

# SQL Server Database Object Naming Standard

Why database object naming standards? Because naming standards:

- Improve communication across projects, systems, and applications and promote consistency of use by representing the organization's vocabulary of terms;
- Help us to identify, document, and manage our database objects;
- Improve the accuracy of searches for a particular object.

The primary audience of this standard is intended to be the administrators of SQL Server and the developers who work with it to build and maintain applications.

Naming conventions are neither inherently right nor wrong. They just provide a standard way of referencing objects in the database.

## **General rules for naming database objects:**

- No spaces;
- 30 characters or less (mainly for potential portability, e.g., Oracle and DB2 limit to 30). The 30-character limit pertains to the lowest-level name, not the name fully qualified by its namespace (see below for more on namespace). This limitation can be overlooked on occasions when a descriptive name requires a few more than 30 characters;
- Use mixed case with initial-capitalization for the descriptive part of a name. Acronyms will normally be capitalized if used in a name, for example, SIC (Standard Industrial Classification) or PPAS (Personal Property Assessment System).;
- A name must be unique. Sounds simple, but...uniqueness is viewed in the context of the name within its namespace, i.e., the scope of the name. A namespace is a set of related objects that cannot have duplicate names. For example, two columns can have the same name only if they are in separate tables (separate namespaces); no two columns in the same table can have the same name;
- A name should be meaningful, descriptive;
- A name should be independent of the hardware or software used, i.e., there should be no references in a name to particular aspects of the hardware or software;
- Reserved words should not be used. Reserved words are keywords that the DBMS employs for its exclusive use;
- A name should not include meaning that could reasonably change over the life of the object.

What is the definition of "name?" A name is a sequence of characters that identify an object to both the DBMS and the user.

We have de facto methods of naming in some places and this proposed standard accommodates the "history" or legacy to a certain extent while also incorporating some best or common practices going forward, and so there may be some apparent inconsistency in syntax from one object to the next. This is especially evident in the use of underscores in some areas and not in others. In some instances accommodation must be made between "grandfathering" and forging ahead with changes.

The components that one who works with SQL Server would commonly encounter are covered by this standard.

## **Database**

Name is based on the organizational unit most associated with the database and subject area of the database:

Organization\_Subject Area {this is the legacy de facto standard}

# SQL Server Database Object Naming Standard

Organization is the data steward, i.e., the department, agency, or office that has primary statutory or business accountability for the data, or the organization that is most commonly associated with the database. The organization is designated by the standard three-character abbreviation as defined in the email system and is capitalized. The subject area describes the basic nature of the content of the database; this is often a term by which it's commonly referred. Examples:

- **ROD\_Phoenix** for the Register of Deeds Phoenix system database.
- **SHF\_DTTS** for the Sheriff's Delinquent Tax Tracking System database.

Note: **JCO** is used as an abbreviation for Johnson County for certain databases of wider scope, e.g., JCO\_RealEstate.

## **Table**

A table is named based on a description of the content of the information in the table. Use whole words where possible (abbreviations should not normally be used, acronyms and common usage excepted); use the proper case for each word. The singular form is used to signify that we're talking about a set of related information. Examples:

- Account
- Employee
- LineItem
- CaseCalendar

**Special case exception:** For look-up tables (reference tables), such as a State table (KS = Kansas), use the prefix "lut", creating lutState in this case. This has the advantages of distinguishing these tables as apart from tables that represent business entities and transactions and sorts them together in table lists. Lookup or reference tables can be defined as those tables that categorize the other data that does describe the organization and its business transactions. The main difference is that lookup tables are not descriptive of things within the enterprise but rather provide a mechanism to validate and verify data values and standardize encoding schemes.

## **View**

Same as table but prefix with "vw", thus vw + description. Examples:

- vwClosedAccount
- vwTaxPayerAudit

## **Application Role**

"approle" + description of the role: Example:

- approlePPASselect
- approleCTTsupervisor

Note: Standard roles are not included since they are only used administratively for controlling rights.

## **Trigger**

"trg" + table name or view name + description + operation upon data (I=insert, U=update, or D=delete).

Examples:

- trgParcelCheckOwnership\_IU
- trgvwCustomerMaintainStatus\_U

# SQL Server Database Object Naming Standard

## User-Defined Datatype

"udt" + description. Examples:

- udtStreetAddress
- udtPersonName

## Index/Key

"PK" + table name or "ix" + table name + column name(s), where PK=primary key and ix=non-primary key index. Use underscore to accommodate legacy usage.

Examples:

- ix\_CustomerLastNameFirstName
- PK\_Account

Note: Legacy usage has capitalized PK (primary key), but not ix (index).

## Check Constraints

"chk" + a description of the nature of the constraint. [Check constraints can apply to multiple columns, and you can apply multiple check constraints to a single column.] Examples:

- chkSalaryRange
- chkTaxAmountLimit

## Foreign Key Constraints/Relationship

"FK" + foreign table name + column name(s). Examples:

FK\_CustomerCustomerID

FK\_AccountAccountID