

# dbt-project

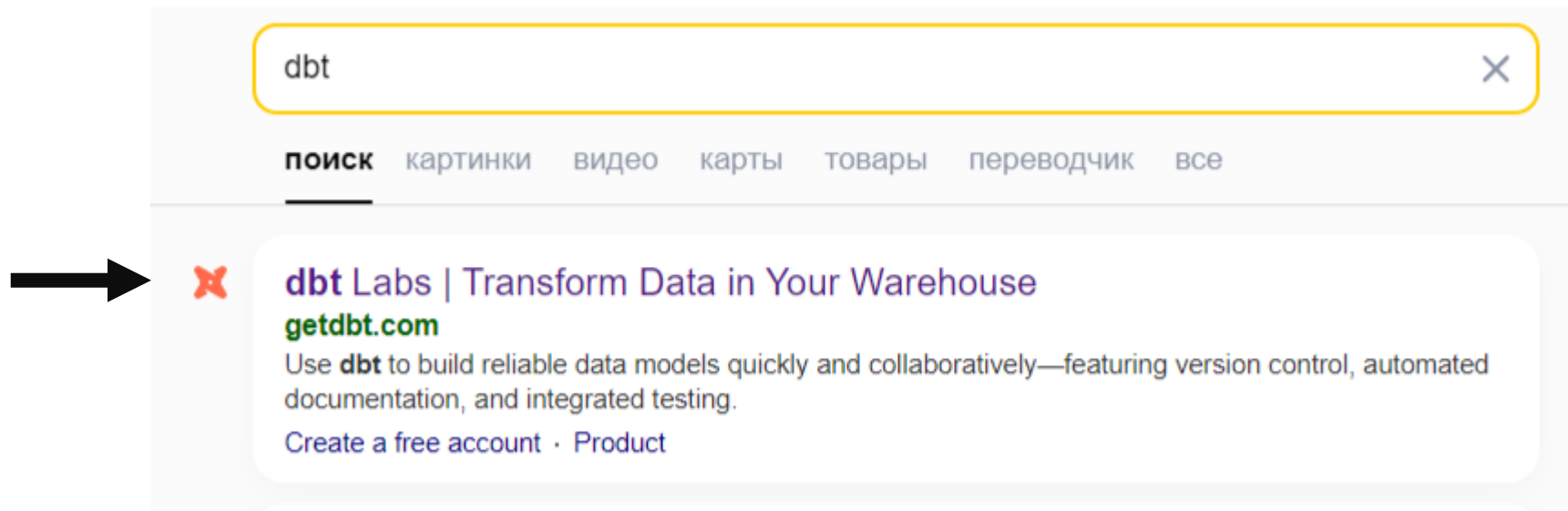
API Youtube/Python + Google Big Query + dbt

## Краткое описание проекта

У меня есть серия исследований каналов Youtube. Данный проект – одна из вариаций на эту тему.

За основу я взяла данные, полученные по **API Youtube**. Эти несколько датасетов я загрузила в **Google Big Query**. И они послужили материалом для проекта **dbt**.

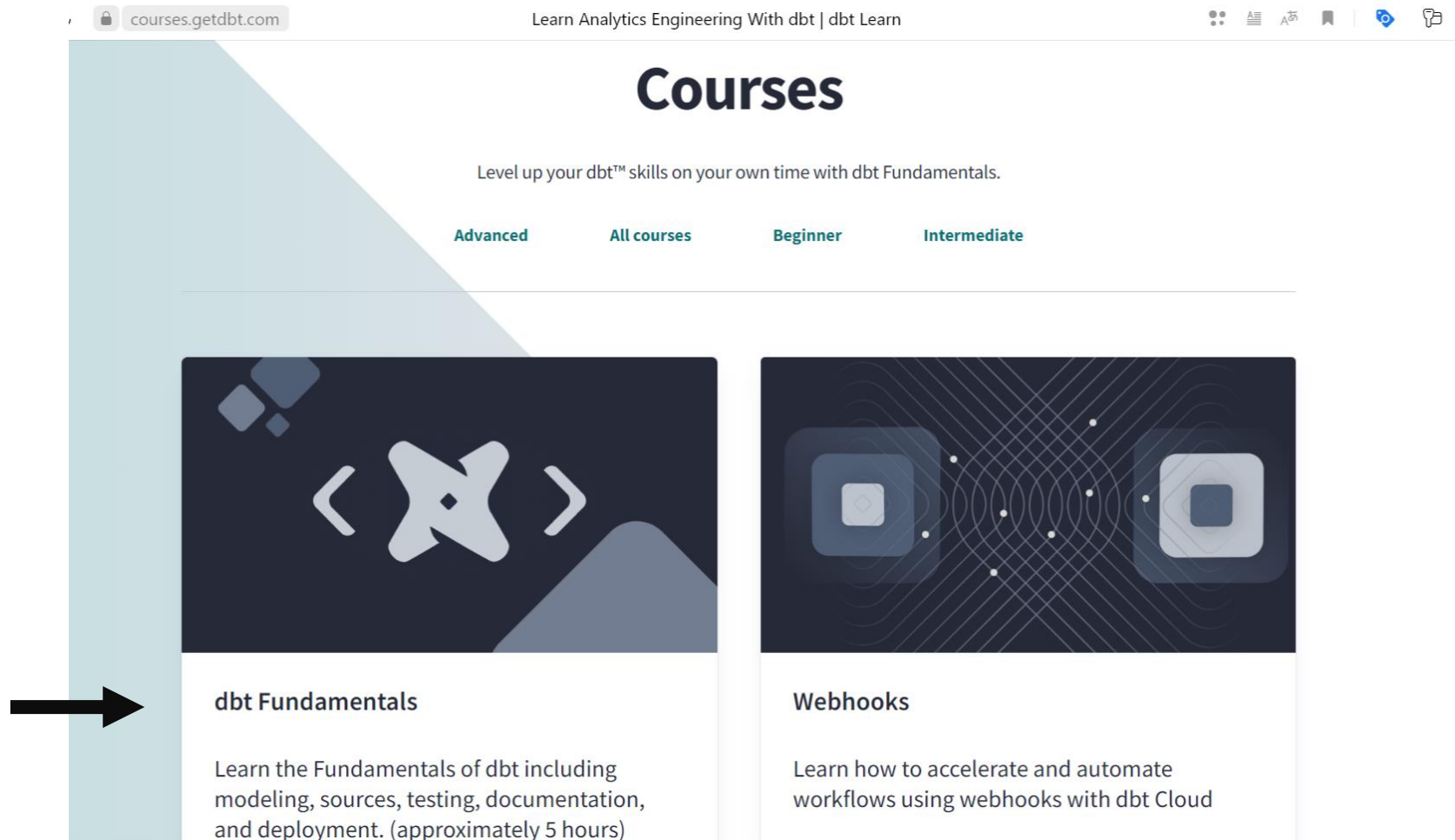
Далее я расскажу и покажу, как устроен мой проект, созданный на **getdbt.com**.

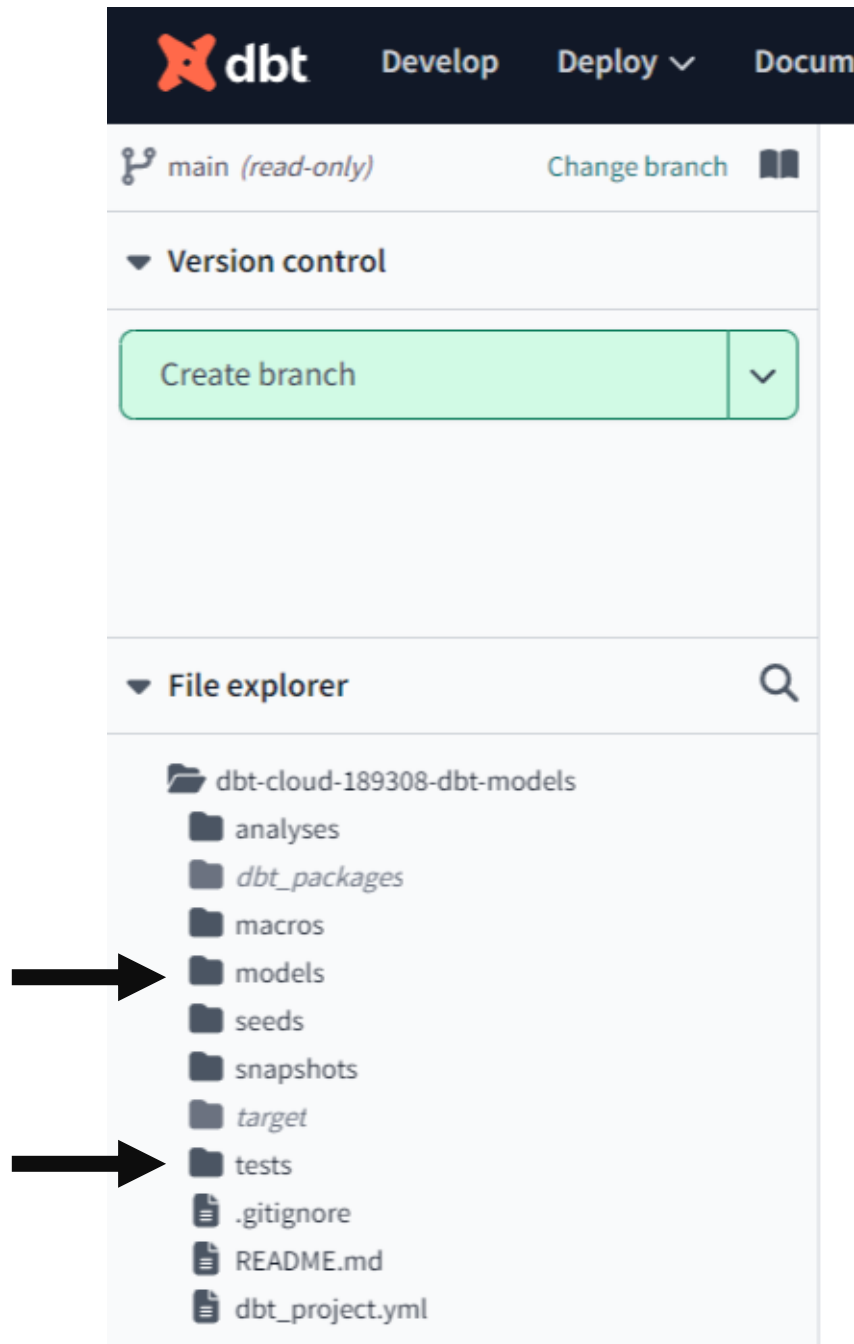


**Изучить основы dbt** можно на бесплатном курсе dbt Fundamentals:

<https://courses.getdbt.com/collections>

Этот курс помог мне освоить dbt и создать данный проект.



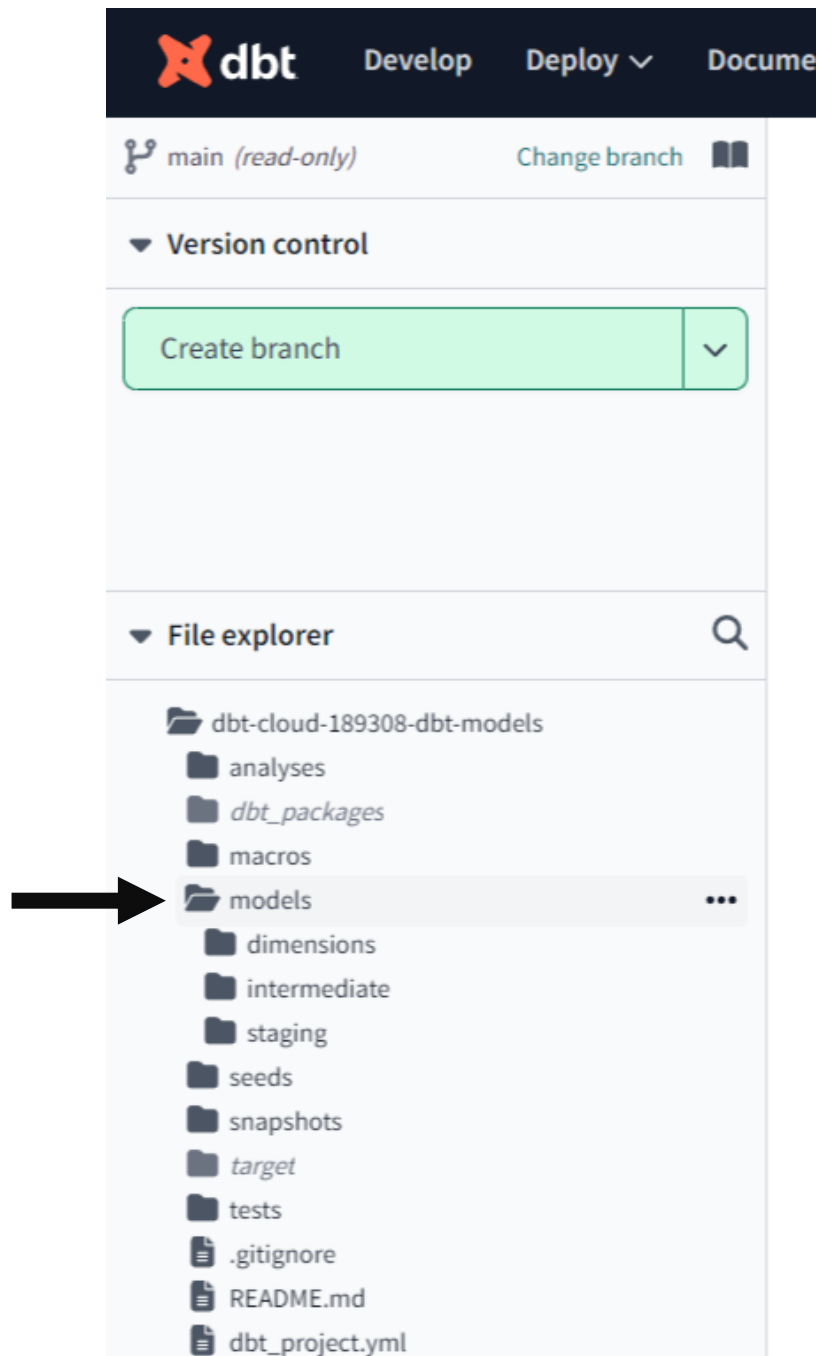


Как выглядит проект:  
это несколько папок.

Основные – это **models** и **tests**.

**models** – здесь находятся dbt-модели,  
разделенные по папкам.  
Это – сердцевина всего проекта.

**tests** – это тесты, они имеют скорее  
вспомогательную функциональность.  
В принципе можно обходиться  
и вовсе без них (но с ними, всё-таки,  
интересней).



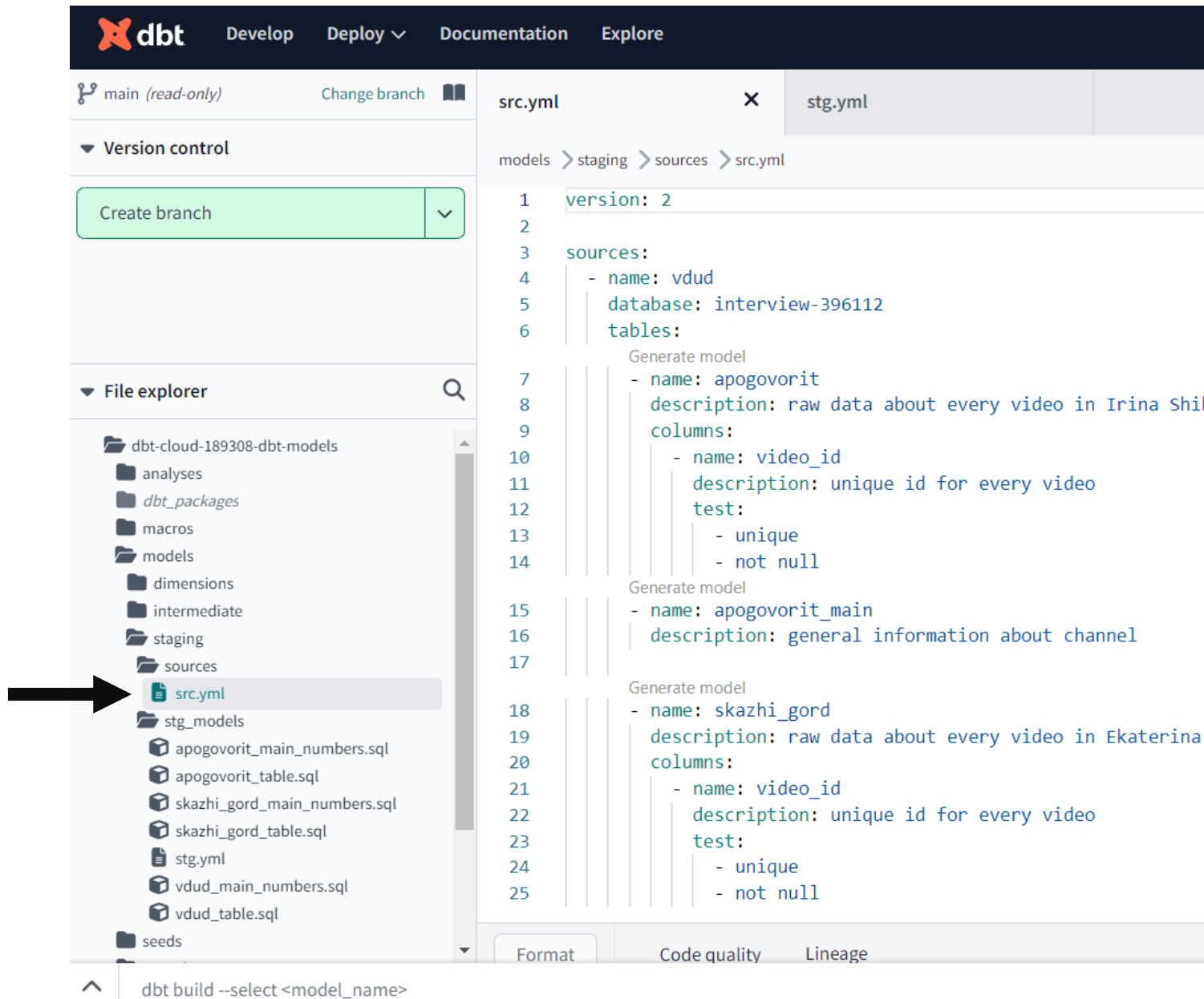
Рассмотрим подробнее **models**.

Модели внутри я распределила по папкам для большей наглядности и структурности.

В моём проекте внутри models 3 папки:

- **dimensions**: итоговый слой
- **intermediate**: слой для преобразований и расчётов
- **staging**: слой для сырых данных

Далее рассмотрим проект по пути следования данных: от «сырого» слоя до документации.



В папке **staging** 2 папки:

**sources** – здесь в файле `src.yml` описано подключение к источнику данных (Big Query) и все подробности о данных. Благодаря файлам `.yml` документация проекта формируется автоматически. Поэтому такие файлы находятся у меня в каждой папке.

**stg\_models** – здесь в файлах `.sql` содержатся первые dbt-модели, а также есть и `yml`-файл

---

\* все упоминаемые авторы youtube-каналов (Дудь, Шихман, Гордеева) - признаны иноагентами

dbt Cloud interface showing the `stg.yml` file in the staging layer. The file defines three models:

- `apogovorit_main_numbers`: description: all information from apogovorit\_main with small change
- `subscriber_count`: description: number of subscribers on the channel
- `video_count`: description: number of videos on the channel

The file also defines tests for each model, including `not_null`, `unique`, and `not_null`.

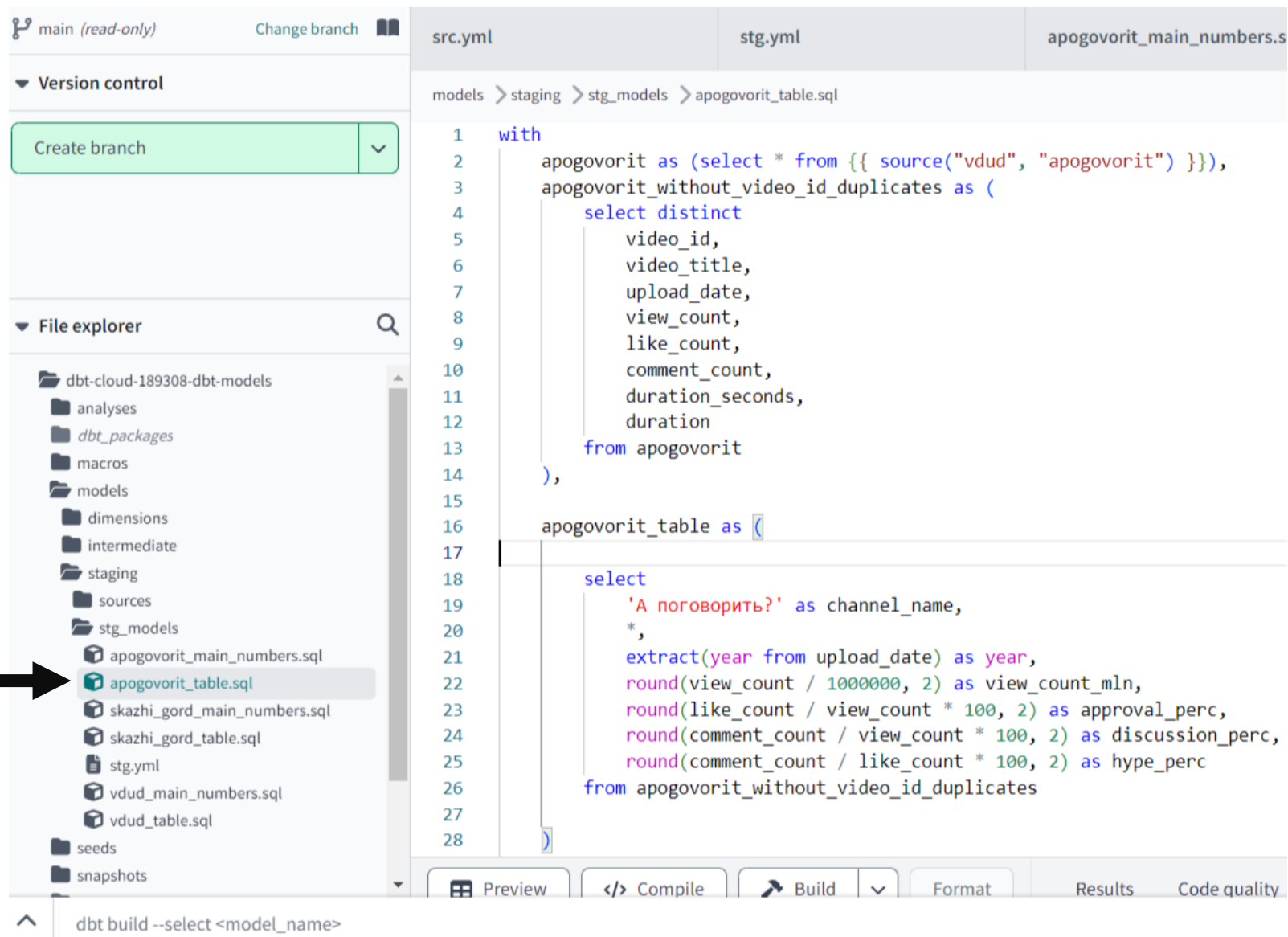
The file explorer on the left shows the project structure, with `stg.yml` highlighted.

В файле `stg.yml` дано подробное описание самих моделей (таблиц):

- название
- описание
- колонки
- название
- описание
- тесты

Таким образом, файлы `.yml` помогают не только создавать и документировать структуру проекта, но и очень простым способом сразу добавлять типовые тесты (`unique`, `not_null`)

-----  
\* все упоминаемые авторы youtube-каналов (Дудь, Шихман, Гордеева) - признаны иноагентами



The screenshot shows the dbt Cloud interface. On the left, the 'File explorer' panel displays a directory structure with folders like 'analyses', 'dbt\_packages', 'macros', 'models', 'dimensions', 'intermediate', 'staging', 'sources', and 'stg\_models'. A black arrow points to the file 'apogovorit\_table.sql' in the 'stg\_models' folder. The main editor area shows the content of this file, which is a SQL model. The file path at the top is 'models > staging > stg\_models > apogovorit\_table.sql'. The SQL code is as follows:

```
1 with
2   apogovorit as (select * from {{ source("vdud", "apogovorit") }}),
3   apogovorit_without_video_id_duplicates as (
4     select distinct
5       video_id,
6       video_title,
7       upload_date,
8       view_count,
9       like_count,
10      comment_count,
11      duration_seconds,
12      duration
13    from apogovorit
14  ),
15  apogovorit_table as (
16
17
18    select
19      'А поговорить?' as channel_name,
20      *,
21      extract(year from upload_date) as year,
22      round(view_count / 1000000, 2) as view_count_mln,
23      round(like_count / view_count * 100, 2) as approval_perc,
24      round(comment_count / view_count * 100, 2) as discussion_perc,
25      round(comment_count / like_count * 100, 2) as hype_perc
26    from apogovorit_without_video_id_duplicates
27
28  )
```

At the bottom of the interface, there is a command bar with the text 'dbt build --select <model\_name>'.

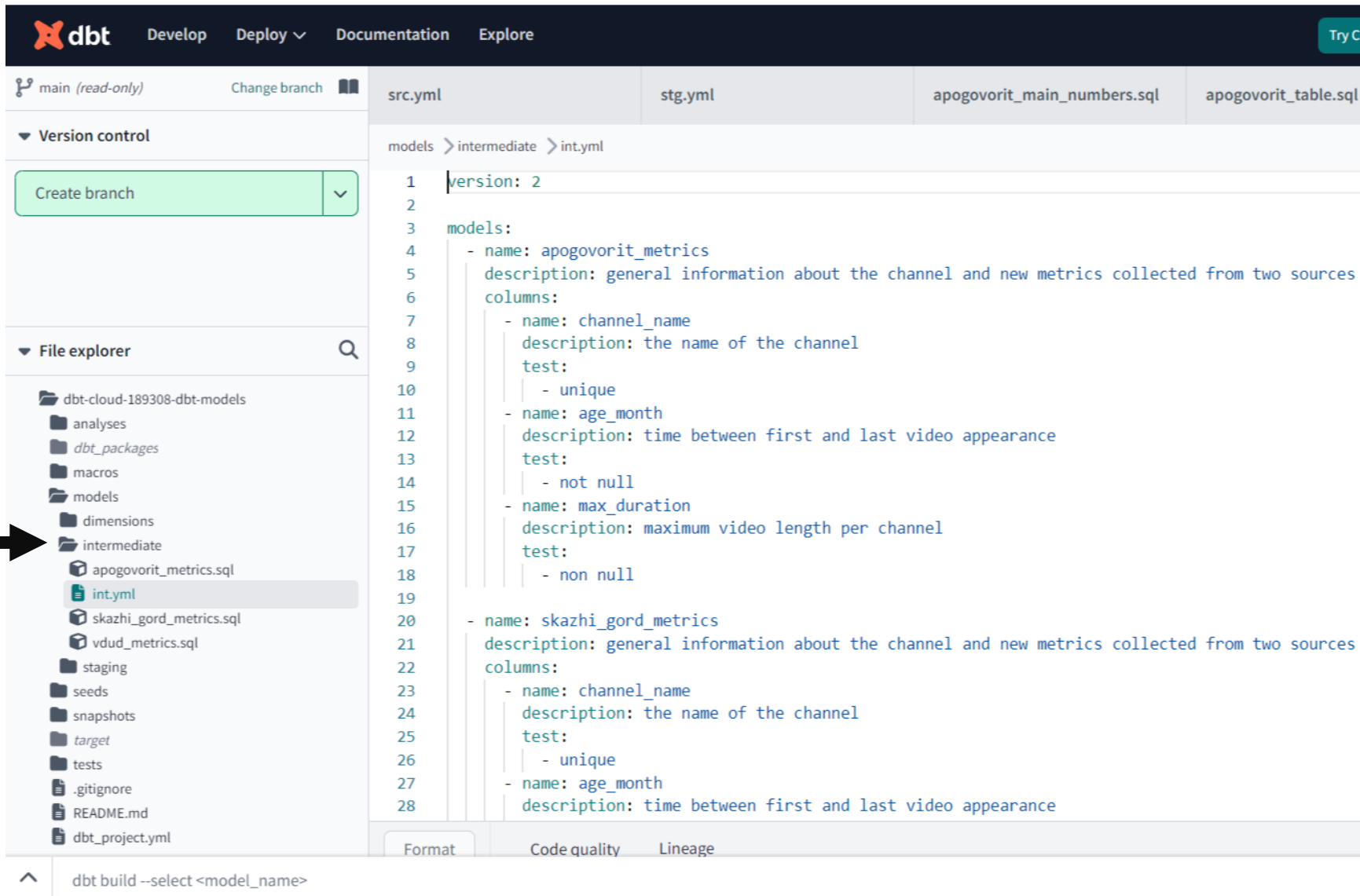
Если посмотреть какой-то sql-файл, то внутри мы увидим вполне привычные sql-выражения, но кроме этого и внутренние ссылки на другие шаги проекта.

Эти ссылки оформляются при помощи двойных фигурных скобок

---

\* все упоминаемые авторы youtube-каналов (Дудь, Шихман, Гордеева) - признаны иноагентами





The screenshot shows the dbt Cloud web interface. On the left, the 'File explorer' sidebar lists the project structure. A black arrow points to the 'intermediate' folder, which contains the 'int.yml' file. The main editor area displays the content of 'int.yml', which defines two models: 'apogovorit\_metrics' and 'skazhi\_gord\_metrics'. The interface includes a top navigation bar with 'dbt', 'Develop', 'Deploy', 'Documentation', and 'Explore' tabs. Below the navigation bar, there's a 'Version control' section with a 'Create branch' button. The bottom of the interface shows a command prompt with the text 'dbt build --select <model\_name>'.

```
1 version: 2
2
3 models:
4   - name: apogovorit_metrics
5     description: general information about the channel and new metrics collected from two sources
6     columns:
7       - name: channel_name
8         description: the name of the channel
9         test:
10           - unique
11       - name: age_month
12         description: time between first and last video appearance
13         test:
14           - not null
15       - name: max_duration
16         description: maximum video length per channel
17         test:
18           - non null
19
20   - name: skazhi_gord_metrics
21     description: general information about the channel and new metrics collected from two sources
22     columns:
23       - name: channel_name
24         description: the name of the channel
25         test:
26           - unique
27       - name: age_month
28         description: time between first and last video appearance
```

Следующая после staging папка – **intermediate**

здесь также есть yml-файл с описанием моделей, их колонок, тестов и т.п.

---

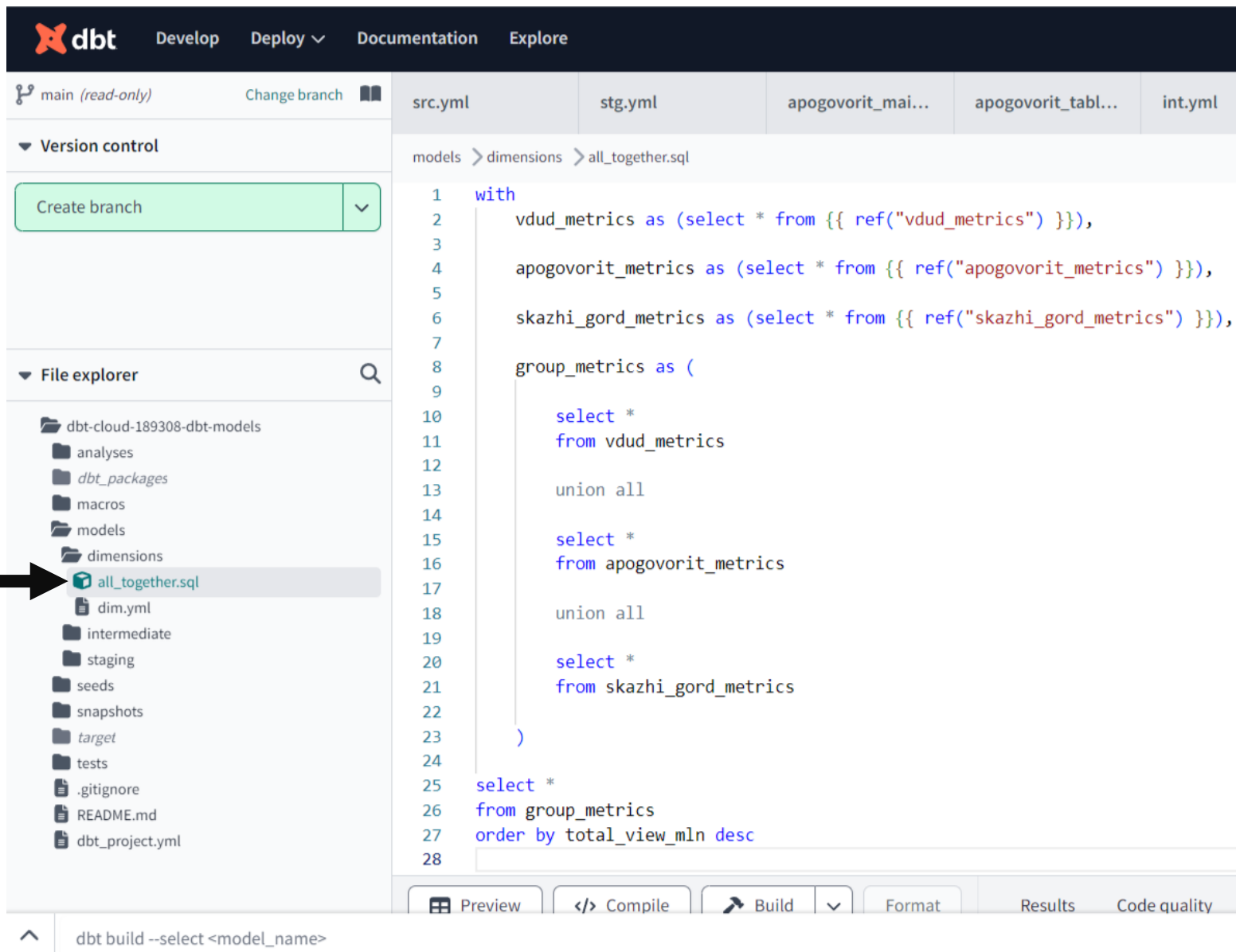
\* все упоминаемые авторы youtube-каналов (Дудь, Шихман, Гордеева) - признаны иноагентами

dbt Cloud interface showing the 'apogovorit\_metrics.sql' model file. The file is located in the 'models' > 'intermediate' directory. The code defines a model with the following structure:

```
1 with
2   apogovorit_table as (select * from {{ ref("apogovorit_table") }}),
3
4   apogovorit_main as (select * from {{ ref("apogovorit_main_numbers") }}),
5
6   apogovorit_table_metrics as (
7
8     select
9       channel_name,
10      min(upload_date) as first_date,
11      max(upload_date) as last_date,
12      date_diff(max(upload_date), min(upload_date), month) as age_month,
13      max(view_count_mln) as max_view_mln,
14      round(avg(view_count_mln), 2) as avg_view_mln,
15      max(duration) as max_duration,
16      round(avg(duration_seconds) / 60) as avg_dur_minutes,
17      max(approval_perc) as max_approval_perc,
18      round(avg(approval_perc), 2) as avg_approval_perc,
19      max(discussion_perc) as max_discussion_perc,
20      round(avg(discussion_perc), 2) as avg_discussion_perc,
21      max(hype_perc) as max_hype_perc,
22      round(avg(hype_perc), 2) as avg_hype_perc
23    from apogovorit_table
24    group by 1
25  ),
26
27  apogovorit_metrics as (
28    select
```

На этом шаге происходят различные расчёты метрик

-----  
\* все упоминаемые авторы youtube-каналов (Дудь, Шихман, Гордеева) - признаны иноагентами



dbt Cloud interface showing the file explorer on the left and the SQL editor on the right. The file explorer highlights the file `all_together.sql` under the `dimensions` folder. The SQL editor displays the following query:

```
1 with
2   vdud_metrics as (select * from {{ ref("vdud_metrics") }}),
3
4   apogovorit_metrics as (select * from {{ ref("apogovorit_metrics") }}),
5
6   skazhi_gord_metrics as (select * from {{ ref("skazhi_gord_metrics") }}),
7
8   group_metrics as (
9
10    select *
11    from vdud_metrics
12
13    union all
14
15    select *
16    from apogovorit_metrics
17
18    union all
19
20    select *
21    from skazhi_gord_metrics
22  )
23
24 select *
25 from group_metrics
26 order by total_view_mln desc
27
```

На последнем шаге происходит объединение данных из разных источников

---

\* все упоминаемые авторы youtube-каналов (Дудь, Шихман, Гордеева) - признаны иноагентами

The screenshot shows the dbt Cloud web interface. At the top, there's a navigation bar with 'dbt' logo and links for 'Develop', 'Deploy', 'Documentation', and 'Explore'. Below this, the 'main (read-only)' branch is selected. The left sidebar contains 'Version control' with a 'Create branch' button and 'File explorer' showing a directory tree. The 'tests' folder is highlighted, and a file named 'views\_likes\_comments\_duration\_not\_negative.sql' is selected, indicated by a black arrow. The main area displays the SQL code for this test, which uses CTEs to define tables and then performs a union of two queries to test for negative values in view\_count, like\_count, or comment\_count. At the bottom, there's a status bar with buttons for 'Preview', 'Compile', 'Build', 'Format', 'Results', and 'Code quality'.

dbt Cloud interface showing a test file in the `tests` directory. The file is named `views_likes_comments_duration_not_negative.sql`. The test code is as follows:

```
1 with
2   vdud_table as (select * from {{ ref("vdud_table") }}),
3   skazhi_gord_table as (select * from {{ ref("skazhi_gord_table") }}),
4   apogovorit_table as (select * from {{ ref("apogovorit_table") }})
5
6 select video_id, view_count, like_count, comment_count
7 from vdud_table
8 where view_count < 0 or like_count < 0 or comment_count < 0
9 union all
10 select video_id, view_count, like_count, comment_count
11 from skazhi_gord_table
12 where view_count < 0 or like_count < 0 or comment_count < 0
13 union all
14 select video_id, view_count, like_count, comment_count
15 from apogovorit_table
16 where view_count < 0 or like_count < 0 or comment_count < 0
17
```

Кроме типовых тестов (not\_null, unique) можно создать свои собственные тесты – они находятся в папке **tests**.

Например, я создала тест на выявление отрицательных значений по количеству

- просмотров,
- лайков,
- комментариев.

---

\* все упоминаемые авторы youtube-каналов (Дудь, Шихман, Гордеева) - признаны иноагентами

The screenshot displays the dbt Cloud web interface. At the top, there's a navigation bar with the dbt logo and links for 'Develop', 'Deploy', 'Documentation', and 'Explore'. Below this, a breadcrumb shows 'main (read-only)' with a 'Change branch' link. The left sidebar contains two main sections: 'Version control' with a 'Create branch' button, and 'File explorer' which lists the project's directory structure. In the 'File explorer', the 'tests' folder is expanded, and the file 'final\_test.sql' is highlighted with a black arrow pointing to it. The main area of the interface shows the content of 'final\_test.sql', which is a SQL test script. The script starts with a CTE 'all\_together' that selects from a ref, followed by a select statement with various max and avg functions. The bottom of the interface has a toolbar with buttons for 'Preview', 'Compile', 'Build', 'Format', and 'Results'. A terminal at the very bottom shows the command 'dbt build --select <model\_name>'.

dbt Cloud interface showing a SQL test file.

Navigation: main (read-only) | Change branch

Version control: Create branch

File explorer: dbt-cloud-189308-dbt-models | analyses | dbt\_packages | macros | models | seeds | snapshots | target | tests | .gitkeep | **final\_test.sql** | views\_likes\_comments\_duration\_not\_neg... | .gitignore | README.md | dbt\_project.yml

SQL Test Code (final\_test.sql):

```
1 with all_together as (select * from {{ ref("all_together") }})
2
3 select channel_name
4 from all_together
5 group by channel_name
6 having
7     max(last_date) < max(first_date)
8     or max(max_approval_perc) < max(avg_approval_perc)
9     or max(max_discussion_perc) < max(avg_discussion_perc)
10    or max(max_hype_perc) < max(avg_hype_perc)
11
```

Toolbar: Preview | Compile | Build | Format | Results

Terminal: dbt build --select <model\_name>

Также в моём проекте есть ещё один собственный тест на определение ошибочных ситуаций – когда максимальные значения по каналу меньше средних значений по тому же каналу (т.е. это тест на адекватность расчётов метрик)

The screenshot shows the dbt Cloud web interface. At the top is a dark navigation bar with the dbt logo and links for 'Develop', 'Deploy', 'Documentation', and 'Explore'. Below this, the left sidebar contains three main sections: 'main (read-only)' with a 'View docs' button, 'Version control' with a 'Create branch' button, and 'File explorer' showing a directory tree. The 'File explorer' tree includes folders like 'analyses', 'dbt\_packages', 'macros', 'models', 'seeds', 'snapshots', 'target', 'tests' and files like '.gitignore', 'README.md', and 'dbt\_project.yml'. The main area displays the 'dbt\_project.yml' file with its YAML content. At the bottom of the editor are buttons for 'Format' and 'Code quality'.

dbt Cloud interface showing the `dbt_project.yml` file.

Navigation bar: dbt, Develop, Deploy, Documentation, Explore.

Left sidebar:

- main (read-only) → View docs
- Version control: Create branch
- File explorer: dbt-cloud-189308-dbt-models
  - analyses
  - dbt\_packages
  - macros
  - models
  - seeds
  - snapshots
  - target
  - tests
  - .gitignore
  - README.md
  - dbt\_project.yml

Right pane (dbt\_project.yml):

```
1 # Name your project! Project names should consist of lowercase
2 # characters and underscores. A good convention is to use snake
3 # case for the project name.
4 name: "vdud"
5 version: "1.0.0"
6 config-version: 2
7
8 # This setting configures the default profile for the project.
9 profile: "default"
10
11 # These configurations are used to generate the dbt project.
12 # The `source-paths` configuration is used to specify the
13 # location of the source files.
14 model-paths: ["models"]
15 analysis-paths: ["analyses"]
16 test-paths: ["tests"]
17 seed-paths: ["seeds"]
18 macro-paths: ["macros"]
19 snapshot-paths: ["snapshots"]
20
21 target-path: "target" # The target directory
```

Buttons: Format, Code quality

Теперь перейдём к документации проекта.

На неё можно перейти при помощи нажатия на иконку с раскрытой книгой.

Напомню, dbt Docs формируются автоматически, если в каждой папке своего проекта вы создаёте и заполняете yaml-файлы.

При открытии dbt Docs мы видим много всего. Для примера я открыла финальную модель all\_together. Документация по модели начинается с деталей, описания, (далее для просмотра спускаемся вниз)...

The screenshot shows the dbt Docs interface in a web browser. The address bar shows 'cloud.getdbt.com' and 'dbt Docs'. The left sidebar displays a project structure with folders like 'skazhi\_gord\_main', 'vdud', and 'vdud\_main'. Under 'Projects', there's a 'models' folder containing 'dimensions', which is highlighted with a black arrow. Inside 'dimensions', the 'all\_together' model is selected. The main content area shows the 'all\_together' view details, including a search bar, tabs for 'Details', 'Description', 'Columns', 'Referenced By', 'Depends On', and 'Code'. The 'Details' tab is active, showing a table with columns: TAGS, OWNER, TYPE, PACKAGE, LANGUAGE, RELATION, ACCESS, and VERSION. The 'Description' tab shows a text box with 'general information about all channels and their new metrics'. The 'Columns' tab shows a table with columns: COLUMN, TYPE, DESCRIPTION, TESTS, and MORE?. A black arrow points to the top right corner of the interface, and a blue circular icon with a white 'E' is visible in the bottom right corner.

cloud.getdbt.com dbt Docs

Search for models...

dbt

skazhi\_gord\_main  
vdud  
vdud\_main

Projects

vdud  
models  
dimensions  
all\_together  
intermediate  
apogovorit\_metrics  
skazhi\_gord\_metrics  
vdud\_metrics  
staging  
stg\_models  
apogovorit\_main\_numbers  
apogovorit\_table  
skazhi\_gord\_main\_numbers  
skazhi\_gord\_table  
vdud\_main\_numbers  
vdud\_table  
tests  
final\_test  
views\_likes\_com... on\_not\_negative

all\_together view

Details Description Columns Referenced By Depends On Code

Details

TAGS	OWNER	TYPE	PACKAGE	LANGUAGE	RELATION	ACCESS	VERSION
untagged		view	vdud	sql	interview-396112.dbt_nmalakhova.all_together	protected	

Description

general information about all channels and their new metrics

Columns

COLUMN	TYPE	DESCRIPTION	TESTS	MORE?
channel_name	STRING	the names of the channels	U	>
total_view mln	FLOAT64	total number of views		>
subscriber mln	FLOAT64	number of subscribers		>
total_video	INT64	total number of videos		>

... далее мы видим колонки с типами данных и кратким описанием каждой...

cloud.getdbt.com dbt Docs

dbt

Search for models...

skazhi\_gord\_main  
vdud  
vdud\_main

Projects

vdud

models

dimensions

all\_together

intermediate

apogovorit\_metrics  
skazhi\_gord\_metrics  
vdud\_metrics

staging

stg\_models

apogovorit\_main\_numbers  
apogovorit\_table  
skazhi\_gord\_main\_numbers  
skazhi\_gord\_table  
vdud\_main\_numbers  
vdud\_table

tests

final\_test  
views\_likes\_com... on\_not\_negative

**all\_together** view

Details Description Columns Referenced By Depends On Code

Columns

COLUMN	TYPE	DESCRIPTION	TESTS	MORE?
channel_name	STRING	the names of the channels	U	>
total_view_mln	FLOAT64	total number of views		>
subscriber_mln	FLOAT64	number of subscribers		>
total_video	INT64	total number of videos		>
first_date	DATE	the first date of video appearance		>
last_date	DATE	the last date of video appearance		>
age_month	INT64	monthes between first and last ...		>
max_view_mln	FLOAT64	maximum view_count		>
avg_view_mln	FLOAT64	average view_count		>
max_duration	TIME	maximum duration		>
avg_dur_minutes	FLOAT64	average duration in minutes		>



## ...ССЫЛКИ, ЗАВИСИМОСТИ, ...

← ↻ 🔒 cloud.getdbt.com dbt Docs

dbt

Search for models...

skazhi\_gord\_main  
vdud  
vdud\_main

Projects

- vdud
  - models
    - dimensions
      - all\_together**
    - intermediate
      - apogovorit\_metrics
      - skazhi\_gord\_metrics
      - vdud\_metrics
    - staging
      - stg\_models
        - apogovorit\_main\_numbers
        - apogovorit\_table
        - skazhi\_gord\_main\_numbers
        - skazhi\_gord\_table
        - vdud\_main\_numbers
        - vdud\_table
    - tests
      - final\_test
      - views\_likes\_com... on\_not\_negative

**all\_together** view

Details Description Columns Referenced By Depends On Code

Referenced By

Tests



final\_test  
unique\_all\_together\_channel\_name

Depends On

Models

skazhi\_gord\_metrics  
vdud\_metrics  
apogovorit\_metrics

Code



... сам код dbt-модели, ...

← ↻ 🔒 cloud.getdbt.com dbt Docs

dbt

Search for models...

skazhi\_gord\_main  
vdud  
vdud\_main

Projects

- vdud
  - models
    - dimensions
      - all\_together**
    - intermediate
      - apogovorit\_metrics
      - skazhi\_gord\_metrics
      - vdud\_metrics
    - staging
      - stg\_models
        - apogovorit\_main\_numbers
        - apogovorit\_table
        - skazhi\_gord\_main\_numbers
        - skazhi\_gord\_table
        - vdud\_main\_numbers
        - vdud\_table
    - tests
      - final\_test
      - views\_likes\_com... on\_not\_negative

**all\_together** view

Details Description Columns Referenced By Depends On Code

Code

Source Compiled copy to clipboard

```
1 with
2   vdud_metrics as (select * from {{ ref("vdud_metrics") }}),
3   apogovorit_metrics as (select * from {{ ref("apogovorit_metrics") }}),
4   skazhi_gord_metrics as (select * from {{ ref("skazhi_gord_metrics") }}),
5   group_metrics as (
6
7     select *
8     from vdud_metrics
9
10    union all
11
12    select *
13    from apogovorit_metrics
14
15    union all
16
17    select *
18    from skazhi_gord_metrics
19  )
```

... и, наконец, моя любимая часть – граф – доступен при нажатии на кнопочку снизу

The screenshot shows the dbt Cloud interface. On the left is a sidebar with a file explorer showing a project structure: 'skazhi\_gord\_main', 'vdud', 'vdud\_main', 'Projects', 'vdud', 'models', 'dimensions', 'all\_together' (selected), 'intermediate', 'apogovorit\_metrics', 'skazhi\_gord\_metrics', 'vdud\_metrics', 'staging', 'stg\_models', 'apogovorit\_main\_numbers', 'apogovorit\_table', 'skazhi\_gord\_main\_numbers', 'skazhi\_gord\_table', 'vdud\_main\_numbers', 'vdud\_table', 'tests', 'final\_test', and 'views\_likes\_com... on\_not\_negative'. The main area displays the 'all\_together' view with tabs for 'Details', 'Description', 'Columns', 'Referenced By', 'Depends On', and 'Code'. The 'Code' tab is active, showing a SQL query. At the bottom right, there is a blue button labeled 'View Lineage Graph' with a downward arrow pointing to it from the text above. Below the button is a circular icon with a graph symbol.

cloud.getdbt.com dbt Docs

Search for models...

dbt

skazhi\_gord\_main  
vdud  
vdud\_main

Projects

- vdud
  - models
    - dimensions
      - all\_together**
    - intermediate
      - apogovorit\_metrics
      - skazhi\_gord\_metrics
      - vdud\_metrics
    - staging
      - stg\_models
        - apogovorit\_main\_numbers
        - apogovorit\_table
        - skazhi\_gord\_main\_numbers
        - skazhi\_gord\_table
        - vdud\_main\_numbers
        - vdud\_table
    - tests
      - final\_test
      - views\_likes\_com... on\_not\_negative

**all\_together** view

Details Description Columns Referenced By Depends On Code

```
3
4   apogovorit_metrics as (select * from {{ ref("apogovorit_metrics") }}),
5   skazhi_gord_metrics as (select * from {{ ref("skazhi_gord_metrics") }}),
6
7   group_metrics as (
8       select *
9       from vdud_metrics
10
11      union all
12
13      select *
14      from apogovorit_metrics
15
16      union all
17
18      select *
19      from skazhi_gord_metrics
20  )
21
22  select *
23  from group_metrics
24  order by total_view_mln desc
25
```

View Lineage Graph

Граф показан в виде визуальной схемы зависимостей моделей (здесь видны соседние шаги для выбранной модели)

← ↺ ↻ 🔒 cloud.getdbt.com

dbt Docs

🌐 📖 🔍 🗂️ 📄 📡

dbt

skazhi\_gord\_main

vdud

vdud\_main

Projects

vdud

models

dimensions

all\_together

intermediate

apogovorit\_metrics

skazhi\_gord\_metrics

vdud\_metrics

staging

stg\_models

apogovorit\_main\_numbers

apogovorit\_table

skazhi\_gord\_main\_numbers

skazhi\_gord\_table

vdud\_main\_numbers

vdud\_table

tests

final\_test

views\_likes\_com... on\_not\_negative

Search for models...

all\_together view

Details Description Columns Referenced By Depends On Code

```
3      apogovorit_metrics as (select * from {{ ref("apogovorit_metrics") }}),
4
5      skazhi_gord_metrics as (select * from {{ ref("skazhi_gord_metrics") }}),
6
7      group_metrics as (
8
9          select *
10         from vdud_metrics
11
12         union all
13
14         select *
15         from apogovorit_metrics
16
17         union all
18
19         select *
20         from skazhi_gord_metrics
21     )
22
23     select *
24     from group_metrics
25     order by total_view_mln desc
```

Lineage Graph

View Fullscreen

skazhi\_gord\_metrics

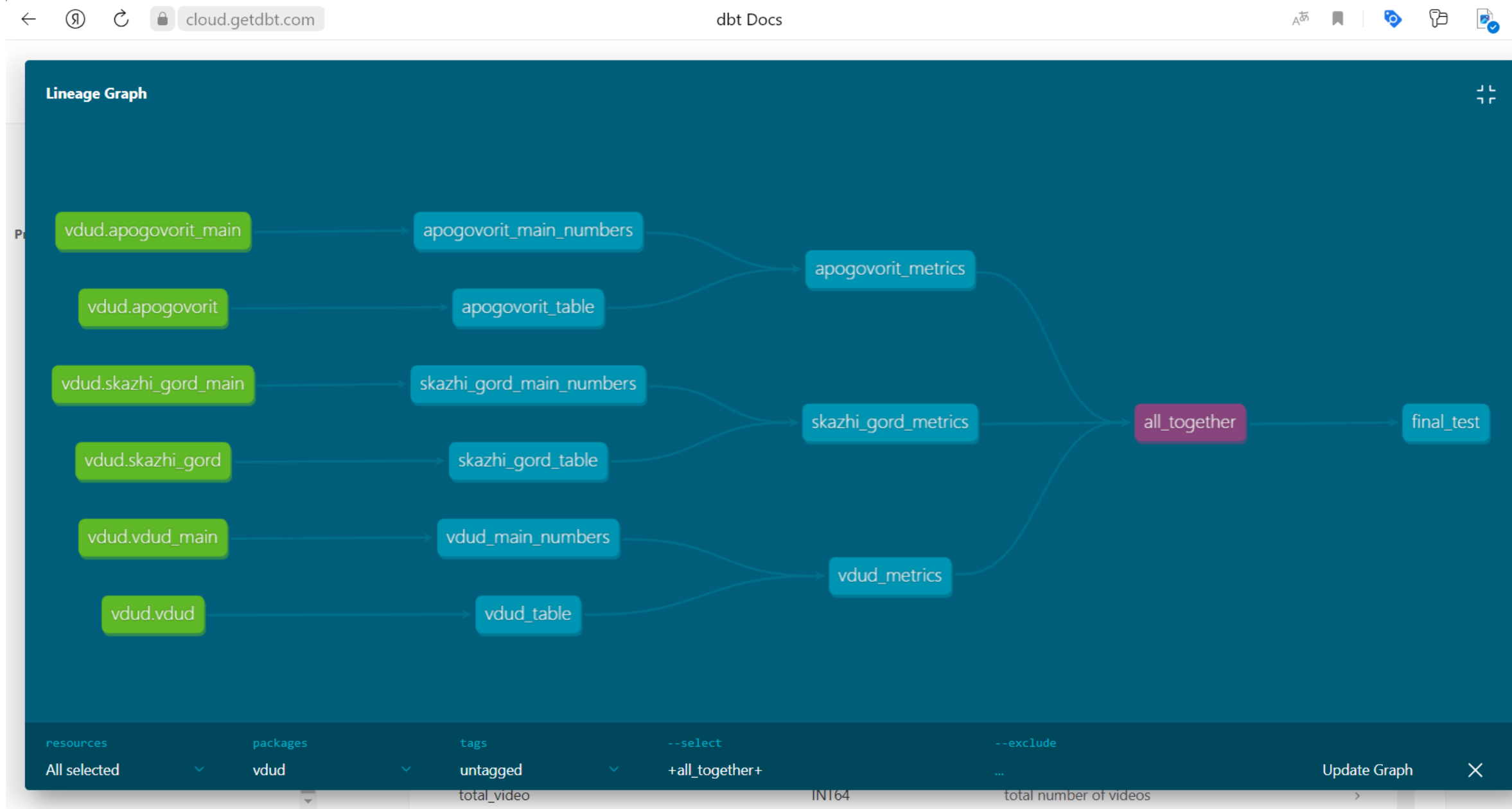
vdud\_metrics

apogovorit\_metrics

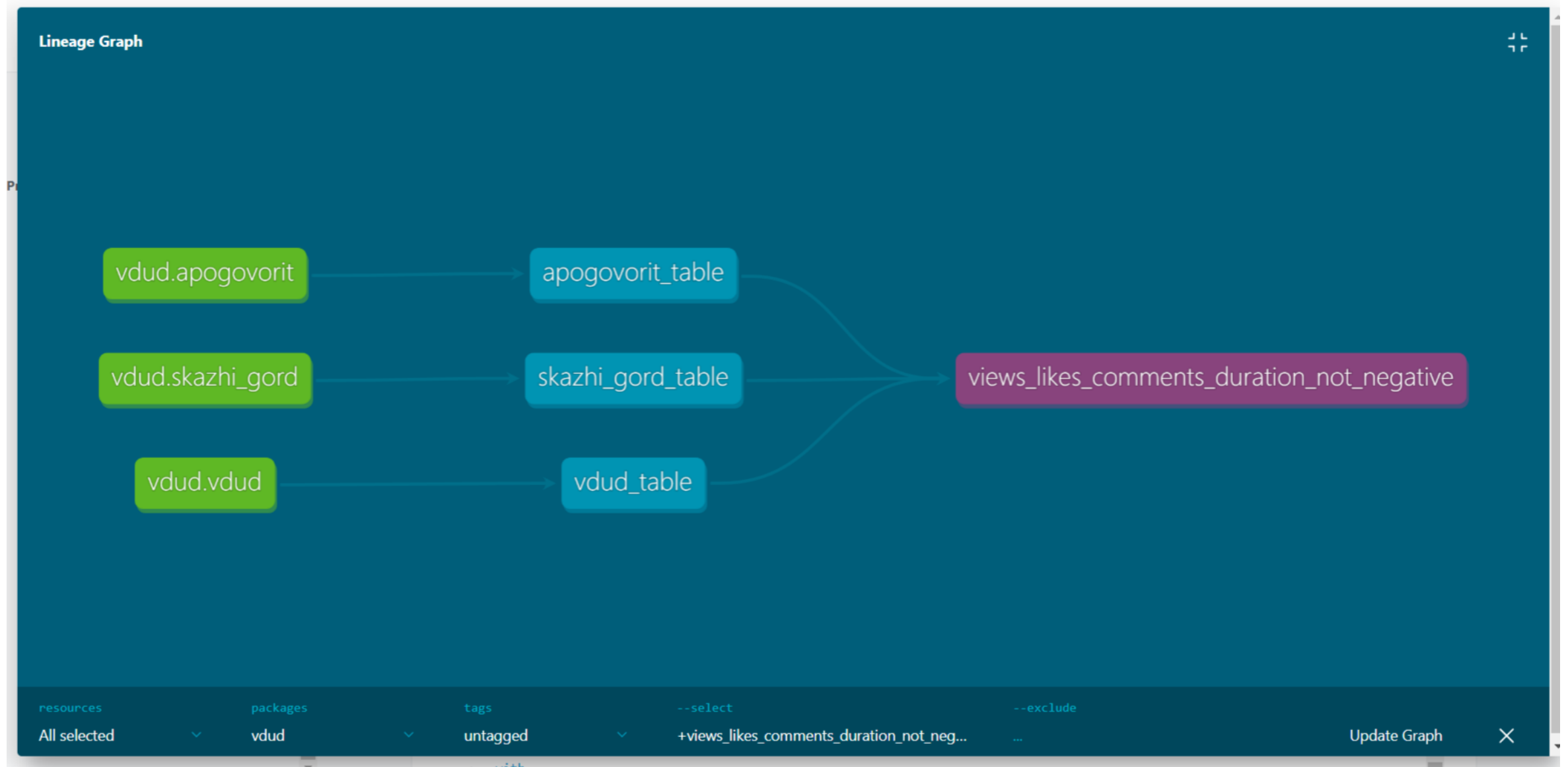
all\_together

final\_test

## При желании граф можно раскрыть целиком

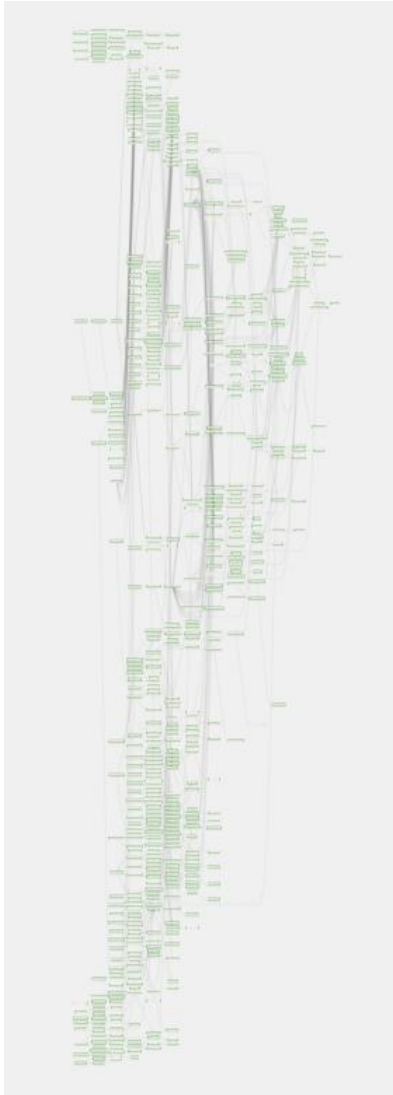


Кстати, графы доступны не только для моделей, но и для тестов  
(views\_likes\_comments\_duration\_not\_negative – это мой первый собственный тест)



Вот такой получился у меня проект.

Он находится по ссылке: <https://cloud.getdbt.com/develop/189308/projects/276942>



В заключение хотелось бы добавить, что dbt плотно используется у меня на работе. Доказательство тому – граф из Airflow с dbt\_dag на картинке слева. Простыми словами это схема взаимосвязей dbt-моделей проекта.

Так что в моей рабочей жизни dbt – уже далеко не fancy инструмент 😊

Больше интересного о моём пути в аналитике данных – в моём телеграм-канале «Дневник аналитика»: [https://t.me/diary\\_musician\\_analyst](https://t.me/diary_musician_analyst)

Welcome!