

## 实验十 数据库索引与存储效率

### 一、实验目的

- 1、创建索引
- 2、体会通过建立索引，提高数据存取效率

### 二、实验内容

对约有 16 万条记录的表，进行单记录插入与所有记录排序查询所需时间的比较，测试使用索引与不使用索引、使用聚集索引与非聚集索引、对唯一值字段与非唯一值字段建索引并排序等情况的执行状况，体会使用索引的作用和意义。

### 三、实验要求

按照实验步骤完成实验内容，并且请以表格的形式给出在你的计算机环境下，每个操作在不同索引下的执行时间，比较时间差异。

### 四、实验步骤

- (1) 创建表 itb1，并插入 16 万条记录。

对约有 16 万条记录的表，进行单记录插入与所有记录排序查询(分别对两个不同字段进行排序)执行耗时的(以毫秒为单位)比较，测试使用索引与不使用索引、使用聚集索引与非聚集索引、对唯一值字段与非唯一值字段建索引并排序等情况的执行状况。从中能领略到使用索引的作用与意义。并能在其他需建索引场合，利用这种测试办法来作分析与比较。

- (1) 创建表 itbl，并插入 16 万条记录。

请在 SQL Server 集成管理器查询子窗口中，选择某用户数据库，执行如下命令，生成 16 万条记录，如图 8-1 所示。

```
CREATE TABLE itbl(id bigint IDENTITY (1, 1) NOT NULL, rq datetime NULL, srq varchar(20) NULL, hh  
smallint NULL, mm smallint NULL, ss smallint NULL, num numeric(12, 3))  
  
declare @i int  
select @i = 1  
while @i <= 160000  
begin  
    insert into itbl(rq,srq,hh,mm,ss,num) values(getdate(),cast(getdate() as varchar(20)),DATE-  
PART(hh, getdate()),DATEPART(mi, getdate()),DATEPART(ss, getdate()),cast(rand(@i) * 100 as numeric  
(12,3)))  
    Select @i = @i + 1  
End
```

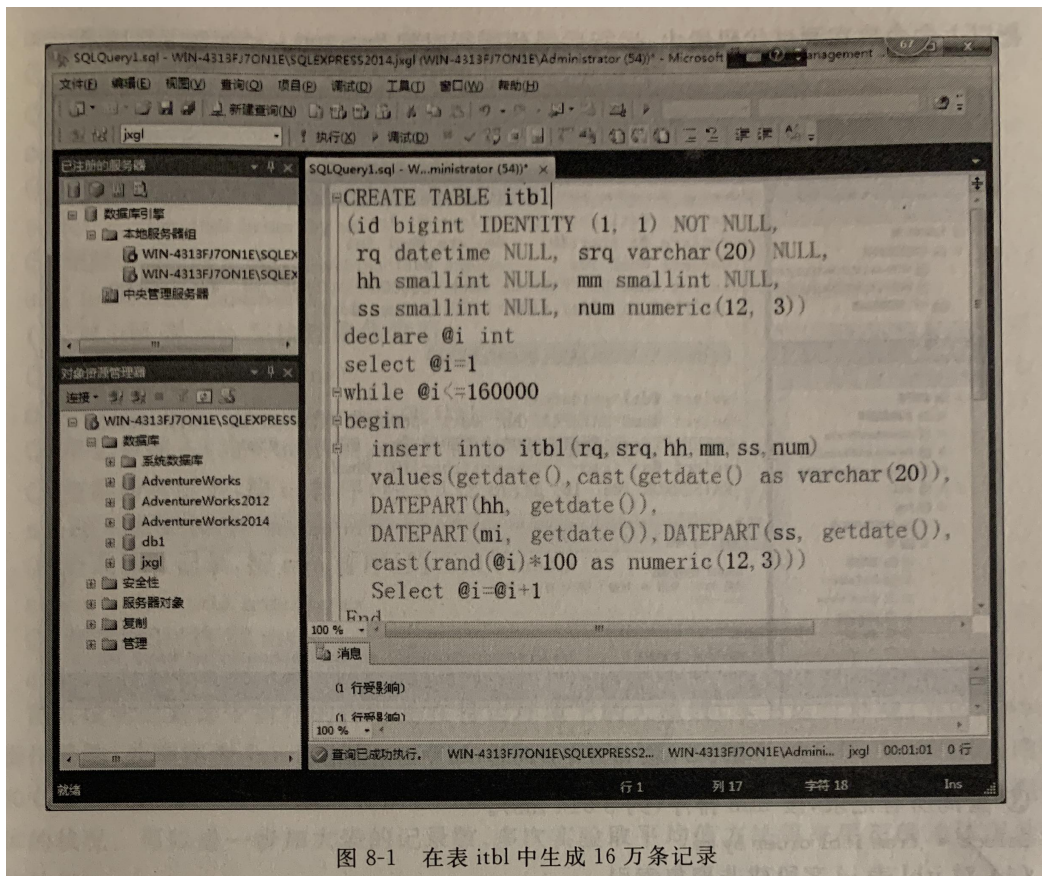
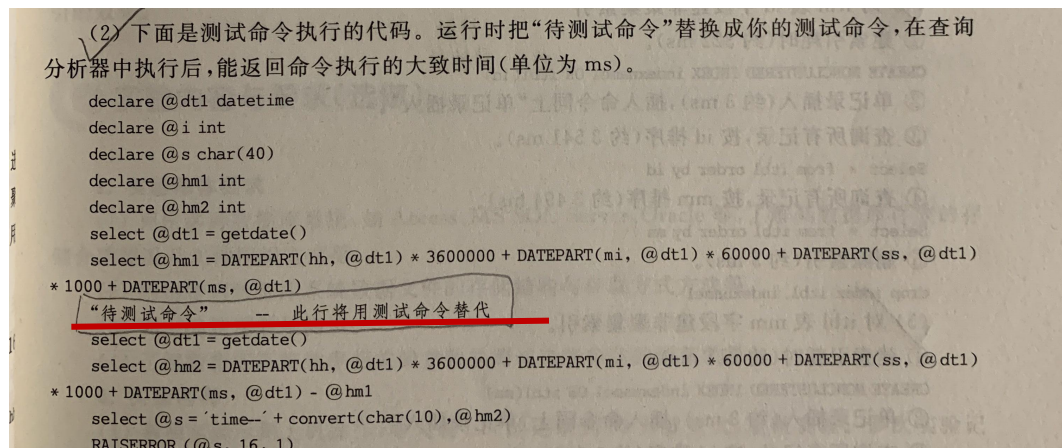


图 8-1 在表 itbl 中生成 16 万条记录

(2) 下面是测试命令执行的代码，运行时请将“待测试命令”替换成你的命令，执行后，能返回执行的大致时间（单位为 ms）。





(3) 未建立索引时, 请按照以下步骤操作。

待测试命令共 3 条, 分别为: 单记录插入, 查询所有记录按 id 排序, 查询所有记录按 mm 排序)

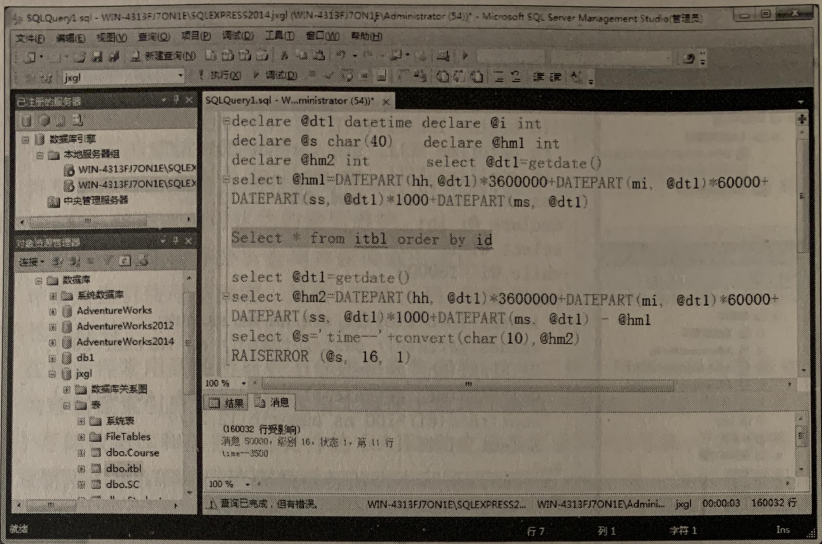
(3) 未建索引时按以下步骤操作。

① 单记录插入(约 3 ms, 给出的毫秒数是在特定环境下得出的, 仅作参考, 下同)。

```
select @i = DATEPART(ms, @dt1)
insert into itb1(rq,srq,hh,mm,ss,num) values(getdate(),cast(getdate() as varchar(20)),DATEPART
(hh, getdate()),DATEPART(mi, getdate()),
DATEPART(ss, getdate()),cast(rand(@i) * 100 as numeric(12,3)))
```

② 查询所有记录, 按 id 排序(约 3 719 ms)

```
Select * from itb1 order by id
```



③ 查询所有记录, 按 mm 排序(约 3 514 ms)。

```
Select * from itb1 order by mm
```

(4) 对 itb1 表 id 字段建非聚集索引

此部分共 5 条语句, 参考下面截图

(5) 对 itb1 表 mm 字段建立非聚集索引

此部分共 5 条语句, 参考下面截图

(6) 对 itb1 表 id 字段建聚集索引

此部分共 5 条语句, 参考下面截图

(7) 对 itb1 表 mm 字段建聚集索引

此部分共 5 条语句, 参考下面截图

(8) 参照实验示例, 增大表 itb1 的记录到 32 万或更大, 重做实验, 多次记录实验耗时, 作分析比较 (选做)。

截图见下页

✓(4) 对 itbl 表 id 字段建非聚集索引

① 建索引耗时(约 322 ms)。

```
CREATE NONCLUSTERED INDEX indexname1 ON itbl(id)
```

② 单记录插入(约 3 ms),插入命令同上“单记录插入”。

③ 查询所有记录,按 id 排序(约 3 541 ms)。

```
Select * from itbl order by id
```

④ 查询所有记录,按 mm 排序(约 3 494 ms)。

```
Select * from itbl order by mm
```

⑤ 删除索引(约 5 ms)。

```
drop index itbl.indexname1
```

✓(5) 对 itbl 表 mm 字段建非聚集索引。

① 建索引耗时(约 211 ms)。

```
CREATE NONCLUSTERED INDEX indexname1 ON itbl(mm)
```

② 单记录插入(约 3 ms),插入命令同上“单记录插入”。

③ 查询所有记录,按 id 排序(约 3 512 ms)。

```
Select * from itbl order by id
```

④ 查询所有记录,按 mm 排序(约 3 487 ms)。

```
Select * from itbl order by mm
```

⑤ 删除索引(约 7 ms)。

```
drop index itbl.indexname1
```

✓(6) 对 itbl 表 id 字段建聚集索引。

① 建索引耗时(约 605 ms)。

```
CREATE CLUSTERED INDEX indexname1 ON itbl(id)
```

② 单记录插入(约 3 ms),插入命令同上“单记录插入”。

③ 查询所有记录,按 id 排序(约 3 017 ms)。

```
Select * from itbl order by id
```

④ 查询所有记录,按 mm 排序(约 3 460 ms)。

```
Select * from itbl order by mm
```

⑤ 删除索引(约 5 ms)。

```
drop index itbl.indexname1
```

✓(7) 对 itbl 表 mm 字段建聚集索引。

① 建索引耗时(约 552 ms)。

```
CREATE CLUSTERED INDEX indexname1 ON itbl(mm)
```

② 单记录插入(约 8 ms),插入命令同上“单记录插入”。

③ 查询所有记录,按 id 排序(约 3 432 ms)。

```
Select * from itbl order by id
```

④ 查询所有记录,按 mm 排序(约 3 017 ms)。

```
Select * from itbl order by mm
```

⑤ 删除索引(约 22 ms)。

```
drop index itbl.indexname1
```

需要说明的是命令执行的耗时是在特定环境下的大概数(本书运行环境:Windows 7 32 位操作系统;处理器为 Intel(R)Core(TM)2 Duo CPU P8600 @2.40 GHz 2.40 Ghz;内存为 2.00 GB(1.90 GB 可用)),因为有很多因素会影响到执行的时间,通过比较能说明一个粗略而大体的状况。可以进一步加大表的记录数、多次实验取平均值方法等来更正确地体现使用索引的效果。