```
# JdbcAppender and PatternLayout
# Define the root logger with file appender
log4j.rootLogger = ALL, DB
# Define the Jdbcappender

log4j.appender.DB=org.apache.log4j.jdbc.JDBCAppender
log4j.appender.DB.URL=jdbc:mysql:///octbatch

# Set Database Driver clas name
log4j.appender.DB.driver=com.mysql.cj.jdbc.Driver

# Set database user name and password
log4j.appender.DB.user=root
log4j.appender.DB.password=root123

# Set the SQL statement to be executed.
log4j.appender.DB.sql=INSERT INTO log_data_tab


(`thread`,`categoryname`,`dateofgeneration`,`methodname`,`lineno`,`message`,`timeel
apsed`,`prioritylevel`)
                  VALUES ('%t','%c',current_timestamp,'%M','%L','%m','%r','%p')

# Define the pattern layout for file appender
log4j.appender.DB.layout=org.apache.log4j.PatternLayout
```

SL4J
====
Simple Logging facade for java
To work with different logging tools , we need to use differnt api's.so moving from one logging tool to another logging tool is complex.
To overcome this problem SL4j has given abstraction over multiple logging tools or api's and provides unified api to work with any logging tool/api.

Logger levels of SL4J are
      => debug<info<trace<warn<error

note: trace can be used for "Auditing" activities.

eg: for user related activities like event handling based code execution will be logged with the support of "trace" log messages.
     eg: button clicked=> actionPerformed() method executed-> this can be logged through "trace" level.

dependencies required are
=========================
```
<dependency>
      <groupId>org.slf4j</groupId>
      <artifactId>slf4j-api</artifactId>
      <version>2.0.7</version>
</dependency>

<!-- https://mvnrepository.com/artifact/org.slf4j/slf4j-simple -->
<dependency>
      <groupId>org.slf4j</groupId>
      <artifactId>slf4j-simple</artifactId>
      <version>2.0.7</version>
      <scope>test</scope>
</dependency>
```

```
<dependency>
	<groupId>org.slf4j</groupId>
	<artifactId>slf4j-log4j12</artifactId>
	<version>2.0.7</version>
</dependency>
```

Note: SL4j generates the log messages by using log4j setup internally based on the instructions collected from log4j.properties
      Springboot internally uses SL4j with log4j to generate the messages ,we can control these messages through "application.properties" file.