实验10 数据库实验

学号	19335162	姓名	潘思晗

一、实验目的

简单了解使用数据库提供的命令查看执行计划。执行计划（execution plan，也叫查询计划或者解释计划）是数据库执行 SQL 语句的具体步骤。

二、实验内容及结果

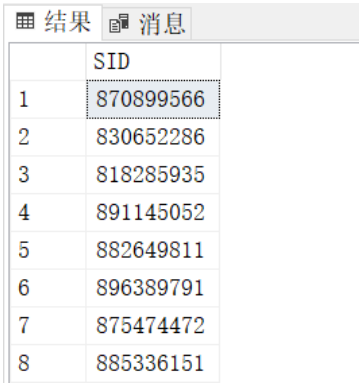
1.使用数据库提供的命令查看执行计划

（1）

SQL语句如下：

```
use school
SET Statistics PROFILE on;
select CHOICES.SID from CHOICES, COURSES where
CHOICES.CID=COURSES.CID and COURSES.CNAME='database';
SET Statistics PROFILE off;
```

点击【执行】之后，查看结果如下：



	SID
1	870899566
2	830652286
3	818285935
4	891145052
5	882649811
6	896389791
7	875474472
8	885336151

	Rows	Executes	StmtText	StmtId	NodeId	Parent	PhysicalOp	LogicalOp	Argument	DefinedValues
1	5898	1	select CHOICES.SID from CHOICES, COURSES where ...	1	1	0	NULL	NULL	NULL	NULL
2	5898	1	|--Hash Match(Inner Join, HASH:([school].[dbo...	1	2	1	Hash Match	Inner ...	HASH:...	[Opt_Bitma...
3	1	1	|--Clustered Index Scan(OBJECT:([school]...	1	3	2	Cluster...	Cluste...	OBJEC...	[school].[...
4	5898	1	|--Filter(WHERE:(PROBE([Opt_Bitmap1004],...	1	4	2	Filter	Filter	WHERE...	NULL
5	2...	1	|--Clustered Index Scan(OBJECT:([sc...	1	5	4	Cluster...	Cluste...	OBJEC...	[school].[...

EstimateRows	EstimateIO	EstimateCPU	AvgRowSize	TotalSubtreeCost	OutputList	Warnings	Type	Parallel	EstimateExecutions
5991	NULL	NULL	NULL	1.643771	NULL	NULL	SELECT	0	NULL
5991	0	0.01828968	17	1.643771	[school...	NULL	PLA...	0	1
1	0.003125	0.000212	47	0.003337	[school...	NULL	PLA...	0	1
2995.5	1.292755	0.3293625	27	1.622117	[school...	NULL	PLA...	0	1
2995.5	1.292755	0.3293625	27	1.622117	[school...	NULL	PLA...	0	1

返回的是CHOICES的SID数据，下方执行计划有多少行表明SQL Server执行了多少个步骤，这里有6行，表明SQLSRVER执行了5个步骤。

参数分析：

Rows: 执行计划的每一步返回的实际行数

Executes: 执行计划的每一步被运行了多少次

StmtText: 执行计划的具体内容。执行计划以一棵树的形式显示。每一行都是运行的一步，都会有结果集返回，也都会有自己的cost

EstimateRows: SQLSERVER根据表格上的统计信息，预估的每一步的返回行数。在分析执行计划时

将Rows和EstimateRows这两列做对比，可以确认SQL Server预估是否正确，以判断统计信息是否有更新，此处对比可知预估准确。

EstimateIO: SQLSERVER根据EstimateRows和统计信息里记录的字段长度，预估的每一步会产生的I/O cost

EstimateCPU: SQLSERVR根据EstimateRows和统计信息里记录的字段长度，以及要做的事情的复杂度，预估每一步会产生的CPU cost

TotalSubtreeCost: SQLSERVER根据EstimateIO和EstimateCPU通过某种计算公式，计算出每一步执行计划子树的cost

Warnings: SQLSERVER在运行每一步时遇到的警告，例如，某一步没有统计信息支持cost预估等。

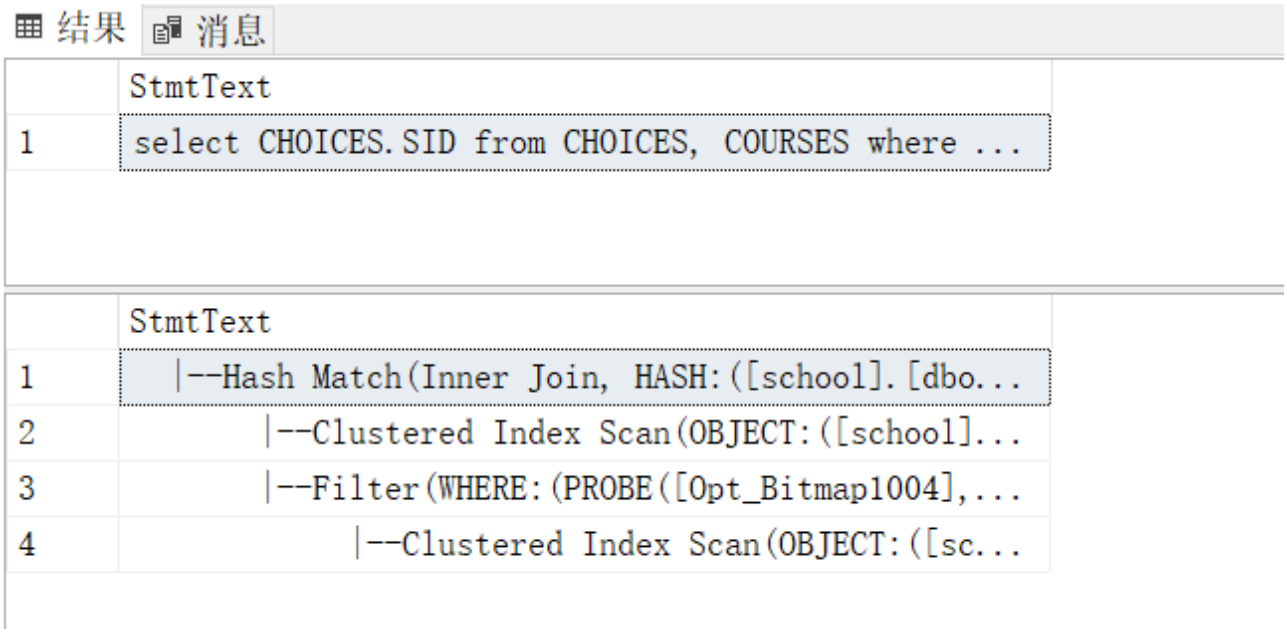
Parallel: 执行计划的这一步是不是使用了并行的执行计划

（2）

SQL语句如下：

```
use school;
Go
set showplan_text on;
GO
select CHOICES.SID from CHOICES, COURSES where
CHOICES.CID=COURSES.CID and COURSES.CNAME='database';
Go
set showplan_text off;
GO
```

输入上述语句，执行之后，可以查询计划如下：



set showplan_text展现执行计划的具体内容，将信息作为一组行返回，这些行形成一个分层树，表示 SQL Server 查询处理器在执行每条语句时所执行的步骤。输出中反映的每个语句都包含一行语句的文本，后跟几行执行步骤的详细信息。该表显示输出包含的列。

第4、第2行：

|--Clustered Index Scan(OBJECT:([school].[dbo].[CHOICES].[PK**CHOICES**3213D0806D12A6DF]))

|--Clustered Index Scan(OBJECT:([school].[dbo].[COURSES].[PK**COURSES**D837D05F569EC771]), WHERE:([school].[dbo].[COURSES].[cname]='database'))

表示是在聚集索引的数据上根据键值取数据。

第1行：

|--Hash Match(Inner Join, HASH:([school].[dbo].[COURSES].[cid])=([school].[dbo].[CHOICES].[cid]), RESIDUAL:([school].[dbo].[CHOICES].[cid]=[school].[dbo].[COURSES].[cid])DEFINE:([Opt_Bitmap1004]))

将之前几步得到的结果集做一个join。因为SQL通过预估知道这两个结果集比较大，所以直接选择了Hash Match的join方法。

（3）

SQL语句如下：

```
use school;
Go
SET SHOWPLAN_ALL ON;
GO
select CHOICES.SID from CHOICES, COURSES where CHOICES.CID=COURSES.CID and COURSES.CNAME='database';
Go
SET SHOWPLAN_ALL OFF;
GO
```

输入上述语句，执行之后，可以看到执行计划显示如下：

结果 消息

	StmtText	StmtId	NodeId	Parent	PhysicalOp	LogicalOp
1	select CHOICES.SID from CHOICES, COURSES where ...	1	1	0	NULL	NULL
2	|--Hash Match(Inner Join, HASH:([school].[dbo...	1	2	1	Hash Match	Inner Join
3	|--Clustered Index Scan(OBJECT:([school]...	1	3	2	Clustered Index Scan	Clustered Index Scan
4	|--Filter(WHERE:(PROBE([Opt_Bitmap1004],...	1	4	2	Filter	Filter
5	|--Clustered Index Scan(OBJECT:([sc...	1	5	4	Clustered Index Scan	Clustered Index Scan

Argument	DefinedValues	EstimateRows	EstimateIO	EstimateCPU	AvgRowSize
1	NULL	5991	NULL	NULL	NULL
HASH:([school].[dbo].[COURSES].[cid])=([school]...	[Opt_Bitmap1004]	5991	0	0.01828968	17
OBJECT:([school].[dbo].[COURSES].[PK__COURSES__...	[school].[dbo].[COURSES].[cid]	1	0.003125	0.000212	47
WHERE:(PROBE([Opt_Bitmap1004],[school].[dbo].[C...	NULL	2995.5	1.292755	0.3293625	27
OBJECT:([school].[dbo].[CHOICES].[PK__CHOICES__...	[school].[dbo].[CHOICES].[sid], [school].[dbo]....	2995.5	1.292755	0.3293625	27

AvgRowSize	TotalSubtreeCost	OutputList	Warnings	Type	Parallel	EstimateExecutions
NULL	1.643771	NULL	NULL	SELECT	0	NULL
17	1.643771	[school].[dbo].[CHOICES].[sid]	NULL	PLAN_ROW	0	1
47	0.003337	[school].[dbo].[COURSES].[cid]	NULL	PLAN_ROW	0	1
27	1.622117	[school].[dbo].[CHOICES].[sid], [school].[dbo]....	NULL	PLAN_ROW	0	1
27	1.622117	[school].[dbo].[CHOICES].[sid], [school].[dbo]....	NULL	PLAN_ROW	0	1

结果与预期一致。

2.利用索引，优化查询性能

对COURSES.CNAME建立索引，查看两种执行计划有何异同。

SQL语句如下：

```
use school
CREATE NONCLUSTERED INDEX INDEX_CNAME on COURSES(CNAME)
SET Statistics PROFILE on;
select CHOICES.SID from CHOICES, COURSES where
CHOICES.CID=COURSES.CID and COURSES.CNAME='database';
SET Statistics PROFILE off;
```

由于之前系统自动为COURSES主键建立了聚集索引，因此此处创建非聚集索引。

结果如下：

	Rows	Executes	StmtText	StmtId	NodeId	Parent	PhysicalOp	LogicalOp
1	5898	1	select CHOICES.SID from CHOICES, COURSES where ...	1	1	0	NULL	NULL
2	5898	1	|--Hash Match(Inner Join, HASH:([school].[dbo...	1	2	1	Hash Match	Inner ...
3	1	1	|--Index Seek(OBJECT:([school].[dbo].[CO...	1	3	2	Index Seek	Index ...
4	5898	1	|--Filter(WHERE:(PROBE([Opt_Bitmap1004],...	1	4	2	Filter	Filter
5	2...	1	|--Clustered Index Scan(OBJECT:([sc...	1	5	4	Cluster...	Cluste...

查看StmtText表如下：

	StmtText
1	|--Hash Match(Inner Join, HASH:([school].[dbo...
2	|--Index Seek(OBJECT:([school].[dbo].[CO...
3	|--Filter(WHERE:(PROBE([Opt_Bitmap1004],...
4	|--Clustered Index Scan(OBJECT:([sc...

可以看到原本第2行的Clustered Index Scan变成了Index Seek，说明表COURSE在cname这个字段上有一个索引，所以SQL可以直接使用这个索引的seek。

三、实验总结

本次实验主要是了解和使用数据库提供的命令查看执行计划。通过本次实验，我熟悉了使用SQL进行执行计划查询分析的基本过程和语法。在之后的学习过程中还需要继续努力和探索。