Luis Valderrama

April 10th, 2021

IT FDN 130 A

Module 01

Databases and Data Tables

# Intro

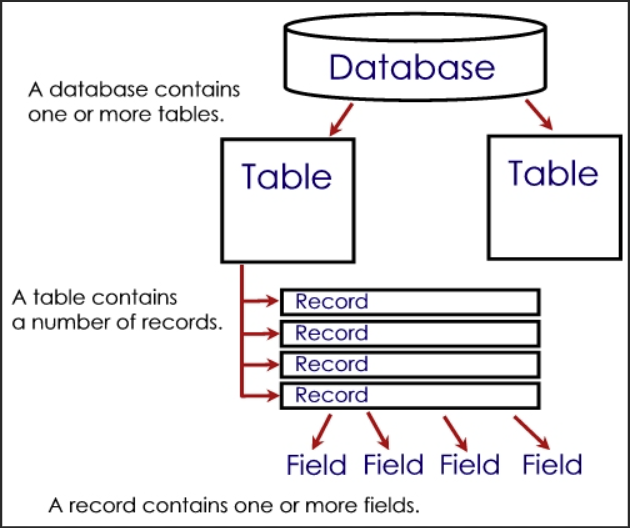
During the course of the first module, I learned the basic concepts and methods to create databases and tables using MS SQL Server Management (MS SQL). This document addresses the basic concepts and difference between databases and data tables, and presents the steps to create my databases, tables, and testing of the SQL script. The process I followed took into consideration the rules of normalization and best practices learned throughout this first module.

# Database Vs. Data Table

*“A database is an organized collection of data. More specifically, a database is an electronic system that allows data to be easily accessed, manipulated and updated”.* (<https://www.techopedia.com/6/28832/enterprise/databases/introduction-to-databases#introduction>, 2020) (external site). Companies, regardless of asset size use relational databases as not only reliable and efficient way to store, retrieve, manage data but also as a powerful business analytics and decision-making tool.

*“The table is the basic data-storage in a relational database*. Tables consist of columns and rows. The columns are attributes or qualities that we want to express, while the rows hold the actual data, with one (or no) items per row”. (<https://www.techopedia.com/6/28832/enterprise/databases/introduction-to-databases#introduction>, 2020) (external site).

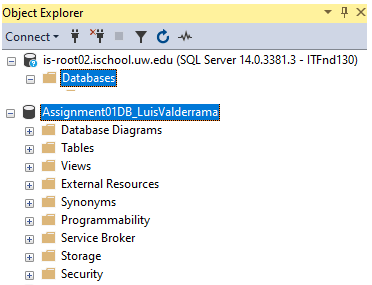
The primary difference between databases and data tables is the fact that databases hold or houses a large number of tables, while tables hold the specific information of the subject in the form of records and attributes (Figure 1). The database not only allows the storage of the tables but it has the mechanism to dynamically interact with the tables with the use of Relational Database Management System (RDMS).



***Figure 1: Database and tables visualization.***

# Creating the Database

There are various ways and applications utilized to create databases, although this document focuses on the two methods of creating databases using MS SQL learned throughout the course of this module. The method I chose to create my first database named *Assignment01DB\_LuisValderrama* was by right click on Database’s folder located in the server, selecting New Database, and assigning the name. (Figure 2.1)



***Figure 2.1: Assignment01DB\_LuisValderrama created within Databases’ folder.***

For my second database, *Mod01LabsLuisV,* I chose to write SQL codes. I began the process by entering a heading and high-level step-by-step of my data design in the form of Notes as a way for the reader to have better visualization of the functions ahead. (Figure 2.2)

--\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--Title: Creating Database and Table Mod01LabLuisV

--Desc: This file is in reference to Mod01Lab #1

--Change Log: 2021.04.08, Luis Valderrama, Created Database,

--two Tables and modified VALUES

--\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--[Data Design]--

/\*

1. Open the SSMS or ADS and connect to a SQL Server.

2. Create a database called Mod01LabsLuisV

3. Change connection's focus to use the new database with the Use command.

***Figure 2.2: Heading and data design.***

Next, I entered the commonly used SQL command, **CREATE DATABASE** (Mod01LabLuisV) and execute. I then changed the focus to the newly created database by entering the **USE** (Mod01LabLuisV) command. (Figure 2.3)

CREATE DATABASE Mod01LabsLuisV;

GO

USE Mod01LabsLuisV;

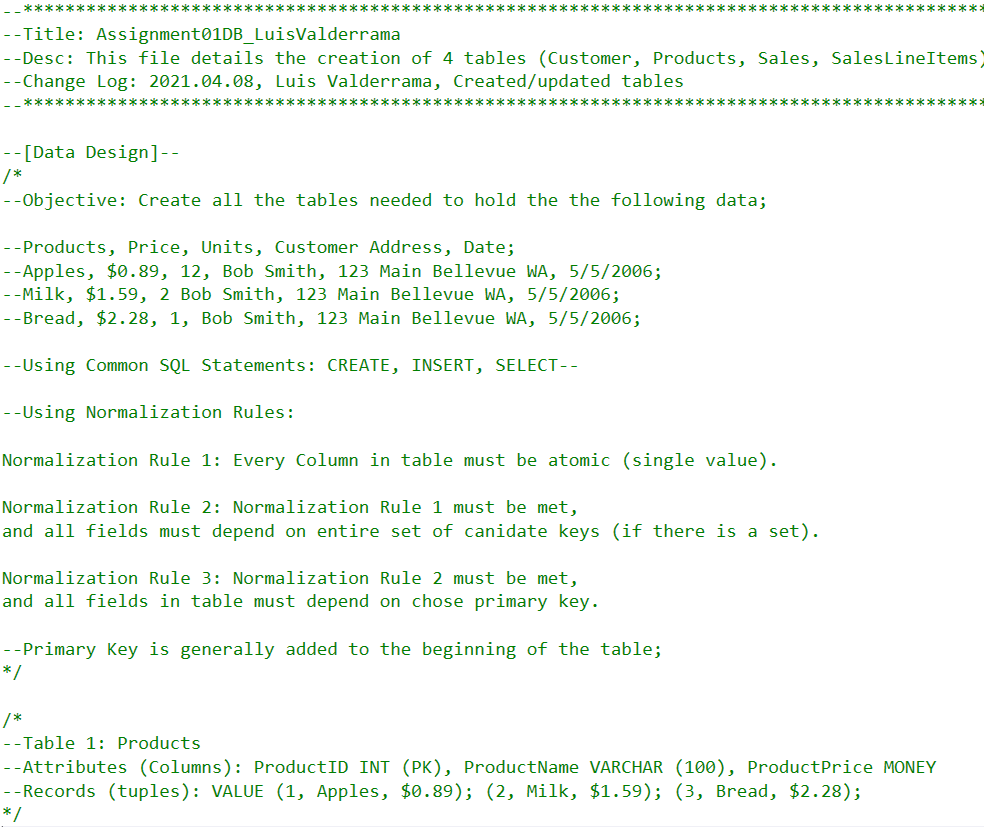
GO

***Figure 2.3: Creating database Mod01LabsLuisV using SQL command codes.***

Note: Regardless of the method to create the database (or tables) it is necessary to refresh the Object Explorer in order for the new database to be visible.

# Creating, and Inserting Values in the Data Table

Similar to creating databases using MS SQL there are also various ways of creating a table. For the purpose of this document, I focus on creating tables by way of writing commonly used SQL codes learned throughout the first module. I find it valuable to first insert a heading, objective and high-level outline of my table design in the form of Notes as a way for the reader to have better visualization of the functions ahead. (Figure 3.1)



***Figure 3.1: Heading and data design helps the user have better visualization.***

As an example, one of the four tables created for this assignment was ‘Products’ using the **CREATE TABLE** (Products)SQL command, followed by entering the column names: ProductID, ProductName, and ProductPrice, Followed by the data types **INT** for the ID column and **VARCHAR** for the remaining columns. Note that ProductID is set up as **PRIMARY KEY**. After the action was executed and Object Explorer refreshed, I added the datapoints to the table using **INSERT INTO** (Products). (Figure 3.2)

/\*

--Table 1: Products

--Attributes (Columns): ProductID INT (PK), ProductName VARCHAR (100), ProductPrice MONEY

--Records (tuples): VALUE (1, Apples, $0.89); (2, Milk, $1.59); (3, Bread, $2.28);

\*/

CREATE TABLE Products

(ProductID INT PRIMARY KEY, ProductName VARCHAR (100), ProductPrice MONEY);

GO

INSERT INTO Products

VALUES (1, 'Apples', 0.89);

GO

INSERT INTO Products

VALUES (2, 'Milk', 1.59);

GO

INSERT INTO Products

VALUES (3, 'Bread', 2.28);

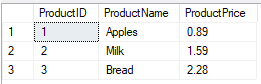
GO

SELECT \* FROM Products

GO

***Figure 3.2: Data design notes, table creating and datapoints addition codes.***

As a quick validation step to ensure the actions took place as intended, I ran a **SELECT \* FROM** (Products) SQL command in order to display the results. (Figure 3.3)

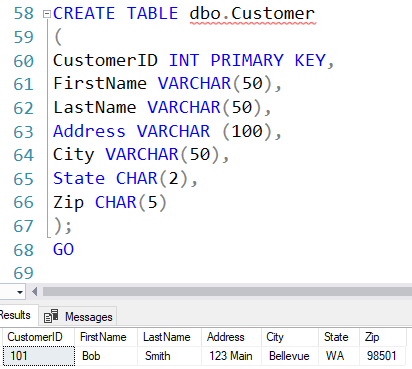


***Figure 3.3: Results of the Product table.***

The four tables created for this assignment were: Customer, Products, Sales, and SalesLineItems. Overall, the creation process is similar to what is presented on page 3 of this document. Although, it took several attempts to get the data to display in way that made sense to the user.

For example, the objective of the assignment was to create as many tables as needed to display a sale to one customer, Bob Smith consisting of three products with different prices, and quantities. Additionally, we have the address information for this customer. After applying ([Database Normalization Rules, or "Normal Forms"](https://docs.microsoft.com/en-US/office/troubleshoot/access/database-normalization-description), 2020) (external site), as well as [Table Relationship Fundamentals](https://docs.microsoft.com/en-us/office/troubleshoot/access/define-table-relationships)*,* 2020) (external site) and the application of primary keys, it was best practice to break the data into the four tables previously mentioned.

Additionally, I learned it was best practice to further break the Customer table into seven attributes in order to eliminate multi-value field and maximize the granularity of the data as shown below. (Figure 3.4)



***Figure 3.4: Customer table script.***

Lastly, a challenging table to create was the SalesLineItems which was intended to be about the sales transaction, more so, than the products or customer. Although, it was necessary to assign a **PRIMARY KEY** to two columns (SalesID and LineItemID) making it a Composite Primary Key to which all columns depend on this.

CREATE TABLE dbo.SalesLineItems

(SalesID INT, LineItemID INT, ProductID INT, Qty INT,

PRIMARY KEY (SalesID, LineItemID) --<Note: Composite Primary Key

);

GO

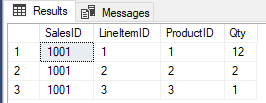
INSERT INTO dbo.SalesLineItems

VALUES (1001, 1, 1, 12),

(1001, 2, 2, 2),

(1001, 3, 3, 1);

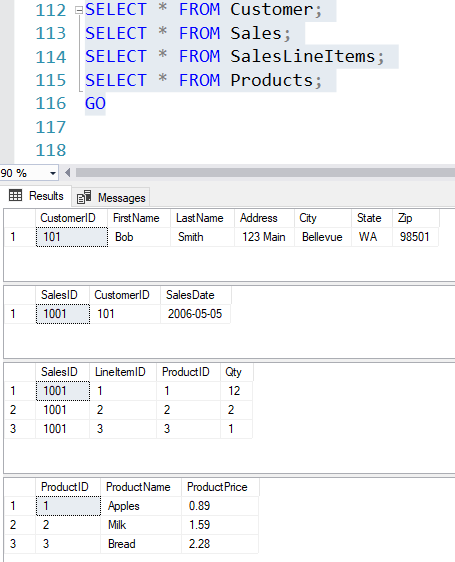
GO



***Figure 3.5: SalesLineItems using a Composite Primary Key***

# Running and Testing the SQL Script

At this point, the database and four data tables have been entered, and updated with datapoint values consistent with the assignment. In order to test the accuracy of the SQL script I used the **SELELCT \* FROM** (tables names), as shown below there are no duplicates, errors and the data returned appears consistent with the directions of the assignment. (Figure 4.1)



***Figure 4.1: SQL Script test run of all tables created within the Assignment01DB\_LuisValderrama database.***

# Summary

To recap, the first module introduced me to the basic concepts and methods to create databases and tables using MS SQL and the difference between them, in addition, to properly create the tables’ attributes by applying Normalization Forms, and taking into consideration Relational Database fundamentals. In my first assignment I was able to create four tables in one database and after several attempts, I understood the best practices in writing the SQL scripts. The testing of the SQL script revealed that the creation process I followed was successful as there were no errors or data duplication displayed in the results.