Analytics Pioneers

COMMUNITY TRAININGS

Mastering Data Lifecycles in BigQuery
- Deleting PII, Preserving Insights -

From the community for the community

Our Vision

Analytics Pioneers was founded to foster the regular exchange of ideas, challenges, and solutions within the digital analytics community. We want to bring the analytics community in Europe closer together and provide a platform where members can learn from each other and have fun.

From the community for the community

The community consists of experts who want to share their knowledge and newcomers to the industry or professionals who want to learn and exchange views on various issues.

Everyone can get involved: Whether as a „normal“ guest at one of our Meetups, with a small presentation/use case/discussion or as a host for one of our next events.

We welcome contributions of any kind, but sales pitches are not allowed!

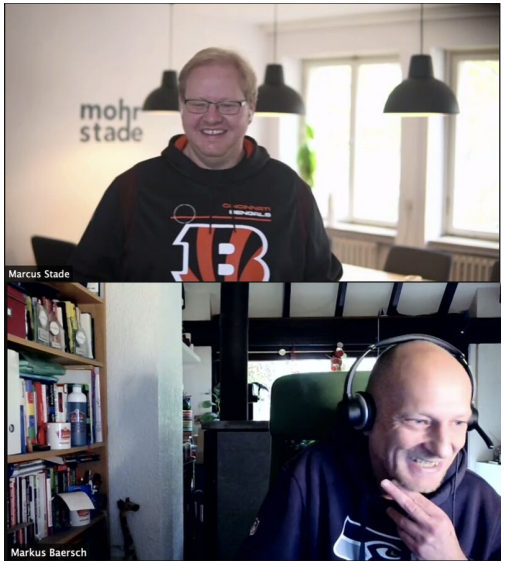
Analytics Pioneers

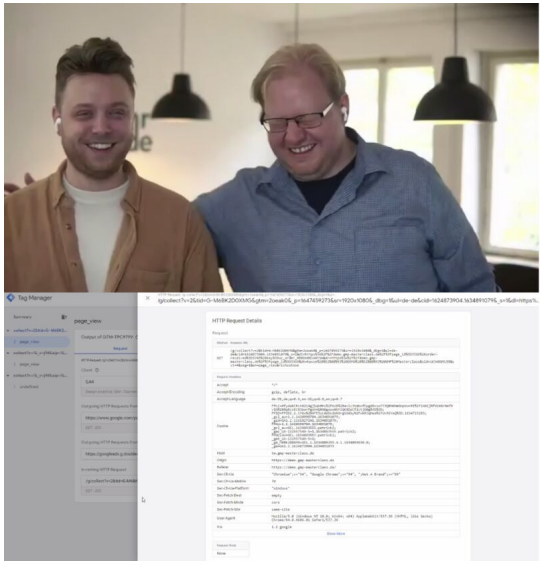
Analytics Pioneers group in 37 cities

Analytics Pioneers Community

Meetups

Community Trainings





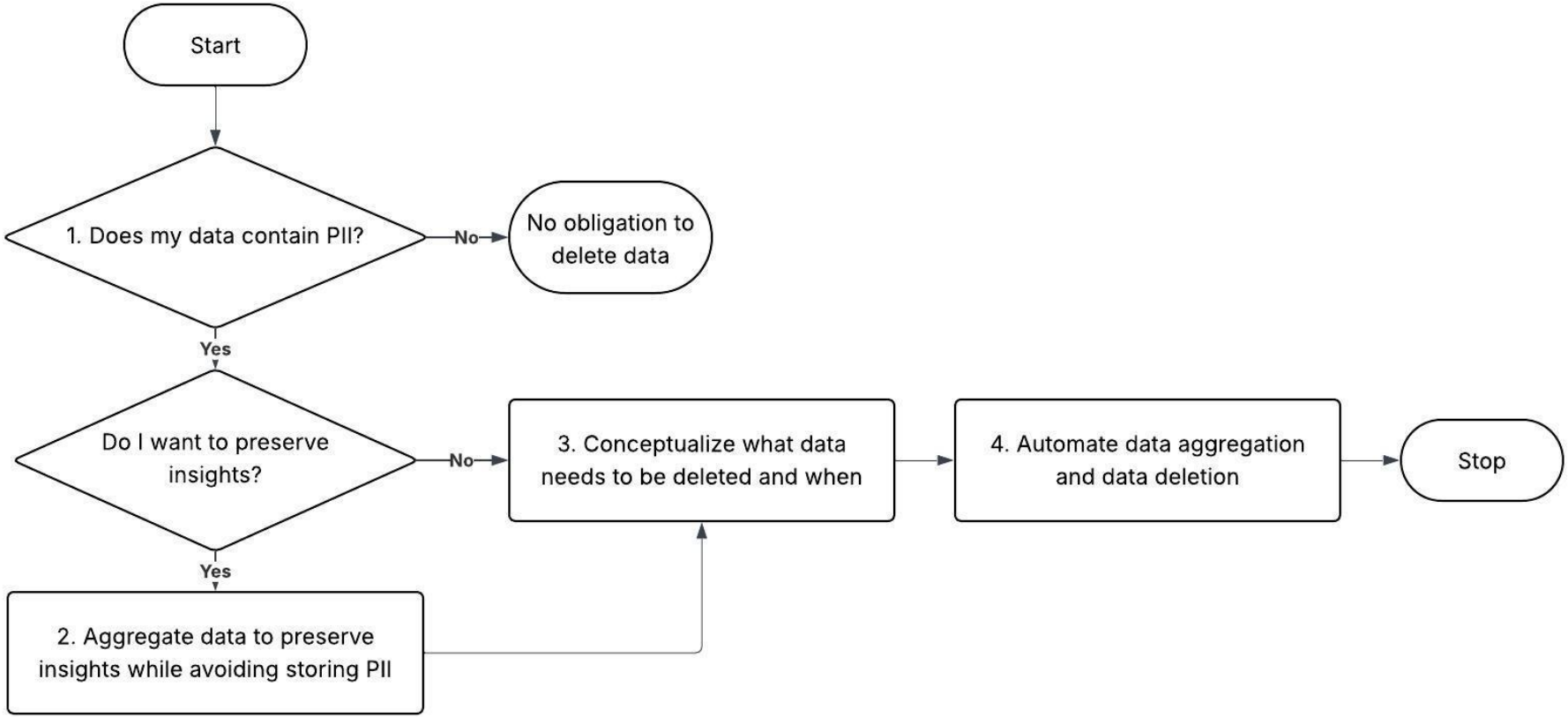
www.analytics-pioneers.com

Slides and Codes:

https://gitlab.com/mstade81/analytics-pioneers-community-trainings

Analytics Pioneers

Agenda



Agenda

1. Does my data contain PII?
2. Aggregate data to preserve insights while avoiding storing PII
3. Conceptualize what data needs to be deleted and when
4. Automate data aggregation and data deletion

Agenda

1. **Does my data contain PII?**
2. Aggregate data to preserve insights while avoiding storing PII
3. Conceptualize what data needs to be deleted and when
4. Automate data aggregation and data deletion

Disclaimer

This presentation is only used for knowledge sharing purposes and **does not serve as legal advice**. We are sharing technical Best Practices and no legally binding consultation. For legal questions and in case of uncertainty, please contact a qualified legal consultant.

Defining Personally Identifiable Information (PII)

PII includes all information that relates to an identified or identifiable **natural person**. This includes:

1. **Direct identifiers** such as name, address, email, or phone number,

2. **Indirect identifiers** such as IP address, cookies, location data, or device IDs,

3. **Particularly sensitive data**, such as health information, biometric or genetic data, which are subject to additional data protection requirements.

Detecting PII among Common Data Sources

Commonly Used Data Sources containing PII include:

- **CRM** Data (user_id, email, name, phone number, address, etc.) from Hubspot, Salesforce, etc.
- Google Analytics **raw** data (user_id, geo location, device info, etc.): analytics_*******
- Google Ads **gclid** info: ads_ClickStats
- Backend data - duh!

Agenda

1. Does my data contain PII?
2. **Aggregate data to preserve insights while avoiding storing PII**
3. Conceptualize what data needs to be deleted and when
4. Automate data aggregation and data deletion

What Data Needs Preservation?

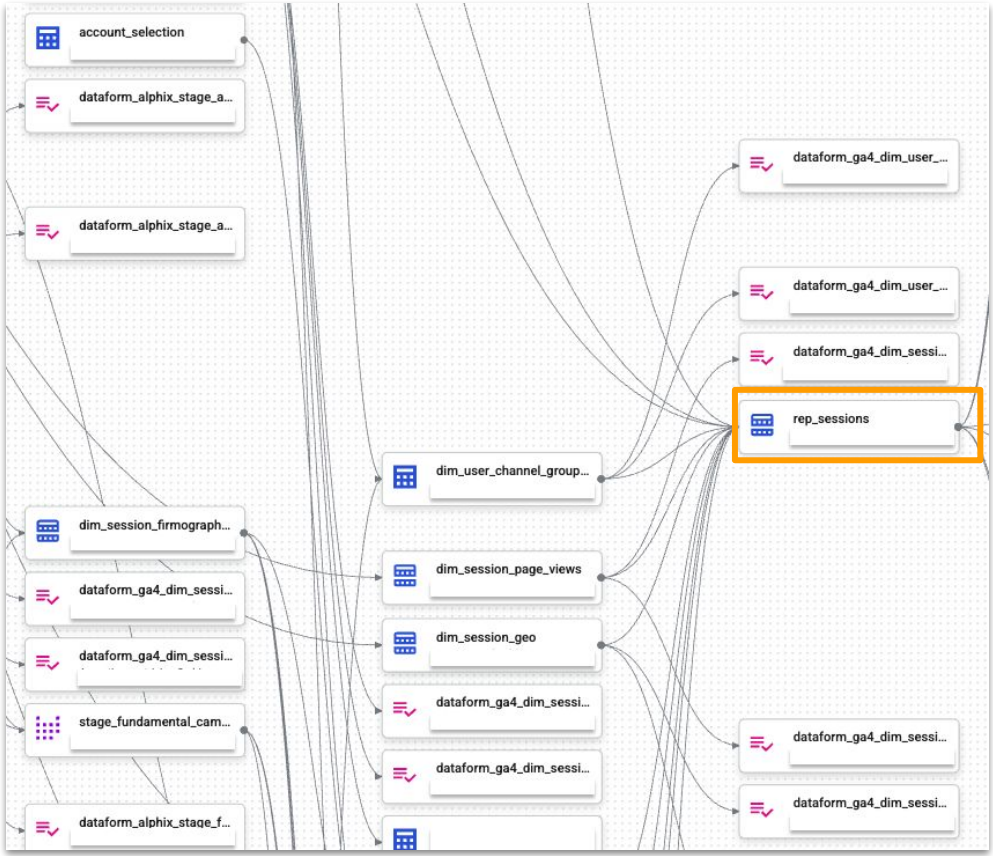
No session-, customer, or even click-id-level data after two years - no problem! Data of such granularity is probably **not even business-relevant** after that time.

It might, however, be relevant to preserve the trends:

- **Geo location**, e.g. “How has the traffic in Eastern Europe developed over time?”
- **Device info,** e.g. “How has the share of iOS traffic changed compared to two years ago?
- **Attribution**, e.g. “What channels are relevant now that were not relevant before?”

Analytics Pioneers

Aggregate Final Models Before Deleting PII Data



Identify the **“final” tables** in the pipeline. Do they contain PII data? If yes -> aggregate!

-> for details see our **github** repo

Tricks for the Aggregation Tables

```
rowConditions: [
    "number_of_sessions > 4",
    "number_of_users > 4"
]
```

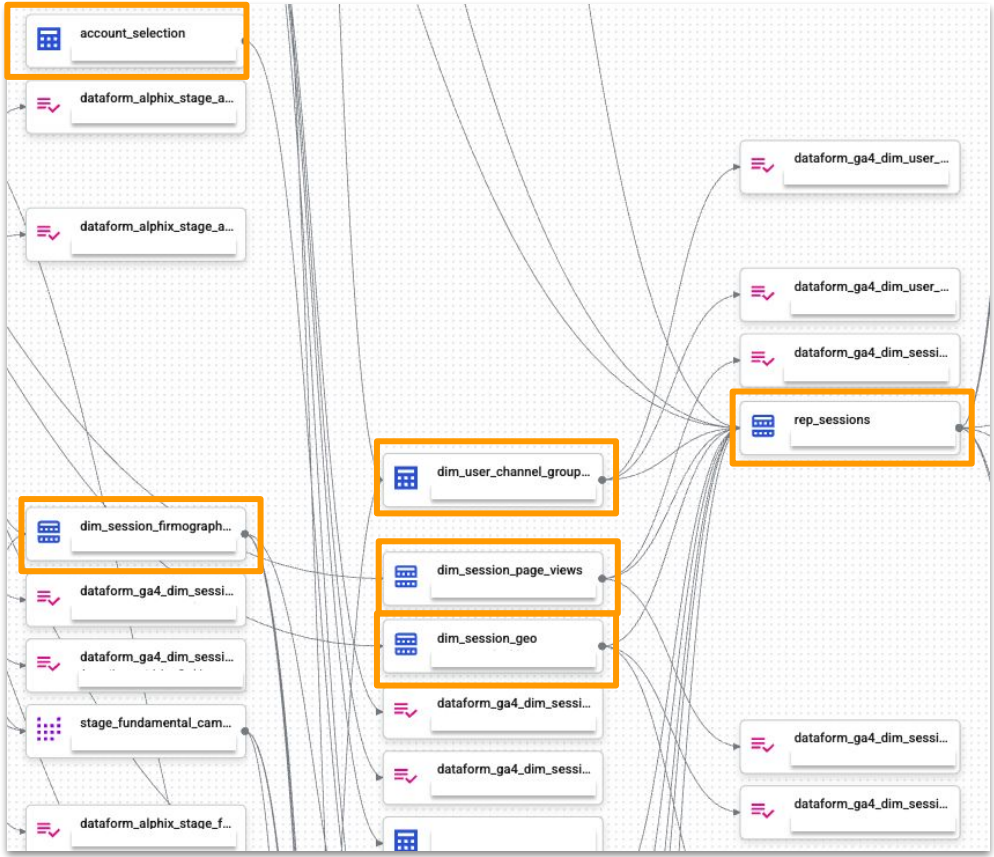
```
HAVING
    number_of_sessions > 4
AND number_of_users > 4
```

- **K-anonymity**! We don't report on categories that have fewer than a certain number of sessions/users (e.g. just one user of our website on January 1st living in Magdeburg)
- Select your **GROUP BY** wisely. If your categories are too specific, k-anonymity result in a lot of lost data
- If daily aggregation results in lost data (k-anonymity again), use **weekly or monthly** aggregation instead (e.g. session_week, session_month)

Agenda

1. Does my data contain PII?
2. Aggregate data to preserve insights while avoiding storing PII
3. **Conceptualize what data needs to be deleted and when**
4. Automate data aggregation and data deletion

What Data Needs To Be Deleted and When



- All tables containing **PII**
- Maximal lifetime guideline: **2 years**
- Depending on the data privacy policy: 2 years since entry has been saved VS 2 years since **last interaction with user**

Agenda

1. Does my data contain PII?
2. Aggregate data to preserve insights while avoiding storing PII
3. Conceptualize what data needs to be deleted and when
4. **Automate data aggregation and data deletion**

Automate Data Aggregation

- Create a separate workflow, or include the aggregated tables in an existing workflow, **executing regularly**
- Use table **partitioning & incrementality** to save compute costs
- **Prevent accidental table deletion** for aggregated tables using the keyword `protected:true`

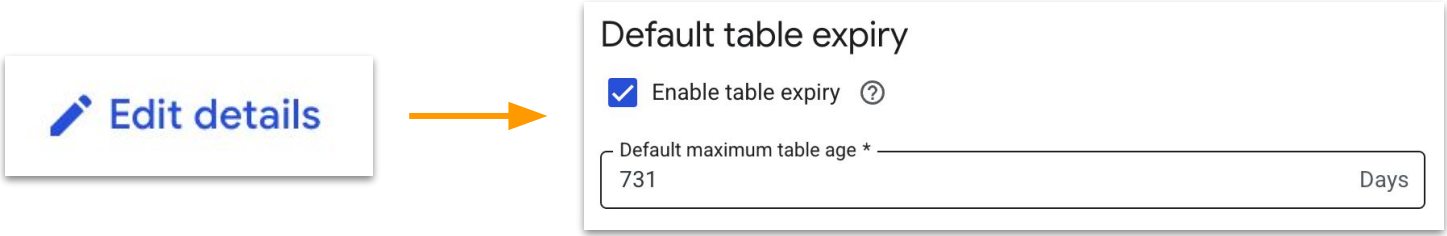
Automate Data Deletion

Table type	Partitioned
Partitioned by	DAY
Partitioned on field	session_date
Partition expiry	731 days
Partition filter	Not required

- Use **table partitioning** in all tables containing PII
- Make use of the **partition expiry** feature, setting it in Dataform directly, or with SQL queries

```
ALTER TABLE your_project.your_dataset.your_table
SET OPTIONS ( partition_expiration_days = 731 );
```

- For tables that only have data for one day, use **table expiry** instead. E.g. with the raw data of GA4:



Analytics Pioneers

Bonus: One Time Data Deletion for GA4 Raw Data

- After setting expiry date for all GA4 tables, only the **new tables** will get it automatically
- All existing tables in the GA4 dataset **need to be modified** to add an expiration date
- Tables with age >2 years can directly be **deleted (SQL Statement)**, but for the other ~731 Tables, a table expiration date needs to be set
- It is possible to set expiration dates one-by-one in the UI, but it is a lengthy and error-prone process (731 tables, 731 dates!)
- To easily set the expiration dates iteratively and without errors, use a **Python script within a BQ Notebook** (see our github repo!), or an **SQL Statement**

Thanks!

Moritz Bauer
Director Marketing
Technology
Follow me on LinkedIn

Marina Sukhanova
Consultant Marketing
Technology
Follow me on LinkedIn

Follow us on LinkedIn