## Assignment 2: Data Transform Activity – SQL vs Rstudio

Tala Ismaeil

040832377

DAT1030

1 July 2024

SQL

**Step 1: Creating a new table called “sales\_genre”**

# CREATE TABLE sales\_genre ( Name string,

# Platform string,

# Year\_of\_Release int,

# Genre string,

# Publisher string,

# NA\_Sales int,

# EU\_Sales int,

# JP\_Sales int,

# Other\_Sales int,

# Global\_Sales int,

# Critic\_score int,

# Critic\_count int,

# User\_score string,

# User\_count int,

# Developer string,

# Rating string );

**Step 2: Selecting particular columns**

# INSERT OVERWRITE TABLE sales\_genre

# SELECT Genre, Global\_Sales, Critic\_Score

# FROM sales;

#to check if the transformation worked

# SELECT\* FROM sales\_genre LIMIT 10;

**Step 3: Rounding the Global\_Sales**

# INSERT OVERWRITE TABLE sales\_genre

# SELECT Global\_Sales

# ROUND (Global\_Sales, 0)

# FROM sales\_genre

# GROUP BY interger;

**Step 4: Filtering the data to only look at Critic\_Score that is greater than 0**

# INSERT OVERWRITE TABLE sales\_genre

# SELECT Critic\_Score FROM sales\_table

# WHERE Critic\_Score > 0;

**Step 5: Ordering the data by Critic\_Score from highest to lowest**

# INSERT OVERWRITE TABLE sales\_genre

# SELECT CRITIC\_SCORE

# FROM sales\_genre

# ORDER BY Critic\_Score DESC;

RSTUDIO

#install.packages("ggplot2")  
library(ggplot2)  
library(tidyverse)

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.4 ✔ readr 2.1.5  
## ✔ forcats 1.0.0 ✔ stringr 1.5.1  
## ✔ lubridate 1.9.3 ✔ tibble 3.2.1  
## ✔ purrr 1.0.2 ✔ tidyr 1.3.1  
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

library("dplyr")

## Step 1: Loading the dataset into Rstudio.

options(scipen=999)  
sales <- read.csv("Video\_Games\_Sales.csv")  
head(sales)

## Name Platform Year\_of\_Release Genre Publisher  
## 1 Wii Sports Wii 2006 Sports Nintendo  
## 2 Super Mario Bros. NES 1985 Platform Nintendo  
## 3 Mario Kart Wii Wii 2008 Racing Nintendo  
## 4 Wii Sports Resort Wii 2009 Sports Nintendo  
## 5 Pokemon Red/Pokemon Blue GB 1996 Role-Playing Nintendo  
## 6 Tetris GB 1989 Puzzle Nintendo  
## NA\_Sales EU\_Sales JP\_Sales Other\_Sales Global\_Sales Critic\_Score Critic\_Count  
## 1 41.36 28.96 3.77 8.45 82.53 76 51  
## 2 29.08 3.58 6.81 0.77 40.24 NA NA  
## 3 15.68 12.76 3.79 3.29 35.52 82 73  
## 4 15.61 10.93 3.28 2.95 32.77 80 73  
## 5 11.27 8.89 10.22 1.00 31.37 NA NA  
## 6 23.20 2.26 4.22 0.58 30.26 NA NA  
## User\_Score User\_Count Developer Rating  
## 1 8 322 Nintendo E  
## 2 NA   
## 3 8.3 709 Nintendo E  
## 4 8 192 Nintendo E  
## 5 NA   
## 6 NA

names(sales)

## [1] "Name" "Platform" "Year\_of\_Release" "Genre"   
## [5] "Publisher" "NA\_Sales" "EU\_Sales" "JP\_Sales"   
## [9] "Other\_Sales" "Global\_Sales" "Critic\_Score" "Critic\_Count"   
## [13] "User\_Score" "User\_Count" "Developer" "Rating"

## Step 2: Creating a new table called “sales\_genre”.

sales <- sales %>%  
 select(Genre, Global\_Sales, Critic\_Score)  
head(sales)

## Genre Global\_Sales Critic\_Score  
## 1 Sports 82.53 76  
## 2 Platform 40.24 NA  
## 3 Racing 35.52 82  
## 4 Sports 32.77 80  
## 5 Role-Playing 31.37 NA  
## 6 Puzzle 30.26 NA

## Step 3: Rounding the Global\_Sales column.

sales$Global\_Sales <- round(sales$Global\_Sales, 0)  
head(sales)

## Genre Global\_Sales Critic\_Score  
## 1 Sports 83 76  
## 2 Platform 40 NA  
## 3 Racing 36 82  
## 4 Sports 33 80  
## 5 Role-Playing 31 NA  
## 6 Puzzle 30 NA

## Step 4: Filtering the data to only look at Critic\_Score that is greater than 0.

sales <- sales %>%   
 filter(Critic\_Score>0)  
  
head(sales)

## Genre Global\_Sales Critic\_Score  
## 1 Sports 83 76  
## 2 Racing 36 82  
## 3 Sports 33 80  
## 4 Platform 30 89  
## 5 Misc 29 58  
## 6 Platform 28 87

## Step 5: Ordering the data by Critic\_Score from highest to lowest.

sales <- sales %>%   
arrange(-Critic\_Score)  
  
head(sales)

## Genre Global\_Sales Critic\_Score  
## 1 Action 11 98  
## 2 Action 10 98  
## 3 Sports 5 98  
## 4 Fighting 0 98  
## 5 Action 21 97  
## 6 Action 16 97