# **Learning Journal - Unit 4**

Computer Science, University of the People

CS 2203-01 Databases 1 - AY2024-T3

Instructor, Irfan Rashid Thoker

February 29, 2024

This week as always was full of new challenges and extensive learning. This week's studies focused mainly on normalization. Both how best to find and implement the different levels of normalization and how to implement and modify our existing structures to make them more efficient. During the week's tasks I had to take our previous data structure and implement the three main normalization levels on them (1NF, 2NF, and 3NF). I had to make sure that I understood the reasons and goals involved in each normalization level and how to properly implement the level to make sure that I improve my database's performance and efficiency and not accidentally harm it and make the performance and efficiency worse.

I learned how to take my existing structure and why it is important to make sure the normalization processes are done correctly and according to the standards set. By taking the existing tables and breaking them down into atomic groups of columns, removing partial and incomplete dependencies and finally getting rid of transitive dependencies. I was able to change the structure into a more reliable, efficient and robust data structure that will better support the current system and future changes.

All this accumulated itself into the programming task for this week where I took all the work from the week and produced a new data structure and ERD for the hospital system that we have been working on in the course.

An interesting issue that I found was working with dates and times in databases. These data types can be a source of much agony for database designers and users alike. The main issues with date-time data types are:

1. Time zones, when designing a global system we need to take into account that every user might be in a different time zone, and so we might have weird issues where data is created “before” the time in a different location. This can create some very weird effects and is a very bad issue when looking at things from a data integrity point of view.
2. Daylight savings, similar to time zones, daylight savings can happen in different parts of the world at different times of the year. Some countries have a pre-set schedule while others might not. This too can cause cases where the time and day that data is created could be inaccurate and inconsistent. Again this makes our data very unpredictable and lacking in integrity.
3. Finally, format, the format of dates and times can change from region to region. Sometimes even in the same region, different users might just choose to use a different format for personal reasons. This makes it hard to predict what date format we might be getting or using. For example, 01-02-2020 might be DD-MM-YYYY or might be MM-DD-YYYY. There is no real way to know from just looking at the data.

Luckily there are some simple and easy ways to circumvent these problems. The main and best solution is to work in a standardized and universal (at least on Earth) format and method of describing a date and time data structure. The UTC, Coordinated Universal Time, is a great tool for making sure we all work with the same format, no matter where we are on the planet or what our current settings are. The UTC format is very simple to understand and works from the largest value to the smallest value, for example, 2024-02-28T20:51:52Z.

This however is not the only tool available to us to help solve the issues revolving around dates and times. Most databases have functions built in to help mitigate the date and time formatting issues and calculations. For instance, SQL Server has the TIMESTAMP datatype which can help with issues of using regular date times.

In conclusion, this week I have learned the theory behind normalization and on top of that I have learned and practiced the practical aspects by implementing the normalization on the hospital system we are working on in the course. On top of that, I have learned about the pitfalls and issues that might arise when working with day-to-day data types such as date time and how seemingly innocent types might hold in them deeper issues and surprises that I need to be aware of.

.

## References

* Learning Guide Unit 1-4  
  <https://my.uopeople.edu/course/view.php?id=7455>
* Sharma, N., Perniu, L., Chong, R. F., Iyer, A., Nandan, C., Mitea, A. C., Nonvinkere, M. & Danubianu, M. (2010). Database fundamentals. IBM Canada.  
  <https://my.uopeople.edu/pluginfile.php/1827130/mod_book/chapter/484065/Database_Fundamentals.pdf>
* Watt, A., & Eng, N. (2014). Database design,  2nd ed. BCcampus, BC Open Textbook Project.   
  <https://opentextbc.ca/dbdesign01/>  
  <https://my.uopeople.edu/pluginfile.php/1827130/mod_book/chapter/484065/Database-Design-2nd-Edition-1560272109.pdf>