**NAMING CONVENTION FOR DATABASE OBJECTS**

**RMS v1.0.2**

**Author:** Son Nguyen

**Create date:** 11-04-2013

**Change log:**

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Author** | **Date** | **Comment** |
| 1 | Son Nguyen | 11-04-2013 | Create document for version 1.0.0 |
| 2 | Lam Vu | 11-05-2013 | Update |
| 3 | Son Nguyen | 11-06-2013 | Update |

1. **Naming**

**Tables:** Rules

* Group related table names.
* PascalCase notation
* Underscores should not be used. Using PascalCase for table name allows for the upper-case letter to denote the first letter of a new word or name.
* Avoid using abbreviations if possible.
* Junction tables, which handle many to many relationships, should be named by concatenating the names of the tables that have a one to many relationship with the junction table.

**Columns:**

* For fields that are the primary key for a table and uniquely identify each record in the table, the name should simply be [tableName] + “ID“

Example: CustomerID

* Columns with data type (date, or datetime) should generally end in \*Date.

Example: InvoiceDate.

**Stored Procs:** Rules: spa\_[<Action>\_]<table/logical instance>

* Examples: spa\_get\_Invoice

get: get 1 record by criterias

save: Insert or Update

list: Get list of record

delete: Delete a record

* If procedure returns a scalar value, or performs an operation like validation, you should use the verb and noun combination. For example, "ValidateLogin".
* Do not prefix stored procedures with something that will cause the system to think it is a system procedure. For example, in SQL Server, if you start a procedure with "sp\_", "xp\_" or "dt\_" it will cause SQL Server to check the master database for this procedure first, causing a performance hit.

**Functions:** Rules : fcn\_<Purpose>

* Example: fcn \_SplitString

**Views:** Rules: vw\_<Purpose>

* Example: vw\_Invoices
* vw\_Customers

**Variables:** All variables should be written in camelCase and include data type

Example : @intCustomerID, @blnIsEnabled

* Integer: int
* Bit: bln
* Uniqueidentifier: gui
* Varchar, Nvarchar: str
* Decimal: dec
* Numeric: num
* Date/Datetime: dtm

**Parameters:** Rule: @{I/O}\_v<ParameterName>

Where {I/O} is input or output parameter.

* Example : @I\_vCustomerID
* @O\_vErrString

**Triggers:** Rules: TR\_<TableName>\_<action>

* Examples: TR\_Branch\_UpdateBranch
* Notes: The use of triggers is discouraged

**Indexes:** Rules: IX\_<TableName>\_<columns separated by \_ or SpecialPurpose>

* Examples: IX\_Invoice\_InvoiceID
* IX\_Invoice\_MonthlyReport

**Primary Keys:** Rules: PK\_<TableName>

* Examples: PK\_Customer

**Constraints:** Rules: <ConstraintType>\_<TableName>\_<FieldName>

* Example: FK\_Invoice\_CustomerID

PK: Primary Key

FK: Foreign Key

CK: Check

UN: Unique

**General Rules**

* Try to limit the name to 50 characters
* Use a letter as the first character of the name. (don't start names with underscores or numbers)
* Limit the use of abbreviations (can lead to misinterpretation of names)
* Limit the use of acronyms (some acronyms have more than one meaning e.g. "ASP")
* Make the name readable.
* Avoid using spaces in names even if the system allows it.
* Do not use SQL keywords as the name of database objects. In cases where this is necessary, surround the object name with the brackets. E.g : [Year]

**Formatting**

* Use upper case for all SQL keywords
  + SELECT, INSERT, UPDATE, WHERE, AND, OR, LIKE, etc.
* Indent code to improve readability
* Comment code blocks that are not easily understandable
  + Use single-line comment markers(–)
  + Reserve multi-line comments (/\*.. ..\*/) for blocking out sections of code
* Use single quote characters to delimit strings.
  + Nest single quotes to express a single quote or apostrophe within a string
    - For example,SET @sExample = ‘SQL”s Authority’
* Use parentheses to increase readability
  + WHERE (color=’red’ AND (size = 1 OR size = 2))
  + JOIN … ON (CustomerID = 1)
* Use BEGIN..END blocks only when multiple statements are present within a conditional code segment.
* Use one blank line to separate code sections.
* Use spaces so that expressions read like sentences.
  + fillfactor = 25, not fillfactor=25
* Format JOIN operations using indents
* Should use TAB as 2 SPACE characters
* Should always use Alias
* Do not select all by “SELECT \*”, put all columns instead.