**Discussion Question: Normalization**

For this module's discussion board assignment respond to **one**the following topics:

1. Can a database be in Third Normal Form (3NF), without achieving First Normal Form (1NF)? Why or why not?

After reading our resources for the week, we learned more about the First Normal Form, Second Normal Form, and Third Normal Form. When dealing with databases, we cannot achieve 3NF without first achieving 1NF. In fact, we must also achieve 2NF to reach 3NF. They are essentially phases; without the first and second phases, we cannot proceed to the third. For example, in 1NF, according to Learn Learn Scratch Tutorials (2020), 1NF must include unique rows and single value cells, which are broken down as much as possible. According to Comeau (2016), the unique rows are known as “atomicity,” and cells holding only single values are referred to as “scalar” (p. 94). Once this is achieved, we can move to 2NF. In the Second Normal Form, all requirements of the First Normal Form must be met as well: “All non-prime attributes should be fully functionally dependent on the candidate key” (Learn Learn Scratch Tutorials, 2020). After 1NF and 2NF are achieved, 3NF occurs when only the primary key can deter all fields (Learn Learn Scratch Tutorials, 2020).

**References**

Comeau, A. (2016). *MySQL explained: your step-by-step guide to database design*. Ostraining.

Learn Learn Scratch Tutorials. (2020). 1st, 2nd and 3rd Normal Form (Database Normalisation) [YouTube Video]. In *YouTube*. https://www.youtube.com/watch?v=J-drts33N8g

1. At what point can the data in a field be considered atomic? Is it possible to carry atomicity too far and how do you know this has happened? Provide an example not found in the text.
2. What is the purpose of a foreign key? Can you have more than one? Provide an example.
3. In an employee database would you store a person's physical age, their date of birth or both? Why or why not?

***Before you submit your thread, put your name in the subject line.***

**Assignment Requirements and Grading:**

1. An initial post is due by **Thursday, 11:59 p.m., CST**.
2. For the initial post to be considered substantive, it should fully cover the topic(s) being presented. Single sentence definitions or responses will not be awarded points.
3. Submit your post by clicking on the **Assignment Link** above, then **Create Thread**. You must create a thread in order to view your peers' posts. Tip: Create your post in a word document and then copy and paste your work into the thread.
4. A minimum of three (3) responses, **to the original threads of other students**,, of 100-200 words each are due by **Sunday, 11:59 p.m., CST**.
5. To view the rubric grading criteria, click on the following link: [Discussion Board Grading Rubric.](https://content.bellevue.edu/cst/csd/rubricdbv3.pdf)

**(50 points)**

Scott, I think you did a great job explaining your answer in your post. I completely agree with you that storing a person’s date of birth rather than their age is much more practical. I think it could be beneficial to include the employee’s age and change the code so it automatically calculates the employee's age and adjusts as time passes. But for the most part, this may just be unnecessary extra work. As you mentioned, it would be highly inefficient to constantly update an employee’s age in the system when the complete birth date serves as a much better option.

Samir, you did a very nice job explaining why a database cannot be in the Third Normal Form without first being in the First Normal Form. Learning more about this helped me when completing our modules this week! I do think that it is important to note that before we get to the Third Normal Form, we also have to get into the Second Normal Form. This means that along with all the rows being unique and the cells all single values. After this, it must be dependent on the candidate key. Finally, all data must be met with the primary key.

Jacob, you did a nice and thorough job on your post this week! I agree with your consensus that storing a person’s date of birth is a much better option than storing physical age. I like how you mentioned that there could be inconsistencies that arise when storing both the physical age and birth date. I think this is especially true if someone were to export the data to a spreadsheet or other file like a PDF. This makes it easy to mistake someone’s age, which could be a bad look for an HR department if they mess up an employee’s information.