Exercise 1

Create a stored procedure called "NameSearch" that allows users to search the Person.Person table for a pattern provided by the user.

The user should be able to search by either first name, last name, or middle name.

You can return all columns from the table; that is to say, feel free to user SELECT *.

The stored procedure should take two arguments:

- **@NameToSearch**: The user will be expected to enter either "first", "middle", or "last". This way, they do not have to remember exact column names.
- **@SearchPattern**: The user will provide a text string to search for.

A record should be returned if the specified name (first, middle, or last) includes the specified pattern anywhere within it. I.e., if the user tells us to search the FirstName field for the pattern "ravi", both the names "Ravi" and "Travis" should be returned.

Hints:

- You will probably want to use LIKE with a wildcard in your WHERE clause.
- To include single quotes in your dynamic SQL, try "escaping" them by typing four consecutive single quotes ('''').
- Try creating a variable to hold the actual column name to search, and then set this variable using IF statements, based on the value passed into the "NameToSearch" parameter by the user. Then simply plug this variable into your dynamic SQL. This is easier than having to execute different queries depending on what was passed in.

Exercise 2

Modify your "NameSearch" procedure to accept a third argument - **@MatchType**, with an INT datatype - that specifies the match type:

- 1 means "exact match"
- 2 means "begins with"
- 3 means "ends with"
- 4 means "contains"

Hint: Use a series of IF statements to build out your WHERE clause based on the @MatchType parameter, then append this to the rest of your dynamic SQL before executing.