

Introduction to dbt

SQL on steroids

Practical use case



Table of contents



01

Use case

What pushed us to use
dbt

02

Dbt

What dbt is and what it is
not

03

Demo time

You did not believe me,
here is a taste

↑
**Question
time**

↑
**Question
time**

01

Use case

Legal company

Why we had to choose dbt over many options



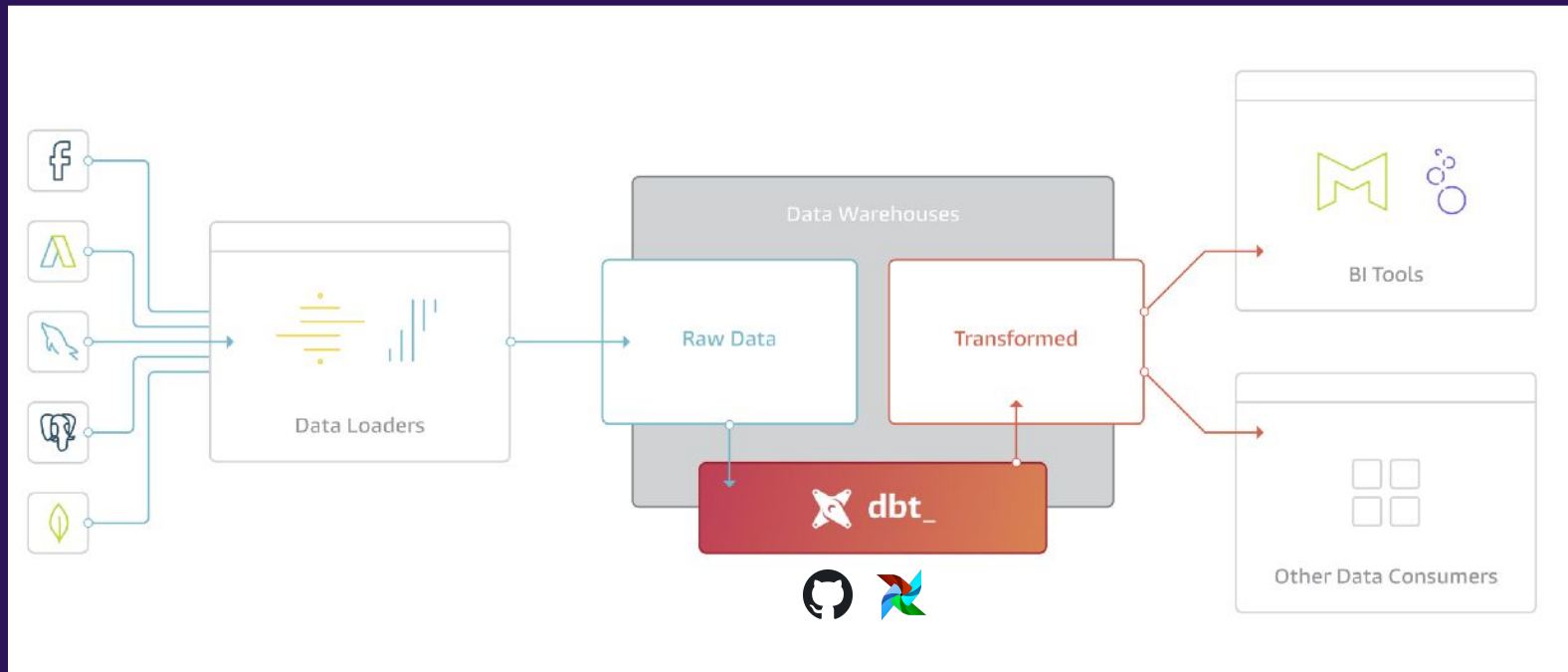
Requirements



- Clean several data sources from a sql database
- Aggregate the cleaned sources to provide insight for the data analysts
- Create a documentation (if possible interactive) for analyst to learn about the data
- Possibility of versioning and of collaboration with multiple teams
- Aggregate only events from aggregator like segment to provide user and session information
 - BUT, we do not know the number of events, their types, etc...



Implementation



02

What is dbt

And what dbt is not

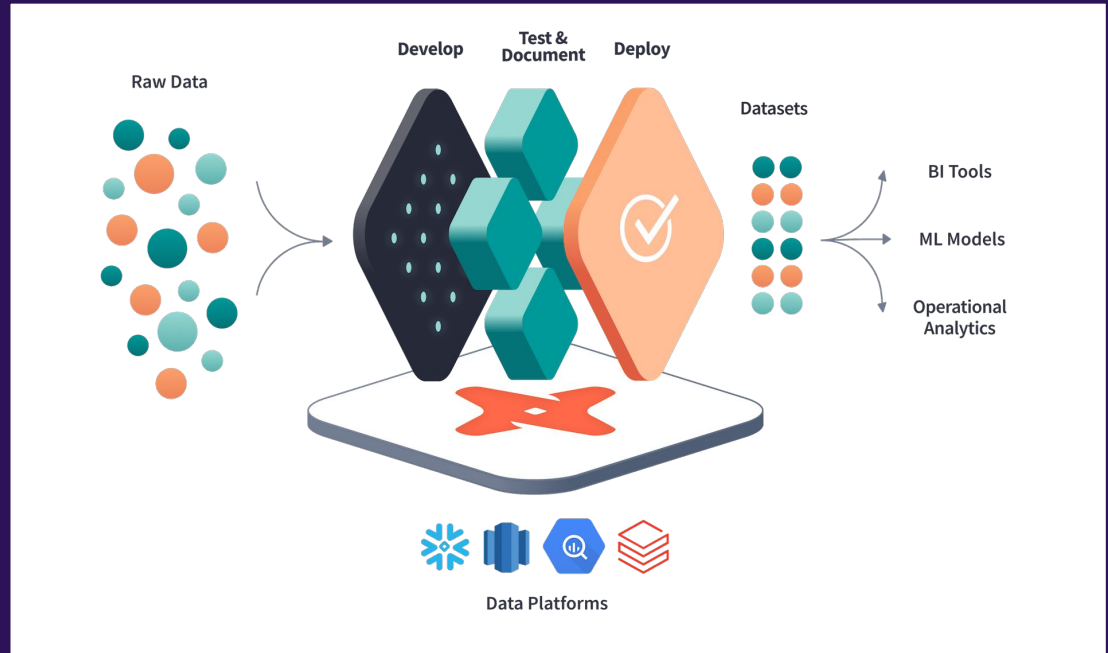
Dbt – Data Building Tool

DBT is :

- A python package
- A way to use “functionnal” SQL
- A documentation generator
- A testing framework

DBT is not:

- A SQL runner
- An end to end platform -> it is an interface





Dbt compilation



```
select
  id,
  firstname,
  lastname,
  {{ phone_to_hash('phone', 'phone') }},
  {{ mail_to_hash('email', 'email') }},
  lsp_type_c,
  convert_timezone('{{ var("timezone") }}', createddate) as createddate
from
  {{ source('salesforce', 'contact') }}
where
  isdeleted = false
  and id not in (select id from {{ ref('contact_test') }})
```

“Code is Law” - Lawrence Lessig

```
with __dbt__cte__contact_test as (

select
  id
from
  stitch_raw.salesforce.contact
where
  isdeleted = false
  and (
    email like 'test.10%'
    or email like 'agtest%'
    or email like 'allicloud.test%'
    or (lower(firstname) like 'test%'
      and lower(lastname) = lower(firstname)))
  and email is not null
)select
  id,
  firstname,
  lastname,
  CAST((MD5_BINARY(NULLIF(UPPER(TRIM(CAST(
    'a' ||
    case
      when len()
      case
        when startswith(
          regexp_replace(phone, '\\D+', ''))
        , '1')
        then
          regexp_replace(phone, '\\D+', ''))
        else '1' ||
          regexp_replace(phone, '\\D+', ''))
      end
    ) <> ''
    then null
    else
      case
        when startswith(
          regexp_replace(phone, '\\D+', ''))
        , '1')
        then
          regexp_replace(phone, '\\D+', ''))
        else '1' ||
          regexp_replace(phone, '\\D+', ''))
      end
    end
    AS VARCHAR(16))), '')) AS BINARY(16)) AS phone
  ,
  CAST((MD5_BINARY(NULLIF(UPPER(TRIM(CAST(email AS VARCHAR(16))), ''))) AS BINARY(16)) AS email
  ,
  lsp_type_c,
  convert_timezone('America/Los_Angeles', createddate) as createddate
from
  stitch_raw.salesforce.contact
where
  isdeleted = false
  and id not in (select id from __dbt__cte__contact_test)
```


Documentation Generation

- Dynamically obtain a dependency graph
- See the tests involved
- Get a description of each tables and fields as well as their types

The screenshot displays the dbt documentation interface in a web browser. The main content area shows the details for the 'snowplow_page_views' incremental model, including its owner (public_integration_test_user), type (table), and package (snowplow). The description states that this table represents a list of page views and information about the time spent and scroll depth. The columns section lists user_custom_id, user_snowplow_domain_id, and user_snowplow_crossdomain_id, all of which are character types.

On the right side, a 'Lineage Graph' overlay is visible, showing the data flow from source tables (snowplow_web_events_time, snowplow_web_events_scroll_depth, snowplow_web_events, snowplow_web_events_internal_bed) to the target table (snowplow_page_views), which then feeds into snowplow_sessions_tmp.

```
graph TD; A[snowplow_web_events_time] --> D[snowplow_page_views]; B[snowplow_web_events_scroll_depth] --> D; C[snowplow_web_events] --> D; E[snowplow_web_events_internal_bed] --> D; D --> F[snowplow_sessions_tmp];
```



Question time

Demo time

Wouhouuuu !!!

Thanks!

Does anyone have any questions?

Contact:

maxime.bonn@gmail.com

Github of demo:

+33 6 08 33 06 47

CREDITS: This presentation template was created by [Slidesgo](#)