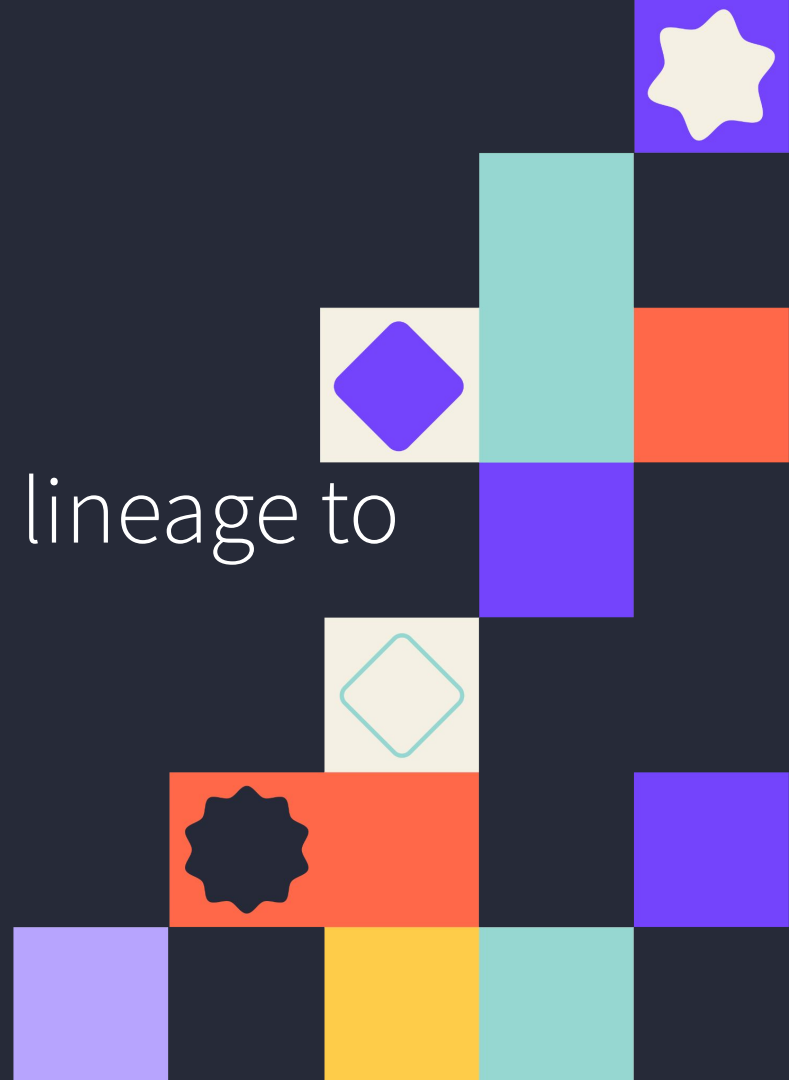




# Leveraging column-level lineage to scale your dbt projects

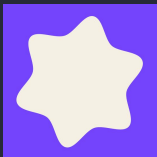
Benoit Perigaud @dbt Labs

October 7-10, 2024





Meet today's presenter from this company



Benoit Perigaud

Senior Resident Architect

dbt Labs





Senior Resident Architect  
in the **Professional  
Service** team at dbt Labs.

Based in Madrid, Spain.

## Who am I

- work with the largest dbt Cloud customers and some of the largest dbt deployments that exist
  - **100s** of dbt projects
- spend a lot of time thinking about scaling
  - number of projects / people / models
- on top of day job, I maintain tools like
  - dbt-project-evaluator
  - dbt-jobs-as-code
  - the dbt Cloud Terraform provider





# Agenda

1

What does scaling dbt mean  
More projects and more models

2

What tools and techniques  
do we have to scale  
Automating quality checks at  
scale

3

Let's talk about column-level  
lineage  
Drill down from the model level  
lineage we used to have

4

Introducing a new tool!  
Who doesn't like a new CLI

5

Demo  
Current use cases covered

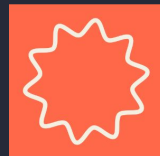
6

Call To Action  
Where YOU can help



# Scaling & tools

Getting Started





dbt can be used by small individual teams or by entire worldwide companies

Once people understand dbt and start developing, the next challenge is **how to scale**

## What does scaling dbt mean

- Scaling the overall **number of projects**
  - unrelated projects
  - or a dbt mesh approach
- Scaling the **number of models**/tests within individual projects
  - 1000+ models
  - 1000+ tests
  - dozens of contributors to the repo

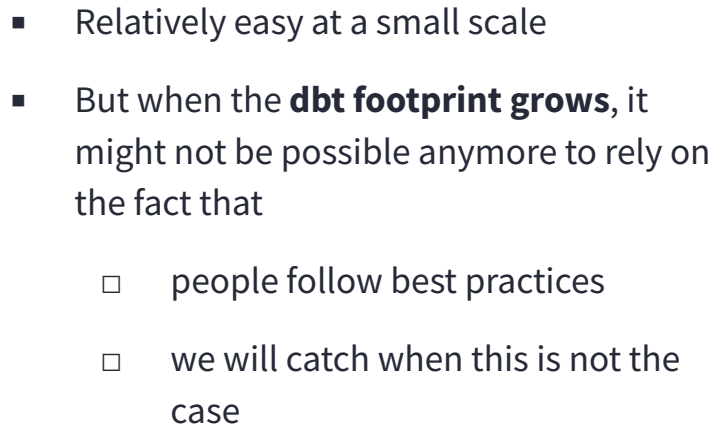
**Often, scaling issues are different from small scale problems**



Version controlling code and the software best practices it enables are great tools to set up rules and guidelines on best practices

## How to scale?

- Follow or adapt the dbt Labs **best practices**
  - how to model your data transformation
  - how to organize your file structure
- Agree on common **standards**
  - on testing
  - on documentation
- Set up a **code promotion** process with reviews of other contributors code to check alignment with those standards







To proceed, hit the Render Lineage button.

## Manual checks are good but **not scalable**

Too much time reviewing code and trying to catch mistakes

## We can't catch everything





The super power of dbt is its ability to integrate with other processes and the **ecosystem** built around it

We need tools and automation to help

And it looks like the dbt community like those tools

- **SQLFluff**: 7.7K ★
  - SQL linting and formatting
  - Not just for dbt but very popular in the dbt community
- **dbt-checkpoint**: 569 ★
  - pre-commit checks on dbt project
- **dbt-project-evaluator**: 434 ★
  - dbt package to enforce rules on dbt projects



## Focus on dbt-project-evaluator



Currently, 28 rules to check that a given dbt project is aligned with our recommended best practices

- Are the modeling layers followed as per design
- Are tests and documentation added
- Are the models named following a unified convention
- etc...

**All of that running in CI/CD**

dbt\_project\_evaluator

Home

Rules

List of rules

Modeling

Testing

Documentation

Structure

Performance

Governance

Customization

Run in CI Check

Querying the DAG

Contributing

### List of the rules currently defined

Type	Friendly name	fact name
Modeling	Staging Models Dependent on Other Staging Models	fct_staging_dependent_on_staging
Modeling	Source Fanout	fct_source_fanout
Modeling	Rejoining of Upstream Concepts	fct_rejoining_of_upstream_concepts
Modeling	Model Fanout	fct_model_fanout
Modeling	Downstream Models Dependent on Source	fct_marts_or_intermediate_dependent_on
Modeling	Direct Join to Source	fct_direct_join_to_source
Modeling	Duplicate Sources	fct_duplicate_sources
Modeling	Hard Coded References	fct_hard_coded_references
Modeling	Multiple Sources Joined	fct_multiple_sources_joined
Modeling	Root Models	fct_root_models



## Focus on dbt-project-evaluator

Many rules from dbt project evaluator are now **automatically calculated** in dbt Explorer

No need to install and configure any package


Project details

- Overview
- Performance
- Recommendations


Resources   File tree   Database

- Models 13
- Sources 6
- Tests 30
- Exposures
- Groups
- Metrics 19
- Semantic Models 6

Project summary



**Models with Tests**  
Define tests on your models to help ensure data quality for consumers.



**Models with Documentation**  
Document your models to ensure col

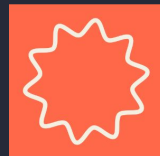
10 Recommendations

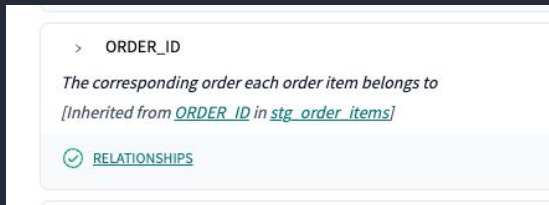
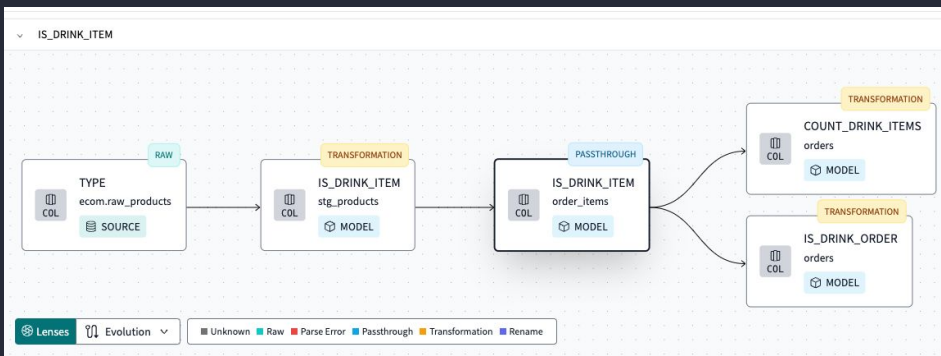
Severity	Category	Rule Name	Resource	Recommendation
High	Testing	<a href="#">Missing Primary Key Test</a>	<a href="#">supplies</a>	`supplies` is missing a primary key test. This test is important to e
High	Testing	<a href="#">Missing Primary Key Test</a>	<a href="#">products</a>	`products` is missing a primary key test. This test is important to e
High	Testing	<a href="#">Missing Primary Key Test</a>	<a href="#">metricflow_time_spine</a>	`metricflow_time_spine` is missing a primary key test. This test i



# Let's talk column-level lineage

For a bit





## Introduction of column-level lineage in dbt Explorer



Identify all the other columns upstream and downstream of a given column



Highlight what type of transformation is done on the column in each model



When upstream columns are documented, retrieve the documentation when looking at downstream columns



## column-level lineage use cases



### Root cause analysis

In production, most issues are due to **unexpected data**

Quickly identify what source might be the cause of an issue



### Impact analysis

What models and reports will be impacted if I **change the logic** of a given column

If I know that a column calculation has a bug in it it, what reports might be reporting incorrect data



### Collaboration and efficiency

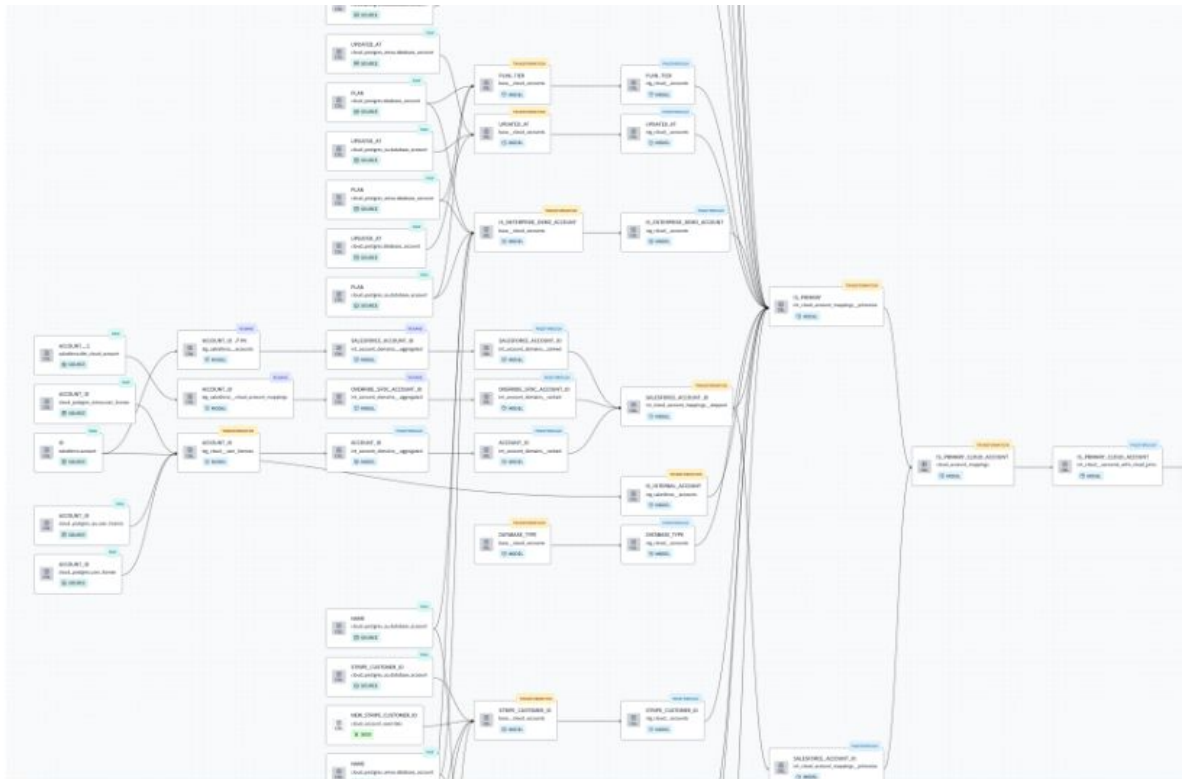
Give **confidence** to data consumer

Easier high level **exploration** of the current data transformation for Analytics Engineers



But then...

When we scale

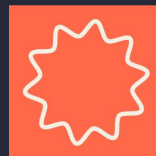


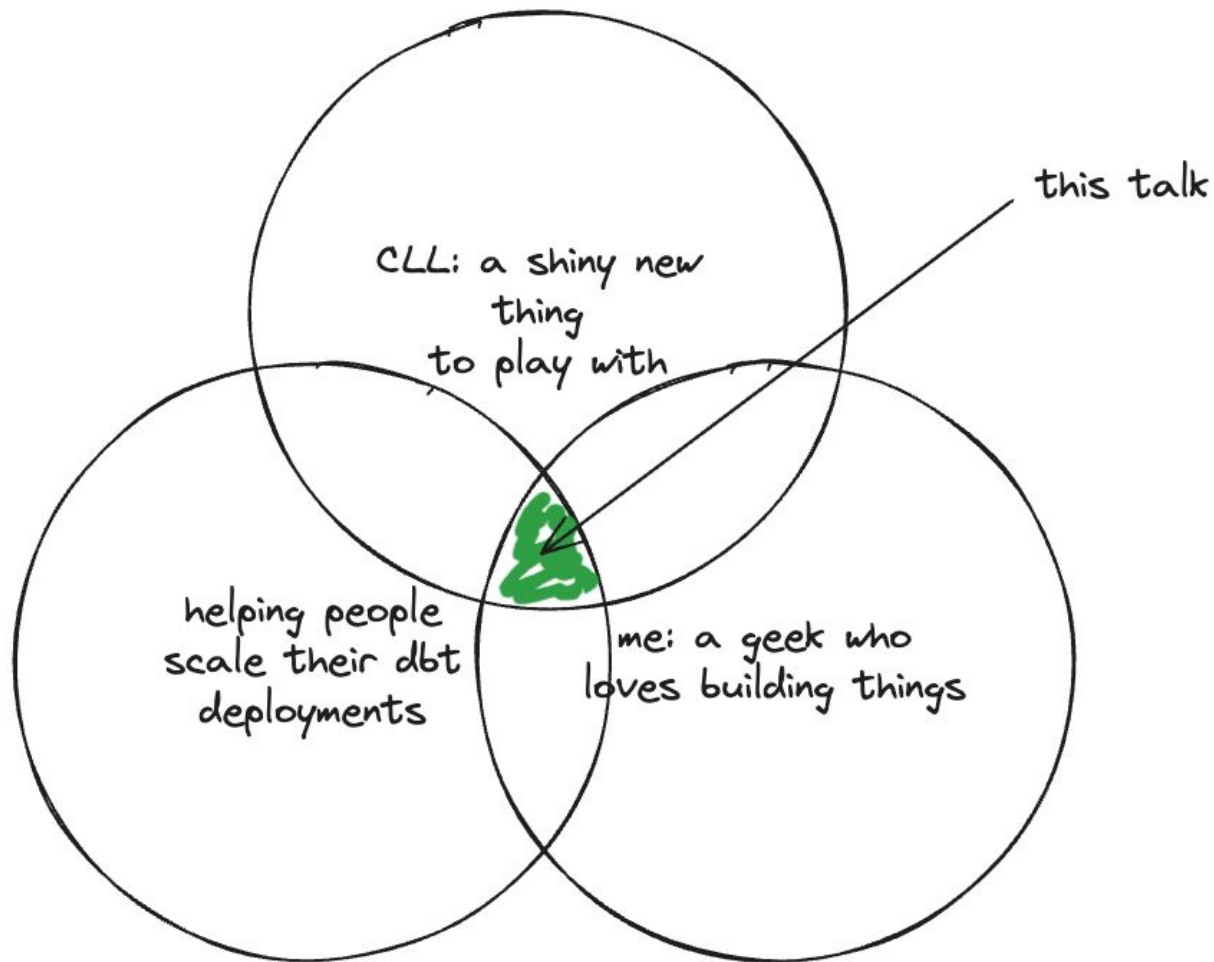




# What next

Getting Started

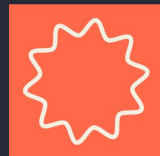






# Introducing **dbt-ctl-evaluator**

(please suggest any other better name)





dbt-cll-evaluator





## dbt-ctl-evaluator

- Open Source CLI tool to query the column-level lineage from the Discovery API and perform automated actions and checks
- A similar inspiration as **dbt-project-evaluator**, but focusing on rules at the column level rather than the model level



It uses some a dbt Cloud internal API, so it works today, but there is not guarantee that it will work tomorrow, next month or next year





Currently, only a few rules are introduced, some of them allowing to **fix your project**.

## Current rules and checks



### Column renames

Are columns renamed outside of staging models?



### Number of dependencies

How many upstream/downstream dependencies each column has



### Missing/mismatching documentation **fixable**

Is documentation missing or different when the transformation is passthrough



### Configuration lineage **fixable**

Is some configuration applied to all downstream and/or upstream columns (e.g. PII data)



## Demo - Column dependencies

```
> dbt-cll-evaluator --start-unique-id model.fishtown_internal_analytics.fct_training_enrollments num-edges
Upstream edges analysis
```

model_name	column_name	num_upstream_edges	comment
analytics.analytics.dim_all_learners	attended_courses	2	
analytics.analytics.fct_training_enrollments	user_email	1	
analytics.analytics.fct_training_enrollments	course_name	1	
analytics.analytics.fct_training_enrollments	attended_courses	1	
analytics.analytics.dim_all_learners	completed_courses	3	
analytics.analytics.fct_training_enrollments	is_completed	1	
analytics.analytics.dim_all_learners	first_started_at	2	
analytics.analytics.fct_training_enrollments	started_at	1	
analytics.analytics.dim_all_learners	is_partner	1	
analytics.analytics.fct_training_enrollments	is_partner	1	
analytics.analytics.dim_all_learners	last_logged_in_at	2	
analytics.analytics.fct_training_enrollments	last_logged_in_at	1	
analytics.analytics.dim_all_learners	learner_email	2	
analytics.analytics.fct_cloud_users	enrolled_in_learn	1	
analytics.analytics.dim_all_learners	learner_email_domain	1	
analytics.analytics.fct_training_enrollments	user_email_domain	1	
analytics.analytics.dim_all_learners	learner_name	2	
analytics.analytics.fct_training_enrollments	user_name	1	
analytics.analytics.dim_all_learners	partner_name	1	

- Can flag columns with no upstream or downstream dependencies
- Can flag **critical columns** with many upstream and/or downstream dependencies



## Demo - Missing or different description (fixable)

```
> dbt-cll-evaluator --start-unique-id model.fishtown_internal_analytics.fct_support_tickets desc-difference  
Different descriptions
```

upstream id	upstream desc	downstream id	downstream desc	comment
analytics.analytics.stg_c...	...	analytics.analytics.cloud...	...	Description is missing downstream
analytics.analytics.fct_s...	...	analytics.analytics.fct_s...	...	Description is different
analytics.analytics.count...	...	analytics.analytics.int_z...	...	Description is missing downstream
analytics.analytics.stg_z...	...	analytics.analytics.int_z...	...	Description is missing downstream
analytics.analytics.stg_z...	...	analytics.analytics.int_z...	...	Description is missing downstream





## Demo - Configuration lineage

```
> dbt-cll-evaluator --start-unique-id model.fishtown_internal_analytics.stg_jira_users config-lineage --key tags --value has_pii --dbt-project-path ~/dev/internal-analytics
```

*Missing config*

column_with_config	column_without_config	model_without_config	comment
analytics.analytics.stg_jira_us...	raw.fivetran_jira.user.name	source.fishtown_internal_analyti...	Upstream column is missing the config tags: has_pii
analytics.analytics.stg_jira_us...	analytics.analytics.fct_jira_issu...	model.fishtown_internal_analytic...	Downstream column is missing the config tags: has_pii
analytics.analytics.stg_jira_us...	analytics.analytics.fct_jira_issu...	model.fishtown_internal_analytic...	Downstream column is missing the config tags: has_pii
analytics.analytics.stg_jira_us...	analytics.analytics.fct_jira_issu...	model.fishtown_internal_analytic...	Downstream column is missing the config tags: has_pii



## Demo - Configuration lineage - with fix

```
dbt-ctl-evaluator --stage unique_id model.fishtown_internal_analytics.stg_jira_users config-lineage --key tags --value has_pii --dbt-project-path ~/dev/internal-analytics --fix
```

Missing config

column_with_config	column_without_config	model_without_config	comment
analytics.analytics.stg_jira_us...	raw.fivetran_jira.user.name	source.fishtown_internal_analyti...	Upstream column is missing the config tags: has_pii
analytics.analytics.stg_jira_us...	analytics.analytics.fct_jira_issu...	model.fishtown_internal_analytic...	Downstream column is missing the config tags: has_pii
analytics.analytics.stg_jira_us...	analytics.analytics.fct_jira_issu...	model.fishtown_internal_analytic...	Downstream column is missing the config tags: has_pii
analytics.analytics.stg_jira_us...	analytics.analytics.fct_jira_issu...	model.fishtown_internal_analytic...	Downstream column is missing the config tags: has_pii

```
2024-09-30 09:40:15.949 | WARNING | dbt-ctl-evaluator.dbt_operations:update_column_config:126 - Could not find node_id source.fishtown_internal_analytics.jira.user in the dbt manifest
2024-09-30 09:40:15.968 | INFO | dbt-ctl-evaluator.dbt_operations:update_column_config:100 - Could not find column assignee_name in the YML file /Users/bper/dev/internal-analytics/models/marts/tasks/_tasks_models.yml for model fct_jira_issues, creating it
2024-09-30 09:40:16.008 | SUCCESS | dbt-ctl-evaluator.dbt_operations:update_column_config:121 - Updated the YML file /Users/bper/dev/internal-analytics/models/marts/tasks/_tasks_models.yml with the tag has_pii for column assignee_name in model fct_jira_issues
2024-09-30 09:40:16.028 | INFO | dbt-ctl-evaluator.dbt_operations:update_column_config:100 - Could not find column creator_name in the YML file /Users/bper/dev/internal-analytics/models/marts/tasks/_tasks_models.yml for model fct_jira_issues, creating it
2024-09-30 09:40:16.037 | SUCCESS | dbt-ctl-evaluator.dbt_operations:update_column_config:121 - Updated the YML file /Users/bper/dev/internal-analytics/models/marts/tasks/_tasks_models.yml with the tag has_pii for column creator_name in model fct_jira_issues
2024-09-30 09:40:16.056 | INFO | dbt-ctl-evaluator.dbt_operations:update_column_config:100 - Could not find column reporter_name in the YML file /Users/bper/dev/internal-analytics/models/marts/tasks/_tasks_models.yml for model fct_jira_issues, creating it
2024-09-30 09:40:16.065 | SUCCESS | dbt-ctl-evaluator.dbt_operations:update_column_config:121 - Updated the YML file /Users/bper/dev/internal-analytics/models/marts/tasks/_tasks_models.yml with the tag has_pii for column reporter_name in model fct_jira_issues
```



```
models:
  - name: fct_jira_issues
    columns:
      - name: assignee_name
        tags:
          - has_pii
      - name: creator_name
        tags:
          - has_pii
      - name: reporter_name
        tags:
          - has_pii
```



What else can we solve  
analyzing our  
transformations **at the  
column level?**

What other use cases?



Column names reuse

What different columns use the  
same column name?



Identify same columns being used  
upstream to create another column

Potentially duplicating of logic or  
different logic for the same concept



Optimize testing

If a column is passthrough and  
tested upstream, do we need to  
retest it entirely?



Any more idea???

## Caveats / Good to know



This is more of a thought experiment than a tool to rely on daily for production loads

**Start discussions** on what CLL can be leveraged for, and what use cases are the key ones, for which we can create a better experience.



As we rely on the Metadata API, there are a couple of limitations today

- we **can't run this tool in CI/CD** as column-level lineage takes some time to be computed
- the API used is **internal**, it might not work in the future

The lineage can't be parsed in some cases

Also today it is run as **+my\_model+**, focusing on one model and all of its upstream/downstream dependencies

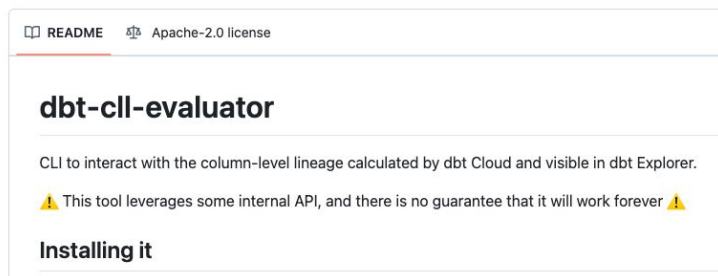


## What's next



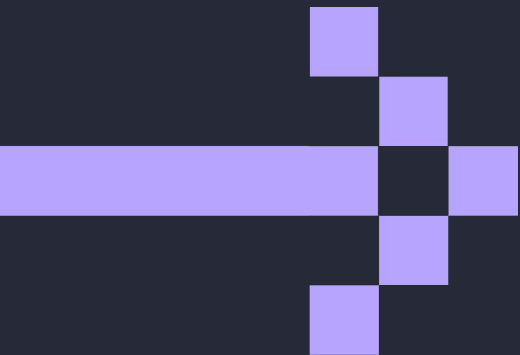
Use the tool if you want to try it out!

<https://github.com/dbt-labs/dbt-cll-evaluator>



Raise feature requests and discussions in the Github repository

**Reach out** to me or to other folks at dbt Labs about what problems you have that could be solved by column-level lineage



Questions?