

Package ‘ERDbuilder’

July 28, 2026

Type Package

Title Entity Relationship Diagrams Builder

Version 2.0.0

Description Build entity relationship diagrams (ERD) to specify the nature of the relationship between tables in a database.

License GPL (>= 3)

Encoding UTF-8

URL <https://github.com/gbasulto/ERDbuilder>,
<https://gbasulto.github.io/ERDbuilder/>

BugReports <https://github.com/gbasulto/ERDbuilder/issues>

Depends R (>= 4.1.0)

Imports DiagrammeR, dplyr

Suggests knitr, rmarkdown, gt, testthat (>= 3.0.0), clipr

VignetteBuilder knitr

Config/testthat/edition 3

Config/roxygen2/version 8.0.0

NeedsCompilation no

Author Jonathan Wood [aut] (ORCID: <<https://orcid.org/0000-0003-0131-6384>>),
Guillermo Basulto-Elias [aut, cre] (ORCID:
<<https://orcid.org/0000-0002-5205-2190>>)

Maintainer Guillermo Basulto-Elias <basulto@iastate.edu>

Repository CRAN

Date/Publication 2026-07-28 03:10:02 UTC

Contents

create_erd	2
format_relationships	4
is_valid_erd	5

perform_join 6

render_erd 8

suggest_relationships 11

validate_erd 13

Index 15

create_erd	Create ERD Object
------------	-------------------

Description

This function serves as a constructor for an Entity-Relationship Diagram (ERD) object. This object encapsulates both the data frames representing the entities and the relationships between these entities. The function takes as its arguments a list of data frames and a list of relationships and returns a list object of class "ERD".

Usage

```
create_erd(df_list, relationships, validate = TRUE)
```

Arguments

df_list	A named list of data frames, where each data frame represents an entity in the ERD. The names of the list elements correspond to the names of the entities.
relationships	A nested named list describing the relationships between entities. The top-level names in this list should correspond to the names in df_list. Each element of this list is itself a list, describing relationships that the corresponding entity has with other entities. The list of acceptable values is specified in "Details."
validate	Logical. If 'TRUE', validate the ERD structure before returning the object. Observed data cardinalities are not checked here.

Details

Possible values in each relationship element of the list include:

- "||" which indicates one and only one
- ">|" which indicates one or more (left table)
- "|<" which indicates one or more (right table)
- ">0" which indicates zero or more (left table)
- "0<" which indicates zero or more (right table)
- "|0" which indicates zero or one (left table)
- "0|" which indicates zero or one (right table)

It is imperative that the names used in df_list and relationships are consistent, as these are used for creating the ERD object and for subsequent operations like rendering and performing joins. Users can effortlessly encapsulate the data and relationships pertaining to an ERD into a single R object with this function, thereby facilitating downstream operations like rendering and joining.

Value

An object of class "ERD", which is a named list containing two elements:

data_frames Named list of data frames identical to `df_list`.

relationship Named list of relationships identical to `relationships`.

The class attribute of this list is set to "ERD".

Examples

```
# Load Packages -----

library(ERDbuilder)
library(dplyr)

# Define entities -----

students_tbl <- data.frame(
  st_id = c("hu1", "de2", "lo3"),
  dep_id = c("water", "evil", "values"),
  student = c("Huey", "Dewey", "Louie"),
  email = c("hubert.duck", "dewfort.duck", "llewellyn.duck"),
  dob = c("04-15", "04-15", "04-15")
)

courses_tbl <- data.frame(
  crs_id = c("water101", "evil205", "water202"),
  fac_id = c("02do", "03pe", "04mi"),
  dep_id = c("water", "evil", "water"),
  course = c("Swimming", "Human-chasing", "Dives")
)

enrollment_tbl <- data.frame(
  crs_id = c("water101", "evil205", "evil205", "water202"),
  st_id = c("hu1", "hu1", "de2", "de2"),
  final_grade = c("B", "A", "A", "F")
)

department_tbl <- data.frame(
  dep_id = c("water", "evil", "values"),
  department = c("Water activities", "Evil procurement", "Good values")
)

faculty_tbl <- data.frame(
  faculty_name = c("Scrooge McDuck", "Donald", "Pete", "Mickey"),
  title = c("Emeritus", "Full", "Assistant", "Full"),
  fac_id = c("01sc", "02do", "03pe", "04mi"),
  dep_id = c("water", "water", "evil", "values")
)

head(students_tbl)
```

```

head(courses_tbl)
head(enrollment_tbl)
head(department_tbl)
head(faculty_tbl)

## Define relationships-----
relationships <- list(
  courses = list(
    enrollment = list(crs_id = "crs_id", relationship = c("|<", "|<")),
    department = list(dep_id = "dep_id", relationship = c(">|", "|<")),
    faculty = list(fac_id = "fac_id", relationship = c(">0", "|<"))
  ),
  enrollment = list(
    students = list(st_id = "st_id", relationship = c(">0", "|<"))
  ),
  students = list(
    department = list(dep_id = "dep_id", relationship = c(">|", "|<"))
  ),
  faculty = list(
    department = list(dep_id = "dep_id", relationship = c(">|", "|<"))
  )
)

## Create ERD object
erd_object <-
  create_erd(
    list(
      students = students_tbl,
      courses = courses_tbl,
      enrollment = enrollment_tbl,
      department = department_tbl,
      faculty = faculty_tbl
    ),
    relationships
  )

## Render ERD -----
render_erd(erd_object, label_distance = 0, label_angle = 15, n = 20)

```

format_relationships *Format Relationships as R Code*

Description

Format Relationships as R Code

Usage

```
format_relationships(relationships, object_name = "relationships")
```

Arguments

`relationships` A relationships list, usually created by `[suggest_relationships()]`.
`object_name` Name to use on the left-hand side of the generated code.

Value

A character string containing R code.

is_valid_erd	<i>Test Whether an ERD Is Valid</i>
--------------	-------------------------------------

Description

Test Whether an ERD Is Valid

Usage

```
is_valid_erd(  
  erd_object,  
  check_data = TRUE,  
  strict = FALSE,  
  allow_incomplete = FALSE  
)
```

Arguments

`erd_object` An object created by `[create_erd()]`.
`check_data` Logical. Check observed cardinalities if 'TRUE'.
`strict` Logical. Treat warnings as invalid if 'TRUE'.
`allow_incomplete` Logical. If 'TRUE', unfinished relationship specifications are reported as warnings rather than errors. This is useful while progressively constructing and rendering an ERD.

Value

A single logical value.

perform_join	<i>Use inner join (unless the other is specified)</i>
--------------	---

Description

The `perform_join` function uses an inner join unless the user specifies the join type.

Usage

```
perform_join(erd_object, tables_to_join, specified_joins = NULL)
```

Arguments

- | | |
|------------------------------|---|
| <code>erd_object</code> | An object of class "ERD", which encapsulates the data frames and the relationships between them. This object is generated using the create_erd function. |
| <code>tables_to_join</code> | A character vector listing the names of tables to join. The first table in this list serves as the main table to which subsequent tables are joined. The tables are joined in the order specified and utilize the relationships defined with the first table. |
| <code>specified_joins</code> | An optional named list where each element's name corresponds to a table in <code>tables_to_join</code> and the value specifies the type of join to perform with that table. The default value is <code>NULL</code> , which activates automated mode (which uses inner joins). |

Details

This orchestrates the joining of multiple tables based on a specified Entity-Relationship Diagram (ERD) object. This function extracts the relationships and join criteria defined within the ERD object and executes the appropriate join operations using R's `dplyr` package.

The function can operate in two modes: automated and user-specified joins. In automated mode, join types are determined by the relationship symbols in the ERD object. In user-specified mode, the types of joins are explicitly provided by the user.

Implementation Details:

- Join Variables: For each pair of tables to be joined, the function extracts the relevant join variables from the ERD object.
- Join Type: Depending on the relationship symbol associated with each pair of tables, the function decides whether to perform an inner join or a left join. This decision is implemented by dynamically invoking the corresponding `dplyr` function ([inner_join](#) or [left_join](#)).
- Aggregation: The function uses [do.call](#) to dynamically execute the appropriate join operation, accumulating the result in the `main_table` variable, which is ultimately returned.

Notes:

- The function iteratively applies the join operations, using the first table in `tables_to_join` as the main table.

- The join operations are performed in the order specified in `tables_to_join`.
- When `specified_joins` is `NULL`, the function operates in automated mode, determining the type of join based on relationship symbols.
- The names in `specified_joins` should match the table names in `tables_to_join` for user-specified mode to function correctly.

Value

A data frame resulting from the join operations conducted between the specified tables, consistent with the relationships indicated in the ERD object. Additionally, the types of joins used are printed to the console.

Examples

```
# Load Packages -----

library(ERDbuilder)
library(dplyr)

# Define entities -----

students_tbl <- data.frame(
  st_id = c("hu1", "de2", "lo3"),
  dep_id = c("water", "evil", "values"),
  student = c("Huey", "Dewey", "Louie"),
  email = c("hubert.duck", "dewfort.duck", "llewellyn.duck"),
  dob = c("04-15", "04-15", "04-15")
)

courses_tbl <- data.frame(
  crs_id = c("water101", "evil205", "water202"),
  fac_id = c("02do", "03pe", "04mi"),
  dep_id = c("water", "evil", "water"),
  course = c("Swimming", "Human-chasing", "Dives")
)

enrollment_tbl <- data.frame(
  crs_id = c("water101", "evil205", "evil205", "water202"),
  st_id = c("hu1", "hu1", "de2", "de2"),
  final_grade = c("B", "A", "A", "F")
)

department_tbl <- data.frame(
  dep_id = c("water", "evil", "values"),
  department = c("Water activities", "Evil procurement", "Good values")
)

faculty_tbl <- data.frame(
  faculty_name = c("Scrooge McDuck", "Donald", "Pete", "Mickey"),
  title = c("Emeritus", "Full", "Assistant", "Full"),
```

```

    fac_id = c("01sc", "02do", "03pe", "04mi"),
    dep_id = c("water", "water", "evil", "values")
  )

head(students_tbl)
head(courses_tbl)
head(enrollment_tbl)
head(department_tbl)
head(faculty_tbl)

## Define relationships-----
relationships <- list(
  courses = list(
    enrollment = list(crs_id = "crs_id", relationship = c("||", "|<")),
    department = list(dep_id = "dep_id", relationship = c(">|", "||")),
    faculty = list(fac_id = "fac_id", relationship = c(">0", "||"))
  ),
  enrollment = list(
    students = list(st_id = "st_id", relationship = c(">0", "||"))
  ),
  students = list(
    department = list(dep_id = "dep_id", relationship = c(">|", "||"))
  ),
  faculty = list(
    department = list(dep_id = "dep_id", relationship = c(">|", "||"))
  )
)

## Create ERD object
erd_object <-
  create_erd(
    list(
      students = students_tbl,
      courses = courses_tbl,
      enrollment = enrollment_tbl,
      department = department_tbl,
      faculty = faculty_tbl
    ),
    relationships
  )

## Render ERD -----
render_erd(erd_object, label_distance = 0, label_angle = 15, n = 20)

# Perform joins -----

perform_join(erd_object, c("courses", "enrollment", "department"))

```


Description

The `render_erd` function graphically renders ERD using DiagrammeR, incorporating pseudo-nodes to depict traditional ERD notations such as cardinality. This function uses edge attributes to append text labels near the end of the edge lines. Specifically, the DiagrammeR label attribute is leveraged to include text labels at the ends of the edges, which effectively convey the intended cardinality and relationship information. This function constructs edge labels from two strings representing the left and right relationship attributes.

Usage

```
render_erd(erd_object, label_distance = 2.5, label_angle = 45, n = 10)
```

Arguments

<code>erd_object</code>	An object of class "ERD", generated using the <code>link{create_erd}</code> function. This object encapsulates the data frames representing the entities and the relationships between these entities.
<code>label_distance</code>	A numeric value that specifies the distance between the edge labels and the lines connecting the nodes. The default value is 2.5.
<code>label_angle</code>	A numeric value that specifies the angle at which the edge labels are displayed. The default value is 45 degrees.
<code>n</code>	The maximum number of rows in each table. The tables will add columns to show all of the variables in the tables such that there are only 'n' rows.

Details

This function is responsible for graphically rendering an Entity-Relationship Diagram (ERD) based on an object of class "ERD". This function leverages the DiagrammeR package to generate a graph that visually represents both the entities and the relationships contained within the ERD object. The entities are represented as nodes, and the relationships as edges. The `render_erd` function thus provides a robust mechanism for visually representing an ERD based on structured data within the R environment. By converting an ERD object into a graphical form, the function aids in a clearer understanding and communication of complex data relationships.

Nodes: Each entity (i.e., data frame) is represented as a node. The node label consists of the entity name and the attribute names within the entity.

Edges: Relationships between entities are represented as edges between the corresponding nodes. Labels at the ends of the edges indicate the type and cardinality of the relationship.

The `label_distance` and `label_angle` parameters control the presentation of edge labels in the ERD to minimize overlap and improve readability.

The function uses a for loop to iterate through the entities and relationships, constructing the necessary DiagrammeR code to render each element. The `nodesep` and `ranksep` parameters in the DiagrammeR code control the node spacing in the rendered ERD, making it easier to visualize complex ERDs.

Value

A DiagrammeR graph object representing the ERD.

Examples

```
# Load Packages -----

library(ERDbuilder)
library(dplyr)

# Define entities -----

students_tbl <- data.frame(
  st_id = c("hu1", "de2", "lo3"),
  dep_id = c("water", "evil", "values"),
  student = c("Huey", "Dewey", "Louie"),
  email = c("hubert.duck", "dewfort.duck", "llewellyn.duck"),
  dob = c("04-15", "04-15", "04-15")
)

courses_tbl <- data.frame(
  crs_id = c("water101", "evil205", "water202"),
  fac_id = c("02do", "03pe", "04mi"),
  dep_id = c("water", "evil", "water"),
  course = c("Swimming", "Human-chasing", "Dives")
)

enrollment_tbl <- data.frame(
  crs_id = c("water101", "evil205", "evil205", "water202"),
  st_id = c("hu1", "hu1", "de2", "de2"),
  final_grade = c("B", "A", "A", "F")
)

department_tbl <- data.frame(
  dep_id = c("water", "evil", "values"),
  department = c("Water activities", "Evil procurement", "Good values")
)

faculty_tbl <- data.frame(
  faculty_name = c("Scrooge McDuck", "Donald", "Pete", "Mickey"),
  title = c("Emeritus", "Full", "Assistant", "Full"),
  fac_id = c("01sc", "02do", "03pe", "04mi"),
  dep_id = c("water", "water", "evil", "values")
)

head(students_tbl)
head(courses_tbl)
head(enrollment_tbl)
head(department_tbl)
head(faculty_tbl)

## Define relationships-----
relationships <- list(
  courses = list(
    enrollment = list(crs_id = "crs_id", relationship = c("||", "<")),
```

```

    department = list(dep_id = "dep_id", relationship = c(">|", "||")),
    faculty = list(fac_id = "fac_id", relationship = c(">0", "||"))
  ),
  enrollment = list(
    students = list(st_id = "st_id", relationship = c(">0", "||"))
  ),
  students = list(
    department = list(dep_id = "dep_id", relationship = c(">|", "||"))
  ),
  faculty = list(
    department = list(dep_id = "dep_id", relationship = c(">|", "||"))
  )
)

## Create ERD object
erd_object <-
  create_erd(
    list(
      students = students_tbl,
      courses = courses_tbl,
      enrollment = enrollment_tbl,
      department = department_tbl,
      faculty = faculty_tbl
    ),
    relationships)

## Render ERD -----
render_erd(erd_object, label_distance = 0, label_angle = 15, n = 20)

```

`suggest_relationships` *Suggest ERD Relationships From a List of Data Frames*

Description

This function inspects a named list of data frames and suggests relationships based on shared column names, ID-like columns, observed key coverage, and observed key uniqueness.

Usage

```

suggest_relationships(
  df_list,
  min_score = 0.65,
  max_composite_cols = 4,
  copy = FALSE,
  object_name = "relationships",
  quiet = FALSE
)

```

Arguments

<code>df_list</code>	A named list of data frames.
<code>min_score</code>	Minimum score required to keep a suggested relationship. Defaults to '0.65'.
<code>max_composite_cols</code>	Maximum number of columns allowed in a composite key candidate. Defaults to '4'.
<code>copy</code>	Logical. If 'TRUE', copy paste-ready R code to the clipboard using the 'clipr' package, if available.
<code>object_name</code>	Name to use when generating paste-ready R code. Defaults to "relationships".
<code>quiet</code>	Logical. If 'FALSE', print a short message about how many relationships were suggested.

Details

The returned object is a nested list in the same format expected by `[create_erd()]`. A data frame with diagnostic information is stored in the "suggestions" attribute.

Value

A nested named list of suggested relationships.

Examples

```
employees <- data.frame(
  emp_id = c(1, 2, 3),
  name = c("Alice", "Bob", "Charlie")
)

assignments <- data.frame(
  emp_id = c(1, 1, 2),
  task = c("A", "B", "C")
)

relationships <- suggest_relationships(
  list(
    employees = employees,
    assignments = assignments
  )
)

relationships
```

validate_erd	<i>Validate an ERD Object</i>
--------------	-------------------------------

Description

Checks an ERD object's structure, table and column references, relationship symbols, join-column types, and optionally the observed cardinalities.

Usage

```
validate_erd(  
  erd_object,  
  check_data = TRUE,  
  strict = FALSE,  
  allow_incomplete = FALSE  
)
```

Arguments

erd_object	An object created by [create_erd()].
check_data	Logical. Check observed cardinalities if 'TRUE'.
strict	Logical. Treat warnings as invalid if 'TRUE'.
allow_incomplete	Logical. If 'TRUE', unfinished relationship specifications are reported as warnings rather than errors. This is useful while progressively constructing and rendering an ERD.

Details

Structural problems are errors. Observed-data problems are warnings because a sample may not fully represent the intended database constraints.

Value

An object of class 'erd_validation' containing 'valid', 'issues', 'n_errors', 'n_warnings', and 'strict'.

Examples

```
departments <- data.frame(  
  dept_id = c(10, 20),  
  department = c("Research", "Operations")  
)  
  
employees <- data.frame(  
  employee_id = 1:3,  
  dept_id = c(10, 10, 20),  
  employee = c("Ana", "Ben", "Chen")  
)
```

```
)  
  
relationships <- list(  
  departments = list(  
    employees = list(  
      dept_id = "dept_id",  
      relationship = c("||", "0<")  
    )  
  )  
)  
  
erd <- create_erd(  
  list(departments = departments, employees = employees),  
  relationships  
)  
  
validate_erd(erd)
```

Index

`create_erd`, [2](#), [6](#)

`do.call`, [6](#)

`format_relationships`, [4](#)

`inner_join`, [6](#)

`is_valid_erd`, [5](#)

`left_join`, [6](#)

`perform_join`, [6](#)

`render_erd`, [8](#), [9](#)

`suggest_relationships`, [11](#)

`validate_erd`, [13](#)