RStudio makes Oracle accessibility from R easier via odbc and connections Pane[1].
Personally, I find it's not so easy.
As it finally works for me, I will detail some snippets here.

After tens of try it seems good to share some tricks[2]. This blog post is also a notepad for me.

Oracle and R configuration is a step where we potentially waste a lot of time.

Many things can cause oracle and R not to work at all:

- it depends on which client is installed (32b, 64b ?)
- wether odbc driver is correctly installed or not
- you have to dissect tnsnames.ora
- investigate on many ORA error's
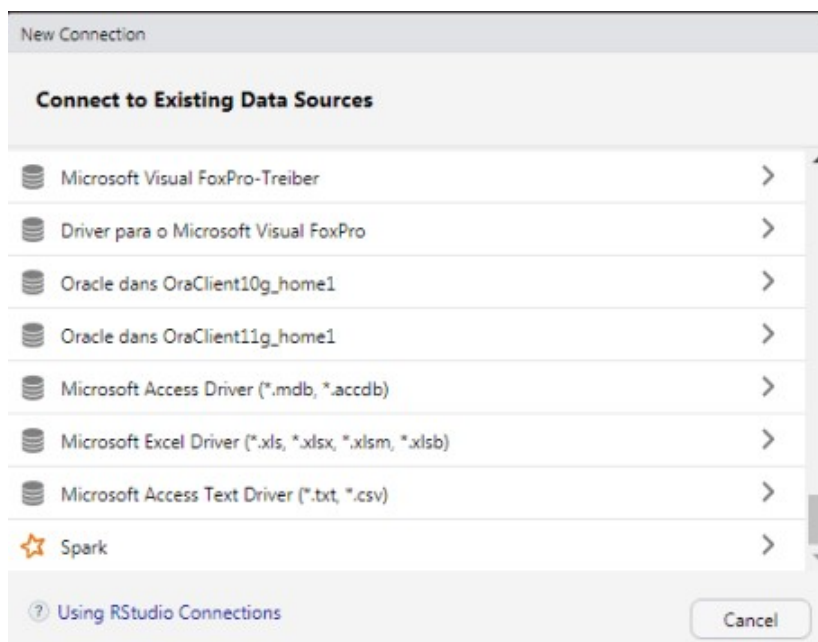- maybe try to clean install Oracle client

Often ROracle is used and it works well, sometimes it doesn't (some oci.dll not found[3], etc.). But it doesn't work with dplyr/dbplyr at the moment.

After several years with ROracle, I'm happy to have both possibilities for query writing and collecting (SQL, and now dplyr)
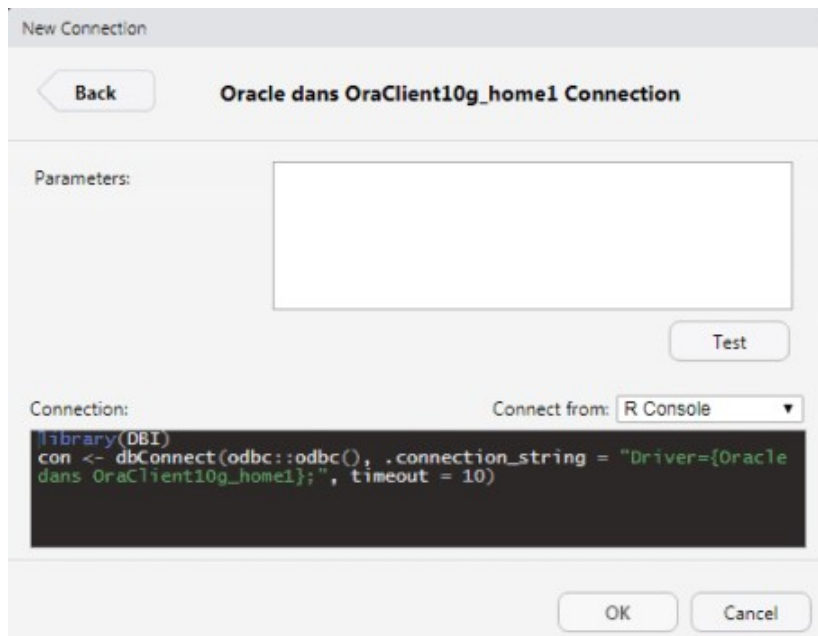
Here we are:

RStudio connection Pane

From connection Pane we take Oracle odbc driver name, we have two here for two Oracle client versions:

And then:

We now have a big component of the connection string.

32b or 64b

If your Oracle client is 32bit, you have to switch to R 32bits, otherwhise it doesn't work (at least for me).

Connection string

Then stackoverflow history helped me[4] to structure the entire string:

```
library(odbc)
library(dplyr)
library(dbplyr)
library(lubridate)

my_oracle <- dbConnect(odbc::odbc(),
                       .connection_string = "Driver={Oracle dans
OraClient10g_home1};DBQ=host:port/db_name;UID=woo;PWD=hoo",
                       timeout = 10)
```

You will find all these informations in tnsnames.ora. Port is probably 1521.

Some dplyr/dbplyr statements

Simple one

```
dplyr::tbl(my_oracle, dbplyr::in_schema('SCHEMA_ONE', 'TB_ONE'))

<SQL>
SELECT *
FROM SCHEMA_ONE.TB_ONE
```

dplyr and dblink

If you have another oracle database with dblinks it may also works like this:

```
dplyr::tbl(my_oracle, dbplyr::in_schema('SCHEMA_B',
'TC_TWO@MYDBTWOLINK'))

<SQL>
SELECT *
```

```
FROM SCHEMA_B.TC_TWO@MYDBTWOLINK
```

List dblinks

```
DBI::dbGetQuery(my_oracle, "SELECT * FROM ALL_DB_LINKS")

<SQL>
SELECT *
FROM ALL_DB_LINKS
```

Catalog of all columns[5]

```
<SQL>
SELECT *
FROM ALL_TAB_COLUMNS
```

Decomposing the connection string

In order to ask for password, we split the connection parts:

```
library(odbc)
library(dplyr)
library(dbplyr)
library(lubridate)

my_oracle <- dbConnect(odbc::odbc(),
                       Driver = "Oracle dans OraClient10g_home1",
                       DBQ = "host:port/db_name",
                       SVC = "DB_SCHEMA", # schema when connection
opens
                       UID = "woo",
                       PWD = "hoo")
```

And then:

```
library(odbc)
library(dplyr)
library(dbplyr)
library(lubridate)

my_oracle <- dbConnect(odbc::odbc(),
                       Driver = "Oracle dans OraClient10g_home1",
                       DBQ = "host:port/db_name",
                       SVC = "DB_SCHEMA",
                       UID = rstudioapi::askForPassword('woo
(username)'),
                       PWD = rstudioapi::askForPassword('Open, Sesame
(password)'))
```

---

1. RStudio documentation for Oracle connections: https://db.rstudio.com/databases/oracle/ ↩

2. see here for a readme in a repo on github: https://github.com/GuillaumePressiat/oracle_odbc_connection_template_for_R ↩

3. see here for ROracle difficulties: https://technology.amis.nl/2017/08/23/r-and-the-oracle-database-using-dplyr-dbplyr-with-roracle-on-windows-10/ ↩

4. how to make a connection string for oracle that includes hostname, instance name, user id, password using system.data.oracleclient? stackoverflow ↩

5. for Oracle catalogs, see here: https://docs.oracle.com/pls/db92/db92.catalog_views?remark=homepage