Janis Gonzalez

WEB 335 Introduction to NoSQL

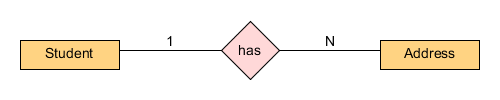
Discussion 3.1 Normalization

March 30,2023

Bellevue University

Database normalization is simply the efficient organization of data in a database. The two goals of the normalization process are to eliminate data that is redundant and ensuring that the data dependencies make sense (Chapple, 2022). Both of these goals help reduce the amount of space a database uses and to ensure that the data is logically stored. There are a series of guidelines that have been developed to ensure that databases are normalized and are referred to as normal forms which are numbers from one through five. The lowest normalization form would be one while the highest form of normalization would be five. The first normal form, or 1NF, sets the fundamentals for an efficiently organized database. This includes eliminating duplicate columns from the same table and creating separate tables for each group of related data as well as identifying each row with a primary key (Chapple, 2022). The second normal form, or 2NF, further addresses the deletion of duplicative data from the database. This would include all the requirements from the first normal form as well as removing subsets of data that apply to multiple rows of a table and then placing those in separate tables (Chapple, 2022). It would also include creating relationships between these new tables with the use of foreign keys. The third normal form, or 3NF, includes all the requirements for 2NF as well as 1NF as well as removes columns that are not dependent on the primary key (Chapple, 2022). There is a third and a half normal form known as the Boyce-Codd Normal Form, 3.5NF, which includes all the previous requirements as well as including that every determinant must be a candidate key (Chapple, 2022). The fourth normal form, or 4NF, meets all the previous requirements as well as understanding that it had no multi-valued dependencies (Chapple, 2022). In MongoDB one-to-many relationships are represented by documents that are embedded within one another with the embedded document model (Sugandhi, 2023). Rather than creating two separate documents, one of the documents is embedded into another which will assist the user in retrieving data with a single query rather than multiple queries.

// Student documents   
{   
 \_id: "kate”,   
 name: “Kate Spengle”,   
 addresses: [   
 {  
 street: “4325 Bacon Street”,   
 city: "San Diego”,  
 state: “CA”,   
 zip: “32423”  
 },   
 {  
 street: “28934 Sparkle Road”,   
 city: "Los Angeles”,   
 state: “CA”,  
 zip: “45435”   
 }   
 ]   
}



*Figure 1. A student has many addresses*

References

Chapple, M. (2022, February 23). *The basics of database normalization*. Lifewire. Retrieved from <https://www.lifewire.com/database-normalization-basics-1019735>

Sugandhi, A. (2023, January 24). *MongoDB relationships: A quick guide*. KnowledgeHut. Retrieved from <https://www.knowledgehut.com/blog/web-development/data-relationships-in-mongodb>