MoneyLion®

dbt_

Why it’s awesome and how we use it in MoneyLion

MoneyLion

WHO AM I

- Boon Keong (BK)
- Work for MoneyLion (US fintech company)
- A data engineer
- Build and maintain data pipeline architectures that enables analytic teams to work more efficiently
- Even if things gets messy 💩

MoneyLion

MOTIVATION

WHAT IS DBT

PROBLEM-SOLUTION

PRODUCTIONISATION

ANALYTICS TRANSFORMED

MoneyLion

TYPICAL ANALYTICS

- Messy (copy paste) SQL code

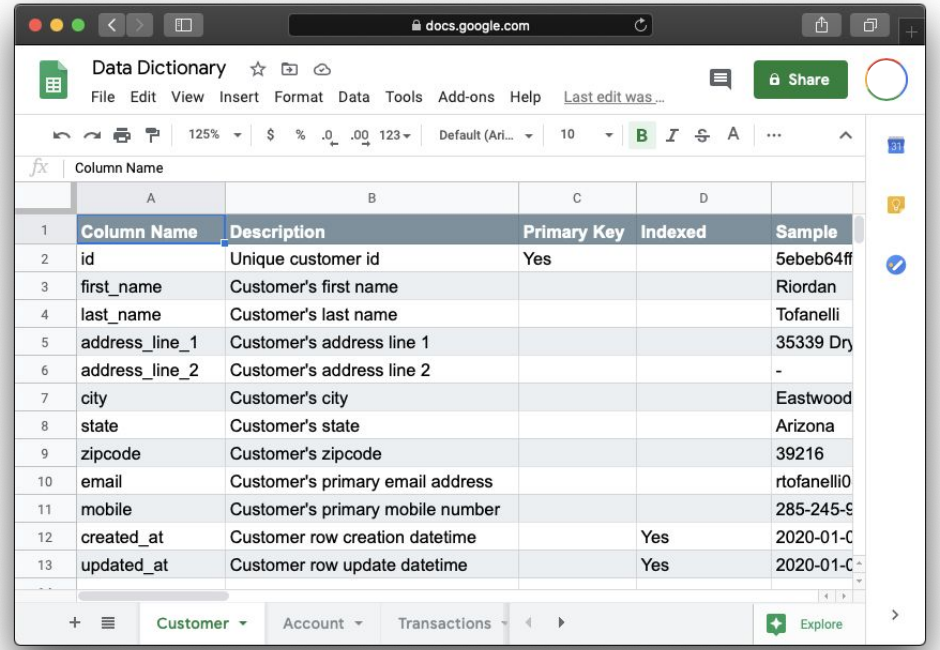
bunch-of-sql.sql

```
--no_cache
with sub_base AS
(select distinct userid
from schema.user_base
where lower(subtype) like ('%employee%') or userid = '1231231231

,t1 AS
(select
id as transaction_id
,user_id as tx_uid
,created_at as transaction_created_at
,amount as transaction_amount
,(txse when transmit_at is null then 'A_TIER' --
      when tier_name != '' then tier_name --
      when amount > 2 then 'B_TIER' --
      else tier_name end) as transaction_tier_name --
,failure_reason as transaction_failure_reason
,transmit_at as transaction_transmit_at
from schema.transactions)

,t2 AS
(select
transaction_id as txs_transaction_id
,txs.amount as txs_amount
,txs.created_at as txs_created_at
,txs.receival_id as txs_receival_id
```

MoneyLion

TYPICAL ANALYTICS

- Messy (copy paste) SQL code
- Little or no documentation

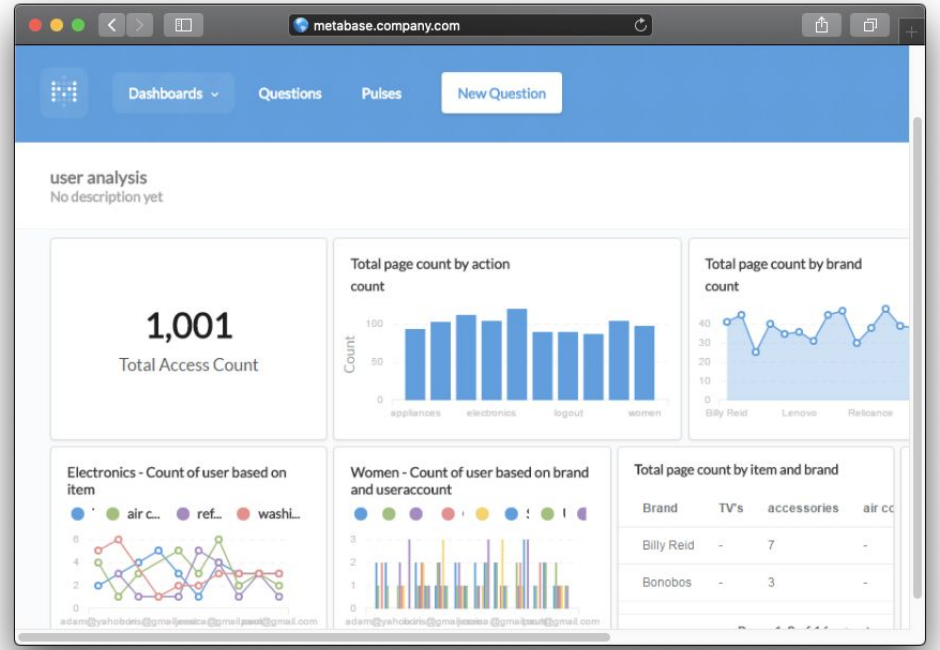


	A	B	C	D	
1	Column Name	Description	Primary Key	Indexed	Sample
2	id	Unique customer id	Yes		5ebeb64ff
3	first_name	Customer's first name			Riordan
4	last_name	Customer's last name			Tofanelli
5	address_line_1	Customer's address line 1			35339 Dry
6	address_line_2	Customer's address line 2			-
7	city	Customer's city			Eastwood
8	state	Customer's state			Arizona
9	zipcode	Customer's zipcode			39216
10	email	Customer's primary email address			rtofanelli0
11	mobile	Customer's primary mobile number			285-245-9
12	created_at	Customer row creation datetime		Yes	2020-01-0
13	updated_at	Customer row update datetime		Yes	2020-01-0

MoneyLion

TYPICAL ANALYTICS

- Messy (copy paste) SQL code
- Little or no documentation
- But simple BI requirements



MoneyLion

PAIN POINTS

- Disorganised & difficult to collaborate

MoneyLion

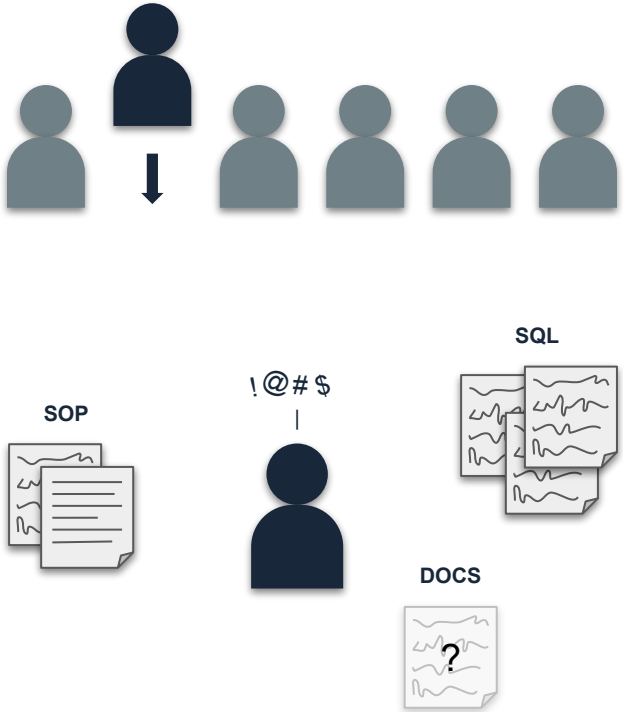
PAIN POINTS

- Disorganised & difficult to collaborate
- Hard to visualise links between tables

MoneyLion

PAIN POINTS

- Disorganised & difficult to collaborate
- Hard to visualise links between tables
- Analysis on-boarding / handover is a pain



MoneyLion

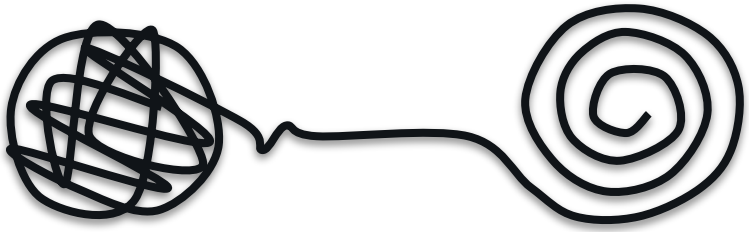
WE WANT

- A way to organised code base for collaboration

MoneyLion

WE WANT

- A way to organised code base for collaboration
- A way to break down complex SQL code for better comprehension



MoneyLion

WE WANT

- A way to organised code base for collaboration
- A way to break down complex SQL code for better comprehension
- A simple and easy way to document tables for handovers or reference

MoneyLion

WE WANT

- A way to organised code base for collaboration
- A way to break down complex SQL code for better comprehension
- A simple and easy way to document tables for handovers or reference

dbt_

MoneyLion

MOTIVATION

WHAT IS DBT

PROBLEM-SOLUTION

PRODUCTIONISATION

ANALYTICS TRANSFORMED

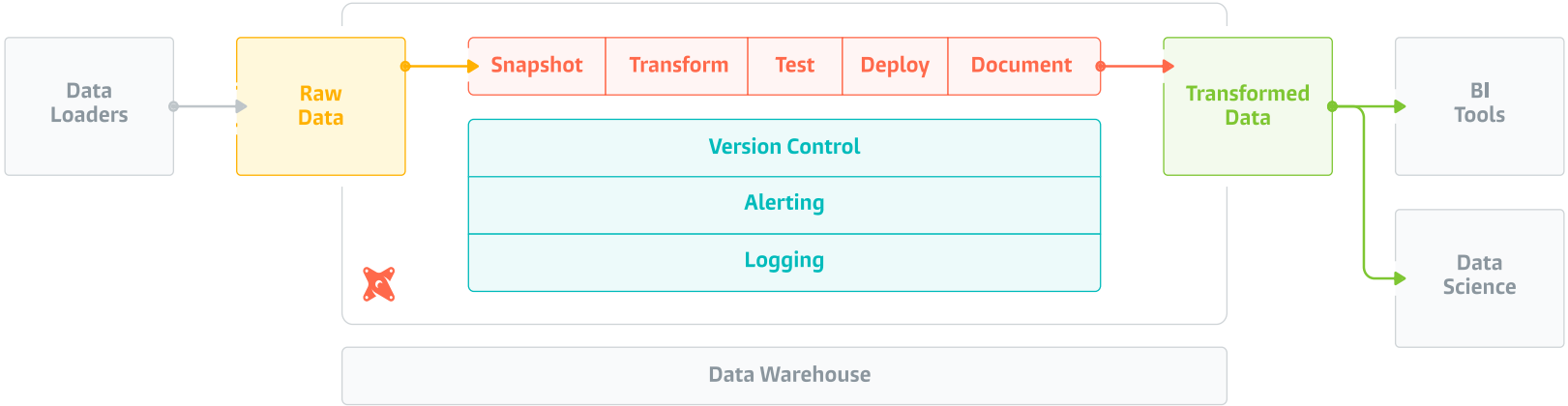
MoneyLion

WHAT IS DBT (data build tool)

- *“a toolkit for building analytics the way developers build applications”*

  —data engineering podcast, episode 81

- Handles the “**T**” in “EL**T**” (Extract-Load-**Transform**)



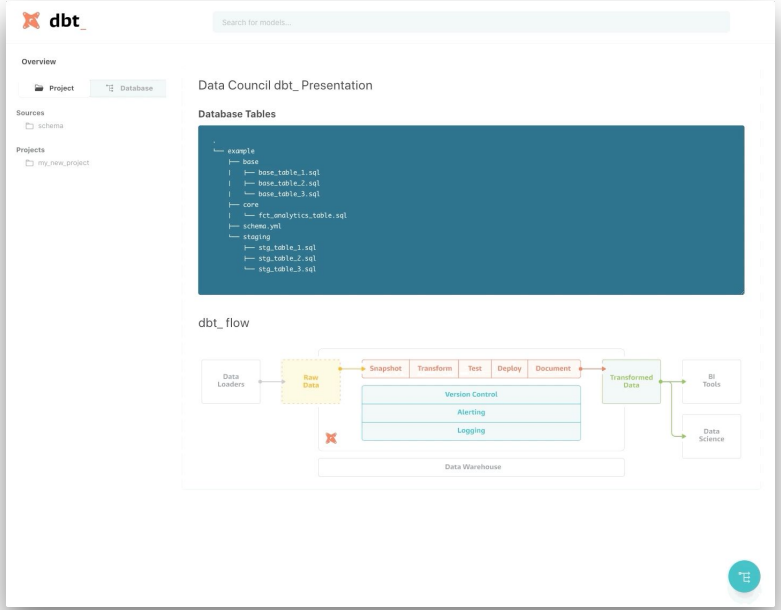
MoneyLion

WHY IT'S AWESOME

BEFORE

```sql
WITH employee AS
(
  SELECT DISTINCT id
  FROM (SELECT id
        FROM schema.subs
        WHERE LOWER(subtype) LIKE ('%employee%')
        OR    id = '123123123123'
        UNION
        SELECT id
        FROM schema.users
        WHERE email LIKE '%@company.com')
),
req_final_cte AS
(
  SELECT id                        AS req_id,
         user_id                   AS user_id,
         created_at                AS laon_reqlied_at,
         effective_date            AS laon_effective_at,
         disbursed_at              AS laon_disbursed_at,
         "type"                    AS release_type,
         status                    AS high_level_status,
         failure_reason            AS req_failure_reason,
         receival_at               AS laon_receival_at,
         txsE
           WHEN laon_receival_at < CURRENT_DATE THEN TRUE
           ELSE FALSE
         END AS laon_due,
         original_maximum_eligible AS original_maximum_eligible,
         max_amount                AS laon_max_amount,
         min_amount                AS laon_min_amount,
         available_amount          AS laon_available_amount,
         txsE
           WHEN receival_at IS NULL AND laon_amount = 2 THEN 'A_TIER'
           WHEN tier_name = '' AND laon_amount > 2 THEN 'B_TIER'
           ELSE tier_name
         END AS laon_tier_name,
         laon_amount       AS laon_amount_on_req
  FROM schema.tx_req
),
tx_cdd_cte AS
(
  SELECT tx.id        AS laon_id,
```

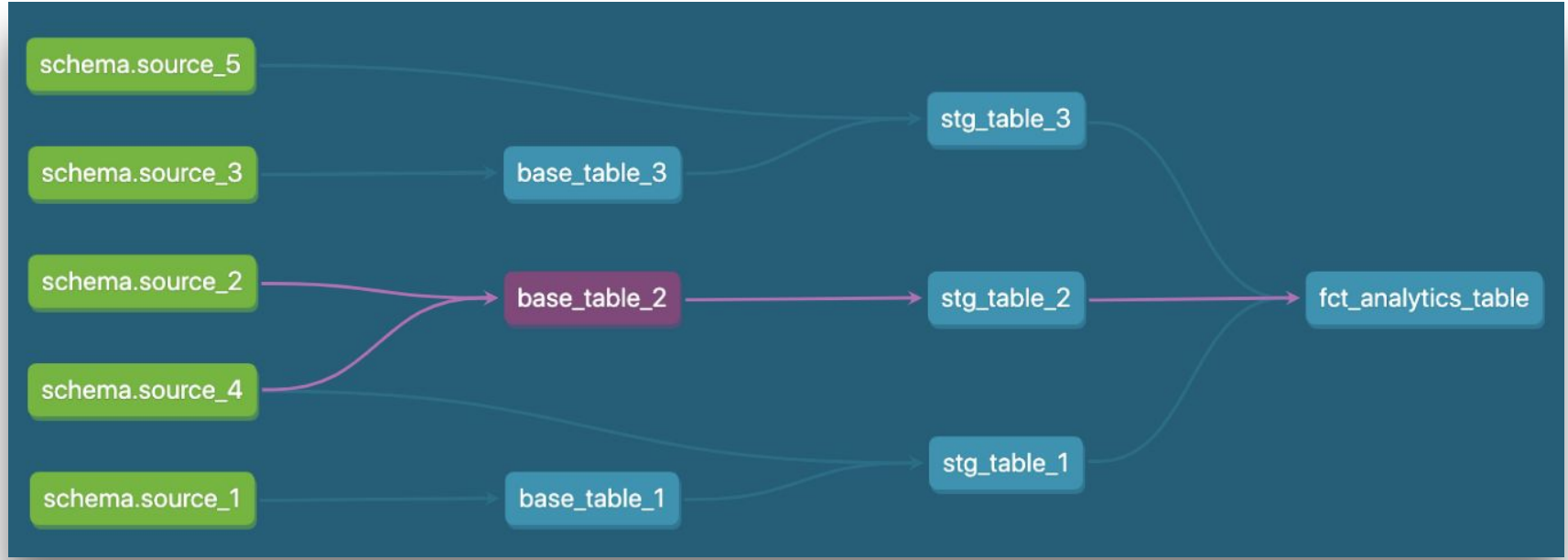
AFTER



MoneyLion

WHY IT'S AWESOME

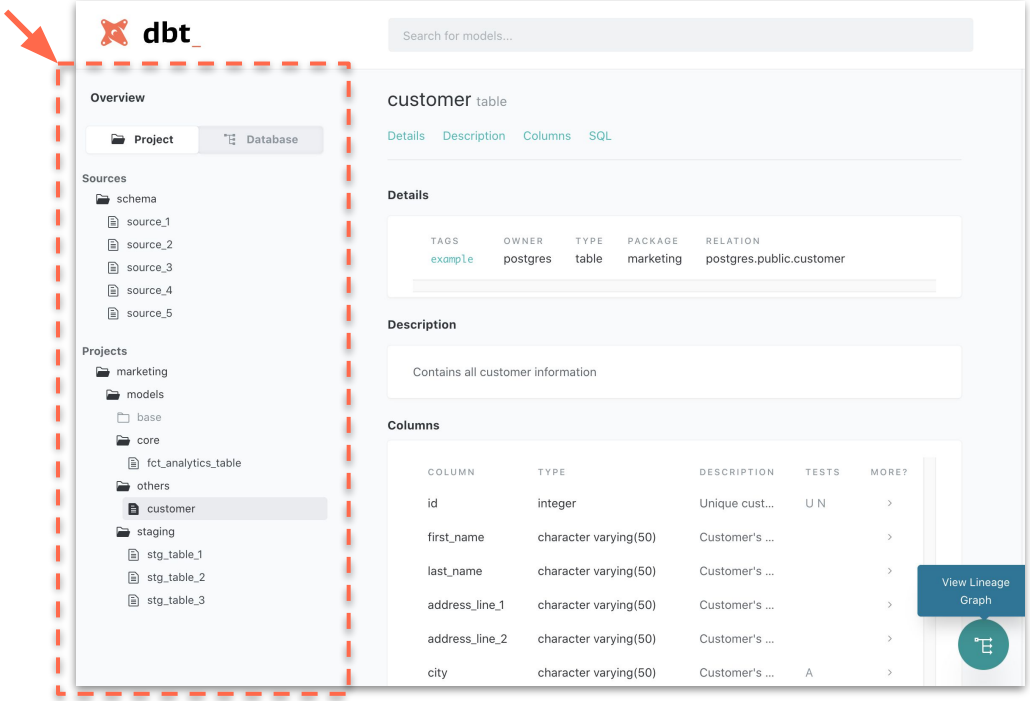
DATA LINEAGE VISUALISATION



MoneyLion

WHY IT'S AWESOME

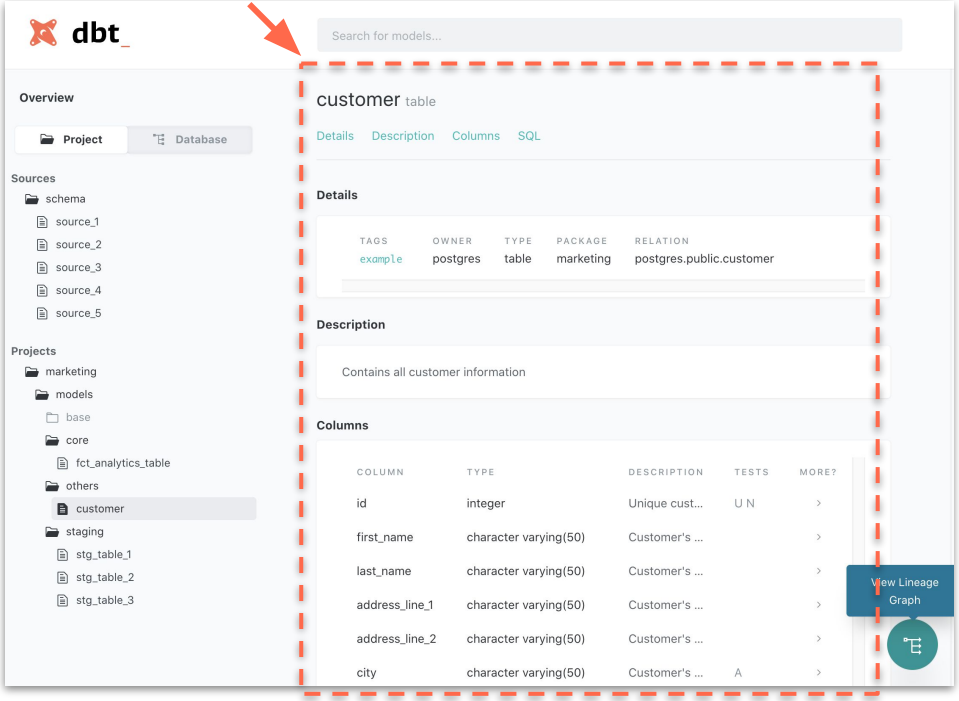
DATA MODELS



MoneyLion

WHY IT'S AWESOME

DATA DICTIONARY



MoneyLion

MOTIVATION

WHAT IS DBT

PROBLEM-SOLUTION

PRODUCTIONISATION

ANALYTICS TRANSFORMED

MoneyLion

#1 - GET ORGANIZED

PROBLEM

- Disorganised SQL code
  - Stored in text files ✘
  - Send via email/messengers ✘
  - Difficult to collaborate ✘

SOLUTION

- Use dbt with GitHub
  - Transparency ✔
  - Version controlled ✔
  - Easier collaboration ✔

MoneyLion

GET ORGANIZED

MoneyLion / dbt Private

Code | Pull requests 0 | Actions | Projects 1 | Security 0 | Insights | Settings

The main analytics repository for dbt projects. Edit

Manage topics

392 commits | 1 branch | 0 packages | 0 releases | 12 contributors

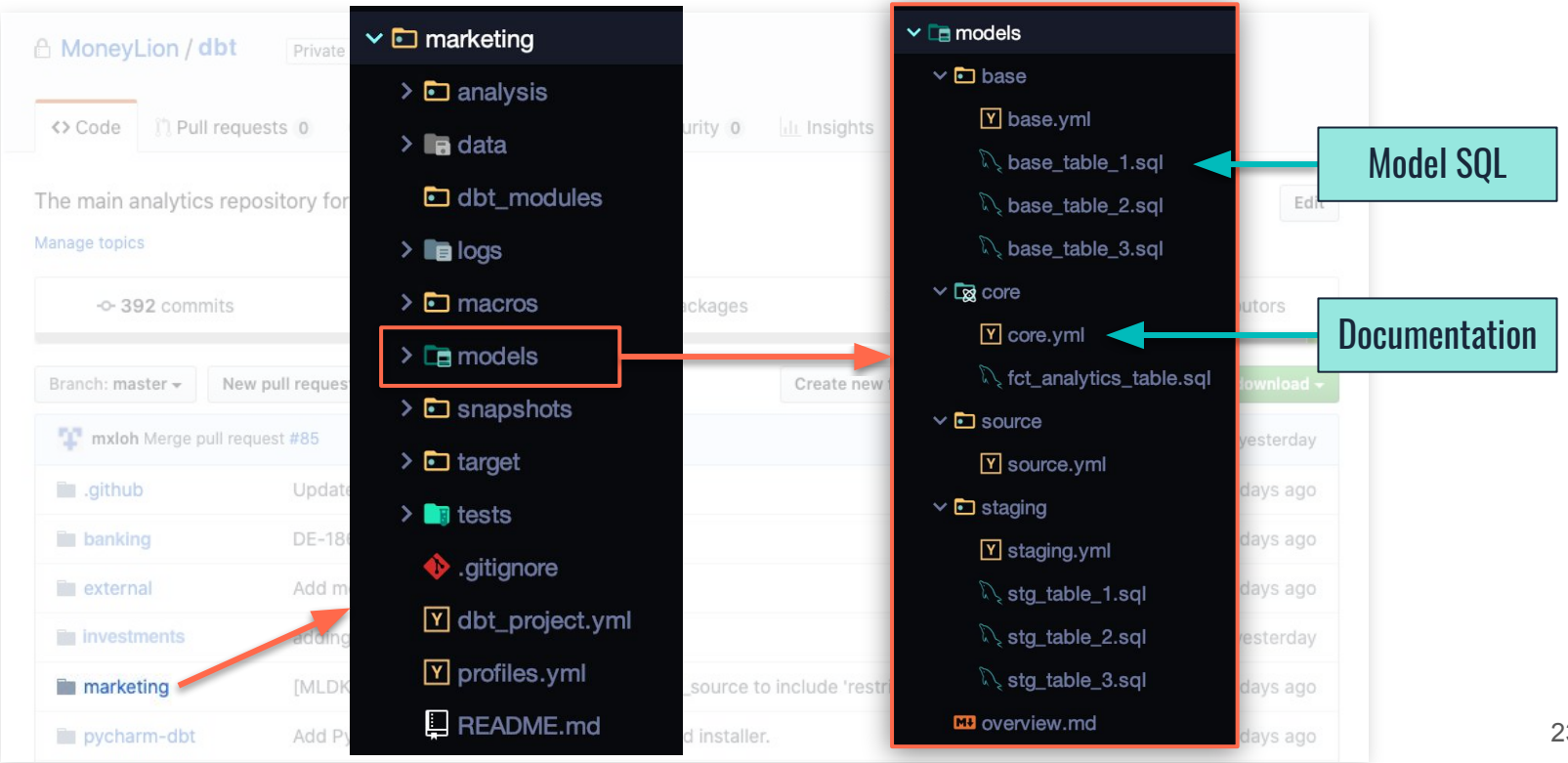
Branch: master | New pull request | Create new file | Upload files | Find file | Clone or download

mxloh Merge pull request #85 ... Latest commit bd8b02a yesterday

.github	Update pull_request_template.md	20 days ago
banking	DE-1865 Lowercased stg_prejoined_b	5 days ago
external	Add more tables	7 days ago
investments	adding a test user in	yesterday
marketing	[MLDK01-18780] - 20200506 - Added media_source to include 'restricted...	13 days ago
pycharm-dbt	Add PyCharm Custom Live Templates files and installer.	14 days ago

MoneyLion

GET ORGANIZED



MoneyLion

#2 - BREAKING IT DOWN

PROBLEM

- SQL too long and complex
  - Hard to understand/visualise ✘
  - Slow/Hard to debug ✘
  - Many repeated CTEs/logic ✘

SOLUTION

- Use **Data Models** concept
  - Reduced cognitive load ✔
  - Easy to debug ✔
  - Reusable code ✔
- Use **Macros** to keep DRY ✔

CTE = Common Table Expression
DRY = Don't Repeat Yourself

MoneyLion

DATA MODELS

BEFORE

```
with cte_1 as (
    select …
    from src_A
),

cte_2 as (
    select …
    from src_B
    where …
)

select * from cte_1
left join cte_2
```

AFTER

```
select …
from src_A
```

cte_1.sql

```
select …
from src_B
where …
```

cte_2.sql

Jinja Templating Language

```
select *
from
  {{ ref('cte_1') }}
left join
  {{ ref('cte_2') }}
```

fct_joined.sql

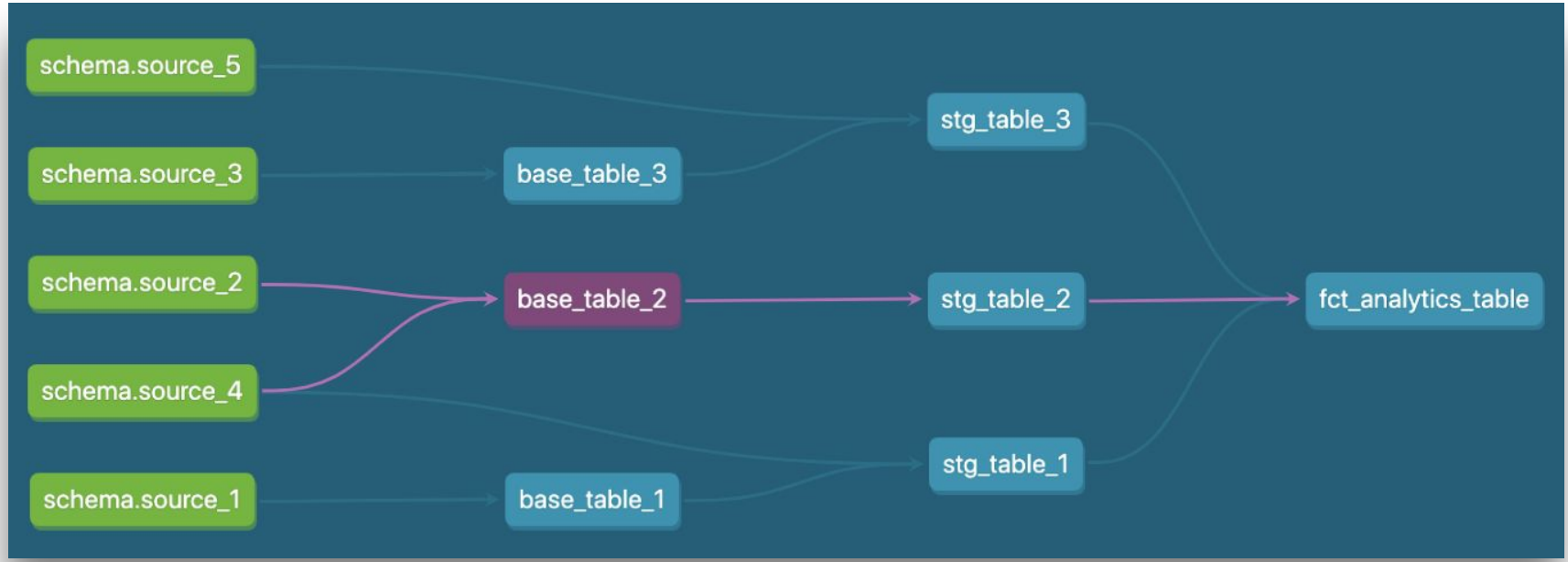
```
select id, name
from {{ref('cte_2')}}
where …
```

Reusable code!

MoneyLion

DATA MODELS

SOURCE	BASE	STAGING	FACT
-	ephemeral	view *(no-binding)*	table



MoneyLion

MACROS

BEFORE

```
select
  case
    when A then 1
    when B then 2
    when C then 3
    when D then 4
    else null
  end as position,
  … … …
from alphabet
group by 1, 2, 3
```

AFTER

Jinja Templating Language

```
{% macro pos(col) %}
  case
    when A then 1
    when B then 2
    when C then 3
    when D then 4
    else null
  end as position
{% endmacro %}
```

macros.sql

```
select
  {{ pos(alphabet) }},
  … … …
from alphabet
{{ group_by(3) }}
```

stg_alphabet.sql

```
select
  {{ pos(alphabet) }},
  … … …
from another_table
```

Reusable code!

MoneyLion

MACROS

BEFORE

```
select
  case
    when alphabet_1 = A then 1
    when alphabet_1 = B then 2
    when alphabet_1 = C then 3
    when alphabet_1 = D then 4
    else null
  end as position_1,
  case
    when alphabet_2 = A then 1
    when alphabet_2 = B then 2
    when alphabet_2 = C then 3
    when alphabet_2 = D then 4
    else null
  end as position_2,
  case
    when alphabet_3 = A then 1
    when alphabet_3 = B then 2
    when alphabet_3 = C then 3
    when alphabet_3 = D then 4
    else null
  end as position_3,
  … … …
from alphabet
group by 1, 2, 3
```

AFTER

```
select
  {{ pos(aphabet_1} }} as position_1,
  {{ pos(aphabet_2} }} as position_2,
  {{ pos(aphabet_2} }} as position_3,
  … … …
from alphabet
{{ dbt_utils.group_by(3) }}
```

MoneyLion

#3 - DOCUMENT

PROBLEM

- Lack of Documentation
  - On-boarding/handover pains ✘
  - Management questions ✘
  - Headache when revisiting ✘

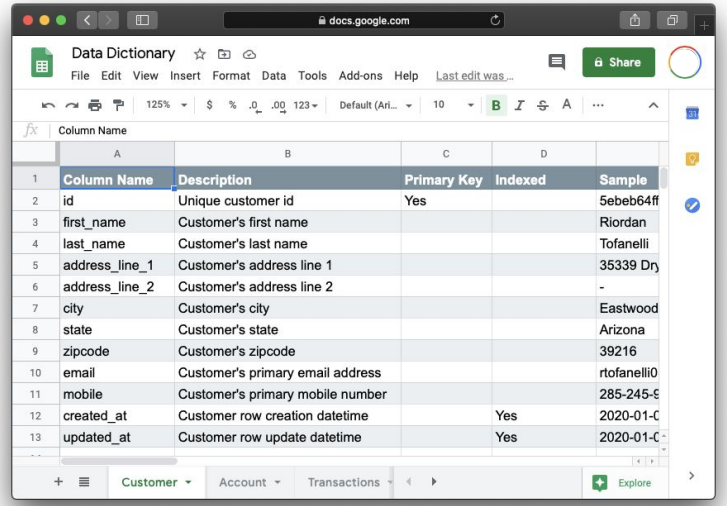
SOLUTION

- Use dbt docs
  - Nice & Sleek UI ✔
  - Visible Lineage ✔
  - Unit Testing & Integrity Checks ✔

MoneyLion

DOCUMENTATION

BEFORE



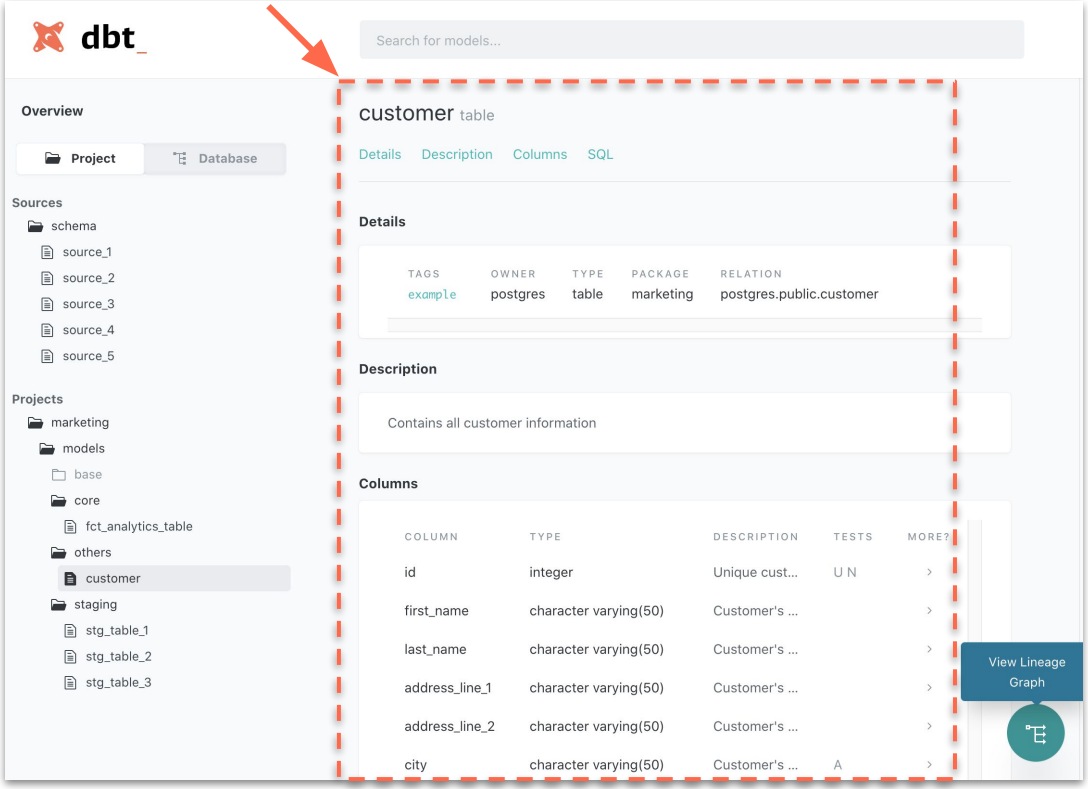
Column Name	Description	Primary Key	Indexed	Sample
id	Unique customer id	Yes		5ebeb64ff
first_name	Customer's first name			Riordan
last_name	Customer's last name			Tofanelli
address_line_1	Customer's address line 1			35339 Dr
address_line_2	Customer's address line 2			-
city	Customer's city			Eastwood
state	Customer's state			Arizona
zipcode	Customer's zipcode			39216
email	Customer's primary email address			rtofanelli0
mobile	Customer's primary mobile number			285-245-9
created_at	Customer row creation datetime		Yes	2020-01-0
updated_at	Customer row update datetime		Yes	2020-01-0

AFTER (in YAML)

```yaml
models:
  - name: customer
    description: "Contains all customer information"

    columns:
      - name: id
        description: "Unique customer id"
        tests: <2 items>

      - name: first_name
        description: "Customer's first name"

      - name: last_name
        description: "Customer's last name"
```

MoneyLion

DATA DICTIONARY



MoneyLion

DATA TESTING & CHECKS

DEFINE TESTS (in YAML)

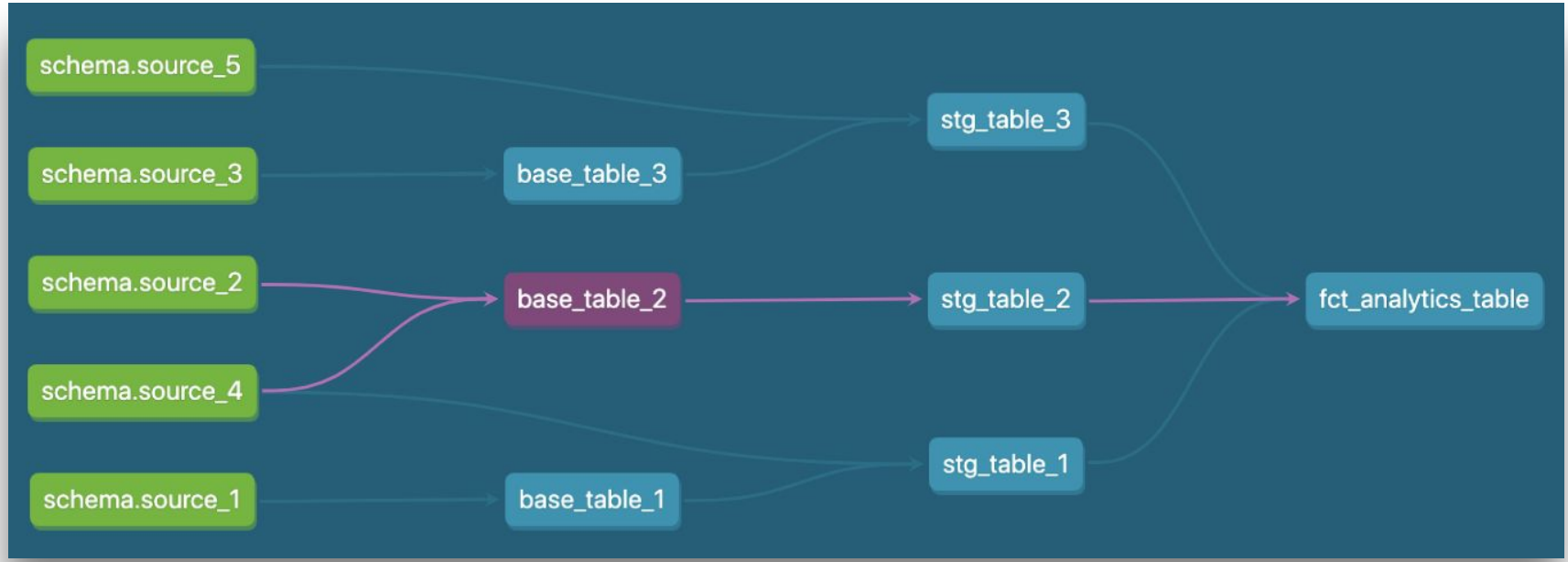
```yaml
columns:
  - name: id
    description: "Unique customer id"
    tests:
      - unique
      - not_null
  ...

  - name: city
    description: "Customer's city"
    tests:
      - accepted_values:
          values:
            - 'Kuala Lumpur'
            - 'Kuching'
            - 'Johor Bahru'
            - 'Melaka'
            - 'Ipoh'
```

RUN TESTS

```bash
$ dbt test
Concurrency: 8 threads (target='dev')

1 of 3 START test accepted_values_customer_city__Kuala_Lumpur__Kuch... [RUN]
2 of 3 START test not_null_customer_id............................... [RUN]
3 of 3 START test unique_customer_id................................. [RUN]
1 of 3 PASS accepted_values_customer_city__Kuala_Lumpur__Kuch... [PASS in 0.09s]
2 of 3 PASS not_null_customer_id..............................  [PASS in 0.09s]
3 of 3 PASS unique_customer_id................................  [PASS in 0.09s]

Finished running 3 tests in 0.77s.

Completed successfully
```

MoneyLion

DATA LINEAGE VISUALISATION



MoneyLion

MOTIVATION

WHAT IS DBT

PROBLEM-SOLUTION

PRODUCTIONISATION

ANALYTICS TRANSFORMED

MoneyLion

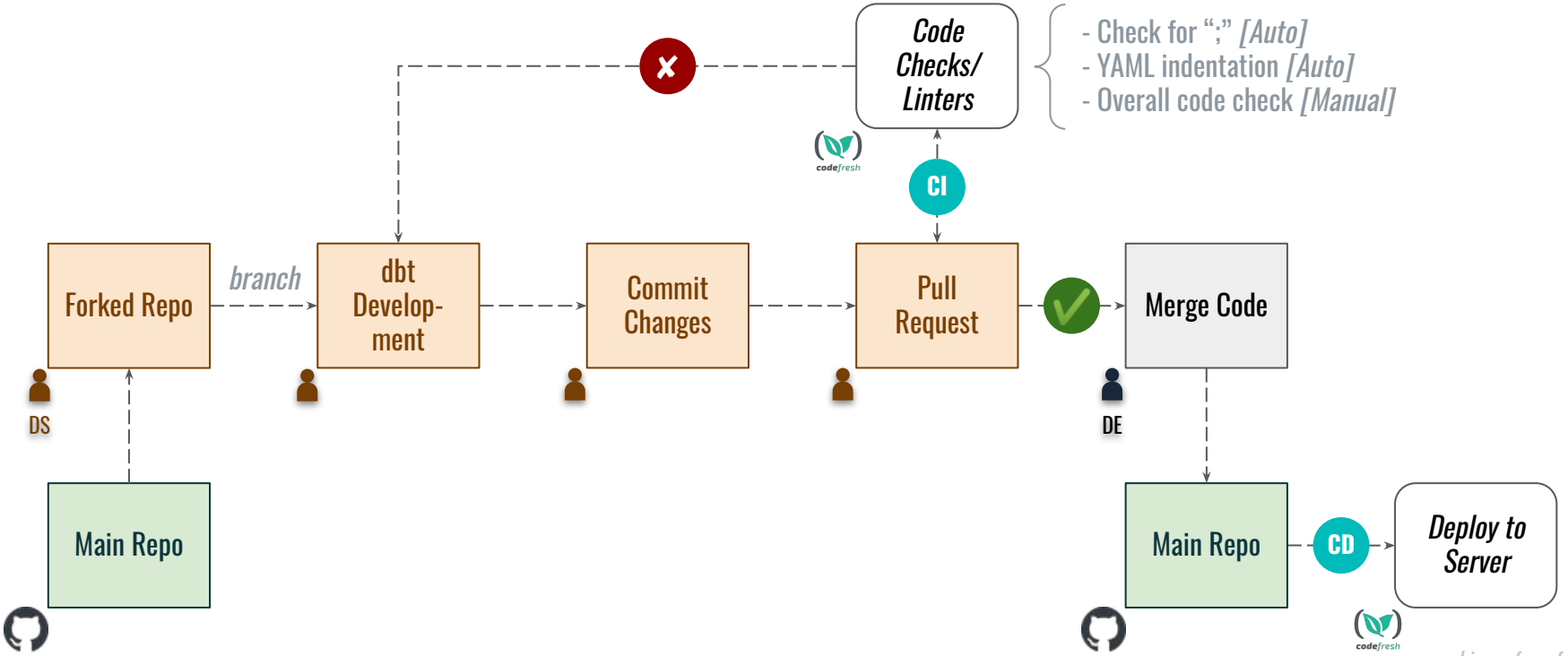
PRODUCTIONISATION

ASK

- How to get people to ***contribute***?
- How do we ***automate the "T"*** pipeline?
- Where do I put the ***documentation*** site?

MoneyLion

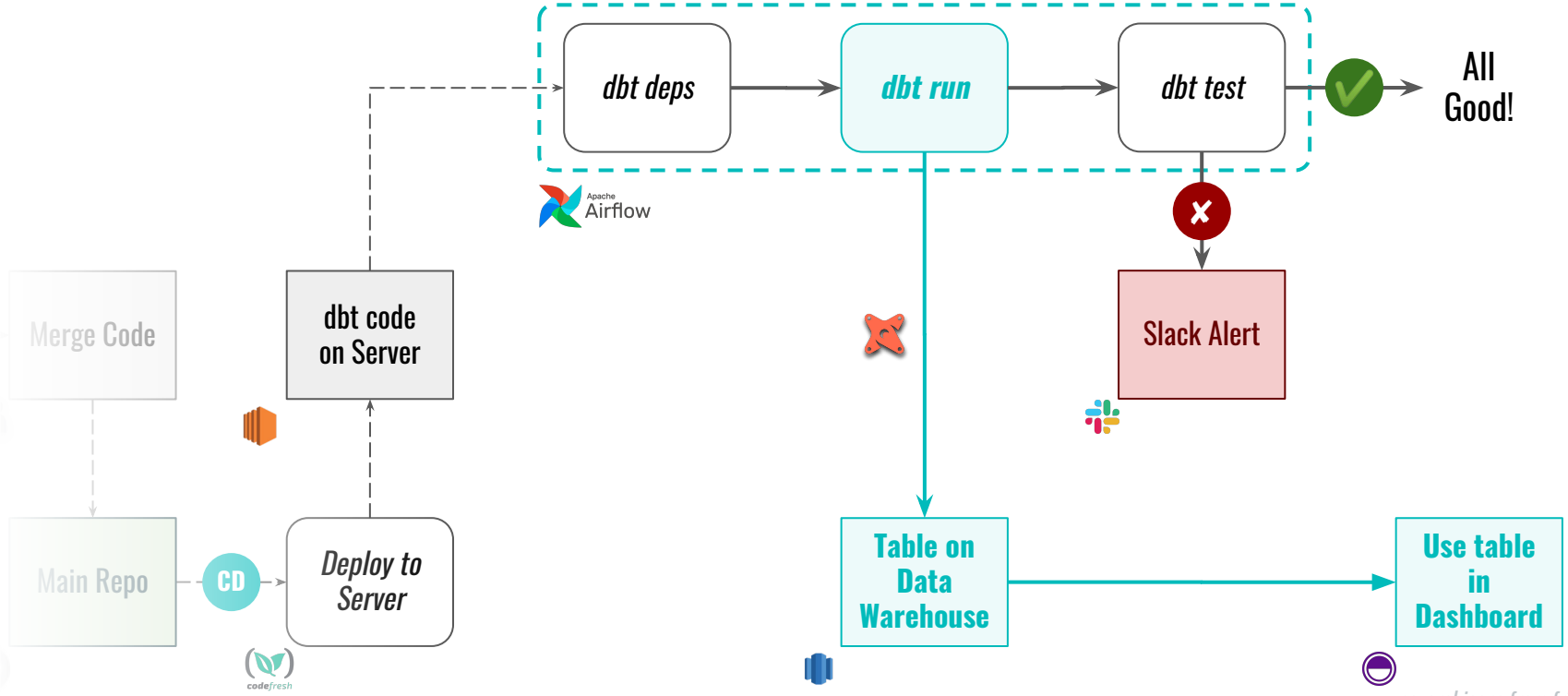
DBT DEVELOPMENT & DEPLOYMENT FLOW



several icons from friconix.com

MoneyLion

DBT EXECUTION FLOW



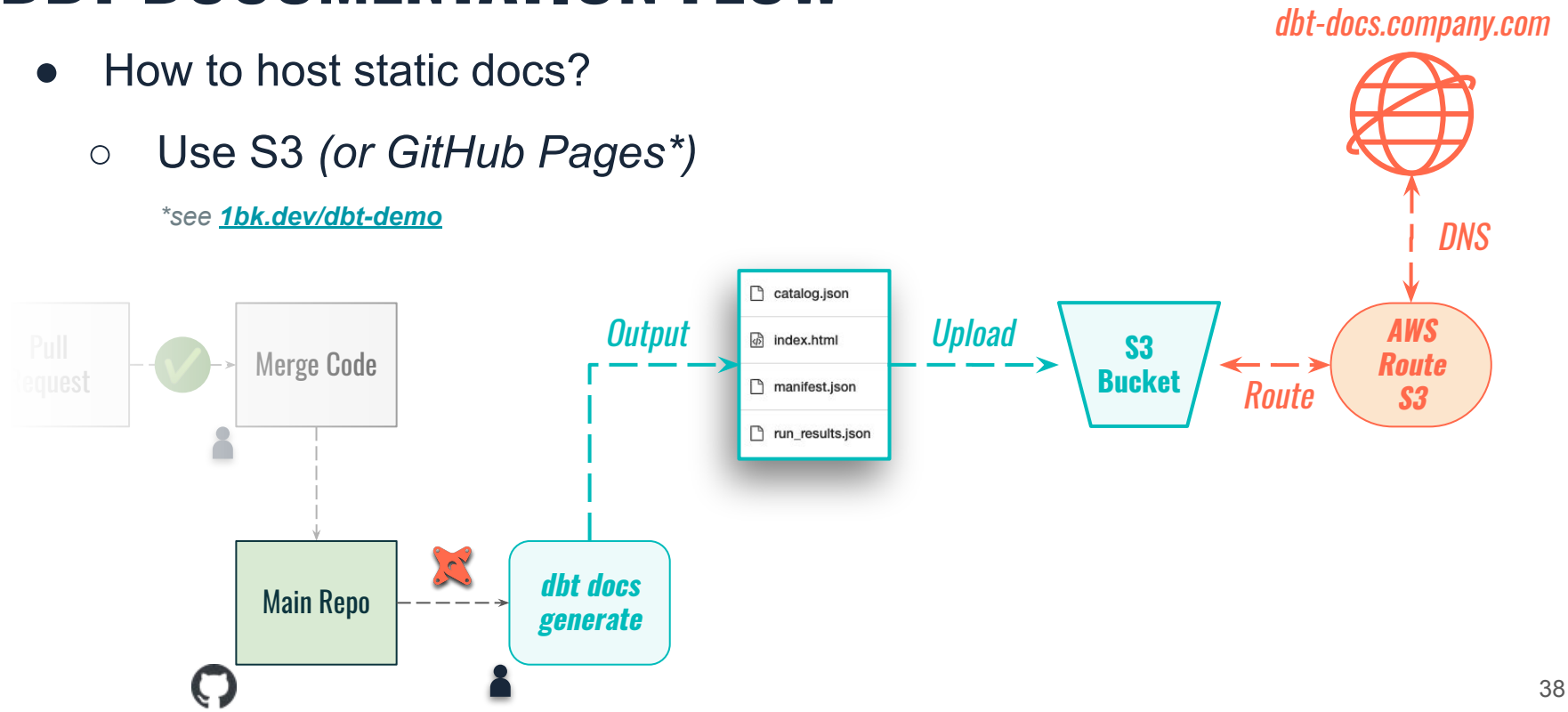
several icons from friconix.com

MoneyLion

DBT DOCUMENTATION FLOW

- How to host static docs?
  - Use S3 *(or GitHub Pages*)*

*see 1bk.dev/dbt-demo



MoneyLion

ATTEMPTING THE CHANGE

- It’s time to get everyone else on-board!
- First, **presentation** to the company.
- Expect low initial buy-in, so invest more time…
- Host multiple hands-on **tutorial sessions**
- Set up Slack channels for support and sharing

MoneyLion

MOTIVATION
WHAT IS DBT
PROBLEM-SOLUTION
PRODUCTIONISATION
ANALYTICS TRANSFORMED

MoneyLion

ANALYTICS TRANSFORMED

BEFORE

- DS pass SQL to DE to run
- Disorganised code
- Messy and complex SQL
- Repetitive code
- Little or no documentation
- Lack proper framework

AFTER

- More shared ownership
- Git integration & version control
- Clear model structure & lineage
- DRY code with Models & Macros
- Sleek documentation site
- Clear analytic workflow

MoneyLion

BUT But but...

MoneyLion

BUT But but...

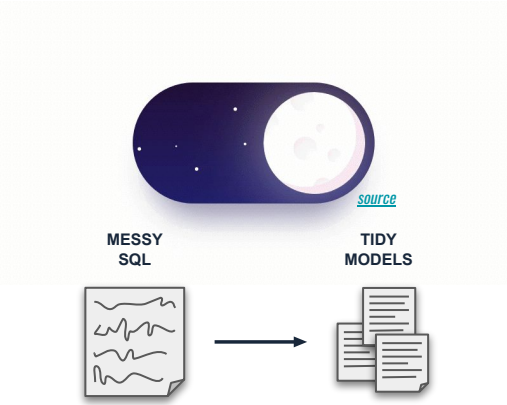
- Busy with BAU
- Time consuming to migrate
- Steep learning curve

CLI?

Git?

Version Control?

YAML?



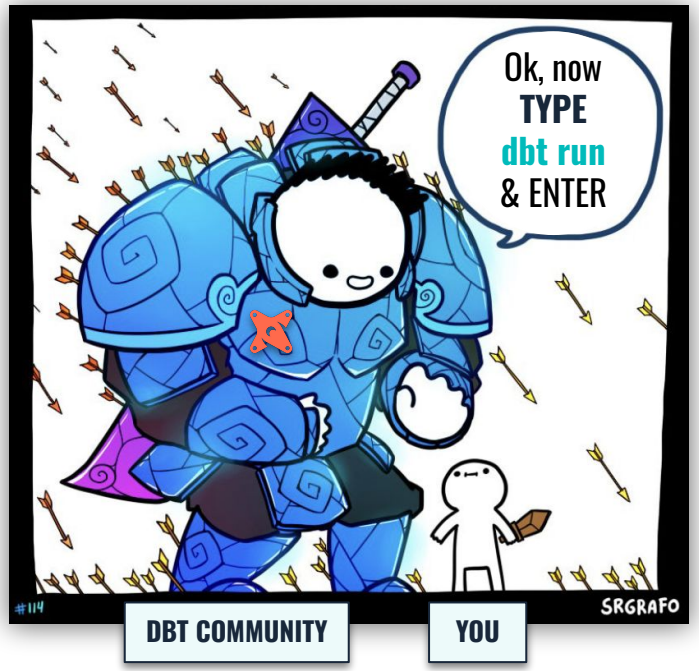
MoneyLion

BUT But but...

- Busy with BAU
- Time consuming to migrate
- Steep learning curve

- Yes, the journey can be though

  … but not as bad as you think!



MoneyLion

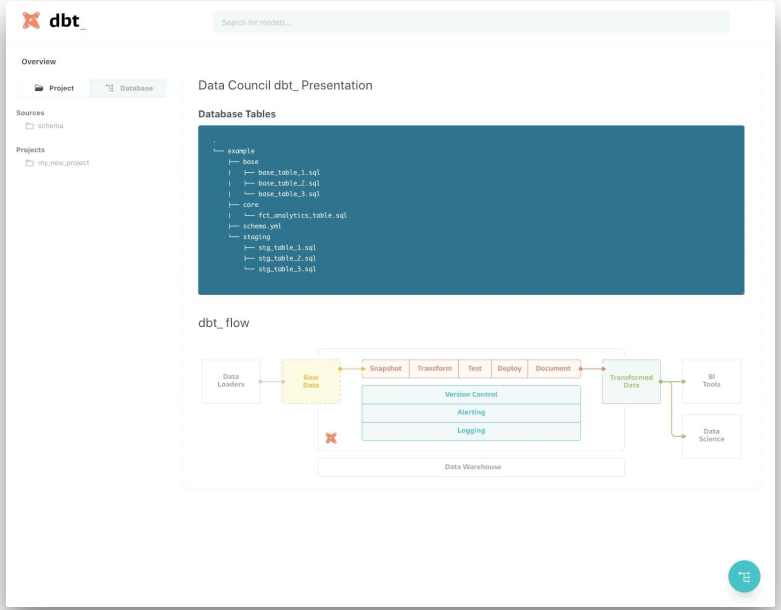
MIGRATED IN < 1 DAY

MONEYLION'S FIRST STEPS

BEFORE

```sql
WITH employee AS
(
  SELECT DISTINCT id
  FROM (SELECT id
        FROM schema.subs
        WHERE LOWER(subtype) LIKE ('%employee%')
        OR    id = '123123123123'
        UNION
        SELECT id
        FROM schema.users
        WHERE email LIKE '%@company.com')
),
req_final_cte AS
(
  SELECT id                        AS req_id,
         user_id                   AS user_id,
         created_at                AS laon_reqlied_at,
         effective_date            AS laon_effective_at,
         disbursed_at              AS laon_disbursed_at,
         "type"                    AS release_type,
         status                    AS high_level_status,
         failure_reason            AS req_failure_reason,
         receival_at               AS laon_receival_at,
         txsE
           WHEN laon_receival_at < CURRENT_DATE THEN TRUE
           ELSE FALSE
         END AS laon_due,
         original_maximum_eligible  AS original_maximum_eligible,
         max_amount                 AS laon_max_amount,
         min_amount                 AS laon_min_amount,
         available_amount           AS laon_available_amount,
         txsE
           WHEN receival_at IS NULL AND laon_amount = 2 THEN 'A_TIER'
           WHEN tier_name = '' AND laon_amount > 2 THEN 'B_TIER'
           ELSE tier_name
         END AS laon_tier_name,
         laon_amount        AS laon_amount_on_req
  FROM schema.tx_req
),
tx_cdd_cte AS
(
  SELECT tx.id          AS laon_id,
```

AFTER



MoneyLion

YOUR FIRST STEPS

- Git clone the Official Tutorial:

  https://github.com/fishtown-analytics/jaffle_shop
- What you need:
  - **Python 3.6 +**
  - A database:
    - **Postgres** *(docker: https://hub.docker.com/_/postgres)*
    - Or others...

      See https://docs.getdbt.com/docs/supported-databases/

MoneyLion

SETUP

Install dbt

Setup and Test DB Connection

Load CSV files into Database

```bash
$ pip install dbt
Successfully installed dbt-0.xx.x

$ vim ./.dbt/profiles.yml
$ dbt debug
  ...
  Connection test: OK connection ok

$ dbt seed
1 of 3 OK loaded seed file public.raw_customers............ [INSERT 100 in 0.48s]
2 of 3 OK loaded seed file public.raw_orders............... [INSERT 99 in 0.48s]
3 of 3 OK loaded seed file public.raw_payments............. [INSERT 113 in 0.46s]

Finished running 3 seeds in 1.36s.

Done. PASS=3 WARN=0 ERROR=0 SKIP=0 TOTAL=3
```

MoneyLion

SETUP

Setup and Test DB Connection

profiles.yml

```yaml
jaffle_shop:
  outputs:
    dev:
      type: postgres
      threads: 8
      host: localhost
      port: 5432
      user: postgres
      pass: docker
      dbname: postgres
      schema: public
  target: dev
```

MoneyLion

EXECUTE

Materialise Tables on Database

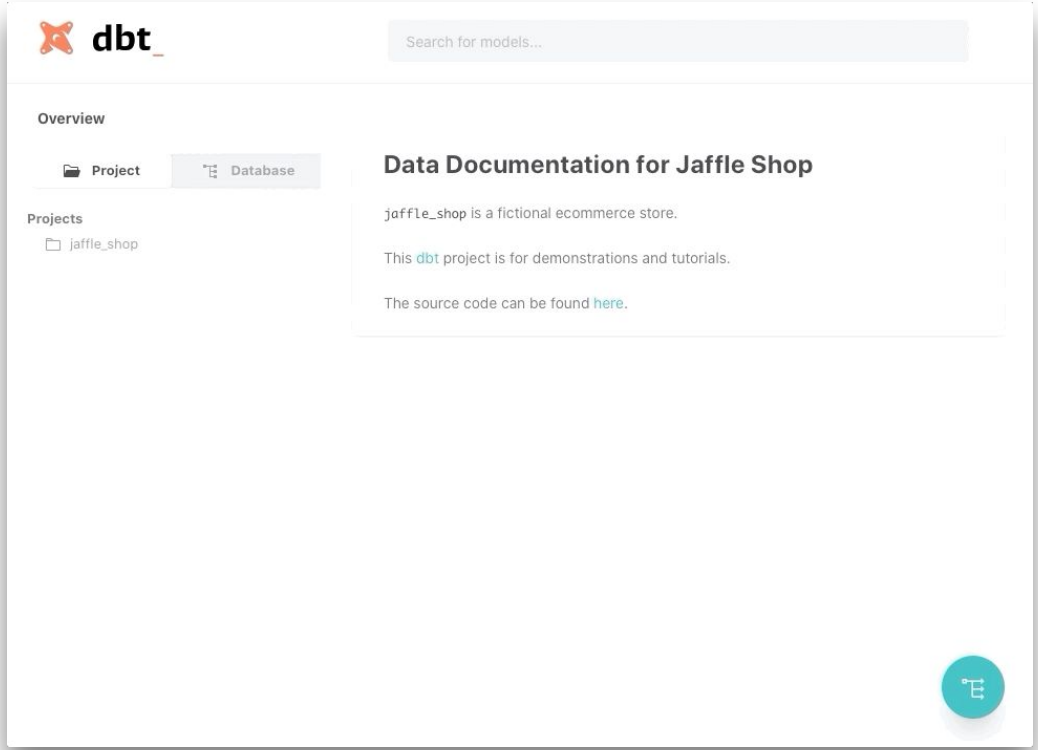
Generate Docs (HTML)

Host Docs Website *(Locally)*

```bash
$ dbt run
Concurrency: 8 threads (target='dev')
...
8 of 8 START table model public.dim_customers............... [RUN]
8 of 8 OK created table model public.dim_customers......... [SELECT 100 in 0.11s]

Finished running 3 view models, 5 table models in 1.29s.

Done. PASS=8 WARN=0 ERROR=0 SKIP=0 TOTAL=8

$ dbt docs generate
Building catalog
Catalog written to /Users/dbt/jaffle_shop/target/catalog.json

$ dbt docs serve
Serving docs at 0.0.0.0:8888
To access from your browser, navigate to: http://localhost:8888
```

MoneyLion

USE (DOCUMENTATION)



MoneyLion®

ONE MORE THING

MoneyLion

OTHER AWESOME FEATURES

- Hooks ***
- Snapshots
- Testing *****
- Custom schema tests ***
- Source freshness checks ***
- Configuring models ***
- Model selection syntax *
- Custom Profiles Directory *****
- Seeds **
- Macros *****
- Environmental Variables ***
- Docs Blocks **
- Using tags *
- Packages

** = level of awesomeness (i.e. we use a lot in MoneyLion)*

MoneyLion

GREAT DBT RESOURCES

OFFICIAL RESOURCES

- Tutorial (fishtown-analytics/jaffle_shop)
- Best Practices ***
- How we structure our dbt projects ***
- How we set up our computers for working on dbt projects
- How Monzo, the UK's favorite mobile-only bank, is rolling out dbt

COMMUNITY

- Join the dbt community (Slack channel)! ***

PERSONAL

- Example dbt docs: 1bk.dev/dbt-demo

MoneyLion®

END