# Application Help for SAP Emarsys

Generated on: 2025-07-28 08:25:21 GMT+0000

SAP Emarsys | Latest

**Public**

Original content: https://help.sap.com/docs/SAP_EMARSYS/f8e2fafeea804018a954a8857d9dfff3?locale=en-US&state=PRODUCTION&version=CLOUD

## Warning

This document has been generated from SAP Help Portal and is an incomplete version of the official SAP product documentation. The information included in custom documentation may not reflect the arrangement of topics in SAP Help Portal, and may be missing important aspects and/or correlations to other topics. For this reason, it is not for production use.

For more information, please visit https://help.sap.com/docs/disclaimer.

# Relational Data

The SAP Emarsys Relational Data Service (RDS) enables you to load your business-specific data to the SAP Emarsys Platform, in order to use it for segmentation and message personalization purposes.

### ℹ Note

Relational Data is found under **Add-ons** > **Relational Data**.

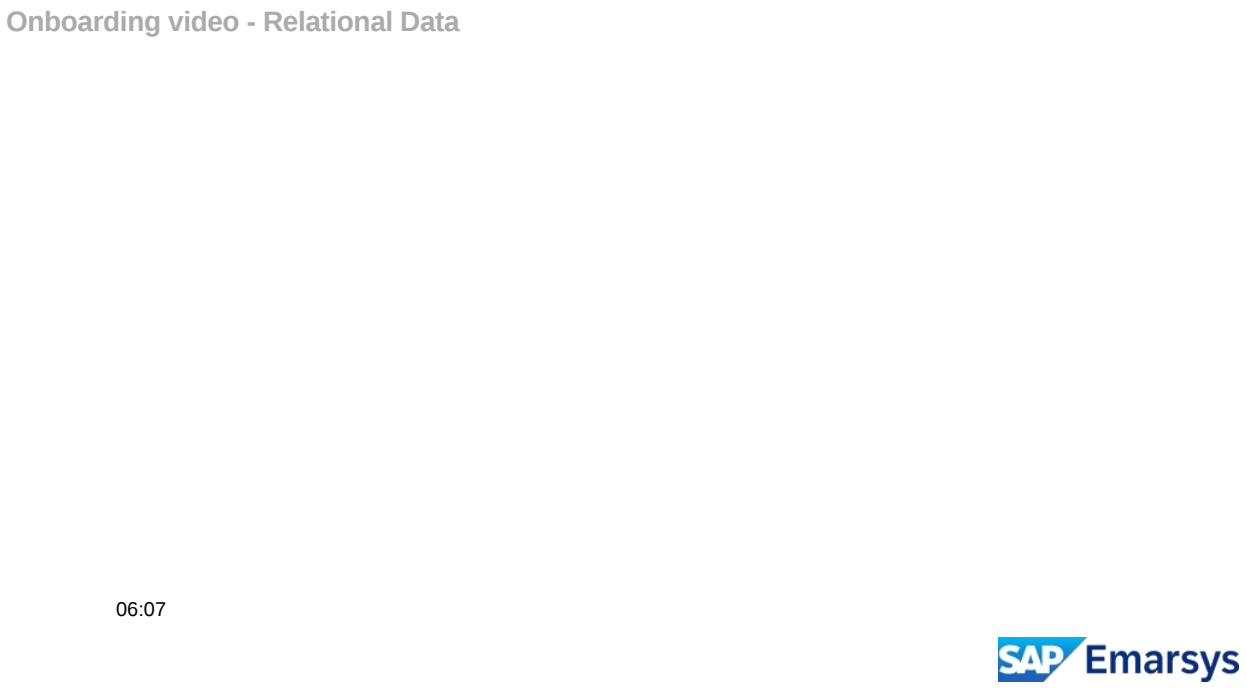
**Product documentation**

- Preparing your hosted database for Relational Data
- Creating relational segments
- Defining Relational Data reference fields
- Running personalization queries on your database
- Troubleshooting your Relational Data setup

### → Tip

As a quick reminder, here is the onboarding video about **Relational Data:**

Disclaimer: The below video is not part of the SAP product documentation. Please read the legal disclaimer for video links before viewing this video.




### ℹ Note

Have a look at a brief and portable summary of this add-on along with some relevant customer testimonies.

## What is Relational Data?

Relational Data allows you to connect your own database containing any kind of business-specific data and make it usable in the SAP Emarsys platform without duplicating the data across. In a relational database structure, you can combine different data sources to make complex data accessible for use in segmentation and personalization.

### Segment definition

You can create segment definitions by connecting different data tables in the core of the system and providing flexible segmentation.

The flexible Segment Template lets you choose what data points to segment on across any of the available data in your database, which can then be used in the SAP Emarsys platform.

Instead of storing the data, Relational Data connects to other databases on execution and only pulling the data that is used, making the whole process compliant with the strictest data privacy regulations.

It provides hundreds of segmentation scenarios for marketers without imposing any IT barriers. For detailed information, see Relational Segments.

### Personalization

Relational Data also allows you to use any of your data to enrich your personalization capabilities with personalization tokens you can integrate into campaigns as easily as any of the other SAP Emarsys data. For detailed information, see Defining Relational Data reference fields.

## Examples: Use cases

**Reminder for retention**

*Contract renewal with incentives*

When a contract is about to expire, an SMS or email will be sent with the focus on a new mobile phone. This is an aggressive campaign with big discounts.

**Upsell**

*An airline wants to send offers for activities and accommodation*

Between the date of booking and before the actual flight date, a campaign is sent with personalized offers for products or activities to buy, and suggestions for accommodation, based on the destination city. This is an automated campaigns sent on a daily basis.

**Abandoned cart**

*E-commerce website brings customers back to complete unfinished purchases*

A web shop is targeting customers who have added products to their cart but have not finished their purchase at the payment step. The campaign is to be sent every six hours. There is a link to the checkout page included to let the customer continue the purchase where they left off. The personalized subject line shows the most expensive product.

**What database connectors are supported?**

Relational Data connects to databases outside of the SAP Emarsys ecosystem, and these databases need to be provided by the customers themselves.

Currently Relational Data has connectors for the following database types:

- Amazon Redshift
- Azure SQL
- Google BigQuery
- Microsoft SQL
- MySQL
- PostgreSQL
- SAP HANA Cloud
- SAP HANA On-Premise
- Snowflake

For more information on how to get started please see Preparing your hosted database for Relational Data.

## Related Information

Relational Data Use Cases
End-user Guides
Troubleshooting

# Checklist for Setting Up Relational Data Manually

This article contains tables detailing all the steps that are necessary for Relational Data to run smoothly in your account. It has been compiled for the person or team responsible for setting up and configuring Relational Data.

We have divided the things to check into three categories:

- Minimum requirements
- Recommended items

By checking off the items in this list you can ensure that Relational Data is set up properly and will be working as intended.

Although the tables below contain items that are not mandatory, best results are achieved if all of these are properly set up.

## Minimum Requirements

These are the requirements that are absolutely necessary for Relational Data to function. Without these, the system can work erratically. These should all be checked off before the system goes into production.

| TO CHECK | HOW TO CHECK? | LINK TO DOCUMENTATION |
|---|---|---|
| **GENERAL** | | |
| • Is Relational Data available in the account (menu item available on the UI)?<br>• Is permission set for the admins who want to use the feature? | • Check if the menu item **Add-ons** > **Relational Data** is present in the account.<br>• Check under **User Management** if admins who will use Relational Data have the required permission. | Learn more about:<br>• Onboarding video: Relational Data<br>• Learn more: User Roles and Permission |
| • Does a connection to a Relational Data exist? | Check under **Add-ons** > **Relational Data** > **Connections** if a connection is available. | |

| TO CHECK | HOW TO CHECK? | LINK TO DOCUMENTATION |
|---|---|---|
| **DATA UPLOAD** | | |
| Do you have an API user that you can use? | Test if they can insert / update / delete entry in the Relational Data database. | Learn more about:<br>• Data exchange resources<br>• Relational Data API end-points |
| **RELATIONAL SEGMENT CREATION** | | |
| Can you create Relational Segments? | Under **Contacts** > **Segments** > **Relational Segments** (Relational Segment template will be needed to run any segment). | Learn more |
| **SELF SERVICE SETUP** | | |
| Check if menu item exists and displays data: **Dashboard** | | Learn more |
| Dashboard widgets / Database usage | Check if you have any data in your Relational Data database (Total database size, Number of rows). | |
| Dashboard widgets / Data import without errors | Check the widget: **Data Import** | |
| Check if menu item exists and diplays data: **Connections** | | Learn more |
| **Connections** menu | Check if the database connection is listed. | Learn more |
| **Connections** menu | Check if connection test is successful. | Learn more |
| Check if menu item exists and displays data: **Tables & Views** | Check if already existing tables / views are listed. | Learn more |
| Check if menu item exists and displays data: **Segment Templates** | Check if you can create new segment template AND if there is already existing segment template then they are listed | |
| Check if menu item exists and diplays data: **Data Quality** | Data Quality page has a collapsible menu on the left. | Learn more |
| Data Quality/Import history | It should show the results of the previously loaded files. | Learn more |
| Data Quality/API errors | It should show the results of the previously sent API requests in case of API errors. | Learn more |
| Data quality/Personalization Health | It can show possible slow response time for Relational Data-based personalization (above 500ms could cause problem). | Learn more |
| **TABLE & VIEWS PRIMARY KEY CHECK** | | |
| No red alert about missing primary key(s) | UI has no alert. | |

## Recommended Items

These checks help the smooth operation of Relational Data.

| TO CHECK | HOW TO CHECK? | LINK TO DOCUMENTATION |
|---|---|---|
| **DASHBOARD** | | |
| For optimal performance, the queries should finish running less than 500 ms. | **Dashboard** menu shows the values for tables and views. If the values are way above 500ms then some Table / View optimization is needed. | Learn more |
| **CONNECTION** | | |
| Database connections can be checked (internal and external) | Details like:<br>• Table/views size / structure can be checked<br>• Database size<br>• SQL queries can be run<br>• Sample data can be checked | |
| **TABLE & VIEW** | | |
| Only create tables and load data that will be used for segmentation or personalization. | Table and column names must use the following conventions:<br>• Lowercase letters only<br>• No spaces between words are allowed, use underscore to connect them instead ('_')<br>• Table and view names should not start with numbers | |
| Reference fields: Reference field is needed to be set for personalization. | Reference fields column shows the setup. | Learn more |
| Personalization settings:<br>1. Default value can be setup.<br>2. In case there is no default value AND the contact has no value then the email will be canceled (with No content cancel reason)<br>3. Personalization Preview can be used for checking available data in the Relational Data database. | • Check Default value<br>• Check sample data from the database | Learn more |
| Upload file name: In case of SFTP upload, you need to setup a file name. | File name column needs to be setup for the table. | |
| **SEGMENT TEMPLATES** | | |
| Is the **Segment Templates** menu available (applicable only when the setup is self service)? | The menu point is available and clickable. | |
| Create segment template | • New segment template window opens with connection and reference field setup. | Learn more |

| TO CHECK | HOW TO CHECK? | LINK TO DOCUMENTATION |
|---|---|---|
| | • Query can be executed and result will be shown. | |
| DATA QUALITY | | |
| **Data Quality** menu is always available with 3 different options based on the account setup. | The menu point is available and clickable. | Learn more |
| **Import History**: This menu point shows data only when the customer uploads data to the Relational Data database via SFTP. | The menu point shows recently uploaded files with their details and result. | |
| API errors: This menu point shows data only when the customer manages the data via API. | Check if there are any API errors other than 0 -> contact with support or check SAP Emarsys API error codes. | Learn more |
| Personalization health: This menu point shows slower or problematic table(s) and view(s). | If the values are way above 500ms then some table / view optimization should be needed. | Learn more |
| RDS API | | |
| Retry mechanism should be implemented on the customer side. | It can happen that the request is not executed on the first attempt so retry mechanism should be implemented. | |

# Relational Data Use Cases

A comprehensive example for the usage of Relational Data.

**Running personalization queries on your database**

This article provides you information on how SAP Emarsys opens connection to your database in order to generate and run queries to get the personalized content for email campaigns with Relational Data.

**Use Case Example**

To walk you through how to run queries we use a real-life use case and its database for our example.

**The Duration of Creating Relational Data Personalization Queries - FAQ**

Frequently asked questions about the duration of data personalization queries.

# Running personalization queries on your database

This article provides you information on how SAP Emarsys opens connection to your database in order to generate and run queries to get the personalized content for email campaigns with Relational Data.

## When does SAP Emarsys open connection to your database?

SAP Emarsys establishes connection to your database with Relational Data Service (RDS) when you either

- utilize Relational Data features,eg.: launching email campaigns with Relational Data personalization or using an Relational Data segment in an Automation Center program,
- browse the User Interface (UI) of the SAP Emarsys Marketing Platform to collect metadata, or

- import data using the Relational Data API.

### Various connection pools

Relational Data uses separate connection pools to avoid interference between segmentation, personalization and data import. This logic ensures that a slow segment running cannot block data import through the Relational Data API.

**Open connection time:** SAP Emarsys attempts to close the connections as fast as possible, if a connection pool has not been used for a while.

**Connection timeout**: All connectors have a configured 5 seconds connection timeout.

**- Exception:** Amazon Redshift has 15 seconds timeout.

### How to generate and run queries for personalization?

Once the connection has been established to your database SAP Emarsys can generate and run queries to get personalized content.

# Use Case Example

To walk you through how to run queries we use a real-life use case and its database for our example.

### Use case example

An airline company wants to send special upsell offers with recommendations in personalized email campaigns to their contacts according to the destination city and date of arrival. The upsell email campaign is sent with personalized offers for services to buy, such as renting cars or booking hotels at discount prices in the destination city, before the date of arrival.

**Example for upsell email content:**

Based on the contact information we determine to which city and at what time the contact is traveling to, furthermore which service is available for him at the best price in the local currency, for the proper service duration at that city, e.g.:

*Hi,*

*We have seen that you will travel to Vienna on the* ***<date_of_arrival>*** *.*

*Did you know that you can buy the following services at a discount price?*

***<service_name>*** *at* ***<company_name> <price> <currency> <service_duration>***

Here are some use cases:

- Setting Up Your Database
- View Upsell_services
- Setting Up Reference Fields
- Setting Up a Default Value
- Personalizing Multiple Contacts
- Personalizing Multiple Parameters

- Limit in Personalization Queries
- Retrying Queries

# Setting Up Your Database

How to set up your Relational Data database.

Relational databases store data in tables, which are made up of named columns and rows of data. The rows are also called relations, hence their name.

For our use case example, you have your customers' bookings data in your bookings table (Columns: `contact_id`, `booking_id`, `date_of_arrival`, `destination_city`).

The upsell services can be obtained from a view called upsell_services. (Columns: `service_type, service_name, company_name, city, price, currency, price_duration, availability`). You also store information about your contact in the built-in SAP Emarsys contact database.

With Relational Data Service SAP Emarsys can connect to your own (external) databases (which needs to be pre-configured). For details, see Preparing your hosted database for Relational Data.

# View Upsell_services

An example of the upsell_services view.

**View upsell_services**

For this query example we use the view called ***upsell_services***.

| service_type | service_name | company_name | city | price | currency | price_duration | availability |
|---|---|---|---|---|---|---|---|
| rent_a_car | Rent a car | ViaRent | Vienna | 10 | Euro | hour | yes |
| rent_a_car | Rent a car | BerliRent | Berlin | 14 | Euro | hour | no |
| rent_a_car | Rent a car | KaiRent | Cairo | 6 | Egyptian Pound | hour | yes |
| rent_a_car | Rent a car | LDRent | London | 8 | Pound | hour | yes |
| accomodation | Accomodation | GreatHotelVienna | Vienna | 130 | Euro | night | yes |
| rent_a_bike | Rent a bike | TwoWheelsBerlin | Berlin | 4 | Euro | hour | yes |
| rent_a_bike | Rent a bike | TwoWheelsBerlin | Vienna | 2 | Euro | hour | yes |
| accomodation | Accommodation | AwesomeHotel | Berlin | 150 | Euro | Night | no |
| baggage_insurance | baggage_insurance | InsuranceLTD | General | 12 | Euro | trip | Yes |

# Setting Up Reference Fields

A guide on how to set up reference fields in your queries.

**Setting up reference fields**

In the email message we want to use the value of `service_name` based on the value of the `city` and its `availability`.

In order to do that, we have to set two reference fields, which are `city` and `availability`. We use these 2 reference fields both together to identify the values for personalization.

ESL

Script to have the name of an available service located in Vienna:

```
rds.myconnection.upsell_services("Vienna","yes")[0].service_name
```

QuerySAP Emarsys runs the following query to have the value of `service_name` where the `city` = "Vienna" and the `availability` = "yes":

```
select distinct service_name from upsell_services where (city = "Vienna"
                                AND availability = "yes") limit x
```

# Setting Up a Default Value

In case you expect to have some values not represented in your view and you want to make sure that even in that case you have a content there you can set a default value.

**Setting up a default value**

ESL `rds.myconnection.upsell_services("Los_Angeles","yes")[0].service_name`

We already have the reference fields `city` and `availability` set up and we added the default value "General" for the `city` reference field.

**Checking if the default value is needed?**

Now, as a default value is set before we run the query to find the personalized content, first we check whether we need to use the default value at all, by running the following query:

Query

```
select distinct city from upsell_services where city = "Los_Angeles"
                                limit 1
```

## Result: Empty

If the default value query's result is empty, SAP Emarsys needs to use the default value instead of the original value.

Query

```
select distinct service_name from upsell_services where (city = "General"
                                AND availability = "yes") limit x
```

### Result: Not Empty

If the default value query's result is not empty, SAP Emarsys runs the query with the original value (not with the default one).

#### Query

```
select distinct service_name from upsell_services where (city =
                                "Los_Angeles" AND availability = "yes") limit x
```

# Personalizing Multiple Contacts

A guide on how to personalize multiple contacts in a query.

**Personalizing multiple contacts**

Generally, SAP Emarsys can batch together and run queries per 500 contacts per personalization source (table or view). However, if the message contains the same source multiple times with different configurations (parameters), then the batch size is multiplied by the number of different configurations in the message and they are grouped together. Each query will contain up to 1,000 parameter groups in the `where` condition.

A parameter group contains multiple reference fields for a personalization token.

In this example, the `contact.1234` field contains the destination cities for each contact:

```
rds.myconnection.upsell_services(contact.1234,"yes")[0].service_name
```

```
rds.myconnection.upsell_services(contact.1234,"no")[0].service_name
```

This would be 2x500 parameter group, because even though it is from the same source (view ***upsell_services***) it has 2 different configurations. So, for the 500 contacts SAP Emarsys runs 1 query containing 1000 parameters in the `where` condition.

#### Query for default handling

```
select distinct city from upsell_services where (city = "Vienna") OR
                                (city = "Los_Angeles") limit 2
```

#### If the default query result is not empty

```
select distinct service_name from upsell_services where ((city =
                                "Los_Angeles" AND availability = "yes") OR (city = "Vienna" AND ava
                                "yes")) limit x
```

# Personalizing Multiple Parameters

A short example of personalizing multiple parameters.

**Personalizing multiple parameters**

If the message contains the same source multiple times with different configurations or parameters, they get merged together into one query.

ESL

```
rds.myconnection.upsell_services("Los_Angeles","yes")[0].service_name
                        rds.myconnection.upsell_services("Los_Angeles","yes")[0].service_type
```

Query

```
select distinct service_name, service_type from upsell_services where (city = "Los_Angeles" AND ava
```

If multiple sources are used in the same campaign, they are not merged together to create a single query. Each individual source will have its own individual query.

# Limit in Personalization Queries

A short guide on the limits of personalization queries.

### Limit in personalization queries

We use `limit` in every query. When you use the same ESL with different indexes, the limit is increased instead of running the query multiple times.

ESL

```
rds.myconnection.upsell_services("Los_Angeles","yes")[0].service_name
                        rds.myconnection.upsell_services("Los_Angeles","yes")[1].service_name
                        rds.myconnection.upsell_services("Los_Angeles","yes")[2].service_name
```

Query

```
select distinct service_name from upsell_services where (city = "Los_Angeles" AND availability = "y
```

In case of the foreach ESL tag, you can define the limit yourself, however the maximum limit is `100`. If undefined, the limit is `100` by default.

# Retrying Queries

A short guide on how to retry queries if needed.

### Retrying queries

If multiple ESL or personalization tokens are present in your campaigns with multiple personalization sources (table or view) and at least **one** times out, **all queries get retried** not just the faulty slow one.

For example, let's say you have the following in the campaign:

```
rds.myconnection.upsell_services("Los_Angeles","yes")[0].service_name
                        rds.myconnection.bookings(contact.1234,"Los_Angeles")[0].date_of_arrival
```

If one bookings table is slow and fails to return the results within due time, both personalization queries will be retried. As a result, following queries run multiple times in your database :

```
select distinct service_name from upsell_services where (city = "Los_Angeles" AND availability = "y
                        select distinct date_of_arrival from bookings where (user_id = 2 AND destir
```

Of course, in case the campaign is sent out to multiple contacts, the conditions can be more complex than the ones described above.

# The Duration of Creating Relational Data Personalization Queries - FAQ

Frequently asked questions about the duration of data personalization queries.



# What is the Time Limit for Relational Data Personlization Queries?

## What is the time limit for Relational Data personalization queries?

For optimal performance, the queries should finish running less than 500 ms.

# What to Do When a Relational Data Personalization Query is Too Slow?

Optimizing the duration of Relational Data personalization queries is very similar to optimizing relational segment template queries. Creating such well-optimized queries requires deep understanding of the type of database you use, the data you store in your database, the use-case you are trying to create as well as general understanding of query optimization.

### ⚠ Caution

We strongly recommend consulting an expert if you experience performance issues.

1. When creating a personalization query, always consider the following:
    - How large will the result set be?
    - How large is the dataset the query scans to determine the results?
2. After you considered these, you can decide:
    - **Whether to use indexes:** Depending on the type of database you use, this might be different, but generally you should consider using indexes. Most of the time with larger datasets, indexes can help performance. Also, you should index reference fields.
    - **Whether to use functions in the where clause:** Some functions can increase query time significantly because they can result in unintended table scans. Before using them we recommend to research their performance impact.
    - **What you really need to include in the query:** Avoid to use select *. Instead, include the specific columns that you need individually.
    - **Whether to use join instead of correlated subqueries**: Most of the time when operating with larger datasets, joins have better performance than correlated subqueries.

- **How to use joins:** Joins can be time-consuming as well. Do not join unused tables, and always try to join on indexed fields. Also, fields on you are joining should have the same data type.
- **Whether to use wildcards:** Using wildcards will definitely slow down your query, especially for huge tables. Try to avoid it if you can.

# End-user Guides

A guide on how to set up your Relational Data queries.



# Defining Relational Data Reference Fields

Details for prerequsites and key considerations when defining reference fields in Relational Data.

> ⚠ **Caution**
> The **Personalization Settings** is only available with the **Self-service** setup.
>
> However, clients are able to use Relational Data for personalization with any of the setup types. (For full-service clients, SAP Emarsys is doing the personalization setup.)

Relational Data with personalization is where you can turn virtually any of your data into meaningful, relevant and persuasive content.

## Prerequisites

For using personalization in Relational Data you need to meet the following requirements:

- Synchronizing data to the Contact Database,
- Synchronizing data to a database connected to Relational Data,
- Overlapping of fields, i.e. Existing in both your Relational Data and in Contact Database.

## Relational database

The relational part of Relational Data is about enabling the SAP Emarsys Platform to contextualize the different data sets on execution. Then to find the right data from your Relational Data database and use it correctly in the campaigns.

To make the Relational Data usable for the personalization service you need to configure your data sources (tables or views) so that they can be used, and to present the correct information. You do this by designating reference fields and further fine-tune this functionality by defining the sort order of your data.

## Reference fields

A reference field is a common field acting as the key that enables data to be matched between the Platform's contact database and the Relational Database itself. This common key enables the personalization service to know where to look in order to pick the

right data and what makes it relational.

### ⚠ Caution

When using a reference field, you must ensure all fields are indexed. Failing to do so will result in slower queries, which can in some cases cause your campaign to deploy with the Relational Data-based content missing.

### i Note

Composite indices (primary keys or indices with multiple columns) are usable only for reference fields that uses all the columns in the index in the right order, or uses just the first column, the first two columns, the first three columns, and so on.

## Data views

In some instances, you might not have the right data point in your table. In this case, you need to create a view that combines tables. This is a usable data view with a correlating data point which can be referenced.

## Key considerations: Reference fields, Sorting fields

- You need a minimum of one reference field to create a personalization preset using this particular data source.
- The personalization service will use the first entry it finds that matches the criteria and the reference field is the criteria you are telling the system to use.
- Depending on your use case you can use either a single reference field, or a combination of reference fields to make sure that your results are presented in the correct order.

### → Tip

The Sorting fields feature lets you preview the fields you want to use in your personalization tokens. If you add those fields in the configuration you can verify that it is showing up correctly when using the Personalization Preview.

# Personalizing Your Relational Data

Follow these steps to make your Relational Data personalization ready.

## Personalizing your Relational Data

1. Defining one or more reference fields
2. Creating a Relational Data personalization preset
3. Create Relational Data tokens

### i Note

Once you have completed these steps your campaigns will include data as part of the personalized content.

You need to complete these steps each time you want to include new data in a personalization token.

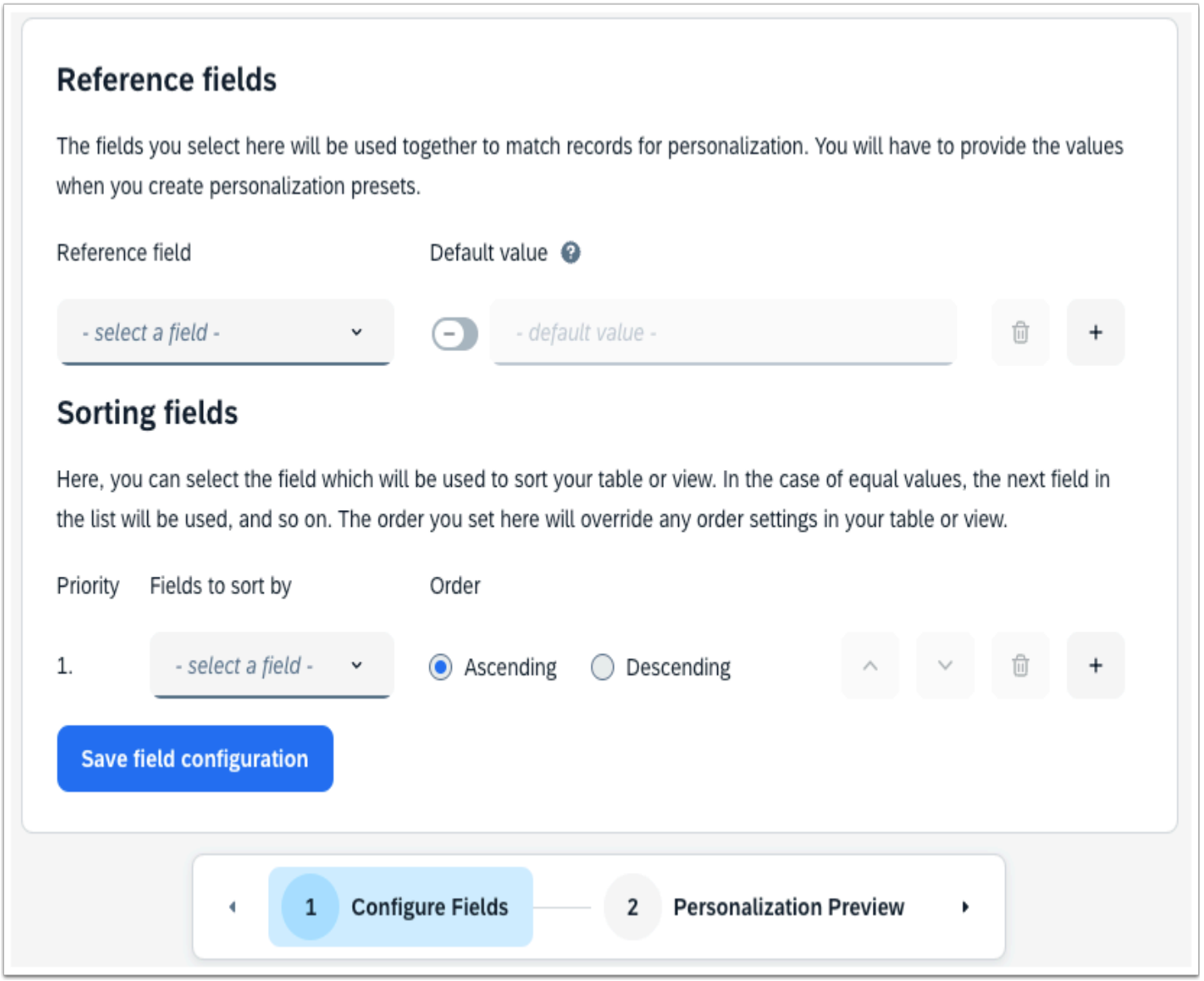
## 1. Defining one or more reference fields

First, you need to set up your Reference fields, so that you could use your tables for personalization.

1. Go to **Add-ons**-> **Relational Data**.

2. In the **tables and views** tab select the table or view that you want to configure.
3. Click the **cogwheel** icon to open the **Personalization Settings**.
4. In the **Configure Fields** tab you can specify which **Reference fields** you want to use, and also how you would like the **Sorting fields** feature to prioritize your results.

   For details, see Defining a single reference field, Defining additional reference fields, Setting up fields to sort.
5. In the **Personalization Preview** tab you can get a preview of the data. You can see how the resulting data looks like and ensure that the right data is used. For details, see Sorting your data correctly.




### Defining a single reference field

1. In the **Reference field** drop-down you can choose from any of the available fields in the table or view that you want to set up. If you cannot select a desired field, then it does not exist in this table or view.
2. The **Default value** toggle allows you to manually specify a fallback value in case the field you use has no data available.

### ℹ Note

The Sorting fields function is not needed when using a single reference field.

### Defining additional reference fields

Once the personalization service runs it will use the first result it finds that matches. If you have a data set with lots of non-unique entries (e.g. historic sales data, multiple subscriptions, etc.) then you need to add multiple reference fields.

When you have data with lots of entries per individual, combining multiple reference fields allows you to find and list the data so that the correct entry is at the top and will be used on launch.

Click the **plus sign** to add another field and repeat it as many time as it is necessary.

> **Note**
> Each time you add a reference field it is removed from the drop-down, so that you do not accidentally add the same one multiple times.

#### Setting up fields to sort

Once you have added all your necessary reference fields you can use the **Sorting fields** feature to define how you want to order your referenced fields.

When you are done click **Save your field configuration**. Check your data is displaying correctly via the **Personalization Preview** tab.

#### Sorting your data correctly

You can configure your personalization tokens using the SAP Emarsys Scripting Language (ESL). Define explicitly the data to use in your token by stating the field name. Specify the result that you want to use, i.e. 1st result, 2nd, 3rd.

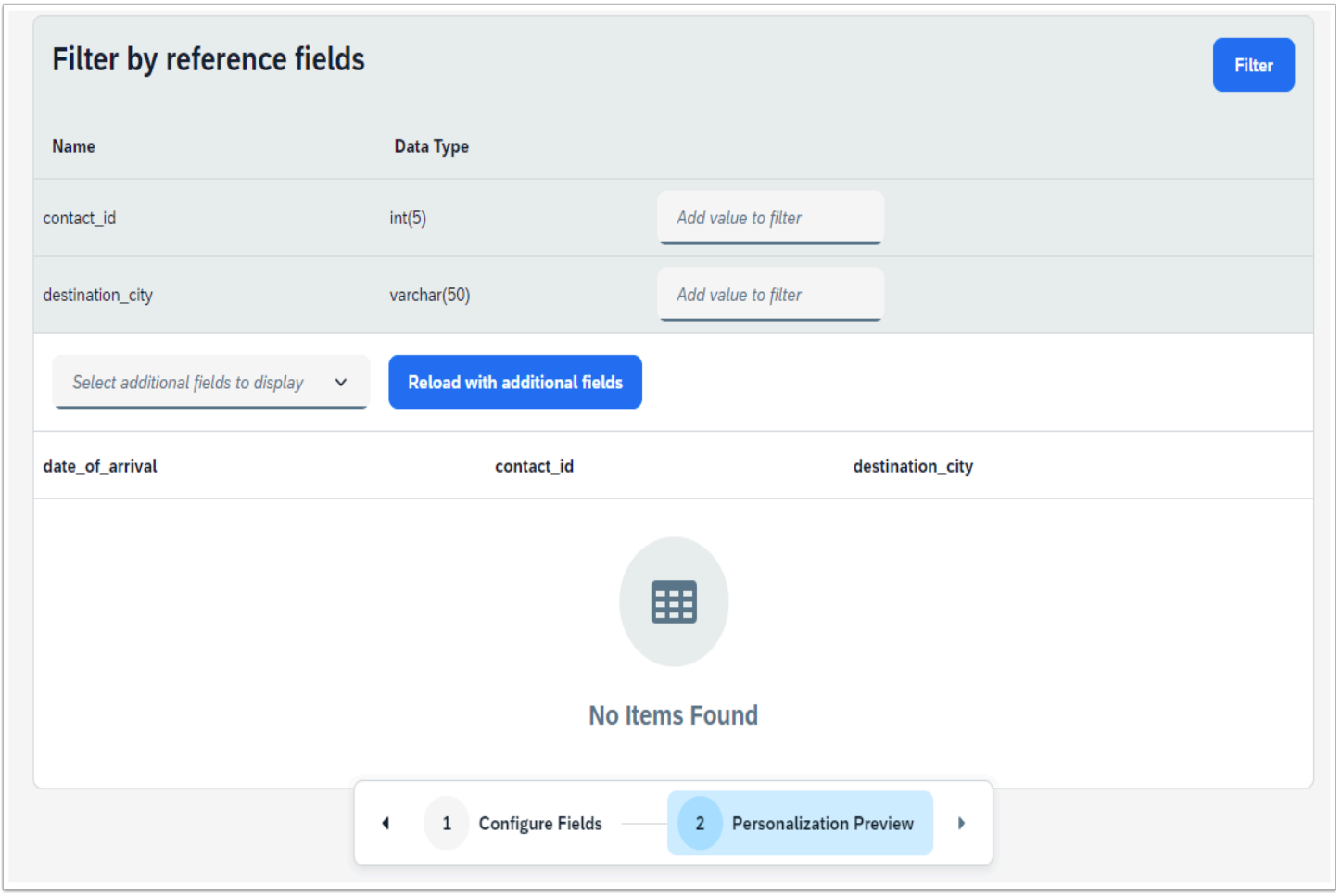
## 2. Personalization Preview

The **Personalization Preview** tab allows you to see what your data will look like to the personalization service. Personalization Preview helps you verify that the desired result is present and correctly displayed.

- Click **Filter**, and you can get a preview of the data for up to 10 contact records.

  You can also narrow the results by entering a result to filter by in your reference field and then clicking Filter to see your results. It is useful if you want to check against a specific parameter (e.g. city name).
- Click **Reload with additional fields** to see output changes when you adjust the results you can add in more reference fields.

> **Caution**
> You cannot make any changes to the way the data is sorted here. To sort your data go back to the **Configure Fields** tab.




To apply changes, go back to the **Reference Fields** tab where you can make any adjustments and then **save your field configuration**.

## 3. Creating a Relational Data personalization preset

Now that your data source has been configured you simply need to create your presets and tokens to make your data usable when the campaign is launched.

For detailed information, see Creating Relational Data tokens.

# Troubleshooting

A guide to troubleshoot your Relational Data setup.

### ℹ Note

To open the **Relational Data** page, go to **Add-ons** > **Relational Data**.

The Relational Data Dashboard

The dashboard is the starting page of the Relational Data, and displays four widgets that show the status of your uploaded data.

Data Quality

This menu item provides insight into the quality of your data, including API errors and personalization health.

# The Relational Data Dashboard

The dashboard is the starting page of the Relational Data, and displays four widgets that show the status of your uploaded data.

## The Relational Data Dashboard

- Personalization health widget
- Slowest tables & views widget

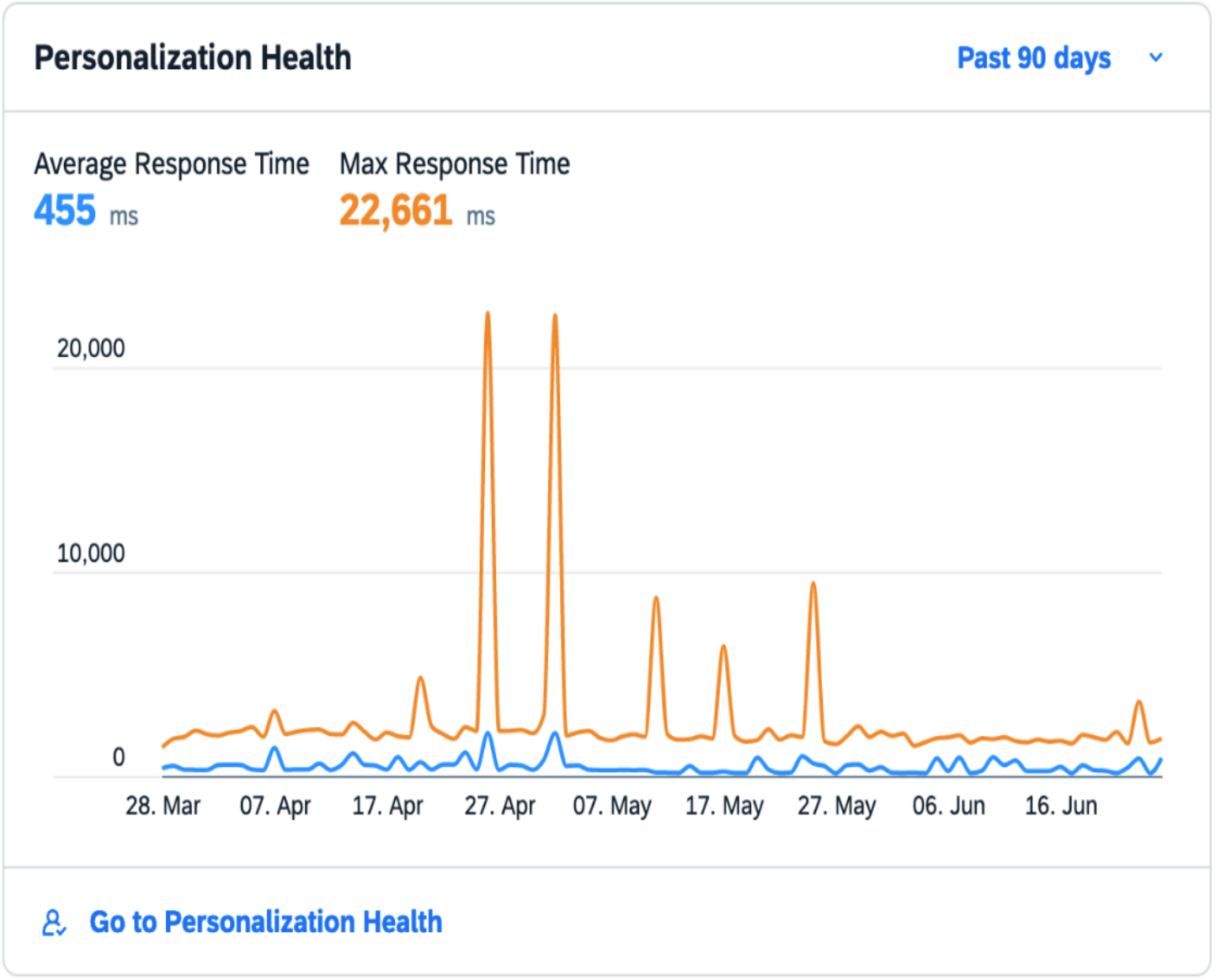
### Personalization health widget

This widget shows the status of your connection:

- Average response time (ms)
- Maximum response time (ms)

**→ Tip**

Click **Go to Personalization Health** to open Personalization health tab, under Data Quality.



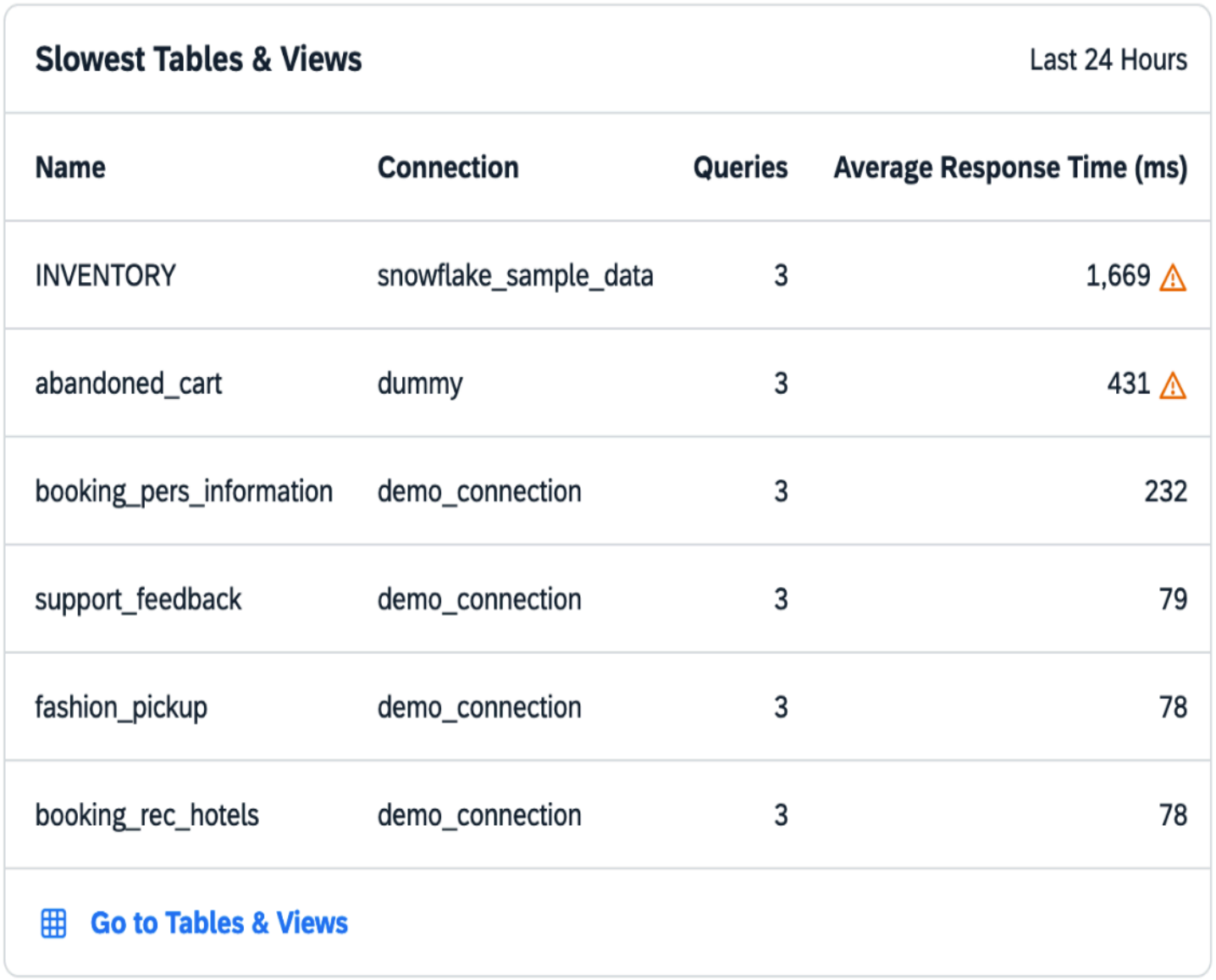

### Slowest tables & views widget

This widget indicates if there is a slow table or view, by showing the following:

- Table/view name
- Connection name
- Number of queries
- Average response time (ms): A warning icon is displayed for connections with a slow average or a max response time.

### → Tip

Click Go to Tables and Views to open Tables & Views tab.




# Data Quality

This menu item provides insight into the quality of your data, including API errors and personalization health.

## Data quality

Data quality page has a collapsible menu on the left, which lets you see the following pages:

- API errors

- Personalization health

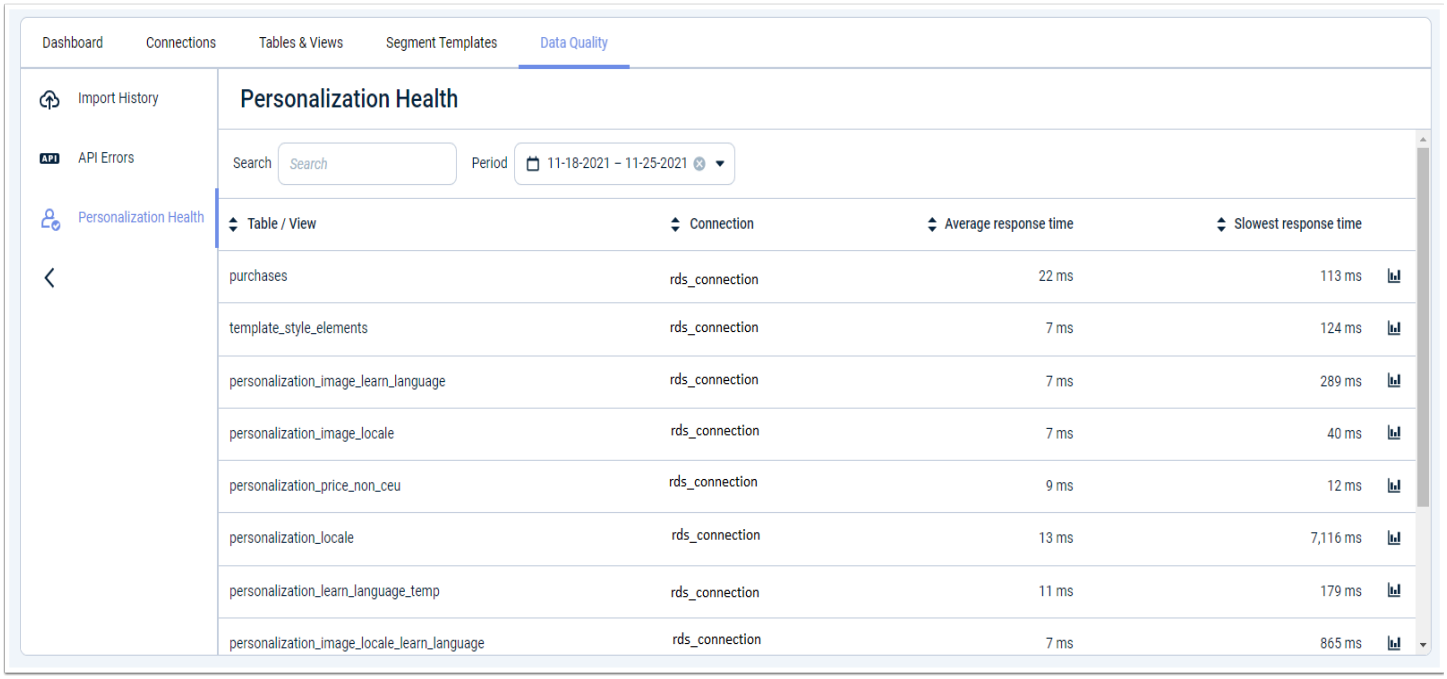
### API errors

This tab provides information about the API errors that might have occurred during the upload:

- **Event time**
- **Table name**
- **Operation (Insert, Upsert, Replace, Update, Delete, Search, Unknown)**
- **Error code** (all error codes are available here )
- **Error message**
- **View packet** - click this button to check the fault request body.

### Personalization health

This tab shows you all tables and views used for personalization, showing the **connection name**, **average response time**, and **the slowest response time**.

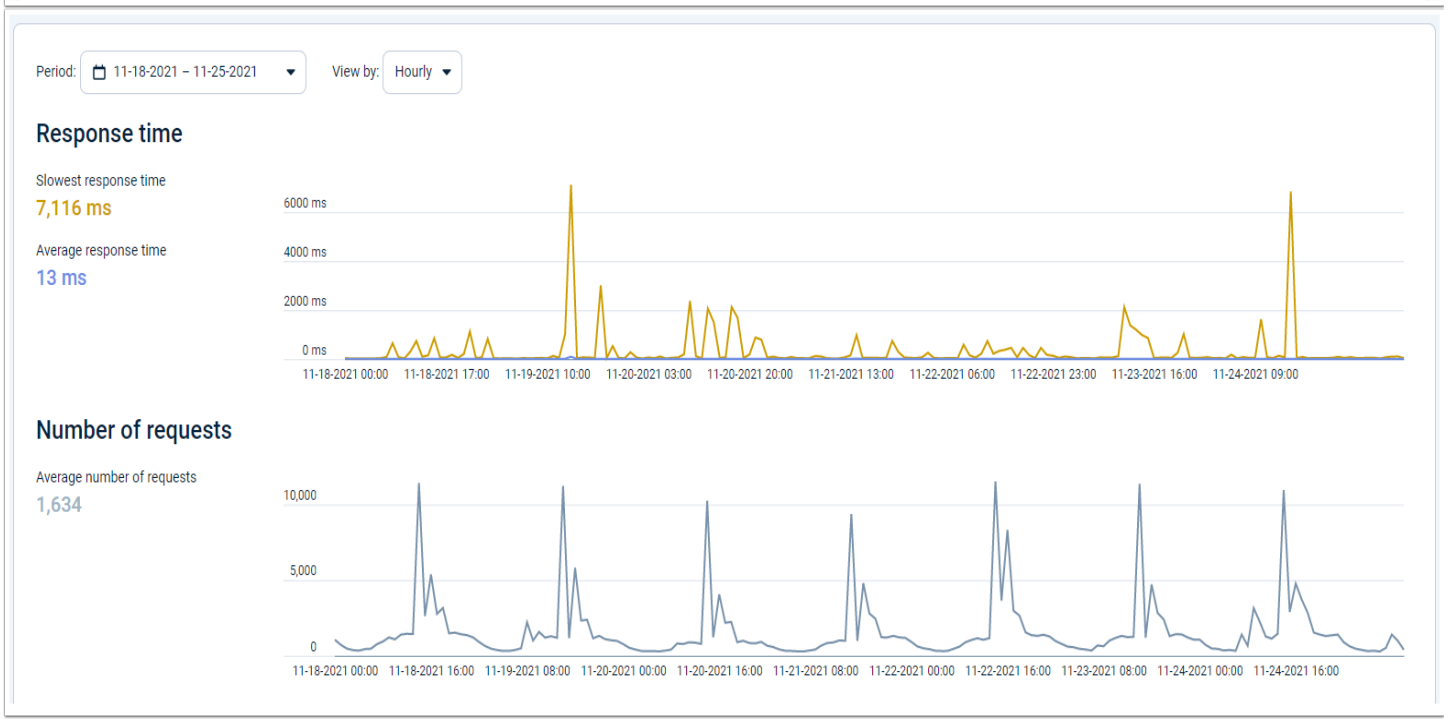



### Personalization health graphs

Click the **graph icon** on the right to drill down into any table or view to get a performance trend over time from the logged response times.

For troubleshooting purposes the view can be toggled from **Daily** to **Hourly** to help narrow down any spikes or issues. This drill down gives you the last 90 days of data to work with, and can be adjusted to focus on any particular scope within that timeframe.


Period: 11-18-2021 – 11-25-2021
View by: Daily
Response time
Slowest response time
7,116 ms
Average response time
14 ms
6000 ms
4000 ms
2000 ms
0 ms
18. Nov
19. Nov
20. Nov
21. Nov
22. Nov
23. Nov
24. Nov
Number of requests
Average number of requests
36,555
40,000
20,000
0
18. Nov
19. Nov
20. Nov
21. Nov
22. Nov
23. Nov
24. Nov


Period: 11-18-2021 – 11-25-2021
View by: Hourly
Response time
Slowest response time
7,116 ms
Average response time
13 ms
6000 ms
4000 ms
2000 ms
0 ms
11-18-2021 00:00
11-18-2021 17:00
11-19-2021 10:00
11-20-2021 03:00
11-20-2021 20:00
11-21-2021 13:00
11-22-2021 06:00
11-22-2021 23:00
11-23-2021 16:00
11-24-2021 09:00
Number of requests
Average number of requests
1,634
10,000
5,000
0
11-18-2021 00:00
11-18-2021 16:00
11-19-2021 08:00
11-20-2021 00:00
11-20-2021 16:00
11-21-2021 08:00
11-22-2021 00:00
11-22-2021 16:00
11-23-2021 08:00
11-24-2021 00:00
11-24-2021 16:00