Jong Tai Kim

11/11/2018

Info 330

Assignment 6 - Midterm

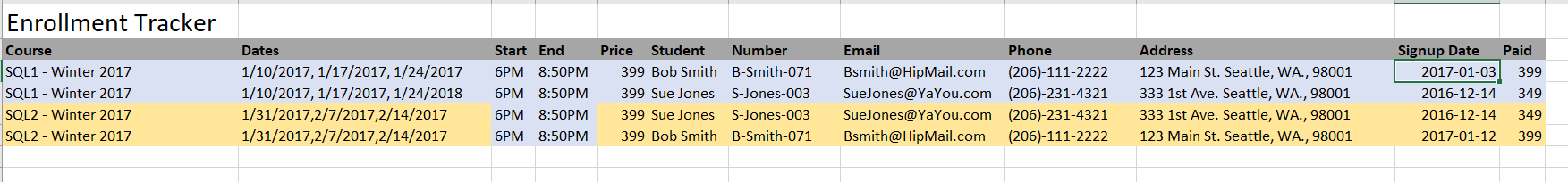
Relational Database

Introduction

For this paper, we will be examining the process of designing and creating relational database based on the information we have learned from the class.

Process of Designing and Creating Relational Database

In most of the time, the raw data we need to analyze is in form of a series of difference information, such as the record of the purchases of products by customers. The problem is that when we try to access those data one record by one, it is not intuitive for us to understand the data as whole data --for instance one record holds too much information at once-- and also it is very redundant that every record would hold customers and products -- same customers may buy again and/or same products are bought again later. To solve the issue of redundancy and , we will apply normalization to the data by creating multiple table that will hold single subject, such as customers table, products table, and orders table for the example above. To normalize, we will require to address that the primary key of given table that will represent the uniqueness of row, such that the columns will be dependent on the primary key.



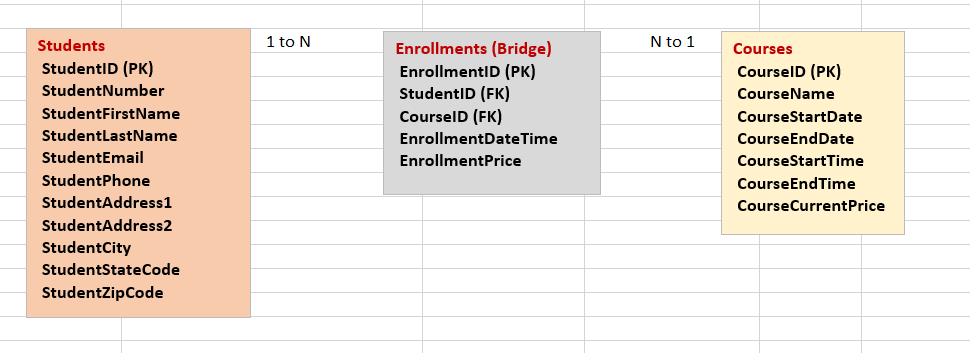


Figure 1,2 before and after normalization of given enrollment data (Root R, Assignment 06 Data and MetaData, 2018).

For instance of the assignment 06, where we were given for normalized tables with pre-normalized data like in the figures above, this shows how the data has been translated into the relational database. The raw data contains all the courses and students and students’ enrollment. To keep the properties of normalization on primary key, it is crucial for us to create bridge table, which will combine the data of students and courses so that we can describe the information of enrollment of each student on each course. Thus, both data of students and courses from students and classes tables would be represented as foreign keys in enrollment table such that data of each student’ enrollment on each course would not violate normalization rule.

Then, after figuring out what to be normalized into tables, we will now create the column of table with data type and the constraints of the tables to not only specify requirement of each column of table but also avoid potential corrupted data into the tables.

In addition, directly accessing to the created table is not a good idea as the tables’ main objective is to store the data. Thus, to do other objectives such as to present the data, it is recommended to create other abstract layers; for presenting the data, we will create views, and for creating, updating, and deleting data, we will create stored procedures. By this way, we will not directly access data to create, read, update and delete the data of the table such that it will increase the security of the data stored in the tables as well as reduce the redundancy of calling tables every time to do the objectives other than the main objective of the table; in other words, it increases the security and the efficiency of our codes creating database. For efficiency, we will avoid calling the stored data directly and other abstract layers would do objectives other than storing data. For security, if we can deny access of the non-authorized users on the stored data and only make them access on the views (or other abstract layer which would not corrupt database), we will make sure that showing the data to strangers would not make them potentially destroy our database.

Summary

In conclusion, this paper have examined on the process of normalization, creating tables with constraints, and assigning abstract layers for each crucial objective and also look at the cases from the lectures and assignments that marks the significance of the normalization and constraints and abstract layers of database.