

索引及应用

1. 简介：索引是系统内部自动维护的一个数据表。作用：极大加快数据的查找速度，这个隐藏的数据表是自动根据某种数据结构算法排好序的。当你进行增删改时，系统同步会更新排序。其查找速度是建立在顺序的基础上的。实现方式：如B+二叉树。创建方式：指定一个表中某个或某些字段为“索引数据字段”。形式：索引类型(字段名)

2. 索引类型

(1) 普通索引

`key(字段名)`

(2) 主键索引

`primary key(字段名)`

(3) 唯一索引

`unique key(字段名)`

注：使用索引字段为搜索条件进行搜索，速度会快很多

(4) 外键索引

`foreign key(banji_id) references banji(id)//外键索引`

注：此时插入学生表中的数据是，`banji_id` 必须是班级表中`id`有存在的值。

删班级表内一条记录，如果学生表中有对应`banji_id`的记录，则删除失败。必须保证删除班级表某条记录时，同步删除学生表中相关记录

3. 应用

百万数据索引查询示例

1. 将有百万数据的数据库导出

```
mysqldump -h localhost -u root -p test > abc/test.sql
```

2. 重命名表

```
alter table tab_test rename to tab_test1
```

3. 在test1添加索引

```
alter table tab_test1 add key(name);
```

```
--+
| tab_millibillion1 | CREATE TABLE `tab_millibillion1` (
|   `id` int(11) NOT NULL AUTO_INCREMENT,
|   `name` varchar(100) NOT NULL COMMENT '姓名',
|   `sex` enum('男','女') DEFAULT NULL COMMENT '性别',
|   `age` tinyint(4) DEFAULT NULL COMMENT '年龄',
|   `fav` set('乒乓球','篮球','足球') DEFAULT '篮球',
|   PRIMARY KEY (`id`)
| ) ENGINE=InnoDB AUTO_INCREMENT=1000001 DEFAULT CHARSET=utf8
|
+-----+
|
+-----+
1 row in set (0.00 sec)

mysql> alter table tab_millibillion1 add key(name);
Query OK, 0 rows affected (6.01 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> show create table tab_millibillion1;
+-----+
|
+-----+
|
+-----+
---+
| Table                | Create Table
|
|
+-----+
|
+-----+
|
+-----+
---+
| tab_millibillion1 | CREATE TABLE `tab_millibillion1` (
|   `id` int(11) NOT NULL AUTO_INCREMENT,
|   `name` varchar(100) NOT NULL COMMENT '姓名',
|   `sex` enum('男','女') DEFAULT NULL COMMENT '性别',
|   `age` tinyint(4) DEFAULT NULL COMMENT '年龄',
|   `fav` set('乒乓球','篮球','足球') DEFAULT '篮球',
|   PRIMARY KEY (`id`),
|   KEY `name` (`name`)
| ) ENGINE=InnoDB AUTO_INCREMENT=1000001 DEFAULT CHARSET=utf8
|
+-----+
```

```
mysql -h localhost -u root -p tab test < abc/test.sql
```

```
mysql> select count(*) as 同名数 from tab_millibillion where name='陈伟南123';
```

```
mysql> select count(*) as 同名数 from tab_millibillion1 where name='陈伟南123';
```

```

+-----+
| 同名数  |
+-----+
|          |
|          |
+-----+
1 row in set (0.00 sec)

```