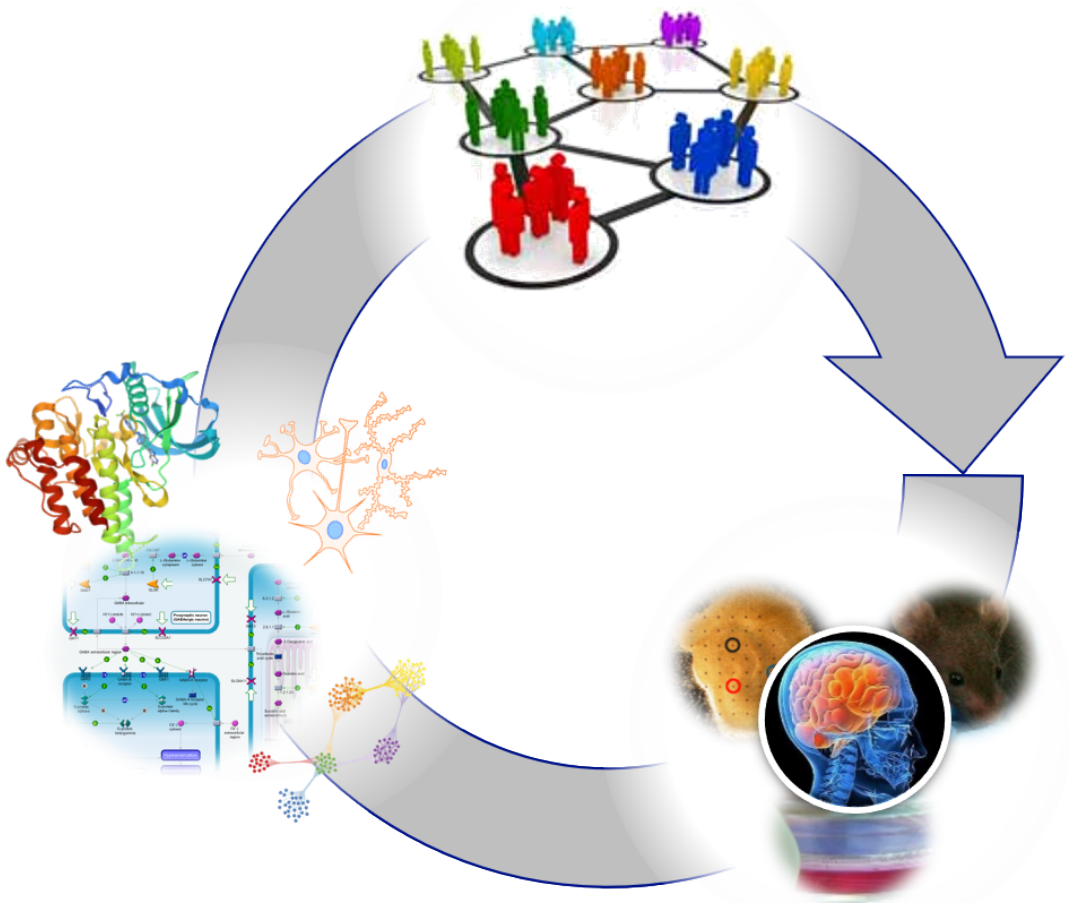
Managing and leveraging knowledge catalogs with TKCat

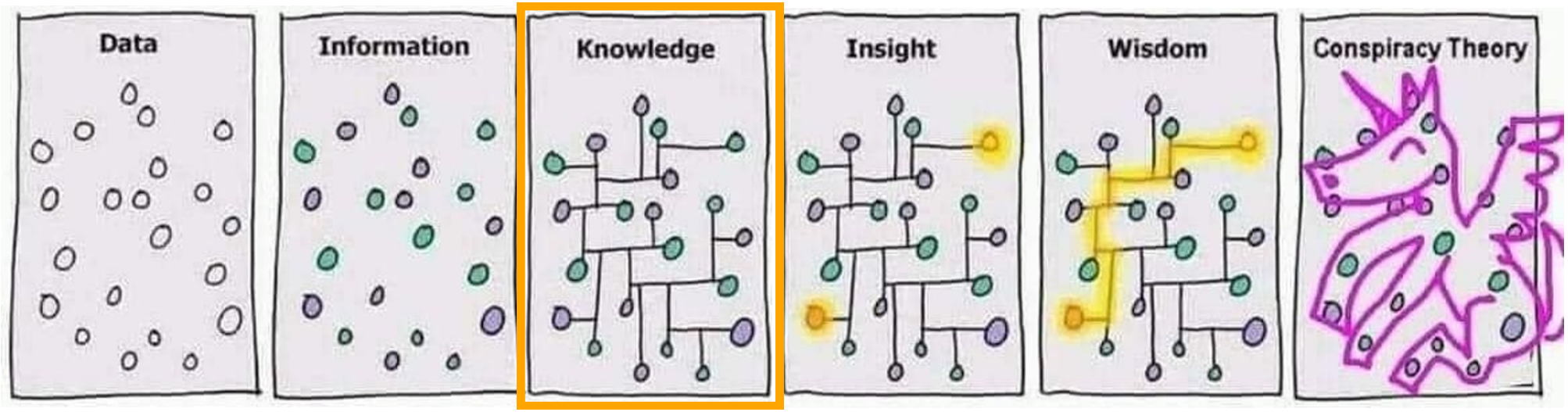
Patrice Godard | useR!2022 | 23 Jun 2022

Inspired by **patients.**
Driven by **science.**

Translational Bioinformatics at UCB



From data to wisdom



***Credit**: from twitter (unknown author)*

Expected features of the knowledge to manage

- Diverse concepts
- Connected concepts

- Tabular data
- To use in R and beyond

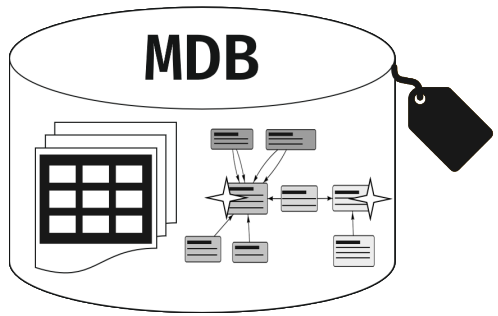
- Non monolytic knowledge: management of independent pieces
- Ready for integration on the basis of shared concepts
- Diverse implementation of shared concepts

- Potentially billions of records
- Tables to be used entirely or by subset
- No frequent updates but to be versioned
- Potential restriction of use

MDB: a Modeled Database for each knowledge resource

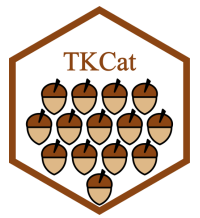
Features

- **Data**: tables and matrices
- **Data model**: Formal description of the data
- **Description**: general information about the data
- **Collections**: tables referring to key concepts
- **Implementation**: in memory, in files, in ClickHouse



TKCat

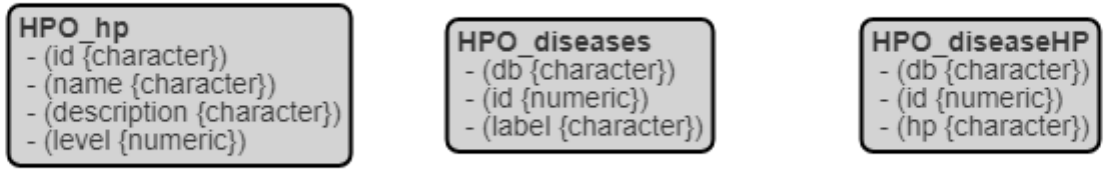
- **Tailored Knowledge Catalog**: a package for managing and using MDBs



Drafting a data model in R with ReDaMoR

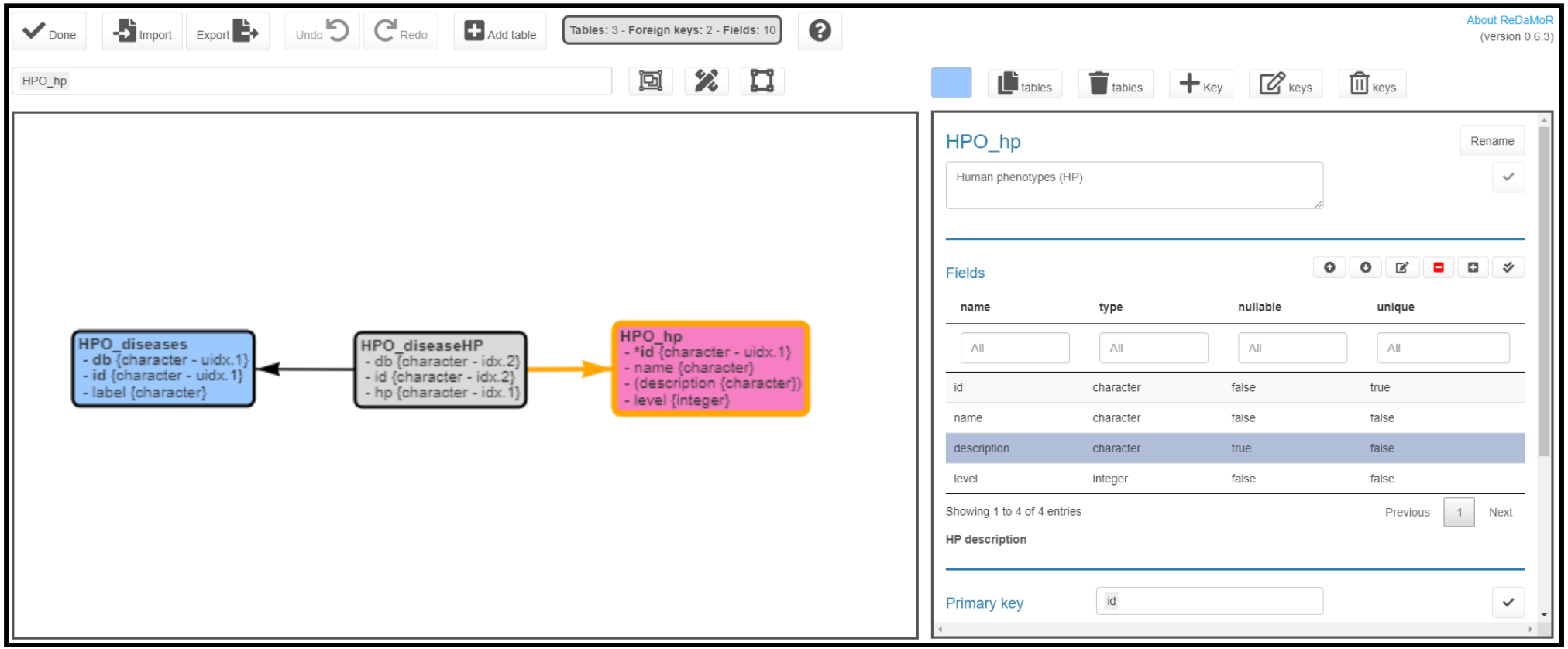
```r
library(readr)
hpo_data_dir <- system.file("examples/HPO-subset", package="ReDaMoR")
HPO_hp <- read_tsv(file.path(hpo_data_dir, "HPO_hp.txt"))
HPO_diseases <- read_tsv(file.path(hpo_data_dir, "HPO_diseases.txt"))
HPO_diseaseHP <- read_tsv(file.path(hpo_data_dir, "HPO_diseaseHP.txt"))
```

```r
library(ReDaMoR)
hpo_model <- df_to_model(HPO_hp, HPO_diseases, HPO_diseaseHP)
plot(hpo_model)
```



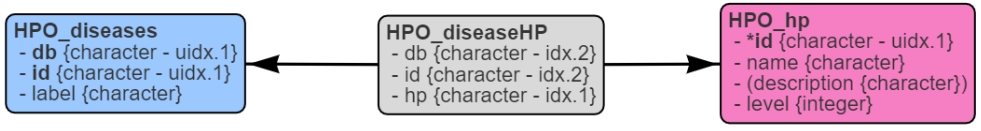
Creating a data model in R with ReDaMoR

```
hpo_model <- model_relational_data(hpo_model)
```



Confronting data to the model

```
confront_data(hpo_model, data=list(
   "HPO_hp"=HPO_hp,
   "HPO_diseaseHP"=HPO_diseaseHP,
   "HPO_diseases"=HPO_diseases
))
```




```
HPO_hp <- mutate(HPO_hp, level=as.integer(level))
HPO_diseases <- mutate(HPO_diseases, id=as.character(id))
HPO_diseaseHP <- mutate(HPO_diseaseHP, id=as.character(id))
confront_data(hpo_model, data=list(
   "HPO_hp"=HPO_hp,
   "HPO_diseaseHP"=HPO_diseaseHP,
   "HPO_diseases"=HPO_diseases
))
```

Confrontation with original data

FAILURE

Check configuration

- **Optional checks**: unique, not nullable, foreign keys
- **Maximum number of records**: Inf

HPO_hp

FAILURE

Field issues or warnings

- description: SUCCESS Missing values 117/500 = 23%
- level: FAILURE Unexpected "numeric"

HPO_diseases

FAILURE

Field issues or warnings

- id: FAILURE Unexpected "numeric"

HPO_diseaseHP

FAILURE

Field issues or warnings

- id: FAILURE Unexpected "numeric"

Confrontation with corrected data

SUCCESS

Check configuration

- **Optional checks**: unique, not nullable, foreign keys
- **Maximum number of records**: Inf

HPO_hp

SUCCESS

Field issues or warnings

- description: SUCCESS Missing values 117/500 = 23%

Creating and using an MDB with TKCat

MDB creation

```
library(TKCat)
hpo <- memoMDB(
   dataTables=list(
      "HPO_hp"=HPO_hp,
      "HPO_diseases"=HPO_diseases,
      "HPO_diseaseHP"=HPO_diseaseHP
   ),
   dataModel=hpo_model,
   dbInfo=list(
      name="miniHPO",
      title="Very small extract of the human pl
      description="For demonstrating ReDaMoR ar
      url="https://hpo.jax.org/app/",
      version="0.1",
      maintainer="Patrice Godard <patrice.godar
   )
)
```

Explore and retrieve information

```
db_info(hpo)
data_model(hpo)
```

```
hpo %>% select(HPO_diseases, HPO_diseaseHP)
hpo %>% pull(HPO_diseases) %>%
   head(3)
```

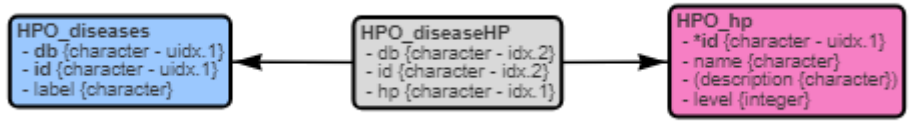
```
## # A tibble: 3 × 3
##   db       id    label
##   <chr>    <chr> <chr>
## 1 DECIPHER 15    NF1-microdeletion syndrome
## 2 DECIPHER 45    Xq28 (MECP2) duplication
## 3 DECIPHER 65    ATR-16 syndrome
```

Leverage the MDB data model: `filter`

```
dims(hpo) %>% select(name, nrow)
```

```
## # A tibble: 3 × 2
##   name           nrow
##   <chr>         <int>
## 1 HPO_hp          500
## 2 HPO_diseases   1903
## 3 HPO_diseaseHP  2594
```

```
data_model(hpo) %>% plot()
```

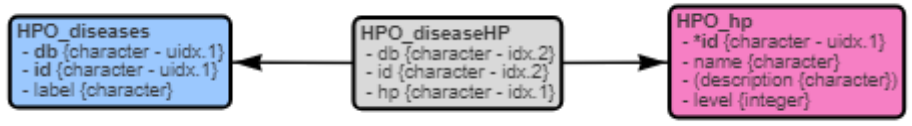



```
fhpo <- hpo %>% filter(HPO_hp=stringr::str_detect(description, "eye"))
```

```
fhpo %>% dims() %>% select(name, nrow)
```

```
## # A tibble: 3 × 2
##   name           nrow
##   <chr>         <int>
## 1 HPO_hp           21
## 2 HPO_diseaseHP   109
## 3 HPO_diseases     99
```

```
data_model(fhpo) %>% plot()
```

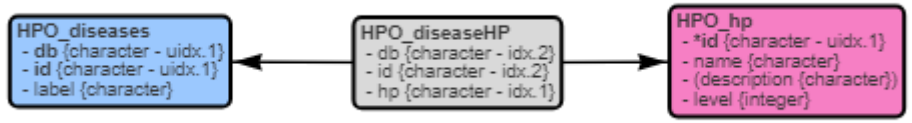


Leverage the MDB data model: `join`

```
dims(fhpo) %>% select(name, nrow)
```

```
## # A tibble: 3 × 2
##   name           nrow
##   <chr>         <int>
## 1 HPO_hp           21
## 2 HPO_diseaseHP   109
## 3 HPO_diseases     99
```

```
data_model(fhpo) %>% plot()
```

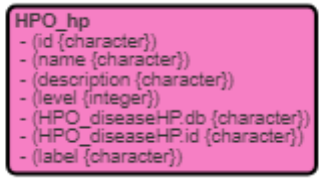



```
jhpo <- fhpo %>% join_mdb_tables(c("HPO_hp", "HPO_diseaseHP", "HPO_diseases"))
```

```
jhpo %>% dims() %>% select(name, nrow)
```

```
## # A tibble: 1 × 2
##   name    nrow
##   <chr>  <int>
## 1 HPO_hp   109
```

```
data_model(jhpo) %>% plot()
```



MDB implementations

In memory (`as_memoMDB()`)

- All the data loaded in R memory
- Fast but greedy
- Convenient for using whole tables

In files (`as_fileMDB()`)

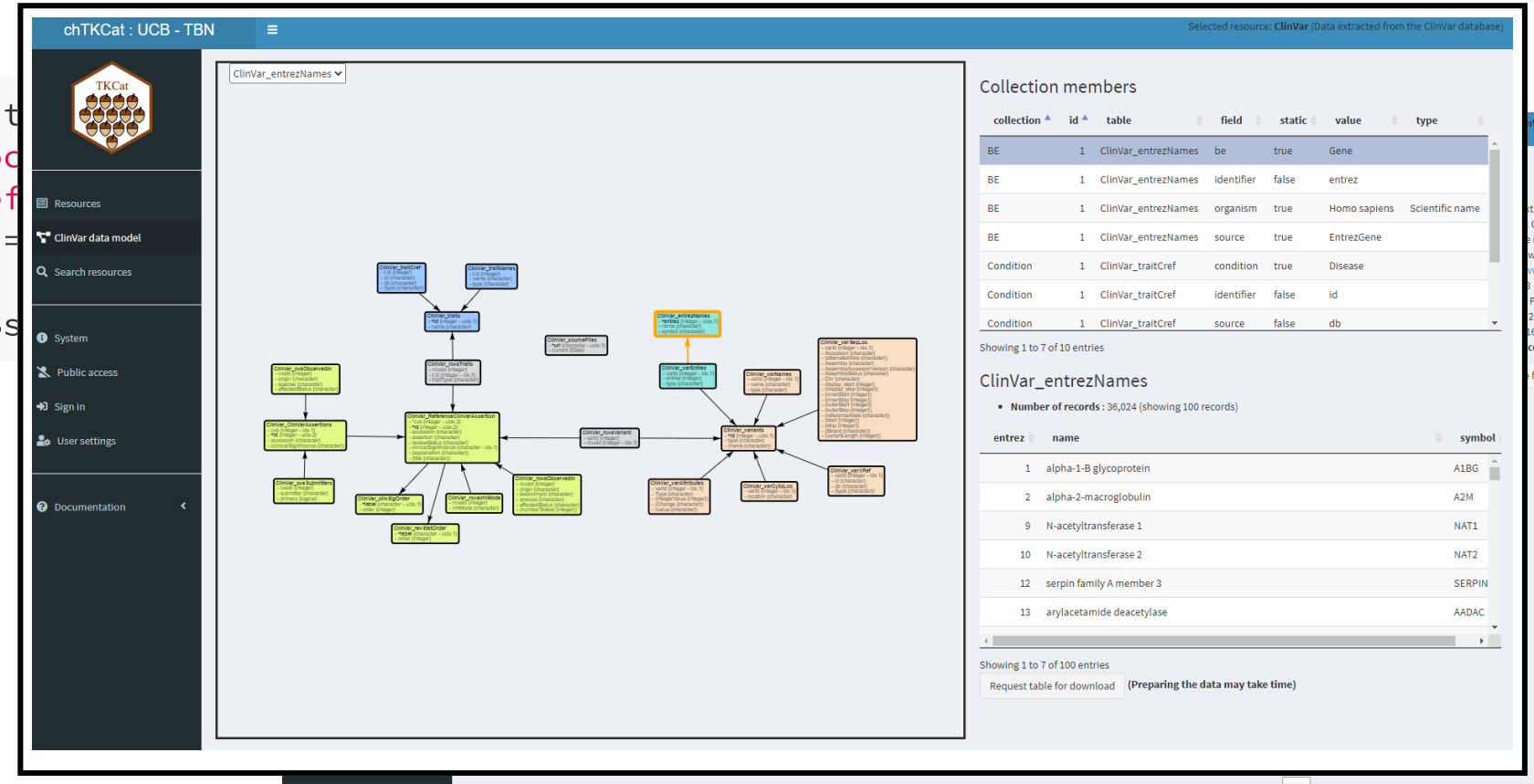
- Data in files until requested (`pull()`, `filter()`, ...)
- Not convenient for subsetting (slow)
- Convenient for archiving and sharing

In ClickHouse DBMS (`as_chMDB()`)

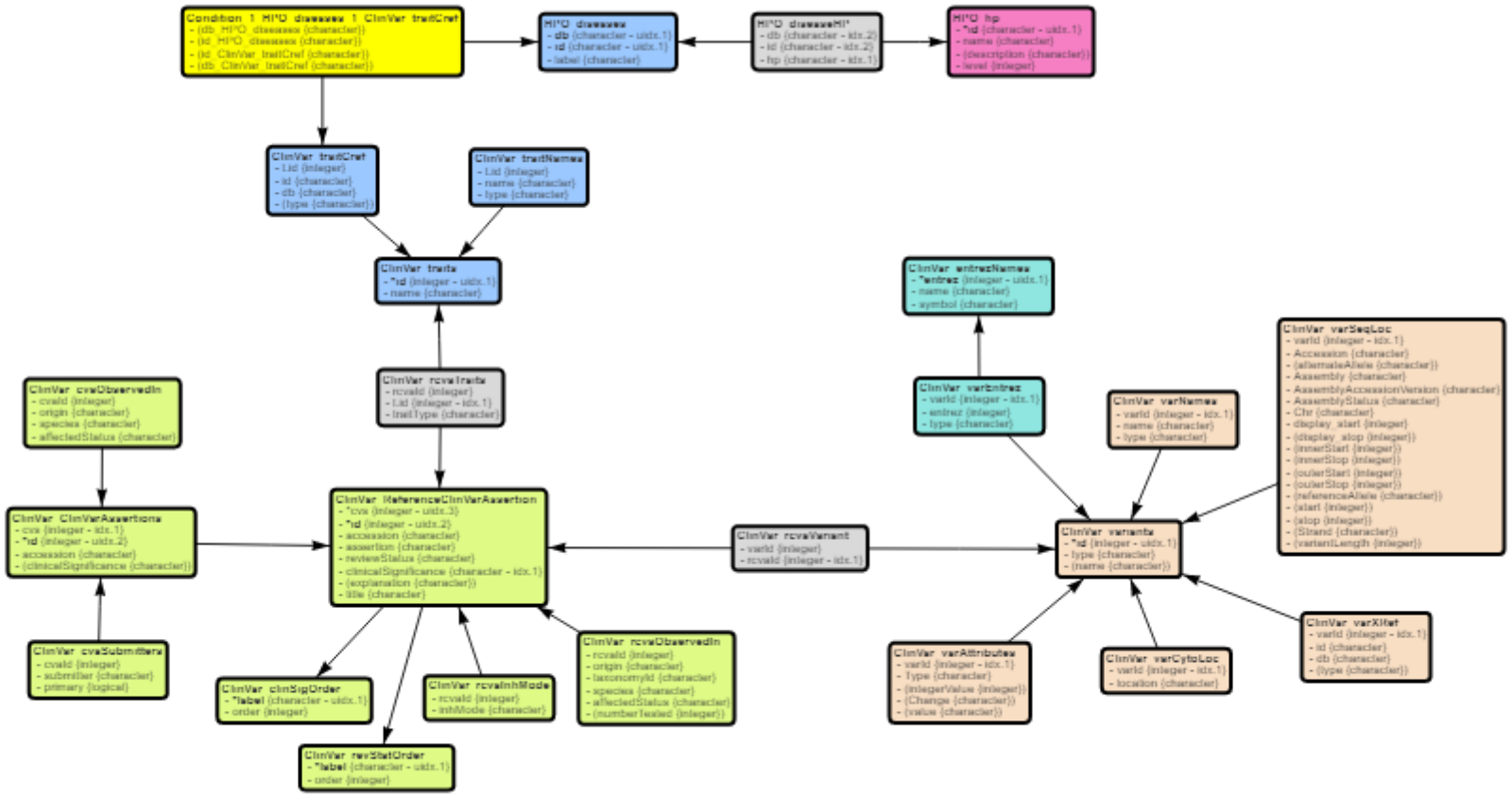
- Data in DBMS until requested (`pull()`, `filter()`, ...)
- Efficient to get subsets (`get_query()`)
- Convenient for sharing and managing access
- Versioning

TKCat: a data warehouse management system

```
k <- chTKCat
    host="loc
    user="def
    password=
)
explore_MDBs
```

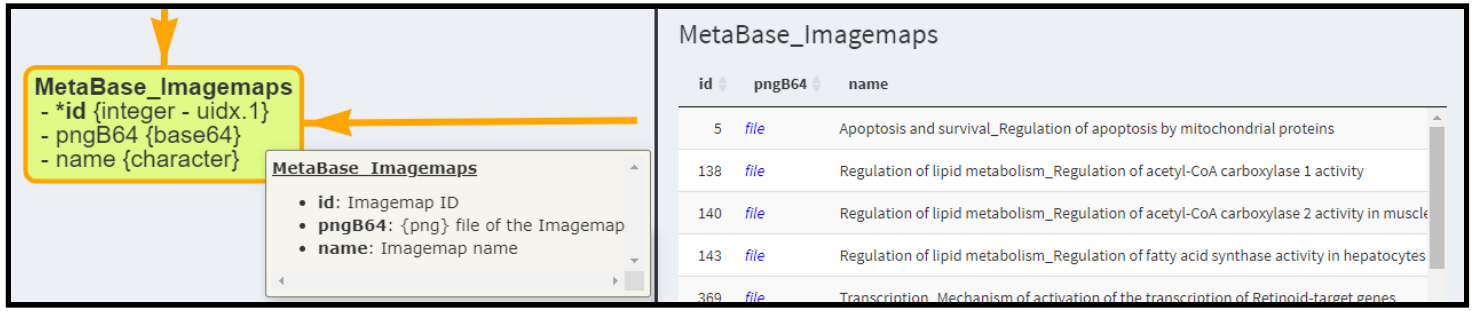


Merging MDBs with collections

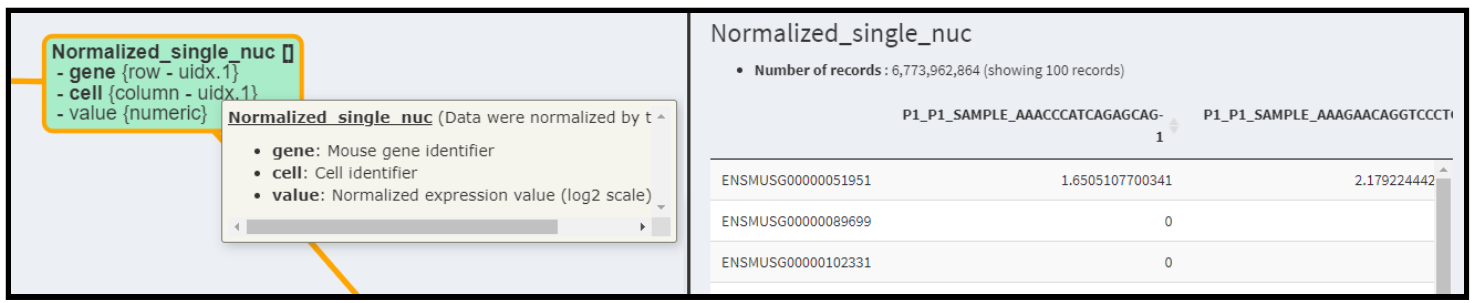


Supported data types

- character, numeric, integer, logic, Date, POSIXct (time)
- base64 (file)



- matrix and sparse matrix



Slides created with xaringan and xaringanthemer

https://patzaw.github.io/TKCat/useR2022/TKCat-useR2022-Patrice-Godard.html

Inspired by **patients.**
Driven by **science.**