Depuración de bases de datos

Matías Deneken

Documentación necesaria para realizar tarea 2

Introducción

El presente documento buscará la creación de bases de datos para SQL. Aquí utilizaremos la fuente de datos propuesta en CANVAS. Sin embargo, se requiere hacer ajustes sustantivos al excel, pues se necesita que las claves foráneas no se repitan, filtro de la bbdd, entre otros.

Depuración de base datos

```
datos <- read_excel("bbdd/fuente_datos.xlsx")
```

Renombraremos el nombre de las variables de interés; además de que creamos un filtro de la base de datos para quedarnos solo con algunas.

```
datos2 <- datos %>% rename( nombre_tipo_inst_1 = `TIPO INST 1`,
                  nombre_tipo_inst_2 = `TIPO INST 2`,
  nombre_tipo_inst_3 = `TIPO INST 3`,
  acreditacion = `ACREDITACIÓN CARRERA O PROGRAMA`,
  region = `REGION SEDE`,
  provincia = PROVINCIA,
  comuna = `COMUNA SEDE`,
  area_conocimiento = `AREA CONOCIMIENTO`,
  oecd_area = `OECD AREA`,
  oecd_subarea = `OECD SUBAREA`,
  area_carrera = `AREA CARRERA GENERICA`,
  cod_tipo_ies = `TIPO IES`,
  cod_ies = `CODIGO IES`,
  nombre_ies = `NOMBRE IES`,
  cod_sede = `CODIGO SEDE`,
  nombre_sede = `NOMBRE SEDE`,
  cod_carrera = `CODIGO CARRERA`,
  nombre_carrera = `NOMBRE CARRERA`,
  pond_nem = `PONDERACION NOTAS`,
  pond_ranking = `PONDERACION RANKING NOTAS`,
  pond_lengua = `PONDERACION LENGUAJE`,
  pond_mat = `PONDERACION MATEMATICAS`,
  pond_hist = `PONDERACION HISTORIA`,
  pond_ciencia = `PONDERACION CIENCIAS`,
  arancel = `ARANCEL ANUAL`,
  cuota_basica = `MATRICULA ANUAL`,
  costo_titulacio = `COSTO TITULACION`,
  anio = `AÑO`,
  matriculados = `TOTAL MATRICULADOS`,
```

```
  matricula_mujer = `MATRICULADOS MUJERES POR CARRERA`,
  matricula_hombre = `MATRICULADOS HOMBRES POR CARRERA`) %>%
  select(
    nombre_tipo_inst_1, cod_tipo_ies, nombre_tipo_inst_2, nombre_tipo_inst_3,
    region, provincia, comuna, area_conocimiento,
    oecd_area, oecd_subarea, area_carrera, cod_ies,
    nombre_ies, cod_sede, nombre_sede, cod_carrera, nombre_carrera,
    pond_nem, pond_ranking, pond_lengua, pond_mat, pond_hist,
    pond_ciencia, arancel, cuota_basica, costo_titulacio, anio,
    matriculados, matricula_mujer, matricula_hombre
  ) %>%   filter(nombre_ies %in% c("UNIVERSIDAD TECNICA FEDERICO SANTA MARIA",
                           "UNIVERSIDAD DIEGO PORTALES",
                           "UNIVERSIDAD DE SANTIAGO DE CHILE",
                           "UNIVERSIDAD DE CONCEPCION",
                           "UNIVERSIDAD DE CHILE",
                           "PONTIFICIA UNIVERSIDAD CATOLICA DE CHILE")) %>%
  mutate(prestigio = case_when(
    nombre_ies %in% c("UNIVERSIDAD DE CONCEPCION", "PONTIFICIA UNIVERSIDAD CATOLICA DE CHILE") ~ 7,
    TRUE ~ 6
  ))
```

Se realiza el procesamiento para las tablas.

```
# Arancel.

arancel <- datos2 %>%
  select(arancel, cuota_basica, cod_carrera) %>%
  distinct() %>%
  mutate(
    cod_arancel = row_number(), # Crear un código único secuencial para arancel
    cod_cuota = dense_rank(cuota_basica) # Crear un código único secuencial para cuota_basica
  )

# Area

area <- datos2  %>% select(oecd_area, cod_carrera) %>%
  distinct()

area <- area %>% mutate(
    cod_area = group_indices(., oecd_area))

# Año

anio <- datos2 %>% select(anio, cod_carrera)

#prestigio

prestigio <- datos2 %>% select(cod_ies, prestigio) %>%
  distinct()

# Nombres IES

nombres_ies_cod <-datos2 %>%  select(cod_ies, nombre_ies, cod_tipo_ies) %>%
  distinct()
```

```r
nombres_ies_cod$cod_tipo_ies <- as.factor(nombres_ies_cod$cod_tipo_ies)
nombres_ies_cod$cod_tipo_ies <- as.numeric(nombres_ies_cod$cod_tipo_ies)

# Carreras

nombres_carrera_cod <-datos2 %>%  select(cod_carrera, nombre_carrera, cod_sede,oecd_area) %>%
  distinct()

nombres_carrera_cod <- nombres_carrera_cod %>% mutate(cod_area = group_indices(., oecd_area))

# Direccion

direccion <- datos2 %>% select(cod_sede,comuna, provincia, region)  %>%  distinct()

# Sede

sede <-datos2 %>% select(nombre_sede, cod_ies) %>% distinct() %>%
  mutate(cod_sede = row_number())

# Sub_area

sub_area <- datos2 %>% select(oecd_subarea) %>%
   #mutate(cod_area = group_indices(., oecd_area)) %>%
  mutate(cod_sub_area = group_indices(., oecd_subarea)) %>%
  distinct()

# Tipo institucionn

inst_1 <- datos2 %>% select(nombre_tipo_inst_1) %>%
  mutate(cod_tipo_inst_1 = group_indices(., nombre_tipo_inst_1)) %>%
  distinct()

inst_2 <- datos2 %>% select(nombre_tipo_inst_2) %>%
  mutate(cod_tipo_inst_2 = group_indices(., nombre_tipo_inst_2)) %>%
  distinct()

inst_3 <- datos2 %>% select(nombre_tipo_inst_3) %>%
  mutate(cod_tipo_inst_3 = group_indices(., nombre_tipo_inst_3)) %>%
  distinct()
```

Además crearemos postulantes y matriculados aleatorizados para poder subirlos a SQL.

```r
# Cargar la librería necesaria
library(dplyr)

# Función para generar RUT aleatorio
generate_rut <- function() {
  paste0(sample(10000000:25000000, 1), "-", sample(0:9, 1))
}

# Función para generar puntajes aleatorios para matriculados
generate_score_matriculados <- function() {
  sample(650:850, 1)
}
```

```r
# Función para generar puntajes aleatorios para postulantes
generate_score_postulantes <- function() {
  sample(450:650, 1)
}

# Función para generar nombres aleatorios
generate_name <- function() {
  first_names <- c("Victor", "Mariana", "Carlos", "Ana", "Luis", "Francisca", "Diego", "Camila", "Javie
  last_names <- c("Osorio", "Larrahona", "Perez", "Diaz", "Gonzalez", "Martinez", "Rodriguez", "Hernand
  paste(sample(first_names, 1), sample(last_names, 1))
}

# Función para generar género aleatorio
generate_gender <- function() {
  sample(c("Femenino", "Masculino"), 1)
}

# Listas para región, provincia y comuna
regions <- c("Metropolitana", "Biobío", "Los Lagos", "Coquimbo", "Aconcagua", "Santiago")
provinces <- c("Santiago", "Valparaíso", "Concepción", "Osorno", "Porvenir", "Coquimbo")
communes <- c("Santiago Centro", "Valparaíso", "Concepción", "Osorno", "Lago Verde", "Porvenir")

# Función para generar direcciones aleatorias
generate_address <- function() {
  list(
    region = sample(regions, 1),
    province = sample(provinces, 1),
    commune = sample(communes, 1),
    street = paste("Calle", sample(1:150, 1)),
    number = sample(1:999, 1)
  )
}

# Crear el DataFrame para matriculados con 250 registros
set.seed(123) # Para reproducibilidad
matriculados_data <- data.frame(
  rut = replicate(250, generate_rut()),
  nombre_matriculado = replicate(250, generate_name()),
  puntaja_Notas = replicate(250, generate_score_matriculados()),
  puntaja_Ranking_Notas = replicate(250, generate_score_matriculados()),
  puntaja_Lenguaje = replicate(250, generate_score_matriculados()),
  puntaja_Matemática = replicate(250, generate_score_matriculados()),
  puntaja_Historia = replicate(250, generate_score_matriculados()),
  puntaja_Ciencias = replicate(250, generate_score_matriculados()),
  puntaja_Otro = replicate(250, generate_score_matriculados()),
  Género = replicate(250, generate_gender()),
  Dirección_Región = character(250),
  Dirección_Provincia = character(250),
  Dirección_Comuna = character(250),
  Dirección_Calle = character(250),
  Dirección_Número = integer(250),
  stringsAsFactors = FALSE
)
```

```r
# Llenar direcciones en el DataFrame de matriculados
for (i in 1:nrow(matriculados_data)) {
  address <- generate_address()
  matriculados_data$Dirección_Región[i] <- address$region
  matriculados_data$Dirección_Provincia[i] <- address$province
  matriculados_data$Dirección_Comuna[i] <- address$commune
  matriculados_data$Dirección_Calle[i] <- address$street
  matriculados_data$Dirección_Número[i] <- address$number
}

# Crear el DataFrame para postulantes con 250 registros
postulantes_data <- data.frame(
  Rut = replicate(250, generate_rut()),
  Nombre_matriculado = replicate(250, generate_name()),
  puntaja_Notas = replicate(250, generate_score_postulantes()),
  puntaja_Ranking_Notas = replicate(250, generate_score_postulantes()),
  puntaja_Lenguaje = replicate(250, generate_score_postulantes()),
  puntaja_Matemática = replicate(250, generate_score_postulantes()),
  puntaja_Historia = replicate(250, generate_score_postulantes()),
  puntaja_Ciencias = replicate(250, generate_score_postulantes()),
  puntaja_Otro = replicate(250, generate_score_postulantes()),
  Género = replicate(250, generate_gender()),
  Dirección_Región = character(250),
  Dirección_Provincia = character(250),
  Dirección_Comuna = character(250),
  Dirección_Calle = character(250),
  Dirección_Número = integer(250),
  stringsAsFactors = FALSE
)

# Llenar direcciones en el DataFrame de postulantes
for (i in 1:nrow(postulantes_data)) {
  address <- generate_address()
  postulantes_data$Dirección_Región[i] <- address$region
  postulantes_data$Dirección_Provincia[i] <- address$province
  postulantes_data$Dirección_Comuna[i] <- address$commune
  postulantes_data$Dirección_Calle[i] <- address$street
  postulantes_data$Dirección_Número[i] <- address$number
}

colnames(matriculados_data) <- tolower(colnames(matriculados_data))
colnames(postulantes_data) <- tolower(colnames(postulantes_data))

postulantes_data  <-  postulantes_data %>% rename(nombre_postulante = nombre_matriculado)
```

Exportación de tablas

Guardamos las tablas en Excel para que pueden ser enviadas al SQL

```r
write_csv(datos2, "bbdd/datos2.csv")
write_csv(arancel, "bbdd/arancel.csv")
write_csv(area, "bbdd/area.csv")
write_csv(direccion, "bbdd/direccion.csv")
```

```r
write_csv(inst_1, "bbdd/inst_1.csv")
write_csv(inst_2, "bbdd/inst_2.csv")
write_csv(inst_3, "bbdd/inst_3.csv")
write_csv(nombres_carrera_cod, "bbdd/nombres_carrera_cod.csv")
write_csv(nombres_ies_cod, "bbdd/nombres_ies_cod.csv")
#write_csv(postulante, "bbdd/postulante.csv")
write_csv(sede, "bbdd/sede.csv")
write_csv(sub_area, "bbdd/sub_area.csv")
write_csv(prestigio, "bbdd/acreditacion.csv")
# Guardar los DataFrames en archivos CSV
write.csv(matriculados_data, "bbdd/matriculados_data.csv", row.names = FALSE)
write.csv(postulantes_data, "bbdd/postulantes_data.csv", row.names = FALSE)
```

Las bases de datos tienen valores únicos en sus atributos dependiendo si es KP. Un ejemplo:

```r
head(nombres_ies_cod)
```

```
## # A tibble: 6 x 3
##   cod_ies nombre_ies                                cod_tipo_ies
##     <dbl> <chr>                                            <dbl>
## 1       3 UNIVERSIDAD DIEGO PORTALES                           3
## 2      70 UNIVERSIDAD DE CHILE                                 1
## 3      71 UNIVERSIDAD DE SANTIAGO DE CHILE                     1
## 4      86 PONTIFICIA UNIVERSIDAD CATOLICA DE CHILE             2
## 5      87 UNIVERSIDAD DE CONCEPCION                            2
## 6      88 UNIVERSIDAD TECNICA FEDERICO SANTA MARIA             2
```

Otro ejemplo:

```r
head(sub_area)
```

```
## # A tibble: 6 x 2
##   oecd_subarea                          cod_sub_area
##   <chr>                                        <int>
## 1 Industria y Producción                          12
## 2 Formación de Personal Docente                   10
## 3 Informática                                     13
## 4 Ciencias Sociales y del Comportamiento           5
## 5 Artes                                            3
## 6 Periodismo e Información                        17
```

Todas las bases de datos podrán verse aquí.