Individual Notes - Christopher Vidal

Course: CSCI 331

Group: #2

Project: Fully Qualified Domains (FQD’s) Fully Qualified Table Names (FQTN’s), and Taxonomies

The CREATE TYPE is used to define user defined data types or table types.

They act as a wrapper around an existing built in type.

Ex: CREATE TYPE udt.Name FROM NVARCHAR(25) NOT NULL;

When this is done the sql server internally treats it as NVARCHAR(25), but the meaning and size are centralized and reusable.

CREATE TYPE *VS* ANSI SQL CREATE DOMAIN

. The CREATE TYPE creates a reusable alias for an existing type, while CREATE DOMAIN creates a constrained data type.

. The only constraint allowed for CREATE TYPE is NOT NULL while CREATE DOMAIN can use more, for instance DEFAULT, CHECK, COLLATE,etc.

. The CREATE TYPE has a data scope while the CREATE DOMAIN has a schema scope.

Syntax:

CREATE TYPE udt.FirstName FROM NVARCHAR(15);

CREATE DOMAIN PHONE AS VARCHAR(15) CHECK (VALUE LIKE ‘+%’);

Why use this?

Consistency: By using this taxonomy every table that uses it will have the same length, format, and rules.

Readability: It's easier to see the purpose of a column and see that its a data type.

Maintainability: It's easier to make changes. If a column definition needs to be changed, you only change it once in the UDT, not in every table.

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

Fully Qualified Table Name(FQTN) identifies a database object with all of its hierarchical components. This becomes essential when working across multiple databases.

Maintain Clarity: Using full qualified names makes the code explicit about where the data lives, avoiding ambiguity, enhancing maintainability, and improving debugging.

Referential Integrity: Using fully qualified names ensures that constraints refer to the correct parent tables. Even if tables with the same names exist in other databases. It also helps enforce foreign key relationships unambiguously. Finally, minimizes accidental cross-linking between unrelated datasets.

Scalability Across Heterogeneous Environments: Fully qualified names support scalability and interoperability by allowing distributed queries, supporting migration and modular design, and facilitating data governance.