

# CONECTAR SQL A R

Alberto Madin Rivera

17/2/2023

## 1. ¿QUÉ ES SQL?

SQL (Structured Query Language) es un lenguaje de programación diseñado para gestionar y manipular bases de datos relacionales. Permite definir, manipular y controlar bases de datos, crear tablas y relaciones entre ellas, insertar, actualizar y eliminar datos, y recuperar información específica de una o varias tablas.

SQL es un lenguaje estándar, lo que significa que su sintaxis es muy similar en diferentes sistemas de gestión de bases de datos (DBMS), como MySQL, Oracle, Microsoft SQL Server, PostgreSQL, entre otros. Además, SQL es fácil de aprender y se utiliza ampliamente en diferentes ámbitos, como la industria, la investigación y la academia.

## 2. ¿POR QUÉ ES IMPORTANTE SQL?

SQL es importante por varias razones:

- Gestión de datos: SQL es el lenguaje utilizado para gestionar bases de datos relacionales, lo que lo convierte en una herramienta fundamental para almacenar, organizar y recuperar datos de manera eficiente.
- Amplia adopción: SQL es un lenguaje de programación estándar que es utilizado en una gran variedad de sistemas de gestión de bases de datos, lo que lo convierte en una habilidad fundamental para cualquier persona que trabaje con datos.
- Facilidad de uso: SQL es fácil de aprender y utilizar, lo que permite a los usuarios de bases de datos crear, modificar y consultar datos sin tener que escribir un código complejo.
- Acceso a la información: SQL permite a los usuarios de bases de datos recuperar información específica de una o varias tablas, lo que permite tomar decisiones informadas y obtener información útil de grandes conjuntos de datos.
- Análisis de datos: SQL también se utiliza ampliamente en el análisis de datos, permitiendo a los usuarios extraer información valiosa de grandes conjuntos de datos, realizar análisis y visualizaciones, y tomar decisiones basadas en datos.

Es importante porque permite a las empresas y organizaciones almacenar y administrar datos de manera eficiente, lo que a su vez les permite tomar decisiones informadas basadas en información precisa y actualizada.

### 3. CONECTAR SQL A R

Para conectar R con una base de datos SQL se pueden seguir los siguientes pasos:

- Instalar el paquete ODBC en R. ODBC es una interfaz de programación de aplicaciones que permite la conexión con diferentes sistemas de bases de datos. Para instalar el paquete ODBC en R, se puede utilizar el siguiente código:

```
install.packages("odbc")
```

- Configurar la conexión con la base de datos. Para hacer esto, se necesita información como el nombre del servidor, la base de datos y las credenciales de acceso. Esta información se puede guardar en un archivo de configuración ODBC. Por ejemplo, se puede crear un archivo de configuración llamado “my\_db\_config.ini” con la siguiente información:

```
[my_db]
Driver = ODBC Driver for SQL Server
Server = my_server
Database = my_database
UID = my_user
PWD = my_password
```

- Establecer la conexión en R. Para hacer esto, se puede utilizar el siguiente código:

```
library(odbc)
con <- dbConnect(odbc(), dsn = "my_db")
```

Este código establecerá una conexión con la base de datos utilizando la configuración definida en el archivo “my\_db\_config.ini”.

- Ejecutar consultas SQL en R. Una vez que se establece la conexión, se pueden ejecutar consultas SQL en R utilizando el siguiente código:

```
result <- dbGetQuery(con, "SELECT * FROM my_table")
```

Este código recuperará todos los datos de la tabla “my\_table” y los almacenará en el objeto “result” en R. A partir de aquí, se pueden realizar análisis de datos, visualizaciones y cualquier otra operación que se desee realizar en R utilizando los datos recuperados de la base de datos SQL.

Es importante tener en cuenta que, en algunos casos, pueden ser necesarias algunas configuraciones adicionales dependiendo del tipo de base de datos y la versión del servidor de la base de datos que se está utilizando.

## 4. REALIZAR UN LEFT JOIN EN SQL DESDE R

Para hacer un left join en SQL desde R se puede utilizar la función `sqldf` del paquete `sqldf`. Esta función permite ejecutar consultas SQL directamente desde R.

A continuación, se muestra un ejemplo de cómo hacer un left join entre dos tablas en R utilizando `sqldf`:

Suponga que tenemos dos data frames en R, `df1` y `df2`, y queremos hacer un left join en la columna “id”:

```
df1 <- data.frame(id = c(1, 2, 3, 4),  
                  name = c("John", "Jane", "Bob", "Mary"))  
df2 <- data.frame(id = c(1, 2, 5),  
                  age = c(25, 30, 35))
```

Para hacer un left join, podemos usar la siguiente consulta SQL en `sqldf`:

```
library(sqldf)  
  
result <- sqldf("SELECT * FROM df1 LEFT JOIN df2 ON df1.id = df2.id")
```

Este código unirá las dos tablas en la columna “id” utilizando un left join y almacenará los resultados en un nuevo data frame llamado “result”.

Es importante tener en cuenta que `sqldf` requiere que las columnas en los data frames tengan nombres únicos. Si hay columnas con el mismo nombre en ambas tablas, se pueden renombrar antes de ejecutar la consulta SQL. Por ejemplo, si ambas tablas tienen una columna llamada “name”, se puede renombrar una de ellas antes de ejecutar la consulta:

```
colnames(df2)[2] <- "age2"  
  
result <- sqldf("SELECT * FROM df1 LEFT JOIN df2 ON df1.id = df2.id")
```

En este caso, la segunda columna en `df2` se ha renombrado como “age2” para evitar conflictos de nombres.