**Naming convention:**

Required:

1. Unique object name within the same database:

Example: you cannot have a table and a view both named “customer\_product”.

1. Since by default SQL is a case-insensitive environment, “customer\_product” and “Customer\_Product” will be evaluated as the same name in the SQL (unless you explicitly ask SQL not to).

Example: you cannot have a table named “customer\_product” and a view named “Customer\_Product”.

1. Don’t exceed 128 characters for table name and column name.
2. Within the limit, be as explicit as possible. (don’t use generic names) (preferred and accepted)

Example:

🗶 prod\_rev

✓ product\_revenue

Suggested:

1. Have a **CONSISTENT** naming convention in your database. camelCasing, underscore-delamination, or a sensible mix of the two are both acceptable, as long as it’s justified by the use case.

Example:

✓ userName (This increase the readability of the code, but = username in the system)

✓ user\_name (my personal preference since it will work for any system/application)

✓ Win10\_KPI\_FY15

Note: **CONSISTENT** is the key: you don’t want userName and device\_name both as column names in the same database.

1. Try everything you can to avoid using space in table name and column name. If you have space in names, you’ll HAVE TO use square brackets every time you refer to them in the script. That means more coding.

Example:

🗶 ms device -> [ms device]

✓ ms\_device

1. Try everything you can to avoid using keyword as table name or column name. Keywords are reserved words in the system and will be highlighted, unless you use [] to quote them. Even this doesn’t necessarily break the code, it still causes unnecessary confusion.

Example:

🗶 datetime -> [datetime]

✓ datetime\_device\_dashboard

1. Use singular format for table and column names.
2. Try not use special characters (#, $, -, ‘, “, /, %...) in names. Many of them will require square brackets around the name. Some of them are even reserved in SQL like “#” (for temp tables) and “@” (for variables).
3. Add prefix to the name of views, functions, stored procedure to help you identify items, however prefixing tables are not generally encouraged unless they are justified by your use case.

Example:

✓ view\_KPIs -> view

✓ sp\_append\_product\_table -> stored procedure

**Style Guide:**

Required:

1. Have to specify schema to refer to tables that are not in the default schema.

Example: [dbo] is the default schema. [device\_archive] is another schema in the db:

[dbo].[all\_up\_report] = [all\_up\_report] -> you can omit [dbo]

[device\_archive].[KPI\_raw] ≠ [KPI\_raw] -> you cannot omit [device\_archive]

Suggested:

1. Capitalize keywords (This is not required, but I always do this in production code).

Example: SELECT, AS, GROUP BY, DATE….

1. Bracket quote schema, table, and column.

Example: [dbo].[ms\_device].[device\_id]

1. Use indentation to help you organize code. White space, new line, “;”, and indentation don’t have gramma meaning, but they can make the script more readable.

Example:



1. The “GO” key word is a batch separator in SQL Server db and Azure SQL db. Use it if you want the script execution to be broken into batches. If you execute SQL command in MySQL or Oracle, it will not be recognized.

**Additional reading:**

Warning: You will find conflicting ideas from the articles. Make your best judgement based on your business case, based on what kind of users your database will serve and what programming languages will be used to access the database.

Naming conventions: <http://www.vertabelo.com/blog/technical-articles/naming-conventions-in-database-modeling>

<http://stackoverflow.com/questions/7662/database-table-and-column-naming-conventions>

<http://www.isbe.net/ILDS/pdf/SQL_server_standards.pdf>

[https://launchbylunch.com/posts/2014/Feb/16/sql-naming-conventi ons/](https://launchbylunch.com/posts/2014/Feb/16/sql-naming-conventi%20ons/)

<http://www.sqlstyle.guide/>

FREE online SQL formatter: <http://www.dpriver.com/pp/sqlformat.htm>

Reserved SQL keyword: <https://msdn.microsoft.com/en-us/library/ms189822.aspx>

j