

Package: qiverse.snowflake (via r-universe)

October 13, 2026

Title Snowflake Connectivity

Version 0.0.3

Description Defines functions needed to connect to Snowflake within an active R session.

License GPL (>= 3) + file LICENSE

Config/testthat/edition 3

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

URL <https://github.com/AUS-DOH-Safety-and-Quality/qiverse/qiverse.snowflake>

BugReports <https://github.com/AUS-DOH-Safety-and-Quality/qiverse/issues>

Depends R (>= 4.1.0)

Imports DBI, odbc, arrow, data.table

Suggests qiverse.azure, qiverse.powerbi, knitr, rmarkdown, testthat (>= 3.0.0), glue

Remotes AUS-DOH-Safety-and-Quality/qiverse/qiverse.azure,
AUS-DOH-Safety-and-Quality/qiverse/qiverse.powerbi

VignetteBuilder knitr

Config/pak/sysreqs cmake make unixodbc-dev libssl-dev

Repository <https://aus-doh-safety-and-quality.r-universe.dev>

Date/Publication 2026-10-13 05:53:40 UTC

RemoteUrl <https://github.com/AUS-DOH-Safety-and-Quality/qiverse>

RemoteRef main

RemoteSha 1ff4db8172ea61894a9627246e65e4850155496f

RemoteSubdir qiverse.snowflake

Contents

create_view_from_definition	2
ingest_dataflow_table	3
ingest_to_snowflake	4
snowflake_con	6
Index	7

create_view_from_definition
<i>Create a View in Snowflake based on an existing table or view</i>

Description

This function creates or replaces a view in the specified target database and schema, using the column definitions from a source table.

Usage

```
create_view_from_definition(  
  con,  
  from_database,  
  from_schema,  
  from_object,  
  to_database,  
  to_schema,  
  to_object,  
  role_name = NULL,  
  warehouse_name = NULL  
)
```

Arguments

con	The connection established by the snowflake_con() function.
from_database	Source database name containing the table, dynamic table or view to select columns from.
from_schema	Source schema name containing the table, dynamic table or view to select columns from.
from_object	Source table, dynamic table or view name to select columns from.
to_database	Target database name where the view will be created.
to_schema	Target schema name where the view will be created.
to_object	Target view name to be created or replaced.
role_name	The role to use for the session.
warehouse_name	The warehouse to use for the session.

Value

A data.frame with output metadata on the final SQL execution

Examples

```
## Not run:
# Create Snowflake azure token
tk <- get_az_tk('sf')
con <- snowflake_con(token = tk)

# Ingest Example To Snowflake
create_view_from_definition <- function(
  con,
  from_database = "DEV_PERSISTENT",
  from_schema = "EXAMPLE",
  from_object = "TEST",
  to_database = "DEV_PRESENTATION",
  to_schema = "EXAMPLE",
  to_object = "TEST",
  role_name = NULL,
  warehouse_name = NULL
)
)

## End(Not run)
```

ingest_dataflow_table *Ingest a PowerBI Dataflow table into Snowflake*

Description

This function uses Snowflake's Azure Blob Storage connectivity to directly import the dataflow table, without needing to first download the file locally and stage.

Usage

```
ingest_dataflow_table(
  con,
  pbi_workspace,
  pbi_dataflow,
  pbi_table,
  database_name,
  schema_name,
  table_name,
  pbi_tk
)
```

Arguments

<code>con</code>	The connection established by the <code>snowflake_con()</code> function.
<code>pbi_workspace</code>	The name of the PowerBI workspace containing the dataflow
<code>pbi_dataflow</code>	The name of the PowerBI dataflow containing the target table
<code>pbi_table</code>	The name of the PowerBI dataflow table to ingest
<code>database_name</code>	The database where the table will be stored.
<code>schema_name</code>	The schema for the table to be stored.
<code>table_name</code>	The name of the table for Snowflake
<code>pbi_tk</code>	The token generated with the correct PowerBI permissions. Use <code>get_az_tk("pbi_df")</code> to create this token.

Value

A data.frame with output metadata on the ingestion process.

Examples

```
## Not run:
# Create Snowflake azure token
tk <- get_az_tk('sf')
con <- snowflake_con(token = tk)

<- get_az_tk('pbi_df')

# Ingest Example To Snowflake
ingest_dataflow_table(
  con = con,
  pbi_workspace = 'My PBI Workspace',
  pbi_dataflow = 'My PBI Dataflow',
  pbi_table = 'MY PBI Table',
  database_name = 'EXAMPLE_DATASET',
  schema_name = 'EXAMPLE',
  table_name = 'TEST',
  pbi_tk = pbi_tk
)

## End(Not run)
```

`ingest_to_snowflake` *Ingest table into Snowflake*

Description

Ingest table into Snowflake

Usage

```
ingest_to_snowflake(  
  data,  
  con,  
  database_name,  
  schema_name,  
  table_name,  
  role_name = NULL,  
  warehouse_name = NULL  
)
```

Arguments

data	This is the R object that we are importing into snowflake. This must be a data.frame, data.table or tibble.
con	The connection established by the snowflake_con() function.
database_name	The database where the table will be stored.
schema_name	The schema for the table to be stored.
table_name	The name of the table for Snowflake
role_name	The role to use for the session. If NULL, no role is set and the user account the default is used.
warehouse_name	The warehouse to use for the session. If NULL, no warehouse is set and the user account the default is used.

Value

A data.frame with output metadata on the ingestion process.

Examples

```
## Not run:  
# Create Snowflake azure token  
tk <- get_az_tk('sf')  
con <- snowflake_con(token = tk)  
  
# Ingest Example To Snowflake  
ingest_to_snowflake(  
  data = data.frame(Alpha = c('a', 'b', 'c'), Number = 1:3),  
  con = con,  
  database_name = 'EXAMPLE_DATASET',  
  schema_name = 'EXAMPLE',  
  table_name = 'TEST'  
)  
  
## End(Not run)
```

snowflake_con

Create Connection to Snowflake

Description

Please ensure that you have the Snowflake ODBC driver installed. This can be found here:

- <https://docs.snowflake.com/en/user-guide/odbc-download.html>.

Also ensure that your default role in Snowflake has been set correctly. If you have not, execute the following command in Snowflake:

```
ALTER USER "MY_USER_ID@EXAMPLE.COM" SET DEFAULT_ROLE = "MY_DEFAULT_ROLE"
```

Usage

```
snowflake_con(server_name = Sys.getenv("snowflake_server"), token)
```

Arguments

server_name	This is the name of Snowflake server you are connecting to. This should include the name, and the server. For example, this value could be "mysnowflakeserver.australia-east"
token	The token generated with the correct Snowflake permissions. Use <code>get_az_tk('sf')</code> to create this token.

Value

Establishes a connection to the Snowflake server, and stores this as an object.

Examples

```
## Not run:
# Create Snowflake azure token
tk <- get_az_tk('sf')
con <- snowflake_con(server_name = 'my-example.server-australiaeast', token = tk)

## End(Not run)
```

Index

`create_view_from_definition`, [2](#)

`ingest_dataflow_table`, [3](#)

`ingest_to_snowflake`, [4](#)

`snowflake_con`, [6](#)