

Narrator

How to end the long tail of data requests?

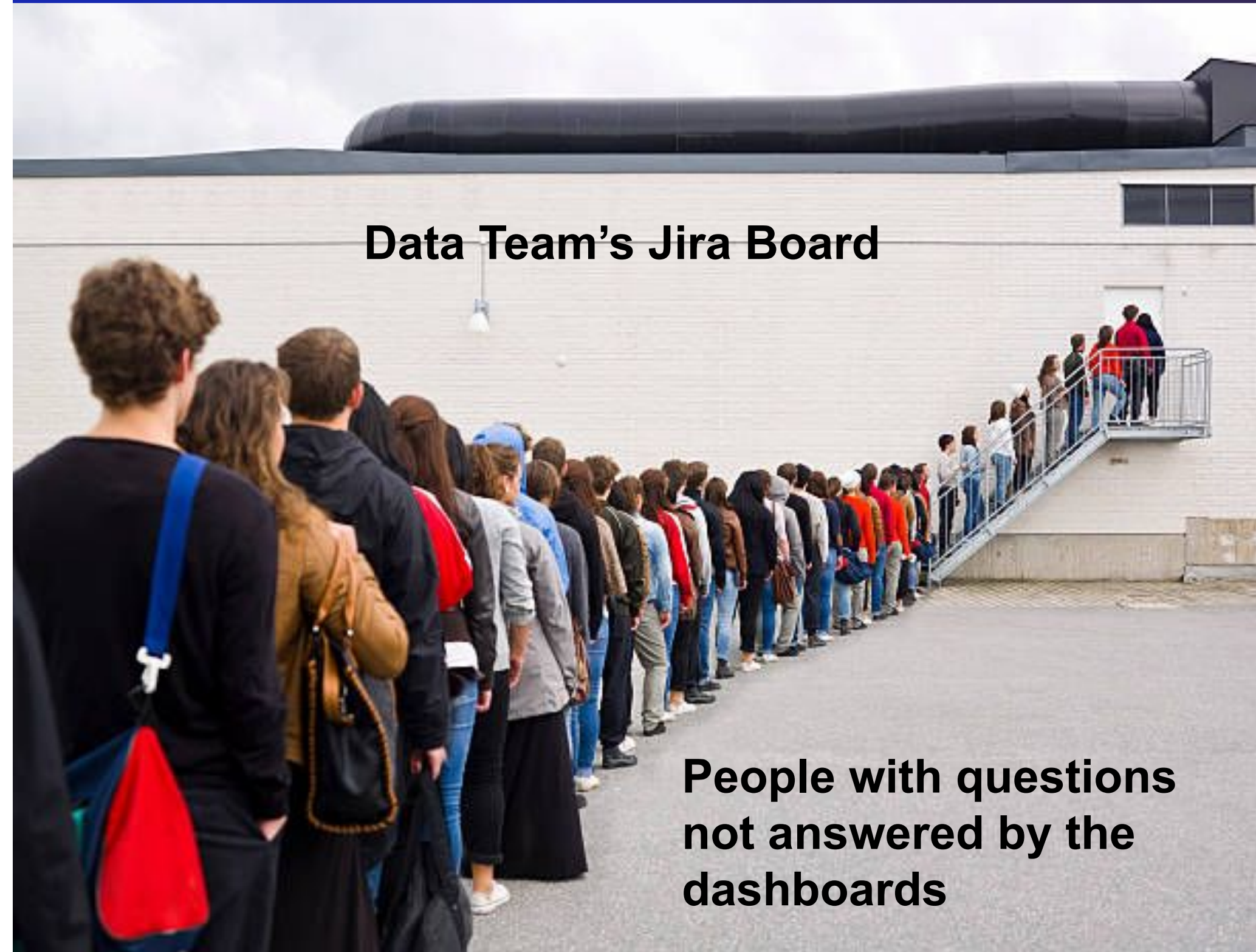
A case for using the Activity Schema as a staging layer

By: Ahmed Elsamadisi

How do we deal with ad-hoc requests today?

1. Let's take a simple **E-Comm** example with only 3 sources (Shopify, Klaviyo, Snowplow)
2. We start with a **couple of tables** that we think will answer **most questions**
 - **order_dim**
 - **email_dim**
 - **web_session_dim**
3. When we get a **question** that these tables can **answer**, everything is great!
 - Q1: How many emails were sent last week?
 - Q2: How many orders came from anyone who received the YOUR_GREAT campaign?
4. We are **crushing** it!!!

**Unfortunately,
questions keep on
coming and they get
SPECIFIC**



Data Team's Jira Board

**People with questions
not answered by the
dashboards**

Q21: How many people who came from the IMPORTANT TEST, ordered again?

1. Go to the dim_order table
2. JOIN Emails to Orders: Well, that doesn't exist so lets **approximate** it with time and add a **last_email_id** with **another join** and max based on time
3. JOIN Order to Next Order: Add a **LEAD** function and grab the **next_order_id**
4. Build a dashboard to show the **conversions** by **last_email_campaign**

Q21: How many people who came from the IMPORTANT TEST, ordered again?

1. Go to the dim_order table
2. JOIN Emails to Orders: Well, that doesn't exist so lets **approximate** it with time and add a `last_email_id`
3. JOIN Order to Next Order: Add a **LEAD** function and grab the `next_order_id`
4. Build a dashboard to show the **conversions** by `last_email_campaign`

How come I see conversions from a campaign we no longer use?

5. Add some more logic to ignore emails that are more than 3 days old



Q28: How many people who came from the IMPORTANT TEST (only if they opened the email within 30 minutes), ordered again within a week?

1. ...
2. Okay, well emails should now have a `first_opened_at` and `last_opened_at` and we can check if either....
3. **NO**, lets create an `opened_email_dim` and JOIN on it, but then make sure I don't **duplicate** incase there **are 2 opens in the last 30 mins**
4. Also don't want to break the current dashboard so lets add `last_email_within_30_id`, and `next_order_within_week_id`

Q42: Does the number of orders a customer have impact likelihood to order again?

1. ...
2. Okay, another join, another aggregation ...
3. Add `total_orders_before` add **window functions**

Q46: Does the # of emails impact it?

1. ...
2. Okay, another join, another aggregation ...
3. Add **total_emails_before** add **window functions**

**You get it, soon you have so many
columns**

So What?

- More columns confuse the source of truth
- Queries are way slower
- Higher risk of mismatched numbers
- Its SQL, so any of the additional joins can duplicate or drop rows

MISTAKE

Build an Ad-Hoc Layer using the Activity Schema

The Activity Schema - a perfect staging layer

Activity Schema

TS	Activity	Customer
2022.01.10		_____
2022.01.12		_____
2022.01.13		_____
2022.01.14		_____
2022.01.15		_____
2022.01.17		_____
2022.01.17		_____

- **Business aligned** terminology so mapping questions to the data is easy
- Everything is happening **in time** so change is easily understood
- Data prep for **definitions** not questions

Learn more at ActivitySchema.com

Querying via Temporal Joins



TS	Customer	Primary	Append
2022.01.03	██████████	●	null
2022.01.04	██████████	●	●
2022.01.07	██████████	●	●
2022.01.08	██████████	●	●
2022.01.08	██████████	●	●
2022.01.11	██████████	●	●
2022.01.12	██████████	●	●
2022.01.13	██████████	●	●
2022.01.15	██████████	●	null
2022.01.17	██████████	●	●
2022.01.20	██████████	●	null
2022.01.20	██████████	●	●
2022.01.21	██████████	●	●

- **No Foreign keys need** so any activity can be related to another activity without data prep
- Runs during **analysis time**
- Cannot **duplicate or drop** rows
- All **SQL**-based so you can use the query in your transformation layer

Learn more at ActivitySchema.com

How does it work?

1. Filter by Customer

Give me all **primary activities** and **secondary activities** for name@example.com

TS	Customer	Activity
2022.01.02	██████████	●
2022.01.03	██████████	●
2022.01.04	██████████	●
2022.01.04	██████████	●
2022.01.05	██████████	●
2022.01.05	██████████	●
2022.01.05	██████████	●
2022.01.07	██████████	●
2022.01.07	██████████	●
2022.01.07	██████████	●
2022.01.07	██████████	●
2022.01.08	██████████	●

2. Filter by time

For each **primary activity** Look at all **first before secondary activity**

TS	Customer	Activity
2022.01.02	██████████	●
2022.01.03	██████████	●
2022.01.04	██████████	●
2022.01.04	██████████	●
2022.01.05	██████████	●
2022.01.05	██████████	●
2022.01.05	██████████	●
2022.01.07	██████████	●
2022.01.07	██████████	●
2022.01.07	██████████	●
2022.01.07	██████████	●
2022.01.08	██████████	●

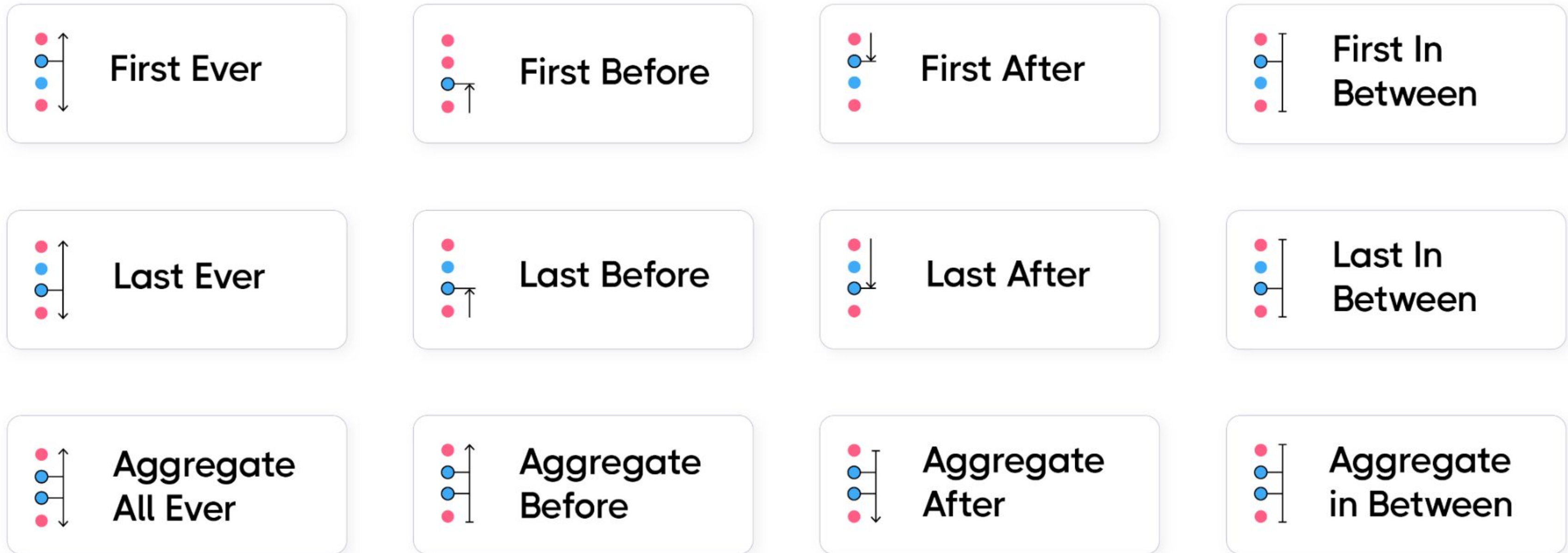
3. Pick one row + append

For each **primary activity** Append all **first before secondary activities**

TS	Customer	Primary	Append
2022.01.03	██████████	●	null
2022.01.04	██████████	●	●
2022.01.07	██████████	●	●
2022.01.07	██████████	●	●

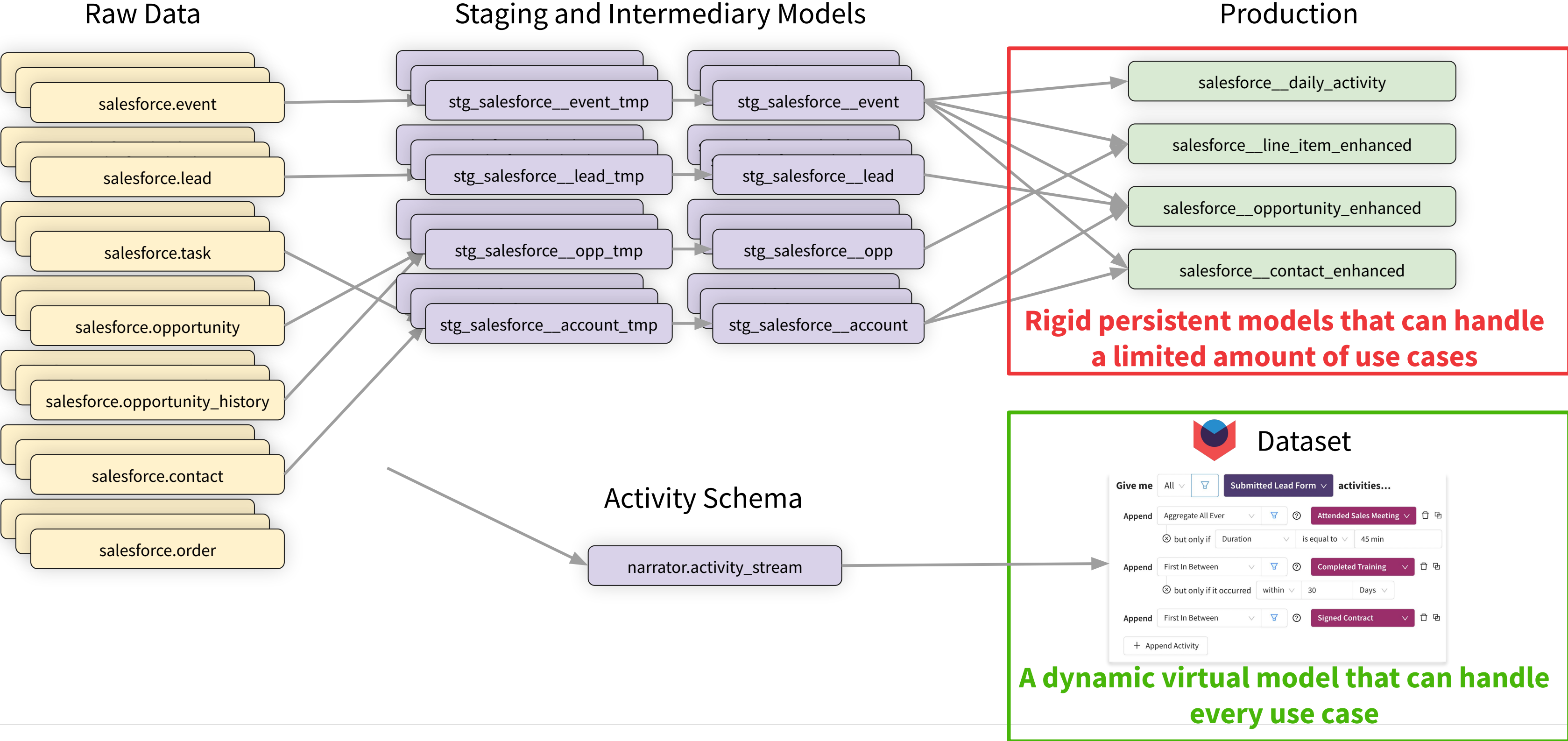
Learn more at ActivitySchema.com

There are only 12 Temporal Joins



Learn more at ActivitySchema.com

Data Modeling for EVERY use case



Demo

Using Narrator's Activity Schema Query tool

Demo

1. Show revenue orders over time
2. Add if the customer came from HOT PRODUCTS campaign
3. Change it to show the last campaign they came from
4. Add the Conversion Rate to next order
5. Show me the Repeat Rate by Last campaign by month
6. Change it to show the last campaign if within 3 days
7. Add total orders before and see if it impacts Repeat Rate
8. Add total emails before and see if that impacts Repeat Rate

https://portal-git-feature-sc-5652.dev.narrator.ai/narrator-demo/datasets/recently_viewed

Datasets

Search Datasets Create New

User	Name	Status	Tags	Viewed	Activities	Used by	Actions
	example	Private		a few seconds ago	Completed Order, Opened Email		
	example - VIZ	Private		an hour ago	Opened Email, Completed Order		
	Data council Demo	Private		11 hours ago	Completed Order, Opened Email		
	A/B Test - CTA Experiment <small>Dataset Auto-generated for Experiment Results: New CTA Experiment analysis</small>	Shared		2 days ago	Started Session, Completed Order	Narratives	
	demo	Private		2 days ago	Opened Email, Completed Order		
	test	Private		3 days ago	Opened Email, Completed Order		
	Oops	Private		9 days ago	Purchase Product		
	Customer LTV <small>Dataset Auto-generated for Customer Lifetime Value analysis</small>	Private		15 days ago	Completed Order	Narratives	
	example	Private		18 days ago	Opened Email, Completed Order		
	Email Conversion	Private		19 days ago	Opened Email, Completed Order	Materialized View	
	Diminishing Returns <small>Dataset Auto-generated for Point of Diminishing Returns - Viewed Webpages analysis</small>	Private		21 days ago	Viewed Webpage, Completed Order	Narratives	
	Opened Email,		

Thank you

Ahmed Elsamadisi