Oxinfer onboarding

Danielle Newby, Martí Català, Edward Burn

2024-07-17T00:00:00+01:00

Table of contents

Preface

Onboarding to the oxinfer group

We've written this book for anyone interested in a working with databases mapped to the OMOP Common Data Model (CDM) in a tidyverse inspired approach. That is, human centered, consistent, composable, and inclusive (see https://design.tidyverse.org/unifying.html for more details on these principles).

New to the OMOP CDM? We'd recommend you pare this book with The Book of OHDSI

New to R? We recommend you compliment the book with R for data science

Citation

TO ADD

License

1 Connect to the database

1.1 Getting started

To connect to databases we will use DBI package and CDMConnector, you can find more information about both packages in their websites:

- DBI package website: https://dbi.r-dbi.org/
- CDMConnector package website: https://darwin-eu.github.io/CDMConnector/

Connect to database (standard way) and set up the environment

To connect to the database, you need to know some parameters of it: server_dbi, port, host, server, user, and password.

server_dbi is different for each one of the databases: e.g.

server_dbi <- “cdm_aurum_202106”

Port is the port used to connect to the database for the moment our databases use port 5432.

port <- “5432”

Host is the IP of the computer that contains the databases. For the moment the host of all our current databases is: 163.1.64.2

host <- “163.1.64.2”

Server is the combination of the host and the server_dbi with a slash “/” in between.

server <- “163.1.64.2/cdm_aurum_202106”

User and password are provided by the database administrator (e.g. Antonella), and they are personal and nontransferable. One of the first things that we will have to do is to change the password, from the default one (given by the administrator) to one of our choices (see STEP 2). The username and password are shared by all the databases that we have on the same port and server, so we need to change it only once.

user <- “…”

password <- “…”

STEP 0 Install the libraries

The libraries necessary to connect to a database are: DBI, RPostgres, dplyr, dbplyr, usethis, DatabaseConnector and here. They are all CRAN libraries so there should not be any problem installing them: if you have any problem please ask for help. To install them run the following commands:

STEP 1 Connect to a database for first time

Execute the following commands, make sure that you fill the parameters of the database accordingly (lines 7-11).

1.2 Load libraries

The following libraries will be used in this chapter: DBI, RPostgres, dplyr, dbplyr, usethis and here. If you do not have them installed you can install them with the following command:

```r
install.packages(c("DBI", "RPostgres", "dplyr", "dbplyr", "usethis", "here"))
```

library(“DBI”)

library(“RPostgres”)

1.3 Set connection details

1.4 Substitute the next 5 lines with the specifications of the database that you

1.5 want to connect

server_dbi <- “...”

user <- “...”

password <- “...”

port <- “...”

host <- “...”

1.6 Connect to the database

db <- dbConnect(RPostgres::Postgres(),

```
                    dbname = server_dbi,
```

```
port = port,
```

```
   host = host,
```

user = user,

```
    password = password)
```

To check that you are connected run the following commands:

library(“dplyr”)

tbl(db, sql(“SELECT * FROM public.person limit 1”))

If you are connected you should see something like this in your terminal:

ShapeBackground pattern

Description automatically generated

STEP 2 Change the password (only to be executed the first time a user connects)

For the new password make sure that you chose a strong password.

After connecting to the database with the temporary password, to change the password, you must execute the following command in the terminal:

dbGetQuery(db, “ALTER USER user WITH PASSWORD ‘new_password’ ”)

Example:

dbGetQuery(db, “ALTER USER martics WITH PASSWORD ‘12345678’ ”)

Disconnect from the database and reconnect with the new password to make sure that the password change was effective.

dbDisconnect(db)

password <- “new_password”

db <- dbConnect(RPostgres::Postgres(),

```
                    dbname = server_dbi,
```

```
 port = port,
```

host = host,

```
    user = user,

    password = password)
```

tbl(db, sql(“SELECT * FROM public.person limit 1”))

If everything worked, you must observe the same output that you observed before.

STEP 3 Set .Renviron

Including username, password and connection details in an R script is something that we must avoid at all costs. Otherwise, when we share our code or upload it in GitHub our account can be vulnerable. So NEVER include the connection details into an R script!

To avoid this potential problem we are going to store the connection details of our database in .Renviron file so that we can use them easily without sharing them.

How .Renviron works? This is a file where we write a list of variables with their values that we can access from any R script. For example, if I add to the file:

NAME = “marti”

When I execute the following command, I will obtain “marti” as a result:

Sys.getenv(“NAME”)

So this two commands would have the same output, but in the second case without having the same .Renviron file someone external won’t be able to read it.

x <- “marti”

x <- Sys.getenv(“NAME”)

To open and set the .Renviron file we have two options:

OPTION 1: open directly the file:

Graphical user interface, application

Description automatically generated

OPTION 2: write the following command in the console:

usethis::edit_r_environ()

Once you have it opened you must add the follow to the file:

DB_USER = “xxx” # Username for the database

DB_PASSWORD = “xxx” # Password for the database, the new one!!

DB_PORT = “5432”

DB_HOST = “163.1.64.2”

DB_SERVER_name_database = “163.1.64.2/xxx” # xxx depends on the database that you

DB_SERVER_DBI_name_database = “xxx” # want to access, see section 4.

Instead of DB_SERVER_DBI_name_database we can write a name that we can remember easily. For example, for CPRD AURUM May 2021 release I would write in .Renviron file:

DB_SERVER_DBI_aurum202106 = “cdm_aurum_202106”

And call it from the main script as:

server_dbi <- Sys.getenv(“DB_SERVER_DBI_aurum202106”)

But as said we can use any name that we want to use.

Important!! Every time that we change .Renviron file, R must be rebooted (Session / Restart R, or Ctrl+Shift+F10) so that the changes are effective.

STEP 4 Check that .Renviron setting works

After rebooting, execute the following code to check if you can connect to the data base:

library(“DBI”)

library(“RPostgres”)

1.7 use your own .Renviron “keys” to select the database that you want to access

server_dbi <- Sys.getenv(“DB_SERVER_DBI_name_database”)

user <- Sys.getenv(“DB_USER”)

password <- Sys.getenv(“DB_PASSWORD”)

port <- Sys.getenv(“DB_PORT”)

host <- Sys.getenv(“DB_HOST”)

db <- dbConnect(RPostgres::Postgres(),

```
                    dbname = server_dbi,

    port = port,

    host = host,

     user = user,

    password = password)
```

```
tbl(db, sql(“SELECT * FROM public.person limit 1”))
```

If you connected correctly to the database, you must observe the same output that you observed before.

Connect to the database (OHDSI way)

The connector used in OHDSI packages is different from the one used in the “standard way”. To create the connection used in OHDSI packages for your database you must use the following commands:

```
library(“DatabaseConnector”)

library(“here”)

server <- Sys.getenv(“DB_SERVER_name_database”)

user <- Sys.getenv(“DB_USER”)

password <- Sys.getenv(“DB_PASSWORD”)

port<- Sys.getenv(“DB_PORT”)

host <- Sys.getenv(“DB_HOST”)

connectionDetails <- DatabaseConnector::downloadJdbcDrivers(“postgresql”,

                                                       here::here())

connectionDetails <- DatabaseConnector::createConnectionDetails(

                            dbms = "postgresql",
```

server = server,

user = user,

password = password,

port = port,

pathToDriver = here::here())

Important considerations

The SQL dialect of our database is: postgresql. This will be needed in some functions (manly OHDSI packages) as an input.

targetDialect <- “postgresql”

IMPORTANT!!

The database works with schemas each schema contains different tables and data. The two schemas that we have access to are:

Public: contains the mapped data and the vocabularies. This schema should NEVER be used to write.

This should be automatically achieved if people have the correct privileges and do not use WebAPI privileges.

Results: should be the only schema where the users have read and write privileges. It contains the cohorts that we create as well as the tables that we upload.

cdm_database_schema <- “public” vocabulary_database_schema <- “public” results_database_schema<- “results”

Available databases

For the moment we have access to 4 databases:

cdm_aurum_202106 - CPRD AURUM 2021_06 release

p20_059_cdm_aurum - CPRD AURUM for protocol 20_059.

p20_000211_cdm_aurum - CPRD AURUM for protocol 20_000211.

cdmgold202007 - CPRD GOLD 2020_07 release.

A suggestion of data to add to the .Renviron file would be:

DB_SERVER_aurum_202106 = “163.1.64.2/cdm_aurum_202106”

DB_SERVER_DBI_aurum_202106 = “cdm_aurum_202106”

DB_SERVER_aurum_protocol_059 = “163.1.64.2/p20_059_cdm_aurum”

DB_SERVER_DBI_ aurum_protocol_059 = “p20_059_cdm_aurum”

DB_SERVER_aurum_protocol_211 = “163.1.64.2/p20_000211_cdm_aurum”

DB_SERVER_DBI_aurum_protocol_211 = “p20_000211_cdm_aurum”

DB_SERVER_gold_202007 = “163.1.64.2/cdmgold202007”

DB_SERVER_DBI_gold_202007 = “cdmgold202007”

References