## 数据库索引

索引是对关系数据库表中某字段或某一列进行预排序的数据结构，通过索引可以实现快速的检索、直接定位到符合条件的记录，而不必扫描整个表。

**创建索引：**#ALTER TABLE table\_name ADD INDEX index\_name (column\_name);

**一次创建多个索引：**

#ALTER TABLE table\_name ADD INDEX index\_name(column\_name1,column\_name2);

索引名称任意。

索引的效率取决于字段是否散列，字段的重复值越少，索引的效率越高，主键索引效率最高。

索引的优点是提高检索效率，缺点是在记录增删更的过程中需要维护索引信息，影响效率。

## 数据表的连接查询(一个内连接，两个外连接)

|  |  |  |
| --- | --- | --- |
| 名称 | 语法 | 说明 |
| 内连接 | <表1> inner join <表2> | 返回两张表中存在连接关系的记录 |
| 外连接 | <表1> left/right outer join <表2> | 返回左表（右表）全部记录和另外一张表的关联记录 |
| 外连接 | <表1> full outer join <表2> | 将两张表的所有记录全部选择出来，并且自动把对方不存在的列填充为NULL |

## Sql语句

1. **查看表创建的sql语句:**

#show create table students;

mysql> **SHOW** **CREATE** **TABLE** students;

+*----------+-------------------------------------------------------+*

| students | **CREATE** **TABLE** `students` ( |

| | `id` bigint(20) **NOT** **NULL** AUTO\_INCREMENT, |

| | `class\_id` bigint(20) **NOT** **NULL**, |

| | `name` **varchar**(100) **NOT** **NULL**, |

| | `gender` **varchar**(1) **NOT** **NULL**, |

| | `score` **int**(11) **NOT** **NULL**, |

| | **PRIMARY** **KEY** (`id`) |

| | ) ENGINE=InnoDB AUTO\_INCREMENT=1 **DEFAULT** CHARSET=utf8 |

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

1 row **in** **set** (0.00 sec)

1. **表格列操作**

列新增

#ALTER TABLE students ADD COLUMN brith VARCHAR(10) NOT NULL;

列修改

#ALTER TABLE student CHANGE COLUMN brith birthday VARCHAR(20) NOT NULL;

删除列

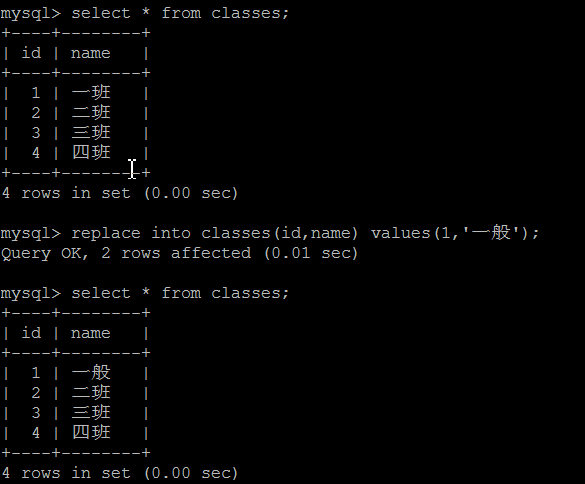
#ALTER TABLE student DROP COLUMN birthday;

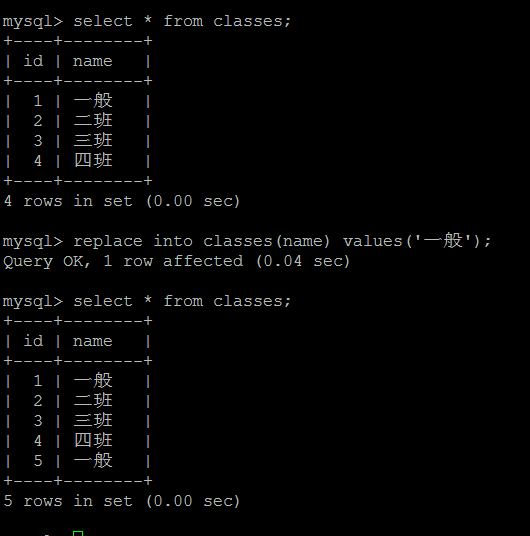
1. 实用sql语句

### 插入或替换记录

REPLACE INTO students …

当记录不存在则插入，存在则替换（主键相同时会替换更新记录信息）





### 插入或更新

如果我们希望插入一条新记录（INSERT），但如果记录已经存在，就更新该记录，此时，可以使用INSERT INTO ... ON DUPLICATE KEY UPDATE ...语句：

**INSERT** **INTO** students (id, class\_id, name, gender, score) **VALUES** (1, 1, '小明', 'F', 99) **ON** DUPLICATE **KEY** **UPDATE** name='小明', gender='F', score=99;

若id=1的记录不存在，INSERT语句将插入新记录，否则，当前id=1的记录将被更新，更新的字段由UPDATE指定。

### 插入或忽略

如果我们希望插入一条新记录（INSERT），但如果记录已经存在，就啥事也不干直接忽略，此时，可以使用INSERT IGNORE INTO ...语句：

**INSERT** IGNORE **INTO** students (id, class\_id, name, gender, score) **VALUES** (1, 1, '小明', 'F', 99);

若id=1的记录不存在，INSERT语句将插入新记录，否则，不执行任何操作。