**2015-2016中山大学本科生实验报告**

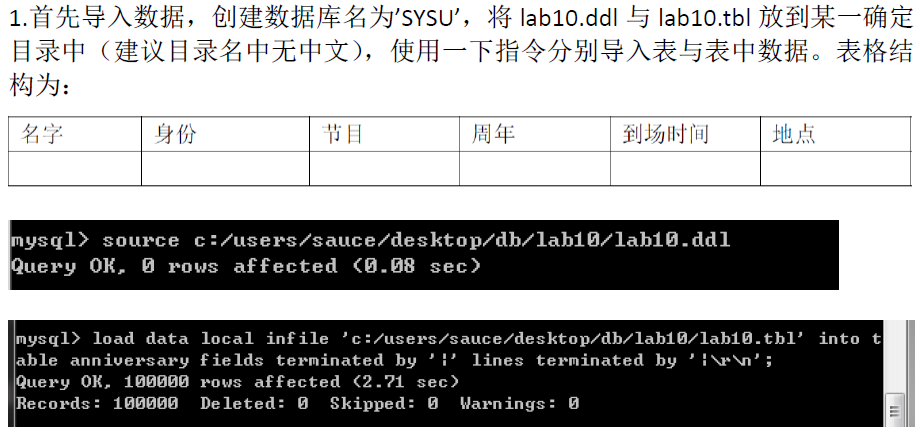
**科目**：数据库系统实验 **教师：**郑贵锋

|  |  |  |  |
| --- | --- | --- | --- |
| 年级 | 13级 | 专业（方向） | 移动信息工程 |
| 学号 | 13354485 | 姓名 | 朱琳 |
| 电话 | 13726231932 | Email | [280273861@qq.com](mailto:280273861@qq.com) |

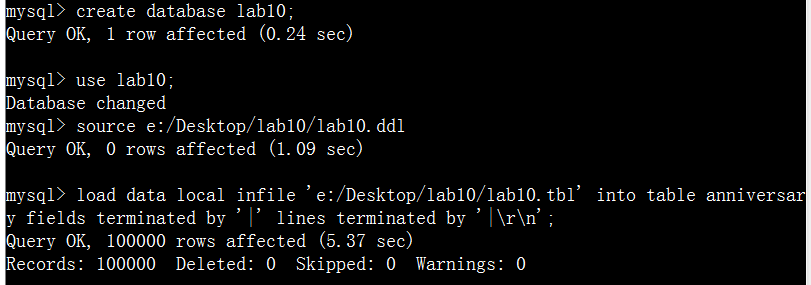
**一 实验目的**



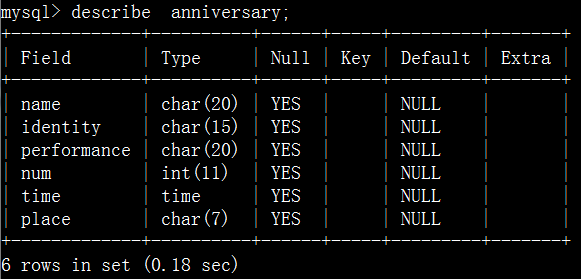
**二 实验题目以及实验结果**



**【结果】**



**【查看下表的结构】**



**2.查询一下内容，记录查询时间。**

**(1)参加了90周年庆的所有人员以及他们的身份。**

Mysql语句：select name,identity from anniversary where num=90;

**【结果】**因为数据太多，只截取最后的结果，看出是1104行，0.12s。



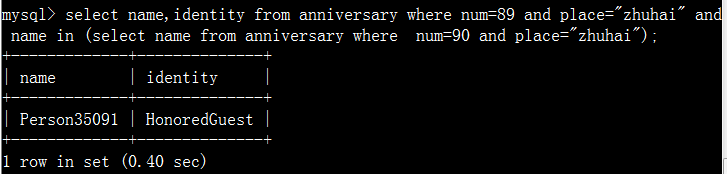
**(2)在东校区50周年庆上参加唱歌活动的所有人员。**

Mysql语句：select name,identity from anniversary where num=50 and place="East" and performance="Singing";

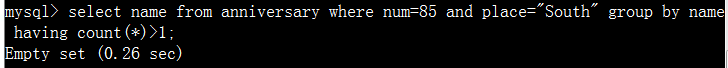
【结果】因为数据太多，只截取最后的结果，看出是21行，0.11s。



**(3)连续参加了珠海校区89和90周年庆的所有人员以及身份。**

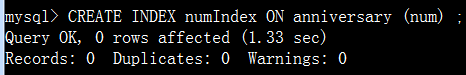


**(4)参加了南校区85周年庆并进行了两项以上节目表演的所有人员。**



1. **在num上建立索引，重做步骤2，对比时间区别。**

**【建立索引】**



**(1)参加了90周年庆的所有人员以及他们的身份。**

Mysql语句：select name,identity from anniversary where num=90;

**【结果】**因为数据太多，只截取最后的结果，看出是1104行，0.12s。



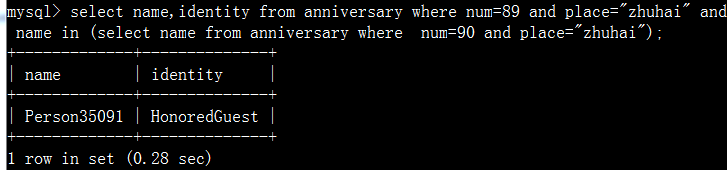
1. **在东校区50周年庆上参加唱歌活动的所有人员。**

Mysql语句：select name,identity from anniversary where num=50 and place="East" and performance="Singing";

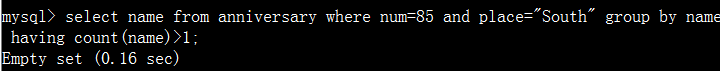
【结果】因为数据太多，只截取最后的结果，看出是21行，0.11s。



1. **连续参加了珠海校区89和90周年庆的所有人员以及身份。**

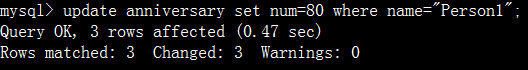


1. **参加了南校区85周年庆并进行了两项以上节目表演的所有人员。**



**4.插入、删除、更新，记录下时间（步骤3中建立的索引不删除）。**

1. **将名字为Person1的num修改为80。**



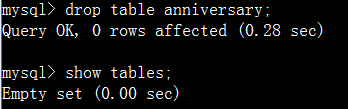
1. **将名字为Person1的人删除。**

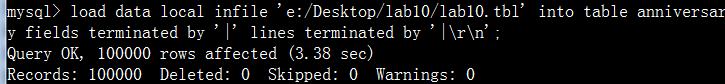


**(3)插入一条记录(Peter, Actor, Singing, 90, 15:40, Zhuhai).**



1. **把表drop掉，重新载入，重做步骤4，对比时间。**





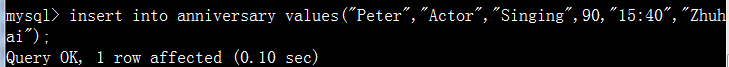
1. **将名字为Person1的num修改为80。**



**(2)将名字为Person1的人删除。**



**(3)插入一条记录(Peter, Actor, Singing, 90, 15:40, Zhuhai).**



1. **设计一个组合索引，要求比步骤2中每一项要更快。**



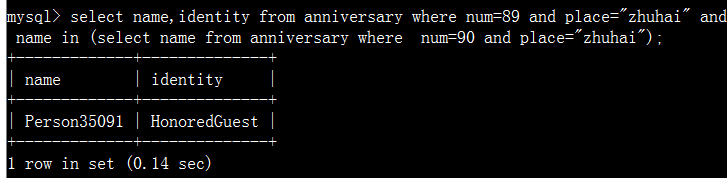
1. **参加了90周年庆的所有人员以及他们的身份。**



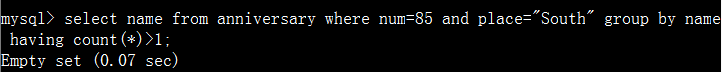
1. **在东校区50周年庆上参加唱歌活动的所有人员。**



1. **连续参加了珠海校区89和90周年庆的所有人员以及身份。**



**(4)参加了南校区85周年庆并进行了两项以上节目表演的所有人员。**



**三 实验结果对比**

1. **第二步和第三步以及第六步时间上的区别**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **第一小题** | **第二小题** | **第三小题** | **第四小题** |
| **不加索引** | 0.36 | 0.11 | 0.40 | 0.26 |
| **加上num索引** | 0.19 | 0.11 | 0.28 | 0.22 |
| **组合索引num,place** | 0.25 | 0.05 | 0.14 | 0.07 |

**【分析】**做了很多遍，才做到理想的效果，虽然也不是完美。这都是建立索引之后第一次运行的结果。若建立索引之后，运行重复的语句，第一次运行以后会非常快。具体的在【实验感想】部分详细描述。

**2.第四步和第五步时间上区分**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **更新** | **删除** | **插入** |
| **有num索引** | 0.47 | 0.59 | 0.08 |
| **不加索引** | 0.43 | 0.57 | 0.10 |

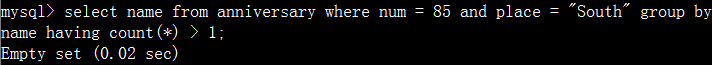
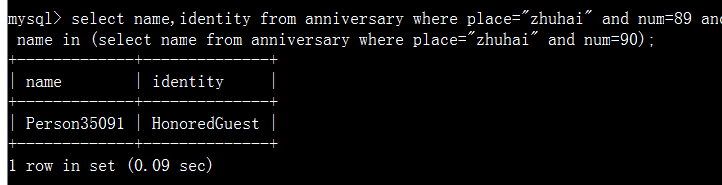
**【分析】**从上述时间对比上可以看出加了索引好像并没有什么用，因为where里面根本没有用到nun这个属性，所以也用不到索引。之所以有索引的时候慢一点可能是因为B+树是个动态的索引，我们增删改数据之后索引也要进行相应的调整，但是对于这种只有一个语句的这些操作，影响是微乎其微的。

**三 实验感想**

1. 这次实验看起来很简单的样子，但是在做的过程中还是有些许的混乱，本来做了一遍感觉没什么问题了，结果还是出现了一些毛病什么的。
2. 关于第四小问那个表演节目两项以上的人什么的，我最初的思路是知道使用group by，然后再COUNT，然后翻了下之前做的关于Group by这个函数的那次实验，发现跟这次的有很大的差别，然后我就有些蒙圈。后来才知道原来需要使用having，真是才疏学浅，之前竟然不知道还有这么个语句。通过查阅资料得知，having字句可以让我们筛选成组后的各种数据，where字句在聚合前先筛选记录。而 having子句在聚合后对组记录进行筛选。总之就是having一般跟在group by之后了，其他的情况用where基本都可以解决。
3. 最坑的就是，添加索引之后，如果第一次执行一个语句，那么运行时间还是正常的，但是第二次调用就变得超级快超级快，具体看如下截图：

以下是添加索引之后第二次或者以上执行语句：





原来都是这样：



前后差别非常大。经过反复验证，都是在添加索引之后的第二次以后运行速度变快的。但是奇怪的是，如果不添加索引，这种现象并不存在。就是说，在没有添加索引的情况下，重复运行指令并不会提升运行速度。

考虑到mysql可能内置了cache，但是没添加索引的时候竟然没有cache现象，这很奇怪啊。并不知道mysql对他们做了什么。。而且运行时间也总是抽风。心真是累。