T-SQL Naming Convention & Style Guidelines

1. Format all T-SQL statements with appropriate line breaks & indentation for readability. Use single (not double) blank lines to separate logical pieces of T-SQL code, and do so liberally.

```sql
SELECT
 Table1.Column1,
 Table1.Column2,
 Table2.Column1
FROM
 Table1
 JOIN Table2 ON Table1.Column4 = Table2.Column6
WHERE
 Table1.Column7 = 44
ORDER BY
 Table2.Column1;
```

2. Use upper case for all T-SQL constructs, except Types:

```sql
SELECT MAX(MyField) FROM MyTable
```

3. User lower case for all T-SQL Types and usernames:

```sql
DECLARE @MyVariable int
```

4. Use Pascal casing for all UDO's:

```sql
CREATE TABLE dbo.MyTable
(
 MyField int
)
```

5. Avoid abbreviations and single character names

```sql
--Correct
DECLARE @Counter int

--Avoid
DECLARE @C int
```

6. Don't use any special or language dependent characters to name objects. Constraints can use the underscore character.
7. Use the following prefixes when naming objects:
   - FK_ - Foreign keys
   - DF_ - Default constraints
   - IX_ - Indexes
8. Name tables in the singular form:

```sql
--Correct
CREATE TABLE dbo.Address

--Avoid
CREATE TABLE dbo.Addresses
```

9. Tables that map many-to-many relationships should be named by concatenating the names of the tables in question separated by an underscore (_), starting with the most central table's name.
10. Primary and Foreign key fields are postfixed with ID.

```
--Correct
CREATE TABLE dbo.[User]
(
 UserID int NOT NULL,
 AddressID int NOT NULL --Foreign key
)
--Avoid
CREATE TABLE dbo.[User]
(
 UserID int NOT NULL,
 AddressFK int NOT NULL --Fieldname indicates its use as a foreign key
)
```

11. Name Stored Procedures as [schema].[Object][Operation].
    When creating Procedures to wrap single INSERT/UPDATE/DELETE statements, operation should be Insert, Update and Delete respectively. Use Pascal casing.
12. Properly arrange statements: Either use one-liners without indentation or multi-liners with indentation. Don't mix the two.

```
--Correct one-liner

SELECT * FROM dbo.MyTable
--Correct multi-liner
SELECT *
FROM dbo.MyTable
WHERE MyTableID IN
(
 SELECT MyForeignTableID
 FROM dbo.MyForeignTable
)
AND MyColumn > 1

--Avoid
SELECT *
FROM dbo.MyTable --Missing indentation
WHERE MyField > 1 AND --Misplaced AND
 Myfield < 3

--Avoid mixing multiline and singleline expressions
SELECT * FROM dbo.MyTable
WHERE MyField > 1
```

13. When creating local scope always indent:

```
BEGIN
 (...)
END
```

14. When using parentheses around multi-line expressions, always put them on their own lines:

```
--Correct
RETURN
(
 (...)
)

--Avoid
RETURN (
 (...) )
```

15. When using IF statements, always BEGIN new scope:

```
--Correct
IF(1 > 2)
BEGIN
 (...)
END
ELSE
BEGIN
 (...)
END

--Avoid
IF(1 > 2)
 (...)
ELSE
 (...)
```

16. Always create scope when defining Procedures and multi statement Functions:

```
--Correct
CREATE PROCEDURE dbo.uspMyProcedure
AS
BEGIN
 (...)
END

--Avoid
CREATE PROCEDURE dbo.uspMyProcedure
AS
 (...)
```

17. Keep table name aliases short, but as meaningful as possible. In general, use the capital letters of a table name as an alias and use the AS keyword to specify aliases for tables or fields.

```
--Correct
SELECT U.Surname,
 A.Street
FROM dbo.[User] AS U
JOIN dbo.Address AS A ON U.AddressID = A.AddressID

--Avoid
SELECT U.Surname,
 Street --Missing alias
FROM Users U --Missing AS
JOIN dbo.Address ON U.AddressID = dbo.Address.AddressID --Missing Alias
```

18. Avoid joining in the where clause, instead use ANSI syntax for joining. Include the reference key last:

```
--Correct
SELECT U.Surname,
 A.Street
FROM dbo.[User] AS U
JOIN dbo.Address AS A ON A.AddressID = U.AddressID

--Avoid
SELECT U.Surname,
 A.Street
FROM dbo.[User] AS U,
 dbo.Address AS A
WHERE U.AddressID = A.AddressID --Joins in the WHERE clause
```

19. Avoid using RIGHT joins - rewrite to LEFT joins.
20. When doing INNER JOIN's, use the INNER keyword:

```
--Correct
SELECT U.Surname,
 A.Street
FROM dbo.[User] AS U
INNER JOIN dbo.Address AS A ON A.AddressID = U.AddressID

--Avoid
SELECT U.Surname,
 A.Street
FROM dbo.[User] AS U
JOIN dbo.Address AS A ON A.AddressID = U.AddressID
```

21. If you use designers to generate DML - reformat it using the design styles defined here. In effect it is disallowed to check in DML from designers into a project repository.
    Using designers to generate DDL however is allowed and encouraged.
22. Code should be self-documenting. That said, SQL is not always the most readable language. Comments should be used as necessary to explain the code.
23. All stored procedures should have comments describing the functionality of the stored procedure. Comments about author, creation date, etc. are discourage as these are tracked by source control.