1. 词法分析器最主要的函数就是一个scan（扫描输入字符串）函数，其他函数或数据都是为它服务的。首先，一个词法分析器应该存储有自己能识别的词类型，这存储在Identifiers和words两个hash\_map<string,word>中，并在其构造函数中就将Word类里的常量放入words中，同时判断用户的输入，在文件中读取所有sql语句，或者用户的输入就是sql语句，直接赋值给词法分析器的sql变量。构造函数结束后即可调用scan函数获取词素，每运行一次scan函数将得到一个Word类的词（包括标点、标识符、关键字等）。

scan函数是这样设计的：

1.过滤空格、回车、制表符，调用readch函数一次从sql字符串里读取一个字符存于peek（char类型），通过整数ith变量记录当前读到的sql语句的位置，读到sql结尾即把peek赋为感叹号，待后面使用。

2.经过第一步多次调用readch过滤空白字符，得到单个字符peek，判断peek是否为标点符号，一旦是标点符号即可返回响应的Word常量类，特别注意peek应该及时重置为空格！此外，lexer有一个bool变量handleNumber用于判断当前是否接受数字，在某些符号（如左括号、小于号等等）后面可以接受数字，即把handleNumber赋为true，待后面使用。

3.接下来调用getATerm函数尝试读取一个词，此函数可以读取两类此，一类是只包括数字、字母和下划线（或者只包含其中一个）的词，一类是以双引号开头和结尾的词（即表项的值），如果字符peek是感叹号，读取到的term设置为空，即sql语句读取完毕。

4.读到一个term后根据handleNumber和判断term是否为数字的函数IsANumber来判断是否应该返回一个Word(term, Tag::NUMBER)，这个Word不需要存到map数据结构中。

5.然后该判断term是否为words里已有的词了，用迭代器遍历words，如果找到了term就返回相应的words[term]，否则继续往下执行。判断term是否为合理的标识符，如果是，在Identifiers（已有关键词的列表）里查找是否有term，如有则返回，否则新建一个Word类存于Identifiers，然后将之返回。

6.最后三步，判断term是否为双引号中的内容、是否为空（返回表示退出的Word），如果都不是，则报错，将行号和term输出。扫描完毕！