Robert Minkler

CSD 310 – Assignment 1.3

Basic Comparison of Relational vs. NoSQL Databases

August 16, 2024

1. ***In the context of relational databases, what are relationships? Describe at least two, and provide an example of their use.***

Relationships are a way to connect related data in a relational database. Relationships can be defined as one-to-one, one-to-many, or many-to-one (*Data relationships in mongodb)*.

An example of a one-to-one relationship in a database that tracks people might be a current spousal relationship. Bob is married to Ginger. There is only room for one current spouse. Instead of storing Bob and Ginger together, they are each stored as individual entities but have a one-to-one relationship pointer that points at one another.

An example of a one-to-many relationship might be in an employee database linking employees to their primary workplace. A single office building can have many different people who have their primary office space in that building. We can link every associate who works in Building A with a one-to-many relationship.

A student and course relationship was an example of a many-to-many relationship discussed in the course materials. A student can be enrolled in many courses at once, and a course can also have many students enrolled in it at once. This can be tracked with a many-to-many relationship.

1. ***What are the advantages of relational databases? What are the advantages of NoSQL databases?***

A relational database stores data in a structured manner within tables of rows and columns. They can be ACID-compliant, ensuring the data's reliability, accuracy, and consistency. This makes them very good at transaction management, where inconsistencies across nodes could be problematic for the reliable recording of transactions (*NoSQL vs SQL databases*).

NoSQL databases excel at storing unstructured or semi-structured data. NoSQL databases can store data in “JSON documents, key-value pairing, family grouping,” or “graph nodes/edges (*NoSQL vs SQL databases*).” They can also scale out across servers and locations in ways that a traditional relational database cannot. The flexibility of NoSQL databases makes them ideal for Agile teams and when a scale-out architecture is preferable (*What is nosql?).*

1. ***What are the disadvantages of relational databases? What are the disadvantages of NoSQL databases?***

Relational databases suffer from the reality that they store their data in “rigid, predefined schemas (*NoSQL vs SQL databases*).” Everything must be stored in a table, so more complex data requires additional tables and foreign keys to point to the data. For example, a person may have more than one address. Storing multiple addresses in a single table with the person’s other information would be challenging. A new table with addresses must be made with foreign keys pointing to the other table. This makes designing, reading, and writing to the database more challenging. Reading is slower as multiple calls to the database are required to retrieve and join the data. Scaling relational databases across multiple machines is also more challenging with a relational database.

Regarding NoSQL, “the lack of structure can make data untrustworthy and difficult to organize (Foote, 2022).” There is also a lack of standards for database schemas, and each unique system has its own advantages and disadvantages (Foote, 2022). NoSQL databases also lack ACID compliance, making them unsuitable for certain jobs. NoSQL databases also lack joins to join and organize complex data (Foote, 2022). Some of these limitations can be overcome by specialized implementations of NoSQL, such as MongoDB Atlas, which is ACID compliant.

1. ***Identify at least two features of MySQL and two features of MongoDB, and describe what they are and how they are used.***

MySQL:

In MySQL JOIN, CROSS JOIN, and INNER JOIN are equivalent commands. JOIN combines rows from two or more related tables and returns the result. It can combine a customer name with invoices or perform similar tasks with data stored across multiple tables.

REPLACE is similar to insert by inserting a row of data into the MySQL database. It first checks if an entry contains the matching primary key. If a matching primary key is located, that row is deleted before the new row is inserted in its place. Because of how REPLACE is implemented, it can be used to insert or replace a row in a table.

MongoDB:

db.collection.insertOne() inserts a single document into the collection and returns the object id on success.

db.collection.remove() will delete all documents that match the supplied filter from the collection or just one document. This can be used to purge several documents from the collection rather than deleting the documents one by one.

*NoSQL vs SQL databases*. MongoDB. (n.d.). <https://mongodb.com/nosql-explained/nosql-vs-sql>

*What is nosql? NoSQL databases explained*. MongoDB. (n.d.-b). <https://www.mongodb.com/resources/basics/databases/nosql-explained>

*Data relationships in mongodb*. Studytonight.com. (n.d.). <https://www.studytonight.com/mongodb/relationships-in-mongodb>

Foote, K. D. (2022, November 17). *NoSQL databases: Advantages and disadvantages*. DATAVERSITY. <https://www.dataversity.net/nosql-databases-advantages-and-disadvantages/>