Package 'implyr'

February 6, 2024

Type Package

Title R Interface for Apache Impala

Version 0.5.0

Maintainer Ian Cook <ianmcook@gmail.com>

Description 'SQL' back-end to 'dplyr' for Apache Impala, the massively parallel processing query engine for Apache 'Hadoop'. Impala enables low-latency 'SQL' queries on data stored in the 'Hadoop' Distributed File System '(HDFS)', Apache 'HBase', Apache 'Kudu', Amazon Simple Storage Service '(S3)', Microsoft Azure Data Lake Store '(ADLS)', and Dell 'EMC' 'Isilon'. See <https://impala.apache.org> for more information about Impala.

URL https://github.com/ianmcook/implyr

BugReports https://github.com/ianmcook/implyr/issues

Depends R (>= 3.6), DBI (>= 1.1.3), dplyr (>= 1.1.2)

Imports assertthat, dbplyr (>= 2.4.0), methods, rlang (>= 1.1.1), tidyselect (>= 1.2.0), utils

Suggests Lahman (>= 3.0-1), lubridate, odbc, RJDBC, rJava (>= 0.4-15), nycflights13, stringr, testthat

SystemRequirements Impala driver to support a 'DBI'-compatible R interface

NeedsCompilation no

License Apache License 2.0 | file LICENSE

Encoding UTF-8

RoxygenNote 7.3.1

Author Ian Cook [aut, cre], Cloudera [cph]

Repository CRAN

Date/Publication 2024-02-06 15:40:02 UTC

R topics documented:

compute 2
copy_to 3
dbDisconnect,src_impala-method 5
dbExecute,src_impala,character-method 6
dbGetQuery,src_impala,character-method 6
db_desc 7
impala_unnest 8
src_databases 8
src_impala 9
tbl 10

Index **12**

`compute` *Force execution of an Impala query*

Description

`compute()` Executes the query and stores the result in a new Impala table

`collect()` Executes the query and returns the result to R as a data frame `tbl`

`collapse()` Generates the query for later execution

Usage

```
## S3 method for class 'tbl_impala'
compute(
  x,
  name,
  temporary = TRUE,
  unique_indexes = NULL,
  indexes = NULL,
  analyze = FALSE,
  external = FALSE,
  overwrite = FALSE,
  force = FALSE,
  field_terminator = NULL,
  line_terminator = NULL,
  file_format = NULL,
  ...
)

## S3 method for class 'tbl_impala'
collect(x, ..., n = Inf, warn_incomplete = TRUE)

## S3 method for class 'tbl_impala'
collapse(x, vars = NULL, ...)
```

Arguments

x	an object with class tbl_impala
name	the name for the new Impala table
temporary	must be set to FALSE
unique_indexes	not used
indexes	not used
analyze	whether to run COMPUTE STATS after adding data to the new table
external	whether the new table will be externally managed
overwrite	whether to overwrite existing table data (currently ignored)
force	whether to silently fail if the table already exists
field_terminator	the deliminter to use between fields in text file data. Defaults to the ASCII control-A (hex 01) character
line_terminator	the line terminator. Defaults to "\n"
file_format	the storage format to use. Options are "TEXTFILE" (default) and "PARQUET"
...	other arguments passed on to methods
n	the number of rows to return
warn_incomplete	whether to issue a warning if not all rows retrieved
vars	not used

Note

Impala does not support temporary tables. When using compute() to store results in an Impala table, you must set temporary = FALSE.

---

copy_to *Copy a (very small) local data frame to Impala*

Description

copy_to inserts the contents of a local data frame into a new Impala table. copy_to is intended to be used only with very small data frames. It uses the SQL INSERT ... VALUES() technique, which is not suitable for loading large amounts of data. By default, this function will throw an error if you attempt to copy a data frame with more than 1000 row/column positions. You can increase this limit at your own risk by setting the option implyr.copy_to_size_limit to a higher number.

This package does not provide tools for loading larger amounts of local data into Impala tables. This is because Impala can query data stored in several different filesystems and storage systems (HDFS, Apache Kudu, Apache HBase, Amazon S3, Microsoft ADLS, and Dell EMC Isilon) and Impala does not include built-in capability for loading local data into these systems.

Usage

```
## S3 method for class 'src_impala'
copy_to(
  dest,
  df,
  name = deparse(substitute(df)),
  overwrite = FALSE,
  types = NULL,
  temporary = TRUE,
  unique_indexes = NULL,
  indexes = NULL,
  analyze = FALSE,
  external = FALSE,
  force = FALSE,
  field_terminator = NULL,
  line_terminator = NULL,
  file_format = NULL,
  ...
)
```

Arguments

dest	an object with class with class src_impala
df	a (very small) local data frame
name	name for the new Impala table
overwrite	whether to overwrite existing table data (currently ignored)
types	a character vector giving variable types to use for the columns
temporary	must be set to FALSE
unique_indexes	not used
indexes	not used
analyze	whether to run COMPUTE STATS after adding data to the new table
external	whether the new table will be externally managed
force	whether to silently continue if the table already exists
field_terminator	the deliminter to use between fields in text file data. Defaults to the ASCII control-A (hex 01) character
line_terminator	the line terminator. Defaults to "\n"
file_format	the storage format to use. Options are "TEXTFILE" (default) and "PARQUET"
...	other arguments passed on to methods

Value

An object with class tbl_impala, tbl_sql, tbl_lazy, tbl

Note

Impala does not support temporary tables. When using `copy_to()` to insert local data into an Impala table, you must set `temporary = FALSE`.

Examples

```
library(nycflights13)
dim(airlines) # airlines data frame is very small
# [1] 16  2

## Not run:
copy_to(impala, airlines, temporary = FALSE)
## End(Not run)
```

`dbDisconnect,src_impala-method`

Close the connection to Impala

Description

Closes (disconnects) the connection to Impala.

Usage

```
## S4 method for signature 'src_impala'
dbDisconnect(conn, ...)
```

Arguments

`conn`	object with class class `src_impala`
`...`	other arguments passed on to methods

Value

Returns `TRUE`, invisibly

Examples

```
## Not run:
dbDisconnect(impala)
## End(Not run)
```

---

dbExecute,src_impala,character-method

Execute an Impala statement that returns no result

Description

Executes an Impala statement that returns no result.

Usage

```
## S4 method for signature 'src_impala,character'
dbExecute(conn, statement, ...)
```

Arguments

conn	object with class class src_impala
statement	a character string containing SQL
...	other arguments passed on to methods

Value

Depending on the package used to connect to Impala, either a scalar numeric that specifies the number of rows affected by the statement, or NULL

Note

This method is for statements that return no result, such as data definition or data manipulation statements. Use dbGetQuery() for SELECT queries.

Examples

```
## Not run:
dbExecute(impala, "INVALIDATE METADATA")
## End(Not run)
```

---

dbGetQuery,src_impala,character-method

Send SQL query to Impala and retrieve results

Description

Returns the result of an Impala SQL query as a data frame.

Usage

```
## S4 method for signature 'src_impala,character'
dbGetQuery(conn, statement, ...)
```

Arguments

conn	object with class class src_impala
statement	a character string containing SQL
...	other arguments passed on to methods

Value

A data.frame with as many rows as records were fetched and as many columns as fields in the result set, even if the result is a single value or has one or zero rows

Note

This method is for SELECT queries only. Use dbExecute() for data definition or data manipulation statements.

Examples

```
## Not run:
flights_by_carrier_df <- dbGetQuery(
  impala,
  "SELECT carrier, COUNT(*) FROM flights GROUP BY carrier"
)
## End(Not run)
```

---

db_desc *Describe the Impala data source*

Description

Describe the Impala data source

Usage

```
## S3 method for class 'impala_connection'
db_desc(x)
```

Arguments

x	an object with class class impala_connection

Value

A string containing information about the connection to Impala

impala_unnest *Unnest a complex column in an Impala table*

Description

`impala_unnest()` unnests a column of type `ARRAY`, `MAP`, or `STRUCT` in a `tbl_impala`. These column types are referred to as complex or nested types.

Usage

```
impala_unnest(data, col, ...)
```

Arguments

`data`	an object with class `tbl_impala`
`col`	the unquoted name of an `ARRAY`, `MAP`, or `STRUCT` column
`...`	ignored (included for compatibility)

Details

`impala_unnest()` currently can unnest only one column, can only be applied once to a `tbl_impala`, and must be applied to a `tbl_impala` representing an Impala table or view before applying any other operations.

Value

an object with class `tbl_impala` with the complex column unnested into two or more separate columns

See Also

Impala Complex Types

src_databases *List all available databases*

Description

Returns a character vector containing the names of all the available databases, in alphabetical order, including the `_impala_builtins` database.

Usage

```
src_databases(src, ...)

src_schemas(src, ...)
```

Arguments

src	object with class class src_impala
...	Optional arguments; currently unused.

Details

src_schemas() is an alias for src_databases()

---

src_impala *Connect to Impala and create a remote dplyr data source*

Description

src_impala creates a SQL backend to dplyr for Apache Impala, the massively parallel processing query engine for Apache Hadoop.

src_impala can work with any DBI-compatible interface that provides connectivity to Impala. Currently, two packages that can provide this connectivity are odbc and RJDBC.

Usage

```
src_impala(drv, ..., auto_disconnect = TRUE)
```

Arguments

drv	an object that inherits from DBIDriver-class. For example, an object returned by odbc or JDBC
...	arguments passed to the underlying Impala database connection method dbConnect. See dbConnect,OdbcDriver-method or dbConnect,JDBCDriver-method
auto_disconnect	Should the connection to Impala be automatically closed when the object returned by this function is deleted? Pass NA to auto-disconnect but print a message when this happens.

Value

An object with class src_impala, src_sql, src

See Also

Impala ODBC driver, Impala JDBC driver

Examples

```
# Using ODBC connectivity:

## Not run:
library(odbc)
drv <- odbc::odbc()
impala <- src_impala(
  drv = drv,
  driver = "Cloudera ODBC Driver for Impala",
  host = "host",
  port = 21050,
  database = "default",
  uid = "username",
  pwd = "password"
)
## End(Not run)

# Using JDBC connectivity:

## Not run:
library(RJDBC)
Sys.setenv(JAVA_HOME = "/path/to/java/home/")
impala_classpath <- list.files(
  path = "/path/to/jdbc/driver",
  pattern = "\\.jar$",
  full.names = TRUE
)
.jinit(classpath = impala_classpath)
drv <- JDBC(
  driverClass = "com.cloudera.impala.jdbc41.Driver",
  classPath = impala_classpath,
  identifier.quote = "`"
)
impala <- src_impala(
  drv,
  "jdbc:impala://host:21050",
  "username",
  "password"
)
## End(Not run)
```

---

tbl *Create a lazy* tbl *from an Impala table*

Description

Create a lazy tbl from an Impala table

Usage

```
## S3 method for class 'src_impala'
tbl(src, from, ...)
```

Arguments

src	an object with class with class src_impala
from	a table name or identifier
...	not used

Value

An object with class tbl_impala, tbl_sql, tbl_lazy, tbl

See Also

in_schema

Examples

```
## Not run:
flights_tbl <- tbl(impala, "flights")

flights_tbl <- tbl(impala, in_schema("nycflights13", "flights"))
## End(Not run)
```

Index

collapse *(*compute*)*, 2
collect *(*compute*)*, 2
compute, 2
copy_to, 3

db_desc, 7
dbConnect, *9*
dbDisconnect,src_impala-method, 5
dbExecute(), *7*
dbExecute,src_impala,character-method, 6
dbGetQuery(), *6*
dbGetQuery,src_impala,character-method, 6

impala_unnest, 8
in_schema, *11*

JDBC, *9*

odbc, *9*
option, *3*

src_databases, 8
src_impala, 9
src_schemas *(*src_databases*)*, 8

tbl, 10