# SQL Generator and Validation Overview

## sql\_generator.py

The sql\_generator.py script is designed to generate SQL queries from a given YAML configuration file. It performs the following functions:  
  
1. Loading YAML Data: The script uses a custom YAML loader to read the input YAML file, which contains the necessary configurations for generating SQL queries.  
  
2. Validation: Before generating the SQL, it validates the structure and data types of the YAML input using a set of predefined validation rules. This ensures that the SQL generated is based on valid and complete information.  
  
3. SQL Query Construction: It constructs the SQL query based on the provided parameters in the YAML file. This includes select columns, join conditions, where clauses, group by clauses, having clauses, order by clauses, and union queries.  
  
4. SQL Formatting: After generating the SQL query, it formats the SQL for better readability using the sqlparse library.  
  
5. SQL Validation: The generated SQL is validated using the sqlglot library to ensure correctness before writing it to a file.  
  
6. Output: Finally, the formatted SQL query is saved to a specified output directory in a .sql file, and logging is used throughout the process to capture any errors or important information.

## validation.py

The validation.py script is responsible for ensuring the correctness of the input YAML structure used for generating SQL queries. It performs the following functions:  
  
1. Custom YAML Loader: It defines a custom YAML loader (LineLoader) that tracks line numbers for each field in the YAML input. This is useful for logging errors with precise line references.  
  
2. String Validation: The script includes a function to validate string fields, ensuring they are not empty and of the correct data type.  
  
3. List Validation: A function checks if a field is a non-empty list and validates each element within the list, allowing for both strings and dictionaries.  
  
4. Join Condition Validation: It validates join conditions to ensure each join entry is a dictionary and contains the necessary fields (join\_type, join\_table, join\_condition) when applicable.  
  
5. Union Validation: Similar validation checks are applied for unions, ensuring that required fields are present and correctly formatted.  
  
6. Overall YAML Validation: The main validation function verifies that the YAML structure conforms to the expected format, providing clear error messages and logging any issues that arise during the validation process.