北京邮电大学

实验报告

课程名称：数据库系统原理

实验名称：数据库物理设计

计算机系2015211312班 姓名刘佳鑫

计算机系2015211312班 姓名万诗婕

教师吴起凡 成绩\_\_\_\_\_\_\_\_\_

2018年5月23日

### 实验目的

1. 理解SQL SERVER数据库的物理设计的几个主要部分。
2. 掌握SQL SERVER数据库的文件（包括主文件、辅助文件和日志文件）和文件组（主文件组、其他文件组）的设计。
3. 掌握SQL SERVER数据库索引的设计，包括聚集索引、非聚集索引和复合索引等，分析索引对select/update/insert/delete的影响。
4. 掌握利用SQL Server Management Studio分析观察SQL语句的查询执行计划，对比查询需求相同、实现方式不一样的不同SQL语句在查询结果、查询执行计划、执行时间上的区别。

### 实验环境

采用SQL Server数据库管理系统作为实验平台。SQL Server 2012。

### 实验内容

1. 查看数据库的文件存储格式
2. 数据库文件设计
3. 数据库文件组的设计
4. 数据库索引的设计及对select操作的影响分析
5. 强制使用索引和强制不使用索引的比较
6. 查看索引空间和数据空间
7. 数据库索引的设计及对insert/update/delete操作的影响分析
8. 聚集索引和非聚集索引的比较

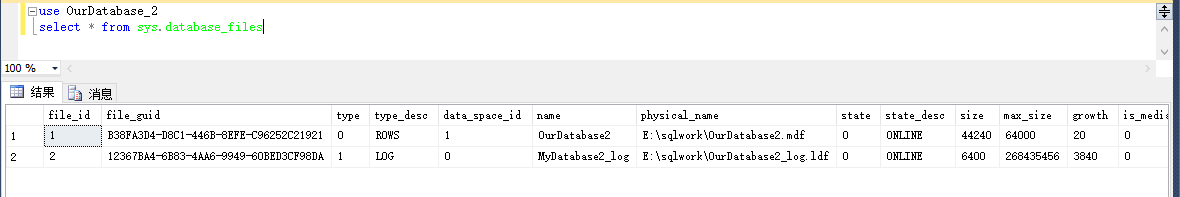
### 实验步骤

**实验1 数据库文件及文件组的设计**

**实验1.1 查看数据库的文件存储格式**

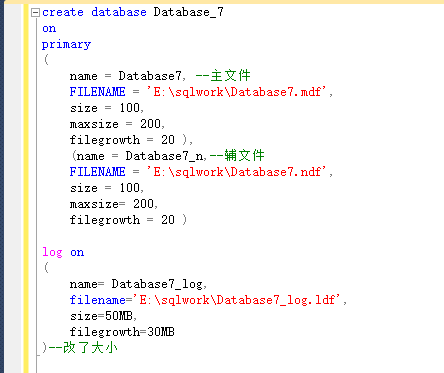
针对在实验1、实验2/3中创建的TD-LTE配置数据库，采用select语句，从数据库元数据视图sys.database\_files中观察数据库文件的组成，包括文件类型、文件名、文件大小、文件存储位置等。

对已有的数据库，可用相应SQL语句查询数据库现有的文件及其属性，这些信息可从数据库元数据视图sys.database\_files中看到。

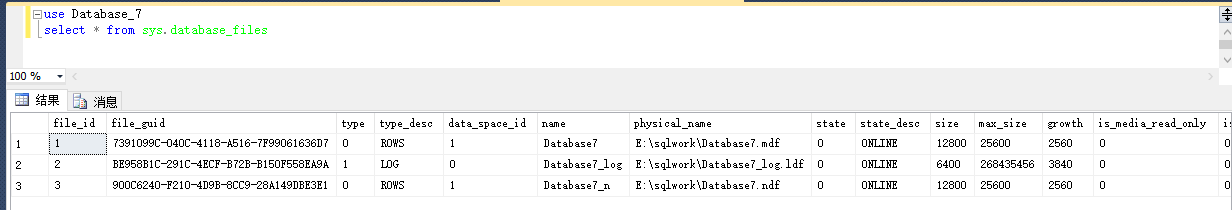


**实验1.2 数据库文件设计**

1. 使用CREATE DATABASE语句创建TD-LTE配置数据库，指定数据库相关属性，如逻辑文件名（NAME）、物理文件名（FILENAME）、初始文件大小（SIZE）、文件最大尺寸（MAXSIZE）、文件成长增量（FILEGROWTH）；

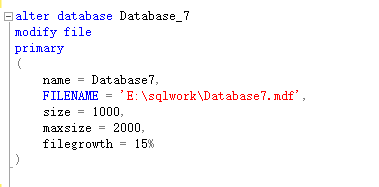


创建后：

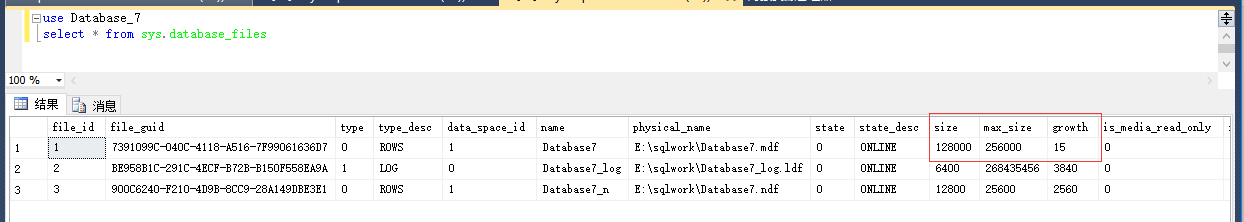


1. 通过ALTER DATABASE语句，对已经创建的数据库文件的属性进行修改；

修改后，发现已经成功修改。



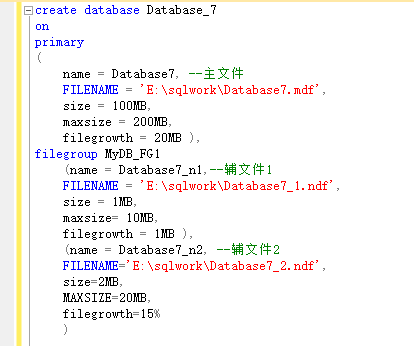
1. 通过数据库元数据视图sys.database\_files中观察创建、修改后的数据库的文件属性。

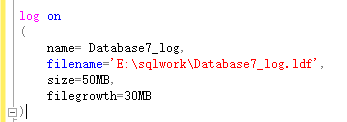


**实验1.3 数据库文件组的设计**

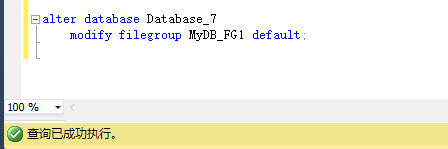
文件和文件组示例（在设计自己的文件组时，注意文件路径。）

（1）创建文件和文件组

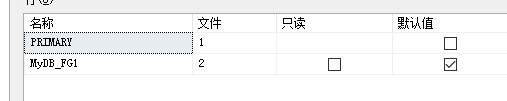




（2）修改默认文件组



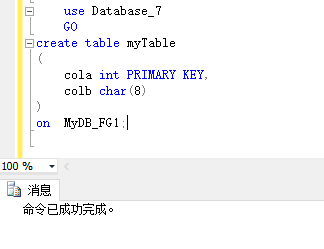
之后再看，成功设置为默认



（3）在用户定义的文件组上新建表：

开始的时候：

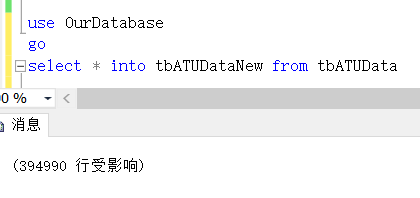
之后发现没有加use 新的database。加了之后就好了。

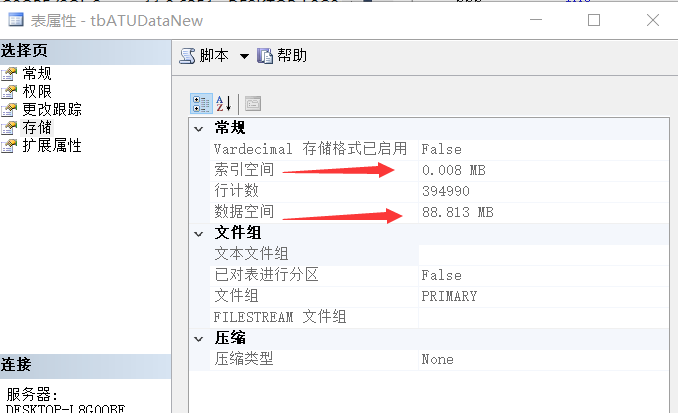


**实验2 数据库索引的设计**

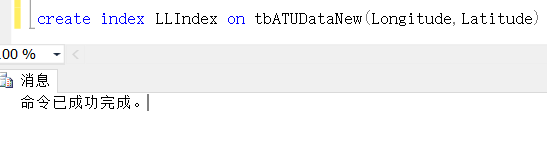
**实验2.1数据库索引的设计及对select操作的影响分析**

**第1步**  **创建tbATUData的备份表tbATUDataNew；**

创建备份表tbATUDataNew

**第2步 查看表属性里面的索引和数据存储空间。**

**第3步 在tbATUDatanew创建索引。**

索引属性建议：可以在Longitude和Latitude上设计索引。

**第4步** **编写select语句，使用索引访问tbATUDatanew；编写select语句，不使用索引方式，访问tbATUData，将2条select语句，同时提交给DBMS执行后，从2个执行结果窗口中，判断执行的结果是否一致，并观察各自执行效果、查询执行计划、时间对比。**

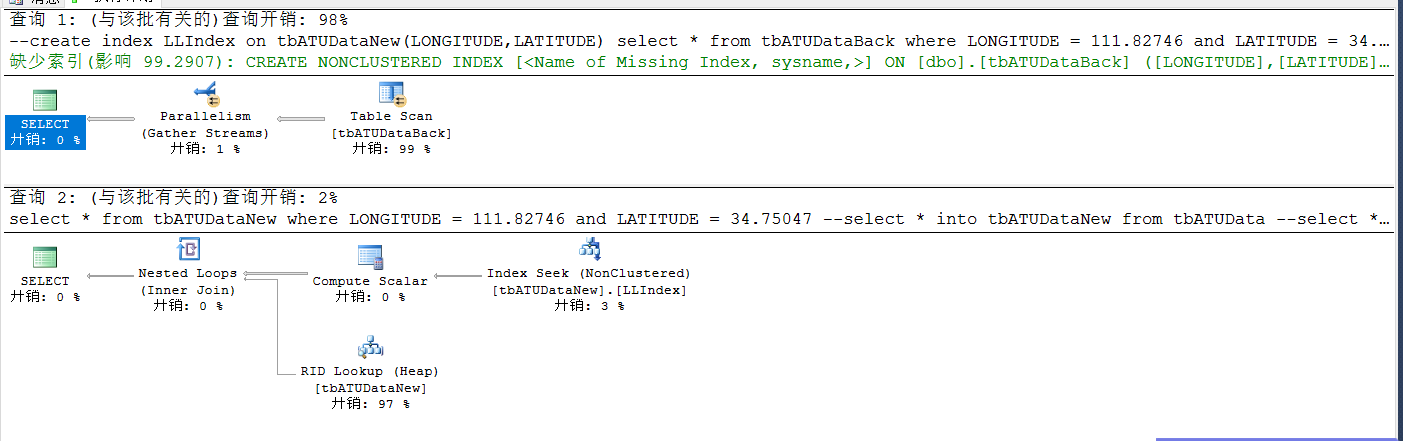
比较有无索引的时间开销

无索引

select \* from tbATUDataBack where LONGITUDE = 111.82746 and LATITUDE = 34.75047

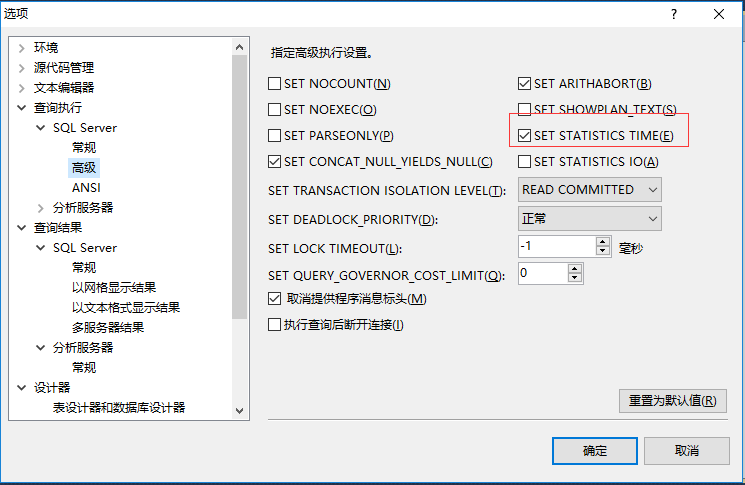
有索引

select \* from tbATUDataNew where LONGITUDE = 111.82746 and LATITUDE = 34.75047

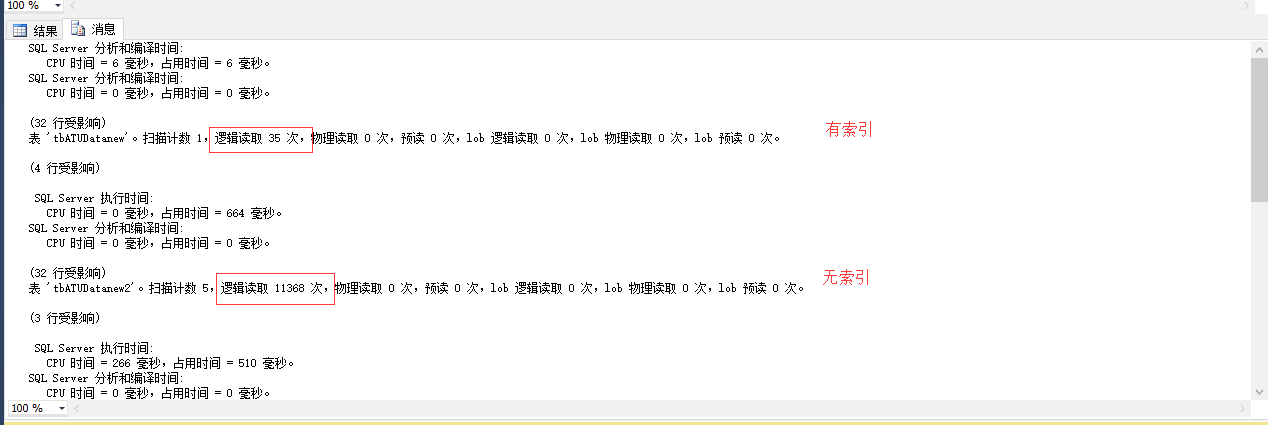
无索引和有索引的执行计划比较

发现只有执行的开销，没有执行时间，设置查看执行时间。

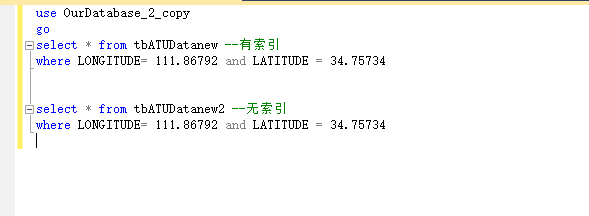
工具->选项->查询执行 如下图

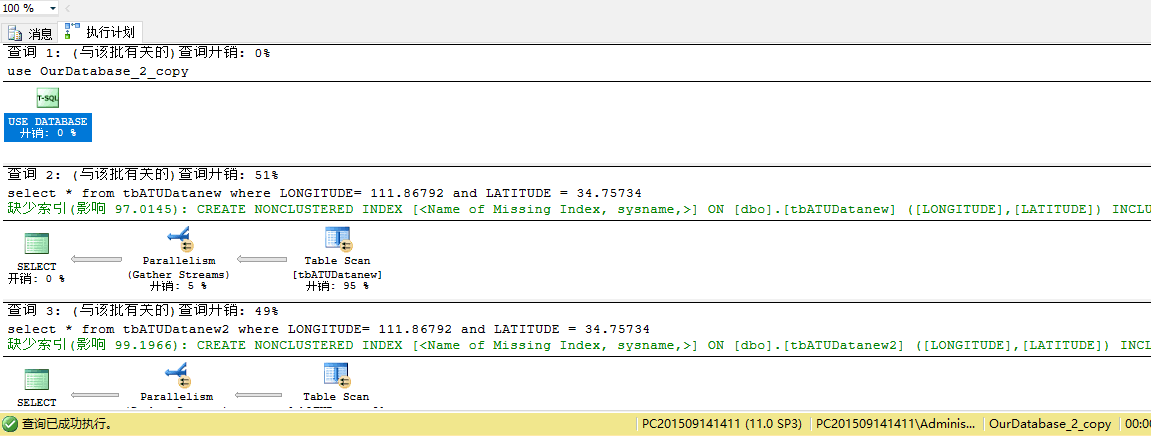


执行时间比较：



在此之前，用了以下这组数据查询，发现是50%和50%，修改数据后成功，猜想是因为此数据在表中位置靠前，索引的作用不是很明显。





**第5步 编写select语句，强制使用索引访问tbATUDatanew；编写select语句，强制不使用索引方式，访问tbATUData，将2条select语句，同时提交给DBMS执行后，从2个执行结果窗口中，判断执行的结果是否一致，并观察各自执行效果、查询执行计划、时间对比。**

强制使用

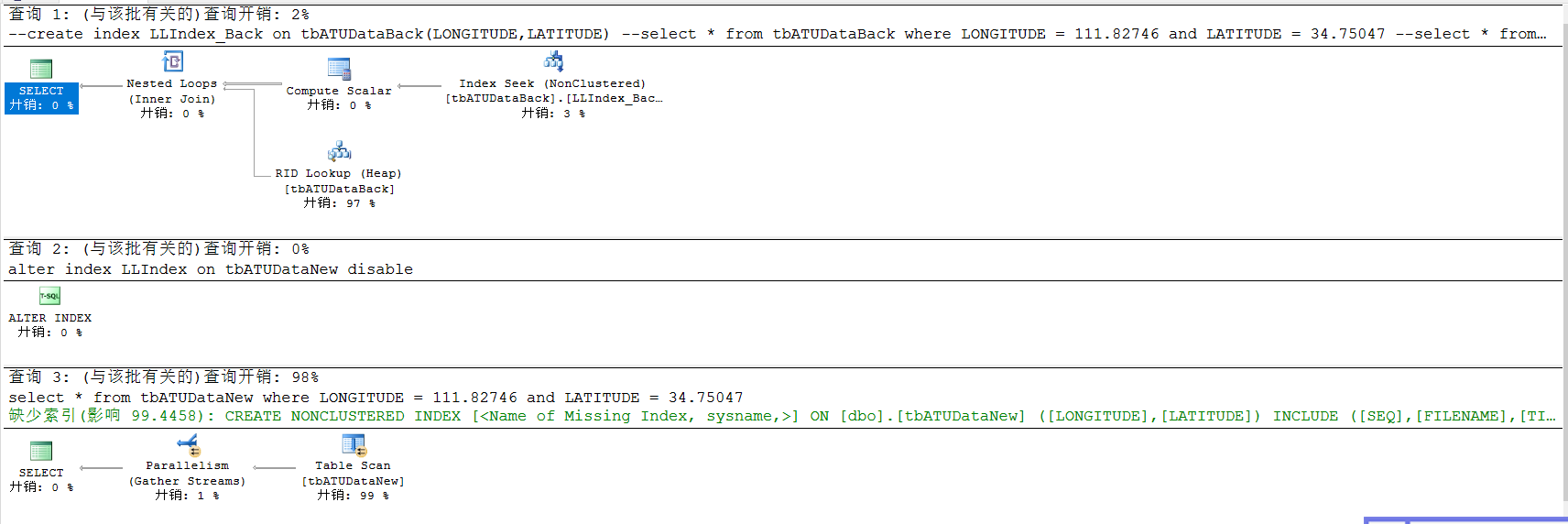
select \* from tbATUDataBack with(index = LLIndex\_Back)

where LONGITUDE = 111.82746 and LATITUDE = 34.75047

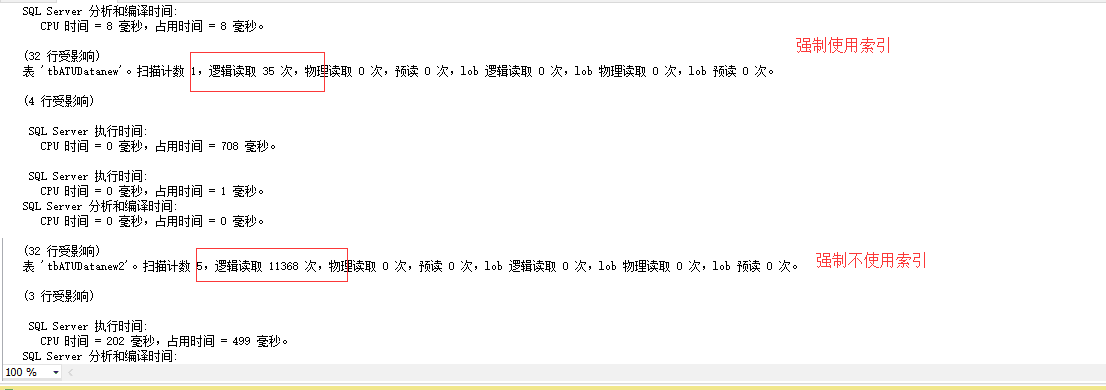
强制不使用

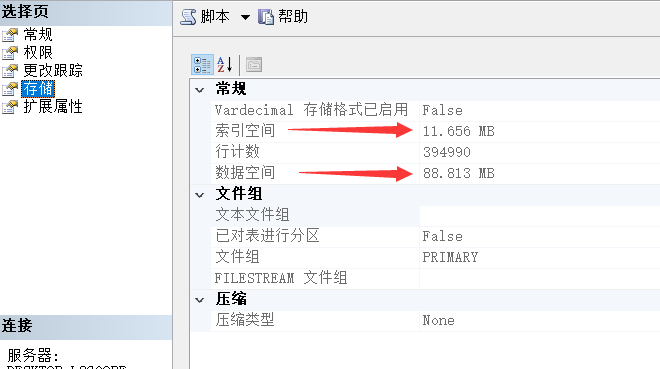
alter index LLIndex on tbATUDataNew disable

select \* from tbATUDataNew

where LONGITUDE = 111.82746 and LATITUDE = 34.75047

执行时间比较



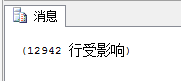
**第6步 查看表属性里面的索引和数据存储空间。**

**实验2.2 数据库索引的设计及对insert/update/delete操作的影响分析**

**第1步 创建tbAdjCell的备份表tbAdjCellNew；**

select\*into tbCellnew

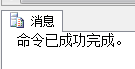
fromtbCell



**第2步 在tbAdjCellNew创建索引。**

索引属性建议：可以在S\_SECTOR\_ID或N\_SECTOR\_ID上设计索引（见表2.2）。

createindexIDontbCellnew(SECTOR\_ID)



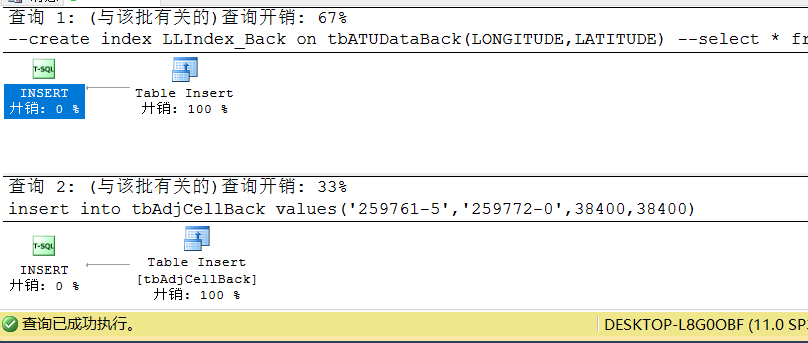
**第3步 在tbATUDatanew编写insert语句，添加一条元组数据（包含索引）；在tbATUData编写同样的一条insert语句，将2条语句同时提交给DBMS执行后，从2个执行结果窗口中，判断执行的结果是否一致，并观察各自执行效果、查询执行计划、时间对比。**

有索引

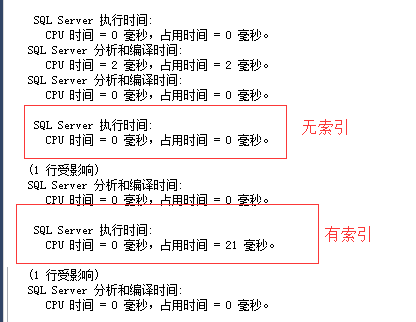
insert into tbAdjCellNew values('259772-0','259761-5',38400,38400)

无索引

insert into tbAdjCellBack values('259761-5','259772-0',38400,38400)



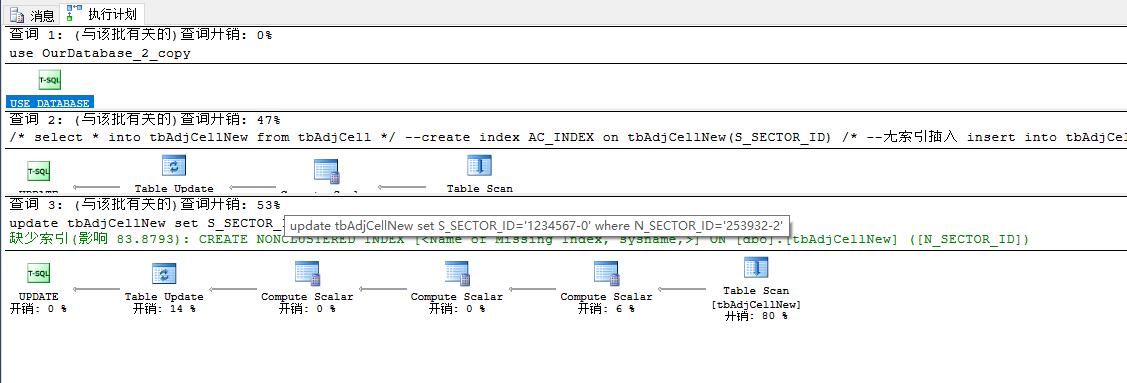
执行时间比较



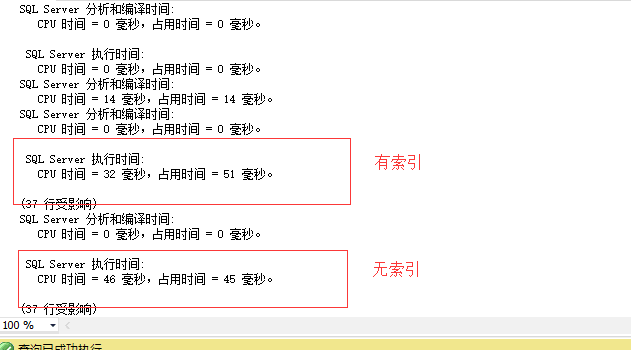
**第4步 在tbATUDatanew编写update语句，修改一条元组数据（包含索引）；在tbATUData编写同样的一条update语句，将2条语句同时提交给DBMS执行后，从2个执行结果窗口中，判断执行的结果是否一致，并观察各自执行效果、查询执行计划、时间对比。**

修改索引：

--无索引  
update tbAdjCellNew2   
set S\_SECTOR\_ID='1234567-0'  
where N\_SECTOR\_ID='253932-2'  
  
--有索引  
update tbAdjCellNew  
set S\_SECTOR\_ID='1234567-0'  
where N\_SECTOR\_ID='253932-2'



执行时间比较



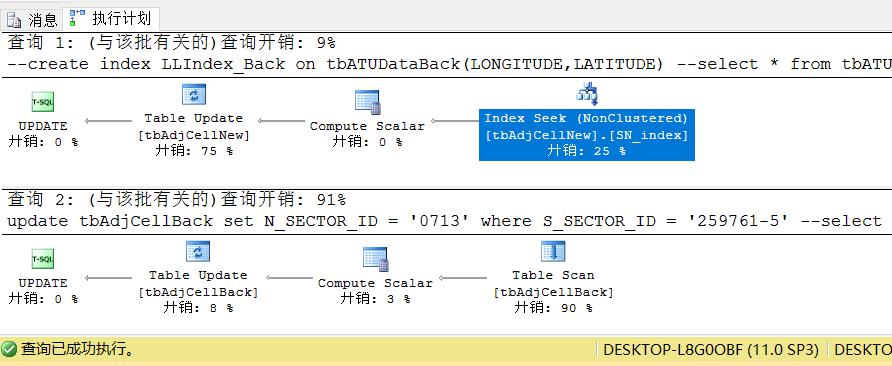
不修改索引：

无索引修改

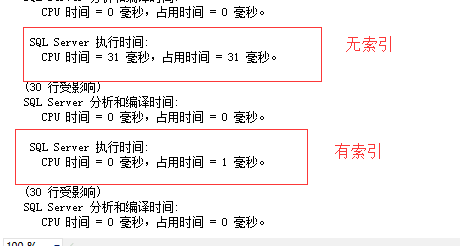
update tbAdjCellNew set N\_SECTOR\_ID = '0713' where S\_SECTOR\_ID = '259772-0'

有索引修改

update tbAdjCellBack set N\_SECTOR\_ID = '0713' where S\_SECTOR\_ID = '259761-5'



执行时间比较



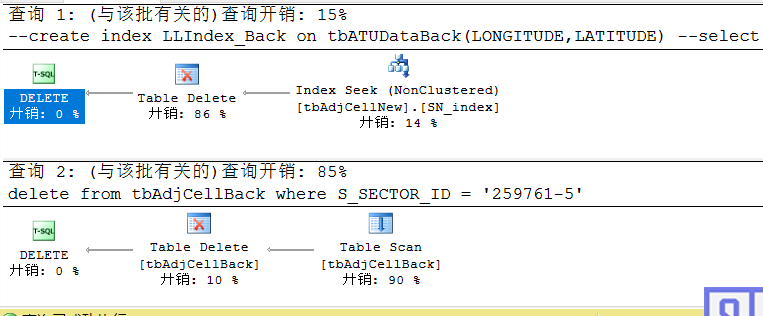
**第5步 在tbATUDatanew编写delete语句，删除第三步添加的数据；在tbATUData编写同样的一条delete语句，将2条语句同时提交给DBMS执行后，从2个执行结果窗口中，判断执行的结果是否一致，并观察各自执行效果、查询执行计划、时间对比。**

无索引查找删除

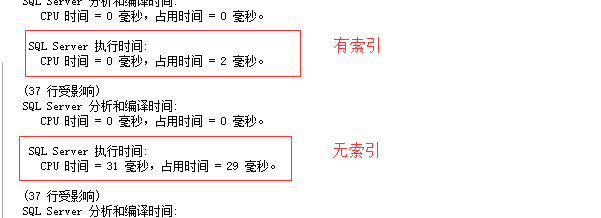
delete from tbAdjCellNew where S\_SECTOR\_ID = '259772-0'

按索引查找删除

delete from tbAdjCellBack where S\_SECTOR\_ID = '259761-5'



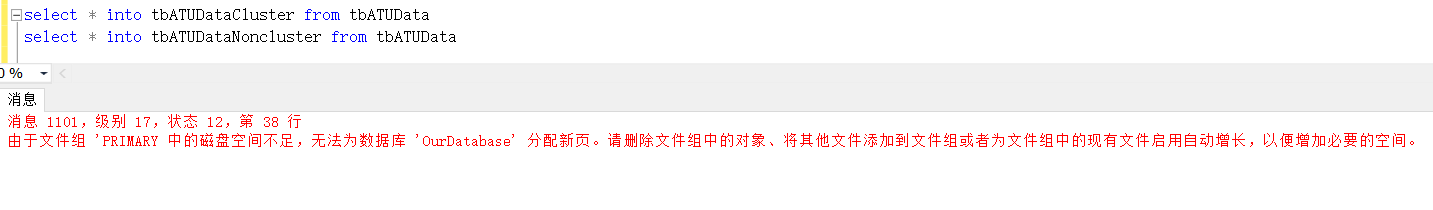
执行时间比较

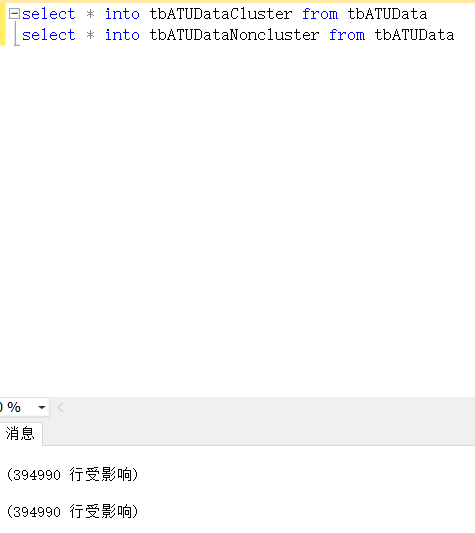


**实验2.3 聚集索引和非聚集索引的比较**

**第1步 创建tbATUData的备份表tbATUDataNew；**

此时遇到了一个问题，提示磁盘空间不足，之后修改了数据库的上限，便可成功的添加新表。





**第2步 在tbATUDatanew创建聚集索引。**

索引属性建议：可以在FileName或CellID 上设计聚集索引，在Longitude和Latitude上设计非聚集索引。

创建聚集索引：

create clustered index ClusterIndex on tbATUDataCluster(CELLID)

执行比较：

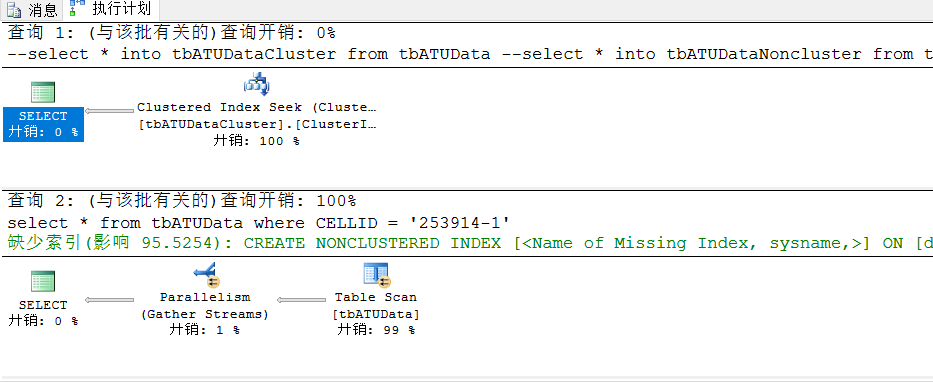
有索引

select \* from tbATUDataCluster where CELLID = '253914-1'

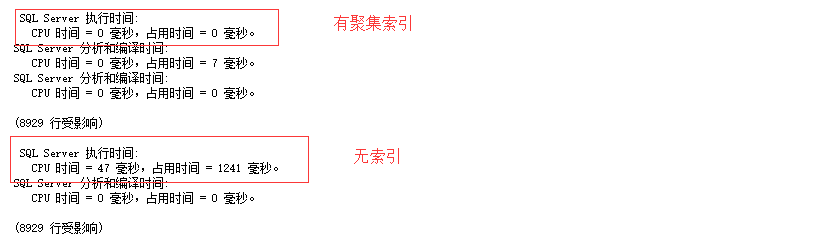
无索引

select \* from tbATUData where CELLID = '253914-1'

有无索引的执行计划比较



执行时间比较：



**第3步 在tbATUData创建非聚集索引。**

在tbATUDataNonCluster创建非聚集索引

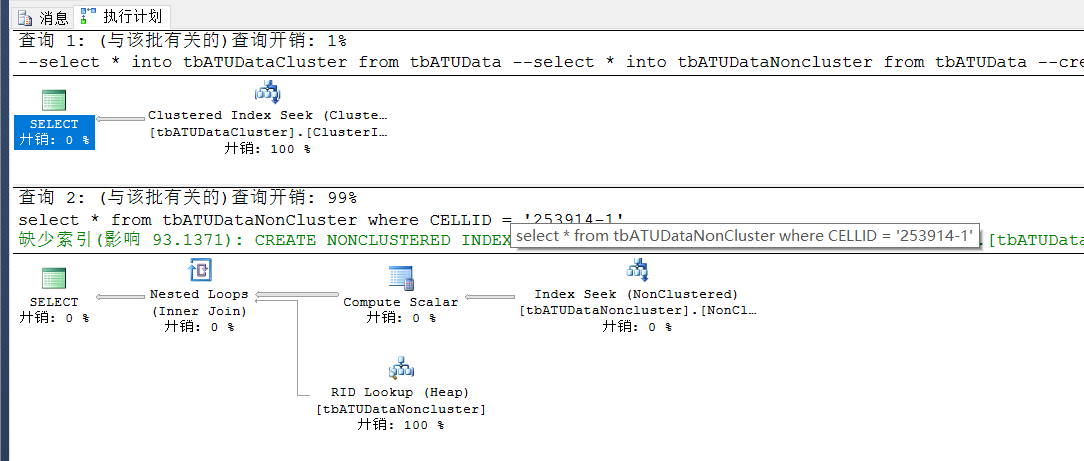
create index NonClusterIndex on tbATUDataNonCluster(CELLID)

聚集索引

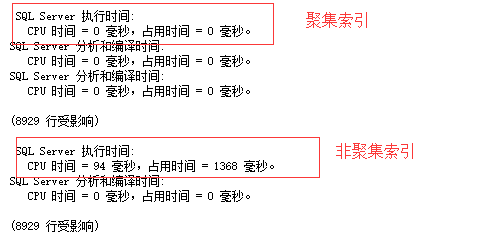
select \* from tbATUDataCluster where CELLID = '253914-1'

非聚集索引

select \* from tbATUDataNonCluster where CELLID = '253914-1'

执行开销比较

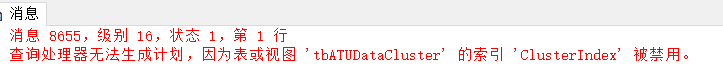
执行时间比较



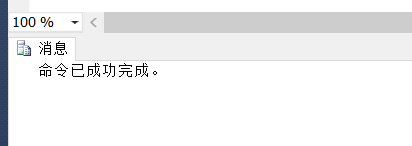
**第4步 在tbATUDatanew强制不使用聚集索引，写一条查询语句观察执行结果。**

alter index ClusterIndex on tbATUDataCluster disable

select \* from tbATUDataCluster where CELLID = '253914-1'



结果显示强制不使用聚集索引会使得表不能访问，这时可以重建聚集索引

alter index ClusterIndex on tbATUDataCluster rebuild

### 实验总结

在此实验中，遇到了以下几个问题：

问题1. 执行计划中只能看到执行开销，却无法看到执行时间。在工具->选项中设置SET STATICS TIME，可以在执行时看到执行时间。

问题2. 执行计划的开销与实验讲义中的示例不符，猜想是因为此数据在表中位置靠前，索引的作用不是很明显。后来得知有无索引的开销和参与比较的表的数据量和数据的选值都有关。

问题3. 最后新加表时，提示磁盘空间不足，原来设置的增长限制最大为500MB，后来修改设置为增长无限制，就能成功添加表。

在此实验中，我们通过实验结果清楚的认识到了索引在增删改查中起到的作用，以及聚集索引和非聚集索引的作用效果的差别。同时，索引并不是什么时候都起到优化的作用，在查询结果和数据总量相当时，索引还会降低效率，所以本份实验报告的测验结果也不完全代表了索引的作用，只是和理论值相似。