***Basic Comparison of Relational vs. NoSQL Databases***

**Assignment 1.3 | Brennan Cheatwood**

In relational databases, relationships connect tables using keys like primary and foreign keys. A one-to-one relationship links one record in a table to only one record in another, like a student ID tied to a single student profile. A one-to-many relationship links one record to multiple records, like an author in an "Authors" table connected to multiple books in a "Books" table.

Relational databases, like MySQL, are great with structured data management and ensure data integrity through ACID compliance, which guarantees accurate transactions. However, their rigid architecture makes adapting to changes difficult, and scaling them horizontally can be complex and costly.

NoSQL databases, like MongoDB, are more flexible and handle unstructured data, such as JSON documents. They scale horizontally quickly, making them ideal for large datasets or real-time applications like social media platforms. However, they may sacrifice data consistency for speed, and their lack of standardization can complicate switching between systems.

MySQL offers features like ACID compliance for data integrity and stored procedures for automating tasks like report generation. MongoDB uses a document-oriented model for flexible data storage and supports horizontal scaling through sharding, which distributes data across servers to handle large datasets efficiently.

Relational databases are better for structured data and applications requiring strong data integrity, while NoSQL databases are better for flexibility and scalability with unstructured data.

**References**:

Amazon Web Services (AWS). (n.d.). What's the difference between MongoDB and MySQL? Retrieved from https://aws.amazon.com/compare/the-difference-between-mongodb-vs-mysql/

MongoDB, Inc. (n.d.). MongoDB vs. MySQL: A comparative guide. Retrieved from https://www.mongodb.com/resources/compare/mongodb-mysql