

Package: rladybugdb (via r-universe)

October 17, 2026

Title R Interface to the 'LadybugDB' Embedded Graph Database

Version 0.4.4

Description Provides an R interface to 'LadybugDB', an embedded columnar graph database using the 'Cypher' query language. 'LadybugDB' was formerly known as 'Kuzu'. This package uses native 'Rcpp' bindings to the 'LadybugDB' C library, requiring no 'Python' runtime.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

NeedsCompilation yes

SystemRequirements C++20, CMake (>= 3.15)

Imports DBI (>= 1.2.0), methods, Rcpp (>= 1.0.0), rlang (>= 1.0.0)

LinkingTo Rcpp

Suggests arrow, bit64, DBItest, tibble, igraph, tidygraph, testthat (>= 3.0.0), withr, knitr, rmarkdown

VignetteBuilder knitr

URL <https://github.com/hadimaster65555/rladybugdb>

BugReports <https://github.com/hadimaster65555/rladybugdb/issues>

Config/pak/sysreqs cmake

Repository <https://hadimaster65555.r-universe.dev>

Date/Publication 2026-09-17 23:27:07 UTC

RemoteUrl <https://github.com/hadimaster65555/rladybugdb>

RemoteRef HEAD

RemoteSha e328b4a228d7bcc3c67851075a621007a8dc8735

Contents

as.data.frame.lb_result	2
as_arrow_table	3
as_igraph	3
as_tbl_graph	4
as_tibble	5
format.lb_interval	5
Ladybug	6
LadybugConnection-class	6
LadybugDB-DBI-methods	6
ladybugdb_install	7
ladybugdb_is_installed	7
ladybugdb_version	8
LadybugDriver-class	8
LadybugResult-class	8
lb_begin	9
lb_close	9
lb_connection	10
lb_copy_from_arrow	10
lb_copy_from_csv	11
lb_copy_from_df	11
lb_database	12
lb_execute	12
lb_fetch	13
lb_fetch_arrow	14
lb_list_extensions	14
lb_next_result	15
lb_query	15
lb_query_summary	16
lb_result_info	16
lb_set_timeout	17
with_lb_connection	17
Index	18

as.data.frame.lb_result

Convert a LadybugDB result to a data frame

Description

Materialization preserves the current streaming cursor, so it is repeatable and does not change subsequent calls to `lb_fetch()`.

Usage

```
## S3 method for class 'lb_result'  
as.data.frame(x, ...)
```

Arguments

- x An lb_result.
- ... Unused.

Value

A data frame.

as_arrow_table	<i>Convert a LadybugDB result to an Arrow Table</i>
----------------	---

Description

Convert a LadybugDB result to an Arrow Table

Usage

```
as_arrow_table(x, ...)
```

Arguments

- x An lb_result.
- ... Unused.

Value

An arrow::Table backed by LadybugDB's Arrow C Data export.

as_igraph	<i>Convert an lb_result to an igraph object</i>
-----------	---

Description

Extracts node and relationship columns from a LadybugDB query result and constructs an [igraph::graph_from_data_frame](#) object.

Usage

```
as_igraph(x, ...)
```

Arguments

x An lb_result object.
 ... Unused; kept for S3 compatibility.

Details

Node columns must have LadybugDB type NODE and relationship columns type REL/RELATIONSHIP.
 When the result contains no structural columns the function aborts with an informative error.

Value

An `igraph::igraph` object.

Examples

```
## Not run:
db <- lb_database(":memory:")
conn <- lb_connection(db)
lb_execute(conn, "CREATE NODE TABLE Person (name STRING, PRIMARY KEY(name))")
lb_execute(conn, "CREATE REL TABLE Knows (FROM Person TO Person)")
lb_execute(conn, "CREATE (:Person {name:'A'})-[:Knows]->(:Person {name:'B'})")
result <- lb_execute(conn, "MATCH (a:Person)-[r:Knows]->(b:Person) RETURN a, r, b")
g <- as_igraph(result)

## End(Not run)
```

as_tbl_graph	<i>Convert an lb_result to a tbl_graph (tidygraph)</i>
--------------	--

Description

Convert an lb_result to a tbl_graph (tidygraph)

Usage

```
as_tbl_graph(x, ...)
```

Arguments

x An lb_result object.
 ... Unused.

Value

A `tidygraph::tbl_graph()` object.

as_tibble	<i>Convert a LadybugDB result to a tibble</i>
-----------	---

Description

Convert a LadybugDB result to a tibble

Usage

```
as_tibble(x, ...)
```

Arguments

x	An lb_result.
...	Passed to as.data.frame() .

Value

A tibble.

format.lb_interval	<i>Format a LadybugDB interval</i>
--------------------	------------------------------------

Description

Format a LadybugDB interval

Usage

```
## S3 method for class 'lb_interval'  
format(x, ...)
```

Arguments

x	An lb_interval value.
...	Unused.

Ladybug	Create a LadybugDB DBI driver
---------	-------------------------------

Description

Create a LadybugDB DBI driver

Usage

Ladybug()

Value

A LadybugDriver.

LadybugConnection-class	LadybugDB DBI connection
-------------------------	--------------------------

Description

LadybugDB DBI connection

Slots

- database Native database object.
- connection Native connection object.
- state Mutable connection state.
- bigint INT64 conversion policy.

LadybugDB-DBI-methods	LadybugDB DBI methods
-----------------------	-----------------------

Description

LadybugDB implements the standard DBI lifecycle, query, binding, fetch, transaction, quoting, and table-metadata generics. LadybugDB’s native schema distinguishes node and relationship tables, so relational helpers outside this documented subset are intentionally left to graph-specific Cypher and the lb_* API. SQL SELECT, dbWriteTable(), and dbAppendTable() are not translated. dbExecute() returns zero after successful DDL/DML because the LadybugDB 0.20.4 C result summary does not expose a stable affected-row count. Applicable driver behavior is exercised with a focused DBItest subset.

See Also

[Ladybug\(\)](#), [DBI::DBI-package](#)

ladybugdb_install	<i>Install the LadybugDB C library (deprecated)</i>
-------------------	---

Description

This function is deprecated. The C library is now bundled with the package and no Python installation is required.

Usage

```
ladybugdb_install(...)
```

Arguments

... Ignored.

Value

Invisible NULL.

Examples

```
## Not run:  
ladybugdb_install() # no longer needed  
  
## End(Not run)
```

ladybugdb_is_installed	<i>Check whether LadybugDB is available</i>
------------------------	---

Description

Returns TRUE if the native LadybugDB C library is loaded and functional.

Usage

```
ladybugdb_is_installed()
```

Value

TRUE or FALSE.

Examples

```
ladybugdb_is_installed()
```

ladybugdb_version	Return the LadybugDB C library version string
-------------------	---

Description

Return the LadybugDB C library version string

Usage

ladybugdb_version()

Value

A character string, e.g. "0.20.4".

Examples

ladybugdb_version()

LadybugDriver-class	LadybugDB DBI driver
---------------------	----------------------

Description

LadybugDB DBI driver

Slots

state Internal driver state.

LadybugResult-class	LadybugDB DBI result
---------------------	----------------------

Description

LadybugDB DBI result

Slots

- connection Owning DBI connection.
- state Mutable result state.
- statement Original Cypher statement.

`lb_begin`*Transaction control*

Description

Transaction control

Usage

```
lb_begin(conn)
```

```
lb_commit(conn)
```

```
lb_rollback(conn)
```

Arguments

`conn` An `lb_connection`.

Value

Invisible conn.

`lb_close`*Close a LadybugDB result, connection, or database*

Description

Closing an object more than once is harmless. A connection cannot be closed while it owns open results, and a database cannot be closed while it owns open connections.

Usage

```
lb_close(x, ...)
```

Arguments

`x` An `lb_result`, `lb_connection`, or `lb_database`.

`...` Unused.

Value

Invisible NULL.

lb_connection	<i>Open a connection to a LadybugDB database</i>
---------------	--

Description

Open a connection to a LadybugDB database

Usage

```
lb_connection(database, num_threads = NULL)
```

Arguments

database	An lb_database object.
num_threads	Positive integer execution thread limit, or NULL.

Value

An lb_connection object.

lb_copy_from_arrow	<i>Load an Arrow object into a LadybugDB table</i>
--------------------	--

Description

Registers a temporary Arrow-backed table through the Arrow C Data Interface, copies its rows into an existing native node table, and unregisters it. No data-frame or CSV conversion is performed.

Usage

```
lb_copy_from_arrow(conn, value, table)
```

Arguments

conn	An lb_connection.
value	An Arrow Table, RecordBatch, or object accepted by <code>arrow::as_record_batch()</code> .
table	Existing target node table name.

Value

The COPY query's lb_result.

lb_copy_from_csv	<i>Load a CSV file into a LadybugDB table</i>
------------------	---

Description

Load a CSV file into a LadybugDB table

Usage

```
lb_copy_from_csv(conn, path, table, header = TRUE, delim = ",", ...)
```

Arguments

conn	An lb_connection.
path	Path to a CSV file.
table	Existing target table name.
header	Does the file contain a header?
delim	One-character delimiter.
...	Named COPY options. Values must be scalar logical, numeric, or character values; option names are validated identifiers.

Value

An lb_result.

lb_copy_from_df	<i>Load an R data frame into a LadybugDB table</i>
-----------------	--

Description

Uses native Arrow transport when the optional arrow package is available. Otherwise it uses the explicit CSV loader as a compatibility fallback.

Usage

```
lb_copy_from_df(conn, df, table)
```

Arguments

conn	An lb_connection.
df	A data frame.
table	Existing target table name.

Value

An lb_result.

lb_database	<i>Open a LadybugDB database</i>
-------------	----------------------------------

Description

Creates or opens a LadybugDB database. LadybugDB v0.20.4 can read the storage formats created by v0.15.x. For incompatible future formats, export with the original engine and import into a fresh database.

Usage

```
lb_database(path = ":memory:", read_only = FALSE, config = list())
```

Arguments

path	File-system path, or ":memory:" for an ephemeral database.
read_only	Open in read-only mode?
config	Named list of LadybugDB system configuration overrides. Valid names are buffer_pool_size, max_num_threads, enable_compression, max_db_size, auto_checkpoint, checkpoint_threshold, throw_on_wal_replay_failure, enable_checksums, enable_multi_writes, and enable_default_hash_index (thread_qos is also accepted on macOS).

Value

An lb_database object.

lb_execute	<i>Execute a Cypher query</i>
------------	-------------------------------

Description

Sends a Cypher query string to LadybugDB and returns an lb_result object. For DDL or DML statements (CREATE TABLE, CREATE, MERGE, ...) the result contains metadata; for MATCH/RETURN it contains rows.

Usage

```
lb_execute(conn, query, parameters = NULL)
```

Arguments

conn	An lb_connection object from lb_connection() .
query	A single character string containing a Cypher query.
parameters	A named list of query parameters for parameterised queries. NULL (default) means no parameters.

Value

An object of class `lb_result`.

Examples

```
## Not run:
db  <- lb_database(":memory:")
conn <- lb_connection(db)
lb_execute(conn, "CREATE NODE TABLE Person (name STRING, PRIMARY KEY(name))")
lb_execute(conn, "CREATE (:Person {name: $name})",
           parameters = list(name = "Alice"))

## End(Not run)
```

lb_fetch
Fetch rows from a LadybugDB result

Description

Fetch rows from a LadybugDB result

Usage

```
lb_fetch(result, n = 10000)
```

```
lb_has_next(result)
```

```
lb_reset(result)
```

Arguments

`result` An `lb_result`.

`n` Maximum rows to fetch. Inf fetches the remainder.

Value

A data frame. Repeated calls advance the result cursor.

lb_fetch_arrow	<i>Fetch a chunk as an Arrow Table</i>
----------------	--

Description

Fetch a chunk as an Arrow Table

Usage

```
lb_fetch_arrow(result, n = Inf)
```

Arguments

result	An lb_result.
n	Maximum rows to fetch. Inf fetches all remaining rows.

Value

An arrow::Table.

lb_list_extensions	<i>Manage LadybugDB extensions</i>
--------------------	------------------------------------

Description

These are thin wrappers around LadybugDB's extension Cypher commands. Remote installation and updating are unavailable in the CRAN build because those operations require a native networking stack. Compatible extensions installed outside R can still be loaded with lb_load_extension().

Usage

```
lb_list_extensions(conn)

lb_install_extension(conn, name)

lb_load_extension(conn, name)

lb_update_extension(conn, name)
```

Arguments

conn	An lb_connection.
name	Official extension name.

Value

lb_list_extensions() returns a data frame; the mutation helpers return an lb_result.

lb_next_result	<i>Retrieve the next result from a multi-statement query</i>
----------------	--

Description

Retrieve the next result from a multi-statement query

Usage

```
lb_next_result(result)
```

Arguments

result	An lb_result.
--------	---------------

Value

The next lb_result, or NULL.

lb_query	<i>Execute a Cypher query and return a data frame</i>
----------	---

Description

Convenience wrapper around [lb_execute\(\)](#) that immediately converts the result to an R data.frame. Equivalent to `as.data.frame(lb_execute(conn, query, ...))`.

Usage

```
lb_query(conn, query, parameters = NULL, ...)
```

Arguments

conn	An lb_connection object from lb_connection() .
query	A single character string containing a Cypher query.
parameters	A named list of query parameters for parameterised queries. NULL (default) means no parameters.
...	Passed to as.data.frame.lb_result() .

Value

A data.frame.

Examples

```
## Not run:
db  <- lb_database(":memory:")
conn <- lb_connection(db)
lb_execute(conn, "CREATE NODE TABLE Person (name STRING, age INT64, PRIMARY KEY(name))")
lb_execute(conn, "CREATE (:Person {name: 'Alice', age: 30})")
df <- lb_query(conn, "MATCH (p:Person) RETURN p.name, p.age")

## End(Not run)
```

lb_query_summary	<i>Return query timing information</i>
------------------	--

Description

Return query timing information

Usage

```
lb_query_summary(result)
```

Arguments

result An lb_result.

Value

A list with compilation and execution times in milliseconds.

lb_result_info	<i>Inspect result metadata</i>
----------------	--------------------------------

Description

Inspect result metadata

Usage

```
lb_result_info(result)
```

Arguments

result An lb_result.

Value

A named list.

lb_set_timeout	<i>Configure and interrupt queries</i>
----------------	--

Description

Configure and interrupt queries

Usage

```
lb_set_timeout(conn, milliseconds)
```

```
lb_interrupt(conn)
```

Arguments

conn	An lb_connection.
milliseconds	Non-negative timeout in milliseconds. Zero disables the timeout.

Value

Invisible conn.

with_lb_connection	<i>Run code with an automatically closed LadybugDB connection</i>
--------------------	---

Description

Run code with an automatically closed LadybugDB connection

Usage

```
with_lb_connection(path = ":memory:", code, ..., num_threads = NULL)
```

Arguments

path	Database path.
code	Code evaluated with conn bound to the open connection.
...	Passed to lb_database() .
num_threads	Passed to lb_connection() .

Value

The value of code.

Index

`as.data.frame()`, 5
`as.data.frame.lb_result`, 2
`as.data.frame.lb_result()`, 15
`as_arrow_table`, 3
`as_igraph`, 3
`as_tbl_graph`, 4
`as_tibble`, 5

`dbBegin`, LadybugConnection-method
(LadybugDB-DBI-methods), 6
`dbBind`, LadybugResult-method
(LadybugDB-DBI-methods), 6
`dbClearResult`, LadybugResult-method
(LadybugDB-DBI-methods), 6
`dbColumnInfo`, LadybugResult-method
(LadybugDB-DBI-methods), 6
`dbCommit`, LadybugConnection-method
(LadybugDB-DBI-methods), 6
`dbConnect`, LadybugDriver-method
(LadybugDB-DBI-methods), 6
`dbDataType`, LadybugDriver-method
(LadybugDB-DBI-methods), 6
`dbDisconnect`, LadybugConnection-method
(LadybugDB-DBI-methods), 6
`dbExecute`, LadybugConnection, character-method
(LadybugDB-DBI-methods), 6
`dbExistsTable`, LadybugConnection, character-method
(LadybugDB-DBI-methods), 6
`dbFetch`, LadybugResult-method
(LadybugDB-DBI-methods), 6
`dbGetInfo`, LadybugConnection-method
(LadybugDB-DBI-methods), 6
`dbGetInfo`, LadybugDriver-method
(LadybugDB-DBI-methods), 6
`dbGetInfo`, LadybugResult-method
(LadybugDB-DBI-methods), 6
`dbGetQuery`, LadybugConnection, character-method
(LadybugDB-DBI-methods), 6
`dbGetRowCount`, LadybugResult-method
(LadybugDB-DBI-methods), 6
`dbGetRowsAffected`, LadybugResult-method
(LadybugDB-DBI-methods), 6
`dbHasCompleted`, LadybugResult-method
(LadybugDB-DBI-methods), 6
`DBI::DBI`-package, 6
`dbIsReadOnly`, LadybugConnection-method
(LadybugDB-DBI-methods), 6
`dbIsValid`, LadybugConnection-method
(LadybugDB-DBI-methods), 6
`dbIsValid`, LadybugDriver-method
(LadybugDB-DBI-methods), 6
`dbIsValid`, LadybugResult-method
(LadybugDB-DBI-methods), 6
`dbListFields`, LadybugConnection, character-method
(LadybugDB-DBI-methods), 6
`dbListTables`, LadybugConnection-method
(LadybugDB-DBI-methods), 6
`dbQuoteIdentifier`, LadybugConnection, character-method
(LadybugDB-DBI-methods), 6
`dbQuoteString`, LadybugConnection, character-method
(LadybugDB-DBI-methods), 6
`dbRollback`, LadybugConnection-method
(LadybugDB-DBI-methods), 6
`dbSendQuery`, LadybugConnection, character-method
(LadybugDB-DBI-methods), 6
`dbSendStatement`, LadybugConnection, character-method
(LadybugDB-DBI-methods), 6
`dbUnloadDriver`, LadybugDriver-method
(LadybugDB-DBI-methods), 6

`format.lb_interval`, 5

`igraph::graph_from_data_frame()`, 3
`igraph::igraph`, 4

Ladybug, 6
`Ladybug()`, 6
LadybugConnection-class, 6
LadybugDB-DBI-methods, 6
`ladybugdb_install`, 7

- ladybugdb_is_installed, 7
- ladybugdb_version, 8
- LadybugDriver-class, 8
- LadybugResult-class, 8
- lb_begin, 9
- lb_close, 9
- lb_commit (lb_begin), 9
- lb_connection, 10
- lb_connection(), 12, 15, 17
- lb_copy_from_arrow, 10
- lb_copy_from_csv, 11
- lb_copy_from_df, 11
- lb_database, 12
- lb_database(), 17
- lb_execute, 12
- lb_execute(), 15
- lb_fetch, 13
- lb_fetch(), 2
- lb_fetch_arrow, 14
- lb_has_next (lb_fetch), 13
- lb_install_extension
 - (lb_list_extensions), 14
- lb_interrupt (lb_set_timeout), 17
- lb_list_extensions, 14
- lb_load_extension (lb_list_extensions),
14
- lb_next_result, 15
- lb_query, 15
- lb_query_summary, 16
- lb_reset (lb_fetch), 13
- lb_result_info, 16
- lb_rollback (lb_begin), 9
- lb_set_timeout, 17
- lb_update_extension
 - (lb_list_extensions), 14
- show, LadybugConnection-method
 - (LadybugDB-DBI-methods), 6
- show, LadybugDriver-method
 - (LadybugDB-DBI-methods), 6
- show, LadybugResult-method
 - (LadybugDB-DBI-methods), 6
- tidygraph::tbl_graph(), 4
- with_lb_connection, 17