

VITA Enterprise Application Division (EAD)
Database Design Standards for Microsoft SQL Server

Introduction

This document's purpose is to define Microsoft SQL server database design standards.

The database being developed or changed should be depicted in an ERD (Entity Relationship Diagram). The Entity-relationship model for relational databases should be in third normal form. The model may be de-normalized where necessary for performance.

The preferred naming convention is lowercase separated by underscores. If another convention (camel case, uppercase, etc.) is used, the database design should be consistent. A production database should only contain tables that are being used with the current applications.

General Naming Conventions for Database Objects

1. Begin all names with a letter.
2. Only lowercase letters, 0-9 and _ (underscore) may be used.
3. Do not use SQL Server reserved words by themselves as names, e.g. database, join, primary, begin or between. Refer to corresponding version of SQL Server Books Online for a list of reserved words.
4. A column name should be unique within the table.
5. Names should be as short as possible while remaining meaningful.
6. Use underscores as delimiters.
7. Avoid prepositions where possible (e.g. use vessel_length rather than length_of_vessel).
8. Avoid meaningless or redundant names (e.g. data_tbl for a table name, col_1 for a column name).

Database Objects Naming Standards

Database objects include:

- I. Tables
- II. Column
 - a. Standard Suffixes
 - b. Audit Columns
 - c. Primary Keys
 - d. Foreign Keys
- III. Indexes
- IV. Views
- V. Triggers
- VI. Schemas
- VII. Security
- VIII. Roles

I. Tables

Table names should be short but meaningful. The maximum length is 128 characters for a table name. Use standard abbreviations (see EAD abbreviation

VITA Enterprise Application Division (EAD)
Database Design Standards for Microsoft SQL Server

list- VITADataAbbrevsV1) where possible and use underscores to separate the abbreviations (e.g. mfg_spec instead of manufacturing_specification or mfgspec).

Avoid generic meaningless table names like detail_data, tmp1 or table1. Table names must be unique throughout a database and describe their full significance within the organization.

Table names should follow these rules:

- Table names should be singular
- Table names will use underscores between words
- Table names should not be longer than 128 characters
- Every table should have a primary or unique key
- Table names should not include embedded spaces

II. Column

Columns should honor the following guidelines:

- Column names must be unique within a table
- Columns names should not be longer than 128 characters
- Column names should be meaningful, use standard abbreviations, and use the correct data type
- Always use VARCHAR or NVARCHAR for text fields and specify a maximum length
- Numeric columns should use correct precision, e.g. NUMERIC(14,2)
- The column naming convention should be consistent.
- Column names should not include spaces

a. Standard Suffixes

It is useful to use a standard suffix to identify common types of fields.

_cd	A code that is known and used by a user to identify an object (usually an alphanumeric code), e.g. mfg_cd.
_nbr	A number code that is known and managed by the user, e.g. part_nbr.
_id	A system generated unique number used internally within the database and not usually known by the user. This is usually generated by a sequence. For primary key columns, the column name will be <table name>_id; for foreign key columns, the column name will be <referenced table name>_id.
_ind	A field used as an indicator field to indicate a specific sub-type or role as identified during the analysis phase.

VITA Enterprise Application Division (EAD)
Database Design Standards for Microsoft SQL Server

b. Primary Key

Numeric single-column primary key names should be <table name>_id, e.g. emp_id.

This key should be the table name appended with “id”. The column should be auto generated. The column type should be “INT” or “BIGINT”.

Every table should have a primary key that is auto generated

c. Foreign Key

The foreign keys should be defined to insure data integrity.

Columns that are used to specify foreign keys in a table will normally have the same name as the key in the referenced table.

If there is more than one foreign key relationship between two tables the differentiation between the foreign key columns should be represented by the definition of the column within the table.

d. Audit Columns

Audit columns should exist on every table unless the table is read-only. These columns are as follows:

Logical Name	Physical Name	Data Type	Nullable
Create date	cret_dttm	DATETIME	Not null
Created by	cret_by	VARCHAR(30)	Not null
Update date	updt_dttm	DATETIME	Not null
Updated by	Updt_by	VARCHAR(30)	Not null

III. Indexes

The DBA will need to find the best use of indexes.

- Indexes and relationships should be carefully created. Adjust the number of indexes to minimize performance issues.
- Limit the number of indexes on tables that will have significant insert and update activity.
- Increase the number of indexes on tables where it will improve performance.
- Consider if your usage would benefit from a clustered index.
- Covering indexes can be used, but be aware of the physical space required for the index.

VITA Enterprise Application Division (EAD) Database Design Standards for Microsoft SQL Server

IV. Views

Views can be used to hide the complexity of the underlying database structure, or customize the data for a set of users. Other uses for views include, controlling access to rows and columns of data and/or to aggregate data for performance.

An indexed view can be used to enhance application performance.

- View names should reflect the purpose of the view and suffixed with `_v`
- View names will use underscores between abbreviations
- View names should not be longer than 128 characters
- Views should be limited in the number of columns they return

V. Triggers

Triggers typically have been used for auditing or DRI (Declarative Referential Integrity). With SQL server version 2008 auditing can be handled with the database. A best practice is to use primary and foreign keys for DRI instead of triggers.

- Triggers can be used to implement business rules
- Other uses of triggers are to performing validation or data modifications
- Suffix the trigger with `_instrg` for inserts, `_updtrg` for updates and `_deltrg` for deletes

VI. Schemas

A schema is a distinct namespace to facilitate the separation, management, and ownership of database objects.

- Use schemas to combine related, logical entities into one physical database to reduce administration overhead.
- The schema groups objects within one database into logical groups.
- Within Microsoft SQL 2005 & 2008, the database can not be backed up by schema.
- A database can not be restored by schema.

VII. Security

By default tables should not have PUBLIC access. Permissions on tables should be granted explicitly to users, roles and or schemas in a way that grants as little access as possible while granting everything needed and attempting to limit the number of roles, users and schemas which need to be managed. Additional VITA database security standards can be found at:

http://www.vita.virginia.gov/uploadedFiles/Library/PSGs/IT_Security_Standard_501_01.pdf

VIII. Roles

Application and database roles are used to grant security permissions to a database.

- To easily manage the permissions in your databases, SQL Server provides several *roles* which are security principals that group other principals. They are like **groups** in the Microsoft Windows operating system. Database-level roles are database-wide in their permissions scope.
- There are two types of database-level roles in SQL Server: *fixed database roles* that are predefined in the database and *flexible database roles* that you can create.
- An application role is a database principal that enables an application to run with its own, user-like permissions. You can use application roles to enable access to specific data to only those users who connect through a particular application.