Java MySQL系列课程02

------ 常用SQL

本文档包含以下内容：

1. 创建表
2. 查看表
3. 修改表
4. 查询表
5. 设置表的字符编码

#### 前言：

SQL是关系型数据库通用的结构化查询语言，掌握好SQL是学习关系型数据库的关键，如果没有任何的SQL基础，推荐两本非常好的MySQL入门书籍《MySQL入门很简单》，《深入浅出MySQL》都是适合初学者使用的。

#### 一 创建表：

创建表的SQL语句格式如下：

CREATE TABLE 表名(属性名 数据类型 [完整性约束条件],

属性名 数据类型 [完整性约束条件],

...

属性名 数据类型 [完整性约束条件]

);

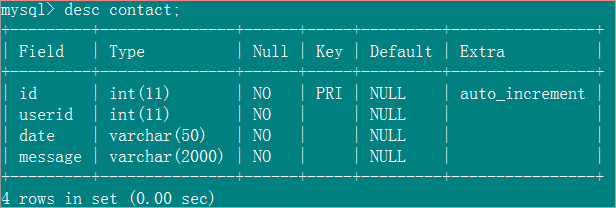
**注意**，创建表结构的时候，要选择合适的数据类型，而且还可以给字段添加完整性约束条件，比如主键，非空键，唯一键等。完整性约束条件都有哪些呢，如下所示：

|  |  |
| --- | --- |
| 约束条件 | 说明 |
| PRIMARY KEY | 修饰的属性为该表的主键，可以区分不同的行记录 |
| FOREIGN KEY | 修饰的属性为该表的外键，关联了父表的主键 |
| NOT NULL | 表示该属性不能为空 |
| UNIQUE | 表示该属性的值是唯一的 |
| AUTO\_INCREMENT | MySQL特色，表示该属性是自增的，整数类型 |
| DEFAULT | 给属性设置默认值 |

#### 二 查看表：

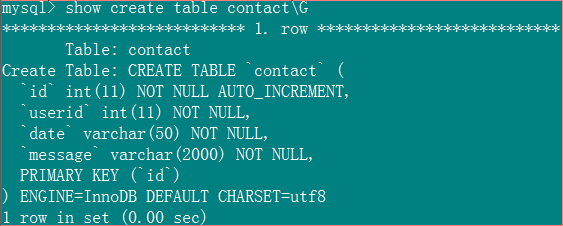
我们经常会查看表的结构、表的存储引擎，表的字符编码等信息，如下操作：

1. **desc Table\_name;**



从上图可以看到，用desc可以查看表的**字段名称，类型，是否非空，约束条件，默认值，备注信息。**

1. **show create table Table\_name\G** （SQL语句可以用；号或者\G结尾，\G结尾打印的格式更清晰）



从上图可以看到，**show create table命令可以打印出创建表的SQL语句，并显示了该编排的存储引擎和字符编码信息。**

**请牢记上面的两个命令，使用的频率很高！**

#### 三 修改表：

如果表在使用的过程中，逐渐发现表的字段不满足设计要求，而且表中已添加大量数据，此时想添加新的字段或者删除已有的字段，可以通过alter命令操作，如下是常用的alter命令**（但是最好还是设计之初，分析好应用场景，一次性设计好表的结构）**：

**修改表名：**

ALTER TABLE 旧表名 RENAME [TO] 新表名;

**修改字段的数据类型：**

ALTER TABLE 表名 MODIFY 属性名 数据类型;

**修改字段名：**

ALTER TABLE 表名 CHANGE 旧属性名 新属性名 新数据类型;

**增加字段：**

ALTER TABLE 表名 ADD 属性名1 数据类型 [完整性约束条件] [FIRST | AFTER 属性名2];

**删除字段：**

ALTER TABLE 表名 DROP 属性名;

**修改字段排列位置：**

ALTER TABLE 表名 MODIFY 属性名1 数据类型 FIRST | AFTER 属性名2;

**修改表的存储引擎：**

ALTER TABLE 表名 ENGINE=InnoDB | MyISAM;

**大家可以用create table创建一个表，操作一下上面的alter语句。**

#### 四 查询表：

SQL的基本查询结构如下：

SELECT 属性列表

FROM 表名

[WHERE 条件表达式1]

[GROUP BY 属性名1 [HAVING 条件表达式2]]

[ORDER BY 属性名2 [ASC | DESC]]

**着重看一下常用的SQL查询都有哪些？面试官经常喜欢问，不同的查询是怎么使用索引的，你知道吗？**

1. **带in的子查询**

[NOT] IN (元素1，元素2，...，元素n)

**如：select \* from user where id in (select stu\_id from grade where average>=60.0);**

**问题：多表联合查询选择子查询好还是连接查询好？**

1. **带BETWEEN AND的范围查询**

[NOT] BETWEEN 取值1 AND 取值2

**如：select \* from user inner join grade on user.id = grade.id where average between 80.0 and 100.0;**

1. **带like的通配符匹配查询**[NOT] LIKE '字符串'

**注意此处LIKE后面的字符串可以携带通配符**

% : 表示0个或任意长度的字符串

\_ ：只能表示单个字符

**如：select \* from news where content like “%亚运会%”;**

1. **空值查询**

IS [NOT] NULL;

**如：select \* from user a left join grade b on a.id = b.userid where b.id is NULL;**

1. **带AND的多条件查询**条件表达式1 AND 条件表达式2 [... AND 条件表达式n]

**如：select \* from user inner join grade on user.id = grade.id where average between 80.0 and 100.0 and user.sex=’男’;**

1. **带OR的多条件查询**

条件表达式1 OR 条件表达式2 [... OR 条件表达式n]

如：**select \* from user inner join grade on user.id = grade.id where grade.math>80.0 or grade.english>80.0;**

1. **去重复查询**

SELECT DISTINCT 属性名

**如：select distinct name from user ;**

**问题：DISTINCT和GROUP BY都能够用来去重，有什么区别呢？**

1. **对结果排序**

ORDER BY 属性名 [ASC | DESC]

**如：select \* from user inner join grade on user.id = grade.id where average between 80.0 and 100.0 order by average desc；**

1. **分组查询**

GROUP BY 属性名 [HAVING 条件表达式]

**如：select name, count(\*) from record group by id;**

1. **union合并查询**

SELECT expression1, expression2, ... expression\_n

FROM tables[WHERE conditions]

UNION [ALL | DISTINCT] **注意：union默认去重，不用修饰distinct，all表示显示所有重复值**

SELECT expression1, expression2, ... expression\_n

FROM tables[WHERE conditions];

**如：SELECT country FROM Websites** **UNION ALL SELECT country FROM apps ORDER BY country;**

1. **Limit分页查询**

不指定初始位置是**LIMIT 记录数**；指定初始位置是**LIMIT 初始位置， 记录数**。

**如：select \* from user limit 10， 20;**

1. **内连接和外连接查询**

**内连接查询：**

SELECT a.属性名1,a.属性名2,...,b,属性名1,b.属性名2... FROM table\_name1 a, table\_name2 b on a.id = b.id where a.属性名 满足某些条件;

SELECT a.属性名1,a.属性名2,...,b,属性名1,b.属性名2... FROM table\_name1 a inner join table\_name2 b on a.id = b.id where a.属性名 满足某些条件;

**如：select a.id,b.average,c.address from user a inner join grade b on a.id = b.userid**

**inner join info c on a.id = c.userid**

**where b.average>=60.0;**

**外连接查询**

**[左连接查询]**

SELECT a.属性名列表, b.属性名列表 FROM table\_name1 a [OUTER] **LEFT JOIN** table\_name2 b on a.id = b.id;

**[右连接查询]**

SELECT a.属性名列表, b.属性名列表 FROM table\_name1 a [OUTER] **RIGHT JOIN** table\_name2 b on a.id = b.id;

**我们一般都会用连接查询代替in子查询进行多表联合查询，子查询的效率远不及连接查询效率高！**

**以上基本包含了所有常用到的SQL查询相关语句，尤其是连接查询，一定要熟练应用！**

#### 五 设置表的字符编码：

有时候MySQL中的表需要存储中文，**需要设置表的字符编码为utf8**，否则默认的字符编码有可能不能正确处理中文，那么在MySQL中，如何设置表的字符编码呢？如下：

通过命令查看MySQL表的字符编码，如下：

**mysql> show variables like 'charac%';**

+--------------------------+---------------------------------------------------------+

| Variable\_name | Value |

+--------------------------+---------------------------------------------------------+

| character\_set\_client | utf8 |

| character\_set\_connection | utf8 |

| character\_set\_database | utf8 |

| character\_set\_filesystem | binary |

| character\_set\_results | utf8 |

| **character\_set\_server | utf8**  **需要设置！**  |

| character\_set\_system | utf8 |

| character\_sets\_dir | C:\Program Files\MySQL\MySQL Server 5.7\share\charsets\ |

+--------------------------+---------------------------------------------------------+

如下设置：

**mysql> set character\_set\_server=utf8;**

Query OK, 0 rows affected (0.00 sec)

字符编码设置utf8成功，MySQL的status**运行状态和运行参数都是通过全局变量来控制的**，用**show status**和**show variables**两个命令可以查看这些信息，用**set**可以设置这些信息，改变配置。