

# Rango

October 22, 2018

## R topics documented:

argumentFormatter . . . . .	2
beginTransaction . . . . .	3
beginTransaction,RangoPostgresConnection-method . . . . .	3
beginTransaction,RangoSQLiteConnection-method . . . . .	4
commit . . . . .	4
commit,RangoPostgresConnection-method . . . . .	5
commit,RangoSQLiteConnection-method . . . . .	5
createObjectFromResult . . . . .	6
dataType . . . . .	6
forceAssignMyNamespace . . . . .	7
formatter . . . . .	7
generateClass . . . . .	8
generateClass,RangoPostgresConnection-method . . . . .	8
generateClass,RangoSQLiteConnection-method . . . . .	9
generateClasses . . . . .	9
getConstraints . . . . .	10
getConstraints,list-method . . . . .	10
getConstraints,RangoObject-method . . . . .	11
getJoinClauses . . . . .	11
getJoinClauses,list-method . . . . .	12
getJoinClauses,RangoObject-method . . . . .	12
getPrimaryKey . . . . .	13
getPrimaryKey,RangoPostgresConnection-method . . . . .	13
getPrimaryKey,RangoSQLiteConnection-method . . . . .	14
getTableStatement . . . . .	14
getTableStatement,list-method . . . . .	15
getTableStatement,RangoObject-method . . . . .	15
initialize . . . . .	16
listTables . . . . .	17
listTables,RangoPostgresConnection-method . . . . .	17
listTables,RangoSQLiteConnection-method . . . . .	18
loadClasses . . . . .	18
nonEmptySlots . . . . .	19
nonEmptySlots,RangoObject-method . . . . .	19
optionalField . . . . .	20
rangoConnection . . . . .	20
RangoConnection-class . . . . .	21

RangoObject-class	21
RangoPostgresConnection-class	21
RangoSQLiteConnection-class	21
reconnect	22
reconnect,RangoPostgresConnection-method	22
reconnect,RangoSQLiteConnection-method	23
requiredFields	23
rEsc	24
retrieve	24
retrieve,list-method	25
retrieve,RangoObject-method	25
rollback	26
rollback,RangoPostgresConnection-method	26
rollback,RangoSQLiteConnection-method	27
sqlEsc	27
store	28
store,list,RangoPostgresConnection-method	28
store,list,RangoSQLiteConnection-method	29
store,RangoObject,RangoPostgresConnection-method	29
store,RangoObject,RangoSQLiteConnection-method	30
stringRep	30
stringRep,character-method	31
stringRep,logical-method	31
stringRep,numeric-method	32
stringRep,POSIXct-method	32
stringRep,RangoObject-method	33
%>=%	33
%>%	34
%<=%	34
%<=>%	35
%<%	35
%like%	36

## Index 37

---

argumentFormatter	<i>generate string that can be used as arguments</i>
-------------------	--

---

### Description

generate string that can be used as arguments

### Usage

```
argumentFormatter(field, item, object, dbc)
```

### Arguments

field	which db field is used
item	the item itself
object	object it is associated with
dbc	RangoConnection

**Value**

string that can be used as arguments

**Author(s)**

Willem Ligtenberg

---

`beginTransaction`      *Begin an new transaction*

---

**Description**

Begin an new transaction

**Usage**

`beginTransaction(object)`

**Arguments**

`object`              the database connection to use

**Author(s)**

Willem Ligtenberg

---

`beginTransaction,RangoPostgresConnection-method`  
*Begin an new transaction*

---

**Description**

Begin an new transaction

**Usage**

```
## S4 method for signature 'RangoPostgresConnection'  
beginTransaction(object)
```

**Arguments**

`object`              the database connection to use

**Author(s)**

Willem Ligtenberg

---

beginTransaction,RangoSQLiteConnection-method  
*Begin an new transaction*

---

**Description**

Begin an new transaction

**Usage**

```
## S4 method for signature 'RangoSQLiteConnection'  
beginTransaction(object)
```

**Arguments**

object            the database connection to use

**Author(s)**

Willem Ligtenberg

---

commit            *Commit the changes in the current transaction*

---

**Description**

Commit the changes in the current transaction

**Usage**

```
commit(object)
```

**Arguments**

object            the database connection to use

**Author(s)**

Willem Ligtenberg

---

commit,RangoPostgresConnection-method  
*Commit the changes in the current transaction*

---

### **Description**

Commit the changes in the current transaction

### **Usage**

```
## S4 method for signature 'RangoPostgresConnection'  
commit(object)
```

### **Arguments**

object            the database connection to use

### **Author(s)**

Willem Ligtenberg

---

commit,RangoSQLiteConnection-method  
*Commit the changes in the current transaction*

---

### **Description**

Commit the changes in the current transaction

### **Usage**

```
## S4 method for signature 'RangoSQLiteConnection'  
commit(object)
```

### **Arguments**

object            the database connection to use

### **Author(s)**

Willem Ligtenberg

createObjectFromResult

*Create an object from the result of the query*

---

### **Description**

Create an object from the result of the query

### **Usage**

```
createObjectFromResult(object, dbc, result, create)
```

### **Arguments**

object	The object that was stored or retrieved
dbc	database connection to use
result	the result of the query
create	Should we create the object if it did not exist?

### **Value**

Object itself

### **Author(s)**

Willem Ligtenberg

---

dataType

*Detect the data type of the argument*

---

### **Description**

Detect the data type of the argument

### **Usage**

```
dataType(columnDataRow)
```

### **Arguments**

columnDataRow	specific row of the description of a table
---------------	--

### **Value**

data type as a string

### **Author(s)**

Willem Ligtenberg

---

forceAssignMyNamespace

*Function to force the assignment of new items in the Rango namespace  
Adapted from Rcpp*

---

### **Description**

Function to force the assignment of new items in the Rango namespace Adapted from Rcpp

### **Usage**

```
forceAssignMyNamespace(x, value)
```

### **Arguments**

x  
value

### **Author(s)**

Willem Ligtenberg

---

formatter

*generate correct string representation of the object*

---

### **Description**

generate correct string representation of the object

### **Usage**

```
formatter(object, dbc)
```

### **Arguments**

object	object to format
dbc	RangoConnection

### **Value**

string representation of the object

### **Author(s)**

Willem Ligtenberg

---

`generateClass`                    *generate S4 classes and functions for a specific table in the data base*

---

**Description**

generate S4 classes and functions for a specific table in the data base

**Usage**

```
generateClass(dbc, tableName)
```

**Arguments**

<code>dbc</code>	<code>RangoConnection</code>
<code>tableName</code>	name of the table

**Value**

string containing the class definition

**Author(s)**

Willem Ligtenberg

---

`generateClass,RangoPostgresConnection-method`  
*generate S4 classes and functions for a specific table in the data base*

---

**Description**

generate S4 classes and functions for a specific table in the data base

**Usage**

```
## S4 method for signature 'RangoPostgresConnection'  
generateClass(dbc, tableName)
```

**Arguments**

<code>dbc</code>	<code>RangoPostgresConnection</code>
<code>tableName</code>	name of the table

**Value**

string containing the class definition

**Author(s)**

Willem Ligtenberg



---

*generateClass,RangoSQLiteConnection-method*  
*generate S4 classes and functions for a specific table in the data base*

---

**Description**

generate S4 classes and functions for a specific table in the data base

**Usage**

```
## S4 method for signature 'RangoSQLiteConnection'  
generateClass(dbc, tableName)
```

**Arguments**

dbc	RangoPostgresConnection
tableName	name of the table

**Value**

string containing the class definition

**Author(s)**

Willem Ligtenberg

---

*generateClasses*      *generate S4 classes and functions for a specific table in the data base*

---

**Description**

generate S4 classes and functions for a specific table in the data base

**Usage**

```
generateClasses(dbc = rangoConnection(), fileName = "dbClasses.R")
```

**Arguments**

dbc	RangoConnection
fileName	file name to write the classes to

**Author(s)**

Willem Ligtenberg

---

getConstraints	<i>Return part of a SQL statement that specifies the constraints</i>
----------------	--

---

**Description**

Return part of a SQL statement that specifies the constraints

**Usage**

```
getConstraints(object, dbc, sql = TRUE, constraints = list())
```

**Arguments**

object	object to inspect
dbc	RangoConnection to use
sql	Boolean, return a sql statement (TRUE), or list (FALSE)
clauses	List of clauses already identified

**Value**

String to insert in a SQL statement

**Author(s)**

Willem Ligtenberg

---

getConstraints,list-method	<i>Return part of a SQL statement that specifies the constraints</i>
----------------------------	--

---

**Description**

Return part of a SQL statement that specifies the constraints

**Usage**

```
## S4 method for signature 'list'  
getConstraints(object, dbc, sql = TRUE,  
  constraints = list())
```

**Value**

String to insert in a SQL statement

**Author(s)**

Willem Ligtenberg

---

`getConstraints,RangoObject-method`*Return part of a SQL statement that specifies the constraints*

---

**Description**

Return part of a SQL statement that specifies the constraints

**Usage**

```
## S4 method for signature 'RangoObject'  
getConstraints(object, dbc, sql = TRUE,  
  constraints = list())
```

**Value**

String to insert in a SQL statement

**Author(s)**

Willem Ligtenberg

---

`getJoinClauses`*Return part of a SQL statement that specifies the join clauses*

---

**Description**

Return part of a SQL statement that specifies the join clauses

**Usage**

```
getJoinClauses(object, dbc, sql = TRUE, clauses = list())
```

**Arguments**

<code>object</code>	object to inspect
<code>dbc</code>	RangoConnection to use
<code>sql</code>	Boolean, return a sql statement (TRUE), or list (FALSE)
<code>clauses</code>	List of clauses already identified

**Value**

String to insert in a SQL statement

**Author(s)**

Willem Ligtenberg

---

getJoinClauses,list-method

*Return part of a SQL statement that specifies the join clauses*

---

**Description**

Return part of a SQL statement that specifies the join clauses

**Usage**

```
## S4 method for signature 'list'  
getJoinClauses(object, dbc, sql = TRUE, clauses = list())
```

**Value**

String to insert in a SQL statement

**Author(s)**

Willem Ligtenberg

---

getJoinClauses,RangoObject-method

*Return part of a SQL statement that specifies the join clauses*

---

**Description**

Return part of a SQL statement that specifies the join clauses

**Usage**

```
## S4 method for signature 'RangoObject'  
getJoinClauses(object, dbc, sql = TRUE,  
  clauses = list())
```

**Value**

String to insert in a SQL statement

**Author(s)**

Willem Ligtenberg

---

getPrimaryKey	<i>Get the primary key for a table</i>
---------------	--

---

**Description**

Get the primary key for a table

**Usage**

```
getPrimaryKey(dbc, tableName)
```

**Arguments**

dbc	database connection to use
tableName	name of the table

**Value**

string that contains the primary key of the table

**Author(s)**

Willem Ligtenberg

---

getPrimaryKey, RangoPostgresConnection-method	<i>Get the primary key for a table</i>
---	--

---

**Description**

Get the primary key for a table

**Usage**

```
## S4 method for signature 'RangoPostgresConnection'  
getPrimaryKey(dbc, tableName)
```

**Arguments**

dbc	database connection to use
tableName	name of the table

**Value**

string that contains the primary key of the table

**Author(s)**

Willem Ligtenberg

getPrimaryKey, RangoSQLiteConnection-method  
*Get the primary key for a table*

---

**Description**

Get the primary key for a table

**Usage**

```
## S4 method for signature 'RangoSQLiteConnection'  
getPrimaryKey(dbc, tableName)
```

**Arguments**

dbc	database connection to use
tableName	name of the table

**Value**

string that contains the primary key of the table

**Author(s)**

Willem Ligtenberg

---

getTableStatement      *Return part of a SQL statement that specifies the tables*

---

**Description**

Return part of a SQL statement that specifies the tables

**Usage**

```
getTableStatement(object, dbc, sql = TRUE, tables = list())
```

**Arguments**

object	object to inspect
dbc	RangoConnection to use
sql	Boolean, return a sql statement (TRUE), or list (FALSE)
tables	List of tables already identified

**Value**

String to insert in a SQL statement

**Author(s)**

Willem Ligtenberg

---

`getTableStatement,list-method`

*Return part of a SQL statetement that specifies the tables*

---

**Description**

Return part of a SQL statetement that specifies the tables

**Usage**

```
## S4 method for signature 'list'  
getTableStatement(object, dbc, sql = TRUE, tables = list())
```

**Value**

String to insert in a SQL statement

**Author(s)**

Willem Ligtenberg

---

`getTableStatement,RangoObject-method`

*Return part of a SQL statetement that specifies the tables*

---

**Description**

Return part of a SQL statetement that specifies the tables

**Usage**

```
## S4 method for signature 'RangoObject'  
getTableStatement(object, dbc, sql = TRUE,  
  tables = list())
```

**Value**

String to insert in a SQL statement

**Author(s)**

Willem Ligtenberg

---

initialize	<i>Initialize the RangoPostgresConnection class</i>
------------	---

---

**Description**

Initialize the RangoPostgresConnection class

Initialize the RangoSQLiteConnection class

**Usage**

```
## S4 method for signature 'RangoPostgresConnection'  
initialize(.Object, host, dbname, user,  
          password, cache)
```

```
## S4 method for signature 'RangoSQLiteConnection'  
initialize(.Object, dbname, cache)
```

**Arguments**

.Object	RangoPostgresConnection object
host	host name
dbname	database name
user	user
password	password
cache	boolean which specifies if we use a local cache
.Object	RangoSQLiteConnection object
dbname	database name
cache	boolean which specifies if we use a local cache

**Value**

RangoConnection object

RangoConnection object

**Author(s)**

Willem Ligtenberg

Willem Ligtenberg



---

listTables	<i>list the tables for the current data base</i>
------------	--

---

**Description**

list the tables for the current data base

**Usage**

```
listTables(dbc)
```

**Arguments**

dbc	RangoConnection
-----	-----------------

**Value**

vector containing table names in the data base

**Author(s)**

Willem Ligtenberg

---

listTables,RangoPostgresConnection-method	<i>list the tables for the current data base</i>
---	--

---

**Description**

list the tables for the current data base

**Usage**

```
## S4 method for signature 'RangoPostgresConnection'  
listTables(dbc)
```

**Value**

vector containing table names in the data base

**Author(s)**

Willem Ligtenberg

---

listTables,RangoSQLiteConnection-method  
*list the tables for the current data base*

---

**Description**

list the tables for the current data base

**Usage**

```
## S4 method for signature 'RangoSQLiteConnection'  
listTables(dbc)
```

**Value**

vector containing table names in the data base

**Author(s)**

Willem Ligtenberg

---

loadClasses                    *generate S4 classes and functions for a specific table in the data base  
and loads them into memory*

---

**Description**

generate S4 classes and functions for a specific table in the data base and loads them into memory

**Usage**

```
loadClasses(dbc)
```

**Arguments**

dbc                    RangoConnection

**Author(s)**

Willem Ligtenberg

---

nonEmptySlots	<i>detect non empty slots of an object</i>
---------------	--

---

**Description**

detect non empty slots of an object

**Usage**

```
nonEmptySlots(object, dbc)
```

**Arguments**

object	object to inspect
dbc	RangoConnection to use

**Value**

vector of non empty slots

**Author(s)**

Willem Ligtenberg

---

nonEmptySlots, RangoObject-method	<i>detect non empty slots of an object</i>
-----------------------------------	--

---

**Description**

detect non empty slots of an object

**Usage**

```
## S4 method for signature 'RangoObject'  
nonEmptySlots(object, dbc)
```

**Value**

vector of non empty slots

**Author(s)**

Willem Ligtenberg

---

optionalField	<i>determine the optional fields of a table</i>
---------------	---

---

**Description**

determine the optional fields of a table

**Usage**

```
optionalField(columnData)
```

**Arguments**

columnData	data of a specific column of the data table
------------	---

**Value**

string with the optional fields concatenated

**Author(s)**

Willem Ligtenberg

---

rangoConnection	<i>Create connection to data base</i>
-----------------	---------------------------------------

---

**Description**

Create connection to data base

**Usage**

```
rangoConnection(dbname, host = NA, user = NA, password = NA,
  type = "PostgreSQL", cache = FALSE, logLevel = loglevels["INFO"])
```

**Arguments**

dbname	database name
host	host name
user	user
password	password
type	database type (PostgreSQL)
cache	boolean that specifies if caching should be used

**Value**

Connection object

**Author(s)**

Willem Ligtenberg

---

RangoConnection-class *Object for the connection to the database*

---

**Description**

Object for the connection to the database

**Author(s)**

Willem Ligtenberg

---

RangoObject-class *Root class to identify classes generated by Rango*

---

**Description**

Root class to identify classes generated by Rango

---

RangoPostgresConnection-class  
*Object for the connection to the Postgres database*

---

**Description**

Object for the connection to the Postgres database

**Author(s)**

Willem Ligtenberg

---

RangoSQLiteConnection-class  
*Object for the connection to the SQLite database*

---

**Description**

Object for the connection to the SQLite database

**Author(s)**

Willem Ligtenberg

---

reconnect

*Reconnect to the database without losing the local cache*

---

**Description**

Reconnect to the database without losing the local cache

**Usage**

```
reconnect(dbc)
```

**Arguments**

dbc                    Old RangoConnection object

**Author(s)**

Willem Ligtenberg

---

reconnect,RangoPostgresConnection-method

*Reconnect to the database without losing the local cache*

---

**Description**

Reconnect to the database without losing the local cache

**Usage**

```
## S4 method for signature 'RangoPostgresConnection'  
reconnect(dbc)
```

**Arguments**

dbc                    Old RangoPostgresConnection object

**Value**

renewed RangoPostgresConnection object

**Author(s)**

Willem Ligtenberg

---

reconnect,RangoSQLiteConnection-method

*Reconnect to the database without losing the local cache*

---

### **Description**

Reconnect to the database without losing the local cache

### **Usage**

```
## S4 method for signature 'RangoSQLiteConnection'  
reconnect(dbc)
```

### **Arguments**

dbc                    Old RangoSQLiteConnection object

### **Value**

renewed RangoSQLiteConnection object

### **Author(s)**

Willem Ligtenberg

---

requiredFields

*determine the required fields of a table*

---

### **Description**

determine the required fields of a table

### **Usage**

```
requiredFields(columnData)
```

### **Arguments**

columnData            data of a specific column of the data table

### **Value**

string with the required fields concatenated

### **Author(s)**

Willem Ligtenberg

---

rEsc	<i>escape some quotes for R string</i>
------	--

---

**Description**

escape some quotes for R string

**Usage**

rEsc(s)

**Arguments**

s	String to escape
---	------------------

**Value**

escaped string

**Author(s)**

Willem Ligtenberg

---

retrieve	<i>query the data base for objects with the specific attributes in the object that is passed.</i>
----------	---

---

**Description**

query the data base for objects with the specific attributes in the object that is passed.

**Usage**

```
retrieve(object, dbc, create = TRUE, returnType = "RangoObject")
```

**Arguments**

object	object to retrieve
dbc	RangoConnection
create	Should a new entry be made in the data base if none exist
returnType	How should the result be returned? Default is as RangoObject, None returns nothing (only useful for store), data.frame returns a data.frame

**Value**

returns (a list of) object with all the information from the data base (or a data.frame if chosen)

**Author(s)**

Willem Ligtenberg



---

retrieve,list-method    *query the data base for objects with the specific attributes in the object that is passed.*

---

**Description**

query the data base for objects with the specific attributes in the object that is passed.

**Usage**

```
## S4 method for signature 'list'  
retrieve(object, dbc, create = FALSE,  
         returnType = "RangoObject")
```

**Value**

returns (a list of) object with all the information from the data base (or a data.frame if chosen)

**Author(s)**

Willem Ligtenberg

---

retrieve,RangoObject-method    *query the data base for objects with the specific attributes in the object that is passed.*

---

**Description**

query the data base for objects with the specific attributes in the object that is passed.

**Usage**

```
## S4 method for signature 'RangoObject'  
retrieve(object, dbc, create = FALSE,  
         returnType = "RangoObject")
```

**Value**

returns (a list of) object with all the information from the data base (or a data.frame if chosen)

**Author(s)**

Willem Ligtenberg

---

rollback	<i>Roll back the changes in the current transaction</i>
----------	---

---

**Description**

Roll back the changes in the current transaction

**Usage**

```
rollback(object)
```

**Arguments**

object	the database connection to use
--------	--------------------------------

**Author(s)**

Willem Ligtenberg

---

rollback,RangoPostgresConnection-method	<i>Roll back the changes in the current transaction</i>
---	---

---

**Description**

Roll back the changes in the current transaction

**Usage**

```
## S4 method for signature 'RangoPostgresConnection'  
rollback(object)
```

**Author(s)**

Willem Ligtenberg

---

rollback,RangoSQLiteConnection-method

*Roll back the changes in the current transaction*

---

### **Description**

Roll back the changes in the current transaction

### **Usage**

```
## S4 method for signature 'RangoSQLiteConnection'  
rollback(object)
```

### **Author(s)**

Willem Ligtenberg

---

sqlEsc

*escape some nasties in SQL strings*

---

### **Description**

escape some nasties in SQL strings

### **Usage**

```
sqlEsc(s)
```

### **Arguments**

s                      String to escape

### **Value**

escaped string

### **Author(s)**

Willem Ligtenberg

---

store	<i>store the object to the data base</i>
-------	--

---

**Description**

store the object to the data base

**Usage**

```
store(object, dbc, force = FALSE, returnType = "RangoObject")
```

**Arguments**

object	object to store
dbc	RangoConnection
force	Should we create a new row even when something like this already exists
returnType	How should the result be returned? Default is as RangoObject, None returns nothing (only useful for store), data.frame returns a data.frame

**Value**

returns (a list of) object with all the information from the data base (or a data.frame if chosen)

**Author(s)**

Willem Ligtenberg

---

store,list,RangoPostgresConnection-method	<i>store the list to the data base</i>
---	--

---

**Description**

store the list to the data base

**Usage**

```
## S4 method for signature 'list,RangoPostgresConnection'
store(object, dbc, force = FALSE,
      returnType = "RangoObject")
```

**Value**

returns (a list of) object with all the information from the data base (or a data.frame if chosen)

**Author(s)**

Willem Ligtenberg

---

store,list,RangoSQLiteConnection-method  
*store the list to the data base*

---

### **Description**

store the list to the data base

### **Usage**

```
## S4 method for signature 'list,RangoSQLiteConnection'  
store(object, dbc, force = FALSE,  
       returnType = "RangoObject")
```

### **Value**

returns (a list of) object with all the information from the data base (or a data.frame if chosen)

### **Author(s)**

Willem Ligtenberg

---

store,RangoObject,RangoPostgresConnection-method  
*store the object to the data base*

---

### **Description**

store the object to the data base

### **Usage**

```
## S4 method for signature 'RangoObject,RangoPostgresConnection'  
store(object, dbc,  
       force = FALSE, returnType = "RangoObject")
```

### **Value**

returns (a list of) object with all the information from the data base (or a data.frame if chosen)

### **Author(s)**

Willem Ligtenberg

---

store, RangoObject, RangoSQLiteConnection-method  
*store the object to the data base*

---

**Description**

store the object to the data base

**Usage**

```
## S4 method for signature 'RangoObject,RangoSQLiteConnection'
store(object, dbc,
      force = FALSE, returnType = "RangoObject")
```

**Value**

returns (a list of) object with all the information from the data base (or a data.frame if chosen)

**Author(s)**

Willem Ligtenberg

---

stringRep                      *Generate a string representation for the object*

---

**Description**

Generate a string representation for the object

**Usage**

```
stringRep(object, dbc)
```

**Arguments**

object	object to represent as a string
dbc	RangoConnection

**Value**

string representation of the object

**Author(s)**

Willem Ligtenberg

---

stringRep,character-method

*Generate a string representation for the object*

---

### **Description**

Generate a string representation for the object

### **Usage**

```
## S4 method for signature 'character'  
stringRep(object, dbc)
```

### **Value**

string representation of the object

### **Author(s)**

Willem Ligtenberg

---

stringRep,logical-method

*Generate a string representation for the object*

---

### **Description**

Generate a string representation for the object

### **Usage**

```
## S4 method for signature 'logical'  
stringRep(object, dbc)
```

### **Value**

string representation of the object

### **Author(s)**

Willem Ligtenberg

---

stringRep,numeric-method

*Generate a string representation for the object*

---

**Description**

Generate a string representation for the object

**Usage**

```
## S4 method for signature 'numeric'  
stringRep(object, dbc)
```

**Value**

string representation of the object

**Author(s)**

Willem Ligtenberg

---

stringRep,POSIXct-method

*Generate a string representation for the object*

---

**Description**

Generate a string representation for the object

**Usage**

```
## S4 method for signature 'POSIXct'  
stringRep(object, dbc)
```

**Value**

string representation of the object

**Author(s)**

Willem Ligtenberg



---

stringRep,RangoObject-method

*Generate a string representation for the object*

---

### **Description**

Generate a string representation for the object

### **Usage**

```
## S4 method for signature 'RangoObject'  
stringRep(object, dbc)
```

### **Value**

string representation of the object

### **Author(s)**

Willem Ligtenberg

---

`%>=%`

*Greater than equals function to allow specification arguments where you want some parameter to be greater than or equal to the following value.*

---

### **Description**

Greater than equals function to allow specification arguments where you want some parameter to be greater than or equal to the following value.

### **Usage**

```
var %>=% x
```

### **Arguments**

var	the name of the parameter
x	the value of the parameter

### **Value**

a string that performs the correct sql query

### **Author(s)**

Willem Ligtenberg

---

**%>%***Greater than function to allow specification arguments where you want some parameter to be greater than the following value.*

---

**Description**

Greater than function to allow specification arguments where you want some parameter to be greater than the following value.

**Usage**

```
var %>% x
```

**Arguments**

var	the name of the parameter
x	the value of the parameter

**Value**

a string that performs the correct sql query

**Author(s)**

Willem Ligtenberg

---

**%<=%***Smaller than equals function to allow specification arguments where you want some parameter to be smaller than or equal to the following value.*

---

**Description**

Smaller than equals function to allow specification arguments where you want some parameter to be smaller than or equal to the following value.

**Usage**

```
var %<=% x
```

**Arguments**

var	the name of the parameter
x	the value of the parameter

**Value**

a string that performs the correct sql query

**Author(s)**

Willem Ligtenberg

---

**%<=>%** *Between function to allow specification arguments where you want some parameter to be between the values specified in the vector.*

---

**Description**

Between function to allow specification arguments where you want some parameter to be between the values specified in the vector.

**Usage**

var %<=>% x

**Arguments**

var                    the name of the parameter  
x                        vector specifying the begin and end value

**Value**

a string that performs the correct sql query

**Author(s)**

Willem Ligtenberg

---

**%<%** *Smaller than function to allow specification arguments where you want some parameter to be smaller than the following value.*

---

**Description**

Smaller than function to allow specification arguments where you want some parameter to be smaller than the following value.

**Usage**

var %<% x

**Arguments**

var                    the name of the parameter  
x                        the value of the parameter

**Value**

a string that performs the correct sql query

**Author(s)**

Willem Ligtenberg

---

%like%

*Like function to allow specification arguments for partial matching.*

---

**Description**

Like function to allow specification arguments for partial matching.

**Usage**

```
var %like% x
```

**Arguments**

var	the name of the parameter
x	vector string to match

**Value**

a string that performs the correct sql query

**Author(s)**

Willem Ligtenberg

# Index

[%<=>%](#), [35](#)  
[%<=%](#), [34](#)  
[%<%](#), [35](#)  
[%>=%](#), [33](#)  
[%>%](#), [34](#)  
[%like%](#), [36](#)

[argumentFormatter](#), [2](#)

[beginTransaction](#), [3](#)  
[beginTransaction](#), [RangoPostgresConnection-method](#), [3](#)  
[beginTransaction](#), [RangoSQLiteConnection-method](#), [4](#)

[commit](#), [4](#)  
[commit](#), [RangoPostgresConnection-method](#), [5](#)  
[commit](#), [RangoSQLiteConnection-method](#), [5](#)  
[createObjectFromResult](#), [6](#)

[dataType](#), [6](#)

[forceAssignMyNamespace](#), [7](#)  
[formatter](#), [7](#)

[generateClass](#), [8](#)  
[generateClass](#), [RangoPostgresConnection-method](#), [8](#)  
[generateClass](#), [RangoSQLiteConnection-method](#), [9](#)

[generateClasses](#), [9](#)  
[getConstraints](#), [10](#)  
[getConstraints](#), [list-method](#), [10](#)  
[getConstraints](#), [RangoObject-method](#), [11](#)  
[getJoinClauses](#), [11](#)  
[getJoinClauses](#), [list-method](#), [12](#)  
[getJoinClauses](#), [RangoObject-method](#), [12](#)  
[getPrimaryKey](#), [13](#)  
[getPrimaryKey](#), [RangoPostgresConnection-method](#), [13](#)  
[getPrimaryKey](#), [RangoSQLiteConnection-method](#), [14](#)

[getTableStatement](#), [14](#)  
[getTableStatement](#), [list-method](#), [15](#)

[getTableStatement](#), [RangoObject-method](#), [15](#)

[initialize](#), [16](#)

[listTables](#), [17](#)  
[listTables](#), [RangoPostgresConnection-method](#), [17](#)  
[listTables](#), [RangoSQLiteConnection-method](#), [18](#)

[loadClasses](#), [18](#)

[notEmptySlots](#), [19](#)  
[notEmptySlots](#), [RangoObject-method](#), [19](#)

[optionalField](#), [20](#)

[rangoConnection](#), [20](#)  
[RangoConnection-class](#), [21](#)  
[RangoObject-class](#), [21](#)  
[RangoPostgresConnection-class](#), [21](#)  
[RangoSQLiteConnection-class](#), [21](#)  
[reconnect](#), [22](#)  
[reconnect](#), [RangoPostgresConnection-method](#), [22](#)  
[reconnect](#), [RangoSQLiteConnection-method](#), [23](#)

[requiredFields](#), [23](#)

[rEsc](#), [24](#)

[retrieve](#), [24](#)  
[retrieve](#), [list-method](#), [25](#)  
[retrieve](#), [RangoObject-method](#), [25](#)

[rollback](#), [26](#)  
[rollback](#), [RangoPostgresConnection-method](#), [26](#)  
[rollback](#), [RangoSQLiteConnection-method](#), [27](#)

[sqlEsc](#), [27](#)  
[store](#), [28](#)  
[store](#), [list](#), [RangoPostgresConnection-method](#), [28](#)  
[store](#), [list](#), [RangoSQLiteConnection-method](#), [29](#)

store, RangoObject, RangoPostgresConnection-method, [29](#)  
store, RangoObject, RangoSQLiteConnection-method, [30](#)  
stringRep, [30](#)  
stringRep, character-method, [31](#)  
stringRep, logical-method, [31](#)  
stringRep, numeric-method, [32](#)  
stringRep, POSIXct-method, [32](#)  
stringRep, RangoObject-method, [33](#)