**Discussion Question: Business Rules + ERDs**

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

1. Explain the primary purpose for constructing a data model for a database. Is a data model always necessary? Why or why not?
2. What are business rules? And how do they relate to creating data models? Provide an example.
3. What is the difference between an Entity Relationship Diagram and an Object Relationship Diagram? Provide an example of when you might use each.
4. What is cardinality in data modeling? How many types of cardinality depiction are there? Which would you prefer and why?

Running into a data model can typically be expected throughout the construction of a database. The primary purpose for constructing a data model for a database is that it allows you to “organize your data to match your application’s needs” (MongoDB, n.d.). A data model is not always necessary. It aids in the process by helping to organize information and data, but it is not required. When tackling more extensive projects, having a data model makes the process easier and less confusing, so if opting out of a data model, it would be more advised on smaller rather than big projects.

Business rules are established by companies that help define the requirements for a project (Comeau, 2016, p. 70). Business rules are related to creating data models because programs need to uphold the same standards and practices as the business for which the program will be used. For example, companies must follow strict rules regarding legality, so a program must implement the rules in the application.

There are different diagrams we can use, and examples are provided to show how we might use each. The Entity Relationship Diagram exclusively depicts entities in an information system, such as their attributes and relationships (Comeau, 2016). Within an ERD, processes or other procedures are not included (Comeau, 2016). An example of when the ERD may be used would be a user and their contact list. It would include all the information of the user and the many relationships to other contacts but no processes. The Object Relationship Diagram focuses on how objects or entities interact with each other. Unlike ERD, ORD includes the functions of the objects mentioned. For example, we could use ORD to determine how an online library catalog should run. It would include the functions of the objects, like how digital rentals are checked out.

Cardinality within data modeling “describes a fundamental characteristic of the relationship between two entities” (Alalouf, n.d.). There are three types of cardinality depiction: one-to-one, one-to-many, and many-to-many (Alalouf, n.d.). Out of the ones mentioned, I prefer one-to-many. I prefer one-to-many because it allows us to show the relationship between two entities, so every time one entity occurs, there are many occurrences of the related entity. However, depending on the need, my answer would change.

**References**

Alalouf, A. (n.d.). Cardinality in Data Modeling: A quick guide to understanding relationship types in database design. In *Temple University*. Retrieved October 28, 2024, from https://community.mis.temple.edu/mis3506digitaldesignfall2018/files/2018/10/Adam-Alalouf\_Cardinality.pdf

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

MongoDB. (n.d.). *Data Models — MongoDB Manual*. Www.mongodb.com. Retrieved October 28, 2024, from https://www.mongodb.com/docs/manual/data-modeling/

***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).

Megan, you did a nice job on your post! I agree with you that a data model is not always necessary. You are spot on when you say that it helps to get all of your ideas out of your head. It also helps map out an application beforehand, making it easier to execute the project. I like how you included that this could benefit teams and collaborators. This could also be true for other stakeholders like potential businesses and users. Having a data model is a way to show others the extent of the program without necessarily needing the actual program up and fully running.

Madison, I really enjoyed reading your post and think you did an excellent job elaborating on business rules. I am glad that you included how business rules can dictate how data is created, stored, and processed. Users need to trust that their data is handled properly and do not want to fear whether the information will be breached or sold to the highest bidder. I like your example about how a customer may have multiple order entities but not multiple customers per order. I cannot imagine the chaos a business would undergo if the same order number were assigned to multiple customers.

Cindy, you did a very good job explaining the primary purpose of creating a data model for a database. I am glad you included that a structured approach helps minimize redundancy. This is true for many reasons. It helps developers avoid spending extra time on something they have already completed, especially in a team environment. Other team members do not have to worry about completing the same work as someone else if it already exists in a data structure. I am curious, though, whether you believe it is necessary to construct a data model for a database. I do not think it is necessary, but the benefits far outweigh the cons, so even though it is not a requirement, it is still best to outline a program first.