**Discussion 2: Design Relational Database**

Suppose you are designing a database to track your exercise sessions and progress toward your fitness goals. At a minimum, you will need to record the details of each session and be able to report on your activity over a specific range of dates. You'll probably want the database to supply information for charts to show your progress and the increasing intensity of your workouts.

Write a primary response to the following discussion topic, touching on each of the bulleted questions:

* Assuming you want to record changes in your weight, would you do this in a separate table or would you record it for each workout?
* How would you design the tables to record separate activities where those like running might be based on time or distance and others such as weight training based on repetitions?
* Would you record the number of calories burned during each activity in the tables or just record enough information so that the software can calculate it later? (This figure can be calculated based on factors such as the activity and your current weight. The complexity of the calculation as factors change over time might cause whatever program was using the data to take longer to generate the reports).

Write at least two secondary responses to your classmates. State whether you agree or disagree with their approach and why discussing the pros and cons. For this question, please keep in mind the issue of normalization and how that would influence your design.

**Designing a Relational Database:**

* To implement restrictions, appropriate relationships and different criteria must be defined between several databases' entities and the relationships.
* The referential integrity of the data must be preserved.
* To easily grasp the database schema and workflow, an ER diagram must be created.

We'll have to establish a dataset for keeping track of fitness metrics within that session. This is indeed a challenging process since several pragmatic considerations must be taken into account, and it is hard to create an effective and non-redundant database.

Suppose let’s keep track of our weight fluctuations throughout the exercise in a different table. Every after session, users can collect substantial weight in a comparatively straightforward format. There seem to be additional entries of periods and private details in the database that stores this stuff.

To keep track of the exercise information, let’s create a different table called “Exercise\_Activity”. Every activity in this table should be labeled as being used as the table's main thing. Several fields are listed in the table to gather exercise information in a variety of methods, such as Exercise\_Activity\_name, Periodicity, Teams, Number\_of\_Hours, Kcal\_Count, Speed, Distance\_Covered, etc.

I think we should keep a record about how many KCals that users burned every after a workout. So, let’s add a column called “Kcal\_Count” to the database where we need to keep track of exercise specifics.

Here’s is the ERD that I created in the MySQL Workbench CE: This includes the relationships between the 3 tables mainly personal\_details, date, and exercise\_activity. Each table has several attributes.

**Discussion 3: Is the RDBMS still dominant?**

The relational model for databases with its structured query language (SQL) has been the dominant model for database systems for many years.

* Why do you think this is so?
* What are the advantages of the relational model?

For many years, the relational database model, with its structured query language (SQL), has been the main database system model. Why? Because data in the relational model is organized into tables, which have rows and columns. To add to the complexity, all relational databases may be used to manage transaction-oriented applications (OLTP), and most non-relational databases in the Document stores and Column stores categories can also be used for OLTP. OLTP databases are operational databases, with frequent, short transactions that include updates and touch a limited amount of data, and where the concurrency of thousands of operations. Because data integrity is critical, they support ACID transactions (Atomicity, Consistency, Isolation, Durability).

* Splitting data into several connected tables in a relational database model has numerous advantages over a flat-file database
* Each row comprises a single record made up of individual data items or attributes that are structured into columns holding elements of the same kind according to the column's rules
* Microsoft SQL Server, Oracle Database, MySQL, and IBM DB2 are the most popular

The advantages of the relational model are:

* The relational model structures data will avoid complexity and it is very simple
* It is very easy to retrieve the Data
* It has Data integrity which is a very essential feature
* It is naturally scalable and extensible and is flexible
* It is free and the impact of the integrity and accuracy of the database is dependable
* It avoids data duplication
* It avoids inconsistent records
* It is Easier to maintain security

**References:**

[1] Mishal Room. (April 2021). 6 Advantages and Disadvantages of Relational Database | Limitations & Benefits of Relational Database, Retrieved from https://www.hitechwhizz.com/2021/04/6-advantages-and-disadvantages-limitations-benefits-of-relational-database.html

**Discussion 4: How structured is your data?**

* There are structured data as we encounter in the numeric fields of a database.
* There are semi-structured data and unstructured data, as we encounter in text files.
* What are your views about structured, semi-structured, and unstructured data?
* Thinking about the analytical work that you have done in the past or are doing now, how much of your time is spent putting data into a structured form (data preparation)?

**Structured Data:**

It has aspects that can be addressed for effective evaluation. This has been structured into a database, which is a structured store. It refers to all data that can be put in a table with rows and columns in a SQL database. The difficulty with maintaining structured data is that it must be precisely described using data field and type. It needs to be supplemented with additional information.

**Semi-structured Data:**

It is the data that is not stored in a relational database but has organizational qualities that make it easier to examine. Some procedures can be stored in a relation database (though this may be difficult for semi-structured data), but semi-structured data exists to save storage. Data in XML format is an illustration.

**Unstructured Data:**

It is data that isn't arranged in a pre-defined way or doesn't have a pre-defined database schema, making it unsuitable for a relational database system. So, there are alternate tools for managing and storing unstructured data. Text and Media logs are just a few examples. Anything else is considered unstructured data. Also, existing data mining tools frequently miss critical info, making unstructured data analysis time-consuming and costly.

**Data Preparation:**

Although data preparation procedures have become more automated in my recent work at Esri, they can still take a long time. Particularly as the amount of data employed in analyses grows. Rather than examining data, I frequently spend the majority of my time locating and purifying it. Normally, users can conduct simple content searches over unstructured textual data. Even though unstructured data analytics solutions are available, no single vendor or toolset stands out as the clear winner. Alteryx, Power BI, MS SQL Server, Power Query, SQL, and Python were the tools I used.

**References:**

[1] Michael Gramlich. (Sep 9, 2020). What are structured, semi-structured, and unstructured data? *Michael Gramlich*. Retrieved from https://www.michael-gramlich.com/what-is-structured-semi-structured-and-unstructured-data/

**Discussion 4: Real-world problem and data transformation**

* Find a real-world article or newspaper of your interest regarding storing, organizing, managing data or information.
* Thinking about the possibility for data to become Big Data, how would you propose to organize and store them?
* What tools, platforms would you choose and why?
* Describe potential problems and risks you might experience in the short and long run.

**Storing, Organizing, Managing Data or Information:**

Let us take an example from the article cited on the Big Data in Healthcare. It talks about data management, analysis, and prospects. For effective processing, the data must be saved in a format file that is readily available and legible. We can better organize ourselves to offer the greatest results if we have more information. As a result, data collecting is an essential component of any company. We may also utilize this information to forecast present trends in specific metrics as well as future events.

**Organize and Store Big Data:**

Healthcare is a multi-faceted system created with the primary purpose of preventing, diagnosing, and treating human health-related diseases or impairments. Huge amounts of data produced from diverse sources are stored in data warehouses. To gain wider and more affordable healthcare alternatives, this information is analyzed utilizing analytic workflows.

**Tools & Platforms:**

To deploy ML and AI methodologies for big data analysis on computing clusters, advanced algorithms are necessary. Such algorithms or software could be written in a programming language suitable for working with huge data (e.g., Python, R, or other languages). Hadoop and Apache Spark are two of the most popular technologies for working with large data. HDFS (Hadoop Distributed File System) is a file system component that allows for scalable, efficient, and replica-based data storage across several nodes in a cluster. IBM Watson is an analytical software for ML-related applications.

**Problems & Risks:**

It would be necessary to have full, correct, and up-to-date metadata for every stored data to have an effective data governance strategy. For scholars and data scientists, metadata would include information such as the date of establishment, the goal, and individual accountable for the data, and past utilization (by who, why, how, and when).

**References:**

[1] Dash, S., Shakyawar, S.K., Sharma, M. et al. Big data in healthcare: management, analysis, and prospects. *J Big Data*6, 54 (2019). https://doi.org/10.1186/s40537-019-0217-0