

**UNIVERSIDAD TECNICA PARTICULAR DE LOJA**

**Facultad de Ingenierías y Arquitectura**

**Fundamentos de Análisis de Datos**

**AUTOR**

Oliver Roberto Saraguro Remache

**DOCENTE**

Ing. Eduardo Encalada

06 de mayo de 2025

**Práctica 2: Cargar datos y generar estadísticas en R, usando librerias "DBI" y "dplyr**

* **Librería DBI**

|  |
| --- |
| La librería DBI en R es una interfaz común para la comunicación entre R y sistemas de bases de datos relacionales (como MySQL, PostgreSQL, SQLite, etc.). Esta librería define una serie de funciones estándar como dbConnect(), dbGetQuery(), dbWriteTable() para realizar consultas, insertar datos, y manejar conexiones de forma segura y eficiente. |

* **Diagrama de tablas de la BD MySQL**

|  |
| --- |
|  |

* **Vista de datos seleccionada para cargar a R (comando SQL)**

|  |
| --- |
| -- SQL  CREATE VIEW vista\_ventas AS  SELECT  c.customerNumber as NumeroCliente,  c.customerName as NombreCliente,  c.country as Pais,  c.city as Ciudad,  c.state as Estado,  o.orderNumber as NumeroOrden,  o.orderDate as FechaOrden,  o.status as EstadoPedido,  od.productCode as CodigoProducto,  p.productName as NombreProducto,  p.productLine as LineaDeProducto,  p.quantityInStock as CantidadStock,  p.buyPrice as PrecioCompra,  od.quantityOrdered as CantidadOrden,  od.priceEach as PrecioUnitario ,  e.employeeNumber AS representanteID,  e.firstName AS Nombre,  e.lastName AS Apellido,  ofi.city AS ciudadOficina,  ofi.country AS paisOficina  FROM customers c  INNER JOIN orders o ON c.customerNumber = o.customerNumber  INNER JOIN orderdetails od ON o.orderNumber = od.orderNumber  INNER JOIN products p ON od.productCode = p.productCode  LEFT JOIN employees e ON c.salesRepEmployeeNumber = e.employeeNumber  LEFT JOIN offices ofi ON e.officeCode = ofi.officeCode; |
|  |

* **Comandos y capturas de lo realizado en R:**
* **Cargar los datos desde MySQL**

|  |
| --- |
| # Instalamos las librerias  install.packages("DBI")  install.packages("RMySQL")  library("DBI")  library("RMySQL")  # Conectamos y se configura la conexion para la bd en MySql  con <- dbConnect(RMySQL::MySQL(),  dbname = "classicmodels",  host = "localhost",  port = 3306,  user = "root",  password = "oliversaraguro")  vista <- paste0(  "SELECT ",  "c.customerNumber as NumeroCliente, ",  "c.customerName as NombreCliente, ",  "c.country as Pais, ",  "c.city as Ciudad, ",  "c.state as Estado, ",  "o.orderNumber as NumeroOrden, ",  "o.orderDate as FechaOrden, ",  "o.status as EstadoPedido, ",  "od.productCode as CodigoProducto, ",  "p.productName as NombreProducto, ",  "p.productLine as LineaDeProducto, ",  "p.quantityInStock as CantidadStock, ",  "p.buyPrice as PrecioCompra, ",  "od.quantityOrdered as CantidadOrden, ",  "od.priceEach as PrecioUnitario, ",  "e.employeeNumber AS representanteID, ",  "e.firstName AS Nombre, ",  "e.lastName AS Apellido, ",  "ofi.city AS ciudadOficina, ",  "ofi.country AS paisOficina ",  "FROM customers c ",  "JOIN orders o ON c.customerNumber = o.customerNumber ",  "JOIN orderdetails od ON o.orderNumber = od.orderNumber ",  "JOIN products p ON od.productCode = p.productCode ",  "LEFT JOIN employees e ON c.salesRepEmployeeNumber = e.employeeNumber ",  "LEFT JOIN offices ofi ON e.officeCode = ofi.officeCode;"  )  # 1 FORMA  practica2 <- dbGetQuery(con, vista)  # 2 FORMA  practica2.1 <- dbGetQuery(con, "SELECT \* FROM vista\_ventas") |
|  |

* **Agregar atributos derivados**

|  |
| --- |
| 1. Total de ventas |
| practica2 <- practica2 %>%  mutate(TotalVenta = CantidadOrden \* PrecioUnitario) |
| 1. Clasificar el total de ventas(baja, media, alta) |
| practica2 <- practica2 %>%  mutate(  ClasificacionVenta = case\_when(  TotalVenta <= 2000 ~ "Baja",  TotalVenta <= 5000 ~ "Media",  TRUE ~ "Alta"  )  ) |

|  |
| --- |
|  |

* **Estadística**

|  |
| --- |
| 1. **Total de ventas por país de cliente** |
| practica2 %>%  group\_by(Pais) %>%  summarise(VentaTotal = sum(TotalVenta)) %>%  arrange(desc(VentaTotal)) %>%  filter(VentaTotal > 200000) %>%  select(Pais, VentaTotal) |
| 1. **Cuales son los 5 productos más vendidos** |
| practica2 %>%  group\_by(NombreProducto) %>%  summarise(TotalCantidad = sum(CantidadOrden)) %>%  arrange(desc(TotalCantidad)) %>%  slice(1:5) |
| 1. **Cantidad de ventas por cada representante de ventas en USA.** |
| practica2 %>%  filter(Pais == "USA") %>%  group\_by(representanteID, Nombre, Apellido) %>%  summarise(TotalVendido = sum(TotalVenta)) %>%  arrange(desc(TotalVendido)) |
| 1. **Saber que productos tienen mayor ganancia promedio por unidad.** |
| practica2 %>%  mutate(GananciaUnitario = round(PrecioUnitario-PrecioCompra, 2)) %>%  group\_by(NombreProducto) %>%  summarise(GananciaPromedio = mean(GananciaUnitario)) %>%  arrange(desc(GananciaPromedio)) %>%  slice(1:5) |
| 1. **Saber las principales 3 líneas de producto que son fuertes en el mercado.** |
| practica2 %>%  group\_by(LineaDeProducto) %>%  summarise(TotalIngresos = sum(TotalVenta)) %>%  arrange(desc(TotalIngresos)) %>%  slice(1:3) |