**Data Modeling**

1. Explain the primary purpose for constructing a data model for a database. Is a data model always necessary? Why or why not?
   1. The main reason for creating a data model is to give structure to the data and show how the parts are related prior to building the database. This helps avoid confusion later when the actual database has been developed. This is why I believe a data model should be developed prior to the creation of the database.
2. What are business rules? And how do they relate to creating data models? Provide an example.
   1. Business rules are the structural requirements for developing the database, tables, and components. The business rules are essential for creating data models because this allow you to choose which system or database you can use.
      1. For example, if the requestor or organization requires the database to be built with embedded related data, you can choose MongoDB.
3. What is the difference between an Entity Relationship Diagram and an Object Relationship Diagram? Provide an example of when you might use each.
   1. ERD will show the relationships between each table in the database. This is the traditional database architecture that explains the table relationships between each column, foreign or primary keys, and other objects within the database.
      1. I would use ERD for developing the design of the database, tables, views, stored procedures within the database. This gives everyone a visual representation of the database architecture and relationships
   2. ORD is focused on the object relationship such as person inheritance of customers, employees, and staff. This allows you to view how the objects relate to each other for storing data.
      1. I would use ORD for developing the visualization of classes, interfaces, and application components on how they interact or relate.
4. What is cardinality in data modeling? How many types of cardinality depiction are there? Which would you prefer and why?
   1. Cardinality shows how many instances of one entity can be related to another. There are three main types: one-to-one, one-to-many, and many-to-many. I prefer one-to-many because it’s the most common and easy to understand, like one customer having many orders.