Research on Fully Qualified Domains (FQDs), Fully Qualified Table Names (FQTNs), and Taxonomies

Fully qualified Domains: It’s a method to manage the data and combine them to make reusable components. Domain integrity refers to constraints on field attributes in a data table, typically indicating data validity. It includes constraints such as the field's value range, data type, and validity rules, and is determined by the field attributes defined when the relation structure is established. Restrictions include data type, default values, rules, constraints, and whether values can be nullable. Domain integrity ensures that invalid values are not entered. So it makes the data from "What is it" to "What does it represent".

To help understand: Its like you making a new Library with defined conditions like

Book1:500+Pages, image. Other conditions….
Book2:10pages. other conditions

So when you are trying to find a book, you don't have to give exact information. But instead you just say the name of the data.

Fully Qualified Table Names: It is the exact address of the data. We know there are many tables in one database, so it will be hard to find specific data in a specific table by just calling the name of it. So we need the exact address of this data.

To help understand: let's say we want to find John. There might be 100+ John in the entire database. If you just write John, this John can exist anywhere. So we need to know the exact

address of this John. [Server Name].[Database Name].[Schema Name].[Table Name].

Use cases for fully qualified names: If it's only used internally, then fully qualified names are basically not used. FQTN are only used when multiple tables involve the same column name. For example, if a user table has a column named "name" and a product table has a field named "name," and a query uses only "name," the database dont know whether to use the user's "name" or the product's "name." This is when the fully qualified table name should be use. The purpose of using fully qualified names is to prevent ambiguity.

Taxonomies: It categorizes the data and organizes the relationship between data. So you only have 1 type of data in a category.

To help understand: Games and Utility Software are both software, but we know they are different. So we put both of them in the Software category, but what they contain is different. Game category might have Half life, CS, and Utility software might have word, excel in it.
So with taxonomies, we can have data managed and have more efficient ways to find it.

1. Review how T-SQL uses the `CREATE TYPE` statement and how it functions similarly to the ANSI SQL `CREATE DOMAIN` standard.

   From what I learned, Create type and Create domain are similar because they both help to create reusable custom

types and design a more consistent database. Which is the same idea as the FQD.

2. Analyze the **benefits of implementing Fully Qualified Domains within a Taxonomy**, focusing on data consistency, reusability, and governance.

   As we know, FQD is to help reusable types. Taxonomie make categories. With those 2 methods, we can make all the data values to be valid,consistent and reusable in different tables.

3. Discuss the **importance of Fully Qualified Table Names** in maintaining clarity, referential integrity, and scalability across heterogeneous database environments.

   FQTN is the most safe way to access specific data in multiple databases. Because it gives all the information to find the specific data.

   FQD, FQTN, Taxonomies all help user to create a database thats clear and easily to modify

My contribution

Give ideas on understanding FQD,FQTN, and taxonomies. Connect the idea to help understand the usage of those topics.