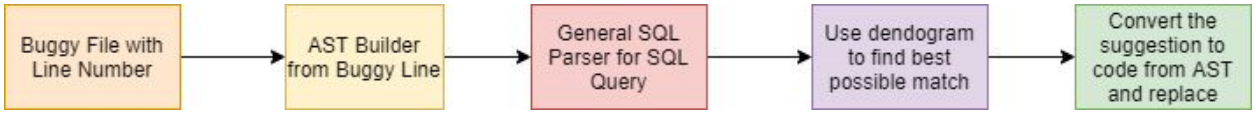
Fix Suggestion

Currently, we are working on building the pipeline of **Getafix**. For this purpose, we built a flow to give suggestions for our buggy code containing probable SQL injection.



We are following these steps:

1. Our code will take a code containing SQL queries.
2. Then, we will create the AST from the buggy line numbers.
3. We will parse the SQL query by GSP(if there any)
4. Use **dendrogram** (Output from the clustering algorithm) to match the AST from step 3. We will use some tree matching algorithm for this step.
5. We have to move back the suggestion to code from AST and replace it to make a compilable code.

We finished up to step 4. We are now working on tree matching algorithms to make a better dendrogram from clustering and give suggestions.

Rule Based Fixer

We are comparing with a rule based fixer from the paper, ***On automated prepared statement generation to remove SQL injection Vulnerabilities by Stephen Thomas, Laurie Williams, Tao Xie.*** We managed to run their code. It works for simple queries. We are going to build a pipeline for this code to compare our Getafix model with them.

Comparison Between Two Approach

1. Rule Based Fixer can fix 14 codes from 61 codes(23%) where our Getafix can fix 31 codes from 61 codes(51%).
2. Rule Based Fixer does not need a previous example or context for fixing where Getafix needs a dataset for learning for giving fix suggestions.
3. Rule Based Fixer adds extra function and loops for fixing.
4. Currently we can not work with queries that are formed using string concatenation in multiple lines. This is our only drawback.