

《数据库系统实验》

实验报告

题目	(实验 7)
姓名	TRY
学号	
班级	计科 X 班

一. 实验环境:

操作系统: windows

图形界面: mysql3.7.31, mysql workbench

二. 实验内容与完成情况:

索引的使用效果测试。参照实验示例上机操作, 增大 test 表的记录到 8 万条或更多, 重做实验。多次记录耗时, 并作分析比较。

2.1 创建 test 表, 创建存储过程, 并调用

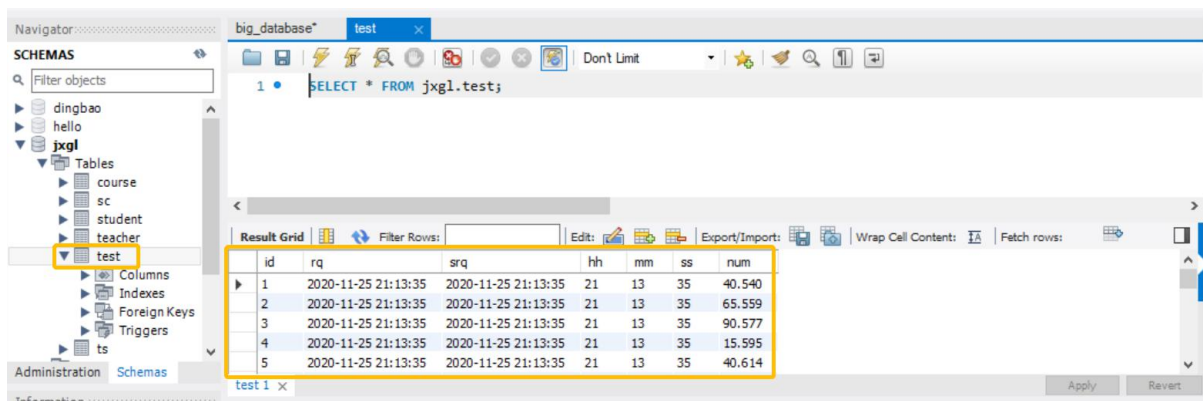
代码:

```
create table test(id int unique auto_increment, rq datetime null, wrrq varchar(20) null,
hh smallint null, mm smallint null, ss smallint null, num numeric(12,3), primary key(id))
auto_increment=1 engine=MyISAM;

delimiter //
create procedure p1()
begin
    set @i=1;
    while @i<=80000 do
        insert into test(rq, srq, hh, mm, ss, num)
            values(now(),now(),hour(now()),minute(now()),second(now()),rand(@i)*100);
        set @i = @i+1;
    end while;
end//

call p1();
```

运行结果:



id	rq	srq	hh	mm	ss	num
79997	2020-11-25 21:13:40	2020-11-25 21:13:40	21	13	40	5.341
79998	2020-11-25 21:13:40	2020-11-25 21:13:40	21	13	40	30.359
79999	2020-11-25 21:13:40	2020-11-25 21:13:40	21	13	40	55.377
80000	2020-11-25 21:13:40	2020-11-25 21:13:40	21	13	40	80.396
NULL	NULL	NULL	NULL	NULL	NULL	NULL

2.2 未建索引时按以下步骤操作

2.2.1 单记录插入

代码:

```
delimiter //
```

```
select @i:=max(id) from test;
```

```
insert into test(rq,srq,hh,mm,ss,num)
```

```
values(now(),now(),hour(now()),minute(now()),second(now()),rand(@i)*100);
```

运行结果:

id	rq	srq	hh	mm	ss	num
79998	2020-11-25 21:13:40	2020-11-25 21:13:40	21	13	40	30.359
79999	2020-11-25 21:13:40	2020-11-25 21:13:40	21	13	40	55.377
80000	2020-11-25 21:13:40	2020-11-25 21:13:40	21	13	40	80.396
80001	2020-11-25 21:17:30	2020-11-25 21:17:30	21	17	30	80.396
NULL	NULL	NULL	NULL	NULL	NULL	NULL

耗时: 0.032sec

0.032 sec / 0.000 sec

2.2.2 查询所有记录, 按 id 排序

代码:

```
select * from test order by id;
```

运行结果:

The screenshot shows a database query editor with the following SQL code:

```

39 insert into test(rq,srq,hh,mm,ss,num)
40 values(now(),now(),hour(now()),minute(now()),second(now()),rand(@i)*100);
41
42 select * from test order by id;
43

```

The result grid shows the following data:

id	rq	srq	hh	mm	ss	num
1	2020-11-25 21:13:35	2020-11-25 21:13:35	21	13	35	40.540
2	2020-11-25 21:13:35	2020-11-25 21:13:35	21	13	35	65.599
3	2020-11-25 21:13:35	2020-11-25 21:13:35	21	13	35	90.577
4	2020-11-25 21:13:35	2020-11-25 21:13:35	21	13	35	15.595
5	2020-11-25 21:13:35	2020-11-25 21:13:35	21	13	35	40.614

耗时：0.110sec

16 21:20:02 select * from test order by id 80001 row(s) returned 0.110 sec / 0.109 sec

2.2.3 查询所有记录，按 mm 排序

代码：

```
select * from test order by mm;
```

运行结果：

The screenshot shows a database query editor with the following SQL code:

```

39 insert into test(rq,srq,hh,mm,ss,num)
40 values(now(),now(),hour(now()),minute(now()),second(now()),rand(@i)*100);
41
42 select * from test order by id;
43
44 select * from test order by mm;

```

The result grid shows the following data:

id	rq	srq	hh	mm	ss	num
76225	2020-11-25 21:13:40	2020-11-25 21:13:40	21	13	40	36.272
76226	2020-11-25 21:13:40	2020-11-25 21:13:40	21	13	40	61.290
76227	2020-11-25 21:13:40	2020-11-25 21:13:40	21	13	40	86.308
76228	2020-11-25 21:13:40	2020-11-25 21:13:40	21	13	40	11.327
76229	2020-11-25 21:13:40	2020-11-25 21:13:40	21	13	40	36.345
76230	2020-11-25 21:13:40	2020-11-25 21:13:40	21	13	40	61.363

耗时：0.031sec

17 21:21:30 select * from test order by mm 80001 row(s) returned 0.031 sec / 0.109 sec

2.2.4 单记录查询

代码：

```
select id from test where id=51;
```

运行结果：

id
51

耗时：0 sec

18 21:34:58 select id from test where id=51 1 row(s) returned 0.000 sec / 0.000 sec

2.3 对 test 表的 mm 字段建立非聚集索引

2.3.1 建立索引耗时

代码：

```
create index indexname1 on test(mm);
```

运行结果：耗时 0.344sec

33 21:48:37 create index indexname1 on test(mm) 80003 row(s) affected Records: 80003 Duplicates: 0 Warnings: 0 0.344 sec

2.3.2 单记录插入

代码：

```
delimiter //
select @i:=max(id) from test;
insert into test(rq,srq,hh,mm,ss,num)
values(now(),now(),hour(now()),minute(now()),second(now()),rand(@i)*100);
```

运行结果：

	id	rq	srq	hh	mm	ss	num
	80000	2020-11-25 21:13:40	2020-11-25 21:13:40	21	13	40	80.396
	80001	2020-11-25 21:17:30	2020-11-25 21:17:30	21	17	30	80.396
	80002	2020-11-25 21:36:53	2020-11-25 21:36:53	21	36	53	5.414
	80003	2020-11-25 21:41:32	2020-11-25 21:41:32	21	41	32	30.432
	80004	2020-11-25 21:49:17	2020-11-25 21:49:17	21	49	17	55.451

耗时：0sec，明显小于上面的 0.032sec

34 21:49:17 select @i:=max(id) from test; insert into test(rq,srq,hh,mm,ss,num) values(now(),now(),... 1 row(s) returned 0.000 sec / 0.000 sec

2.3.3 查询所有记录，按 id 排序

代码：

```
select * from test order by id;
```

运行结果：

	id	rq	srq	hh	mm	ss	num
▶	1	2020-11-25 21:13:35	2020-11-25 21:13:35	21	13	35	40.540
	2	2020-11-25 21:13:35	2020-11-25 21:13:35	21	13	35	65.559
	3	2020-11-25 21:13:35	2020-11-25 21:13:35	21	13	35	90.577
	4	2020-11-25 21:13:35	2020-11-25 21:13:35	21	13	35	15.595
	5	2020-11-25 21:13:35	2020-11-25 21:13:35	21	13	35	40.614

耗时：0.062sec，明显小于上面的 0.110sec

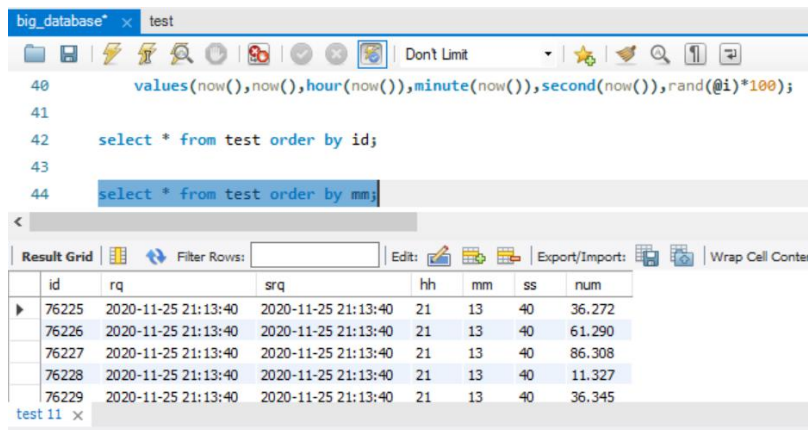
37 21:51:18 select * from test order by id 80004 row(s) returned 0.062 sec / 0.094 sec

2.3.4 查询所有记录，按 mm 排序

代码：

```
select * from test order by mm;
```

运行结果：



耗时：0.047sec，多于上面无索引使得 0.031sec

39 21:52:17 select * from test order by mm 80005 row(s) returned 0.047 sec / 0.109 sec

2.3.5 单记录查询

代码：

select id from test where id=51;

运行结果：

Result Grid	
id	
51	
NULL	

耗时：0sec，和上面的一样

40 21:53:25 select id from test where id=51 1 row(s) returned 0.000 sec / 0.000 sec

2.3.6 删除索引

代码：

drop index indexname1 on test;

运行结果及耗时：

42 21:54:20 drop index indexname1 on test 80005 row(s) affected Records: 80005 Duplicates: 0 Warnings: 0 0.281 sec