restore-adventureworks-postgres-on-docker.R

jds

2019-06-27

This is a utility job to create the Docker container with the adventureworks database that's used in the book. This job depends on the output from `load-adventureworks-to-postgres-on-docker.R`

```
library(tidyverse)
```

```
## -- Attaching packages ---------------------------------------------------- tidyverse 1.2.1 --
```

```
## v ggplot2 3.2.0          v purrr   0.3.2
## v tibble  2.1.3          v dplyr   0.8.1
## v tidyr   0.8.3.9000     v stringr 1.4.0
## v readr   1.3.1          v forcats 0.4.0
```

```
## -- Conflicts ------------------------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(DBI)
library(RPostgres)
library(glue)
```

```
##
## Attaching package: 'glue'
```

```
## The following object is masked from 'package:dplyr':
##
##     collapse
```

```
library(here)
```

```
## here() starts at /Users/jds/Documents/Library/R/r-system/sql-pet
```

```
require(knitr)
```

```
## Loading required package: knitr
```

```
library(dbplyr)
```

```
##
## Attaching package: 'dbplyr'
```

```
## The following objects are masked from 'package:dplyr':
##
##     ident, sql
```

```
library(sqlpetr)
library(inspectdf)

wd <- here()

# Verify Docker is up and running and list all containers:
sp_check_that_docker_is_up()
```

```
## [1] "Docker is up, running these containers:"
```

```
## [2] "CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS
## [3] "264103bb5444        postgres:10         \"docker-entrypoint.s...\"   2 days ago          Up 44 h
```

```
# sp_show_all_docker_containers()

# Remove previous container if it exists.

# sp_docker_remove_container("adventureworks")
sp_docker_remove_container("adv2")
```

```
## [1] 0
```

```
# create new adventureworks container
docker_cmd <- glue(
  "run ", # Run is the Docker command.  Everything that follows are `run` parameters.
  "--detach ", # (or `-d`) tells Docker to disconnect from the terminal / program issuing the command
  " --name adv2 ", # tells Docker to give the container a name: `sql-pet`
  "--publish 5432:5432 ", # tells Docker to expose the PostgreSQL port 5432 to the local network with 5
  "--mount ", # tells Docker to mount a volume -- mapping Docker's internal file structure to the host
  'type=bind,source="', wd, '",target=/petdir', # not really used, but could be later
  " postgres:10 " # tells Docker the image that is to be run (after downloading if necessary)
)
cat("docker ", docker_cmd)
```

```
## docker  run --detach  --name adv2 --publish 5432:5432 --mount type=bind,source="/Users/jds/Documents/
```

```
system2("docker", docker_cmd, stdout = TRUE, stderr = TRUE)
```

```
## [1] "102d9e481affadcfd4ccf3c96e3cf274ed6d69c5cbe30558bab29b45f8a4bc51"
```

```
Sys.sleep(2)

# create the adventureworks database in the Docker container
system2("docker", "exec -i adv2 psql -U postgres -c 'CREATE DATABASE adventureworks;' ")

# restore the adventureworks tar file.  This might come from github rather than locally
system2("docker", glue(
  "exec -i adv2 pg_restore -U postgres ",
  " -d adventureworks /petdir/book-src/adventureworks.sql "
))

# Wait for Docker to finish its business
Sys.sleep(4)

# sp_docker_start("adv2")

con <- sp_get_postgres_connection(
  host = "localhost",
  port = 5432,
  user = "postgres",
  password = "postgres",
  dbname = "adventureworks",
  seconds_to_test = 10
)

adventureworks_schemas <-  tbl(con, in_schema("information_schema", "schemata")) %>%
  select(catalog_name, schema_name, schema_owner) %>%
```

```
  collect()
```

```
## Warning: `overscope_eval_next()` is deprecated as of rlang 0.2.0.
## Please use `eval_tidy()` with a data mask instead.
## This warning is displayed once per session.
```

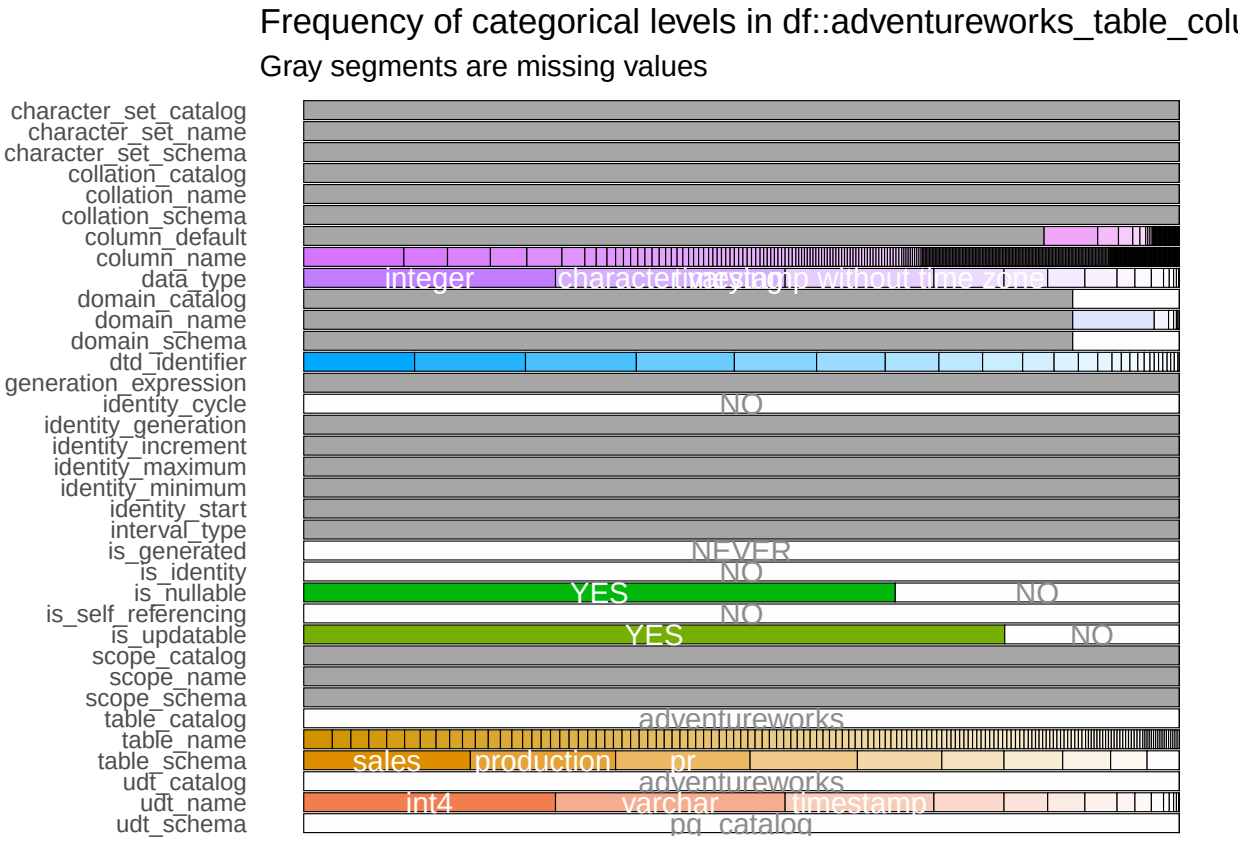
```
## Warning: `overscope_clean()` is deprecated as of rlang 0.2.0.
## This warning is displayed once per session.
```

```
adventureworks_schemas
```

```
## # A tibble: 16 x 3
##    catalog_name   schema_name        schema_owner
##    <chr>          <chr>              <chr>
##  1 adventureworks pg_toast           postgres
##  2 adventureworks pg_temp_1          postgres
##  3 adventureworks pg_toast_temp_1    postgres
##  4 adventureworks pg_catalog         postgres
##  5 adventureworks public             postgres
##  6 adventureworks information_schema postgres
##  7 adventureworks hr                 postgres
##  8 adventureworks humanresources     postgres
##  9 adventureworks pe                 postgres
## 10 adventureworks person             postgres
## 11 adventureworks pr                 postgres
## 12 adventureworks production         postgres
## 13 adventureworks pu                 postgres
## 14 adventureworks purchasing         postgres
## 15 adventureworks sa                 postgres
## 16 adventureworks sales              postgres
```

```
adventureworks_table_columns <- tbl(con, in_schema("information_schema", "columns")) %>%
  collect() %>%
  filter(!str_detect(table_schema,"pg_|information_schema" ))

cat_info <- inspect_cat(adventureworks_table_columns)
cat_info %>% show_plot()
```




```
# select the most useful columns
adventureworks_table_columns <- adventureworks_table_columns %>%
  select(table_schema, table_name, column_name, dtd_identifier, data_type, column_default)

adventureworks_table_columns %>%
  filter(table_name == "stateprovince") %>%
  select(column_name, dtd_identifier, data_type, column_default)
```

```
## # A tibble: 8 x 4
##   column_name     dtd_identifier data_type      column_default
##   <chr>           <chr>          <chr>          <chr>
## 1 stateprovincec~ 2              character      <NA>
## 2 countryregionc~ 3              character var~ <NA>
## 3 name            5              character var~ <NA>
## 4 territoryid     6              integer        <NA>
## 5 isonlystatepro~ 4              boolean        true
## 6 rowguid         7              uuid           uuid_generate_v1()
## 7 modifieddate    8              timestamp wit~ now()
## 8 stateprovinceid 1              integer        nextval('person.stateprovi~
```

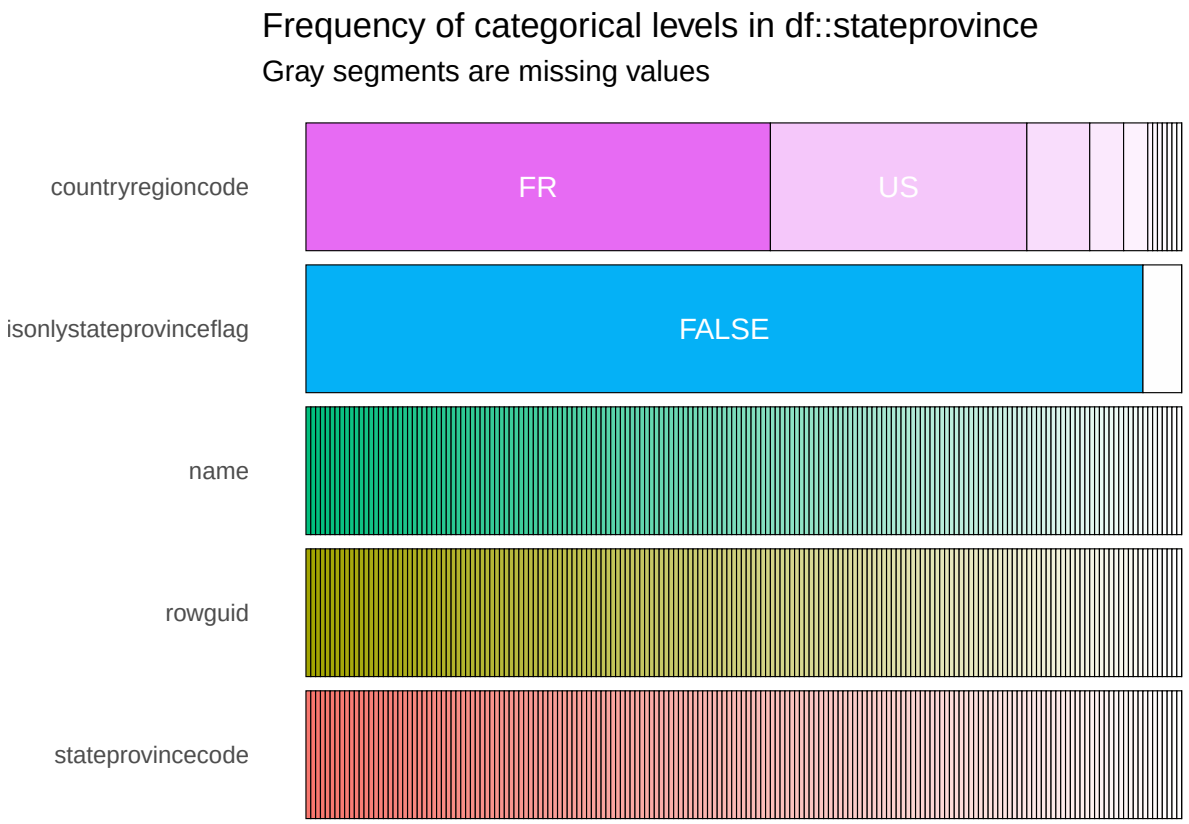
```
stateprovince <- tbl(con, in_schema("person", "stateprovince")) %>%
  collect()

stateprovince
```

```
## # A tibble: 181 x 8
##    stateprovinceid stateprovinceco~ countryregionco~ isonlystateprov~ name
##              <int> <chr>            <chr>            <lgl>            <chr>
```

```
##  1              1 "AB "           CA              FALSE           Albe~
##  2              2 "AK "           US              FALSE           Alas~
##  3              3 "AL "           US              FALSE           Alab~
##  4              4 "AR "           US              FALSE           Arka~
##  5              5 "AS "           AS              TRUE            Amer~
##  6              6 "AZ "           US              FALSE           Ariz~
##  7              7 "BC "           CA              FALSE           Brit~
##  8              8 "BY "           DE              FALSE           Baye~
##  9              9 "CA "           US              FALSE           Cali~
## 10             10 "CO "           US              FALSE           Colo~
## # ... with 171 more rows, and 3 more variables: territoryid <int>,
## #   rowguid <chr>, modifieddate <dttm>
```

```
inspect_cat(stateprovince) %>% show_plot
```




```
# list all the tables in the database (except information_schema tables)
adventureworks_tables <- adventureworks_table_columns %>%
  count(table_name) %>% rename(n_columns = n)

# deal with multiple schemas:

dbListTables(con)
```

```
## character(0)
```

```
dbExecute(con, "set search_path to production, person, public;") # watch for duplicates!
```

```
## [1] 0
```

```
dbListTables(con)
```

```
##  [1] "countryregion"
##  [2] "businessentityaddress"
##  [3] "phonenumbertype"
##  [4] "person"
##  [5] "address"
##  [6] "personphone"
##  [7] "emailaddress"
##  [8] "password"
##  [9] "contacttype"
## [10] "vadditionalcontactinfo"
## [11] "location"
## [12] "culture"
## [13] "document"
## [14] "illustration"
## [15] "productdescription"
## [16] "productsubcategory"
## [17] "productphoto"
## [18] "productproductphoto"
## [19] "transactionhistory"
## [20] "productmodel"
## [21] "scrapreason"
## [22] "transactionhistoryarchive"
## [23] "productinventory"
## [24] "productlistpricehistory"
## [25] "productreview"
## [26] "vproductmodelcatalogdescription"
## [27] "vproductmodelinstructions"
## [28] "unitmeasure"
## [29] "workorder"
## [30] "productdocument"
## [31] "addresstype"
## [32] "productcategory"
## [33] "billofmaterials"
## [34] "businessentitycontact"
## [35] "workorderrouting"
## [36] "productmodelillustration"
## [37] "stateprovince"
## [38] "productmodelproductdescriptionculture"
## [39] "businessentity"
## [40] "productcosthistory"
## [41] "product"
```

```
dbListFields(con, "person")
```

```
##  [1] "businessentityid"      "persontype"
##  [3] "namestyle"             "title"
##  [5] "firstname"             "middlename"
##  [7] "lastname"              "suffix"
##  [9] "emailpromotion"        "additionalcontactinfo"
## [11] "demographics"          "rowguid"
## [13] "modifieddate"
```

```
tbl(con, "person") %>% collect(n = 10)
```

```
## # A tibble: 10 x 13
##    businessentityid persontype namestyle title firstname middlename
##               <int> <chr>      <lgl>     <chr> <chr>     <chr>
##  1                1 EM         FALSE     <NA>  Ken       J
##  2                2 EM         FALSE     <NA>  Terri     Lee
##  3                3 EM         FALSE     <NA>  Roberto   <NA>
##  4                4 EM         FALSE     <NA>  Rob       <NA>
##  5                5 EM         FALSE     Ms.   Gail      A
##  6                6 EM         FALSE     Mr.   Jossef    H
##  7                7 EM         FALSE     <NA>  Dylan     A
##  8                8 EM         FALSE     <NA>  Diane     L
##  9                9 EM         FALSE     <NA>  Gigi      N
## 10               10 EM         FALSE     <NA>  Michael   <NA>
## # ... with 7 more variables: lastname <chr>, suffix <chr>,
## #   emailpromotion <int>, additionalcontactinfo <chr>, demographics <chr>,
## #   rowguid <chr>, modifieddate <dttm>
```

```
dbExecute(con, "set search_path to production, person, public;") # watch for duplicates!
```

```
## [1] 0
```

```
# verify that table names are unique across all schemas
adventureworks_tables %>%
  count(table_name) %>%
  filter(n > 1)
```

```
## # A tibble: 0 x 2
## # ... with 2 variables: table_name <chr>, n <int>
```

```
index_list <- tbl(con,  "pg_indexes") %>%
  select(schemaname, tablename, indexname, indexdef) %>%
  arrange(tablename) %>%
  collect() %>%
  filter(!str_starts(tablename, "pg_"))
index_list
```

```
## # A tibble: 71 x 4
##    schemaname tablename     indexname               indexdef
##    <chr>      <chr>         <chr>                   <chr>
##  1 person     address       PK_Address_AddressID    "CREATE UNIQUE INDEX \"~
##  2 person     addresstype   PK_AddressType_Addres~ "CREATE UNIQUE INDEX \"~
##  3 production billofmateri~ PK_BillOfMaterials_Bi~ "CREATE UNIQUE INDEX \"~
##  4 person     businessenti~ PK_BusinessEntity_Bus~ "CREATE UNIQUE INDEX \"~
##  5 person     businessenti~ PK_BusinessEntityAddr~ "CREATE UNIQUE INDEX \"~
##  6 person     businessenti~ PK_BusinessEntityCont~ "CREATE UNIQUE INDEX \"~
##  7 person     contacttype   PK_ContactType_Contac~ "CREATE UNIQUE INDEX \"~
##  8 person     countryregion PK_CountryRegion_Coun~ "CREATE UNIQUE INDEX \"~
##  9 sales      countryregio~ PK_CountryRegionCurre~ "CREATE UNIQUE INDEX \"~
## 10 sales      creditcard    PK_CreditCard_CreditC~ "CREATE UNIQUE INDEX \"~
## # ... with 61 more rows
```

```
index_list %>% count(schemaname)
```

```
## # A tibble: 5 x 2
##   schemaname         n
```

```
##   <chr>          <int>
## 1 humanresources     6
## 2 person            14
## 3 production        27
## 4 purchasing         5
## 5 sales             19
```

```
tbl(con, "pg_tables") %>% count(schemaname)
```

```
## # Source:   lazy query [?? x 2]
## # Database: postgres [postgres@localhost:5432/adventureworks]
##   schemaname         n
##   <chr>              <int64>
## 1 purchasing          5
## 2 pg_catalog         62
## 3 production         25
## 4 person             13
## 5 humanresources      6
## 6 information_schema  7
## 7 sales              19
```

```
employee <- tbl(con, dbplyr::in_schema("humanresources", "employee"))

department_history  <- tbl(con, dbplyr::in_schema("humanresources", "employeedepartmenthistory"))

employee %>%
  left_join(department_history, by = c("businessentityid") ) %>%
  count(businessentityid, sort = TRUE)
```

```
## Warning: `chr_along()` is deprecated as of rlang 0.2.0.
## This warning is displayed once per session.
```

```
## # Source:     lazy query [?? x 2]
## # Database:   postgres [postgres@localhost:5432/adventureworks]
## # Ordered by: desc(n)
##    businessentityid n
##               <int> <int64>
##  1              250 3
##  2                4 2
##  3               16 2
##  4              234 2
##  5              224 2
##  6               70 1
##  7              272 1
##  8               87 1
##  9              169 1
## 10              176 1
## # ... with more rows
```

```
DBI::dbDisconnect(con)

# if R has not finished disconnecting, Docker returns a "137" when the
#   container is stopped.
Sys.sleep(4)

system2("docker", "stop adv2", stdout = TRUE, stderr = TRUE)
```

```
## [1] "adv2"
```