Qloud Cover Writeback – User Guide

Stretch | www.stretch.se

For questions and inquiries contact support:
servicedesk@stretch.dk

Contents

Introduction

The following is a guide that provides an overview of how to use the Qloud Cover Writeback. Before using this tool, please ensure that the steps in the Installation Guide have been completed.

With the Qloud Cover Writeback Extension you can define as many input columns as desired. The rows in the input columns are linked to existing dimensions in your Qlik data model. These input columns are defined in the Extension and not in the Qloud Cover Writeback GUI. On the Qloud Cover Writeback GUI you will define your Data Warehouse connection and create your Writeback tables in the given Data Warehouse.

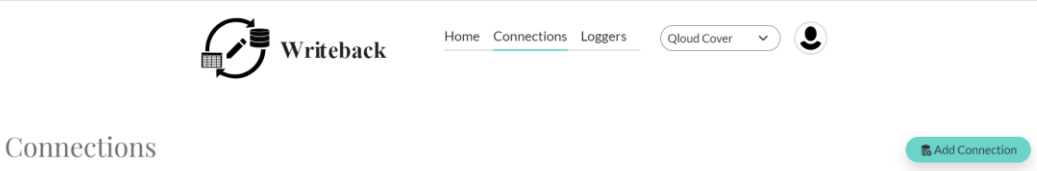
Step 1. Preparing the Data Warehouse

To prepare the Data Warehouse for a Writeback connection we recommend that you prepare the following:

- A dedicated Database/Workgroup/Warehouse/Schema etc. depending on which Warehouse you are connecting to.
- A dedicated Writeback user.
- The Writeback user needs access to the following:
    - Create tables.
    - Create views.
    - Access to the created environment (Database, Warehouse, Schema, etc.).
- It is recommended that the Writeback configuration operates in its own area.
- For **Amazon Redshift** it is required to create a user with IAM.
- If using **Microsoft SQL Server** and Qloud Cover Writeback SaaS the SQL Server must be available for Writeback Qloud Cover SaaS. We have an option set whitelist of IP addresses. Contact Stretch support for more.

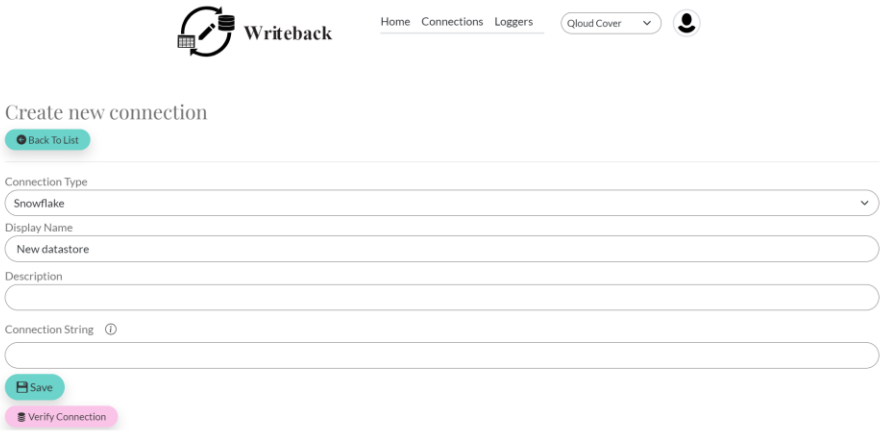
Step 1. Creating a Connection

- Go to Connections in the Qloud Cover Writeback GUI and press Add Connection



- Under *Create new connection* choose which connection you which to connect to.
- The connection string will be removed when inserted and cannot be retrieved afterwards. It is therefore recommended that the Connection String is created locally.

Connect to Snowflake



- **Display Name**: The name of the connection, which is used for displaying the name of the connection.
- **Description** (Optional): Is used for displaying a description of the connection.
- **Connection String**: The connection string which is used to authenticate the connection to the Snowflake Account.

  The following fields are required**:**

  Account=account;Host=hosturl;User=username;Password=password;Warehouse=warehouse;Db=db;Schema=schema;

- It is important that the Connection String are in this exact format, see below for an example:

  Account=vo42714;Host= vo42714.west-europe.azure.snowflakecomputing.com;User=writeback_user;Password=password;Warehouse=writeback_warehouse;Db=writeback_db;Schema=writeback_schema;

- The Writeback user needs access to:
  - Create tables.
  - Create views.
  - Access to the Warehouse, Db and Schema.
- Press Verify Connection to ensure that the connection is setup properly.

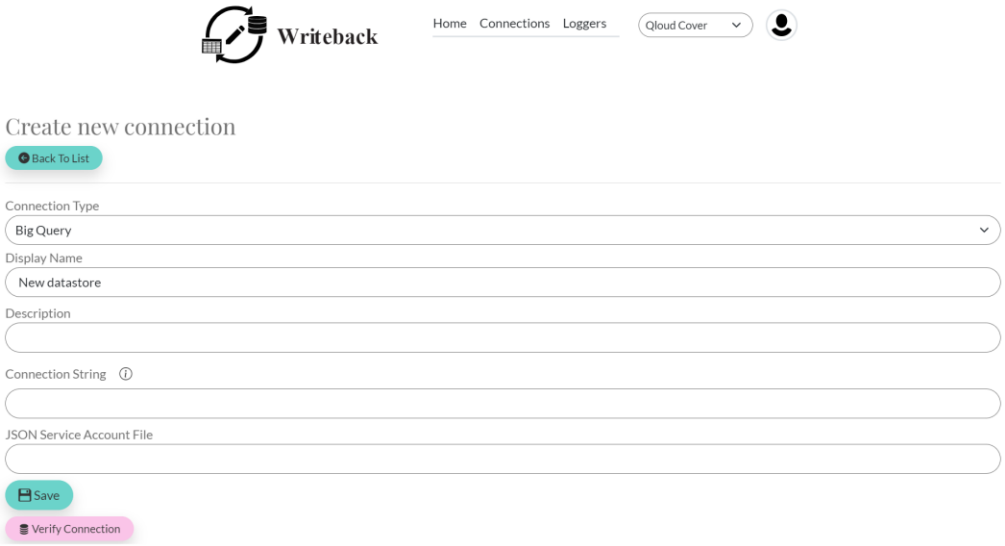
- Ensure that the user you are connecting with has assigned a role in Snowflake that has the necessary permissions. If this is not possible to do via. Snowflake it can be done in by adding the parameter to the connection string:

  Role=role
  See more at https://docs.snowflake.com/en/sql-reference/sql/grant-role

Connect to Google Big Query

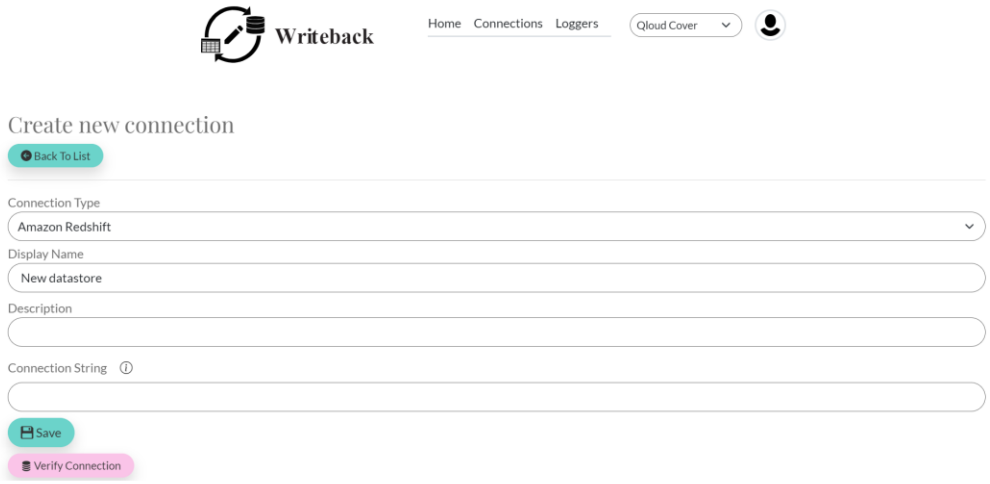
Setting up the User Account for Writeback.

- Create a service account on the google cloud console: https://cloud.google.com/iam/docs/service-accounts-create
- See above for which accesses the Writeback user should.
- Create a JSON key for the created account/user: https://cloud.google.com/iam/docs/keys-create-delete
- This will be used to authenticate Writeback with the specified project.



- **Display Name**: The name of the connection, which is used for displaying the name of the connection.
- **Description** (Optional): Is used for displaying a description of the connection.
- **Connection String:** The connection string must contain the project id that the Writeback configuration should connect to.
- **Example Connection String:** projectid = writeback_projectid
- **JSON Service Account File:** Paste the content of the JSON Key in this field.
- Press Verify Connection to ensure that the connection is setup properly.

Connect to Amazon Redshift



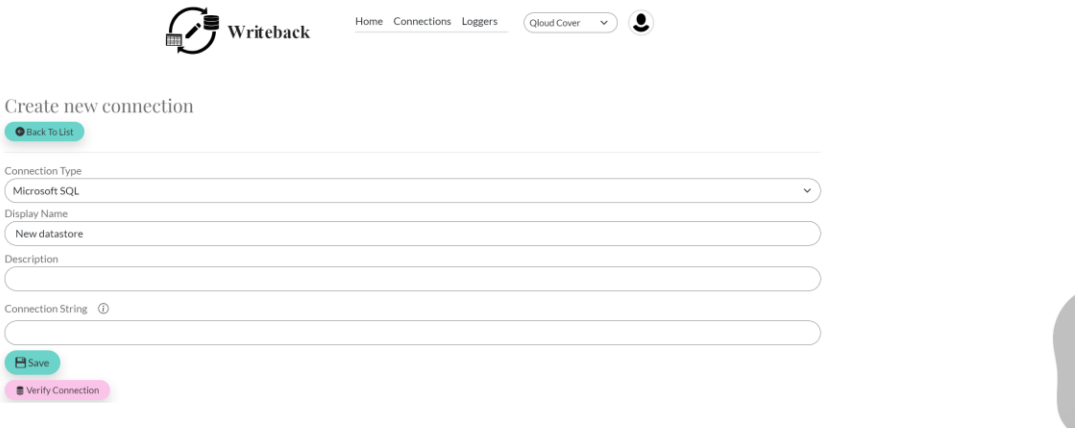
- **Display Name**: The name of the connection, which is used for displaying the name of the connection.
- **Description** (Optional): Is used for displaying a description of the connection.
- **Connection String**: The connection string which is used to authenticate the connection to the Snowflake Account.

  The following fields are required for setting up a connection with Redshift:

  AccessKeyId='accessKeyID';SecretAccessKey='secretAccessKey';Region='region';Workgroup='workgroup';Database='database';Schema='schemaname';MaxQueryTimeMS='maxTimeMS';RetryTimeMS='retryTimeMS';

- **Schema**: Specify the schema to store tables and views in for the given connection. Leave blank to use the default public schema.
- **MaxQueryTimeMS**: a parameter that specifies how long the Writeback service should spent executing queries at Redshift before terminating.
  - **Recommended value** is 20000 (20 seconds).
- **RetryTimeMS:** a parameter that specifies how long the Writeback service should wait when attempting to retry queries (in case of for example waiting for servers to startup).
  - **Recommended value** is 500.
- **Region**: must be in the format: eu-north-1. (as an example)
- Press Verify Connection to ensure that the connection is setup properly.

Connect to Microsoft SQL Server



- **Display Name**: The name of the connection, which is used for displaying the name of the connection.
- **Description** (Optional): Is used for displaying a description of the connection.
- **Connection String**: The connection string which is used to authenticate Microsoft SQL Server**.** The following values are required:

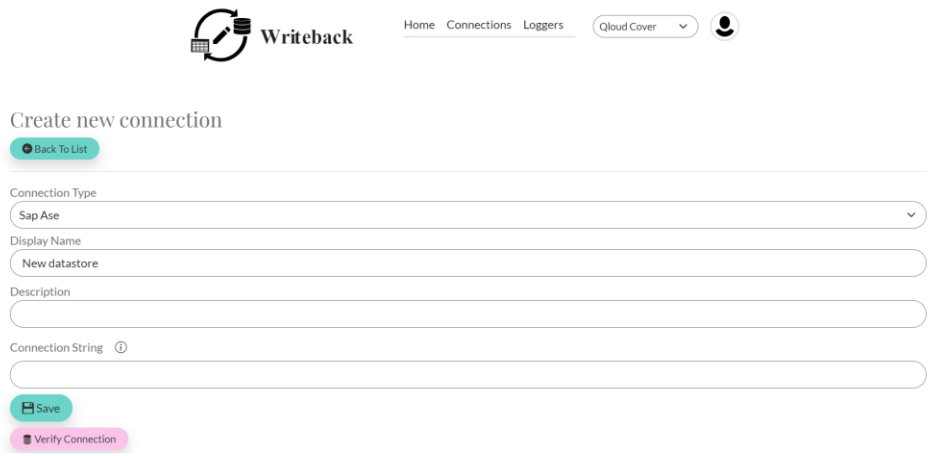
  Server=url-to-azure.com;Database=database;User Id=userID;Password=password;

- Port will default to SQL Server default port: 1433. If you wish to use another port it can be specified in the connection string:

  Server=url-to-azure.com,myPortNumber;Database=database;User Id=userID;Password=password;

- Press Verify Connection to ensure that the connection is setup properly.

Connect to SAP Agentry Cloud Edition (ACE)



- **Display Name**: The name of the connection, which is used for displaying the name of the connection.
- **Description** (Optional): Is used for displaying a description of the connection.
- **Connection String**: The connection string which is used to authenticate to SAP ACE**.** The following values are required:

  Data Source='data-source';Port='port';Database='sap-database';UID='uid';PWD='pwd';

- Press Verify Connection to ensure that the connection is setup properly.

Step 2. Create a Logger Configuration

Each Writeback table created will be linked to a Logger Configuration which will store all the Log Files containing user data and activity from the Writeback extension.

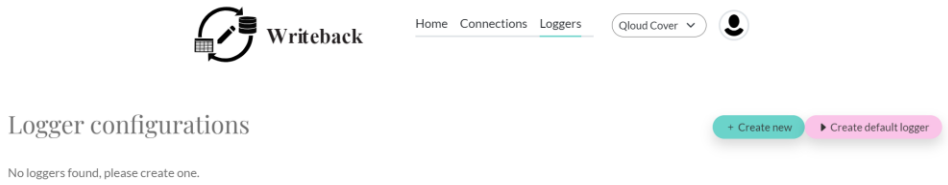
The options are either using the Build in Logger or configuring your own logger to a Cloud Storage. At the moment there is only support for Azure Blob Storage. Contact support for inquiries of other Cloud Storages.

Qloud Cover Writeback Default Logger Configuration

For the Default Logger Configuration all log files are stored in an Azure Blob Storage Account. All the data is stored securely and crypted, with no access to the files by the Qloud Cover Team. If you want to retrieve the filed, then contact servicedesk.

This is useful if you do not want to create your own Azure Blob Storage Account or are trying the product on a free trail.

See below for setup:



- Go to the Loggers tab.
- Press Create default logger.
- The built in Default logger is now created.

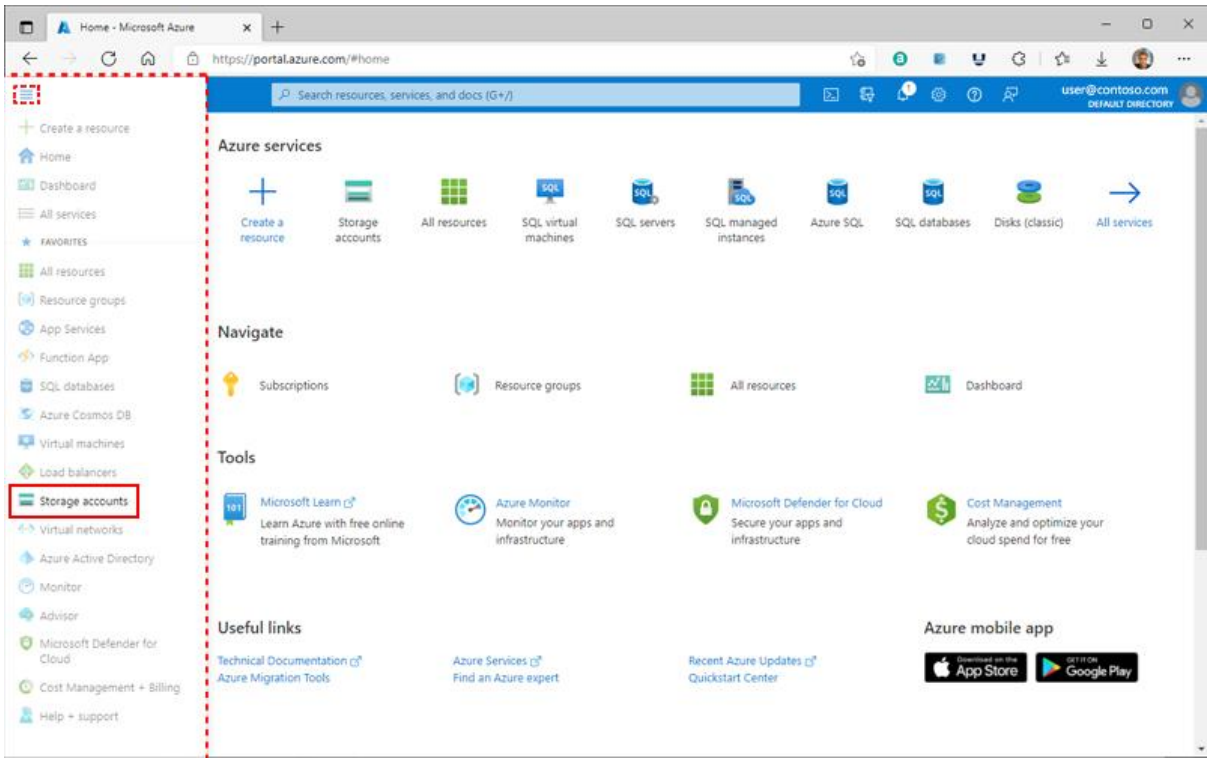
Configure own Azure Blob Storage

Preparing the Azure Blob Storage

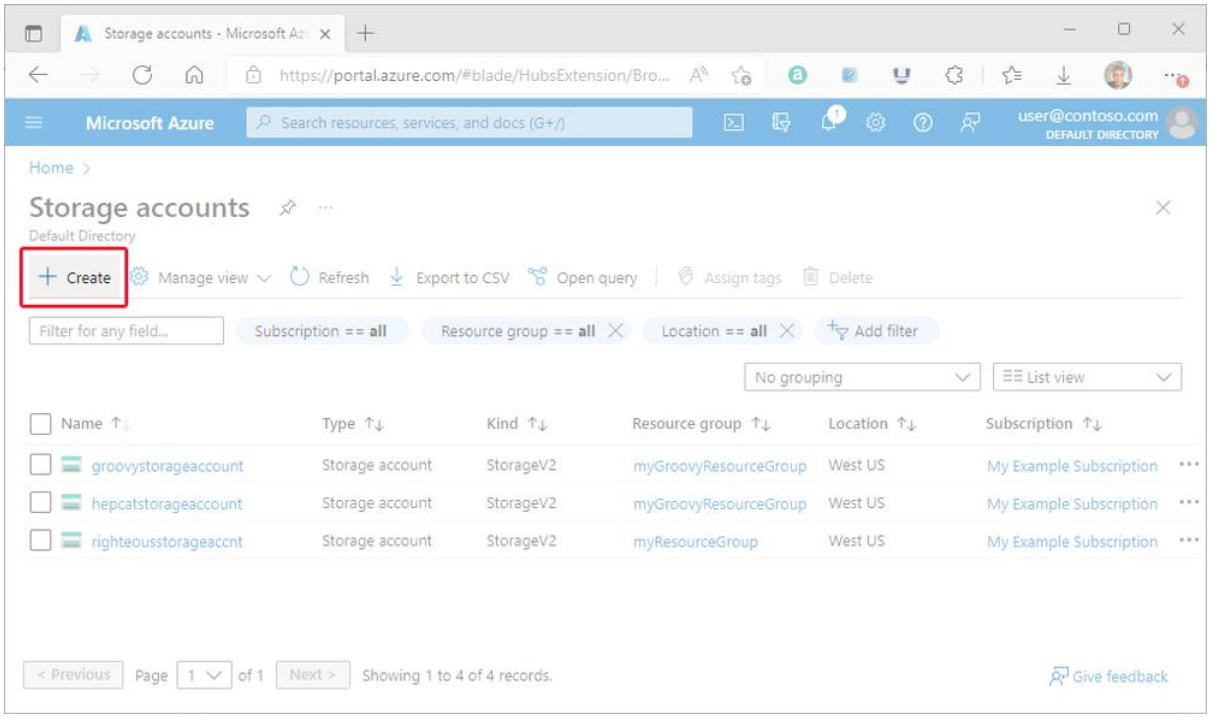
To create an Azure Blob Storage account with the Azure portal, follow these steps:

1. **Navigate to the Azure Portal:** Navigate to and login with your credentials at https://portal.azure.com

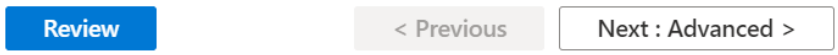
2. **Navigate to Storage Accounts:** From the left portal menu, select Storage accounts to display a list of your storage accounts. If the portal menu isn't visible, click the menu button to toggle it on.



1. **Create Storage Account:** On the Storage accounts page, select Create.



2. Choose a subscription
3. Choose the resource group (a new one can be created with a name of your choosing)
4. Give the storage account a relevant name like: **qloudcoverstorageaccount**
5. Choose the region (depends on the region you're in, we suggest (**Europe) North Europe** for Denmark
6. Choose performance **Standard**
7. Choose redundancy (choice is up to you, but we recommend): **Geo-Redundant storage (GRS)**
8. Click **Next : Advanced >**



9. Keep everything default except for under the step **Blob Storage**, select **Cool** as the access tier:

Basics | **Advanced** | Networking | Data protection | Encryption | Tags | Review

Certain options have been disabled by default due to the combination of storage account performance, redundancy, and region.

Security

Configure security settings that impact your storage account.

Require secure transfer for REST API operations ☑

Allow enabling public access on individual containers ☑

Enable storage account key access ☑

Default to Azure Active Directory authorization in the Azure portal ☐

Minimum TLS version: Version 1.2

Permitted scope for copy operations (preview): From any storage account

Hierarchical Namespace

Hierarchical namespace, complemented by Data Lake Storage Gen2 endpoint, enables file and directory semantics, accelerates big data analytics workloads, and enables access control lists (ACLs) Learn more

Enable hierarchical namespace ☐

Access protocols

Blob and Data Lake Gen2 endpoints are provisioned by default Learn more

Enable SFTP ☐

To enable SFTP, 'hierarchical namespace' must be enabled.

Enable network file system v3 ☐

To enable NFS v3 'hierarchical namespace' must be enabled. Learn more about NFS v3

Blob storage

Allow cross-tenant replication ☑

Access tier

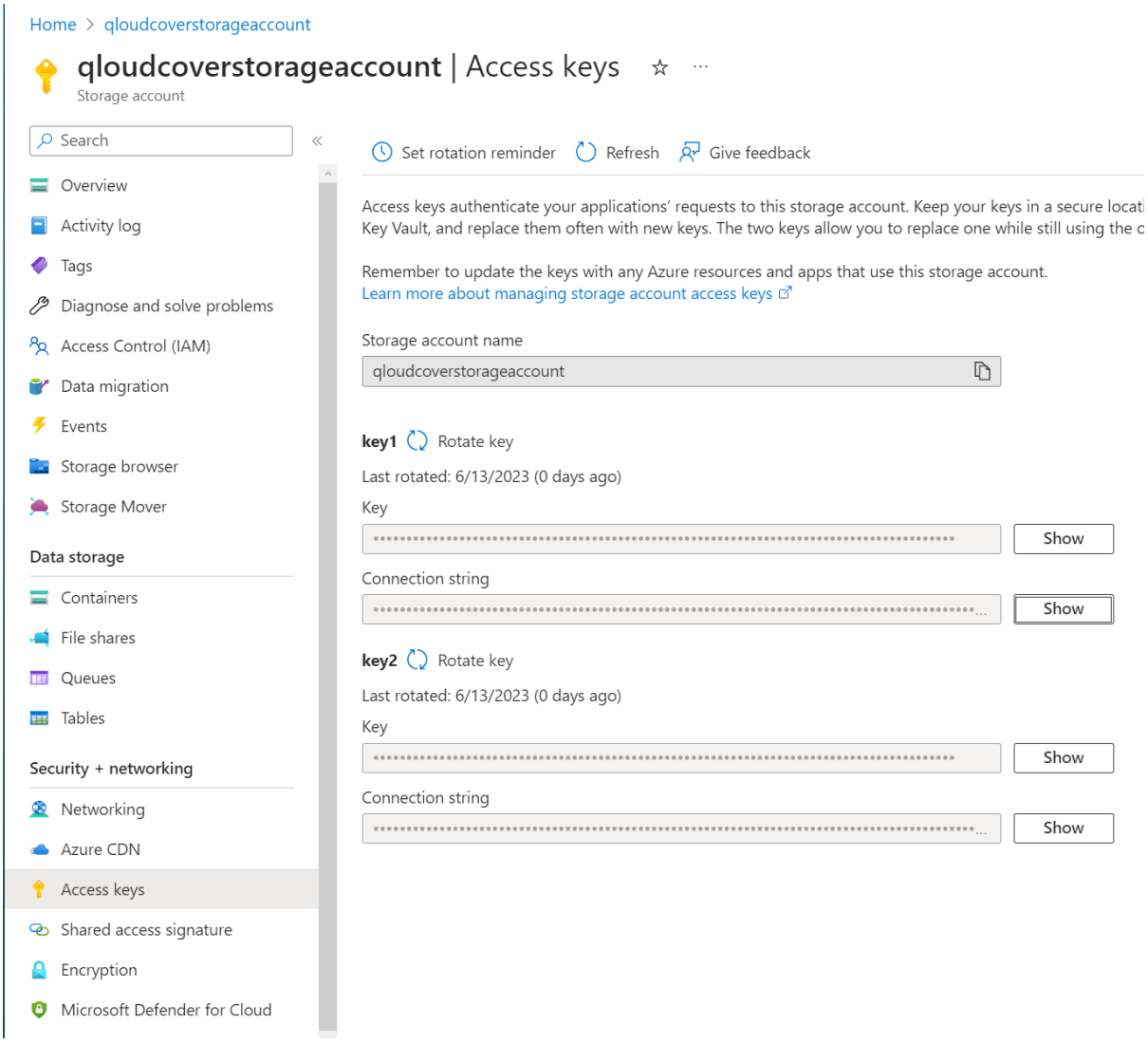
- ○ **Hot:** Frequently accessed data and day-to-day usage scenarios
- ◉ **Cool:** Infrequently accessed data and backup scenarios

Azure Files

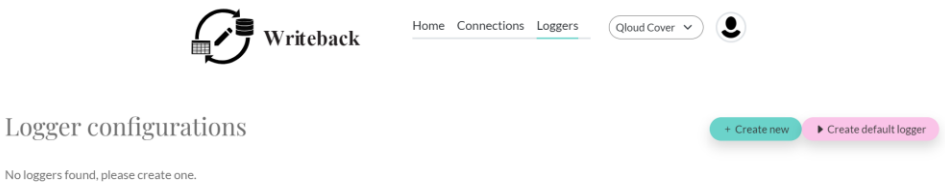
Enable large file shares ☐

10. The next steps should also be kept default, click on **Review** on the top menu bar and click **Create**

11. Navigate to the storage account panel, and click on the **Access keys** tab in the left-hand panel.
12. Under the **key1** section, click the **Show** button next to the *Connection String* box, and copy the connection string. This string will be used in the next step, in which the connections are created in Qloud Cover Writeback.



Configure the Logger

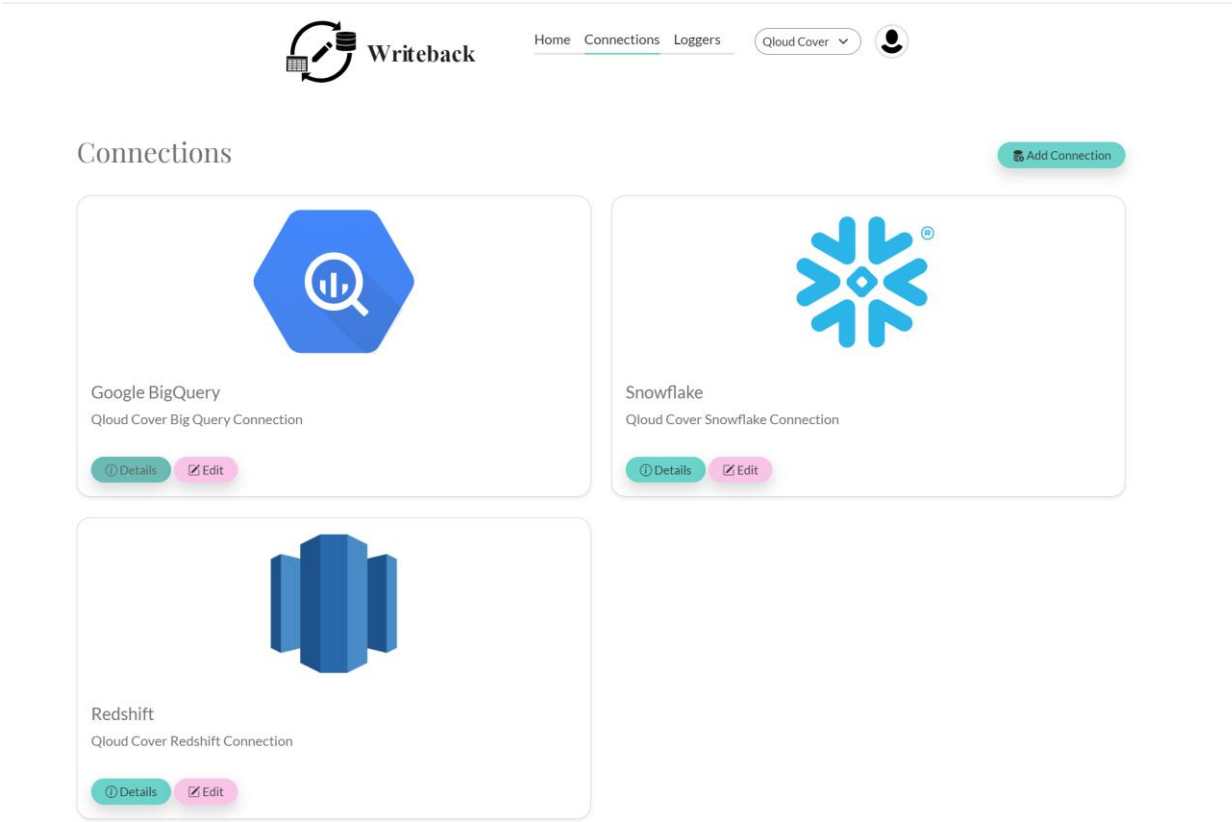


- Go the the Loggers tab
- Press Create New

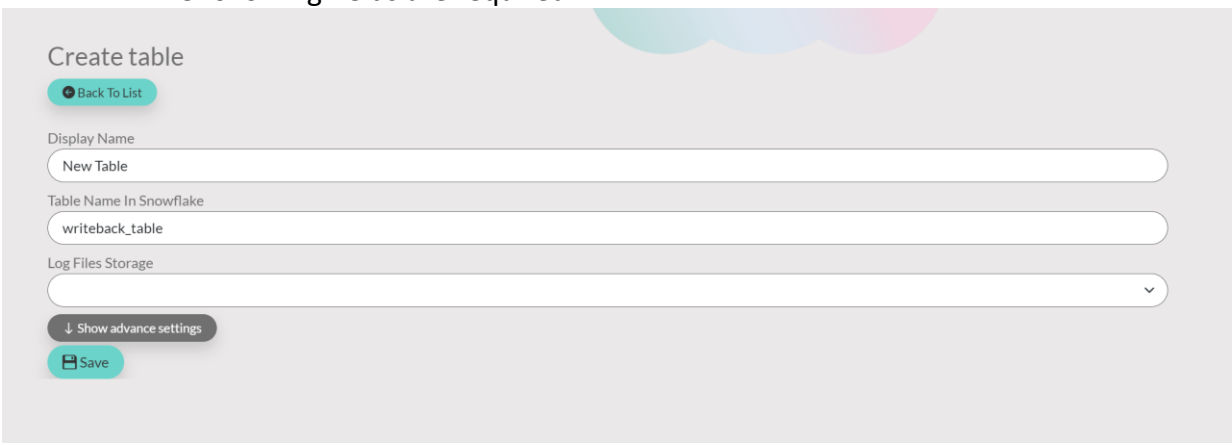
- **Log Files Storage**: Specified which Storage Provider you want to use.
    - At the moment the only option is Azure Blob Storage.
- **Name**: The name of the Logger Configuration.
- **Description**: Description of what the intended use of the Logger Configuration
- **Connection String**: Copy paste the retrieved Connection String from step 12 in the section above.
- **Container Name**: The container name of the Container where the Log Files are stored.
- **Log Event Level**: The Log Event Level of which actions should be logged. The options are:
    - **Verbose**: Everything will be logged. This includes normal activity.
    - **Debug**: Only relevant information for debugging and troubleshooting, but not normal activity.
    - **Information**: Only things that are relevant to be informed.
    - **Warning**: Only warnings from the system.
    - **Error**: Only errors.
    - **Critical**: Only Critical Errors. Is currently not being used.

- Our recommendation is to use Verbose or Debug.
- It can always be changed.
- Press verify connection and ensure that everything is functioning correctly.

Step 3. Creating a new Table Configuration



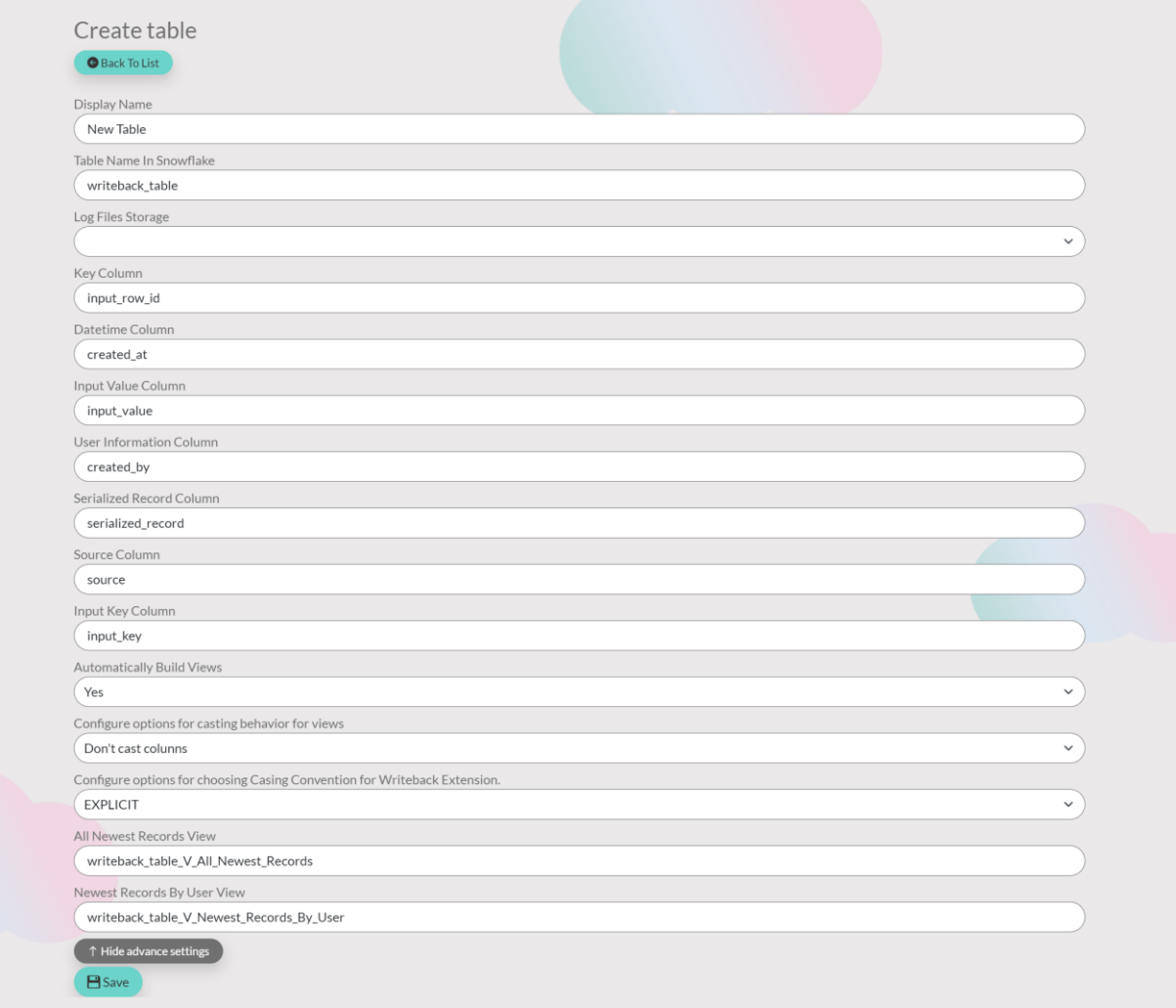
- To create a new Table Configuration press Details on the desired Connection.
- Press Add table.
- The following fields are required:



- **Display Name:** The display name for the Table.
- **Table Name:** The table name for the table created at the Data Warehouse.
- **Log Files Storage:** Choose which Logger Configuration you want to use for the given Table Configuration.
- **(Only for Big Query) Dataset Name:** The name of the Dataset the Writeback should connect to.

Advanced settings:

The following advanced settings can be optionally configured:



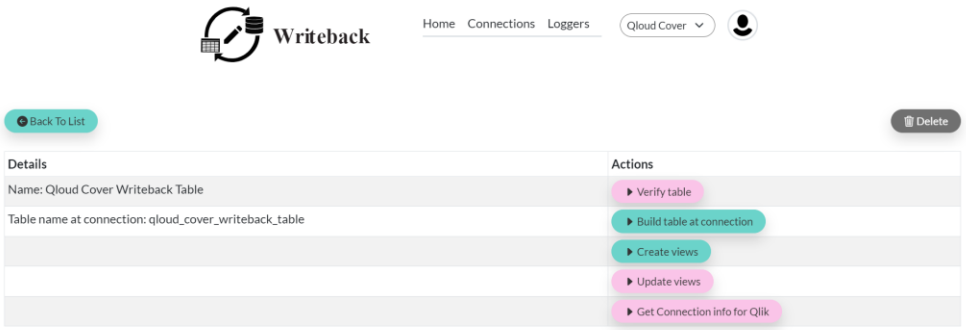
- **Key Column**: Name of the Key Column in the table. The keys are defined in the Extension. See more at Setting up the Extension below.
- **Datetime Column:** The name of the Datetime Column.
- **Input Value Column:** The name of the column where the user input is stored.
    - This column does not specify the input definitions at the Writeback extension. But this column contains the rows of all the input columns in the Writeback extension.
- **User Information Column:** The name of the column where the user information is stored. The user information is taken from the given Qlik app where the Extension is being used.

- **Serialized Record Column:** The name of the Column with a serialized record. This column contains a JSON value of the entire row when a new input is being created by the user.
    - It is recommended to keep the default value.
- **Source Column:** Stores information of the source of where the Extension is being used. E.g. Qlik Cloud or Qlik Sense.
- **Input Key Column:** This column enables a multi key function in the Extension.
    - It is recommended to keep the default value.
- **Automatically Build Views:** Choose if you want to automatically build views when saving the table.
- **Casting behaviour for views:** Configure if you want to autocast the datatype from the Qlik Extension to the views created. Default is Don't cast columns
    - *Number* will be mapped to *Integer.*
    - *Boolean* will be mapped to *Boolean*
    - *Date* will be mapped to *Datetime.*
    - Everything else will be mapped to *String/Varchar/Text.*
- **Casting Convention (Only for Snowflake):** By default, all columns, table name and views names will match the case is they are entered, but this can be configured. **This is disabled for other datastores than Snowflake. Contact Support if you wish to change this.**
    - **Explicit:** Match exact casing conventin as it is entered in the configuration.
    - **Uppercase:** Convert all column names, viewnames and table name to Uppercase.
    - **Lowecase:** Convert al column names, viewnames and table name to Lowercase.
- **All Newest Records View:** The name of the built in view All Newest Records.
    - This view shows all the newest records for the created input columns in the Writeback Extension.
- **Newest Records By User:** The name of the built in view Newest Records By User.
    - This view shows all the newest records by user(s) for the created input columns in the Writeback Extension
- **Log Files Storage:** Choose which Logger Configuration you want to use for the given Table Configuration.

Useful information – Table and Views

1. When saving for the first time the table and views will automatically be created.
2. The following views are being built:
3. All Newest Records Base View: The newest records by user base view will contain _base in the view name.
    a. This view is a base view that can be used by users to create other views if desired.
    b. This view is created automatically on save.

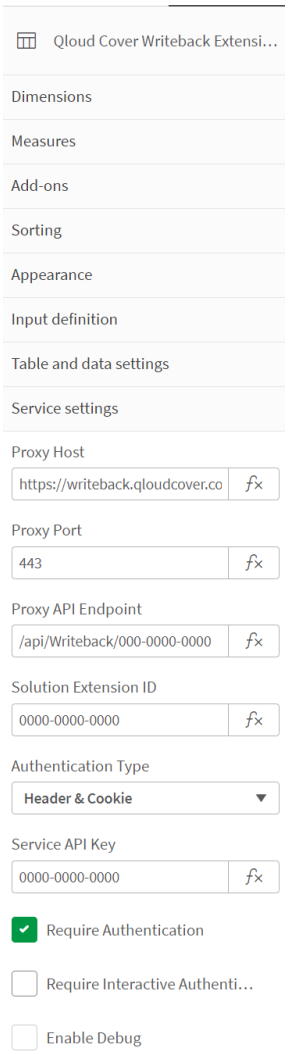
4. Newest Records By User Base View: The newest records by user base view will contain _base in the view name.
    a. This view is a base view that can be used by users to create other views if desired.
    b. This view is created automatically on save.
5. All Newest Records: This view is a view of All Newest Records inputted from the Extension.
    a. This view is not automatically created. Because there is no data in the table, this view cannot be created automatically.
6. Newest Records By User: This view is a view of All Newest Records By Users inputted from the Extension
    a. This view is not automatically created. Because there is no data in the table, this view cannot be created automatically.
7. When the Writeback Extension is being used the views can be updated by pressing "Update views" on the button. (See screenshot below)



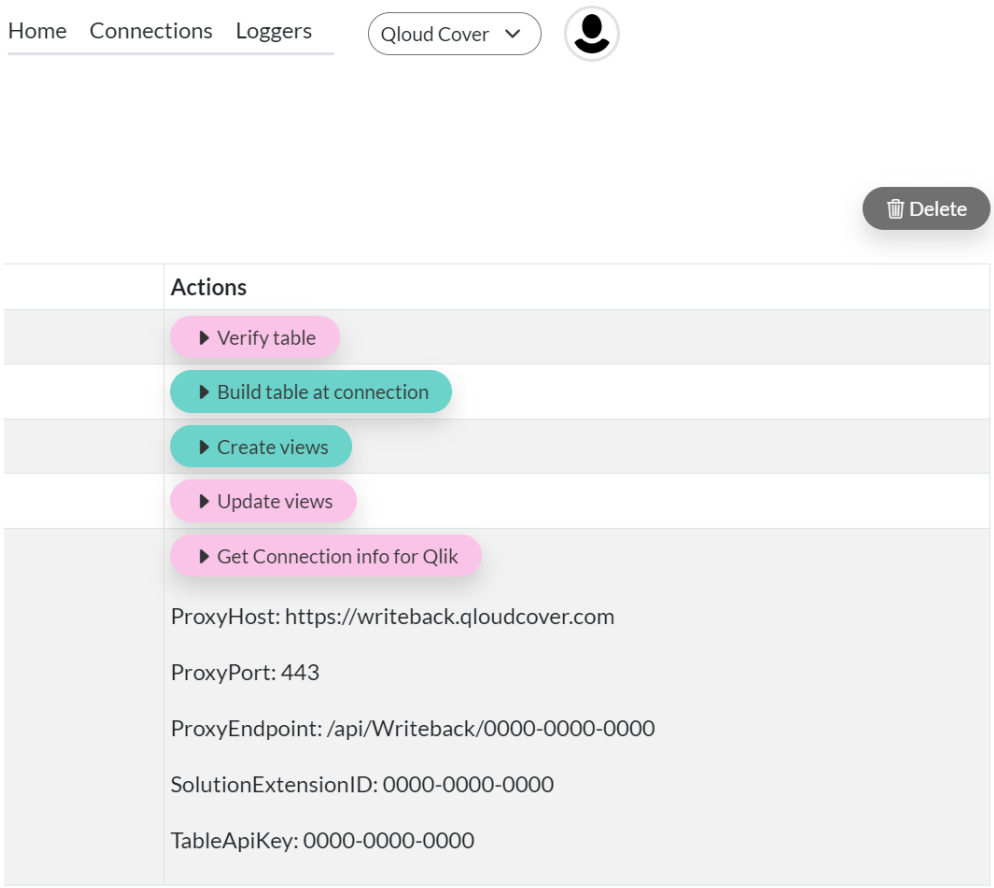
8. When adding a new input column in the Writeback Extension the views must be updated. This can be done by pressing "Update views".
9. If the Views should be deleted, they can be built again by pressing "Create Views".

Step 4. Linking the Writeback extension to the created Table

- Navigate to the Writeback Extension in Qlik.
- Press Edit Sheet and Edit the Extension.
- Go to Service Settings-
- The following values are required:



- Go to Qloud Cover Writeback and locate the table you want to connect to.
- Press Details and find the button **Get Connection info for Qlik**.
- Insert the Information in the respective fields.
- Ensure that Authentication Type is **Header & Cookie.**
- Ensure that Require Authentication is chosen.

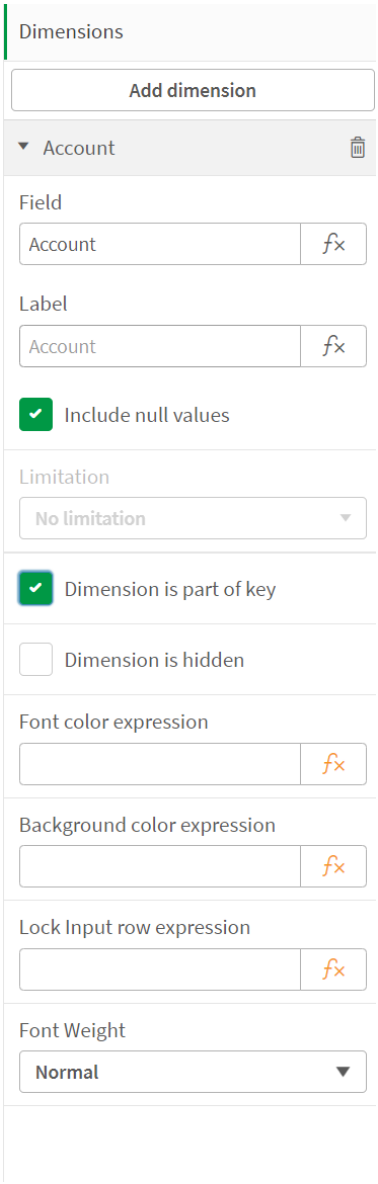


- Navigate to Table and data settings and ensure that “Use Unique Keys when Fetching” is enabled.

Step 5. Linking to the Qlik data model

When linking the Writeback extension to the Qlik data model, you define the dimensions to be used as part of the key. The keys can be created on the granularity you wish to comment on. The keys can be created like a composite key, consisting of multiple dimensions or it can be created on just on dimension, e.g. an Order Id.

- Go to the dimensions section of the Writeback extension.
- As the first dimension (the top of the list), add a dimension.
    - This dimension will be used as the key to link the Qlik data model and the Qloud Cover Writeback Solution.
    - Ensure to mark dimension as “Dimension is part of key”.
    - If desired you can create multiple dimensions and use them as part of the key.
- The key can also be typed in an dimension with “Division &'|' & Department &'|' & Account”.
    - Division, Department and Account are Dimensions in the Qlik data model and | is the separator used in combining the keys
- To display dimensions and measures added to the Writeback extensions choose the desired dimensions and measures and add them without selecting “Dimension is part of key”. These dimensions will then appear in the Writeback extension.
- You can define a Set Expression to disable the row bound to the dimension with a set expression.


Dimensions
Add dimension
Account
Field
Account
Label
Account
Include null values
Limitation
No limitation
Dimension is part of key
Dimension is hidden
Font color expression
Background color expression
Lock Input row expression
Font Weight
Normal

Step 6. Setting up Input definition

- Go to the “Input definition” of the configuration.
- Press add header and configure the following options:
  - **Header:** The display name of the column in the Writeback extension
  - **Dataid:** The unique ID of the column. **Must be unique and can not be changed.**
  - **Placeholder text:** Placeholder text to display on empty rows for the column.
  - **Disable input columns:** *Optional field* where you can define set expressions to disable the given input column.
  - **Type:** The type in the input. The following types are supported:
    - Textbox
    - Text
    - Number
    - Checkbox
    - Date
    - Dropdown
    - Radio button list
    - Slider
  - **Add attributes:** If necessary, you can add attributes to the input. This can e.g. be if the input should only consist of fixed values.
  - **Add values:** You can choose values to be either fixed or Dynamic. Fixed values you will manually enter the values to choose from in Dropdown or Radio Button.
    Values can also be bound to **Dynamic** input which is configured by choosing two dimensions:
    - Use | to seperate values and ~ to seperate values and label: value1~label1|value2~label2.
    - **Example**: *Concat(distinct Dim&'~'&Dim,'|')*, or with dual *Concat(distinct text(dim)&'~'&num(dim),'|')*
  - **Regex validation**: For an input definition with type *number* you can add Regex validation. This will validate the user input and mark it with red if it does not validate with the Regex.
    - **Important** the Regex string must start with and end with /.
    - **Example:** */d+.?d*/*
    -

When adding Input Definitions *the build-in views must be updated. This can be done by pressing the Update Views button. This button is only shown when users are Editing the sheet.*

Common Issues and Solutions

Redshift

Problem:
When creating a new table in Redshift via the Writeback service, the table does not appear in Redshift.

Solution:
Sometimes the Redshift Query Editor can be slow to show newly created table from external sources. Verify that the table is created with the query:

select * from database.schema.tablename;

Problem:

I get the error “Query Exceeded MaxTime (..)”

Solution:

Because the Redshift servers shuts down (if not specified not to) after a period of inactivity, when fetching a result from Redshift the servers sometimes need to shut down. This can exceed the specified MaxTime, meaning that the Writeback service stopped waiting for the Query to complete. Try to run the given Action and check if the error disappears.

Problem:

I can only find two Base Views in the datastore. It can see there should also be two pivot views.

Solution:

The pivot views are only being built when there is data in the extension. If the pivot views does not exists, ensure that there is data in the extension. When adding new input definitions, the views must also be rebuilt to ensure that they are updated.

Problem:

For SAP Adaptive Enterprise Server there is known limitation that it stores empty strings as Whitespace Characters. This means that when a row is deleted in the extension placeholder text will not reapper.
https://help.sap.com/docs/SAP_ASE/993b1744f83f452e94b5c4168e8774d8/aa18390dbc2b1014a444f4a7188a7c05.html?version=16.0.0.0

Solution:

There is currently no solution to this problem. We can add custom logic to solve this problem, but it will increase loading time. Contact support for more.

Releases

Qloud Cover Writeback

- Qloud Cover Writeback Release 2.1 – 02/07/2024
  - Add functionality to specify schema to store tables and views in with Redshift. See more at "**Connect to Redshift**".
  - Add functionality to build tables with serialized_record column with max string length (VARCHAR(MAX)). Is only needed for Redshift, MSSQL and SAP ACE. BigQuery and Snowflake defaults string to max.
  - Add timestamp in Millichanges to created_at column in Redshift.
  - No changes to Qlik Extension.
  - Moved to dotnet 7.0

- Qloud Cover Writeback Release 2.2 – 01/09/2024
  - Add functionality to cast views with data types from Qlik Extension.
  - Add columns conversion for **snowflake** to choose from either using explicit, all uppercase or lowercase. See more in user guide under "*Snowflake".*
  - Moved to dotnet 8.0

- Qloud Cover Writeback Release 2.3 – 01/10/2024
  - Update driver MSSQL DataStore, and added support for MSSQL DataStore in SaaS

- Qloud Cover Writeback Release 2.4 – 01/11/2024
  - Migrate Qloud Cover Writeback Client Managed to windows platform and added support for hosting application on-premise.

Qloud Cover Writeback Extension

- Qloud Cover Writeback Extension Release 3.2 – 01/09/2024
  - Add Regex validation to Number input definition.
  - Add Dynamic values to Dropdown. Can be fetched from dimensions.
  - Add button for updating and building views in Extension.