

## Package 'SqlRender'

---

`createRWrapperForSql`     *Create an R wrapper for SQL*

---

### Description

`createRWrapperForSql` creates an R wrapper for a parameterized SQL file. The created R script file will contain a single function, that executes the SQL, and accepts the same parameters as specified in the SQL.

### Usage

```
createRWrapperForSql (sqlFilename, rFilename, packageName,  
                     createRoxygenTemplate = TRUE)
```

### Arguments

<code>sqlFilename</code>	The SQL file.
<code>rFilename</code>	The name of the R file to be generated. Defaults to the name of the SQL file with the extension reset to R.
<code>packageName</code>	The name of the package that will contains the SQL file.
<code>createRoxygenTemplate</code>	If true, a template of Roxygen comments will be added.

### DrillwerForSql

**Usage**

```
loadRenderTranslateSql (sqlFileName, packageName, dbms = "sql server", ...)
```

**Arguments**

<code>sqlFileName</code>	The source SQL file
<code>packageName</code>	The name of the package that contains the SQL file
<code>dbms</code>	

**Value**

Returns a string containing the SQL.

renderSqlFile	Render a SQL file
---------------	-------------------

Description	
renderSqlFile	Renders SQL code in a file based on parameterized SQL and parameter values





**Examples**

```
## Not run:  
sql <- "SELECT * FROM @table_name"  
writeSql(sql, "myParamStatement.sql")  
  
## End(Not run)
```



