

Best practice with BigQuery

NIKO SPIES

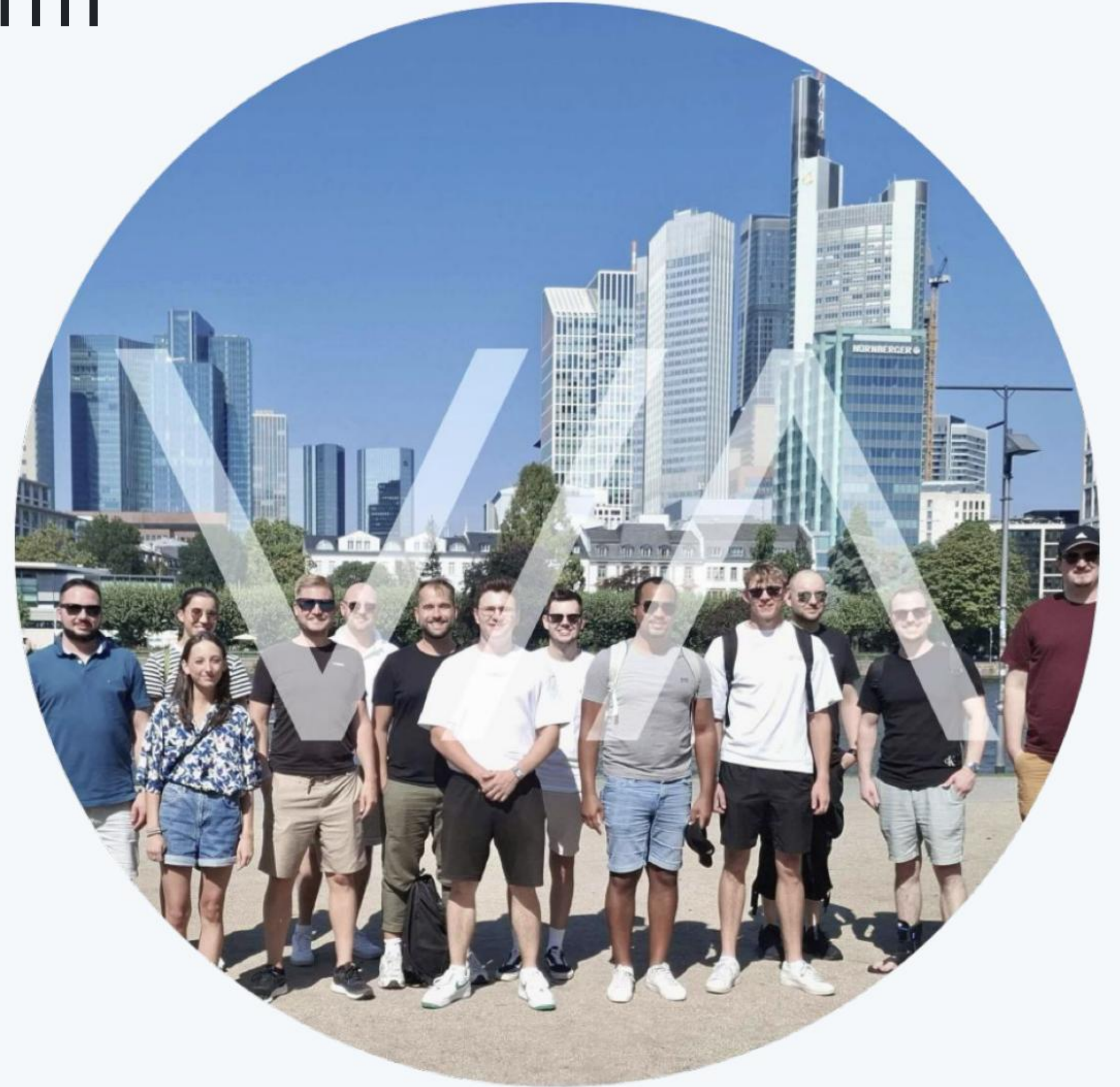
Mehr Umsatz für Online-Shops

- E-commerce entry in 2019 with your own shop
- CRO expert for startups and medium-sized companies
- Insights into >200 shops



Verify.io® - A/B Testing Platform

- ∞ Unlimited Traffic
- 🍪 No cookies
- 🎯 Single Source of Truth
- 📊 Shopify & Big Query integration



This is what it's about today:

- **Introduction: What is BigQuery anyway?**
- **Practical example: A/B testing**
- **How I work with BigQuery**
- **Tips & tools from practice**
- **Connection GA4 BigQuery**



introduction

What is BigQuery?

- **BigQuery is a cloud-based data warehouse from Google**
 - stores and analyses very large amounts of data (e.g. GA4 raw data)
- **Allows you to analyze large amounts of data (e.g. GA4 data) very quickly using SQL** • ask your own questions to the data – without restrictions due to predefined reports
- **Ideal for your own analyses, A/B testing and data-driven work**
 - Use for individual analyses, segmentations, test evaluations and reporting

How does data get into BigQuery?

- **Automatically exported from GA4**
 - via activated BigQuery link in the GA4 property settings
- **GA4 raw data export (automatic)**
 - the events recorded in GA4 are stored 1:1 as raw data in BigQuery • Example: a click on “Add to cart” is saved as a single event with timestamp, user ID, etc.
- **Additional own data sources possible**
 - e.g. CRM, shop, log files, APIs, CSV/Excel
- **Flexibly import your own data**
 - via API, file upload or schedule (scheduled jobs)

Events in BigQuery

	events_20250622	06-22 ▼	 Abfrag
Schema	Details	Vorschau	Tabellen-Exp
☐	user_pseudo_id		
☐	▶ privacy_info		
☐	▶ user_properties		
☐	user_first_touch_timestamp		
☐	▶ user_ltv		
☐	▶ device		
☐	▶ geo		
☐	▶ app_info		
☐	▶ traffic_source		
☐	stream_id		
☐	platform		
☐	▶ event_dimensions		
☐	▼ ecommerce		
☐	total_item_quantity		
☐	purchase_revenue_in_usd		
☐	purchase_revenue		
☐	refund_value_in_usd		
☐	refund_value		
☐	shipping_value_in_usd		
☐	shipping_value		
☐	tax_value_in_usd		
☐	tax_value		
☐	unique_items		
☐	transaction_id		
☐	▶ items	8	

Advantages and differences to GA4 data

- **Special tests and combinations possible** ÿ e.g.,
analyze users in connection with certain events or behavior patterns
- **Any length of time can be analyzed** ÿ in
BigQuery you store the data yourself — no 14-month limit like in GA4
- **No sampling problems**
ÿ All data is available, even for long test runs or small segments. No extrapolations.
- **Data secured & exportable**
ÿ Raw data permanently stored in BigQuery, downloadable at any time



Advantages and differences to GA4 data

Differences to GA4 directly:

- GA4 Standard storage: 2 to 14 months depending on the settings ÿ Funnel and exploration data (e.g. segments, behavior flows) are deleted after expiration
- BigQuery Sandbox: 60 days table lifetime — Upgrade for permanent storage possible (low Cost)
- BigQuery offers full data sovereignty ÿ no dependence on the retention settings in GA4. Data can be exported.



This is what Big Query costs

- **Free quota included** ÿ up to 10 GB storage and 1 TB queries per month free
- **Notifications for limits possible** ÿ automatic email warnings for storage or query limits can be set
- **Billing based on actual usage** ÿ only what is stored or queried is charged (no fixed price)
- **No fear of cost traps**
ÿ You can see the current usage in your Google Cloud account at any time



Query A/B Test & Dataset

- Test ran for 45 days
- Query 6.7 GB
- I could do this query 150 times to consume 1 TB

Abfrageergebnisse		
Jobinformationen		
Ergebnisse		
Dia		
ZEILE	total_users	users_with_trans...
1	10093	2414

Abfrageergebnisse

Ergebnisse speichern

Öffnen in

Jobinformationen

Ergebnisse

Diagramm

JSON

Ausführungsdetails

Ausführungsgrafik

Job-ID

Nutzer

nikospies1@gmail.com

Ort

EU

Erstellungszeit

23.06.2025, 15:36:56 UTC+2

Beginn

23.06.2025, 15:36:56 UTC+2

Ende

23.06.2025, 15:36:59 UTC+2

Dauer

2 Sek

Verarbeitete Byte

6,69 GB

In Rechnung gestellte Byte

6,69 GB

Example quota calculation

- Assumption: Shop with 100,000 users & 1,500 purchases per month
• approx. 10–15 events per user
• approx. 1.5 million events / month



x 100,000



x 1,500

- Data size: approx. 1 KB per event (GA4 events with standard data + custom dimensions)
• 1.5 million events × 1 KB = approx. 1.5 GB per month

Example quota calculation



x 100,000



x 1,500

- **Free quota: 10 GB storage**
- Result: after about 6–7 months the free tier (10 GB) would be reached
- After that: each additional GB costs only about 0.02 USD / month
 - Example: with 15 GB storage = approx. 0.10 € / month

Example quota calculation



x 100,000



x 1,500

- **Query quota: 1 TB per month free**
- Example: Query with 6.7 GB (see previous slide)
• approx. 150 queries per month free of charge
- In practice: with typical A/B tests and analyses you almost always stay in the free tier
• with 5 - 10 test evaluations per month • usually well below the 1 TB limit

Practical examples

What is SQL?

- **Language for data queries** ÿ
with SQL you can retrieve specific information from a database
- **In BigQuery, I use SQL to evaluate GA4 data** ÿ e.g., how often was something purchased, which user group performed better
- **Small commands – big impact**
ÿ With just a few lines of SQL, I can create my own reports and analyses

Basics for evaluating tests

- **GA4 tracking fully and cleanly set up**
 - e.g. Purchase event, all important events arrive in GA4 • Consent banner correctly integrated and set
- **Testing tool integrated and events recorded in GTM** •
 - the testing tool sends relevant events (e.g. “Testing Tool Event”) to GA4
- **Testing events contain unique information** • e.g.
 - “experiment_id” and “variant_name” in the event • so that the test groups can be evaluated cleanly later

Basics for evaluating tests

16 cookie_consent_update

15 Einwilligungsbefehl „upd...

14 Set

13 Set

12 Trigger-Gruppe

11 **verify**

10 Fenster geladen

9 DOM ist bereit.

8 Container geladen

7 template_load

6 Initialisierung

Ausgabe von

Tags

Variablen

Datenschicht

Datenschichtwerte nach dieser Nachricht:

1 {

2 developer_id: {dN2ZKMj: true, dY2Q2ZW: true},

3 gtm: {uniqueEventId: 877, priorityId: 2, start: 1750760063328},

4 event: "verify",

5 ads_data_redaction: false,

6 url_passthrough: false,

7 templateType: "index",

8 verify_experimentId: 21952,

9 verify_experimentName: "sidemenu-0011",

10 verify_variationId: null,

11 verify_variationName: "Original",

12 verify_abTest: "21952_sidemenu-0011:Original",

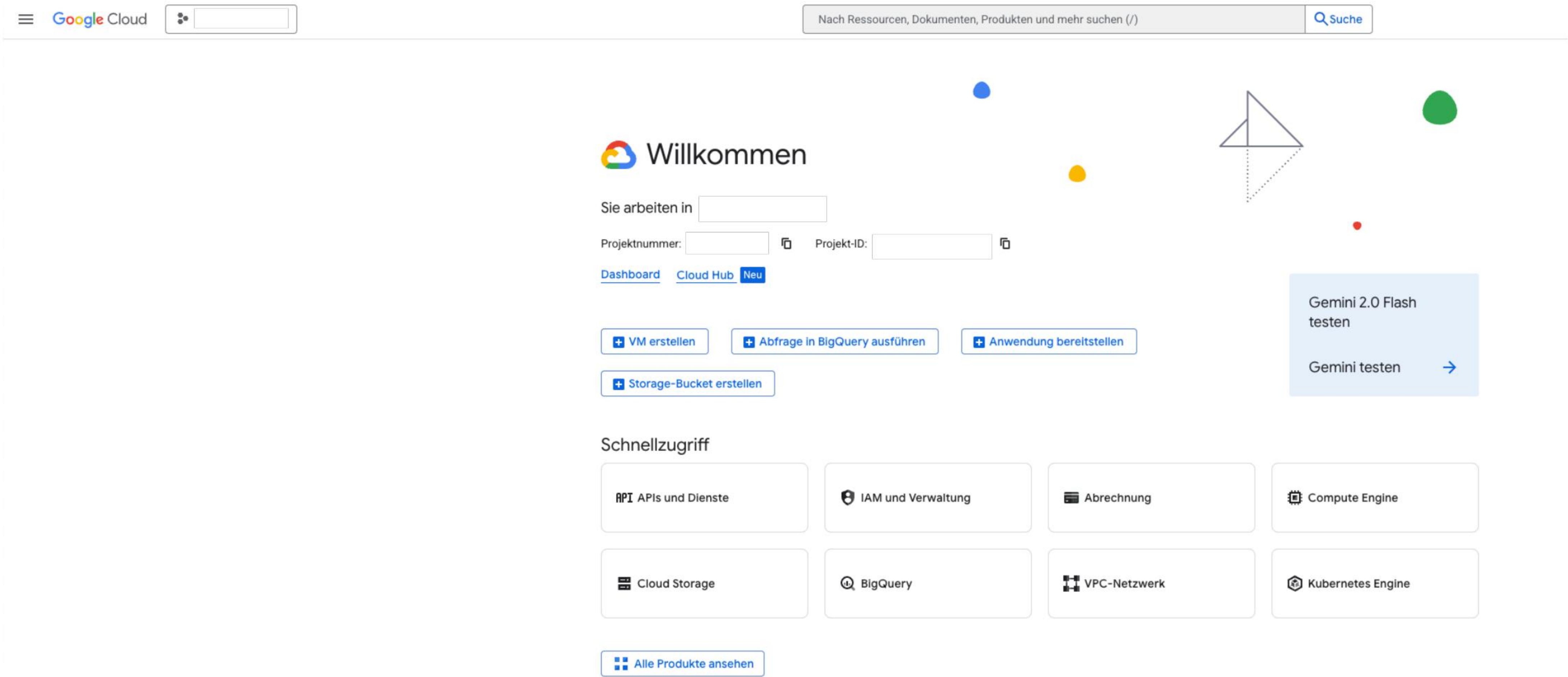
13 verify_abTestLong: "21952_sidemenu-0011:Original",

14 verify_abTestShort: "21952:Original",

15 verify_success: 1

16 }

Where can we query?



Where can we query?

Google Cloud

Cloud Hub

Cloud-Übersicht

Lösungen

Kürzlich aufgerufen

Neu

Angepinnte Produkte

APIs und Dienste

Marketplace

Compute Engine

Cloud Storage

VPC-Netzwerk

Cloud SQL

Sicherheit

Google Maps Platform

Abrechnung

Kubernetes Engine

BigQuery

Cloud Run

Produkte

IAM und Verwaltung

Vertex AI

Monitoring

Studio

Pipelines und Integration

Datenübertragung

Dataform

Geplante Abfragen

Planung

Governance

Sharing (Analytics Hub)

Richtlinien-Tags

Metadatenkuratierung

Verwaltung

Monitoring

Job-Explorer

Kapazitätsverwaltung

BI Engine

Notfallwiederherstellung

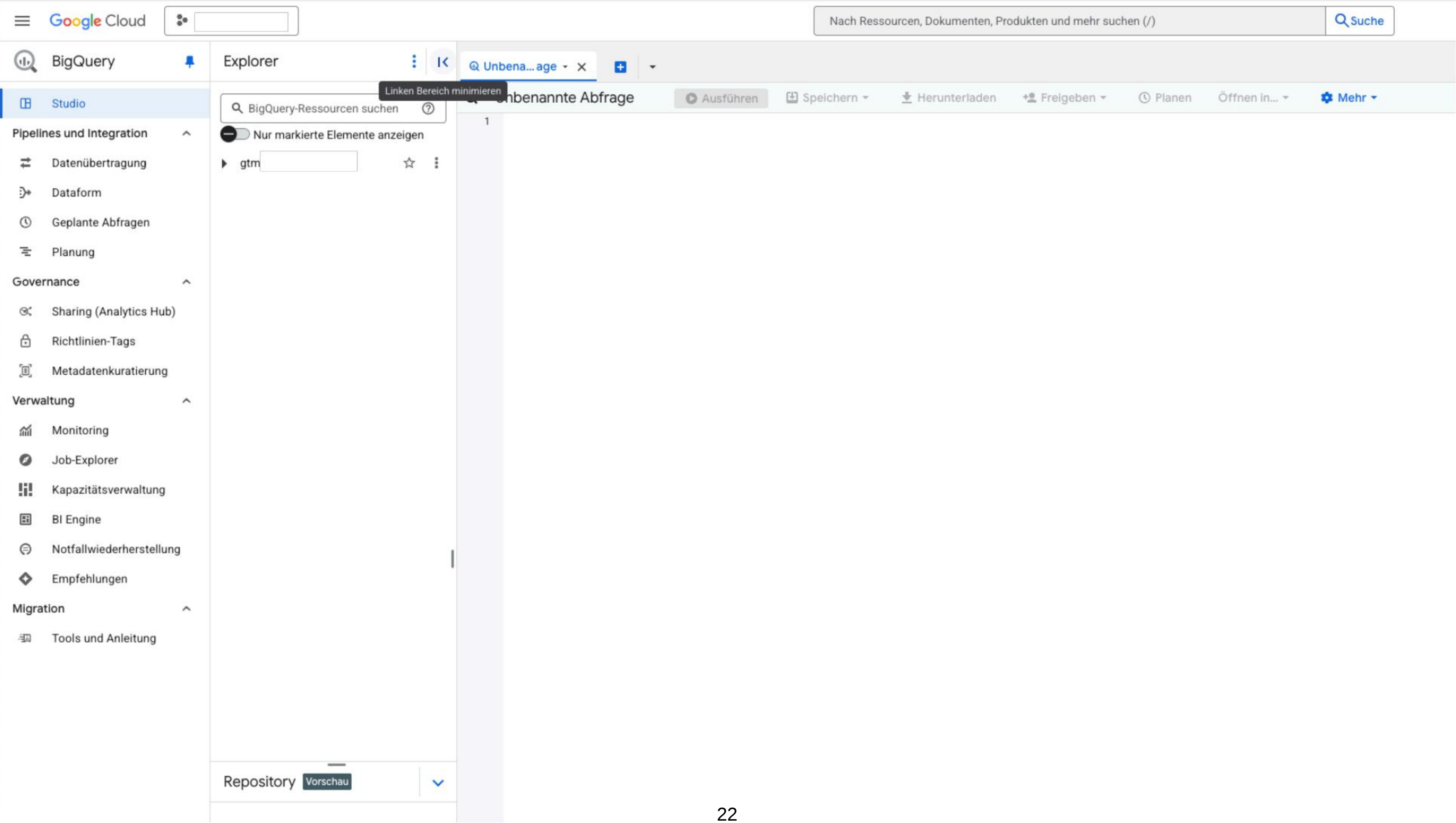
Empfehlungen

Migration

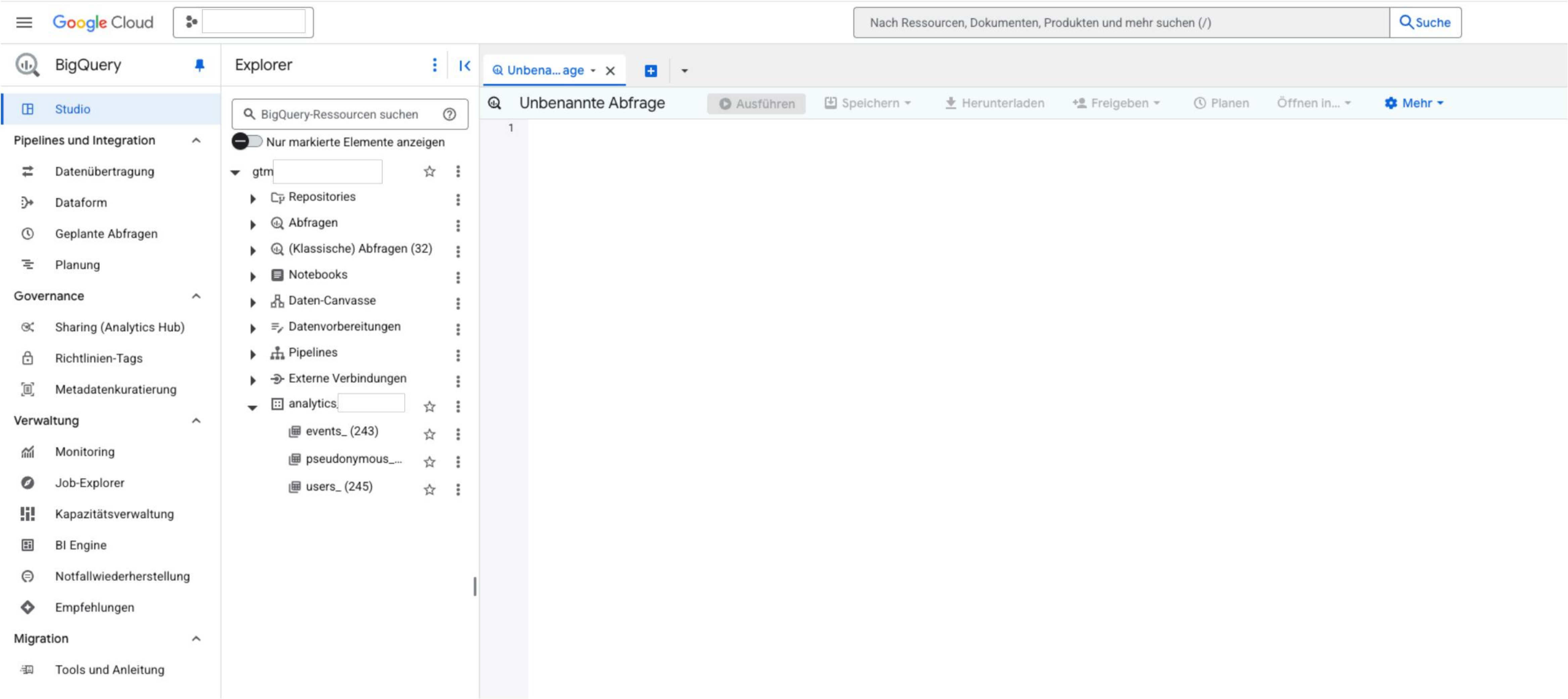
Tools und Anleitung

21

Where can we query?



Where can we query?



Where can we query?

BigQuery-Ressourcen suchen

Nur markierte Elemente anzeigen

gtm

Repositories

Abfragen

(Klassische) Abfragen (32)

Projektanfragen

2024-12-04 18:15:15

AA Test - Original

AA Test - Variante

BFCM Original

BFCM Variante

PUR Events

collection_001.1 - Kohorten

collection_001.1 - Original

collection_001.1 - Variante

collection_001.1 - Original

Ausführen

(Klassische) Abfrage speichern

Freigeben

Planen

```
1 SELECT
2   (SELECT COUNT(DISTINCT user_pseudo_id)
3     FROM `gtm`
4     WHERE event_name = 'Abtesting'
5           AND (SELECT value.string_value FROM UNNEST(event_params) WHERE key = 'verify_abTestShort') = '17145:Original'
6           AND _TABLE_SUFFIX BETWEEN '20250210' AND '20250325') AS total_users,
7   COUNT(DISTINCT user_pseudo_id) AS users_with_transactions,
8   ROUND(SUM(ecommerce.purchase_revenue), 2) AS Revenue,
9   ROUND((COUNT(DISTINCT user_pseudo_id) /
10    (SELECT COUNT(DISTINCT user_pseudo_id)
11      FROM `gtm`
12      WHERE event_name = 'Abtesting'
13            AND (SELECT value.string_value FROM UNNEST(event_params) WHERE key = 'verify_abTestShort') = '17145:Original'
14            AND _TABLE_SUFFIX BETWEEN '20250210' AND '20250325')) * 100, 2) AS CR,
15   ROUND(SUM(ecommerce.purchase_revenue) / COUNT(DISTINCT user_pseudo_id), 2) AS AOV,
16   ROUND(SUM(ecommerce.purchase_revenue) /
17     (SELECT COUNT(DISTINCT user_pseudo_id)
18       FROM `gtm`
19       WHERE event_name = 'Abtesting'
20             AND (SELECT value.string_value FROM UNNEST(event_params) WHERE key = 'verify_abTestShort') = '17145:Original'
21             AND _TABLE_SUFFIX BETWEEN '20250210' AND '20250325'), 2) AS RPU
22 FROM `gtm`
23 WHERE _TABLE_SUFFIX BETWEEN '20250210' AND '20250325'
24 AND ecommerce.purchase_revenue != 0
25 AND user_pseudo_id IN (
26   SELECT user_pseudo_id
27     FROM `gtm`
28     WHERE event_name = 'Abtesting'
29           AND (SELECT value.string_value FROM UNNEST(event_params) WHERE key = 'verify_abTestShort') = '17145:Original'
30           AND _TABLE_SUFFIX BETWEEN '20250210' AND '20250325');
```

Abfrage abgeschlossen

Abfrageergebnisse

Jobinformationen

Ergebnisse

Diagramm

JSON

Ausführungsdetails

Ausführungsgrafik

ZEILE	total_users	users_with_trans...	Revenue	CR	AOV	RPU
1	22819	745	67191.17	3.26	90.19	2.94

How I evaluate tests

```
SELECT
  (SELECT COUNT(DISTINCT user_pseudo_id)
   FROM `gtm-123456789.analytics_123456789.events_*` WHERE event_name
   = 'Abtesting'
   AND (SELECT value.string_value FROM UNNEST(event_params) WHERE key = 'verify_abTestShort') = '17145:Original'
   AND _TABLE_SUFFIX BETWEEN '20250210' AND '20250325') AS total_users,
  COUNT(DISTINCT user_pseudo_id) AS users_with_transactions,
  ROUND(SUM(ecommerce.purchase_revenue), 2) AS Revenue,
  ROUND((COUNT(DISTINCT user_pseudo_id) /
    (SELECT COUNT(DISTINCT user_pseudo_id)
     FROM `gtm-123456789.analytics_123456789.events_*` WHERE event_name
     = 'Abtesting'
     AND (SELECT value.string_value FROM UNNEST(event_params) WHERE key = 'verify_abTestShort') = '17145:Original'
     AND _TABLE_SUFFIX BETWEEN '20250210' AND '20250325')) * 100, 2) AS CR,
  ROUND(SUM(ecommerce.purchase_revenue) / COUNT(DISTINCT user_pseudo_id), 2) AS AOV,
  ROUND(SUM(ecommerce.purchase_revenue) /
    (SELECT COUNT(DISTINCT user_pseudo_id)
     FROM `TABLE ID` WHERE
     event_name = 'Abtesting'
     AND (SELECT value.string_value FROM UNNEST(event_params) WHERE key = 'verify_abTestShort') = '17145:Original'
     AND _TABLE_SUFFIX BETWEEN '20250210' AND '20250325'), 2) AS RPU FROM
`gtm-123456789.analytics_123456789.events_*` WHERE _TABLE_SUFFIX
BETWEEN '20250210' AND '20250325'
AND ecommerce.purchase_revenue != 0 AND
user_pseudo_id IN ( SELECT
  user_pseudo_id FROM
  `gtm-123456789.analytics_123456789.events_*` WHERE event_name =
  'Abtesting'
  AND (SELECT value.string_value FROM UNNEST(event_params) WHERE key = 'verify_abTestShort') = '17145:Original'
  AND _TABLE_SUFFIX BETWEEN '20250210' AND '20250325');
```

Abfrageergebnisse

Ergebnisse speichern

Jobinformationen

Ergebnisse

Diagramm

JSON

Ausführungsdetails

Ausführungsgrafik

ZEILE	total_users	users_with_trans...	Revenue	CR	AOV	RPU
1	22819	745	67191.17	3.26	90.19	2.94

How I evaluate tests

```
SELECT
(SELECT COUNT(DISTINCT user_pseudo_id)
FROM `TABLE ID`
```

BigQuery-Ressourcen suchen

Nur markierte Elemente anzeigen

▼

Repositories

Abfragen

(Klassische) Abfragen (32)

Notebooks

Daten-Canvasse

Datenvorbereitungen

Pipelines

Externe Verbindungen

▼

events_ (242)

pseudonymous_users_ (243)

users_ (244)

events_20250...

2025-06-22

Abfrage

Öffnen in

Freigeben

Schema

Details

Vorschau

Tabellen-Explorer

Vorschau

Statistiken

LI

Tabelleninformationen

Tabellen-ID	events_20250622
Erstellt	
Zuletzt geändert	
Tabellenablauf	NIE
Speicherort der Daten	
Standardsortierung	
Standard-Rundungsmodus	ROUNDING_MODE_UNSPECIFIED
Groß- und Kleinschreibung wird nicht berücksichtigt	false
Beschreibung	
Label	
Primärschlüssel	
Tags	

- We count all users in the test
- We filter the event source
- We filter the period
- We filter the test
- We filter the event

```
SELECT
  (SELECT COUNT(DISTINCT user_pseudo_id)
   FROM `gtm-123456789.analytics_123456789.events_*` WHERE
    event_name = 'Abtesting'
   AND (SELECT value.string_value FROM UNNEST(event_params) WHERE key = 'verify_abTestShort') = '17145:Original'
   AND _TABLE_SUFFIX BETWEEN '20250210' AND '20250325') AS total_users,

  COUNT(DISTINCT user_pseudo_id) AS users_with_transactions,

  ROUND(SUM(ecommerce.purchase_revenue), 2) AS Revenue,

  ROUND((COUNT(DISTINCT user_pseudo_id) /
   (SELECT COUNT(DISTINCT user_pseudo_id)
    FROM `gtm-123456789.analytics_123456789.events_*` WHERE
     event_name = 'Abtesting'
    AND (SELECT value.string_value FROM UNNEST(event_params) WHERE key = 'verify_abTestShort') = '17145:Original'
    AND _TABLE_SUFFIX BETWEEN '20250210' AND '20250325')) * 100, 2) AS CR,

  ROUND(SUM(ecommerce.purchase_revenue) / COUNT(DISTINCT user_pseudo_id), 2) AS AOV,

  ROUND(SUM(ecommerce.purchase_revenue) /
   (SELECT COUNT(DISTINCT user_pseudo_id)
    FROM `gtm-123456789.analytics_123456789.events_*` WHERE
     event_name = 'Abtesting'
    AND (SELECT value.string_value FROM UNNEST(event_params) WHERE key = 'verify_abTestShort') = '17145:Original'
    AND _TABLE_SUFFIX BETWEEN '20250210' AND '20250325'), 2) AS RPU

FROM `gtm-123456789.analytics_123456789.events_*`

WHERE _TABLE_SUFFIX BETWEEN '20250210' AND '20250325'
AND ecommerce.purchase_revenue != 0 AND
user_pseudo_id IN ( SELECT
  user_pseudo_id FROM
  `gtm-123456789.analytics_123456789.events_*` WHERE
   event_name = 'Abtesting'
   AND (SELECT value.string_value FROM UNNEST(event_params) WHERE key = 'verify_abTestShort') = '17145:Original'
   AND _TABLE_SUFFIX BETWEEN '20250210' AND '20250325');
```

total_users ÿ Number of all users in the test

- We count buyers

- We sum up sales

```
SELECT
  (SELECT COUNT(DISTINCT user_pseudo_id)
   FROM `gtm-123456789.analytics_123456789.events_*` WHERE
    event_name = 'Abtesting'
   AND (SELECT value.string_value FROM UNNEST(event_params) WHERE key = 'verify_abTestShort') = '17145:Original'
   AND _TABLE_SUFFIX BETWEEN '20250210' AND '20250325') AS total_users,

  COUNT(DISTINCT user_pseudo_id) AS users_with_transactions,

  ROUND(SUM(ecommerce.purchase_revenue), 2) AS Revenue,

  ROUND((COUNT(DISTINCT user_pseudo_id) /
    (SELECT COUNT(DISTINCT user_pseudo_id)
     FROM `gtm-123456789.analytics_123456789.events_*` WHERE
      event_name = 'Abtesting'
     AND (SELECT value.string_value FROM UNNEST(event_params) WHERE key = 'verify_abTestShort') = '17145:Original'
     AND _TABLE_SUFFIX BETWEEN '20250210' AND '20250325')) * 100, 2) AS CR,

  ROUND(SUM(ecommerce.purchase_revenue) / COUNT(DISTINCT user_pseudo_id), 2) AS AOV,

  ROUND(SUM(ecommerce.purchase_revenue) /
    (SELECT COUNT(DISTINCT user_pseudo_id)
     FROM `gtm-123456789.analytics_123456789.events_*` WHERE
      event_name = 'Abtesting'
     AND (SELECT value.string_value FROM UNNEST(event_params) WHERE key = 'verify_abTestShort') = '17145:Original'
     AND _TABLE_SUFFIX BETWEEN '20250210' AND '20250325'), 2) AS RPU

FROM `gtm-123456789.analytics_123456789.events_*`

WHERE _TABLE_SUFFIX BETWEEN '20250210' AND '20250325'
AND ecommerce.purchase_revenue != 0 AND
user_pseudo_id IN ( SELECT
  user_pseudo_id FROM
  `gtm-123456789.analytics_123456789.events_*` WHERE
  event_name = 'Abtesting'
  AND (SELECT value.string_value FROM UNNEST(event_params) WHERE key = 'verify_abTestShort') = '17145:Original'
  AND _TABLE_SUFFIX BETWEEN '20250210' AND '20250325');
```

users_with_transactions ÿ Number of buyers

purchase_revenue ÿ Total sales during the test period

- We calculate the CR
- We calculate the AOV

```
SELECT
  (SELECT COUNT(DISTINCT user_pseudo_id)
   FROM `gtm-123456789.analytics_123456789.events_*` WHERE
    event_name = 'Abtesting'
   AND (SELECT value.string_value FROM UNNEST(event_params) WHERE key = 'verify_abTestShort') = '17145:Original'
   AND _TABLE_SUFFIX BETWEEN '20250210' AND '20250325') AS total_users,

  COUNT(DISTINCT user_pseudo_id) AS users_with_transactions,

  ROUND(SUM(ecommerce.purchase_revenue), 2) AS Revenue,

  ROUND((COUNT(DISTINCT user_pseudo_id) /
   (SELECT COUNT(DISTINCT user_pseudo_id)
    FROM `gtm-123456789.analytics_123456789.events_*` WHERE
     event_name = 'Abtesting'
    AND (SELECT value.string_value FROM UNNEST(event_params) WHERE key = 'verify_abTestShort') = '17145:Original'
    AND _TABLE_SUFFIX BETWEEN '20250210' AND '20250325')) * 100, 2) AS CR,

  ROUND(SUM(ecommerce.purchase_revenue) / COUNT(DISTINCT user_pseudo_id), 2) AS AOV,

  ROUND(SUM(ecommerce.purchase_revenue) /
   (SELECT COUNT(DISTINCT user_pseudo_id)
    FROM `gtm-123456789.analytics_123456789.events_*` WHERE
     event_name = 'Abtesting'
    AND (SELECT value.string_value FROM UNNEST(event_params) WHERE key = 'verify_abTestShort') = '17145:Original'
    AND _TABLE_SUFFIX BETWEEN '20250210' AND '20250325'), 2) AS RPU

FROM `gtm-123456789.analytics_123456789.events_*`

WHERE _TABLE_SUFFIX BETWEEN '20250210' AND '20250325'
AND ecommerce.purchase_revenue != 0 AND
user_pseudo_id IN ( SELECT
  user_pseudo_id FROM
  `gtm-123456789.analytics_123456789.events_*` WHERE
  event_name = 'Abtesting'
  AND (SELECT value.string_value FROM UNNEST(event_params) WHERE key = 'verify_abTestShort') = '17145:Original'
  AND _TABLE_SUFFIX BETWEEN '20250210' AND '20250325');
```

AOV (Average Order Value) ÷ Sales divided by number of buyers (only users with purchases)

```
SELECT
  (SELECT COUNT(DISTINCT user_pseudo_id)
   FROM `gtm-123456789.analytics_123456789.events_*` WHERE
   event_name = 'Abtesting'
   AND (SELECT value.string_value FROM UNNEST(event_params) WHERE key = 'verify_abTestShort') = '17145:Original'
   AND _TABLE_SUFFIX BETWEEN '20250210' AND '20250325') AS total_users,

  COUNT(DISTINCT user_pseudo_id) AS users_with_transactions,

  ROUND(SUM(ecommerce.purchase_revenue), 2) AS Revenue,

  ROUND((COUNT(DISTINCT user_pseudo_id) /
   (SELECT COUNT(DISTINCT user_pseudo_id)
    FROM `gtm-123456789.analytics_123456789.events_*` WHERE
    event_name = 'Abtesting'
    AND (SELECT value.string_value FROM UNNEST(event_params) WHERE key = 'verify_abTestShort') = '17145:Original'
    AND _TABLE_SUFFIX BETWEEN '20250210' AND '20250325')) * 100, 2) AS CR,

  ROUND(SUM(ecommerce.purchase_revenue) / COUNT(DISTINCT user_pseudo_id), 2) AS AOV,
```

- We calculate the RPU

```
ROUND(SUM(ecommerce.purchase_revenue) /
 (SELECT COUNT(DISTINCT user_pseudo_id)
  FROM `gtm-123456789.analytics_123456789.events_*` WHERE
  event_name = 'Abtesting'
  AND (SELECT value.string_value FROM UNNEST(event_params) WHERE key = 'verify_abTestShort') = '17145:Original'
  AND _TABLE_SUFFIX BETWEEN '20250210' AND '20250325'), 2) AS RPU
```

```
FROM `gtm-123456789.analytics_123456789.events_*`

WHERE _TABLE_SUFFIX BETWEEN '20250210' AND '20250325'
AND ecommerce.purchase_revenue != 0 AND
user_pseudo_id IN ( SELECT
  user_pseudo_id FROM
  `gtm-123456789.analytics_123456789.events_*` WHERE
  event_name = 'Abtesting'
  AND (SELECT value.string_value FROM UNNEST(event_params) WHERE key = 'verify_abTestShort') = '17145:Original'
  AND _TABLE_SUFFIX BETWEEN '20250210' AND '20250325');
```

RPU (Revenue per User) ÷ Revenue divided by all users in the test (even without purchase)

```
SELECT
  (SELECT COUNT(DISTINCT user_pseudo_id)
   FROM `gtm-123456789.analytics_123456789.events_*` WHERE
   event_name = 'Abtesting'
   AND (SELECT value.string_value FROM UNNEST(event_params) WHERE key = 'verify_abTestShort') = '17145:Original'
   AND _TABLE_SUFFIX BETWEEN '20250210' AND '20250325') AS total_users,

  COUNT(DISTINCT user_pseudo_id) AS users_with_transactions,

  ROUND(SUM(ecommerce.purchase_revenue), 2) AS Revenue,

  ROUND((COUNT(DISTINCT user_pseudo_id) /
   (SELECT COUNT(DISTINCT user_pseudo_id)
    FROM `gtm-123456789.analytics_123456789.events_*` WHERE
    event_name = 'Abtesting'
    AND (SELECT value.string_value FROM UNNEST(event_params) WHERE key = 'verify_abTestShort') = '17145:Original'
    AND _TABLE_SUFFIX BETWEEN '20250210' AND '20250325')) * 100, 2) AS CR,

  ROUND(SUM(ecommerce.purchase_revenue) / COUNT(DISTINCT user_pseudo_id), 2) AS AOV,

  ROUND(SUM(ecommerce.purchase_revenue) /
   (SELECT COUNT(DISTINCT user_pseudo_id)
    FROM `gtm-123456789.analytics_123456789.events_*` WHERE
    event_name = 'Abtesting'
    AND (SELECT value.string_value FROM UNNEST(event_params) WHERE key = 'verify_abTestShort') = '17145:Original'
    AND _TABLE_SUFFIX BETWEEN '20250210' AND '20250325'), 2) AS RPU
```

```
FROM `gtm-123456789.analytics_123456789.events_*`

WHERE _TABLE_SUFFIX BETWEEN '20250210' AND '20250325'
AND ecommerce.purchase_revenue != 0 AND
user_pseudo_id IN ( SELECT
  user_pseudo_id FROM
  `gtm-123456789.analytics_123456789.events_*` WHERE
  event_name = 'Abtesting'
  AND (SELECT value.string_value FROM UNNEST(event_params) WHERE key = 'verify_abTestShort') = '17145:Original'
  AND _TABLE_SUFFIX BETWEEN '20250210' AND '20250325');
```

- We filter out 0€ purchases

Scripts for customization

```

SELECT
  (SELECT COUNT(DISTINCT user_pseudo_id)
   FROM `gtm-123456789.analytics_123456789.events_*` WHERE
    event_name = 'Abtesting'
    AND (SELECT value.string_value FROM UNNEST(event_params) WHERE key = 'verify_abTestShort') = '17145:Original'
    AND _TABLE_SUFFIX BETWEEN '20250210' AND '20250325') AS total_users,

  COUNT(DISTINCT user_pseudo_id) AS users_with_transactions,

  ROUND(SUM(ecommerce.purchase_revenue), 2) AS Revenue,

  ROUND((COUNT(DISTINCT user_pseudo_id) /
   (SELECT COUNT(DISTINCT user_pseudo_id)
    FROM `gtm-123456789.analytics_123456789.events_*` WHERE
     event_name = 'Abtesting'
     AND (SELECT value.string_value FROM UNNEST(event_params) WHERE key = 'verify_abTestShort') = '17145:Original'
     AND _TABLE_SUFFIX BETWEEN '20250210' AND '20250325')) * 100, 2) AS CR,

  ROUND(SUM(ecommerce.purchase_revenue) / COUNT(DISTINCT user_pseudo_id), 2) AS AOV,

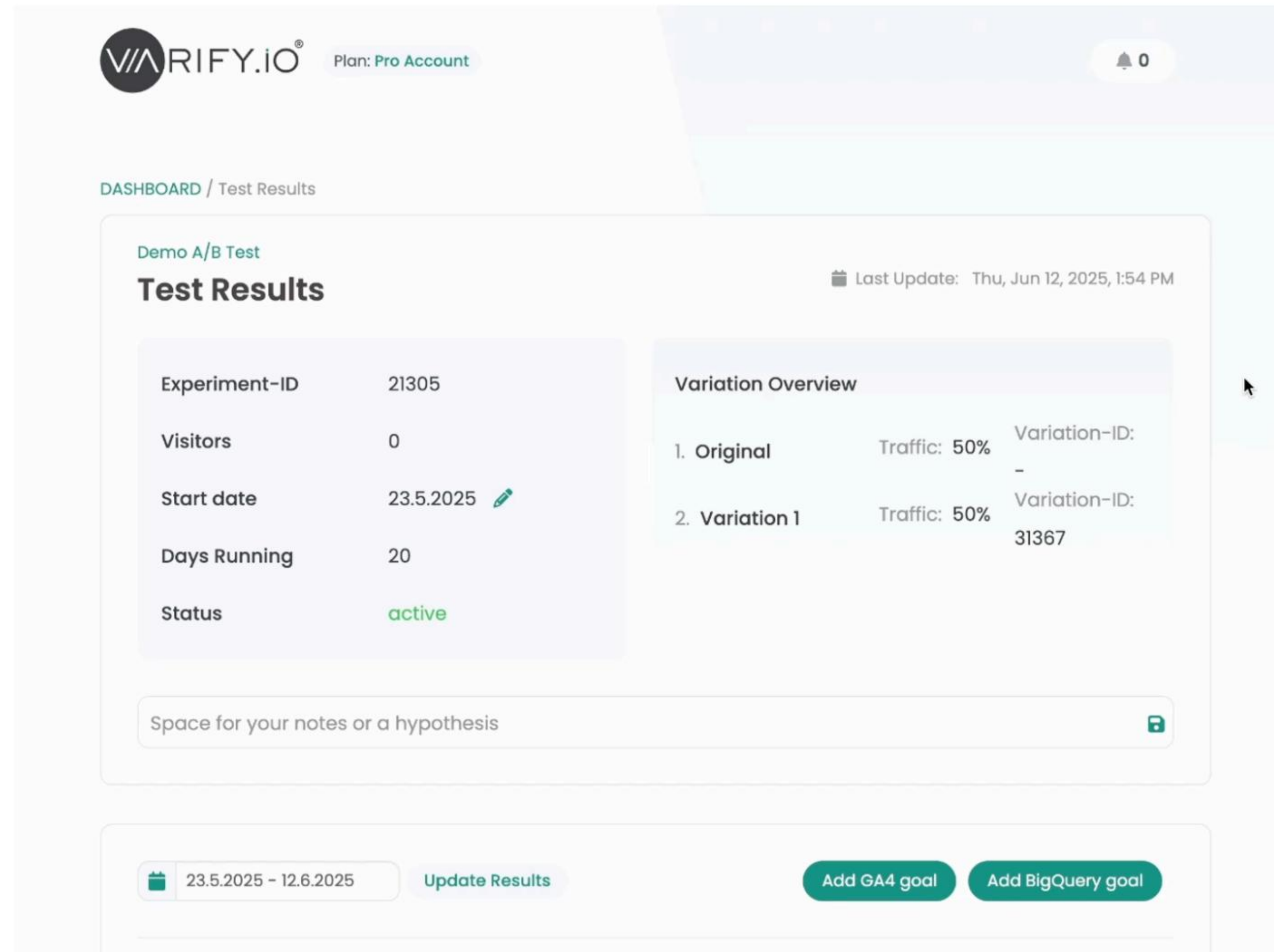
  ROUND(SUM(ecommerce.purchase_revenue) /
   (SELECT COUNT(DISTINCT user_pseudo_id)
    FROM `gtm-123456789.analytics_123456789.events_*` WHERE
     event_name = 'Abtesting'
     AND (SELECT value.string_value FROM UNNEST(event_params) WHERE key = 'verify_abTestShort') = '17145:Original'
     AND _TABLE_SUFFIX BETWEEN '20250210' AND '20250325'), 2) AS RPU

FROM `gtm-123456789.analytics_123456789.events_*`

WHERE _TABLE_SUFFIX BETWEEN '20250210' AND '20250325'
AND ecommerce.purchase_revenue != 0 AND
user_pseudo_id IN ( SELECT
  user_pseudo_id FROM
  `gtm-123456789.analytics_123456789.events_*` WHERE event_name
= 'Abtesting'
  AND (SELECT value.string_value FROM UNNEST(event_params) WHERE key = 'verify_abTestShort') = '17145:Original'
  AND _TABLE_SUFFIX BETWEEN '20250210' AND '20250325');

```

Example: Safe Space



No IT degree is required!

ChatGPT 4o 

 Gemein:

Bitte ergänze die SQL Abfrage und füge ganz unten eine weitere Bedingung ein, dass nutzer, die über 300€ bestellt haben nicht berücksichtigt werden

```
SELECT
(SELECT COUNT(DISTINCT user_pseudo_id)
FROM gtm-w8s7h6x2-
WHERE event_name = 'Abtesting'
AND (SELECT value.string_value FROM UNNEST(event_params)
WHERE key = 'verify_abTestShort') = '17145:Original'
AND _TABLE_SUFFIX BETWEEN '20250210' AND '20250325')
AS total_users,
COUNT(DISTINCT user_pseudo_id) AS users_with_transactions,
ROUND(SUM(ecommerce.purchase_revenue), 2) AS Revenue,
ROUND((COUNT(DISTINCT user_pseudo_id) /
(SELECT COUNT(DISTINCT user_pseudo_id)
FROM gtm-w8s7h6x2-
WHERE event_name = 'Abtesting'
AND (SELECT value.string_value FROM
UNNEST(event_params) WHERE key = 'verify_abTestShort') =
```

Stelle irgendeine Frage

  Tools

ChatGPT kann Fehler machen. Überprüfe wichtige Informationen. Siehe [Cookie-Voreinstellungen](#).

No IT degree is required!

Hier ist deine ergänzte Abfrage.

Ich habe ganz unten eine weitere Bedingung eingefügt, die nur Nutzer mit einer **Gesamtbestellsumme ≤ 300 €** berücksichtigt:

sqlKopierenBearbeiten

```
SELECT
  (SELECT COUNT(DISTINCT user_pseudo_id)
   FROM `ecommerce.events_*`
   WHERE event_name = 'Abtesting'
        AND (SELECT value.string_value FROM UNNEST(event_params) WHERE key = 'verify_abTestS'
              AND _TABLE_SUFFIX BETWEEN '20250210' AND '20250325') AS total_users,
  COUNT(DISTINCT user_pseudo_id) AS users_with_transactions,
  ROUND(SUM(ecommerce.purchase_revenue), 2) AS Revenue,
  ROUND((COUNT(DISTINCT user_pseudo_id) /
         (SELECT COUNT(DISTINCT user_pseudo_id)
          FROM `ecommerce.events_*`
          WHERE event_name = 'Abtesting'
               AND (SELECT value.string_value FROM UNNEST(event_params) WHERE key = 'verify_abTe'
                     AND _TABLE_SUFFIX BETWEEN '20250210' AND '20250325')) * 100, 2) AS CR,
  ROUND(SUM(ecommerce.purchase_revenue) / COUNT(DISTINCT user_pseudo_id), 2) AS AOV,
  ROUND(SUM(ecommerce.purchase_revenue) /
        (SELECT COUNT(DISTINCT user_pseudo_id,
```

No IT degree is required!

collection_001.1 - Original

Ausführen

(Klassische) Abfrage speichern

Freigeben

Planen

```
1 SELECT
2   (SELECT COUNT(DISTINCT user_pseudo_id)
3    FROM `gtm-`
4    WHERE event_name = 'Abtesting'
5          AND (SELECT value.string_value FROM UNNEST(event_params) WHERE key = 'verify_abTestShort') = '17145:Original'
6          AND _TABLE_SUFFIX BETWEEN '20250210' AND '20250325') AS total_users,
7   COUNT(DISTINCT user_pseudo_id) AS users_with_transactions,
8   ROUND(SUM(ecommerce.purchase_revenue), 2) AS Revenue,
9   ROUND((COUNT(DISTINCT user_pseudo_id) /
10    (SELECT COUNT(DISTINCT user_pseudo_id)
11     FROM `g-`
12     WHERE event_name = 'Abtesting'
13           AND (SELECT value.string_value FROM UNNEST(event_params) WHERE key = 'verify_abTestShort') = '17145:Original'
14           AND _TABLE_SUFFIX BETWEEN '20250210' AND '20250325')) * 100, 2) AS CR,
15   ROUND(SUM(ecommerce.purchase_revenue) / COUNT(DISTINCT user_pseudo_id), 2) AS AOV,
16   ROUND(SUM(ecommerce.purchase_revenue) /
17    (SELECT COUNT(DISTINCT user_pseudo_id)
18     FROM `g-`
19     WHERE event_name = 'Abtesting'
20           AND (SELECT value.string_value FROM UNNEST(event_params) WHERE key = 'verify_abTestShort') = '17145:Original'
21           AND _TABLE_SUFFIX BETWEEN '20250210' AND '20250325'), 2) AS RPU
22 FROM `gtm-`
23 WHERE _TABLE_SUFFIX BETWEEN '20250210' AND '20250325'
24 AND ecommerce.purchase_revenue != 0
25 AND user_pseudo_id IN (
26   SELECT user_pseudo_id
27   FROM `g-`
28   WHERE event_name = 'Abtesting'
29         AND (SELECT value.string_value FROM UNNEST(event_params) WHERE key = 'verify_abTestShort') = '17145:Original'
30         AND _TABLE_SUFFIX BETWEEN '20250210' AND '20250325');
31
```

✓ Diese Abfrage verarbeitet bei der Ausführung 2,02 GB.

Abfrageergebnisse

Jobinformationen	Ergebnisse	Diagramm	JSON	Ausführungsdetails	Ausführungsgrafik	
ZEILE	total_users	users_with_trans...	Revenue	CR	AOV	RPU
1	22819	745	67191.17	3.26	90.19	2.94

Unbenannte Abfrage

Ausführen

Speichern

Herunterladen

Freigeben

Planen

```
16 ROUND(SUM(ecommerce.purchase_revenue) /
17   (SELECT COUNT(DISTINCT user_pseudo_id)
18    FROM `g-`
19    WHERE event_name = 'Abtesting'
20          AND (SELECT value.string_value FROM UNNEST(event_params) WHERE key = 'verify_abTestShort') = '17145:Original'
21          AND _TABLE_SUFFIX BETWEEN '20250210' AND '20250325'), 2) AS RPU
22 FROM `g-`
23 WHERE _TABLE_SUFFIX BETWEEN '20250210' AND '20250325'
24 AND ecommerce.purchase_revenue != 0
25 AND user_pseudo_id IN (
26   SELECT user_pseudo_id
27   FROM `g-`
28   WHERE event_name = 'Abtesting'
29         AND (SELECT value.string_value FROM UNNEST(event_params) WHERE key = 'verify_abTestShort') = '17145:Original'
30         AND _TABLE_SUFFIX BETWEEN '20250210' AND '20250325'
31 )
32 AND user_pseudo_id IN (
33   SELECT user_pseudo_id
34   FROM (
35     SELECT user_pseudo_id, SUM(ecommerce.purchase_revenue) AS total_revenue
36     FROM `g-`
37     WHERE _TABLE_SUFFIX BETWEEN '20250210' AND '20250325'
38     AND ecommerce.purchase_revenue != 0
39     GROUP BY user_pseudo_id
40   )
41   WHERE total_revenue <= 300
42 );
43
```

✓ Abfrage abgeschlossen

Abfrageergebnisse

Jobinformationen		Ergebnisse	Diagramm	JSON	Ausführungsdetails	Ausführungsgrafik	
ZEILE	total_users	users_with_trans...	Revenue	CR	AOV	RPU	
1	22819	723	57764.1	3.17	79.9	2.53	

Use of events from the GTM

- In A/B tests, there are often important events that are forgotten in the evaluation
- Example: if I test in the sidecart, I need to know if the sidecart was even opened?
- Such events can be created via Google Tag Manager and sent to GA4
 - e.g. event “sidecart opened”
 - e.g. B. Event “all search open”
- In BigQuery I can use these events later in the query
 - as a filter or segment e.g. only evaluate users who have opened the sidecart
- Each event can be used as an additional filter
 - important: it must be laid out cleanly and fired correctly in the GTM

Use of events from the GTM

×

21 Sichtbarkeit von Elementen >

GA4 - Event - Sidecart geöffnet

✓

Ausgelöst

Tag-Details

Properties

Name	Wert
Typ	Google Analytics: GA4-Ereignis
Auslösestatus	Erfolgreich
E-Commerce-Daten senden	false
Include user-provided data from your website	false
Ereignisname	"sidecart_geöffnet"
Mess-ID	"ALT - GA4 - ID"

Use of events from the GTM

```
SELECT
  (SELECT COUNT(DISTINCT user_pseudo_id)
   FROM `gtm-123456789.analytics_123456789.events_*` WHERE
    event_Name = 'Abtesting'
   AND (SELECT value.string_value FROM UNNEST(event_params) WHERE key = 'verify_abTestShort') = '14767:Original'
   AND _TABLE_SUFFIX BETWEEN '20241212' AND '20250121'
  AND user_pseudo_id IN
    ( SELECT user_pseudo_id
      FROM `gtm-123456789.analytics_123456789.events_*` WHERE
        event_name = 'sidecart_open'
        AND _TABLE_SUFFIX BETWEEN '20241212' AND '20250121')) AS total_users,
```

My workflow

My workflow

1. Test planning



2. Test preparation



3. Test implementation



4. Control / Set up BigQuery queries



5. Evaluation of cohorts, segments, etc.



Special queries

SELECT

```

    user_pseudo_id,
    ROUND(SUM(ecommerce.purchase_revenue), 2) AS Purchase_Revenue, MAX(ecommerce.transaction_id)
    AS Transaction_ID, FORMAT_DATETIME('%T',
    DATETIME(TIMESTAMP_MICROS(event_timestamp), 'Europe/Berlin')) AS Berlin_Time FROM `cg-sky-12345.123456.events_*` WHERE _TABLE_SUFFIX BETWEEN
'20231128' AND '20231128'

    AND ecommerce.purchase_revenue != 0 AND
    ecommerce.purchase_revenue > 60 AND
    user_pseudo_id IN ( SELECT
        user_pseudo_id FROM `cg-
        sky-12345.123456.events_*` WHERE event_name
        = 'ablyft_abtest'
        AND (SELECT value.string_value FROM UNNEST(event_params) WHERE key = 'experiment_name') = 'test_cart_002'
        AND (SELECT value.string_value FROM UNNEST(event_params) WHERE key = 'variation_name') = 'Variation #1'
        AND _TABLE_SUFFIX BETWEEN '20231128' AND '20231128'
        AND user_pseudo_id IN
            ( SELECT user_pseudo_id
              FROM `cg-sky-12345.123456.events_*` WHERE
                event_name = 'sidecart_open'
                AND _TABLE_SUFFIX BETWEEN '20231128' AND '20231128'
            )
        )
    )
GROUP BY user_pseudo_id, event_timestamp
ORDER BY Berlin_Time;
```

Special queries

- Other software: ablyft
- Other parameters to query
- Shorter query because there are no calculations

Tips & tools from practice

ChatGPT as SQL Assistant

- I often use ChatGPT to help me write SQL queries
- ChatGPT helps with:
 - Syntax and structure of queries
 - Converting ideas into SQL
 - Optimizing and explaining existing queries
- Especially useful for more complex questions and filter logic
 - saves a lot of time when creating and customizing queries
- I have a BigQuery project and work with a GBT that is SQL professional

Looker Studio

- Often used to visualize BigQuery data in dashboards
- For A/B tests, you can, for example, build your own test dashboards
 • e.g. display purchases per variant over time
- I don't use it myself at the moment — because I do the analysis directly in BigQuery / SQL.
- Must be set so that data is not synchronized all the time!



Looker Studio

Looker Studio

Experiment ID: 11110 (1) ▾

Event Name: page_view (1) ▾

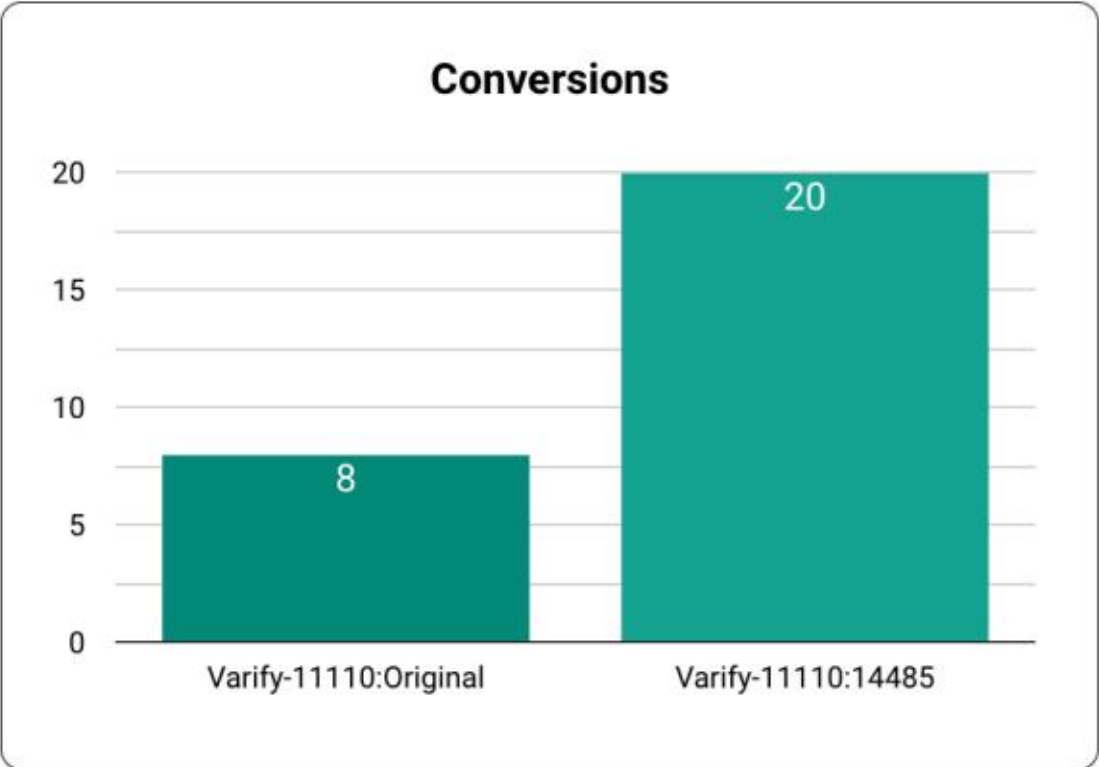
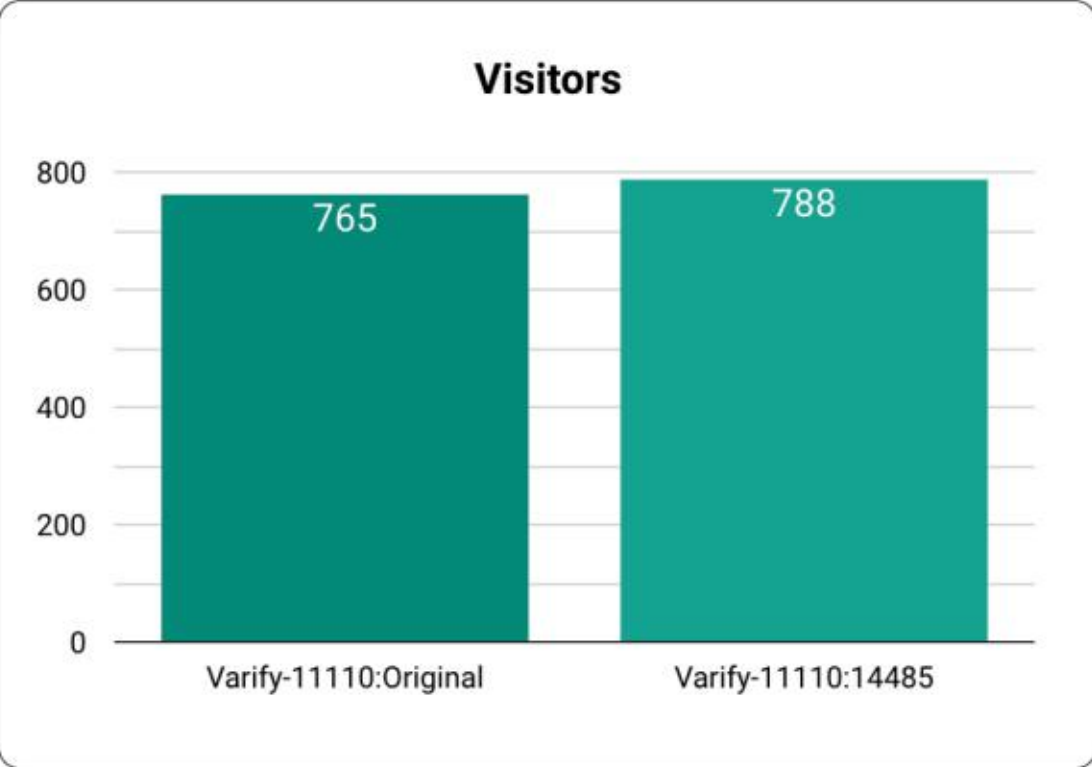
01.01.2024 - 31.12.2024 ▾

Varify.io BigQuery Looker Studio Dashboard



Varify experiment data is based on BigQuery data. The statistical calculation of significance is conducted using a two-sided Chi-Square test. The significance calculation is only valid if the number of conversions is less than the number of visitors. It is not applicable for revenue values. Improvements and significance calculations are always made in comparison to the original.

		Visitors	Conversions	Conversion Rate	Improvement	Confidence	Significance
1.	Varify-11110:Original	765	8	1.05%	-	-	Nicht signifikant
2.	Varify-11110:14485	788	20	2.54%	142.70%	97%	Signifikant (p < 0....



Safe spaces

	Visitors	Conversions	Conversion Rate	Improvement	Confidence	Significant
Original	84,895	1,730	2.04%	-	-	-
Variation 1	84,897	1,790	2.11%	4.46	83.8%	No
 No statistically significant differences were found. Please continue testing to ensure a sufficient sample size.						

Goal 3: Add to carts					GA4	⋮
	Visitors	Conversions	Conversion Rate	Improvement	Confidence	Significant
Original	84,698	6,314	7.45%	-	-	-
Variation 1	84,717	6,754	7.97%	6.94%	100.0%	Yes

Goal 4: Add to carts (count)					Big Query	⋮
	Visitors	Conversions	Conversion Rate	Improvement	Confidence	Significant
Original	84,895	6,340	7.47%	-	-	-
Variation 1	84,897	6,790	7.99%	7.27%	100.0%	Yes

Goal 5: Abtesting					GA4	⋮
	Visitors	Event Count	Conversion Rate	Improvement	Confidence	Significant
Original	84,698	84,662	99.96%	-	-	-
Variation 1	84,717	84,687	99.96%	0.01%	77.0%	No
 No statistically significant differences were found. Please continue testing to ensure a sufficient sample size.						



BigQuery Access

Connecting GA4 & BigQuery

Linking GA4 & BigQuery

Produktverknüpfungen

Mit diesen Einstellungen können Sie festlegen, welche Produkte mit dieser Property verknüpft werden



Google AdSense-Verknüpfungen



Google Ads-Verknüpfungen



Ad Manager-Verknüpfungen



BigQuery-Verknüpfungen



Display & Video 360 – Verknüpfungen



Floodlight-Verknüpfungen



Merchant Center-Verknüpfungen



Google Play-Verknüpfungen



Search Ads 360-Verknüpfungen



Search Console-Verknüpfungen



BigQuery-Verknüpfungen

Suchen

Verknüpfen

Projekt-ID	Projektname	Projektnummer
------------	-------------	---------------

Linking GA4 & BigQuery

Details zur erstellten Verknüpfung

Projekt-ID

Projektname

Projektnummer

Standardort für Datenpoolerstellung ⓘ

Europäische Union (eu)

Erstellt von

Erstellungsdatum

07.05.2025

Datenkonfigurationen

Ereignisdaten

Datenstreams und Ereignisse

Legen Sie fest, welche Datenstreams und Ereignisse exportiert werden sollen. Alle Ereignismengen sind Schätzwerte. Ob das Tageslimit durchgesetzt wird, hängt von der tatsächlich exportierten Menge ab. [Weitere Informationen](#)

GESCHÄTZTE GESAMTZAHL DER TÄGLICHEN ereignisse, die zu exportieren sind

0,02 / 1 Million(en) (Tageslimit) ⓘ

2 von 2 Streams ausgewählt

Keine Ereignisse ausgeschlossen

[Datenstreams und Ereignisse konfigurieren](#)

☐ Werbe-IDs für App-Streams einbeziehen

Exporttyp

☒ Täglich

Die Daten werden einmal täglich komplett exportiert

☐ Streaming (bestmöglich)

Fortlaufender Export, innerhalb von Sekunden nach Ereigniseintritt. Daten werden bestmöglich exportiert, ohne Garantie für Vollständigkeit. [Weitere Informationen](#)

Nutzerdaten

Alle Nutzer mit Aktivitäten für den aktuellen Tag werden exportiert, wenn sich eines der [Attribute des Nutzers](#) geändert hat. Der Export von Nutzerdaten wird pausiert, wenn mehr Ereignisdaten exportiert werden, als das Limit erlaubt. [Weitere Informationen zum Exportieren von BigQuery-Nutzerdaten](#)

Exporttyp

☒ Täglich

Die Daten werden einmal täglich komplett exportiert

Selected daily to avoid streaming large amounts of data all the time!

51

Or via varify.io via auto setup

Analytics

Google Analytics 4

Data Transfer

Manual

Test Evaluation

GA4 and Varify

Consent Mode

Per Consent

Google Authenticator Status: ● connected

Disconnect

Account

Property

BJJ Grappling

BJJ-Grappling.de - GA4

 Use Bigquery Data ● activated

Off 

Save

Privacy

Privacy Settings

Next Steps

- Compare data for the single source of truth
- Try out simple SQL scripts
- Exchange with experts
- Try BigQuery for free via varify.io
(As of July 31)
- Member of our (really) exclusive Slack
Become a community

Mehr Umsatz für deinen Online-Shop
Fragen zum Webinar? Lass uns sprechen!



Niko Spies
CRO Expert

Steffen Schulz
CEO Varify.io

