Author: Tianze Shan

Denormalization

Definition:

Denormalization is a database optimization technique in which we add redundant data to one or more tables. This can help us avoid costly joins in a relational database. Note that denormalization does not mean not doing normalization. It is an optimization technique that is applied after doing normalization.

Example:

Course_ID	Teacher_ID
cs101	1
cs102	2
cs103	3

Teacher_ID	Name
1	Tom
2	Jerry
3	Bob

In a normalized database, we might have a Courses table and a Teachers table. Each entry in Courses would store the Teacher_ID for a Course but not the teacherName. When we need to retrieve a list of all Courses with the Teacher's name, we would do a join between these two tables.
In some ways, this is great; if a teacher changes his or her name, we only have to update the name in one place.
The drawback is that if tables are large, we may spend an unnecessarily long time doing joins on tables.

Denormalization vs. Normalization

	Denormalization	Normalization
Pros	● Faster data reads ● Simpler queries writing ● Requires less compute on reading operations ● Makes data available quickly	● Faster writes ● No redundant data ● Less database complexity ● Data always consistent
Cons	● Slower writes ● More database complexity ● Potential for data inconsistency ● Requires more storage	● Slower reads ● Heavy querying ● Table joins required ● Indexing not as efficient due to table joins

Denormalization in data warehousing and NoSQL databases

Denormalization often plays an important role in relational data warehouses. Because data warehouses contain massive data sets and may host many concurrent connections, optimizing read performance and minimizing expensive join operations is important.

Document data stores databases often underlie content management systems for web profile pages that benefit from read optimizations. The goal of denormalization in this context is to reduce the amount of time needed to assemble pages that use data from different sources. In such cases, maintaining data consistency becomes the job of the application and, in turn, the application developer.

Columnar databases such as Apache Cassandra also benefit greatly from denormalized views, as they can use high compression to offset higher disk usage and are designed for high read access.

Bibliography

"Denormalization in Databases." GeeksforGeeks, 7 July 2021, www.geeksforgeeks.org/denormalization-in-databases/.

Wright, Gavin, and Jack Vaughan. "What Is Denormalization and How Does It Work?" SearchDataManagement, TechTarget, 6 May 2021, searchdatamanagement.techtarget.com/definition/denormalization.