

## 1 Create a Slack Account!

Available for Android & iOS



## 2 Go to getdbt.com/community & add your info

### Join the dbt Community

Sign up to get an invite to dbt Slack, and subscribe to the dbt Community newsletter.

First, we need some basic information:

First Name

Last Name

Work email\*

Company

Next

### Join the dbt Community

Help us learn about our community

These questions are optional, but answering them helps us make dbt even better

How did you hear about dbt?

What is your title at work?

Which database do you use?

Where are you located?

Next

## 3 Review and acknowledge Community rules

### Join the dbt Community

One last (but important) thing!

You're about to receive an invite to dbt Slack, dbt Slack is special.

We currently have thousands of data professionals in this community, many of whom genuinely love this community. It's filled with smart, kind, and helpful people who share our commitment to elevating the analytics profession.

We are committed to maintaining the spirit of this community, and as such have written some guidelines to help new members understand how to best participate in our Slack. As a reference, you can always view the rules of the road here, but we'll show you what you need to know before joining on the following pages.

Read the rules

Already a member? [Sign in to Slack](#)

### Join the dbt Community

Thanks for reading the rules of the road

Thank you for your help in ensuring the health of our community. We're glad to have you with us. Click the link below to receive an invitation to dbt Slack.

I agree to these rules ☒ I agree

Send me emails about dbt ☒ Sure

Send me my Slack invite

Already a member? [Sign in to Slack](#)

## 4 Click on invite link and follow instructions

Welcome to the community

Now go get your invite **right here!** See you in Slack!

Already a member? [Sign in to Slack](#)

## 5 Join **#local-berlin** channel to chat with us!

If you are unable to join - please email [anna.filippova@dbtlabs.com](mailto:anna.filippova@dbtlabs.com) for an invite

# Welcome

---

## Berlin dbt Meetup

Host: TIER Mobility

**Join dbt community Slack**

[getdbt.com/community](https://getdbt.com/community)

#local-berlin



# Victoria Perez Mola

*Senior Analytics Engineer @  
TIER Mobility*

*(she/her)*

*I love using technology to make lives  
easier, yet I don't know how to use a  
blow dryer*



[linkedin.com/in/victoriaperezmola](https://www.linkedin.com/in/victoriaperezmola)



[victoria.mola@tier.app](mailto:victoria.mola@tier.app)

# Sowmia naraynan Sivasubramanian

*Head of Analytics Platform @  
TIER Mobility*

*(he/him)*

*I believe in 80/20 rule. Yes, it took  
80% of my time to fill up the fun fact  
here and you know the rest..*



[linkedin.com/in/sowmianaraynan](https://www.linkedin.com/in/sowmianaraynan)



[sowmia.subramanian@tier.app](mailto:sowmia.subramanian@tier.app)

# Data @ TIER Mobility



# TIER Mobility in numbers

Operating in:

**17**  
Countries

**150**  
Cities

**16**  
Capital Cities



Partnerships:



**30+** with Public Transportation  
(local and regionals)

**15+** with Private Mobility Partners

**+55**  
Nationalities  
in TIER team

Multimodality:

**120.000+**  
Live vehicles



**6 million+**  
riders

Fundraising

**650 million USD**



Key investors:



MUBADALA

SoftBank

Key Partners:



talabat



## We have a diverse set of roles in data

### Data Engineers

Make sure data goes from source to destination and new data points are added

### Analytics Engineers

Make sure data is accessible for everyone and maintain the data warehouse and analysts/scientists have clean data



### Data Scientists

Work on clean data and build advanced analytical insights or machine learning models.

### Data Analysts

Take raw or clean data and build insights and dashboards.

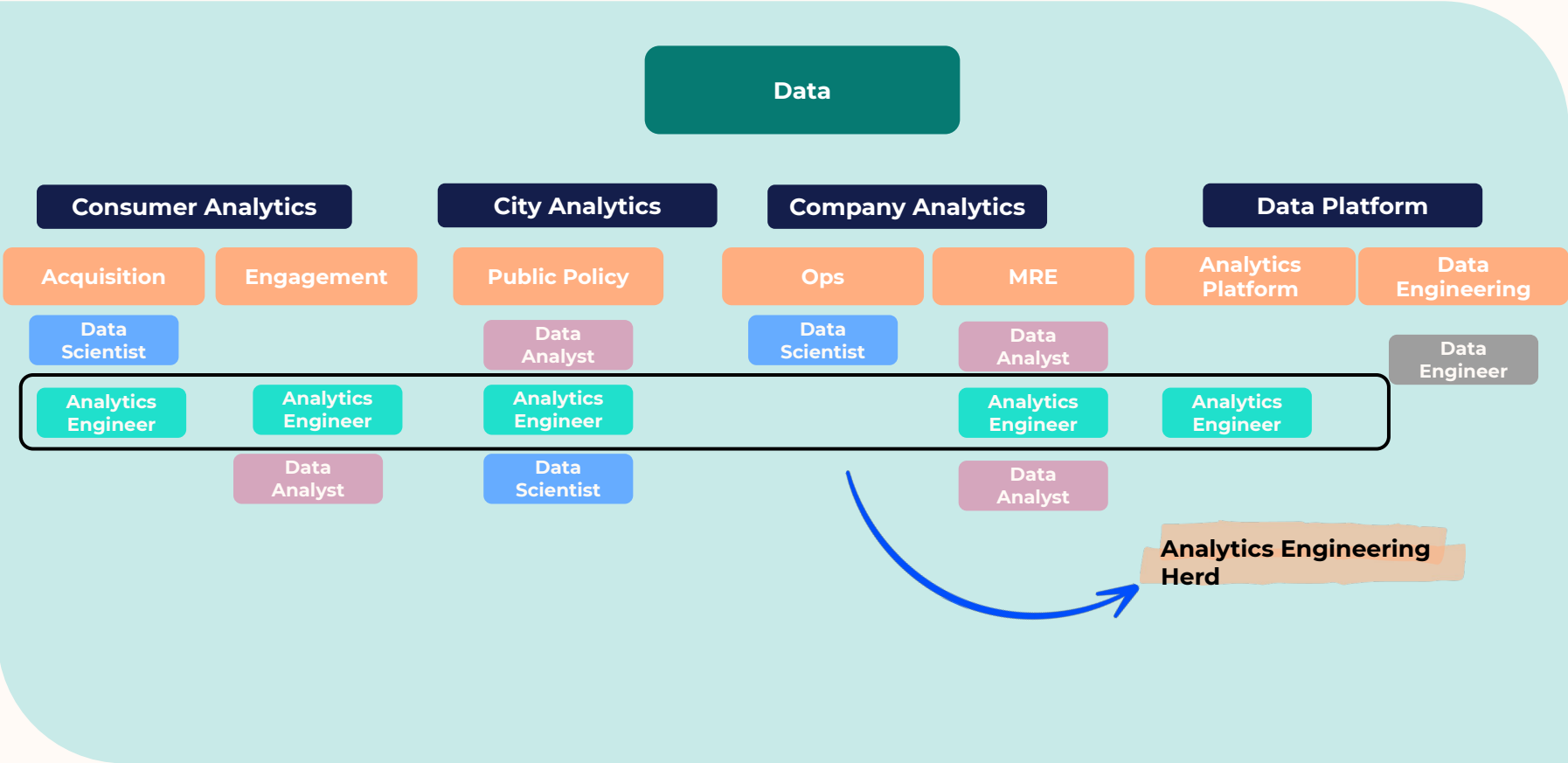


### Head / Directors of Data Analytics

Manages team capacity, acts as product manager and together with stakeholders drives the team's roadmap

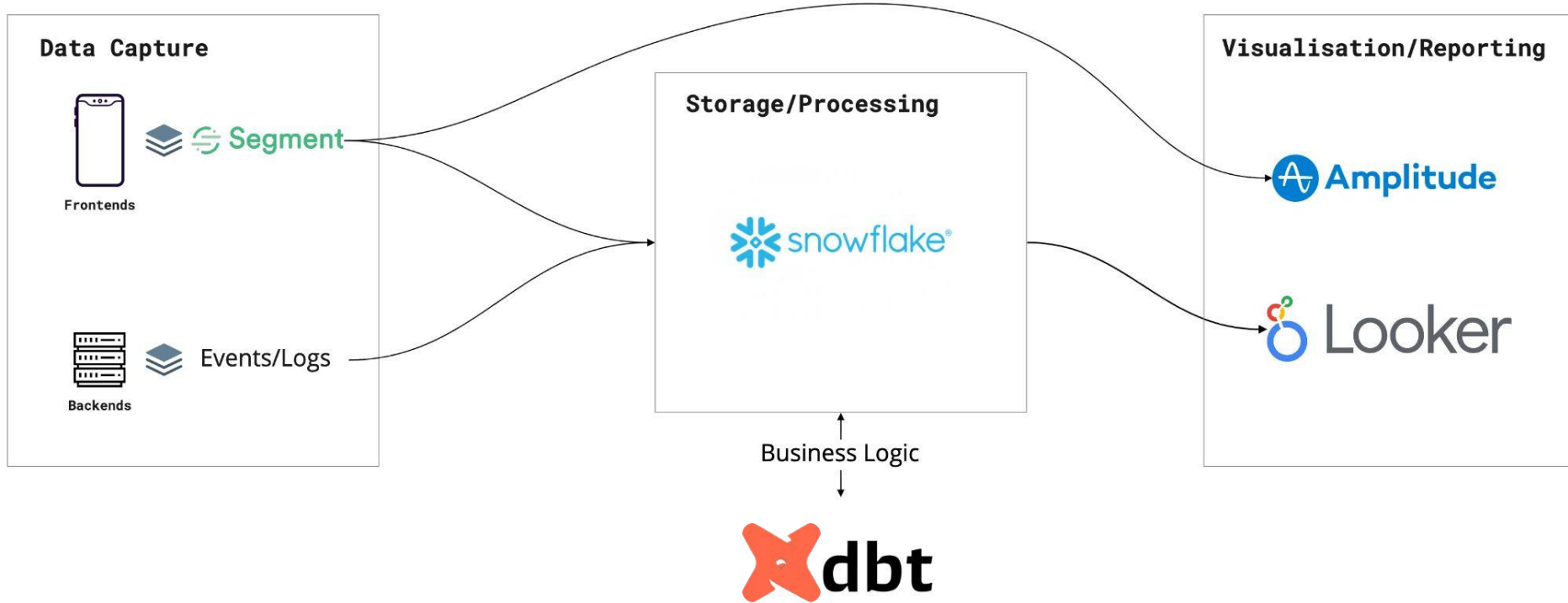
Currently **50 people** in the data analytics team but we continue to grow.

# Data team





## Data stack from a bird's eye view



# And we are hiring...

If you are interested in our mission and would like to join our team, feel free to apply  
**<https://about.tier.app/jobs>**

TIER Mobility SE · Eichhornstr. 3 · 10785 Berlin · Germany  
[www.tier.app](http://www.tier.app) · [info@tier.app](mailto:info@tier.app)

TIER

# What's going on in the dbt Community?



*The Analytics Engineering Podcast*



*dbt Community Newsletter*



*Coalesce 2021 Conference*



# **coalesce**

December 6-10, 2021



# coalesce

The Analytics Engineering Conference

**DECEMBER 6-10, 2021**



# Join dbt Community Slack

*[getdbt.com/community](https://getdbt.com/community)*

*add #local-berlin*

# New to dbt Slack?

## Got a question?

Use a **?** emoji before your message. e.g. **?** Why is the sky blue?

## I have the same question!

If you see another question that you want answered, add a **+** reaction — this helps us prioritize questions.

## My question got answered:

If your question gets solved (e.g. someone in the thread answers it for you), react with a **✓** emoji.

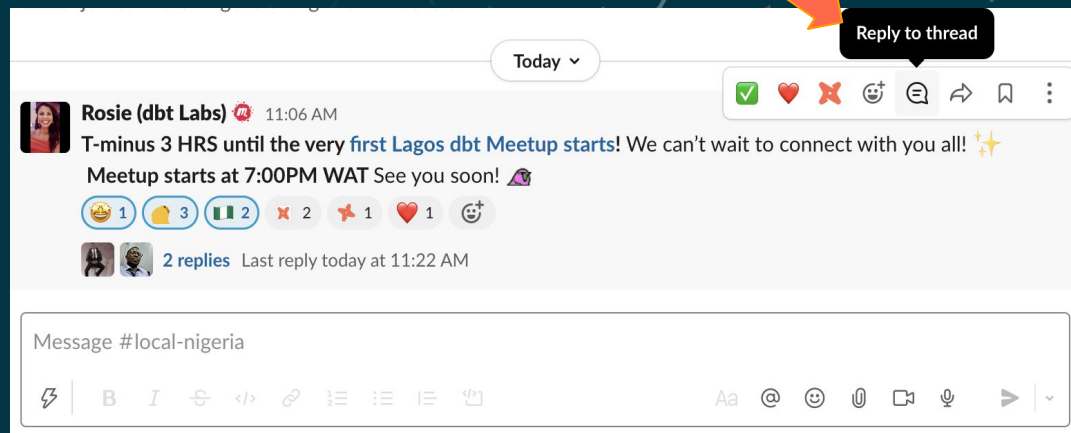
## I'm ready to interact!

Awesome! Non-question banter is strongly encouraged — use all the reactions and gifs you want!

## KEY THINGS TO REMEMBER:


**Be Respectful and use Slack threads**

We  threads!



Today ▾



**Reply to thread**

 **Rosie (dbt Labs)** 11:06 AM








T-minus 3 HRS until the very **first Lagos dbt Meetup starts!** We can't wait to connect with you all! ✨

Meetup starts at 7:00PM WAT See you soon! 🎉

👑 1 🎁 3 🇳🇬 2 ✖ 2 ⭐ 1 ❤ 1 😊

  **2 replies** Last reply today at 11:22 AM

Message #local-nigeria

⚡ | **B** *I*        Aa @ 😊 📎 📺 🔊 ▶ ▾

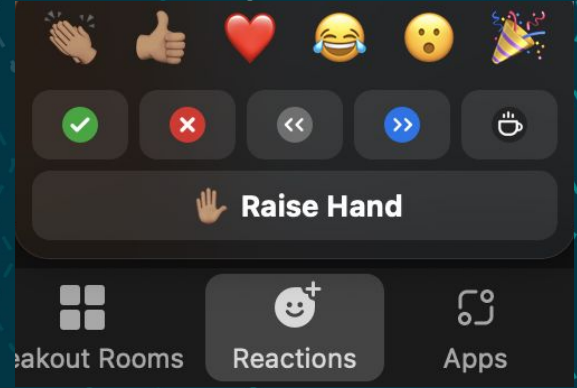
# Zoom Etiquette



**Video On**  
(if possible)



**Mute Mic**  
(to avoid background noise)



**Use Zoom Controls**  
(raise hand and other reactions)



# Agenda

- 6:30 - Welcome Remarks
- 6:40 - Coalesce teaser w/ Icebreaker
- 6:50 - Modeling Structures (tips & tricks w/ designing your models) by Alex Ryan
- 7:20 - Incremental Models: It means no worries for the rest of your runs by Faisal El-Shami
- 7:45 - Closing Remarks & Feedback Session
- 8:00 - *Optional informal chat*

# Speakers for Today

---



Alex Ryan



Faisal El-Shami

# Alex Ryan

*Data Scientist @TIER Mobility*

*(he/him)*



alexander.ryan@tier.app

# An Essence of Data Modelling

Alexander Ryan

18th November 2021

TIER



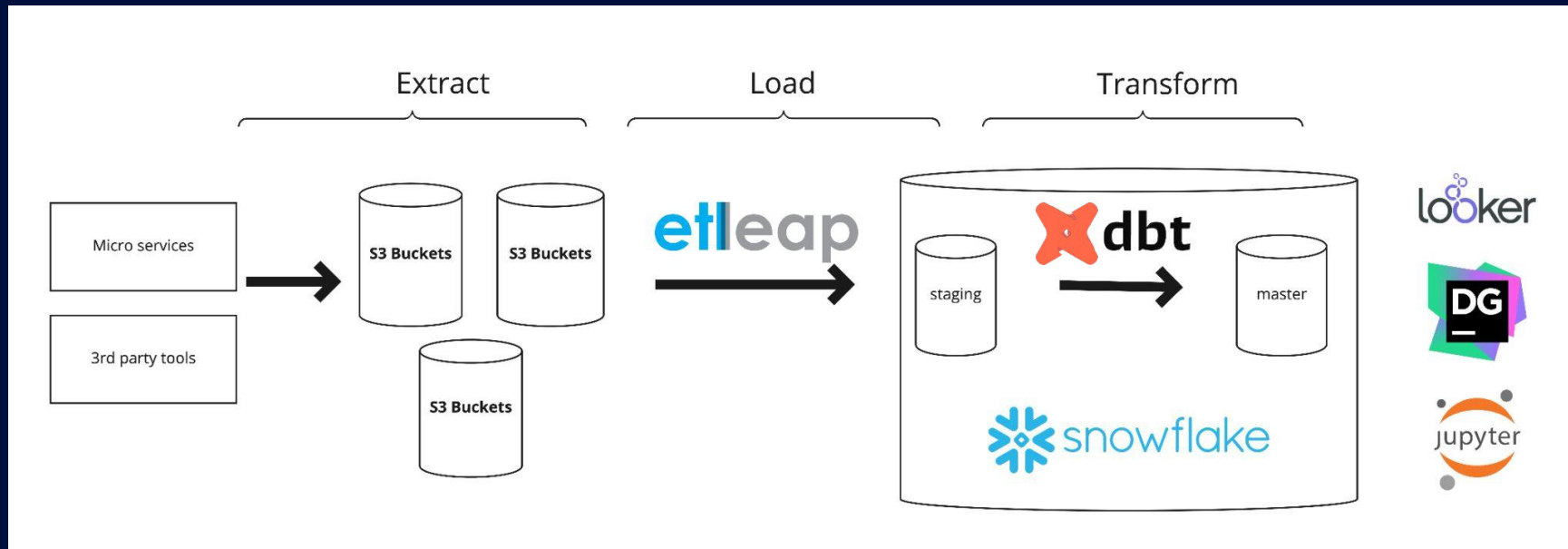
# Contents

1. Defining Data Modelling
2. A Generalised Structure of Transformation Layers
3. CTEs
4. Happy families are all alike: Families of models
5. Words, words, words: Renaming fields
6. Git it? Got it. Good.
7. A Brief Interlude into my Workflow
8. List of Banal Platitudes



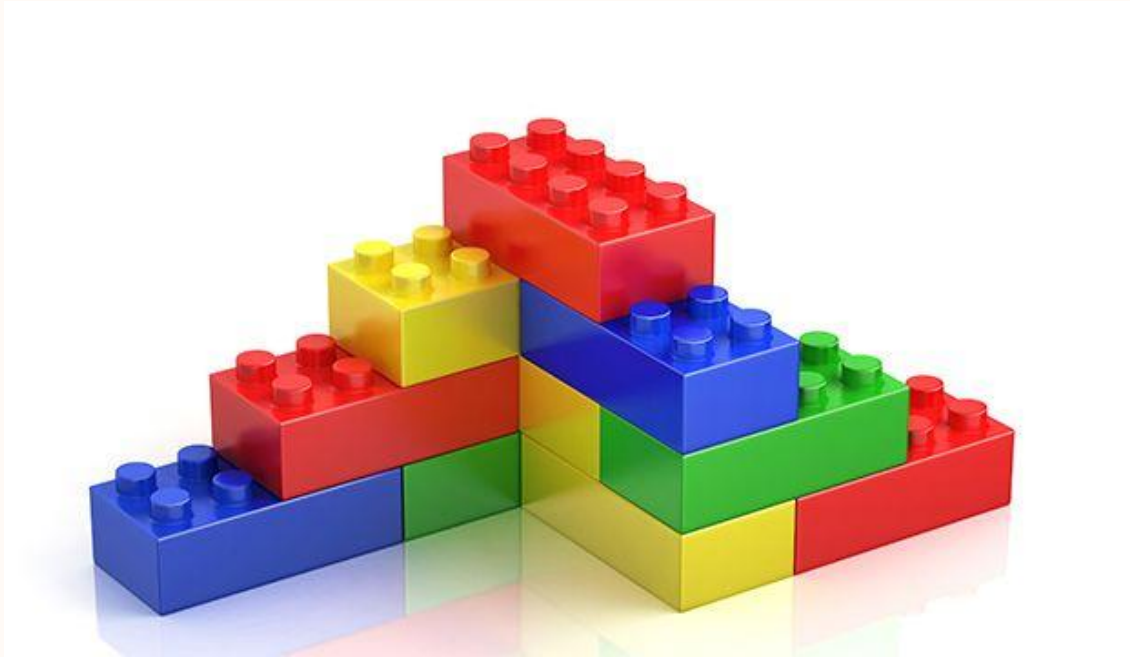
# **What We Talk About When We Talk About... Data Modelling**

# ELT Implementation @ TIER



## CTEs: Common Table Expressions

- What is a CTE? -> Modular select statements that can be chained together in a dbt model
- Each CTE should do one core thing
- Each cte should have a label that describes the said function performed on the data.
- Don't write out each separate field or column -> use \* where possible.
- Include \* at the very end of the list of columns in the SELECT statement -> newly created field will appear first





## CTEs part 2. Mortal sins: Nested Queries

Aim to have CTE logic to flow in one direction

Building sequentially on top of one another

Write each new CTE as such: **,customers as (**

When in a SQL Runner IDE, to see result of a particular CTE,  
comment out the above line

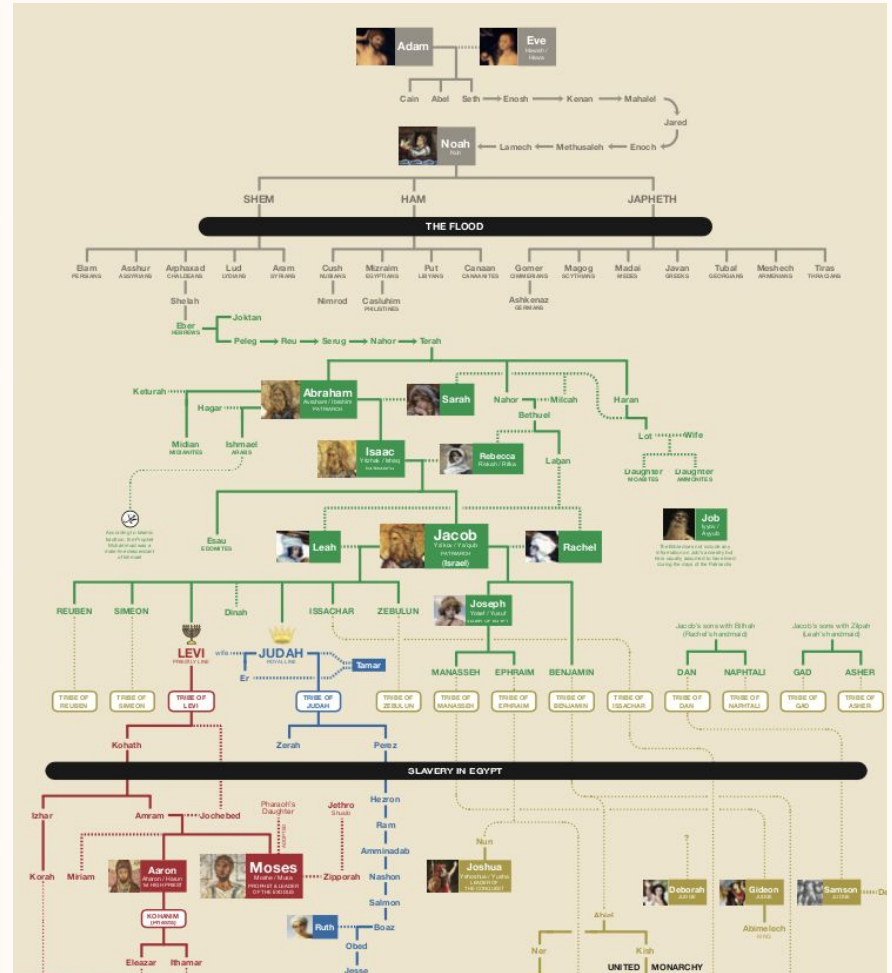
When using dbt run, to see result of a particular CTE, change the  
final select \* to another cte above

```
•with processing as (  
  select  
    customer_id,  
    count(distinct rental_id)  
  from (select * from customers) as customers  
  left join (select * from rentals) as rentals using customer_id  
)  
  
•select *  
  from processing
```

```
•with rentals as (  
  select *  
  from rentals  
)  
  
•,customers as (  
  select *  
  from customers  
)  
  
•,processing as (  
  select customer_id, count(distinct rental_id)  
  from customers  
  left join rentals using customer_id  
)  
  
•select *  
  from processing
```

# Happy families are all alike

- Families of interrelated models are better than developing isolated models.
- When you see repeating logic in multiple CTEs, think about spinning off the logic into its own data model
- This should reduce the total number of models, thereby reducing maintenance costs for analytic engineers
- Take Note: Analytic Engineers will need to be careful making changes upstream, and be mindful of how their changes will affect downstream models



## Words, words, words: Renaming fields

- Aim for **consistent** naming across data models in transformation steps
- Renaming should occur in the final core model that you expose to stakeholders
- Poorly named, or inconsistent naming can be frustrating to manage, and lead to more time reading documentation.
- Consistent names of fields allow one to follow the field through the different chains of transformations, which can assist with debugging models

Unknown or inaccessible field  
"user\_order\_facts.lifetime\_orders" referenced in  
"users.lifetime\_orders". Check for typos and  
missing joins.

▼ 2 occurrences

users:79 (ecommerce:order\_items)

users:79 (ecommerce:orders)

## Git it? Got it. Good.

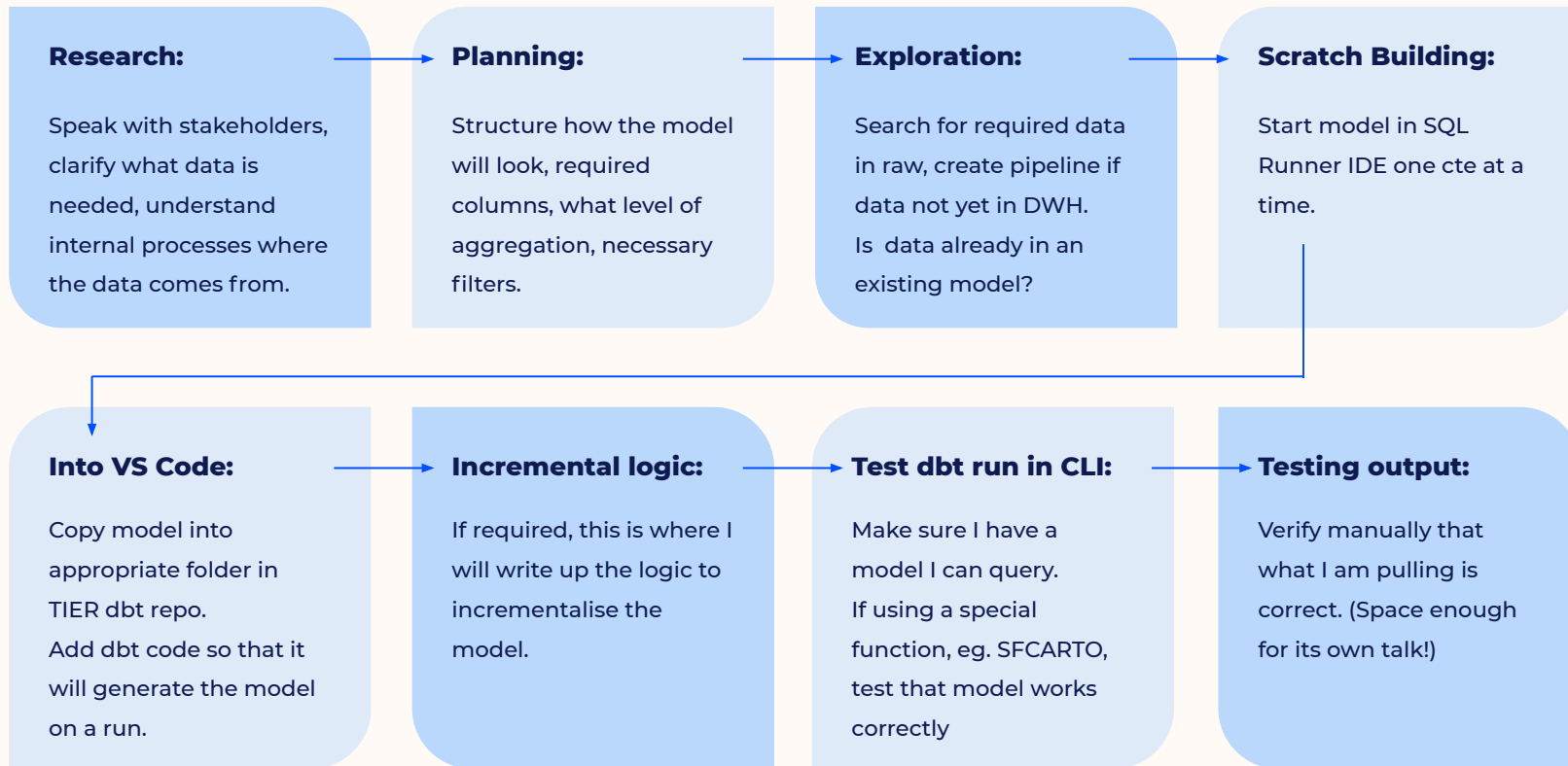


TIER

- **dbt + git** allows a team of data experts to work together, share code, maintain production branch and development branches, and ultimately provide a level of data integrity that is regularly shipped with software
- Enables junior members of the team to learn from the senior analytic engineers.
- Set a general **style guide** for the whole data team across all dbt models
- Add this as part of your PR review process when pulling changes to models or adding new models

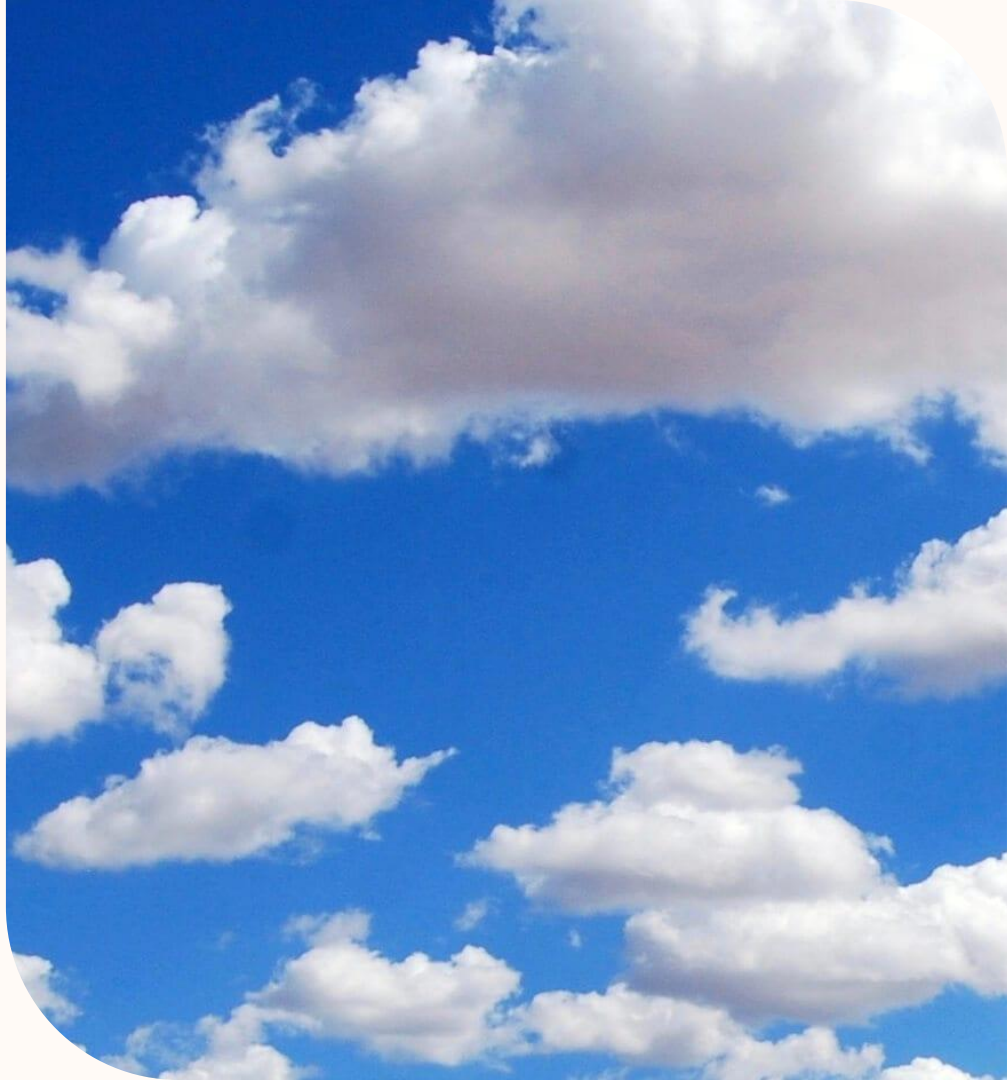


## A Brief Interlude into my Workflow



## List of Banal Platitudes

- Learn how other Analytic Engineers structure their models.
- Build sequentially, one cte at a time.
- Don't be afraid to go back and improve prior cte logic.
- Robust models are better than finely coded models with elaborate structures that break easily.
- **Be kind to yourself.** No one builds perfect models from scratch.



# Thank you!





---

*Please post your questions in #local-berlin*



# Faisal El-Shami

*Analytics Engineer @TIER Mobility*

*(he/him)*

*I always say gm,  
No matter what time it is*



[linkedin.com/in/faisalelshami](https://www.linkedin.com/in/faisalelshami)



[faisal.elshami@tier.app](mailto:faisal.elshami@tier.app)

# Contents

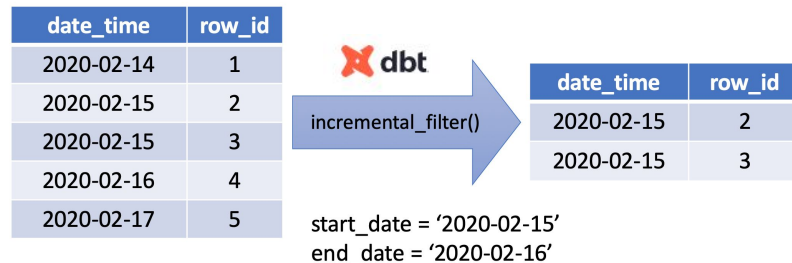
1. What are incremental Models?
2. When to Use Incremental Models
3. How to Incrementalize Models
4. Additional Tips
5. Usage of Incremental models at TIER



# 1. WHAT ARE INCREMENTAL MODELS

Incremental models are built as **tables** in your data warehouse.

- After the first dbt model run, dbt **transforms only the rows in your source data that you tell dbt to filter for**, inserting them into the table that has already been built (the target table).
- This reduces the build time by just transforming new records
- They require **extra configuration** and one should be more careful
- They adapt very well to **event-style data**, may not be applicable to other types



## 2. WHEN TO USE INCREMENTAL MODELS

### **SOURCE DATA IS HUGE & MODELS TAKE TOO LONG TO RUN**

the transformations on the source data are computationally expensive (that is, take a long time to execute), for example, complex Regex functions are being used to transform the data source

### **HISTORICAL DATA DOESN'T CHANGE**

### **SAVE ON TIME AND RESOURCES**

### **MODELS MUST BE UPDATED FREQUENTLY**

For example hourly model updates

## 2. WHEN **NOT TO USE INCREMENTAL MODELS**

**SOURCE DATA IS LITTLE!**

**DATA IS CONSTANTLY CHANGING**

**MODELS RELY HEAVILY ON WINDOW FUNCTIONS**

*Note: Can be very difficult to apply  
Incremental Logic, but not impossible.*



### 3. How to Incrementalize a Model

- **Define materialization in the model config**

Incremental models are defined with **select** statements, with the the materialization defined in a config block.

- **Identify the uniqueness constraint of the model (if any)**

If the **unique key** of an existing row in your target table matches one of your incrementally transformed rows, the existing row will be (technically) **updated**.

- **Define row filtering for an incremental run**

- What are the new rows?
- Use **is\_incremental()** and **{{ this }}**

```
1  {{config(
2    → materialized = 'incremental',
3      tags = ["tag"],
4    → unique_key = 'event_id'
5      )
6  }}
7
8  SELECT
9
10     event_id, --A Unique ID per event
11     timestamp_utc,
12     column1,
13     column2
14
15  FROM  {{ ref('table_name') }}
16
17  WHERE 1 = 1
18
19  (
20    {% if is_incremental() %}
21      AND timestamp_utc >= (SELECT max(timestamp_utc) FROM {{ this }})
22    {% endif %}
23  )
24
```

## 4. Additional Tips

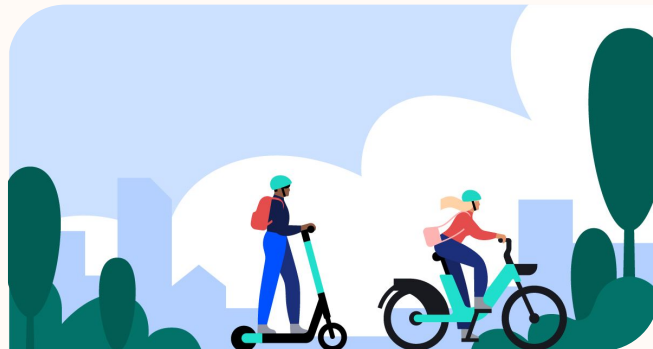
- `{{ this }}` points to the model you're working on, not the table you're selecting from
- If your model has a part that can't be incremental because of **window functions** and some part that can, *you may consider divide the model into two: one incremental + one table*
- **Unique key doesn't ensure uniqueness**, it's only an indicator for dbt on how to identify the rows that should be "updated".
- If you won't be **"updating"** rows then avoid using a unique key, it requires more resources



## 4. Additional Tips

- For more complex incremental models that make use of CTEs, **you should consider the impact of the position of the `is_incremental()` macro on query performance**. On some warehouses, filtering your records early can vastly improve the run time of your query!
- When adding new columns to source models , or make a major modification, you must use the **`--full--refresh`** to apply changes on all data

```
$ dbt run --full-refresh --select my_incremental_model+
```

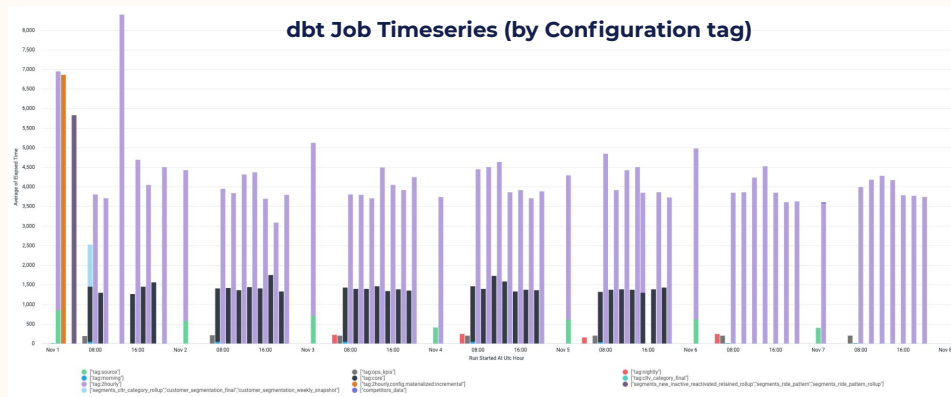




## 5. Incremental Usage At TIER

## TIER dbt models in a nutshell

- **400+ dbt Models, 22% are incremental**
- **Frequent dbt Runs** ( every 1-2 hours)
- Run **--Full--refresh** on incremental models on Weekends
- Natural complexity progression with company Growth ( Heavy Computations )



### Some dbt Runtime example of my Models:

Model	Table	Incremental
Model A (1 bn+ rows)	55 minutes	42 seconds
Model B (1 bn+ rows)	1.2 hours	1 minute
Model C (millions of rows + Heavy computation)	25 minutes	3 minutes

## 5. Incremental Usage At TIER

### The Safe Approach

```
1  {{config(
2      materialized = 'incremental',
3      tags = ["tag"],
4      unique_key = 'event_id'
5  )
6  }}
7
8  SELECT
9
10     event_id, --A Unique ID per event
11     timestamp_utc,
12     column1,
13     column2
14
15 FROM  {{ ref('table_name') }}
16
17 WHERE 1 = 1
18
19 {% if is_incremental() %}
20     AND timestamp_utc >= (SELECT max(timestamp_utc) FROM {{ this }})
21 {% endif %}
22
23
24
```

- The Unique Key is used to update records
- Notice the use of ' $\geq$ ' in 'is\_incremental()'

**Safe way to incrementalize, works straight forward with simple models**

## 5. Incremental Usage At TIER

### Using the 'pre\_hook' Configuration

```
1  {{config(
2    tags = ["your_tag"],
3    materialized = 'incremental',
4    pre_hook = '
5      {% if is_incremental() %}
6        delete from {{this}} where date_hour_utc >= dateadd(hour, -24, getdate())
7      {% endif %}'
8  )
9  }}
10
11
12 WITH CTE AS (
13
14   SELECT
15
16     date_hour_utc ,
17     column1,
18     column2
19
20
21   FROM {{ ref ('table') }}
22
23   WHERE 1 = 1
24
25   {% if is_incremental() %}
26     AND date_hour_utc >= dateadd(hour, -24, getdate())
27   {% endif %}
28
29 )
30
31 SELECT * FROM CTE
32
```

#### Pre\_hook statement

- A SQL statement (or list of SQL statements) to be run **before** the model is built
- We use this method when relying on a unique key is not applicable

#### Not Ideal and can have issues:

- **First dbt run always Fails**
- **Process 24 hours of data every time vs referring to the latest timestamp**

## 5. Incremental Usage At TIER

### Incremental with Unions

```
1  {{
2    config(
3      schema = '{{target.schema}}',
4      tags=["tag_name"],
5      materialized='incremental'
6    )
7  }}
8
9
10 -- select new data from active pipeline
11
12 with base as (
13   select * from {{ ref('table_a') }}
14
15   {% if is_incremental() %}
16     where timestamp_utc >=(select max(timestamp_utc) from {{ this }})
17   {% endif %}
18 ),
19
20 -- union new data with some of the already materialised model's data as
21 processing_incremental_data as (
22   select * from base
23
24   {% if is_incremental() %}
25     union all
26     select * from {{ this }}
27     where timestamp_utc < (select max(timestamp_utc) from {{ this }})
28   {% endif %}
29 ),
30
31 -- continue query
```

TIER

- This approach is used on large data sets where we want to just process the latest updates.
- Useful when we have **window functions** and historical data is needed for business logic application
- **Base CTE:** takes all the unprocessed data. You can apply transformations here!
- **Processing\_incremental\_data:** Select \* from base **CTE** and unions that to all the existing data in the table **{{ this }}**

## 5. Incremental Usage At TIER

### Method: Elephant by Chunks

```
1 {{
2   config(
3     schema = '{{target.schema}}',
4     tags=["tag_name"],
5     materialized='incremental',
6     unique_key='event_id'
7   )
8 }}
9
10 -- select new data from active pipeline
11
12 with base as (
13
14   select * from {{ ref('table_a') }}
15
16   {% if is_incremental() %}
17
18     where timestamp_utc > (select max(timestamp_utc) from {{ this }})
19
20     {% if var('incr_chunk_days',default=false) %}
21
22       -- if optionally set to upper bound the incremental filter to only x days
23       and timestamp_utc <
24         (select dateadd(day,{{ var("incr_chunk_days") }}, max(timestamp_utc)) from {{ this }})
25
26     {% endif %}
27
28   {% endif %}
29 ),
30
```

```
$ dbt run --models model_name --var 'incr_chunk_days: 90
```

- Add the variable **incr\_chunk\_days** in the incremental blocks of the model as seen in the **base** CTE
- When running dbt you can add the **--var** with the number of days (in this example) that is needed to be process in chunks
- This allows you to reprocess a chunk of the data rather than the full pipeline.



---

*Please post your questions in #local-berlin*

# Thank you!

*Stay tuned for upcoming events and other content.*

Berlin dbt Meetup Group  
#local-berlin Slack Channel  
[about.tier.app/jobs](https://about.tier.app/jobs)

**Before you leave...**

*Please share your feedback with us!*

