SQL formatting standards for NCOI Team

*References:* [*Redgate - tSQL formatting standards*](https://www.simple-talk.com/sql/t-sql-programming/transact-sql-formatting-standards-(coding-styles)/)

[*Apex SQL formating standards*](http://solutioncenter.apexsql.com/sql-formatting-standards-capitalization-indentation-comments-parenthesis/)

**Case (Capitalization)**

* Keywords are uppercase, the data types lowercase, and the object names camel case

Example:

IF OBJECT\_ID(‘ProductDocs’, ‘U’) IS NOT NULL

DROP TABLE ProductDocs

GO

CREATE TABLE ProductDocs

(

  DocID int NOT NULL IDENTITY,

  DocTitle nvarchar(50) NOT NULL,

  DocFileName nvarchar(400) NOT NULL,

  CONSTRAINT PK\_ProductDocs\_DocID PRIMARY KEY CLUSTERED (DocID ASC)

)

GO

# Object References

* Include the schema name *only* if it’s not dbo

SELECT OnlinePaymentID, ReferenceOrderId

FROM Imported.OnlinePayments

* Do not include the DB’s name in the object’s name
* The asterisk (\*) wildcard is not permitted in SELECT lists

SELECT \* FROM ProductDocs

VS.

SELECT DocID,

DocTitle

FROM ProductDocs

* Always specify column names in the INSERT and UPDATE statements, even if they have default values
* Do not use column numbers instead of column names in T-SQL statements

# Aliases

* The keyword AS should always be used for both column aliases and table aliases
* Tables should have meaningful abbreviations instead of single letter aliases

SELECT(cnt.FirstName + ' ' + cnt.LastName) FullName,

   emp.LoginID,

emp.Title

FROM HumanResources.Employee emp

   INNER JOIN Person.Contact cnt

   ON emp.ContactID = cnt.ContactID

ORDER BY cnt.LastName

# Commas

* Commas should be places after each selected column, at the end of row, as each column should be placed on a new line
* In the order by statement, multiple columns should be on the same line with a comma and a space after

SELECT FirstName,

   MiddleName,

   LastName,

   City,

   StateProvinceName

FROM HumanResources.vEmployee

WHERE JobTitle LIKE ‘Production Technician%’

ORDER BY StateProvinceName, City

# Spacing and Aligning

* In the SELECT statement, each column is written on a new line, except the first one
* Main statements start at the beginning of the line, but the rest of the code is indented (like the example above)
* For indentation we use TABS, not spaces
* Each join is indented once and it’s ON clause is indented even more

SELECT sa.StudentID

FROM Sync\_StudentAssignments AS sa

INNER JOIN Sync\_StudentAssignmentInstances AS sai

ON sa.StudentAssignmentID = sai.StudentAssignmentID

AND sa.StudentAssigmentStatusID = 1

INNER JOIN Sync\_Persons AS p

ON sa.StudentID = p.PersonID

# Code Blocks

* BEGIN/END statements should be indented and the inner statements should be indented even further in
* Parenthesis should be opened and closed on new line
* Any subquery should be indented as to offer the possibility of using that query individual. Indented inside brackets, but each bracket on a new line, as mentioned above.

DECLARE @c varchar(10), @d varchar(10)

SET @c = ‘one’

SET @d = ‘two’

IF (@c = @d) OR (@c = ‘two’)

   BEGIN

      PRINT ‘The first condition is correct.’

   END

ELSE

   BEGIN

      PRINT ‘The first condition is incorrect.’

END

SELECT cnt.FirstName, cnt.LastName, emp.SickLeaveHours,

   (

      SELECT AVG(SickLeaveHours)

      FROM HumanResources.Employee

   ) AS AvgSickLeave

FROM HumanResources.Employee emp

   JOIN Person.Contact cnt

   ON emp.ContactID = cnt.ContactID

WHERE emp.EmployeeID = 1

# Comments

* Always used at the beginning of views/SPs/functions

/\*=============================================

Author:Name+SurnameInitial

Create date: dd.mm.yyyy

Description: short description of code

Changes:

NameI : dd.mm.yyyy – issueNumber – description of change

Examples:

RobertC : 26.06.2013 – 112233 - added new column TaxExempted

DanielZ : 09.07.2013 – 223344 - select TaxExempted as SystemName

MirceaR : 18.10.2013 – 334455 - add SendInvoiceViaSettingId and SendInvoiceViaSettingName

=============================================\*/

* Code should be self-explanatory, no need for comments. If there are IDs used, or other information that needs some other explanation, add a comment on it
* Use /\* \*/ comments, for it contains the entire comment so there is no room for mistakes in adding pieces of code after a comment, so that code will be also commented

# Naming Conventions

* Tables – singular/plural?
* Views – suffix List
* SPs – action
* Names must be CamelCase
* No underscores are allowed in object names
* Primary and Foreign key should be explicitly named

# [SET Statements](https://msdn.microsoft.com/en-us/library/ms190356.aspx) use

# SET ANSI\_NULLS or SET QUOTED\_IDENTIFIER use the setting specified at stored procedure creation time. If used inside a stored procedure, any SET setting is ignored.

SET ANSI\_NULLS is ON, a SELECT statement that uses WHERE***column\_name***=**NULL**returns zero rows even if there are null values in***column\_name.*** In a future version of SQL Server, ANSI\_NULLS will always be ON and any applications that explicitly set the option to OFF will generate an error.

When SET QUOTED\_IDENTIFIER is ON, identifiers can be delimited by double quotation marks, and literals must be delimited by single quotation marks

# Keep in Mind/ToDOs while SQL coding

* Create **Primary and Foreign Keys explicit**, in the CREATE TABLE statements. Name them based on the naming convention above.
* **Create indexes when you create the table**. Create **clustered index over the primary key** and non-clustered indexes on other columns that might need it. That’s a decision particular to every table. MOST tables need a **non-clustered index on Deletion\_Date** (all that have this column should have index on it).
* **Use SQLSearch** to find place you might have affected when changing something
* When changing a view/SP **check the performance** impact by including the execution plan at run time. See at least the percentage difference so you do not lower performance.
* When creating a view/SP **try more variants of code** and check them against actual execution plan. See if you can improve performance.