这个框架主要参考mybatis框架的思想。

将分离出来的sql语句写在一个rb文件中，这个rb文件中有一个完整的Ruby Hash实例（从代码形态上看，它有点像JSON），它类似于使用mybatis框架时含有sql语句的xml文档。一个典型例子如下：

{

dbType: "mysql",

mapper: {

select: {

getPerson: {

resultType: "String",

sql: %Q{

select

id

from

person p

where

name = #{person\_name}

}

},

}

}

}

所有的sql语句都写在mapper键后的花括号中，其中，将所有select语句写在select键后面的花括号中。该例子中，getPerson便代表了一个sql语句段，这个getPerson键后面的Hash实例中的sql键的值（后文将直接称为“getPerson动作的sql语句”）即为sql语句段，它是一个Ruby String实例（在Ruby中，被%Q{}包围的字符将成为一个字符串对象）。这个地方写成

sql: “select id from person p where name = #{person\_name}”

也没问题，只要sql键后面的表达式可以生成一个字符串对象（生成是指由Ruby的解释器执行并产生）。

在这个sql语句中，由#{}包围的person\_name是一个Ruby代码串（在这个例子中，它可能是一个Ruby变量或一个Ruby方法）。Ruby在解析读取这个表示sql语句的字符串实例时，会在某个上下文中执行这个代码串，执行结果会被转换为字符串，代替#{person\_name}。