Jacob Hayes

2023 May 22

Relationships refer to the nature of linked information in databases. Broadly, these relationships fall into one of three categories – one to one, one to many, and many to many. Additionally, when linking two tables together, one is the primary table and one is its child table. In a one-to-one relationship, each record in the primary table relates to exactly one record in the child table, and vice-versa (since this relationship is symmetrical, either table could be considered the primary).

However, in a one-to-many relationship, a record in the primary table can relate to *zero or more* records in the child table, while each record in the child table relates to exactly one record in the primary table. (An example of this kind of relationship would be customer addresses and online orders – a customer COULD have only one order, they could have many, they could even have zero if they signed up for a mailing list but didn’t order anything yet; but an order will relate to a single customer.)

Relational databases are useful for preventing duplicate information, which saves time and storage costs. It also allows for the table to be normalized, which ensures organization and readability, which is always useful. However, non-relational databases have the advantage of being human-readable, since they’re essentially JSON. Additionally, non-relational databases can be scaled out horizontally, with multiple machines having databases held in sync. This contrasts with one of the major disadvantages of relational databases – scaling a relational database up requires a lot more processing to return queries. Non-relational databases also have high availability, due to the large amount of redundant information; however, this also means that they tend to be larger and less space-efficient than relational bases.

Continuing from that, MySQL is a relational database system, while MongoDB is nonrelational. MongoDB allows for sharding databases, which increases availability and read/write speed; it also has an API specifically for inserting many records at once, as opposed to row-by-row insertion. MySQL allows for more rigid data structures to be easily adhered to; it also provides the benefits of more tightly connecting records by shared data, due to the nature of relationship-oriented database management.

Reference

DePaul University. (n.d.). Database Relationships. Database relationships. <https://condor.depaul.edu/gandrus/240IT/accesspages/relationships.htm>

MongoDB. (n.d.-a). Comparing the differences - mongodb vs mysql. MongoDB. https://www.mongodb.com/compare/mongodb-mysql

MongoDB. (n.d.-b). Relational vs. non-relational databases. MongoDB. <https://www.mongodb.com/compare/relational-vs-non-relational-databases>