Luis Valderrama

January 12, 2022

DATAANA 320 A

Lesson 02 Assignment

SQL & Data

# Intro

In Lesson 02, we revisit basic SQL statements, commands, clauses, operators, and how these SQL uses can be leveraged and applied in RStudio to which are utilized in the case study for the retail firm AdventureWorks Inc. This document addresses the business case, business questions, steps to get the data while contrasting SQL and R. Finally, analysis and a recommendation are presented.

# Case Study and Business Question

The retail firm AdventureWorks Inc. is relocating their Research and Development unit to streamline the unit and decrease corporate pressure. A recommendation is requested to select a new location.

The business question presented in the Lesson 02 Assignment document is based on the premises of geographical proximity among the employees in this unit, and is as follows:

***“What location would you recommend moving the Research and development unit?”***

The question is answered below under [Analysis and Recommendation](#_Analysis_and_Recommendation).

# Data Sources

In this section, a contrast of SQL and RStudio functions to get data.

## Get Data in SQL

Using SQL, I examined the data available in the **AdventureWorks2016CTP3** database. The three **schema.tables** used for this assignment are: HumanResources.Employee, Person.BusinessEntityAddress, and Person.Address. (Figure 1)

Graphical user interface, text

Description automatically generated

***Figure 1: View of schema.tables from* AdventureWorks2016CTP3 DB.**

## Get Data in RStudio

In RStudio, I use .csv files instead of connecting to the database and extracting data from the tables. The files (Employee, BusinessEntityAddress, and Address). The files are saved on my directory for impot. But first, the necessary packages and libraries are installed, including **‘sqldf’** which enables a “SQL-like” programing language. (Figure 2).

## Step 1: Install Packages and Libraries

install.packages( "data.table" )

install.packages( "sqldf" )

install.packages( "curl" )

library(data.table) # Loading `data.table` package

library(sqldf) # Loading `sqldf` package

library(curl) # Loading `curl` package

library(tidyverse) # Loading tidyverse

***Figure 2: Installing Packages and Loading Libraries in RStudio.***

A folder\_path function saves the path to where the data files are located. (Figure 3)

## Step 2: Data File Folder Location Setup

folder\_path = "C:/Users/lvald/OneDrive/Desktop/"

***Figure 3: Creating Folder Path to Data Files.***

The .csv files are read using **fread** function from **data.table**, followed by a quick data glimpse. (Figure 4)

## Step 3: Read Data Files

# Read .csv filesRead .csv files

Employee = fread(paste(folder\_path, "Employee.csv", sep = ""), header = TRUE)

BusinessEntityAddress = fread(paste(folder\_path, "BusinessEntityAddress.csv", sep = ""), header = TRUE)

Address = fread(paste(folder\_path, "Address.csv", sep = ""), header = TRUE)

# glimpse the data in the .csv files

Employee %>% glimpse()

BusinessEntityAddress %>% glimpse ()

Address %>% glimpse()

***Figure 4: Reading CSV Files using fread (data.table).***

# Data Wrangling and Transformation

In this section, the data is transformed in both, SQL and RStudio for comparison.

## Selecting Data, Table Joins and Filtering In SQL

Using a **SELECT** statement, I extract the attributes from the tables. (Figure 5)

--Attribute selection

SELECT

[BusinessEntityID] = E.BusinessEntityID

,[LoginID] = E.LoginID

,[JobTitle] = E.JobTitle

,[City] = A.City

***Figure 5: Attributes SQL Code.***

Since some of the attributes were presented in the various tables, it was necessary to join the tables using **HumanResources.Employee** as the main table, applying an **INNER JOIN**, I joined **Person.BusinessEntityAddress** “bridge” table to the **HumanResources.Employee** using the attribute, **BusinessEntityID,** as their unique identifier, and with an **INNER JOIN**, the **Person.Address** table is joined to the **Person.BusinessEntityAddress** using the attribute, **AddressID** as unique identifier. (Figure 6)

--Joins and Tables

--HR.Employee table PK = BusinessEntityID (The AdvWorksOLTPSchemaVisio schema doc shows EmployeeID as PK??)

FROM HumanResources.Employee AS E

--Person.BusinessEntityAddress was not found in the dvWorksOLTPSchemaVisio schema doc. Assume the Foreign Key = BusinessEntityID

INNER JOIN Person.BusinessEntityAddress AS BEA

ON E.BusinessEntityID = BEA.BusinessEntityID

--Person.Address PK = AddressID

INNER JOIN Person.Address AS A

ON BEA.AddressID = A.AddressID

***Figure 6: Joins and Tables SQL Code.***

The initial result displayed 290 rows of data which included numerous job title. Since the request on the lesson was to narrow the scope to isolate the employees involved in Research and Development it was necessary to filter the data using **WHERE** clause, plus **LIKE** operator and ‘%**’** wildcard enabling the filtering of the employees’ job titles, targeting the job titles starting with ‘Research’. The data is ordered by job title using **ORDER BY** Command as shown below on (Figure 7).

--Filtering data by Job Title starting with 'Research...'

WHERE E.JobTitle LIKE ('Research%')

ORDER BY E.JobTitle, A.City

;

***Figure 7: Use of WHERE Clause, LIKE operator, ‘%’ Wildcard, and the ORDER BY.***

## Selecting Data, Table Joins and Filtering in RStudio

As mentioned above, installing the package named **‘sqldf’** enables the use of “SQL-like” syntax. For example, the tables (.csv files) are joined using the **merge** command and the corresponding unique identifiers**. (**Figure 8)

## Step 4: Joining Tables

# Inner Join Employee and Business Entity Address using BusinessEntityID PK/FK

df\_Employee\_BusinessEntityAddress = merge(Employee, BusinessEntityAddress, by = c( "BusinessEntityID" ))

#Inner Join of the Address table to the newly created dataset for Employee and BusinessEntityAddress using AddressID as PK/FK

df\_Employee\_BusinessEntityAddress\_Address = merge(df\_Employee\_BusinessEntityAddress, Address, by = c("AddressID"))

# Glimpse the dataset

df\_Employee\_BusinessEntityAddress\_Address %>% glimpse()

***Figure 8: Table Joins Using RStudio.***

Other functions in SSMS can be applied in RStudio to select, filter and order the data which in my example they are used to extract the columns, filter the data based on job title and arrange the results by job title. The final extract is then written to a .csv. (Figure 9)

## Step 5: Select, Filter and Arrange

# R&D Unit

df\_RandD\_Unit<- df\_Employee\_BusinessEntityAddress\_Address %>%

select(BusinessEntityID, LoginID, JobTitle, City) %>%

filter(grepl('Research', JobTitle)) %>%

arrange(JobTitle)

# View the final results

df\_RandD\_Unit%>% head()

## Step 6: Write the Final Results to a .csv

write.csv(df\_RandD\_Unit, "LuisValderrama-L02\_AssignmentResults.csv", row.names = TRUE)

## END

***Figure 9: Selecting, Filtering, Arranging Data, and Writing to File in RStudio.***

# Analysis and Recommendation

There are many factors to consider when making a recommendation or a major decision such as relocating a department, including but not limited to financial, employee seniority, skillsets, hierarchy, proximity to suppliers and major clients, proximity to major airports and other transportation option, as well as other external collaborators, etc.… For the purposes of this analysis and recommendation we are focusing solely on geographical location convenience.

The results generated by the SQL and RStudio queries as detailed in the sections above suggest that four employees are currently working in Research and Development, two Engineers and two Managers, and three out of four employees live in the Seattle greater area. Specifically, Everett, Kenmore, and Bellevue, while one lives in the San Fransico area. (Figure 10)

Graphical user interface

Description automatically generated

***Figure 10.***

With that, the initial plausible recommendation is to relocate based on geographical tiers. Tier 1 being by state, which choosing Washington state would be most convenient as it only requires the relocation of one employee. The tier 2 is relocation by city. The Seattle greater area is one of the largest cities in the Pacific Northwest with access to resources that could facilitate the Research and Development department such as technology, transportation, and other tools important for this department. The city showing adequate geographical characteristics is Bothell which is established near the intersection of a major Interstates I405 and Hwy 522, but also have relatively close proximity to I5 and other major state highways. The city of Bothell is also near the half-way point between Everett and Bellevue to which are the cities where two of the four employees reside while the employee living in Kenmore will have a relatively short commute. The total distance between Everett and Bellevue is approximately 28 miles, while Bothell is just about 8 miles south of the half-way point. Additionally, the employee relocating from San Francisco could consider the Bothell area for relocation.