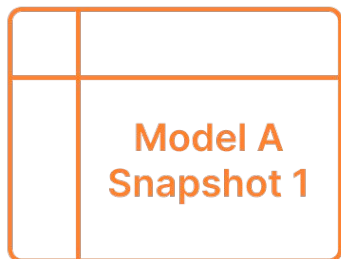


Snapshot Properties

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

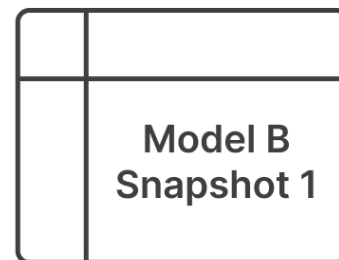
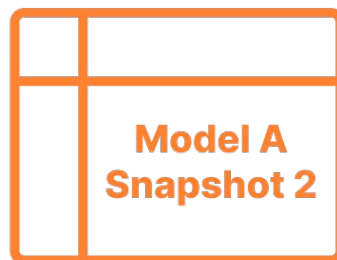
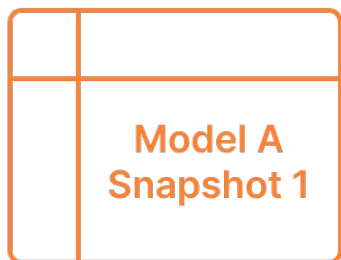
Snapshots in the Data Warehouse

Data Warehouse

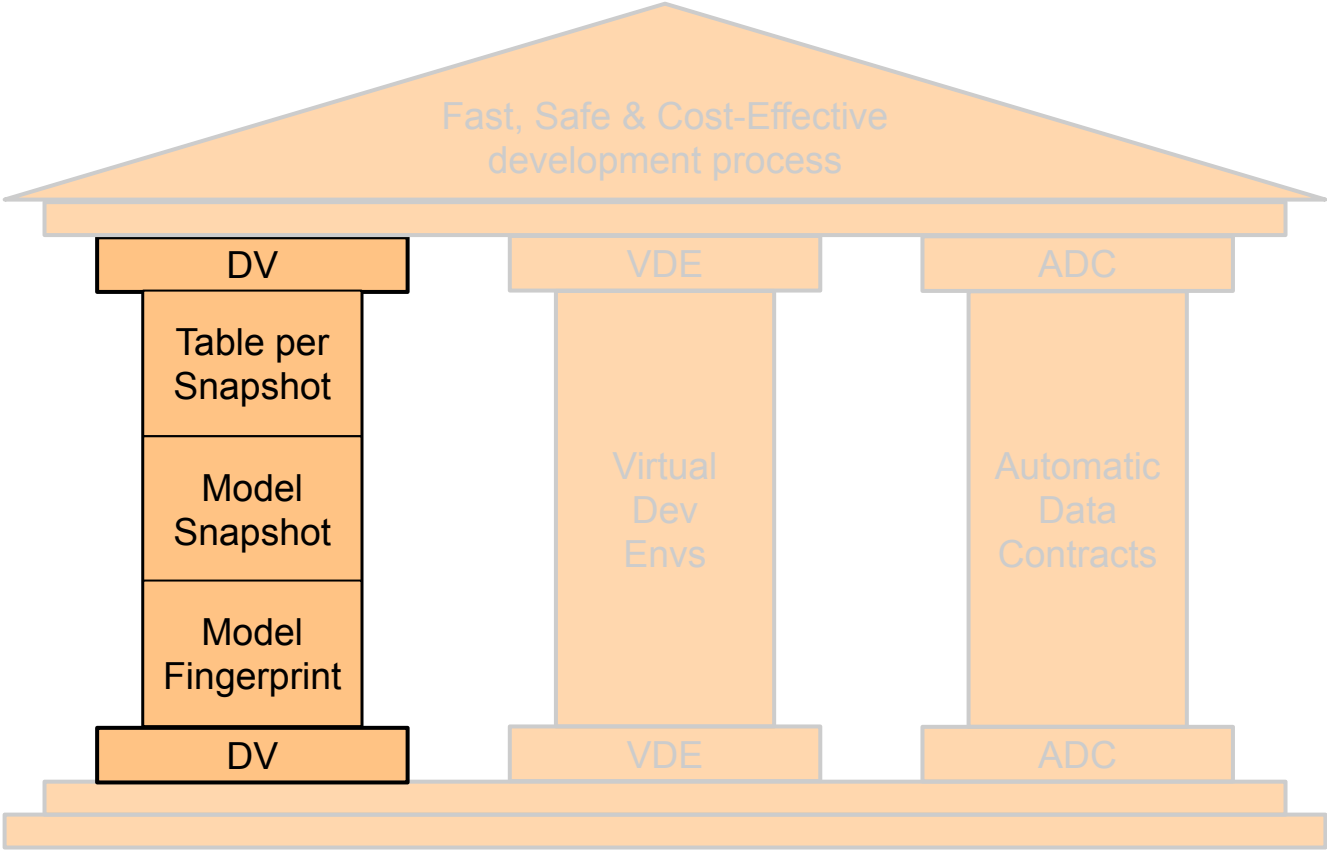


Snapshots in the Data Warehouse

Data Warehouse



Pillar of Data Versioning



Virtual Data Environments

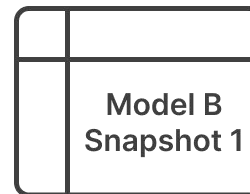
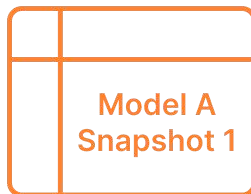
Virtual Layer

Physical Layer

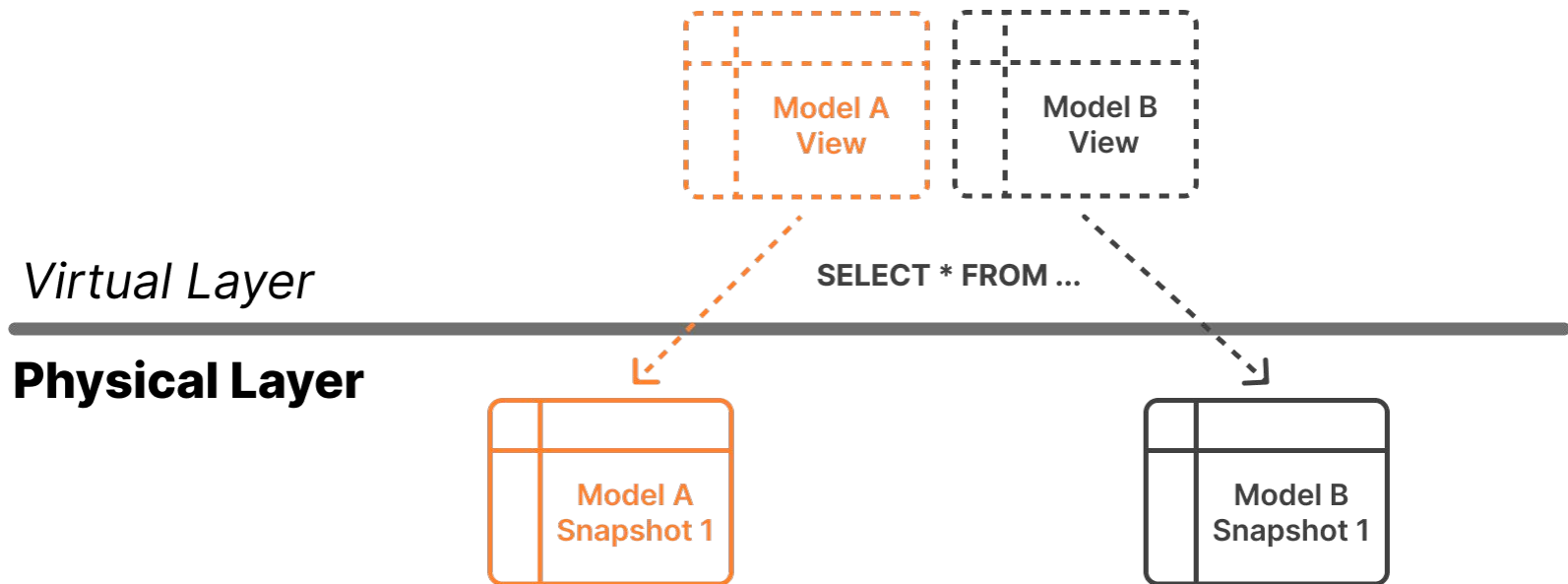
Virtual Data Environments

Virtual Layer

Physical Layer

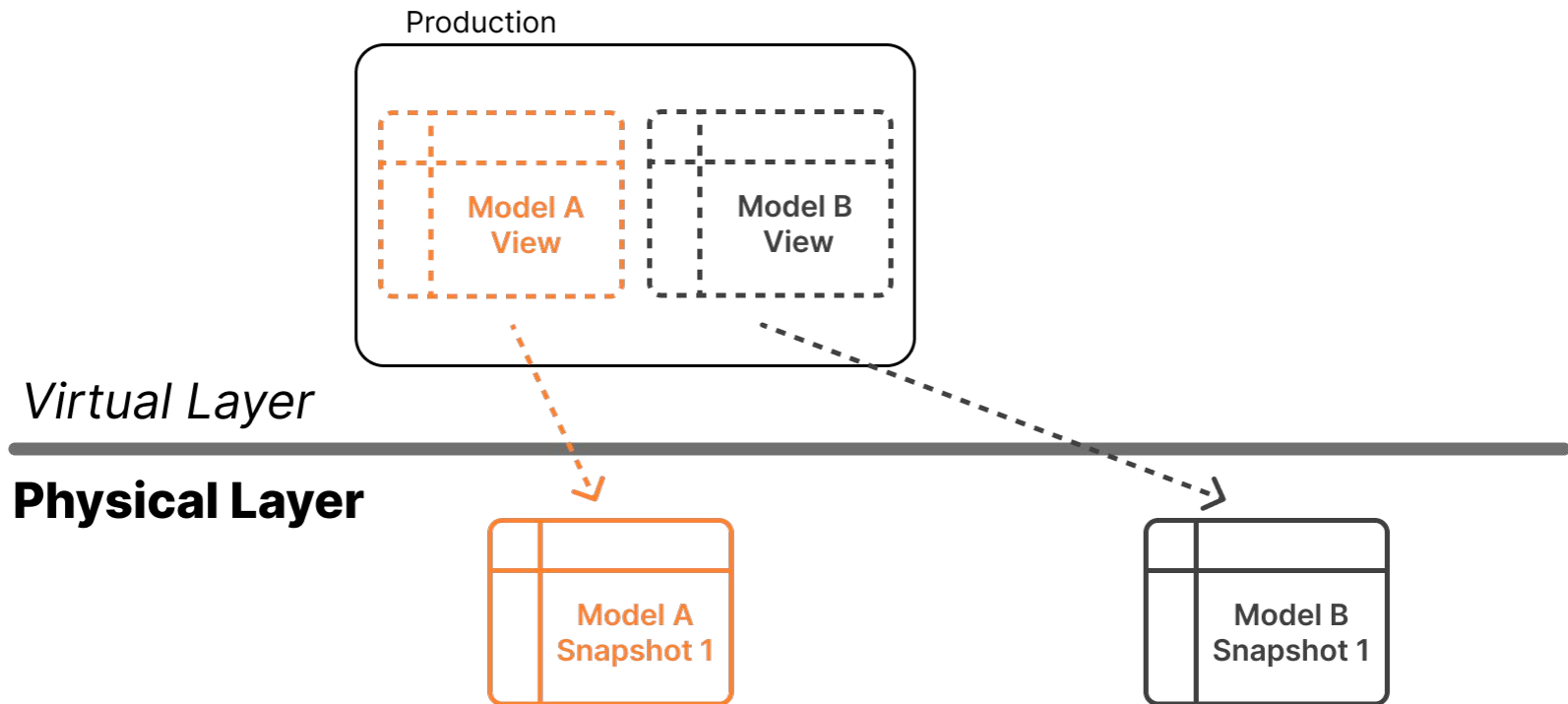


Virtual Data Environments



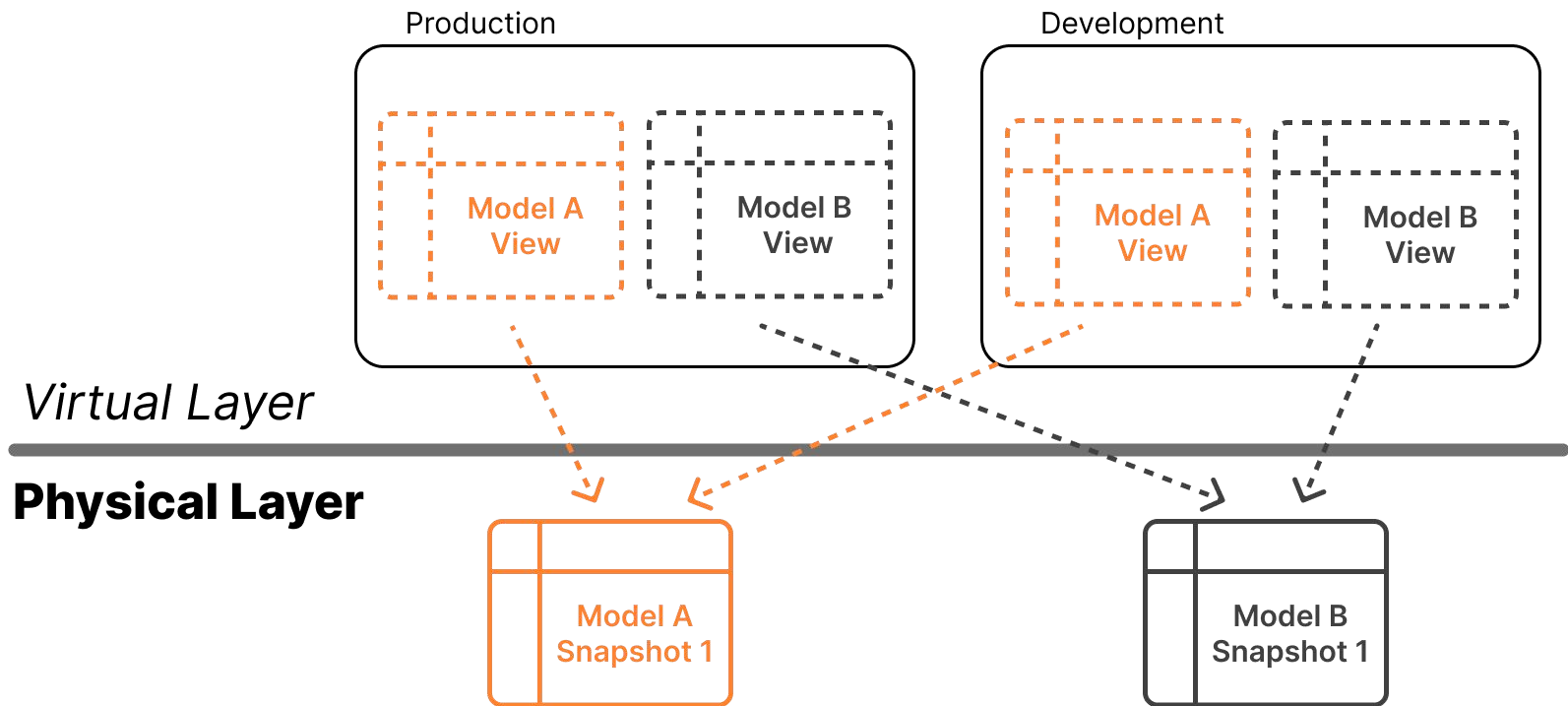
Virtual Data Environments: Flow

1. Starting production environment



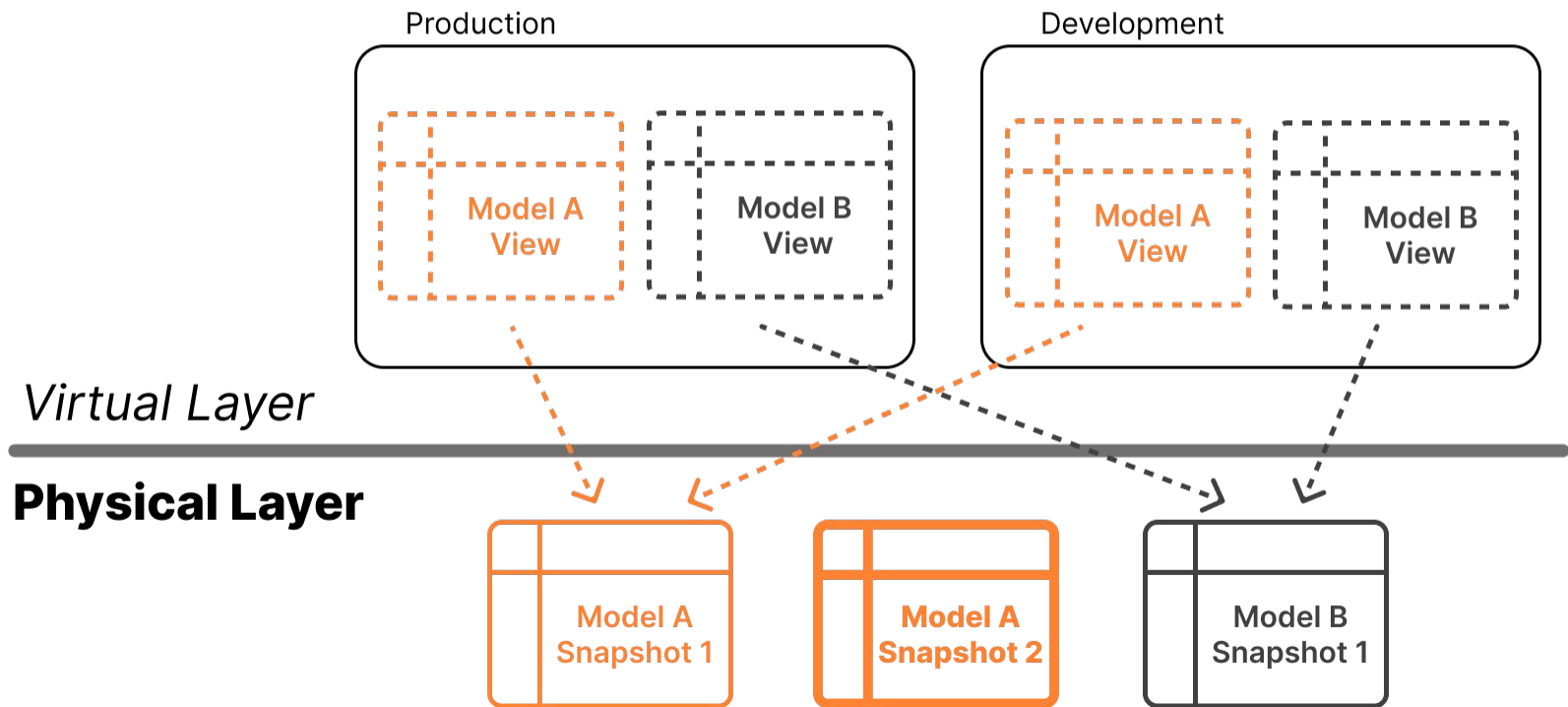
Virtual Data Environments: Flow

2. Create development as a mirror of production



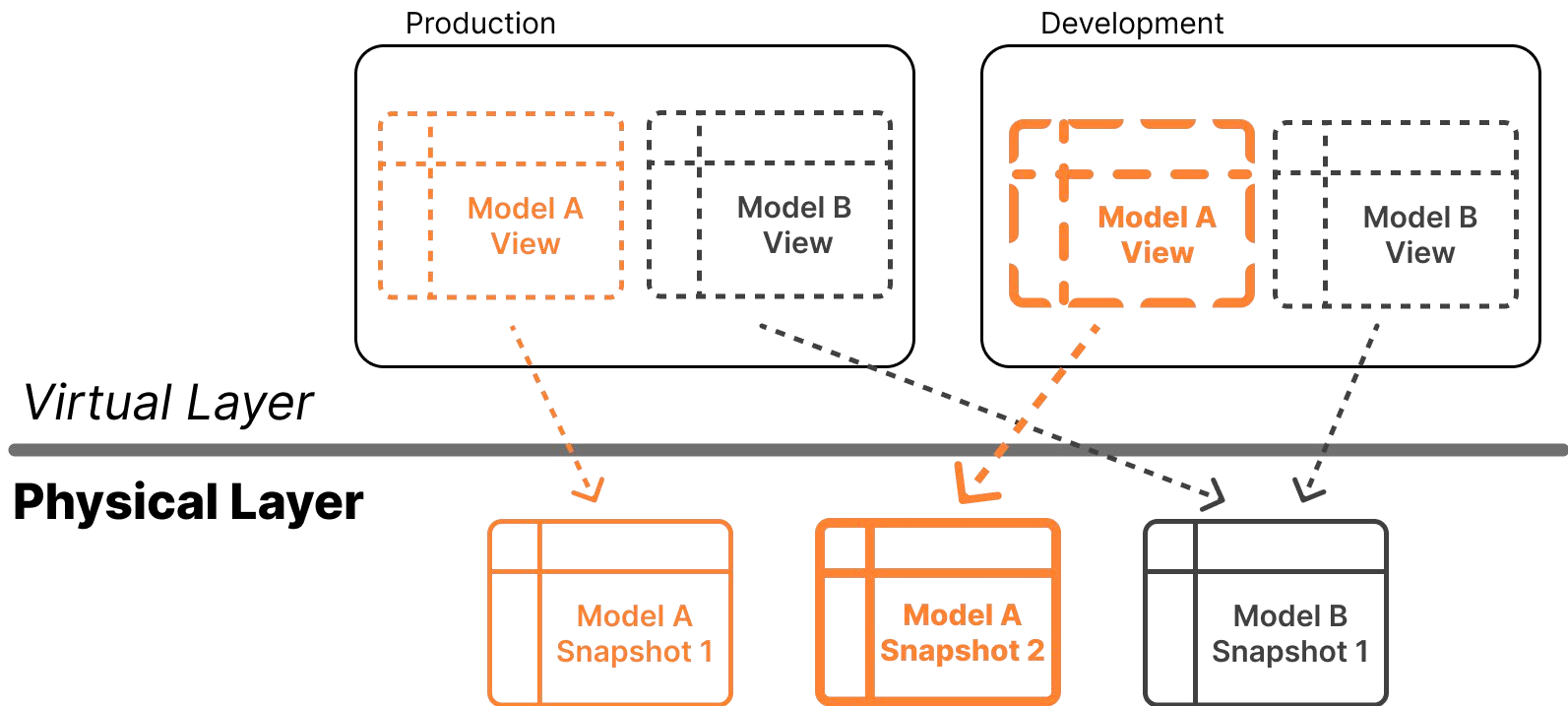
Virtual Data Environments: Flow

3. Make a change to Model A



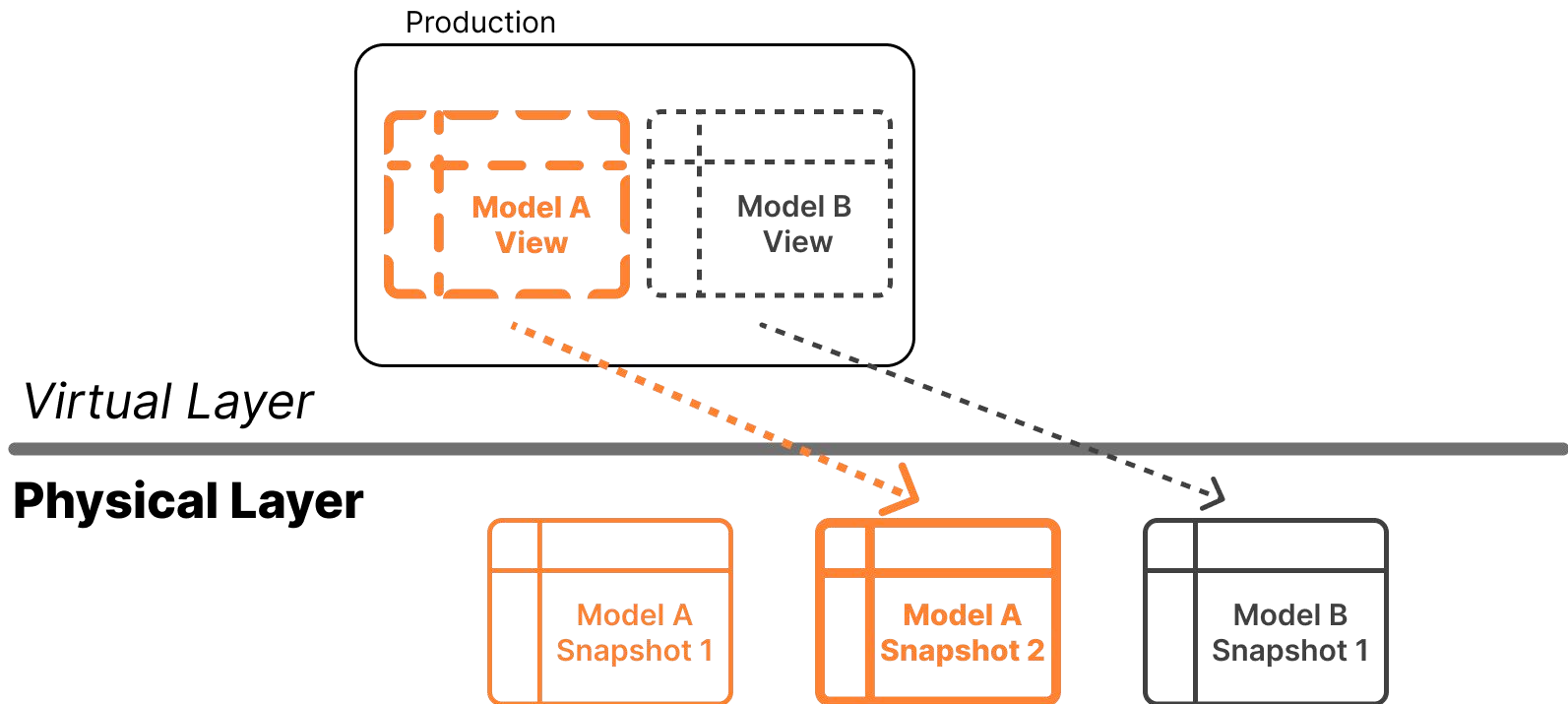
Virtual Data Environments: Flow

3. Make a change to Model A



Virtual Data Environments: Flow

4. Deploy model to production



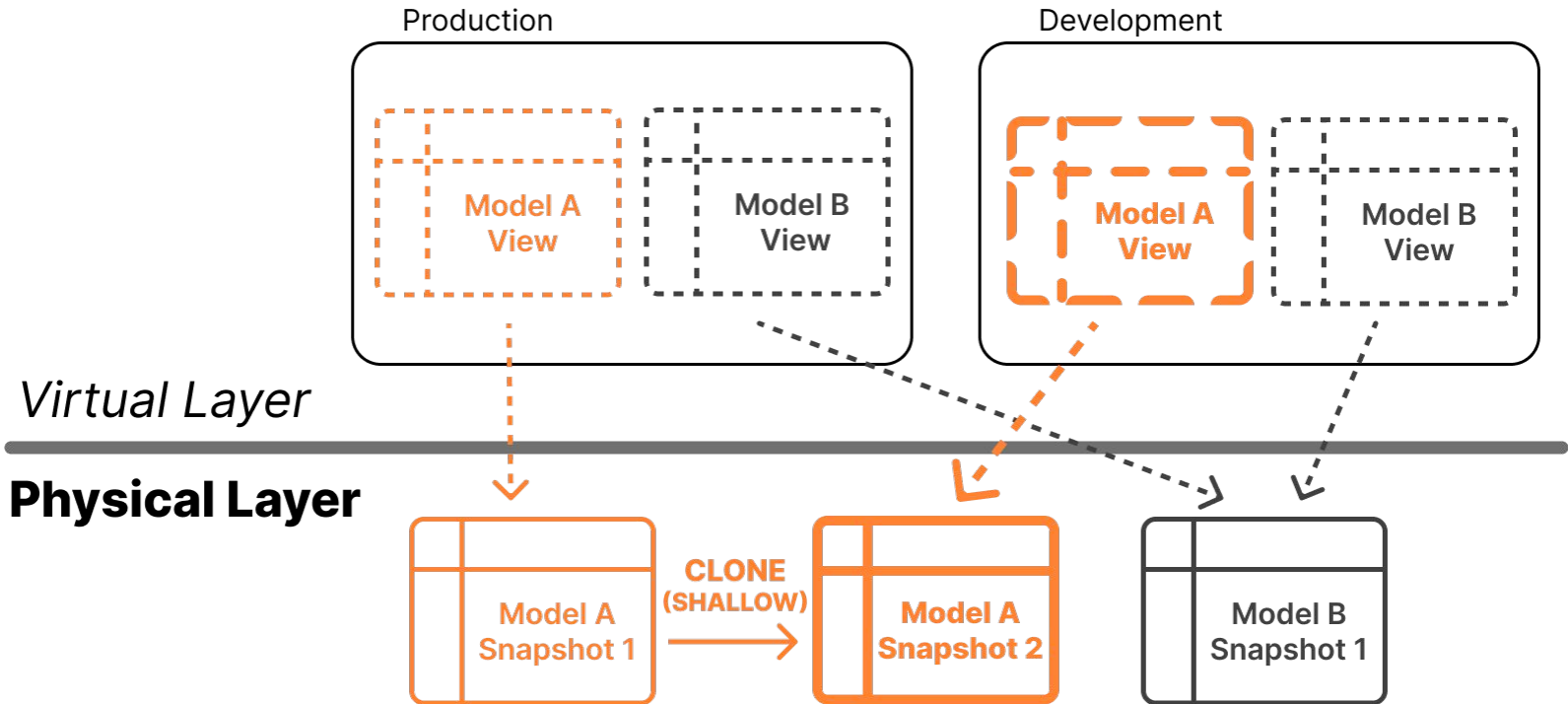
It's nice and all, but what if...

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

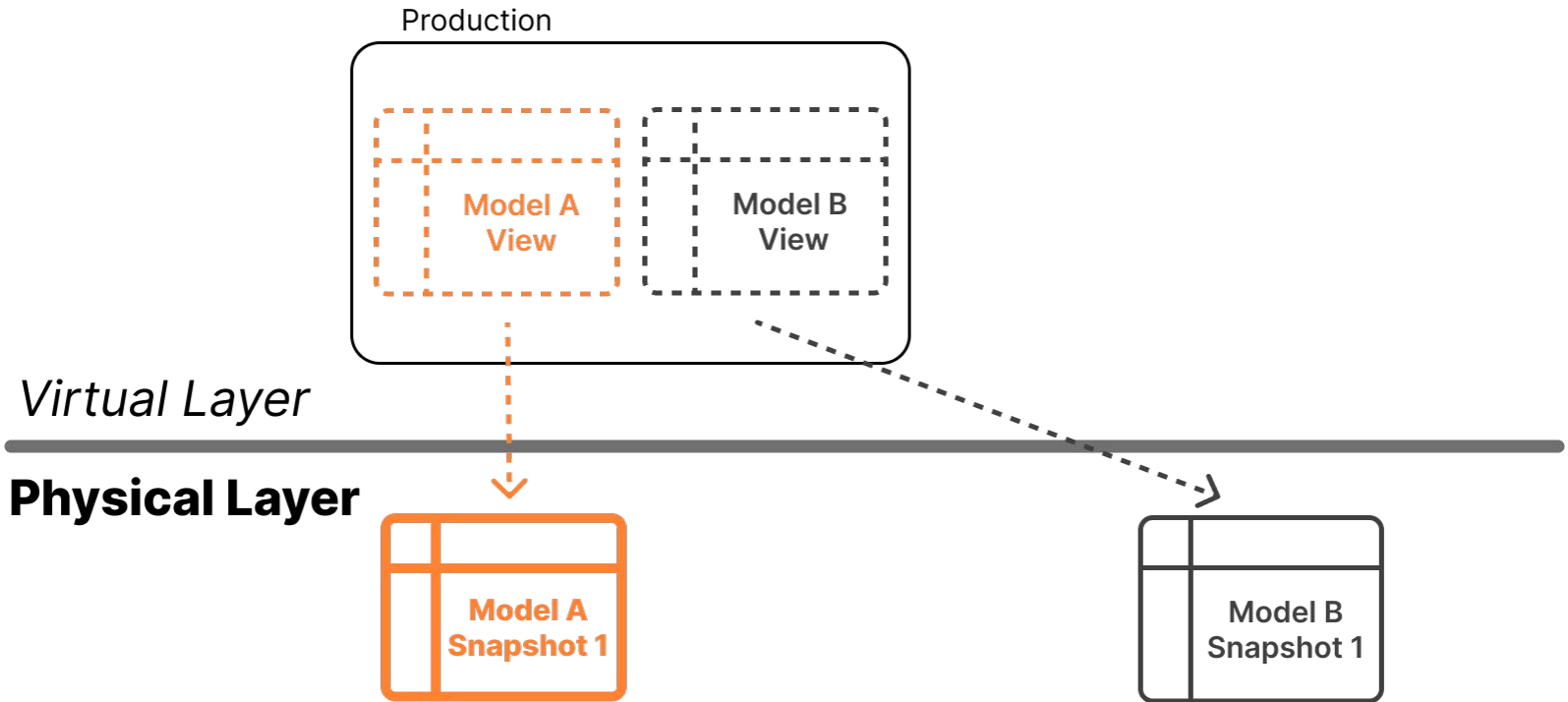
It's nice and all, but what if..



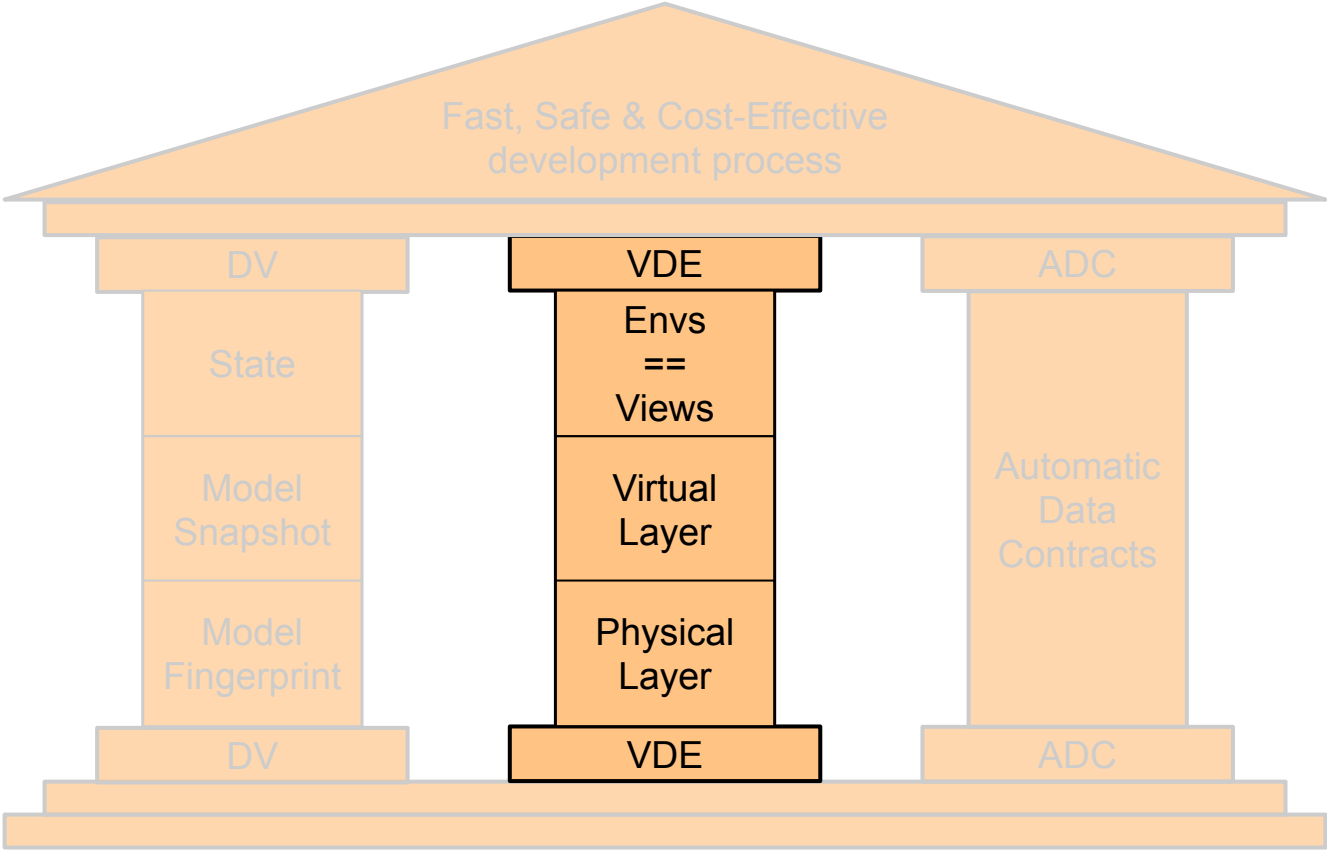
Forward-only Changes



Forward-only Changes



Pillar of Virtual Data Environments

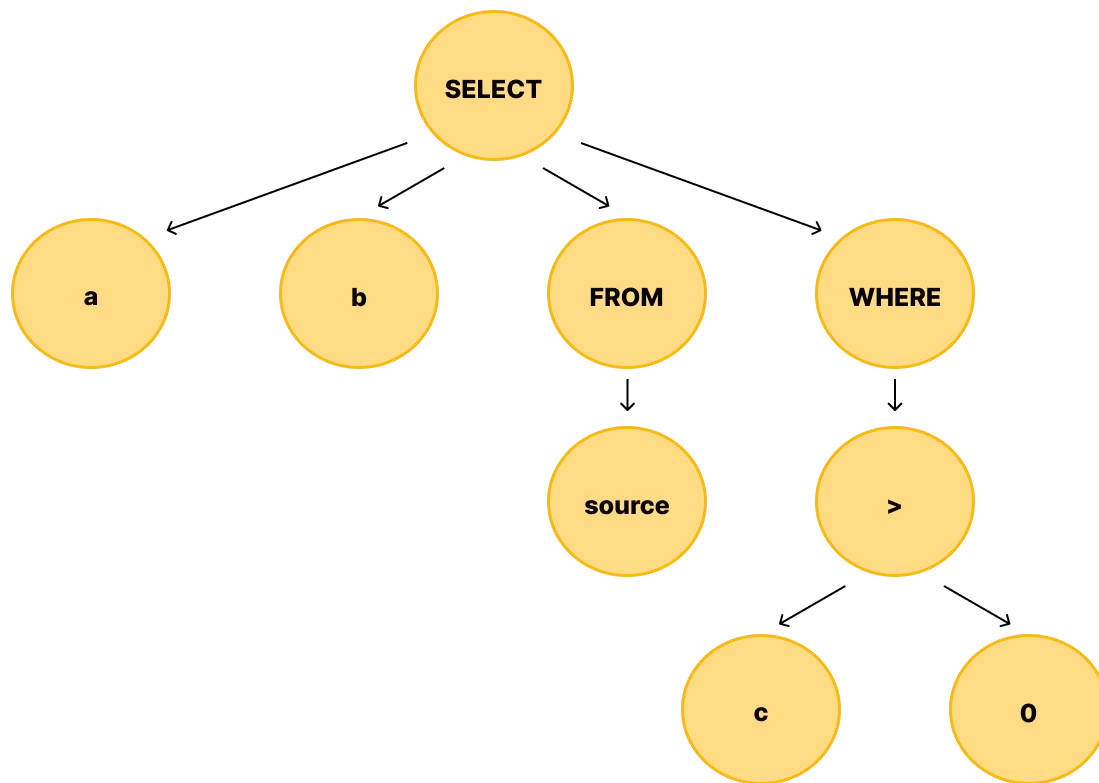


Semantic Understanding

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

Semantic Understanding

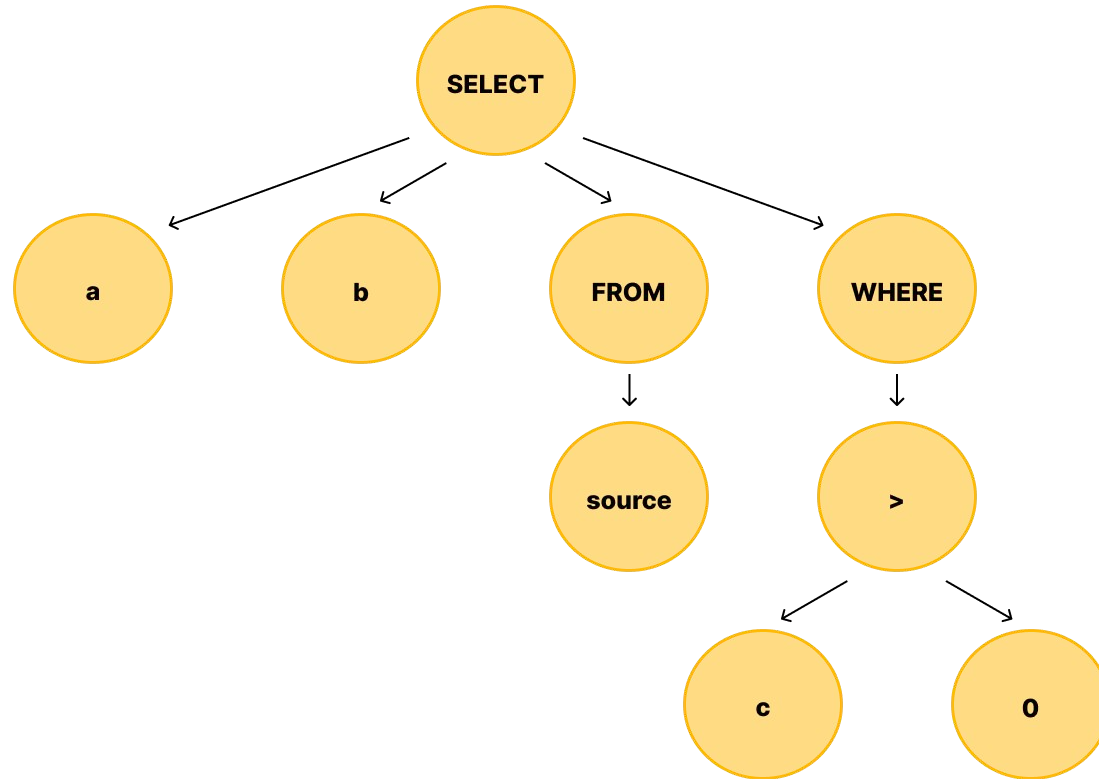
SELECT a, b FROM source WHERE c > 0;



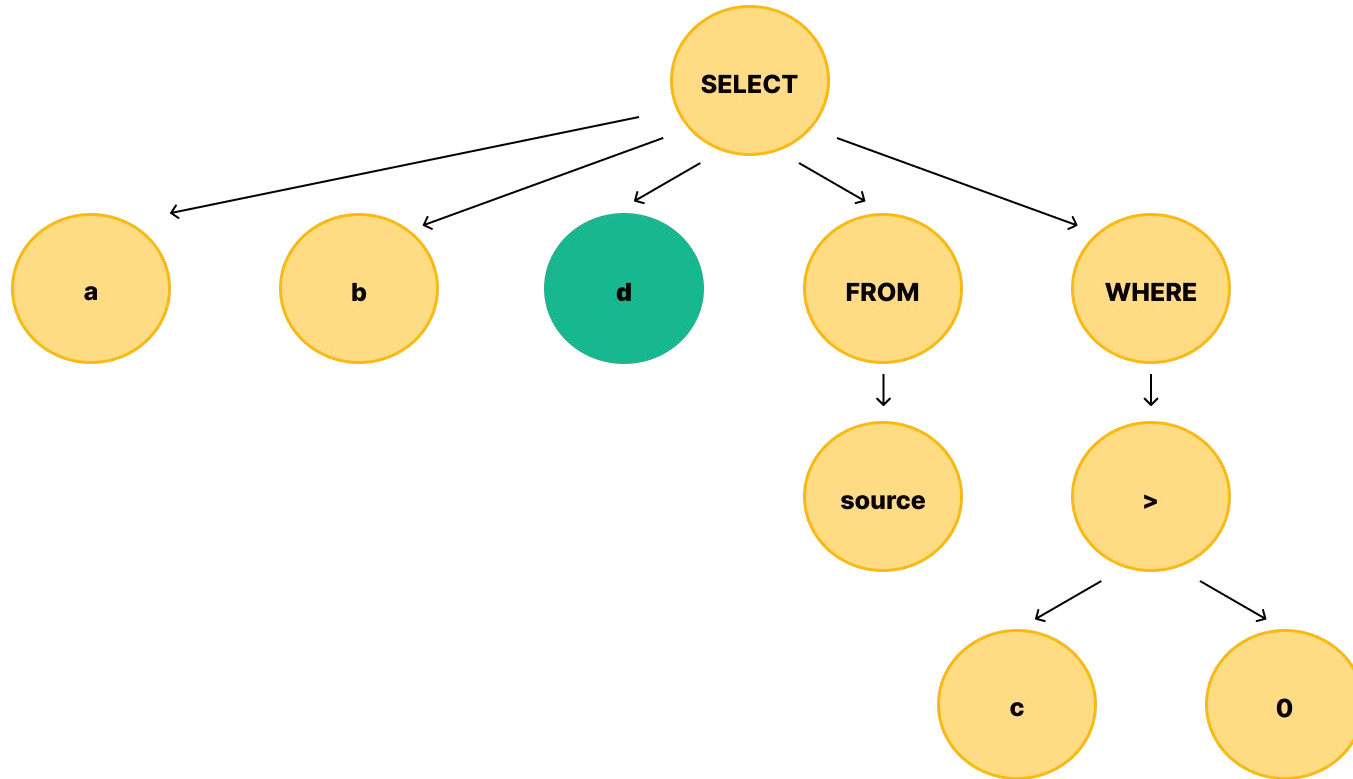
Semantic Understanding

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

Analyzing the Change

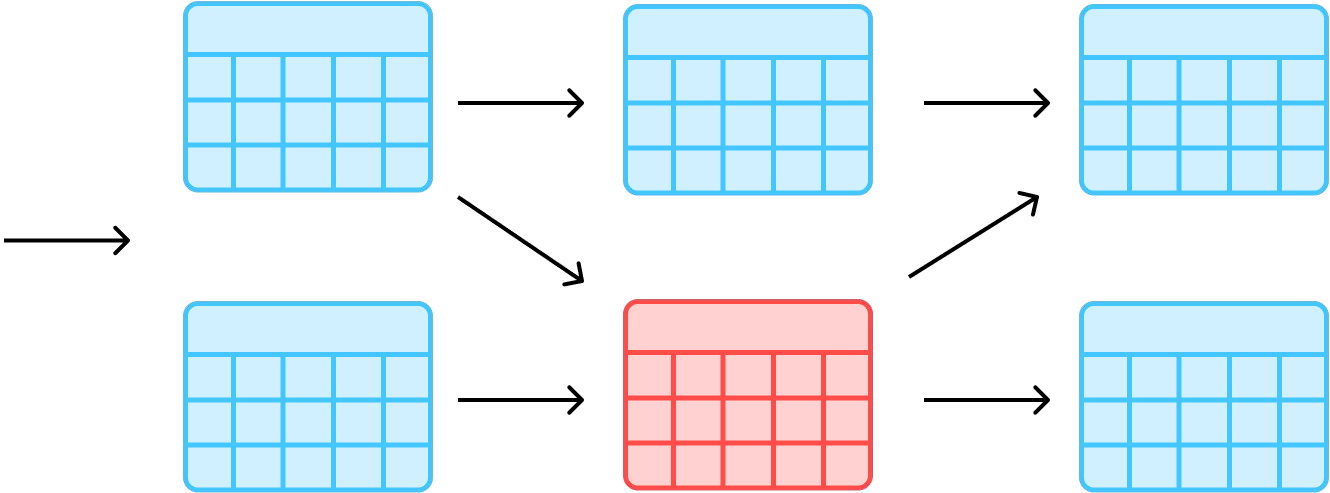


Analyzing the Change



Non-breaking Change

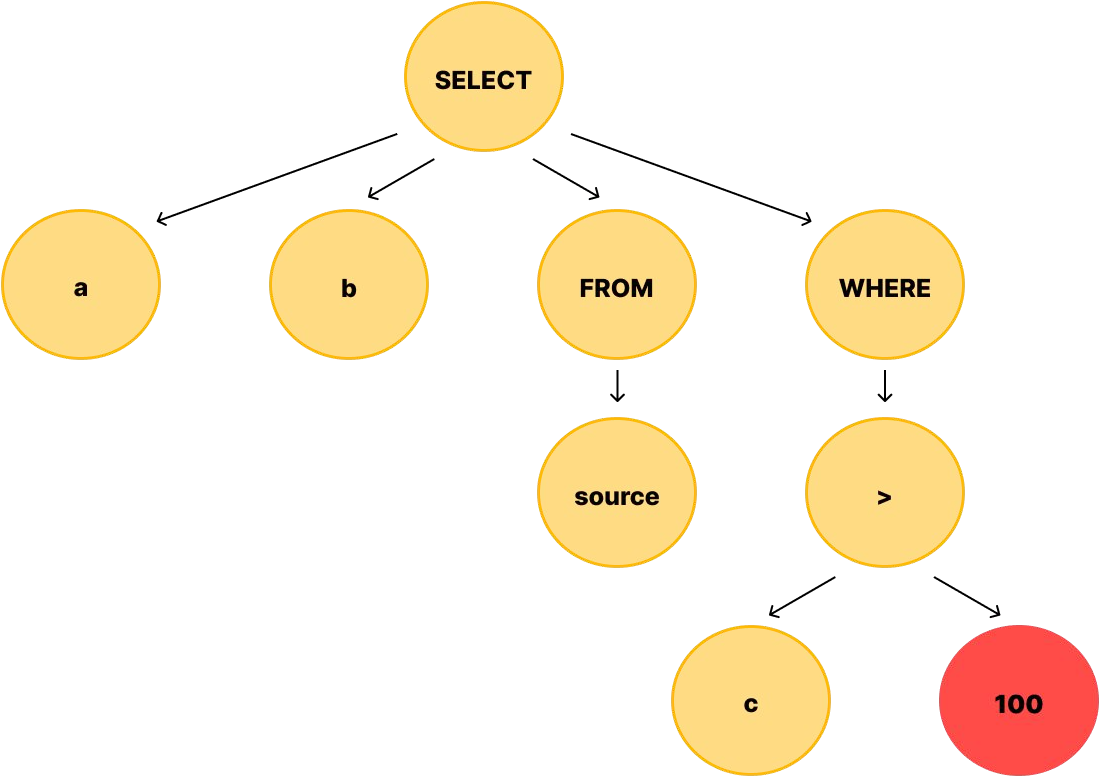
```
SELECT  
  a,  
  b,  
  d  
FROM source  
WHERE c > 0
```



Breaking Change

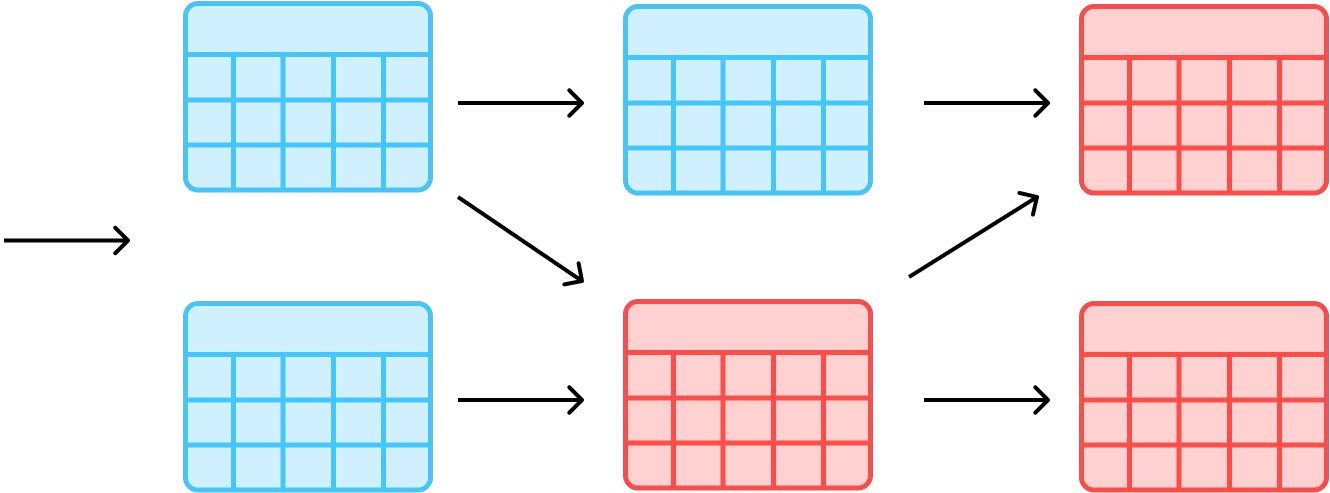
```
SELECT a, b FROM source WHERE c > 100
```


Breaking Change

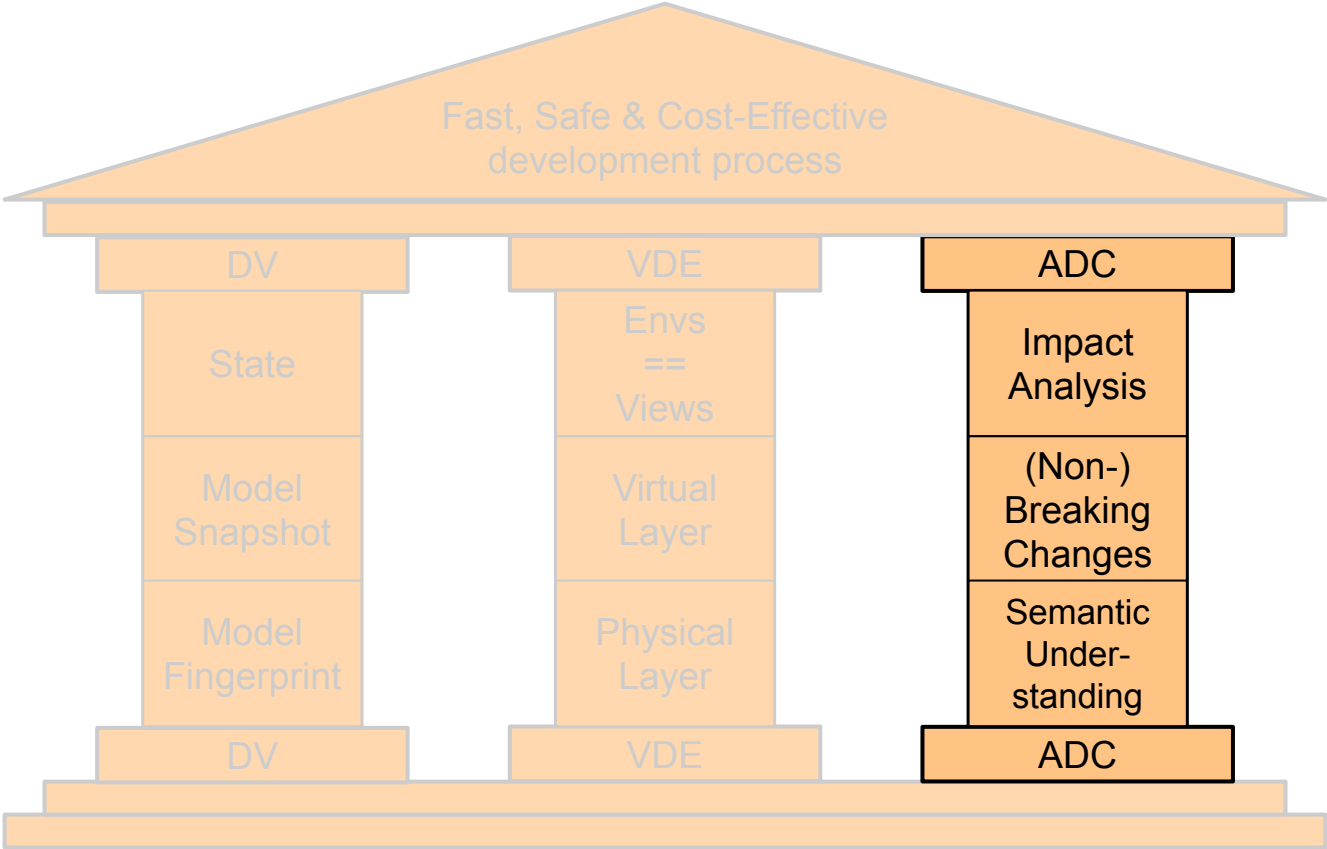


Breaking Change

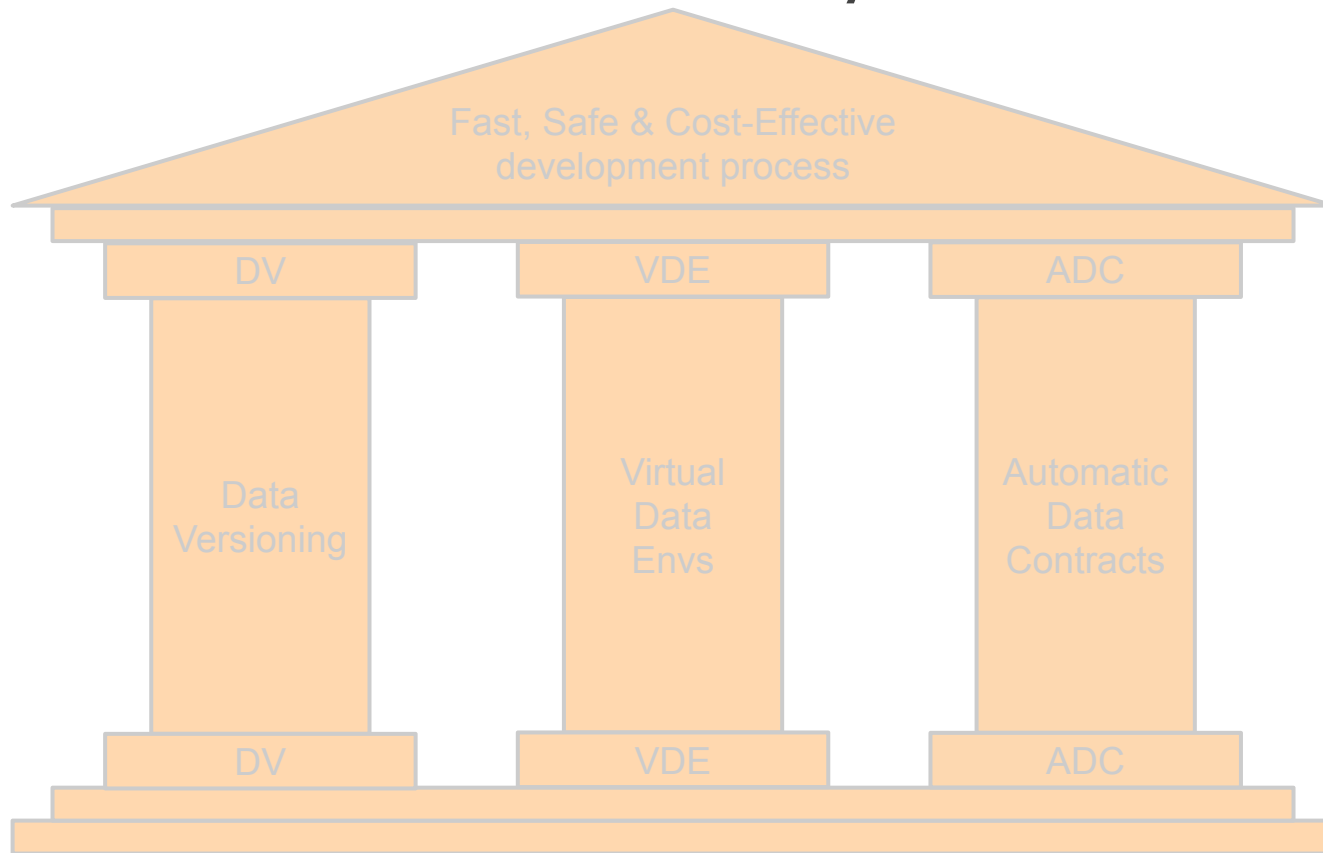
```
SELECT  
  a,  
  b,  
FROM source  
WHERE c > 100
```



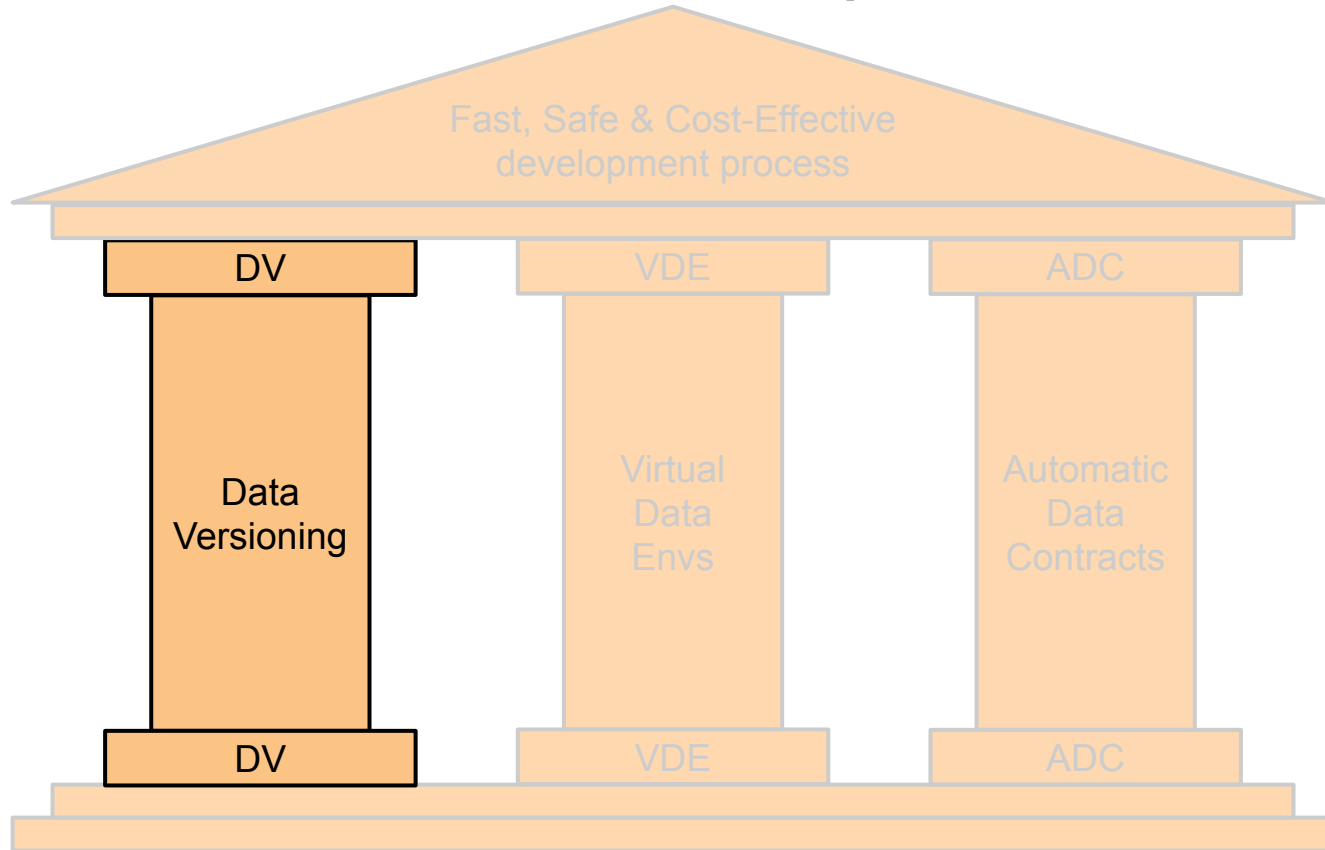
Pillar of Automatic Data Contracts



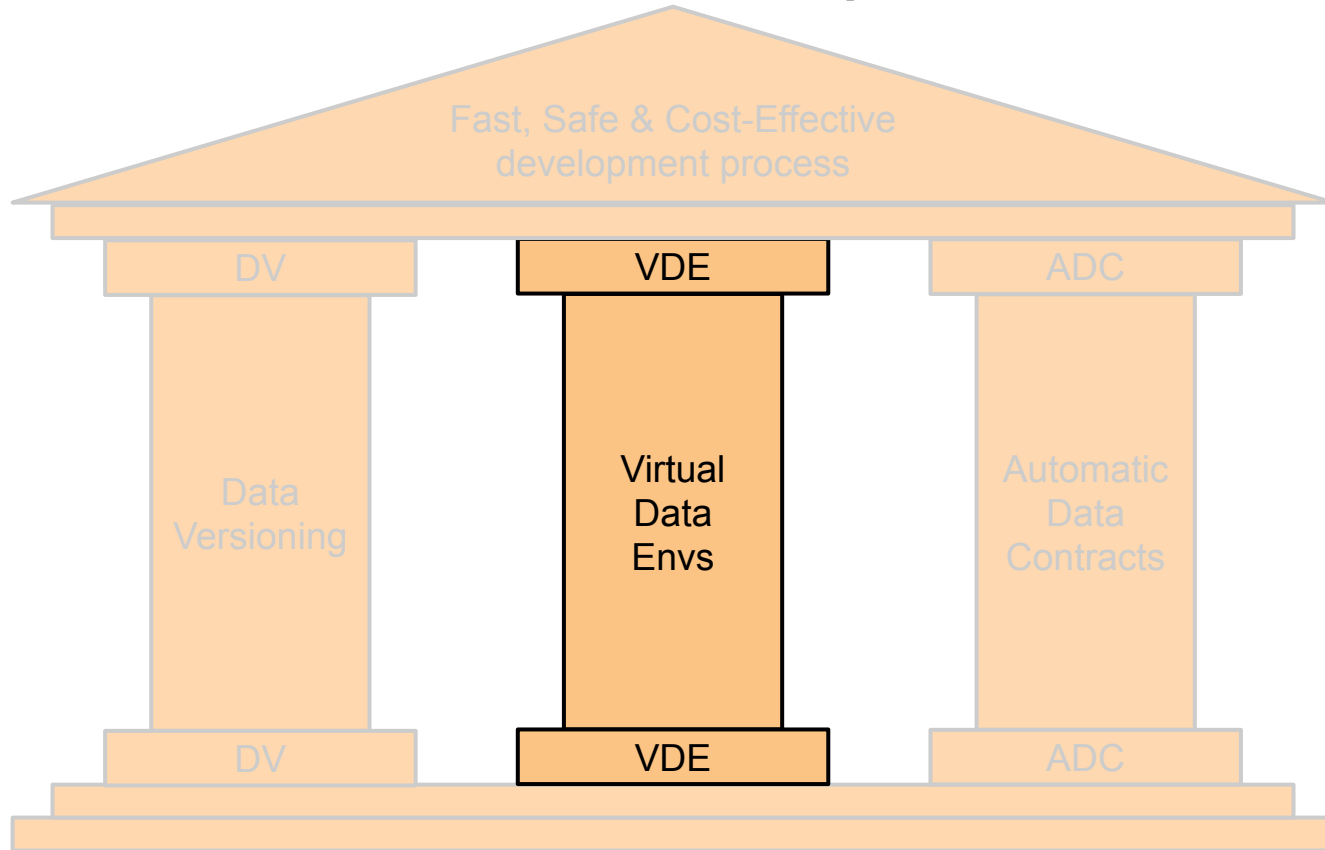
Three Pillars of Data Productivity



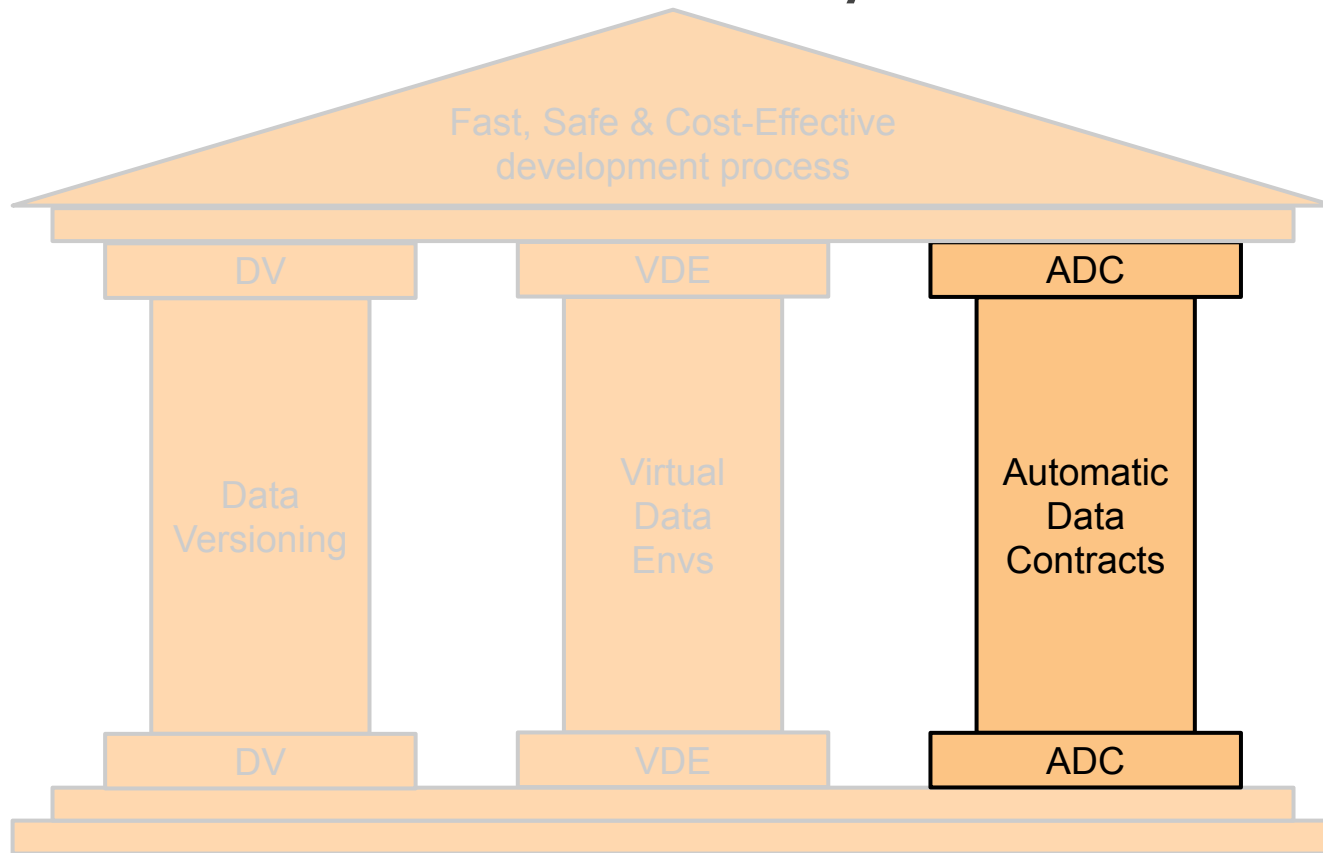
Three Pillars of Data Productivity



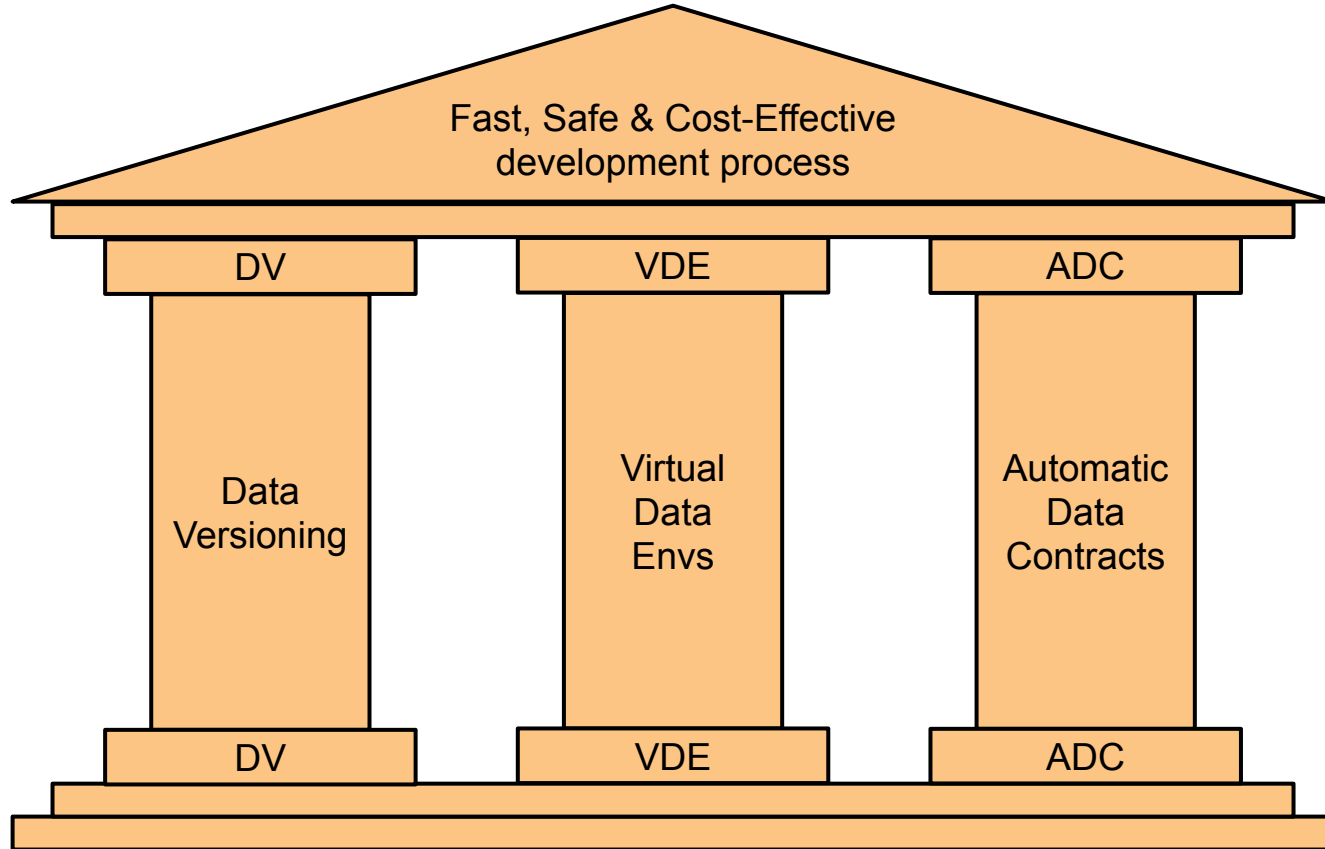
Three Pillars of Data Productivity



Three Pillars of Data Productivity



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

The screenshot displays the SQLMesh by tobiko interface. At the top, the logo "SQLMesh by tobiko" is on the left, and "Documentation" with a star icon is on the right. Below the logo, there are navigation arrows and a "Plan" button with a refresh icon. To the right of the "Plan" button are "prod" and "No Changes" buttons, and a "No Errors" button. The main area shows a SQL query in a dark theme editor:

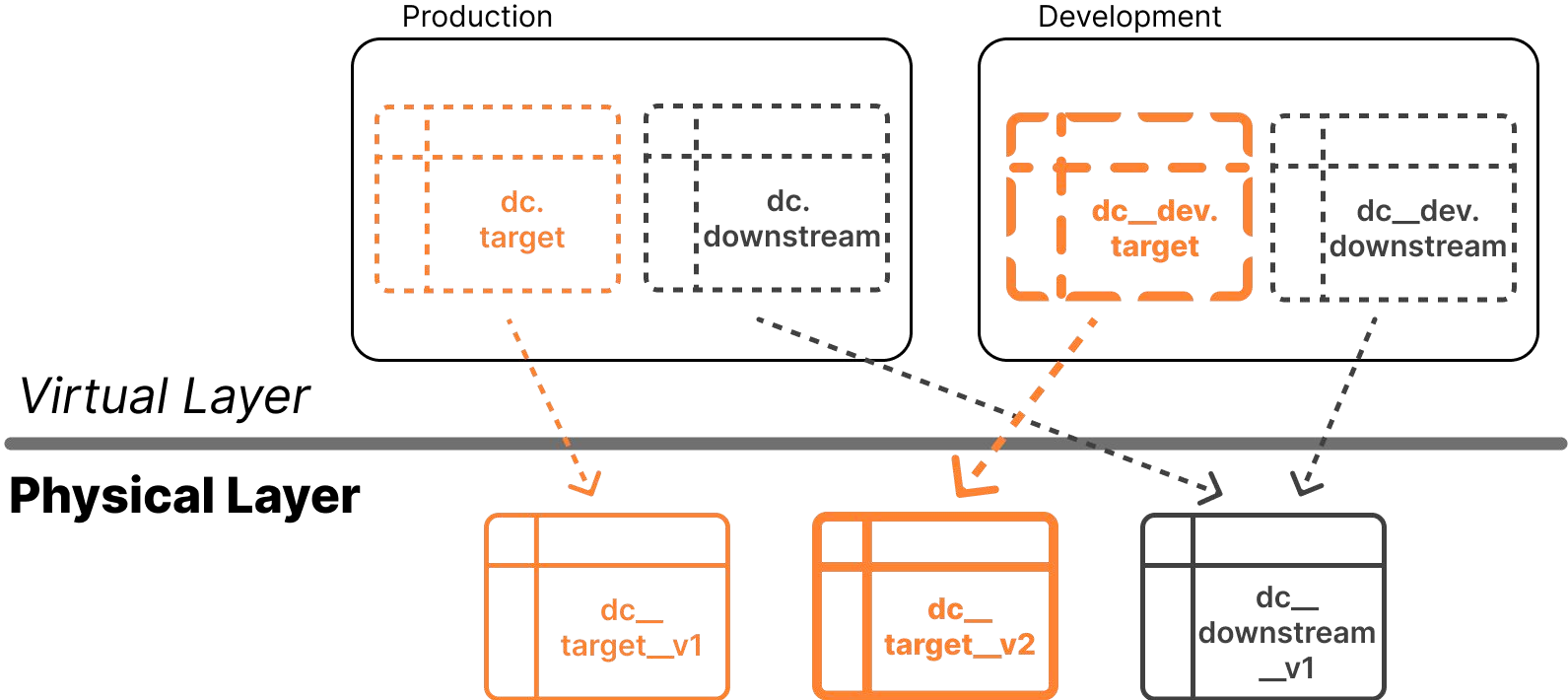
```
1 MODEL (  
2   name dc.target  
3 );  
4  
5 SELECT  
6   a,  
7   b  
8 FROM dc.source  
9 WHERE  
10  c > 0
```

Below the query, there are status indicators: "Saved: ● Formatted: ● Language: SQL Dialect: duckdb SQLMesh Type: model". A "Lineage" tab is active, showing a lineage diagram for "dc.target". The diagram has a search bar with "Find" and a "Show" dropdown. It displays three nodes: "dc.source" (4), "dc.target" (3), and "dc.downstream" (2). Each node shows its schema: "dc.source" has columns a, b, c; "dc.target" has columns a, b; and "dc.downstream" has columns a, b. Arrows indicate the flow of data from "dc.source" to "dc.target" and then to "dc.downstream". A vertical toolbar on the left of the diagram contains icons for zooming and locking.

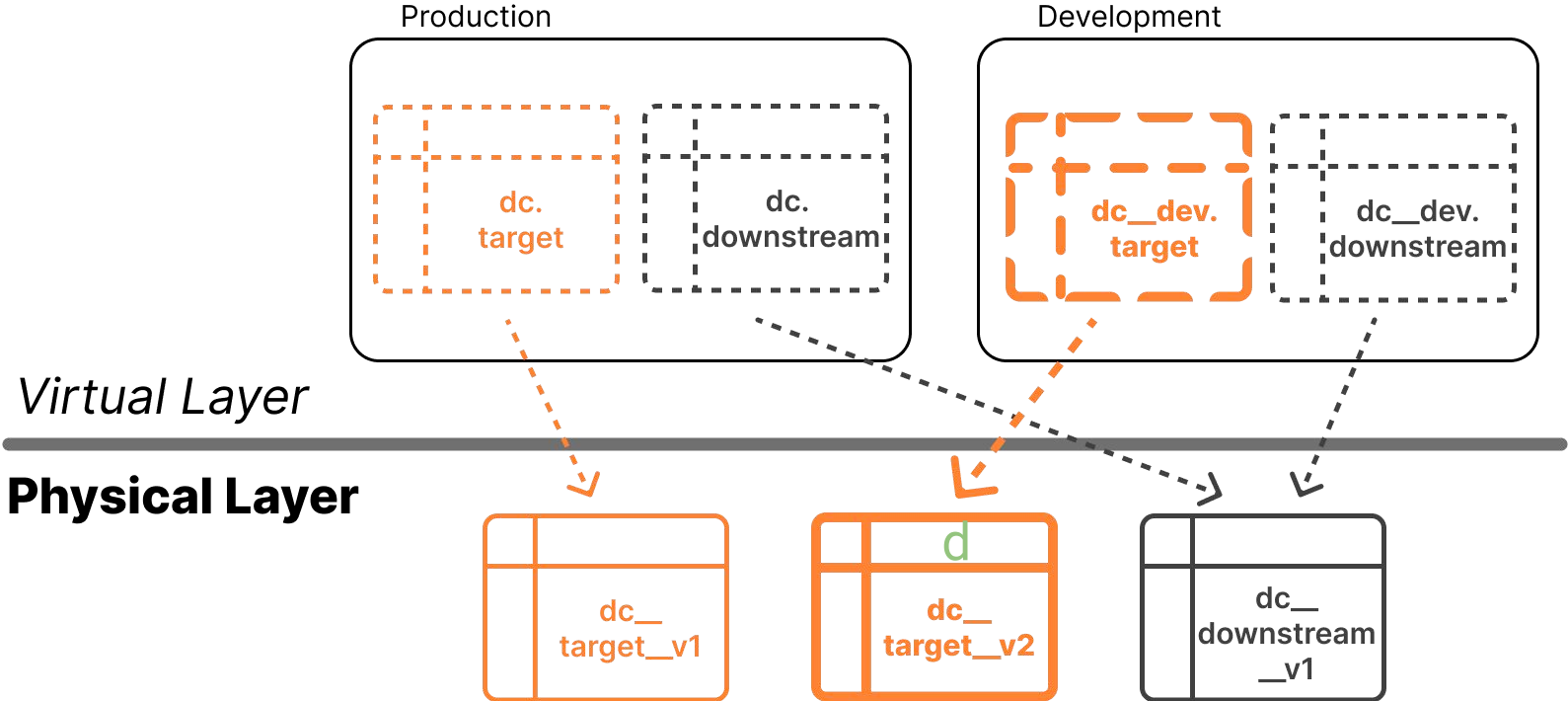
SQLMesh: 0.72.2

© 2024 Tobiko Data, Inc. All rights reserved.

Non-breaking Change



Non-breaking Change



Breaking Change

The screenshot displays the SQLMesh by tobiko interface. At the top, there is a navigation bar with a 'Plan' button, a dropdown menu set to 'dev', and status indicators for 'No Changes' and 'No Errors'. The main editor shows a SQL file named 'target.sql' with the following code:

```
1 MODEL (  
2   name dc.target  
3 );  
4  
5 SELECT  
6   a,  
7   b  
8 FROM dc.source  
9 WHERE  
10  c > d
```

Below the editor, a status bar indicates: 'Saved: ● Formatted: ● Language: SQL Dialect: duckdb SQLMesh Type: model'. A 'Lineage' button is visible, and the lineage diagram below it shows the data flow:

- dc.source** (4) with columns a, b, c, d
- dc.target** (2) with columns a, b
- dc.downstream** (2) with columns a, b

The lineage diagram shows a flow from 'dc.source' to 'dc.target', and then from 'dc.target' to 'dc.downstream'. A search bar at the top of the lineage view shows 'All: 3 Upstream/Downstream: 2'.

SQLMesh: 0.72.2 © 2024 Tobiko Data, Inc. All rights reserved.

Deploy to Prod

The screenshot displays the SQLMesh interface. At the top, the logo "SQLMesh by tobiko" is visible on the left, and "Documentation" with a star icon is on the right. A navigation bar includes a "Plan" button with a play icon, a "dev" dropdown menu, and two status indicators: "No Changes" and "No Errors".

The main editor area shows a file named "target.sql" with the following SQL code:

```
1 MODEL (  
2   name dc.target  
3 );  
4  
5 SELECT  
6   a,  
7   b  
8 FROM dc.source  
9 WHERE  
10  c > 100
```

Below the editor, a status bar indicates: "Saved: ● Formatted: ● Language: SQL Dialect: duckdb SQLMesh Type: model".

The "Lineage" section is active, showing a diagram for "dc.target". It indicates "All: 3" and "Upstream/Downstream: 2". A search bar labeled "Find" and a "Show" dropdown are present. The lineage diagram consists of three nodes on a grid:

- A node labeled "dc.source" with a count of 4. It contains a table with columns a, b, c, and d, each with a data type of INT.
- A node labeled "dc.target" with a count of 2, highlighted with an orange border. It contains a table with columns a, b, and c, each with a data type of INT.
- A node labeled "dc.downstream" with a count of 2. It contains a table with columns a and b, each with a data type of INT.

On the left side of the lineage diagram, there is a vertical toolbar with icons for zooming in (+), zooming out (-), refreshing, and locking.

At the bottom left, the version "SQLMesh: 0.72.2" is displayed. At the bottom right, the copyright notice "© 2024 Tobiko Data, Inc. All rights reserved." is shown.

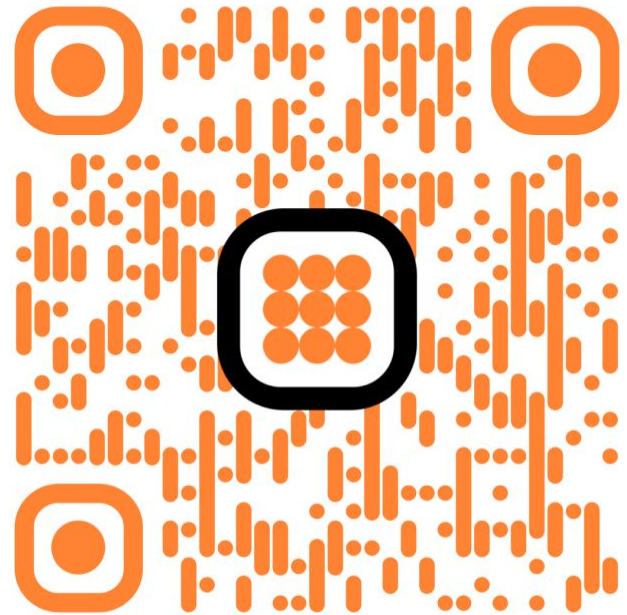


Join us on Slack!

<https://tobikodata.com/slack>

Thank you!

Scan me!



Backup Slides

Non-breaking Change

non_breaking_change.mv by tobiko Documentation

Plan prod No Changes No Errors

```
target.sql
1 MODEL (
2   name dc.target
3 );
4
5 SELECT
6   a,
7   b
8 FROM dc.source
9 WHERE
10  c > 0
```

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

Lineage

dc.target All: 3 Upstream/Downstream: 2 Find Show

The lineage diagram illustrates the data flow between three tables: `dc.source`, `dc.target`, and `dc.downstream`. `dc.source` (4 columns: a, b, c, d) is the upstream table. `dc.target` (2 columns: a, b) is the current table, highlighted in orange, and is the downstream of `dc.source`. `dc.downstream` (2 columns: a, b) is the downstream of `dc.target`. The diagram shows a grid of data points with arrows indicating the flow from left to right.

SQLMesh: 0.72.2 © 2024 Tobiko Data, Inc. All rights reserved.

Non-breaking Change

The screenshot displays the SQLMesh web interface. At the top, the file name is 'non_breaking_change.mv' and the user is 'tobiko'. The environment is set to 'prod' (Production Environment). The status bar shows 'No Changes' and 'No Errors'. A red arrow points to the 'dev' environment selection box in the environment dropdown menu.

```
1 MODEL (  
2   name dc.target  
3 );  
4  
5 SELECT  
6   a,  
7   b  
8 FROM dc.source  
9 WHERE  
10  c > 0
```

Below the code editor, the status bar indicates: Saved: ● Formatted: ● Language: SQL Dialect: duckdb SQLMesh Type: model.

The Lineage view shows a dependency graph for 'dc.target'. It includes three nodes: 'dc.source' (4), 'dc.target' (2), and 'dc.downstream' (2). The 'dc.target' node is highlighted in orange, indicating it is the current model being viewed. The lineage shows 'dc.source' as an upstream dependency and 'dc.downstream' as a downstream dependency.

© 2024 Tobiko Data, Inc. All rights reserved.

Non-breaking Change

non_breaking_change.mgv by tobiko

Documentation

Plan dev Changes 1 1 1 No Errors

```
1 MODEL (  
2   name dc.target  
3 );  
4  
5 SELECT  
6   a,  
7   b,  
8   d  
9 FROM dc.source  
10 WHERE  
11   c > 0
```

Directly Modified
dc_dev.target

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

Lineage

dc.target All: 3 Upstream/Downstream: 2 Find Show

The lineage diagram shows three data nodes on a grid background. On the left is 'dc.source' (ID 4) with columns a, b, c, and d. In the center is 'dc.target' (ID 3) with columns a, b, and d. On the right is 'dc.downstream' (ID 2) with columns a and b. Arrows indicate data flow: from 'dc.source' to 'dc.target', and from 'dc.target' to 'dc.downstream'. The 'dc.target' node is highlighted with an orange border.

SQLMesh: 0.72.2 © 2024 Tobiko Data, Inc. All rights reserved.

Non-breaking Change

The screenshot displays the SQLMesh web interface. At the top, the file name is `non_breaking_change.mv` by `tobiko`. The main editor shows a SQL query in `target.sql`:

```
1 MODEL (  
2   name dc.target  
3 );  
4  
5 SELECT  
6   a,  
7   b,  
8   d  
9 FROM dc.source  
10 WHERE  
11   c > 0
```

A notification box on the right indicates "Indirectly Modified" for `dc_dev.downstream`. Below the editor, the status bar shows "Saved: ● Formatted: ● Language: SQL Dialect: duckdb SQLMesh Type: model".

The "Lineage" section is active, showing a dependency graph for `dc.target`. It contains three nodes:

- `dc.source` (ID: 4) with columns `a`, `b`, `c`, and `d`, all of type `INT`.
- `dc.target` (ID: 3) with columns `a`, `b`, and `d`, all of type `INT`.
- `dc.downstream` (ID: 2) with columns `a` and `b`, both of type `INT`.

Arrows indicate dependencies: `dc.target` depends on `dc.source`, and `dc.downstream` depends on `dc.target`. The `dc.target` node is highlighted in orange, indicating it is the current focus.

At the bottom left, the version is `SQLMesh: 0.72.2`. At the bottom right, the copyright notice is `© 2024 Tobiko Data, Inc. All rights reserved.`

Non-breaking Change

non_breaking_change.mgw by tobiko

Documentation

Plan dev Changes 1 1 No Errors

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

The screenshot shows the Tobiko Data Mesh interface for configuring a change. The top bar includes a 'Plan' button, a dropdown menu set to 'dev', and a 'Changes' summary showing 1 change and 1 error, with 'No Errors' displayed. The main content area is titled 'dc_dev.target' and shows a configuration for 'dc_dev.downstream'. The 'Non-Breaking Change' option is selected, with a description: 'It will exclude all indirect models caused by this change'. Other options include 'Breaking Change', 'Forward-Only Change', 'Indirect Breaking Change', and 'Indirect Non-Breaking Change'. Below the options is a code editor showing a SQL query with a change in the 'SELECT' clause: 'a,' is added and 'b' is removed. The bottom of the interface features 'Apply Changes And Backfill' and 'Start Over' buttons. The footer shows '© 2024 Tobiko Data, Inc. All rights reserved.'

non_breaking_change.mgw by tobiko

Documentation

Plan dev Changes 1 1 No Errors

dc_dev.target

dc_dev.downstream

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
```

Apply Changes And Backfill Start Over

SQLMesh: 0.72.2 © 2024 Tobiko Data, Inc. All rights reserved.

Non-breaking Change

non_breaking_change.m9w by tobiko

Documentation

Plan dev Changes 11 1 No Errors

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 1

dc_dev.target

Tests Skipped

Apply Changes And Backfill Start Over

Non-breaking Change

non_breaking_change.mgw by tobiko

Documentation

Plan dev No Changes No Errors

- 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

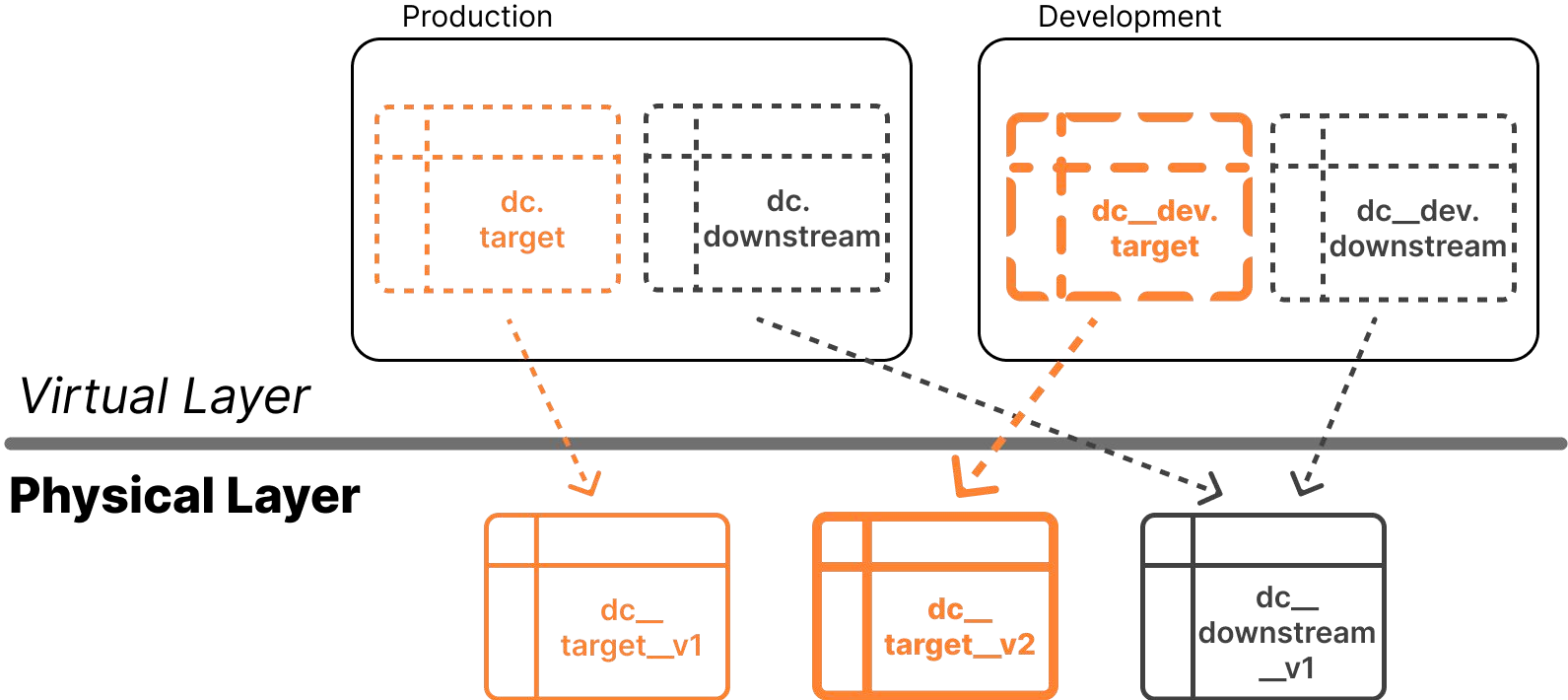
- Environment Promoted

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

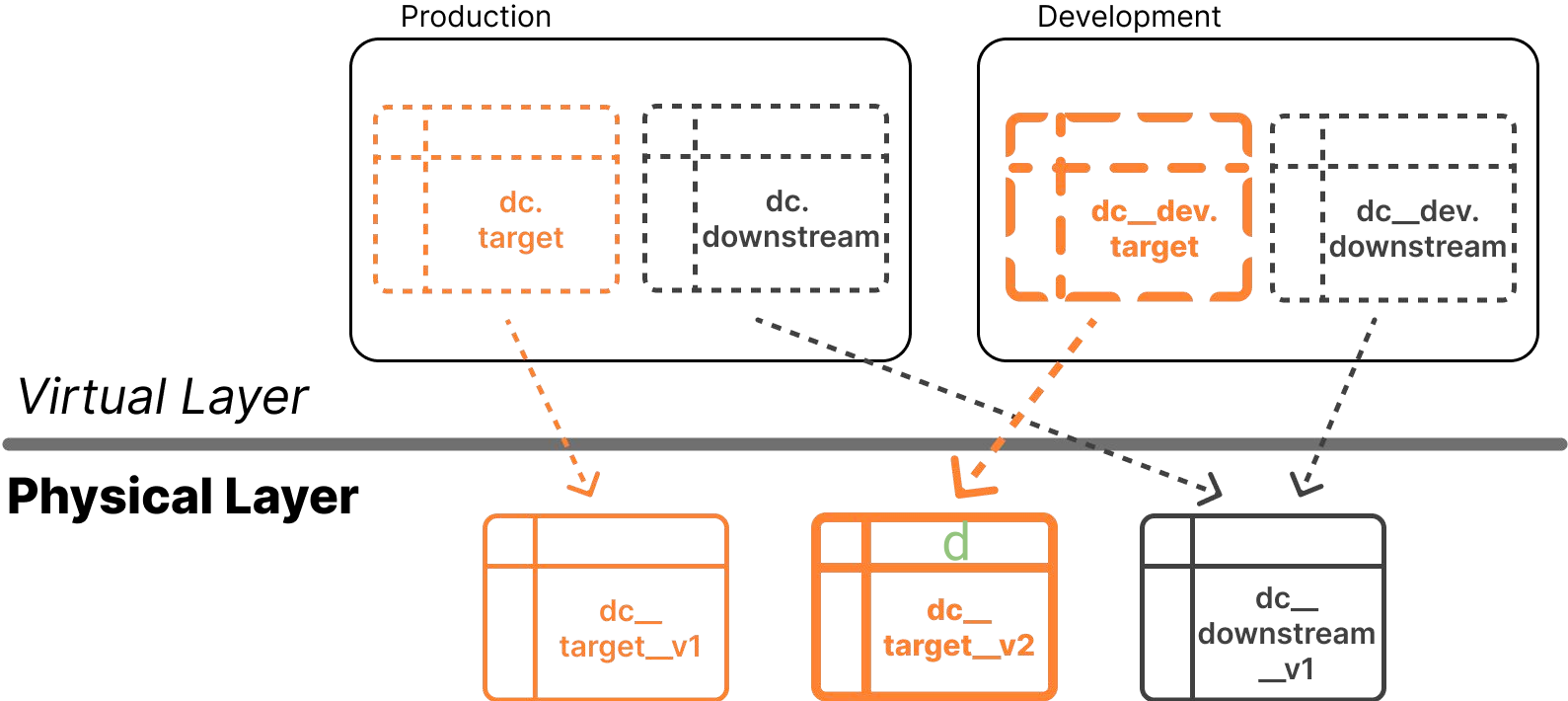
Go Back

© 2024 Tobiko Data, Inc. All rights reserved.

Non-breaking Change



Non-breaking Change



Breaking Change

The screenshot displays the SQLMesh interface by Tobiko. At the top, the logo 'SQLMesh by tobiko' is visible on the left, and 'Documentation' with a sun icon is on the right. Below the logo, there are navigation icons and a file explorer showing 'target.sql'. The main code editor contains the following SQL query:

```
1 MODEL (  
2   name dc.target  
3 );  
4  
5 SELECT  
6   a,  
7   b  
8 FROM dc.source  
9 WHERE  
10  c > 100
```

A red arrow points to the line 'c > 100'. A notification box on the right says 'Directly Modified' with a refresh icon and 'dc_dev.target'. Below the code editor, there is a status bar: 'Saved: ● Formatted: ● Language: SQL Dialect: duckdb SQLMesh Type: model'. A 'Lineage' tab is active, showing a diagram with three nodes: 'dc.source' (4 nodes), 'dc.target' (2 nodes, highlighted in orange), and 'dc.downstream' (2 nodes). The diagram shows data flow from 'dc.source' to 'dc.target' and then to 'dc.downstream'. The bottom of the screen shows 'SQLMesh: 0.72.2' on the left and '© 2024 Tobiko Data, Inc. All rights reserved.' on the right.

Breaking Change

The screenshot shows the SQLMesh interface with the following elements:

- Header:** SQLMesh by tobiko logo, Documentation link, and a status bar with 'Plan' (green), 'dev' (dropdown), 'Changes 1 1 2' (orange), and 'No Errors' (grey).
- Interval:** Two date input fields: '2023-12-13' (The start datetime of the interval) and '2022-12-13' (The end datetime of the interval).
- Additional Options:** A section with a '+' icon.
- Changes:** A section with a green minus icon and a red arrow pointing to a 'Breaking Change' notification for the 'dc_dev.target' model.
- Modified Indirectly:** A section with an orange plus icon for 'dc_dev.downstream'.
- Backfills:** A section with a green plus icon for 'Models 2', listing 'dc_dev.downstream' and 'dc_dev.target'.
- Buttons:** 'Apply Changes And Backfill' (blue) and 'Start Over' (grey).

Breaking Change

SQLMesh by tobiko

Documentation

Plan dev Changes 1 2 No Errors

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

Apply Changes And Backfill Start Over

Breaking Change

SQLMesh by tobiko Documentation

Plan dev No Changes No Errors

- 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

[Go Back](#)

Deploy to Prod

The screenshot shows the SQLMesh interface with a model named 'target.sql' open. The model's SQL code is as follows:

```
1 MODEL (  
2   name dc.target  
3 );  
4  
5 SELECT  
6   a,  
7   b  
8 FROM dc.source  
9 WHERE  
10  c > 100
```

A dropdown menu is open, showing deployment options for the 'dev' environment. The 'prod' environment is selected, with a red arrow pointing to it. The 'prod' environment is labeled '(remote)' and 'Production Environment'. Below the dropdown, there is an 'Environment' input field and an 'Add' button.

Below the code editor, the status bar shows: Saved: ● Formatted: ● Language: SQL Dialect: duckdb SQLMesh Type: model

The 'Lineage' section is visible, showing a dependency graph for the model 'dc.target'. The graph shows three nodes: 'dc.source' (4), 'dc.target' (2), and 'dc.downstream' (2). The 'dc.target' node is highlighted in orange, indicating it is the current model. The 'dc.source' node has columns 'a', 'b', and 'd' with data types 'INT', 'INT', and 'INT' respectively. The 'dc.target' node has columns 'a' and 'b' with data types 'INT' and 'INT' respectively. The 'dc.downstream' node has columns 'a' and 'b' with data types 'INT' and 'INT' respectively.

At the bottom of the interface, the version 'SQLMesh: 0.72.2' is displayed on the left, and the copyright notice '© 2024 Tobiko Data, Inc. All rights reserved.' is on the right.

Deploy to Prod

The screenshot displays the SQuidMesh interface for a deployment to the 'prod' environment. The interface includes a top navigation bar with 'Plan', 'prod', 'Changes 1', and 'No Errors' buttons. A sidebar on the left contains icons for a folder, a document with a '3' notification, a warning triangle, and a play button. The main content area is titled 'Prod Environment' and features an 'Additional Options' section with a plus icon. Below this, a 'Changes' section is expanded, showing 'Modified Directly' with 'dc.target' and a 'Breaking Change' button, and 'Modified Indirectly' with 'dc.downstream' and a plus icon. At the bottom, there are checkboxes for 'No Backfills' and 'Tests Skipped', a 'Virtual Update' section, and two large buttons: 'Apply Virtual Update' and 'Start Over'. The footer shows 'SQuidMesh: 0.72.2' on the left and '© 2024 Tobiko Data, Inc. All rights reserved.' on the right.

deploy_to_Prod.mv Mesh by tobiko Documentation

Plan prod Changes 1 No Errors

Prod Environment

Additional Options +

Changes

Modified Directly

dc.target Breaking Change +

Modified Indirectly

dc.downstream +

✓ No Backfills

✓ Tests Skipped

Virtual Update

Apply Virtual Update Start Over

SQuidMesh: 0.72.2 © 2024 Tobiko Data, Inc. All rights reserved.

Deploy to Prod

deploy_to_Prod.mv Mesh by tobiko Documentation

Plan prod Changes 1 No Errors

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

Apply Virtual Update Start Over

SQLMesh: 0.72.2 © 2024 Tobiko Data, Inc. All rights reserved.

Deploy to Prod

The screenshot shows a deployment tool interface with a dark theme. At the top, it says "deploy_to_prod.mov" and "Created by @tobiko". On the right, there are buttons for "Run", "prod" (with a dropdown arrow), "Changes" (with a count of 1), and "No Errors". The main area is titled "Prod Environment" and contains several sections: "Start Date (UTC)", "Additional Options", "Changes" (with a plus icon), "No Backfills" (checked), "Tests Skipped" (checked), and "Virtual Update" (with a plus icon). A modal dialog is open in the center with the title "Applying Plan Directly On Prod Environment!". The dialog text asks, "Are you sure you want to apply your changes directly on prod? Safer choice will be to select or add new environment first." Below the text is a section titled "Select or Add Environment" containing a dropdown menu with "prod" selected, a text input field with "Environment", and an "Add" button. At the bottom of the dialog are two buttons: "Yes, Run prod" (red) and "No, Cancel" (white with a black border). In the background, there is a large "Apply Virtual Update" button and a "Start Over" button. At the bottom of the screen, there is a video player control bar showing "0:22 / 0:33" and "0.72.2", and a footer with "© 2024 Toobit Data, Inc. All rights reserved." and some icons.

deploy_to_prod.mov Created by @tobiko

Documentation

Run prod Changes 1 No Errors

Prod Environment

Start Date (UTC)

The start date/time of the instance

Additional Options

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

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

prod Environment Add

Yes, Run prod No, Cancel

Apply Virtual Update Start Over

0:22 / 0:33 0.72.2

© 2024 Toobit Data, Inc. All rights reserved.

Deploy to Prod

The screenshot displays the SSO Mesh interface for a deployment to the Prod Environment. The interface is dark-themed and includes a top navigation bar with the text "deploy_to_Prod.mev" and "SSO Mesh by tobiko". On the right side of the top bar, there is a "Documentation" link with a star icon. Below the top bar, there is a breadcrumb navigation area with a folder icon and navigation arrows. To the right of the breadcrumb are three buttons: "Plan" (green with a play icon), "prod" (with a dropdown arrow), "No Changes", and "No Errors".

The main content area is titled "Prod Environment" and contains a list of deployment status items:

- Changes** (green play icon)
- ✓ No Backfills
- ✓ No Tests
- Virtual Update Completed** (green play icon)

Below these items, it states "Evaluation started at 2024-02-26 21:18:04".

- Snapshot Tables Created** (green play icon)
- ✓ No Models To Restate
- Environment Promoted** (green play icon)

At the bottom of the main content area, it states "Evaluation stopped at 2024-02-26 21:18:05".

A "Go Back" button is located at the bottom right of the main content area.

The bottom status bar shows "SSO Mesh: 0.72.2" on the left and "© 2024 Tobiko Data, Inc. All rights reserved." on the right.



Join us on Slack!

<https://tobikodata.com/slack>

Thank you!

Scan me!

