Limitation of PSR Algo(Rule Based Technique)

1. The method and Class which contains the SQL Query should have an access controller such as Public, Private, Protected.
2. If there is any method with such access controllers i.e. Public, Private, Protected before the original method which contains SQL query, PSR Algo can not handle the fixing.
3. We have to make sure there is no other variable that has the same name as this algo generated variable.
4. The PSR-Algorithm uses the objects involved in the SQL to create the prepared statement. The required objects include the execution method, the string objects containing the SQL statement, and the *Connection* or *Statement* object. The PSR-Algorithm can not work without these objects.
5. Can not handle if there are more than one execution of queries in a single function.
6. Can not handle embedded value in field type query. Such as:
   *String sql = "SELECT * FROM tblAchievements WHERE fldStatus='Active'";*
7. Can not handle multi-line and multipart String. Such as:
   String sqlQuery =
   *"Select * from User where email=" +*
   forgotObj.getEmail() +
   "and userName=" +
   forgotObj.getUserName();

Previously, we had some bugs running the PSR Algo.
After carefully running the code again: We got **24 compilable solutions** out of 61 codes. Our methodology finds **32 solutions**.

For the limitation no 5, PSR Algo can not handle second order SQL injection for this flow:
Fetch data from the database(which may contain the vulnerability) and use this for another query within the same function.

PSR Algo can handle Lambda Function and Anonymous inner class functions. Previously, we had some errors on our part to run their code. So, we thought that they could not handle it. We are sorry for our mistake.

Update on Extension to PHP

We can successfully run the **php parser** and **php differencer**. Now, we have to work to build a dataset and integrate it with our pipeline.