Prácticas de Bases de Datos Relacionales con R

Máster en Data Science y Big Data con R

Gisela Ruiz Renzi

22 Octubre 2023

Contents


```
#install.packages("RMySQL")
library(RMySQL)
```

```
## Loading required package: DBI
```

```
library(DBI)
```

5. Tarea a realizar en la práctica.

A continuación se pide que realicéis las siguientes tareas.

5.1. Tarea 1: Crear conexión a vuestra base de datos personal

Mira el fichero con la información de los usuarios e identifica tu nombre de usuario (será el nombre del correo con el que accedes al Campus Virtual quitando "@alu.uclm.es") y el nombre de la base de datos que te ha sido asignada. Se trata de que crees la conexión a MySQL usando el driver **RMySQL**. Guarda el nombre de esta variable porque tendrás que seguir usándola en el resto de los ejercicios.

Pistas:

- Pista 1. La IP del Host donde tenemos el servidor es 172.20.48.118,
- Pista 2. Recuerda que al host sólo se puede acceder a través de la VPN de la UCLM.

- Pista 3. Las bases de datos de los alumnos tienen la forma BD_1, BD_10 y cada uno tenéis la vuestra propia.

```
print ("Escribe aquí todas las instrucciones R para conectarte a tu base de datos usando R con el usuari
```

```
## [1] "Escribe aquí todas las instrucciones R para conectarte a tu base de datos usando R con el usuari
```

```
usuario   =   'GiselaJudit.Ruiz'
passwd    =   'MdsR.2023#'
nombrebd  =   'BD_15'
servidor  =   '172.20.48.70'
puerto    =   3306
mibbdd_15 = dbConnect(MySQL(), user=usuario, password=passwd, dbname=nombrebd,
                  host=servidor, port= puerto)
summary (mibbdd_15)
```

```
## <MySQLConnection:0,0>
##   User:   GiselaJudit.Ruiz
##   Host:   172.20.48.70
##   Dbname: BD_15
##   Connection type: 172.20.48.70 via TCP/IP
##
## Results:
```

```
dbGetInfo(mibbdd_15)
```

```
## $host
## [1] "172.20.48.70"
##
## $user
## [1] "GiselaJudit.Ruiz"
##
## $dbname
## [1] "BD_15"
##
## $conType
## [1] "172.20.48.70 via TCP/IP"
##
## $serverVersion
## [1] "8.0.34-0ubuntu0.20.04.1"
##
## $protocolVersion
## [1] 10
##
## $threadId
## [1] 1191
##
## $rsId
## list()
```

```
#dbDisconnect(mibbdd_15)
```

5.2. Tarea 2: Crear tablas con datos

Esta tarea tiene como objetivo crear una base de datos de películas que han protagonizado determinados actores.

Inicialmente te sugiero almacenar los siguientes hechos:

- Kevin Costner protagonizó en 1991 la película cuyo título en España fue "Robin Hood: el Príncipe de los Ladrones" y en versión original "Robin Hood: Prince of Thieves" siendo la productora "Warner Bros"
- Russell Crowe protagonizó en 2010 la película de la productora "Universal Pictures" cuyo título en España fue "Robin Hood" y en versión original fue "Robin Hood".
- Russell Crowe protagonizó en 2000 la película de la productora "Dreamworks Pictures" titulada "Gladiator"

Pistas:

- Pista 1. Piensa cuántas tablas necesitas crear (recuerda el ejemplo de los libros)
- Pista 2. Piensa en los atributos que debería tener cada una de las tablas. En la descripción de los hechos anteriores tienes enlaces a IMDB de los actores y de las películas donde puedes completar los atributos que podrían ser de tu interés. En la tarea 5 te propondré algunas consultas que son de mi interés sobre la base de datos de mi clientes. Analízalas para inferir qué atributos deberías añadir a las diferentes tablas. A modo de ejemplo, te animo a que crees un atributo nacionalidad en la tabla del actor o clasificación de la película y lo dejes en blanco para crear una actualización en la tarea 4.
- Pista 3. Crea un data frame por cada una de las tablas, y aunque no hemos visto cómo garantizar desde el punto de vista tecnológico la integridad referencial, asegúrate que no se viola respetando los valores que creas oportunos.
- Pista 4. Piensa qué instrucción de **RMySQL** debería utilizar para crear las tablas.
- Pista 5. Tu usuario tiene asignados permisos para la creación de las tablas. En caso de que te de algún error al crear la tabla por falta de permisos, contacta con el profesor.

```
print ("Escribe aquí todas las instrucciones R para crear la base de datos de películas y actores")
```

```
## [1] "Escribe aquí todas las instrucciones R para crear la base de datos de películas y actores"
```

```
#Tabla Actor

CodActor <- c("kcostner", "rcrowe")
Nombre <- c("Kevin", "Russell")
Apellido <- c("Costner", "Crowe")
Nacionalidad <- c("", "")

dfActor <- data.frame(CodActor, Nombre, Apellido, Nacionalidad)

dbWriteTable(mibbdd_15, "Actor", dfActor, overwrite = TRUE, row.names =FALSE )
```

```
## [1] TRUE
```

```
#Tabla Película

CodPelicula <- c("rhood","robinh","gladiador")
TituloEsp <- c("Robin Hood: el Príncipe de los Ladrones","Robin Hood", "Gladiator(El Gladiador)" )
TituloVO <- c("Robin Hood: Prince of Thieves", "Robin Hood", "Gladiator")
Publico <- c("", "", "")
Productora <- c("Warner Bros", "Universal Pictures", "Dreamworks Pictures")
Anio <- c(1991, 2010, 2000)
dfPelicula <- data.frame(CodPelicula, TituloEsp, TituloVO, Publico, Productora, Anio)
dbWriteTable(mibbdd_15, "Pelicula", dfPelicula, overwrite = TRUE, row.names =FALSE )
```

```
## [1] TRUE
```

```
#Tabla Protagoniza

CodActor <- c("kcostner", "rcrowe", "rcrowe")
CodPelicula <- c("rhood","robinh","gladiador")
Anio <- c(1991, 2010, 2000)
dfProtagoniza <- data.frame(CodActor, CodPelicula, Anio)
dbWriteTable(mibbdd_15, "Protagoniza", dfProtagoniza, overwrite = TRUE, row.names =FALSE )
```

```
## [1] TRUE
```

```
dbListTables(mibbdd_15)
```

```
## [1] "Actor"       "Pelicula"    "Protagoniza"
```

5.3. Tarea 3: Insertar nuevos registros en la base de datos.

En esta tarea lo que se persigue es que añadas al menos tres nuevos registros en la base de datos, usando cada vez un método diferente de los que hemos visto. Para ello te sugiero que añadas los siguientes hechos, aunque puedes elegir los que tú prefieras. * Los datos de la película “Matrix” protagonizada por Keanu Reaves * Los datos de la película “Pretty Woman” protagonizada por Julia Roberts * Los datos de la película “Logan” protagonizada por Hugh Jackman

Pistas:

- Pista 1. Recuerda que todo el trabajo en dos de los tres métodos se hace a nivel de data frame.
- Pista 2. Utiliza los enlaces a IMDB para completar la información a insertar.
- Pista 2. Comprueba que la inserción de datos ha tenido éxito.

```
print ("Escribe aquí el código correspondiente a las tres inserciones que se indican en en el enunciado
```

```
## [1] "Escribe aquí el código correspondiente a las tres inserciones que se indican en en el enunciado
```

```
#PELÍCULA MATRIX

#Tabla Actor
dfActor <- dbReadTable(mibbdd_15, "Actor")
dfNuevoRegistroActor <- as.list(dfActor)
```

```
dfNuevoRegistroActor$CodActor <- c("Kreeves")
dfNuevoRegistroActor$Nombre <- c("Keanu")
dfNuevoRegistroActor$Apellido <- c("Reeves")
dfNuevoRegistroActor$Nacionalidad <- c("canadiense")
dfActor <- rbind (dfActor, dfNuevoRegistroActor)
dbWriteTable(mibbdd_15, "Actor", dfActor, overwrite = TRUE, row.names =FALSE)
```

```
## [1] TRUE
```

```
#Tabla Película
dfPelicula <- dbReadTable(mibbdd_15, "Pelicula")
dfNuevoRegistroPelicula <- as.list(dfPelicula)
dfNuevoRegistroPelicula$CodPelicula <- c("matrix")
dfNuevoRegistroPelicula$TituloEsp <- c("Matrix")
dfNuevoRegistroPelicula$TituloVO <- c("Matrix")
dfNuevoRegistroPelicula$Publico <- c("+18")
dfNuevoRegistroPelicula$Productora <- c("Warner Bros")
dfNuevoRegistroPelicula$Anio <- c(1999)
dfPelicula <- rbind (dfPelicula, dfNuevoRegistroPelicula)
dbWriteTable(mibbdd_15, "Pelicula", dfPelicula, overwrite = TRUE, row.names =FALSE)
```

```
## [1] TRUE
```

```
#Tabla Protagoniza
dfProtagoniza <- dbReadTable(mibbdd_15, "Protagoniza")
dfNuevoRegistroProtagoniza <- as.list(dfProtagoniza)
dfNuevoRegistroProtagoniza$CodActor <- c("Kreeves")
dfNuevoRegistroProtagoniza$CodPelicula <- c("matrix")
dfNuevoRegistroProtagoniza$Anio <- c(1999)
dfProtagoniza <- rbind (dfProtagoniza, dfNuevoRegistroProtagoniza)
dbWriteTable(mibbdd_15, "Protagoniza", dfProtagoniza, overwrite = TRUE, row.names =FALSE)
```

```
## [1] TRUE
```

```
#PELÍCULA PRETTY WOMAN

#Tabla Actor
dfDatoActorNuevo <- as.list(dfActor)
dfDatoActorNuevo$CodActor <- "jroberts"
dfDatoActorNuevo$Nombre <- "Julia"
dfDatoActorNuevo$Apellido <- "Roberts"
dfDatoActorNuevo$Nacionalidad <- "estadounidense"
dfDatoActorNuevo <- data.frame (dfDatoActorNuevo)
dbWriteTable(mibbdd_15, "Actor", dfDatoActorNuevo, append = TRUE, row.names =FALSE)
```

```
## [1] TRUE
```

```
#Tabla Pelicula
dfDatoPeliculaNuevo <- as.list(dfPelicula)
dfDatoPeliculaNuevo$CodPelicula <- "prwoman"
```

```r
dfDatoPeliculaNuevo$TituloEsp <- "Mujer Bonita"
dfDatoPeliculaNuevo$TituloVO <- "Pretty Woman"
dfDatoPeliculaNuevo$Publico <- "+13"
dfDatoPeliculaNuevo$Productora <- "Touchstone Pictures"
dfDatoPeliculaNuevo$Anio <- 1990
dfDatoPeliculaNuevo <- data.frame (dfDatoPeliculaNuevo)
dbWriteTable(mibbdd_15, "Pelicula", dfDatoPeliculaNuevo, append = TRUE, row.names =FALSE)
```

```
## [1] TRUE
```

```r
#Tabla Protagoniza
dfDatoProtagonizaNuevo <- as.list(dfProtagoniza)
dfDatoProtagonizaNuevo$CodActor <- "jroberts"
dfDatoProtagonizaNuevo$CodPelicula <- "prwoman"
dfDatoProtagonizaNuevo$Anio <- 1990
dfDatoProtagonizaNuevo <- data.frame (dfDatoProtagonizaNuevo)
dbWriteTable(mibbdd_15, "Protagoniza", dfDatoProtagonizaNuevo, append = TRUE, row.names =FALSE)
```

```
## [1] TRUE
```

```r
#PELÍCULA LOGAN

#Tabla Actor
SentenciaSQL_InsercionActor ="insert into Actor value ('hjackman', 'Hugh', 'Jackman', 'australiana')"
dbSendQuery (mibbdd_15, SentenciaSQL_InsercionActor)
```

```
## <MySQLResult:-1864603136,0,40>
```

```r
#Tabla Pelicula
SentenciaSQL_InsercionPelicula ="insert into Pelicula value ('logan', 'Logan', 'Logan', '+16', '20th Ce
dbSendQuery (mibbdd_15, SentenciaSQL_InsercionPelicula)
```

```
## <MySQLResult:NA,0,41>
```

```r
#Tabla Protagoniza
SentenciaSQL_InsercionProtagoniza ="insert into Protagoniza value ('hjackman','logan', 2017)"
dbSendQuery (mibbdd_15, SentenciaSQL_InsercionProtagoniza)
```

```
## <MySQLResult:12,0,42>
```

5.4. Tarea 4: Actualizar los datos de la base de datos

Esta tarea tiene como objetivo que hagas actualizaciones a las tablas de la base de datos que estás creando. En la tarea 2 te animaba a que dejaras en blanco el campo "nacionalidad" de cada uno de los actores. Este puede ser un buen ejemplo de actualización. Otro ejemplo de actualización podría ser el cambio de clasificación de una película ("mayores de 18 años", "apta para todos los públicos", ...)

Pistas:

- Pista 1. Recuerda cómo recuperar el contenido de toda la tabla en un data frame

- Pista 2. Piensa en cómo volver a guardar todo el data frame en la tabla una vez hechos los cambios.
- Pista 3. Sigue los enlaces en IMDB para completar los datos de las películas
- Pista 4. Comprueba que la actualización de datos ha tenido éxito.

```
print ("Escribe aquí el código correspondiente a las tres actualizaciones que se indican en el enunciad
```

```
## [1] "Escribe aquí el código correspondiente a las tres actualizaciones que se indican en el enunciad
```

```
#"Nacionalidad Kevin Costner"

dfActor <- dbReadTable(mibbdd_15, "Actor")
dfActor$Nacionalidad[1] <- "estadounidense"
dbWriteTable(mibbdd_15, "Actor", dfActor, overwrite = TRUE, row.names =FALSE  )
```

```
## [1] TRUE
```

```
dfActorModificado <- dbReadTable(mibbdd_15, "Actor")
#dfActorModificado

#"Nacionalidad Russell Crowe"

dfActor <- dbReadTable(mibbdd_15, "Actor")
dfActor$Nacionalidad[2] <- "neozelandesa"
dbWriteTable(mibbdd_15, "Actor", dfActor, overwrite = TRUE, row.names =FALSE  )
```

```
## [1] TRUE
```

```
dfActorModificado2 <- dbReadTable(mibbdd_15, "Actor")
dfActorModificado2
```

```
##   CodActor  Nombre Apellido   Nacionalidad
## 1 kcostner   Kevin  Costner estadounidense
## 2   rcrowe Russell    Crowe   neozelandesa
## 3  Kreeves   Keanu   Reeves     canadiense
## 4 jroberts   Julia  Roberts estadounidense
## 5 hjackman    Hugh  Jackman    australiana
```

```
#"Actualización atributo Público"

SentenciaSQL_InsercionDato1 ="UPDATE Pelicula SET Publico = 'TP' WHERE CodPelicula ='rhood'"
dbSendQuery(mibbdd_15, SentenciaSQL_InsercionDato1)
```

```
## <MySQLResult:6029404,0,57>
```

```
SentenciaSQL_InsercionDato2 ="UPDATE Pelicula SET Publico ='+7' WHERE CodPelicula ='robinh'"
dbSendQuery(mibbdd_15, SentenciaSQL_InsercionDato2)
```

```
## <MySQLResult:1,0,58>
```

```
SentenciaSQL_InsercionDato3 ="UPDATE Pelicula SET Publico ='+12' WHERE CodPelicula ='gladiador'"
dbSendQuery(mibbdd_15, SentenciaSQL_InsercionDato3)
```

```
## <MySQLResult:0,0,59>
```

```
dfPeliculaActualizado <- dbReadTable(mibbdd_15, "Pelicula")
dfPeliculaActualizado
```

```
##   CodPelicula                                TituloEsp
## 1       rhood Robin Hood: el Príncipe de los Ladrones
## 2      robinh                                Robin Hood
## 3   gladiador                Gladiator(El Gladiador)
## 4      matrix                                   Matrix
## 5     prwoman                               Mujer Bonita
## 6       logan                                      Logan
##                          TituloVO Publico             Productora Anio
## 1 Robin Hood: Prince of Thieves      TP            Warner Bros 1991
## 2                     Robin Hood      +7     Universal Pictures 2010
## 3                      Gladiator     +12    Dreamworks Pictures 2000
## 4                         Matrix     +18            Warner Bros 1999
## 5                  Pretty Woman     +13    Touchstone Pictures 1990
## 6                         Logan     +16 20th Century Fox Films 2017
```

5.5. Tarea 5: Consultas a la base de datos.

En esta tarea el objetivo es realizar las tres siguientes consultas sencillas a la base de datos. Las consultas serán sencillas mientras todos los datos salgan de una única tabla. En caso contrario, tendremos consultas multitabla en las que tendrás que hacer uso de la integridad referencial (*a esto se le suele llamar "cruzar tablas"*). No son consultas complicadas, pero como no las hemos visto aquí no las vamos a pedir. No obstante, si te animas a hacerlas, bienvenidas serán. Te propongo las siguientes consultas:

- Lista todos los actores con nacionalidad estadounidense
- Lista todos las películas con una clasificación de "Apta para todos los públicos"
- Lista todas las películas realizadas entre 1990 y 2000.

Pistas:

- Pista 1. Recuerda la importancia del Glosario de Términos, del Catálogo de Datos, y del Diccionario de Datos para identificar los atributos que serán objeto de las consultas. Aunque no lo pido como una tarea específica, tienes herramientas para obtener la información que puedes llegar a necesitar.
- Pista 2. Recuerda que el tipo de consultas SQL que tienes que escribir son las de selección.
- Pista 3. Recuerda el tipo de función **RMySQL** que tienes que usar para poder guardar los resultados en data frames y procesarlos con R.
- Pista 4. Trata de usar los dos métodos que hemos visto para realizar la consulta.
- Pista 5. Comprueba los resultados mostrando los data frames que se han generado.

```
print ("Introduce aquí el código R correspondiente a la realización de las consultas pedidas")
```

```
## [1] "Introduce aquí el código R correspondiente a la realización de las consultas pedidas"
```

```r
#"Consulta 1"

SentenciaSQL_actorestadounidense = "Select Nombre, Apellido from Actor where Nacionalidad = 'estadounid

Consulta_actorestadounidense = dbGetQuery(mibbdd_15, SentenciaSQL_actorestadounidense)
summary(Consulta_actorestadounidense)
```

```
##     Nombre            Apellido
##  Length:2           Length:2
##  Class :character   Class :character
##  Mode  :character   Mode  :character
```

```r
Consulta_actorestadounidense
```

```
##   Nombre Apellido
## 1  Kevin  Costner
## 2  Julia  Roberts
```

```r
#"Consulta 2"

SentenciaSQL_ConsultaTodoPublico = "Select TituloEsp, TituloVO from Pelicula where Publico = 'TP'"

Consulta_TodoPublico = dbGetQuery(mibbdd_15, SentenciaSQL_ConsultaTodoPublico)
summary(Consulta_TodoPublico)
```

```
##   TituloEsp           TituloVO
##  Length:1           Length:1
##  Class :character   Class :character
##  Mode  :character   Mode  :character
```

```r
Consulta_TodoPublico
```

```
##                                TituloEsp                      TituloVO
## 1 Robin Hood: el Príncipe de los Ladrones Robin Hood: Prince of Thieves
```

```r
#"Consulta 3"

SentenciaSQL_Peliculas90_00 = "Select TituloEsp, TituloVO from Pelicula where Anio between '1990' and '
Peliculas90_00 <- dbSendQuery(mibbdd_15, SentenciaSQL_Peliculas90_00);
dbGetInfo(Peliculas90_00)
```

```
## $statement
## [1] "Select TituloEsp, TituloVO from Pelicula where Anio between '1990' and '2000'"
##
## $isSelect
## [1] 1
##
## $rowsAffected
## [1] -1
##
```

```
## $rowCount
## [1] 0
##
## $completed
## [1] 0
##
## $fieldDescription
## $fieldDescription[[1]]
## NULL
```

```
print(paste("Consulta Peliculas entre 1990 y 2000:", dbGetStatement(Peliculas90_00)))
```

```
## [1] "Consulta Peliculas entre 1990 y 2000: Select TituloEsp, TituloVO from Pelicula where Anio betwe
```

```
Consulta_Pelicula90_00_condbSendQuery <- dbFetch(Peliculas90_00, n=-1)
print(paste("Número de elementos devueltos en la consulta", dbGetRowCount(Peliculas90_00)))
```

```
## [1] "Número de elementos devueltos en la consulta 4"
```

```
summary(Consulta_Pelicula90_00_condbSendQuery)
```

```
##   TituloEsp           TituloVO
##  Length:4           Length:4
##  Class :character   Class :character
##  Mode  :character   Mode  :character
```

```
Consulta_Pelicula90_00_condbSendQuery
```

```
##                                TituloEsp                      TituloVO
## 1 Robin Hood: el Príncipe de los Ladrones Robin Hood: Prince of Thieves
## 2                 Gladiator(El Gladiador)                     Gladiator
## 3                                 Matrix                        Matrix
## 4                           Mujer Bonita                 Pretty Woman
```

5.6. Tarea 6: Eliminar registros de la base de datos

En esta última tarea se trata de que elimines al menos dos de los registros que has añadido anteriormente. A modo de ejemplo, te sugiero las siguientes eliminaciones, aunque puedes realizar las que consideres más adecuadas:

- Elimina todas las películas protagonizadas por Kevin Costner anteriores a 1992.
- Elimina todos los datos relacionados con Keanu Reeves.

Pistas_:

- Pista 1. Elige bien la tabla de donde vas a borrar los datos
- Pista 2. Algunas de las peticiones realizadas implican borrar datos de una o más tablas (la segunda petición es la que tiene trampa)
- Pista 3. Asegúrate que la base de datos queda consistente tras el borrado
- Pista 4. Comprueba los resultados mostrando la base de datos tras las eliminaciones.

```r
print ("Introduce aquí el código R correspondiente a la eliminación de los datos propuestos")
```

```
## [1] "Introduce aquí el código R correspondiente a la eliminación de los datos propuestos"
```

```r
#"Eliminar películas protagonizadas por Kevin Costner anteriores a 1992"

SentenciaSQL_EliminarProtagonizaCostner ="delete from Protagoniza where CodActor = 'Kcostner' and Anio <
dbSendQuery (mibbdd_15, SentenciaSQL_EliminarProtagonizaCostner)
```

```
## <MySQLResult:-1846944840,0,64>
```

```r
SentenciaSQL_EliminarPeliculaCostner ="delete from Pelicula where CodPelicula = 'rhood'"
dbSendQuery (mibbdd_15, SentenciaSQL_EliminarPeliculaCostner)
```

```
## <MySQLResult:8,0,65>
```

```r
#"Eliminar todos los datos relacionados con Keanu Reeves"

SentenciaSQL_EliminarProtagonizaReeves ="delete from Protagoniza where CodActor = 'Kreeves'"
dbSendQuery (mibbdd_15, SentenciaSQL_EliminarProtagonizaReeves)
```

```
## <MySQLResult:-1846944840,0,66>
```

```r
SentenciaSQL_EliminarPeliculaReeves ="delete from Pelicula where CodPelicula = 'matrix'"
dbSendQuery (mibbdd_15, SentenciaSQL_EliminarPeliculaReeves)
```

```
## <MySQLResult:1,0,67>
```

```r
SentenciaSQL_EliminarActorReeves ="delete from Actor where CodActor = 'Kreeves'"
dbSendQuery (mibbdd_15, SentenciaSQL_EliminarActorReeves)
```

```
## <MySQLResult:8,0,68>
```

```r
tblActor <- dbReadTable(mibbdd_15, "Actor")
SentenciaSQL_TablaPelicula ="Select * from Pelicula"
TblPelicula_condbGetQuery = dbGetQuery (mibbdd_15, SentenciaSQL_TablaPelicula)
TblProtagoniza_condbReadTable = dbReadTable(mibbdd_15, "Protagoniza")
tblActor
```

```
##   CodActor  Nombre Apellido   Nacionalidad
## 1 kcostner   Kevin  Costner estadounidense
## 2   rcrowe Russell    Crowe   neozelandesa
## 3 jroberts   Julia  Roberts estadounidense
## 4 hjackman    Hugh  Jackman    australiana
```

```r
TblPelicula_condbGetQuery
```

```
##   CodPelicula               TituloEsp     TituloVO Publico
## 1      robinh             Robin Hood   Robin Hood      +7
## 2   gladiador Gladiator(El Gladiador)    Gladiator     +12
## 3     prwoman           Mujer Bonita Pretty Woman     +13
## 4       logan                  Logan        Logan     +16
##              Productora Anio
## 1     Universal Pictures 2010
## 2    Dreamworks Pictures 2000
## 3    Touchstone Pictures 1990
## 4 20th Century Fox Films 2017
```

```
TblProtagoniza_condbReadTable
```

```
##   CodActor CodPelicula Anio
## 1   rcrowe      robinh 2010
## 2   rcrowe   gladiador 2000
## 3 jroberts     prwoman 1990
## 4 hjackman       logan 2017
```

```
dbDisconnect(mibbdd_15)
```

```
## [1] TRUE
```