

# 生成语句

2025 年 5 月 23 日

## 一、主要任务：

1. 编写程序来获取数据库中的所有 schemas。
2. 生成以上表的 DDL 语句。
3. 生成一个表的插入 SQL 语句。

## 二、实现方法：

1. 获得数据库的表可以用 `select TABLE_NAME from information_schema.tables where table_schema=.....`，获取属性栏可以使用 `select ...from information_schema.columns where table_schema =... and table_name =...`。在 application 中配置数据库信息后 mybatis 可以直接连接到数据库中，使用 mapper 层写下形如上述的查询语句即可调用

2. 使用 `show create table ...` 可以直接生成出该表的 DDL 语句。

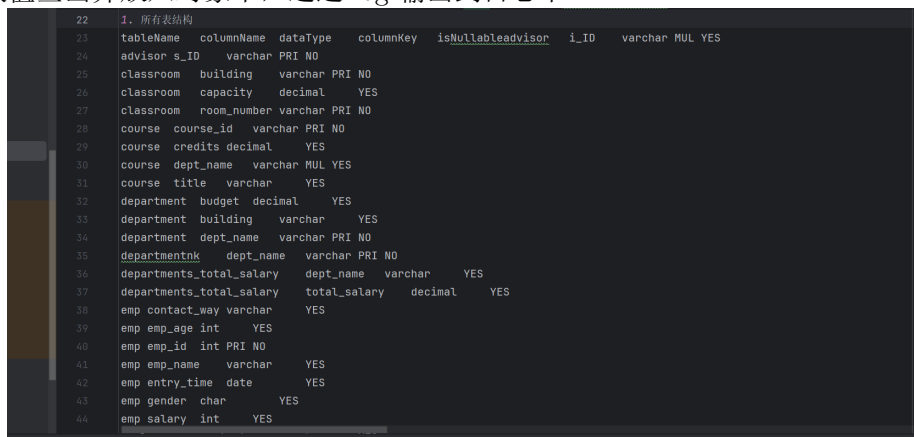
3. 使用 `SELECT COLUMN_NAME FROM INFORMATION_SCHEMA.COLUMNS WHERE TABLE_SCHEMA = DATABASE() AND TABLE_NAME = #tableName` 来获得指定的表的所有列字段

## 三、成果展示：

1.Mapper 中查询表的代码

```
1      @Select("SELECT TABLE_NAME AS tableName,
2              COLUMN_NAME AS columnName,
3              "DATA_TYPE AS dataType, COLUMN_KEY AS columnKey,
4              "IS_NULLABLE AS isNullable, COLUMN_DEFAULT AS
5              columnDefault" +
6              "FROM INFORMATION_SCHEMA.COLUMNS" +
              "WHERE TABLE_SCHEMA = DATABASE()")
      List<TableSchema> getAllTableSchemas();
```

通过 select 依次将表名，列名，数据类型，主键/外键，是否可空，默认数值查出并放入对象中，通过 log 输出到日志中



```
22 1. 所有表结构
23 tableName columnName dataType columnKey isNullableadvisor i_ID varchar MUL YES
24 advisor s_ID varchar PRI NO
25 classroom building varchar PRI NO
26 classroom capacity decimal YES
27 classroom room_number varchar PRI NO
28 course course_id varchar PRI NO
29 course credits decimal YES
30 course dept_name varchar MUL YES
31 course title varchar YES
32 department budget decimal YES
33 department building varchar YES
34 department dept_name varchar PRI NO
35 departmentnk dept_name varchar PRI NO
36 departments_total_salary dept_name varchar YES
37 departments_total_salary total_salary decimal YES
38 emp contact_way varchar YES
39 emp emp_age int YES
40 emp emp_id int PRI NO
41 emp emp_name varchar YES
42 emp entry_time date YES
43 emp gender char YES
44 emp salary int YES
```

## 2. 查询 ddl 语句

```
1      @Select("SHOW CREATE TABLE_${tableName}")
2      Map<String, String> getTableDDL(@Param("
3          tableName") String tableName);
4
5      @Override
6      public Map<String, String> generateAllDDL() {
7          List<TableSchema> schemas = viewMapper.
8              getAllTableSchemas();
9          Set<String> tables = schemas.stream()
10              .map(TableSchema::getTableName)
11              .collect(Collectors.toSet());
12
13          Map<String, String> ddlMap = new
14              LinkedHashMap<>();
15          tables.forEach(table ->
16              ddlMap.put(table, viewMapper.
17                  getTableDDL(table).get("Create_
18                      Table"))
19              );
20          return ddlMap;
21      }
```

service 层先调用 mapper 层查询所有表结构的函数，对其中的每个表

都调用一次查找 ddl 语句的函数，输出到日志中

```
116 所有表DDL
117 CREATE TABLE `mylog5` (
118   `id` int NOT NULL AUTO_INCREMENT,
119   `tb_name` varchar(20) DEFAULT NULL,
120   `colname` varchar(20) DEFAULT NULL,
121   `event` varchar(20) DEFAULT NULL,
122   `oldvalue` varchar(20) DEFAULT NULL,
123   `newvalue` varchar(20) DEFAULT NULL,
124   `date` datetime DEFAULT NULL,
125   PRIMARY KEY (`id`)
126 ) ENGINE=InnoDB AUTO_INCREMENT=41 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
127 null
128 CREATE TABLE `departmentnk` (
129   `dept_name` varchar(20) NOT NULL,
130   PRIMARY KEY (`dept_name`)
131 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
132 CREATE TABLE `student` (
133   `ID` varchar(5) NOT NULL,
134   `name` varchar(20) NOT NULL,
135   `dept_name` varchar(20) DEFAULT NULL,
136   `tot_cred` decimal(3,0) DEFAULT NULL,
137   PRIMARY KEY (`ID`),
138   KEY `student_fk1` (`dept_name`),
139   CONSTRAINT `student_fk1` FOREIGN KEY (`dept_name`) REFERENCES `department` (`dept_name`) ON DELETE SET NULL,
```

### 3. 构建 insert 语句

```
1      @Select("SELECT COLUMN_NAME FROM
2      INFORMATION_SCHEMA.COLUMNS" +
3      "WHERE TABLE_SCHEMA=DATABASE() AND TABLE_NAME
4      =#{tableName}")
5      List<String> getTableColumns(@Param("tableName")
6      String tableName);
7
8      @Override
9      public String generateInsertSQL(String
10      tableName) {
11          List<String> columns = viewMapper.
12          getTableColumns(tableName);
13          String columnsPart = String.join(",",
14          columns);
15          String valuesPart = columns.stream()
16          .map(col -> "#{ " + col + "}")
17          .collect(Collectors.joining(","));
18          return String.format("INSERT INTO %s (%s) VALUES (%s)",
19          tableName, columnsPart, valuesPart);
20      }
21
22      @Override
23      public Map<String, String>
```

```

17         generateAllInsertTemplates() {
18
19             Set<String> tables = viewMapper.
20                 getAllTableSchemas().stream()
21                 .map(TableSchema::getTableName)
22                 .collect(Collectors.toSet());
23
24             Map<String, String> inserts = new
25                 LinkedHashMap<>();
26             tables.forEach(table ->
27                 inserts.put(table, this.
28                     generateInsertSQL(table))
29             );
30             return inserts;
31         }

```

获取每个表的表名，分别调用 generateInsertSQL，针对表名查询出表的字段名并且拼接进插入语句

```

插入SQL模板
-- Table: mylog5
INSERT INTO mylog5 (colname, date, event, id, newvalue, oldvalue, tb_name) VALUES ({colname}, {date}, {event}, {id}, {r
-- Table: stu_view
INSERT INTO stu_view (Birthdayplace, dept_name, Id, Name) VALUES ({Birthdayplace}, {dept_name}, {Id}, {Name})
-- Table: departmentnk
INSERT INTO departmentnk (dept_name) VALUES ({dept_name})
-- Table: student
INSERT INTO student (dept_name, ID, name, tot_cred) VALUES ({dept_name}, {ID}, {name}, {tot_cred})
-- Table: idnamedepartment1
INSERT INTO idnamedepartment1 (Deptname, Location) VALUES ({Deptname}, {Location})
-- Table: emp
INSERT INTO emp (contact_way, emp_age, emp_id, emp_name, entry_time, gender, salary) VALUES ({contact_way}, {emp_age}, {e
-- Table: classroom
INSERT INTO classroom (building, capacity, room_number) VALUES ({building}, {capacity}, {room_number})
-- Table: section
INSERT INTO section (building, course_id, room_number, sec_id, semester, time_slot_id, year) VALUES ({building}, {course_i
-- Table: students
INSERT INTO students (dept_name, ID, name, tot_cred) VALUES ({dept_name}, {ID}, {name}, {tot_cred})
-- Table: studentsc
INSERT INTO studentsc (dept_name, ID, Name) VALUES ({dept_name}, {ID}, {Name})
-- Table: faculty
INSERT INTO faculty (dept_name, ID, name) VALUES ({dept_name}, {ID}, {name})

```

#### 四、结论：

1. 可以用 select 查找语句，在 information\_schema. ... 中查找，获得数据库、表名、属性等内容。
2. 使用 springboot 集成的 mybatis 框架操作数据库是十分简单便捷的