## Database Assignment 3

**Name:** Muhammad Shayaan e Subah

**Section:** A

**Registration number:** FA22-BSE-065

## **Definition of Denormalization:** Denormalization is the process of adding precomputed redundant data to an otherwise normalized relational database to improve read performance.

Database optimization is an essential step to improve website performance. Typically, developers normalize a relational database, meaning they restructure it to reduce data redundancy and enhance data integrity. However, sometimes normalizing a database isn’t enough, so to improve database performance even further developers go the other way around and resort to database denormalization.

## **When to de-normalize a database:**

Before talking about denormalization, we need to discuss about normalization first.

Normalization is the process of organizing a database to reduce redundancy and ensure data integrity. This means putting each piece of data in its proper place. When a database is normalized, updating data is fast because each piece of data is stored only once. However, retrieving data can be slower because queries need to gather information from many different tables.

In many modern applications, fast data retrieval is crucial. This is where denormalization comes in. Denormalization is the opposite of normalization. Instead of organizing data to avoid redundancy, denormalization involves deliberately putting the same data in multiple places, increasing redundancy.

To conclude, the main reason is to make data retrieval much faster. However, denormalization is not a one-size-fits-all solution. It should only be used for specific reasons and purposes.

## **Pros and Cons of Denormalization**

**Pros**

1. Improved Performance:

Denormalization can lead to significant improvements in database query performance. The database can quickly process read operations by reducing the number of table joins required to retrieve data.

1. Simplified Data Model:

Denormalization can simplify the data model by reducing the database's number of tables and relationships. This can make the database easier to understand and maintain.

1. Reduced Complexity:

With fewer tables and relationships to manage, database administrators can spend less time on maintenance and more time optimizing performance.

1. Enhanced Reporting and Analysis:

Efficient Data Access: Denormalization can improve the efficiency of reporting and analytical queries by reducing the need for complex joins. This is especially useful in data warehousing where quick access to summarized data is important.

**Cons**

1. Increased Data Redundancy:

Denormalization introduces redundancy into the database, which can lead to data inconsistencies and higher storage costs.

1. Reduced Flexibility:

Denormalization can make it more difficult to change the database schema in the future. Because data is duplicated across multiple tables, changing a single field may require updating multiple tables.

1. Higher Maintenance Costs:

Denormalization can increase the database's complexity, making it more difficult to maintain and troubleshoot.

1. Increased Risk of Anomalies:

The likelihood of encountering anomalies during data operations is higher in a denormalized database. Update, insert, and delete anomalies can occur more frequently, necessitating careful management and additional checks to maintain data integrity.

## **References:**

1. RubyGarage Blog: "Database Denormalization with Examples." RubyGarage. Accessed on May 29, 2024, from <https://rubygarage.org/blog/database-denormalization-with-examples>.

2. TechTarget: "Denormalization." TechTarget. Accessed on May 29, 2024, from <https://www.techtarget.com/searchdatamanagement/definition/denormalization>.

3. InvGate Blog: "Denormalization in Databases." InvGate Blog. Accessed on May 29, 2024, from <https://blog.invgate.com/denormalization-in-databases>.