

mapper文件中的sql语句不需要写parameterType（不管传入何种类型的参数都不需要写），mybatis会自动推断传入参数类型，如果写错了类型，反而会导致sql执行失败

mybatis获取参数的两种方式

- #{}
- \${}

\${}会产生sql注入和单引号等问题，所以尽量使用#{}

单引号问题

#{}

使用对象为PreparedStatement，采用占位符的方式获取参数，不需要手动添加单引号问题

```
1 mapper接口:
2 Boolean insert(@Param("id")Integer id, @Param("name")String name);
3
4 mapper.xml文件中使用${}接收参数要手动添加单引号
5 insert into emp(id, name) values(#{id}, #{name})
```

\${}

使用对象为Statement，采用字符串拼接的方式操作SQL，要注意单引号的问题。例如：传入的参数是一个字符串类型，接收参数时要手动添加单引号：

```
1 mapper接口:
2 Boolean insert(@Param("id")Integer id, @Param("name")String name);
3
4 mapper.xml文件中使用${}接收参数要手动添加单引号
5 insert into emp(id, name) values(${id}, '${name}')
```

传递单个参数（参数类型为String，或基本数据类型及其包装类）

#{}

由于采用占位符方式，mybatis接受的参数只与传入参数的顺序有关，因此可以使用任意的名字获取参数，例如：

```
1 mapper接口:
2 Boolean insert(String name);
3
4 mapper.xml文件中，#{ }使用任意名称都能获取传入参数的值
5 insert into emp(name) values(#{bbb})
```

`${}`

```
1 mapper接口:
2 Boolean insert(String name);
3
4 mapper.xml文件中, 使用${value}或${_parameter}获取传入的参数的值
5 insert into emp(name) values(${value})
6 或
7 insert into emp(name) values(${_parameter})
```

如果用以下方式获取参数值：

```
1 mapper接口:
2 Boolean insert(String name);
3
4 mapper.xml文件中
5 insert into emp(name) values(${name})
```

会报类似如下的错误：

```
RuntimeException: There is no getter for property named 'eid' in 'class java.lang.String'
```

传入参数类型为java对象

`#{}和${}`都可以直接通过属性名获取值，但`${}`需要注意单引号问题

`#{}`

```
1 mapper接口:
2 Boolean insert(Emp emp);
3
4 mapper.xml文件中
5 insert into emp values("#{id}", "#{name})
```

`${}`

```
1 mapper接口:
2 Boolean insert(Emp emp);
3
4 mapper.xml文件中
5 insert into emp values(${id}, '${name}')
```

传递多个参数

当传递多个参数时，mybatis会将这些参数放入map集合中，而放入map集合的方式有两种：

方式一：

低版本mybatis：0，1，2，3 ... N-1作为key 高版本mybatis：arg0，arg1，arg2，arg3 ... argN-1作为key

方式二：param1，param2，param3，paramN作为key

#{}

低版本的mybatis用#{0},#{1}或#{param1},#{param2}方式接受参数，高版本的mybatis用#{arg0},#{arg1}或#{param1},#{param2}方式接受参数，我用的1.3.0版本的mybatis-starter就不能使用#{0},#{1}方式

```
1 mapper接口:
2 Boolean insert(String name, int age);
3
4 mapper.xml文件中
5 insert into emp(name, age) values(#{0}, #{1}) 或
6 或
7 insert into emp(name, age) values(#{param1}, #{param2})
8
```

\${}

用\${param1},\${param2}方式接受参数，但要注意上面提到的单引号问题

```
1 mapper接口:
2 Boolean insert(String name, int age);
3
4 mapper.xml文件中
5 insert into emp(name, age) values('${param1}', ${param2})
```

用map传递参数

上面传递多参数时提到mybatis会自动将参数放入map集合中，但我们也可以手动将参数放入map集合中，用map来传递参数：

#{}

```
1 mapper接口:
2 Map<String, Object> map = new HashMap<>(4);
3 map.put("name", "小明");
4 map.put("age", 23);
5 Boolean insertEmp(map);
6
7 mapper.xml文件:
8 insert into emp(name, age) values(#{name}, #{age})
```

`${}`

```
1 mapper接口:
2 Map<String, Object> map = new HashMap<>(4);
3 map.put("name", "小明");
4 map.put("age", 23);
5 Boolean insertEmp(map);
6
7 mapper.xml文件:
8 insert into emp(name, age) values('${name}', ${age})
```

参入类型为List和Array

当传入参数为List或Array时，mybatis会将List或Array放入map中，List以list为键，Array以array为键

这种情况常出现使用foreach遍历的时候

```
1 <foreach collection="list" item="id" open="(" close=")" separator="," >
2   #{id}
3 </foreach>
```

@Param注解

@Param注解可以给传递的参数重新指定key名称

```
1 mapper接口:
2 Boolean insert(@Param("ename")String name, @Param("age")int age);
3
4 mapper.xml文件:
5 insert into emp(name, age) values(#{ename}, #{age})
```