

Fully Qualified Domains:

A domain is a specific data type that may include certain constraints. Essentially, it acts as a reusable or 'global' constraint. For instance, when multiple tables require the same validation rules, such as for phone numbers, instead of defining those rules repeatedly, we can create a single domain named `PhoneNumber`, assign it the necessary constraints, and apply it wherever needed. This ensures consistency and reduces the chance of errors. A fully qualified domain is a domain with a clearly unique name that reflects its position within a taxonomy or hierarchy. This clarity helps prevent confusion when different departments or systems use domains with similar names.

Example:

Not qualified domain: `Email`

Qualified domains: `Nissan.Employees.Email`, `Nissan.Customers.Email`

Fully Qualified Table Names:

In many database systems, objects with identical names can exist in different locations. For example, there may be two separate `Employees` tables, one in the Sales database and another in the Admin database. Referring to just `Employees` isn't specific enough; a complete reference is required, such as:

`SalesDB.dbo.Employees`

`AdminDB.dbo.Employees`

This structure establishes a parent-child hierarchy that clearly identifies where each table resides. Just as a fully qualified domain specifies the meaning of data, a fully qualified table name precisely indicates where the data is stored.

Taxonomies:

A data taxonomy provides a structured way to classify and organize data. It establishes a hierarchy of categories and subcategories, ensuring that data remains organized and easy to locate. As databases expand, they can become complex and difficult to navigate, leaving users uncertain about what information exists. By developing a clear taxonomy, these challenges are minimized. One can think of it like a tree, where the parent branches represent broad categories, and the leaves or nodes represent the more specific subcategories.