HARD SOLUTION

Introduction

The `DBmaps` package provides tools for exploring, mapping, and merging tables in a relational database. This vignette demonstrates how to use the key functions in the package: `create.DB.map`, `multi.merge`, and `visualize.DB.map`.

Example:

Creating a Database Map (create.DB.map)

The `create.DB.map` function constructs a map of relationships between tables in a database. It specifies which tables can be joined, the linking variables, the type of join, and optional aggregations.

Creating a Database Map (`create.DB.map`)

The `create.DB.map()` function helps visualize relationships between tables in a database.

```
# Define example tables
table1 <- data.table(id = c(1, 2), value1 = c("A", "B"))
table2 <- data.table(id = c(2, 3), value2 = c("C", "D"))
table3 <- data.table(id = c(3, 4), value3 = c("E", "F"))

# Define relationships between tables
relationships <- list(
  list(table1 = "table1", table2 = "table2", key = "id", type =
"inner"),
  list(table1 = "table2", table2 = "table3", key = "id", type =
"left")
)

# ✅ Use a named list for tables
db_map <- create.DB.map(tables = list(
  table1 = table1,
  table2 = table2,
```

```
  table3 = table3
),
by = list(
    list(table1 = "table1", table2 = "table2", by.x = "id", by.y =
"id"),
    list(table1 = "table2", table2 = "table3", by.x = "id", by.y =
"id")
  )
)

# Print map summary
print(db_map)
```

OUTPUT

```
   table_x table_y can_join   by.x   by.y join_type  agg_x  agg_y
   <char>  <char>   <lgcl> <list> <list>    <char> <char> <char>
1:  table1  table2     TRUE     id     id     inner   <NA>   <NA>
2:  table2  table3     TRUE     id     id     inner   <NA>   <NA>
```

Merging Tables (`multi.merge`)

The `multi.merge()` function merges multiple tables based on specified keys and join types.

```
# Merge tables using inner and left joins
merged_result <- multi.merge(
  tables = list(
    table1 = table1,
    table2 = table2,
    table3 = table3
  ),
  by = "id",
  types = c("inner", "left")
)
```

"Initial Table:TABLE1"

```
      id value1
   <num> <char>
1:     1      A
2:     2      B
```

TABLE2

```
       id value2
    <num> <char>
1:     2      C
2:     3      D
```

"Result after this first merge: Inner Merge"

```
Key: <id>
      id value1 value2
   <num> <char> <char>
1:     2      B      C
```

[1] "Merging with table 3 using left join"

TABLE3

```
       id value3
    <num> <char>
 1:     3      E
 2:     4      F
```

```
[1] "Result after this merge:"
Key: <id>
      id value1 value2 value3
   <num> <char> <char> <char>
1:     2      B      C   <NA>
```

```
View the merged result

print(merged_result)
```

```
Key: <id>
      id value1 value2 value3
   <num> <char> <char> <char>
1:     2      B      C   <NA>
```

Visualizing Database Relationships (`visualize.DB.map`)

The `visualize.DB.map()` function creates a graphical representation of the database schema.

```
# Visualize database map
visualize.DB.map(db_map)
```