

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

Presented By Group#2

Zarrin, Shiwlee, Azmain, Eusan, Haiyan, Christoper and Izaz



*Understanding
Database
Naming and
Structure
Concepts*

- This project explores three key concepts in relational database design:
- **Fully Qualified Domains (FQDs)**
- **Fully Qualified Table Names (FQTNs)**
- **Taxonomies**
- These concepts help organize, validate, and govern data in large systems.
- A **taxonomy-driven design** ensures consistency, reduces errors, and improves collaboration across database environments.

*Why These
Concepts
Matter*

Clear naming rules
ensure **accuracy and governance**.

Standardization reduces confusion
and duplication.

FQDs + FQTNs maintain data
integrity across systems.

These practices are critical in large
enterprise databases.

Defining Reusable Data Rules

In T-SQL, the CREATE TYPE command defines a reusable data type such as “CREATE TYPE EmailAddress FROM VARCHAR(255) NOT NULL.”

In ANSI SQL, the equivalent is “CREATE DOMAIN EmailAddress AS VARCHAR(255) NOT NULL.”

Both approaches enforce consistency, reduce redundancy, and make databases easier to maintain.

Why Use FQDs in Databases

- Define global, reusable data definitions.

- **Benefits:**

- Consistent data validation rules
- Simplified maintenance
- Easier compliance & auditing
- Supports taxonomy-based governance
- FQDs make data standards repeatable across schemas.



Example: Reusable Domain for Phone Numbers

A Fully Qualified Domain can standardize phone number formatting across all tables.

Example: “CREATE DOMAIN PhoneNumber AS CHAR(10) CHECK (VALUE NOT LIKE '%[^0-9]%');”

This enforces a uniform 10-digit format, ensuring every schema follows the same validation rule and improving overall data consistency.

Organizing Data Through Taxonomies

A taxonomy is a structured classification system for organizing database objects.

It defines how different components relate to each other in a hierarchy such as Database → Schema → Table → Column → Domain.

This clear organization promotes logical structure, easier navigation, and better understanding of how data connects within a system.

Improving Governance and Scalability

- Enforces standardization across all databases.
- Teams can:
 - Apply uniform naming conventions
 - Maintain metadata consistency
 - Scale systems without confusion
- Aligns design with **data governance policies**.

Clarity and Structure in Table References

A Fully Qualified Table Name shows the complete path to a table, written as DatabaseName.SchemaName.TableName. For example, “SELECT * FROM SalesDB.dbo.Customers;” identifies exactly where the table exists.

Using FQTNs prevents confusion between similar table names in different databases and ensures clear, precise SQL references.

Integrity and Cross-Database Access

- Maintain **referential integrity**.
- Prevent naming conflicts.
- Enable cross-database joins safely.
- Support scalable, cloud-based systems.
- Improve readability of SQL scripts.

Domains vs Tables – Key Differences

Together, they create **clarity**
+ **consistency** from data
definition to storage.

Concept	Defines	Focus Area	Example
FQD	Data Type or Rule	Column level	CREATE DOMAIN Email AS VARCHAR(255)
FQTN	Table Location	Table reference	SalesDB.dbo.Customers

Enterprise Applications

- **Banks** → Use FQDs for standardized account IDs.
- **Hospitals** → Use FQTNs for shared patient data.
- **E-commerce** → Taxonomies for product categorization.
- **Results: Reliable data, fewer errors, easier governance.**

*FQDs &
FQTNs in
Data
Protection*

- Prevent unauthorized access through precise naming.
- Audit trails can track schema-level activities.
- Consistent naming simplifies **role-based permissions**.
- Enhances **data transparency** and compliance.

Applying These Concepts Effectively

- Always use meaningful schema + table names.
- Create reusable domains for validation.
- Document taxonomies and naming conventions.
- Review periodically for accuracy and scalability.
- Follow ANSI SQL standards when possible.

Key Takeaways from Our Research

- CREATE TYPE / CREATE DOMAIN → Reusable data types.
- FQDs → Consistency and data governance.
- FQTNs → Clarity and cross-system integrity.
- Taxonomies → Structured organization and scalability.
- Together, they build **clean, efficient, and reliable** databases.

- These principles make modern databases more consistent, secure, and easier to manage across teams.