原生SQL语句的编写

String sql = "SELECT P.ID, P.USERNAME, P.PASSWORD, P.FULL\_NAME, "

"P.LAST\_NAME,P.CREATED\_ON, P.UPDATED\_ON " +

"FROM PERSON P, ACCOUNT A " +

"INNER JOIN DEPARTMENT D on D.ID = P.DEPARTMENT\_ID " +

"INNER JOIN COMPANY C on D.COMPANY\_ID = C.ID " +

"WHERE (P.ID = A.ID AND P.FIRST\_NAME like ?) " +

"OR (P.LAST\_NAME like ?) " +

"GROUP BY P.ID " +

"HAVING (P.LAST\_NAME like ?) " +

"OR (P.FIRST\_NAME like ?) " +

"ORDER BY P.ID, P.FULL\_NAME";

使用SQL语句构建器编写SQL语句

private String selectPersonSql() {

return new SQL() {{

SELECT("P.ID, P.USERNAME, P.PASSWORD, P.FULL\_NAME");

SELECT("P.LAST\_NAME, P.CREATED\_ON, P.UPDATED\_ON");

FROM("PERSON P");

FROM("ACCOUNT A");

INNER\_JOIN("DEPARTMENT D on D.ID = P.DEPARTMENT\_ID");

INNER\_JOIN("COMPANY C on D.COMPANY\_ID = C.ID");

WHERE("P.ID = A.ID");

WHERE("P.FIRST\_NAME like ?");

OR();

WHERE("P.LAST\_NAME like ?");

GROUP\_BY("P.ID");

HAVING("P.LAST\_NAME like ?");

OR();

HAVING("P.FIRST\_NAME like ?");

ORDER\_BY("P.ID");

ORDER\_BY("P.FULL\_NAME");

}}.toString();

}

SQL语句构建器的两种使用风格

1.匿名内部类风格

// 匿名内部类风格

public String deletePersonSql() {

return new SQL() {{

DELETE\_FROM("PERSON");

WHERE("ID = #{id}");

}}.toString();

}

2.Builder/Fluent风格

public String insertPersonSql() {

String sql = new SQL()

.INSERT\_INTO("PERSON")

.VALUES("ID, FIRST\_NAME", "#{id}, #{firstName}")

.VALUES("LAST\_NAME", "#{lastName}")

.toString();

return sql;

}

3.动态条件（参数需要使用final修饰，以便返回值中的匿名内部类使用）

public String selectPersonLike(final String id, final String firstName, final String lastName) {

return new SQL() {{

SELECT("P.ID, P.USERNAME, P.PASSWORD, P.FIRST\_NAME, P.LAST\_NAME");

FROM("PERSON P");

if (id != null) {

WHERE("P.ID like #{id}");

}

if (firstName != null) {

WHERE("P.FIRST\_NAME like #{firstName}");

}

if (lastName != null) {

WHERE("P.LAST\_NAME like #{lastName}");

}

ORDER\_BY("P.LAST\_NAME");

}}.toString();

}