

Lecture 3, Fall 2024/2025

数据库系统实验

Yubao Liu (刘玉葆)

School of Data and Computer Science Sun
Yat-sen University

- 本节课提纲

- 实验目的

- 实验内容

- 实验示例

- 练习

- 实验目的

熟悉SQL的数据查询语言，

能够使用SQL语句对数据库进行单表查询、连接查询

- 实验内容

- 1) 单表查询

- 查询的目标表达式为所有列、指定列或指定列的运算。
 - 用 **DISTINCT**保留字消除重复行
 - 对查询结果排序和分组。
 - 集合分组使用集函数进行各项统计

- 2)连接查询

- 笛卡儿连接和等值连接。
 - 自连接
 - 外连接
 - 复合条件连接
 - 多表连接

- 实验示例

以 school数据库为例，在该数据库中存在4张表格，分别为

- students (sid, sname , email , grade)
- teachers (tid, tname , email , salary)
- courses (cid, cname , hour)
- choices (no, sid , tid , cid , score)

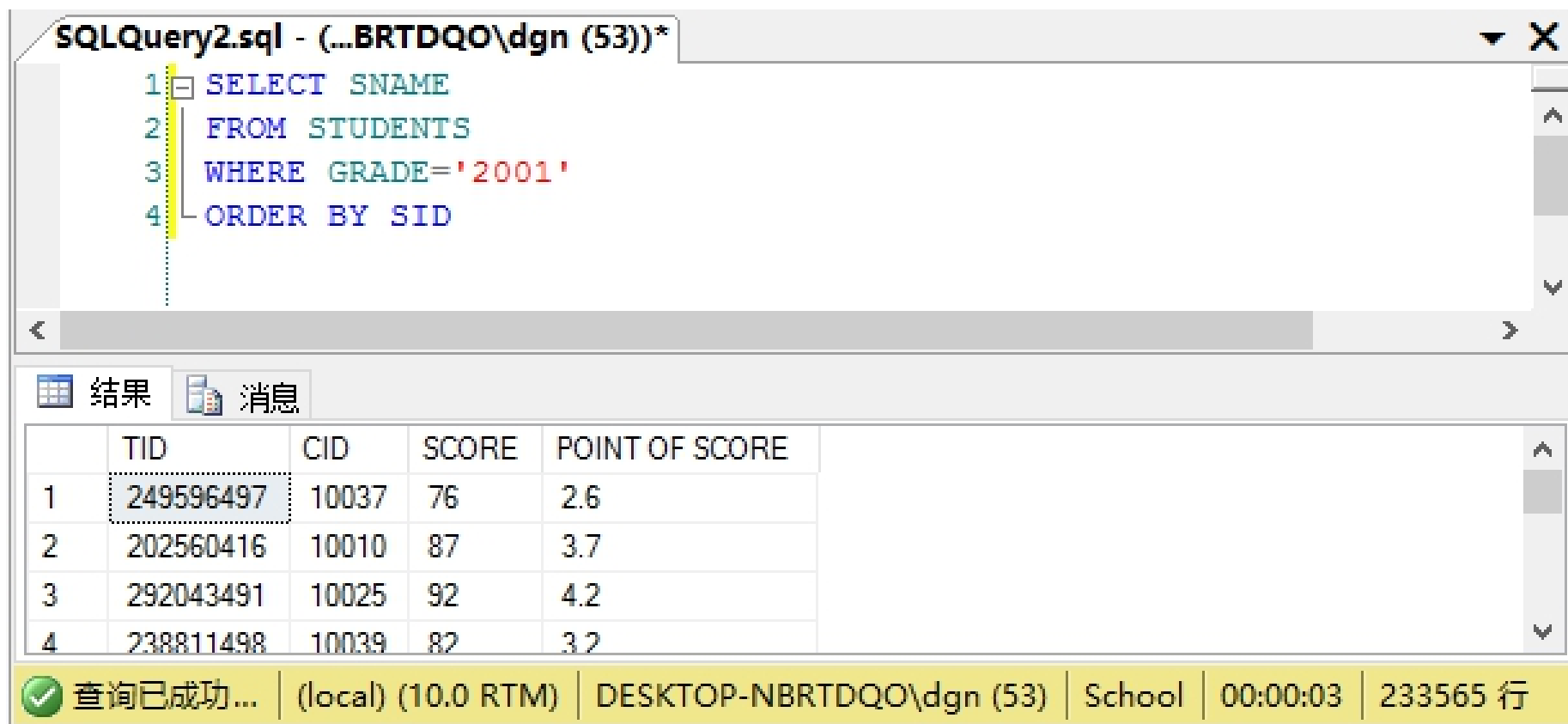
在数据库中，存在这样的关系，学生可以选择课程。一个课程对应一个教师。

在CHOICES表中保存学生的选课记录。

• 实验示例

一. 单表查询

1. 查询年级为2001所有学生的名称，按编号升序排列。



The screenshot shows a SQL query window titled "SQLQuery2.sql - (...BRTDQO\dgn (53))*". The query text is:

```
1 SELECT SNAME
2 FROM STUDENTS
3 WHERE GRADE='2001'
4 ORDER BY SID
```

Below the query editor, there are two tabs: "结果" (Results) and "消息" (Messages). The "结果" tab is active, displaying a table with the following data:

	TID	CID	SCORE	POINT OF SCORE
1	249596497	10037	76	2.6
2	202560416	10010	87	3.7
3	292043491	10025	92	4.2
4	238811498	10039	82	3.2

At the bottom of the window, a status bar displays the following information: "查询已成功..." (Query successful...), "(local) (10.0 RTM)", "DESKTOP-NBRTDQO\dgn (53)", "School", "00:00:03", and "233565 行" (233565 rows).

实验示例

2. 查询学生的选课成绩合格的课程成绩，并把成绩换算为积点。（60分对应积点1，每增加1分，积点增加0.1）

SQLQuery2.sql - (...BRTDQO\dgn (53))*

```
1 SELECT TID,CID,SCORE, (1+(SCORE-60)*0.1) as 'POINT OF SCORE'
2 FROM CHOICES
3 WHERE SCORE>60
```

结果 消息

	TID	CID	SCORE	POINT OF SCORE	别名
1	249596497	10037	76	2.6	
2	202560416	10010	87	3.7	
3	292043491	10025	92	4.2	
4	238811498	10039	82	3.2	

查询已成功... (local) (10.0 RTM) DESKTOP-NBRTDQO\dgn (53) School 00:00:03 233565

• 相当于给‘积点’取了个别名。

• 原本表中无定义“积点”这个属性，但可以通过score属性计算而得。

- 实验示例

3. 查询课时是48或者64的课程名称

(注：可使用 in 或者 or 语句)

SQLQuery2.sql - (...BRTDQO\dgn (53))*

```
1 SELECT CNAME
2 FROM COURSES
3 WHERE HOUR IN ('48', '64')
```

<

结果 消息

	CNAME
1	computer graphics
2	java
3	design pattern
4	real-time system

✓ 查询已成功执行。 | (local) (10.0 RTM) | DESKTOP-

SQLQuery2.sql - (...BRTDQO\dgn (53))*

```
1 SELECT CNAME
2 FROM COURSES
3 WHERE (HOUR='48') or (HOUR = '64')
```

<

结果 消息

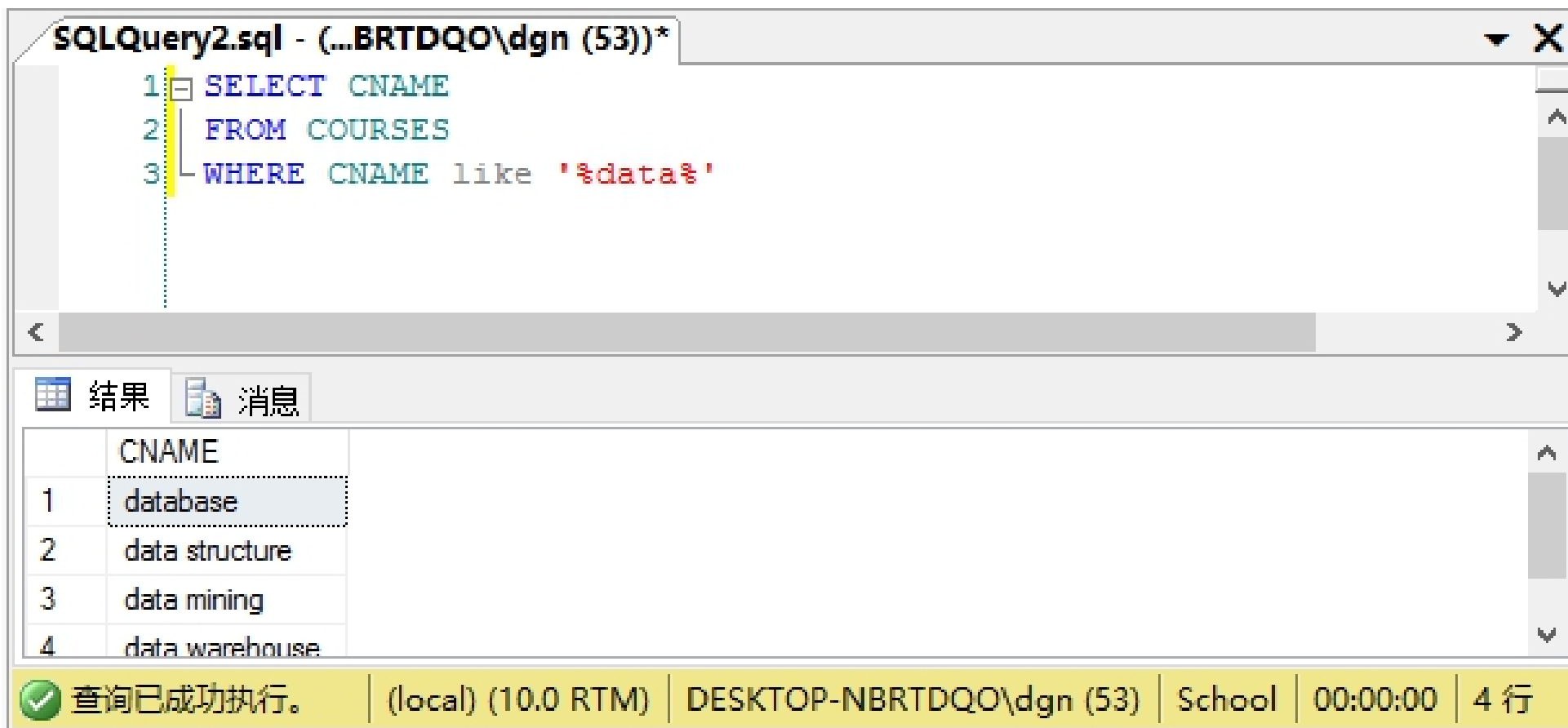
	CNAME
1	computer graphics
2	java
3	design pattern
4	real-time system

✓ 查询已成功执行。 | (local) (10.0 RTM) | DESKTOP-NBR

- 实验示例

- 4. 查询所有课程名称中含有**data**的课程编号

(注：使用模糊查询like)



The screenshot shows a SQL query editor window titled "SQLQuery2.sql - (...BRTDQO\dgn (53))*". The query text is:

```
1 SELECT CNAME
2 FROM COURSES
3 WHERE CNAME like '%data%'
```

Below the query editor, there is a tabbed interface with "结果" (Results) and "消息" (Messages). The "结果" tab is active, displaying a table with the following data:

	CNAME
1	database
2	data structure
3	data mining
4	data warehouse

At the bottom of the window, a status bar indicates: "查询已成功执行。" (Query executed successfully.) | (local) (10.0 RTM) | DESKTOP-NBRTDQO\dgn (53) | School | 00:00:00 | 4 行

- 实验示例

5. 查询所有选课表中的课程号（不重复显示）

（注：使用distinct去重）

SQLQuery2.sql - (...BRTDQO\dgn (53))*

```
1 SELECT CID FROM CHOICES
```

<

结果 消息

	CID
299548	10050
299549	10050
299550	10050

查询已成功... | (local) (10.0 RTM) | DESKTOP-

去重前记录数

SQLQuery2.sql - (...BRTDQO\dgn (53))*

```
1 SELECT DISTINCT CID FROM CHOICES
```

<

结果 消息

	CID
48	10029
49	10047
50	10017

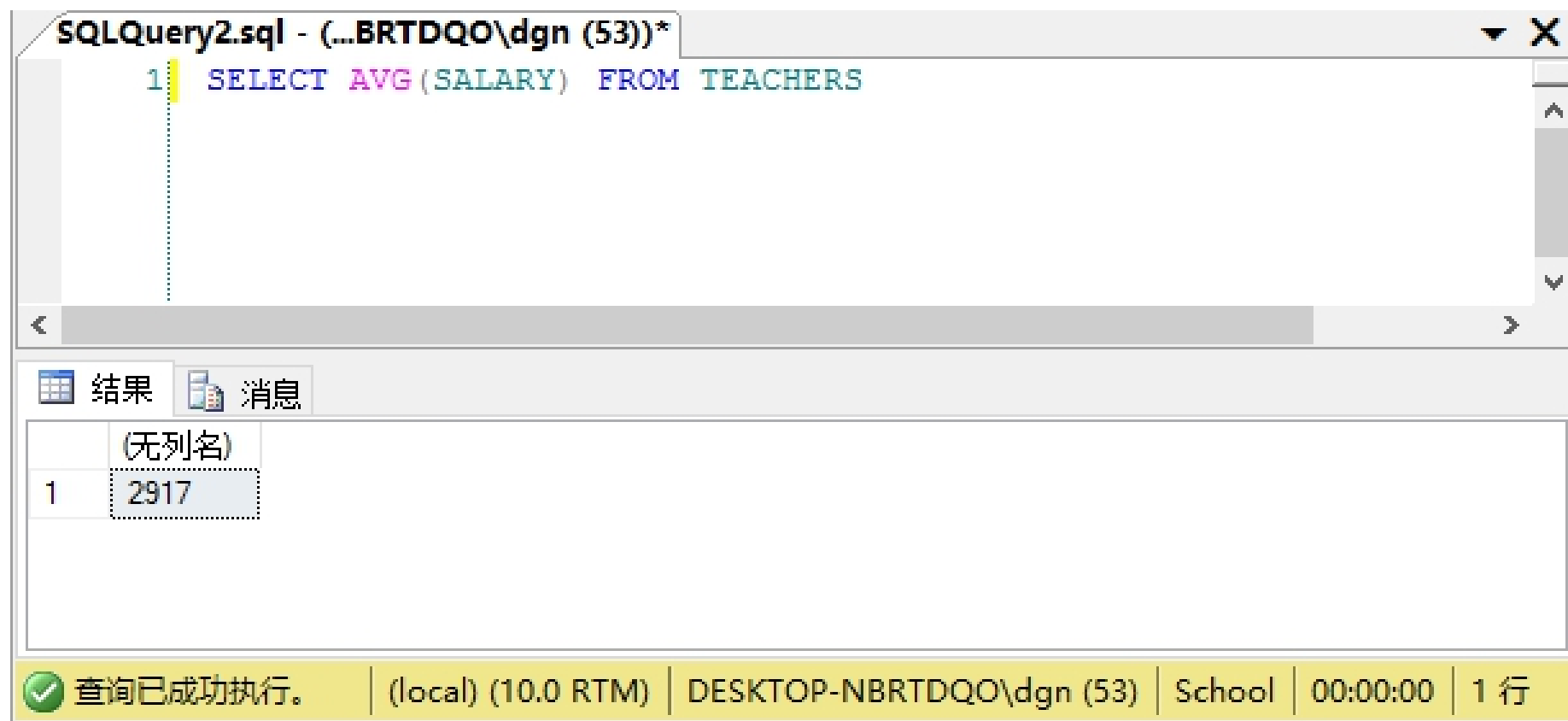
查询已成功执行。 | (local) (10.0 RTM) | DESKTOP-NBRTDQO

去重后记录数

- 实验示例

6.统计所有老师的平均工资

(注：用AVG()函数求平均数)



The screenshot shows a SQL query execution window titled "SQLQuery2.sql - (...BRTDQO\dgn (53))*". The query text is "1 SELECT AVG(SALARY) FROM TEACHERS". Below the query, there are two tabs: "结果" (Results) and "消息" (Messages). The "结果" tab is active, displaying a table with one row and one column. The column header is "(无列名)" (No column name) and the value is "2917". The status bar at the bottom indicates "查询已成功执行。" (Query executed successfully.) and provides additional information: "(local) (10.0 RTM) | DESKTOP-NBRTDQO\dgn (53) | School | 00:00:00 | 1 行" (1 row).

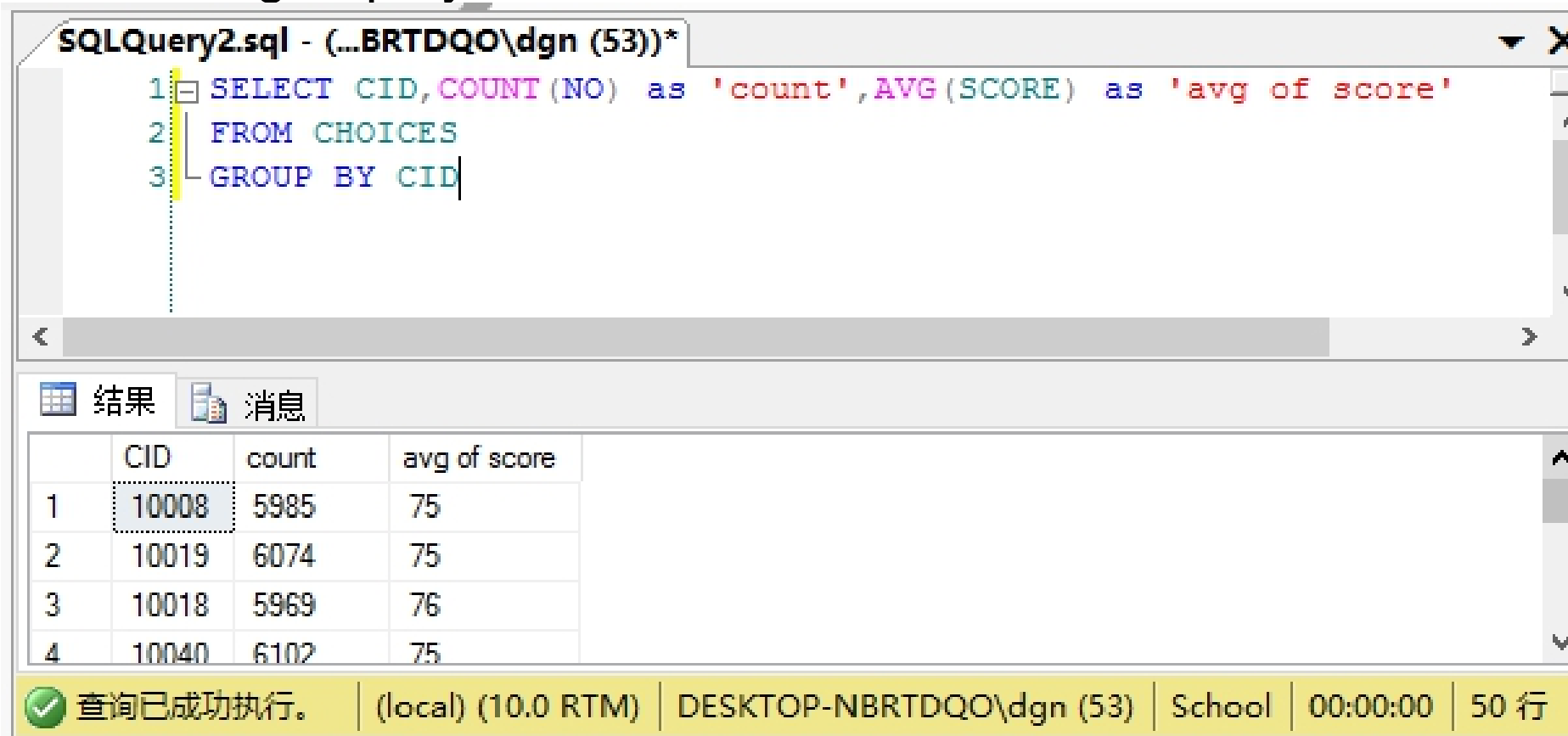
(无列名)
2917

查询已成功执行。 | (local) (10.0 RTM) | DESKTOP-NBRTDQO\dgn (53) | School | 00:00:00 | 1 行

- 实验示例

7.统计每个课程的选课人数和平均成绩

(注：使用group by)



The screenshot displays the SQL Server Enterprise Manager interface. The top pane shows a query window titled "SQLQuery2.sql - (...BRTDQO\dgn (53))*" containing the following SQL code:

```
1 SELECT CID, COUNT(NO) as 'count', AVG(SCORE) as 'avg of score'
2 FROM CHOICES
3 GROUP BY CID
```

The bottom pane shows the "Results" tab with a table of 4 rows and 4 columns. The columns are labeled "CID", "count", and "avg of score". The first row is highlighted with a dashed border.

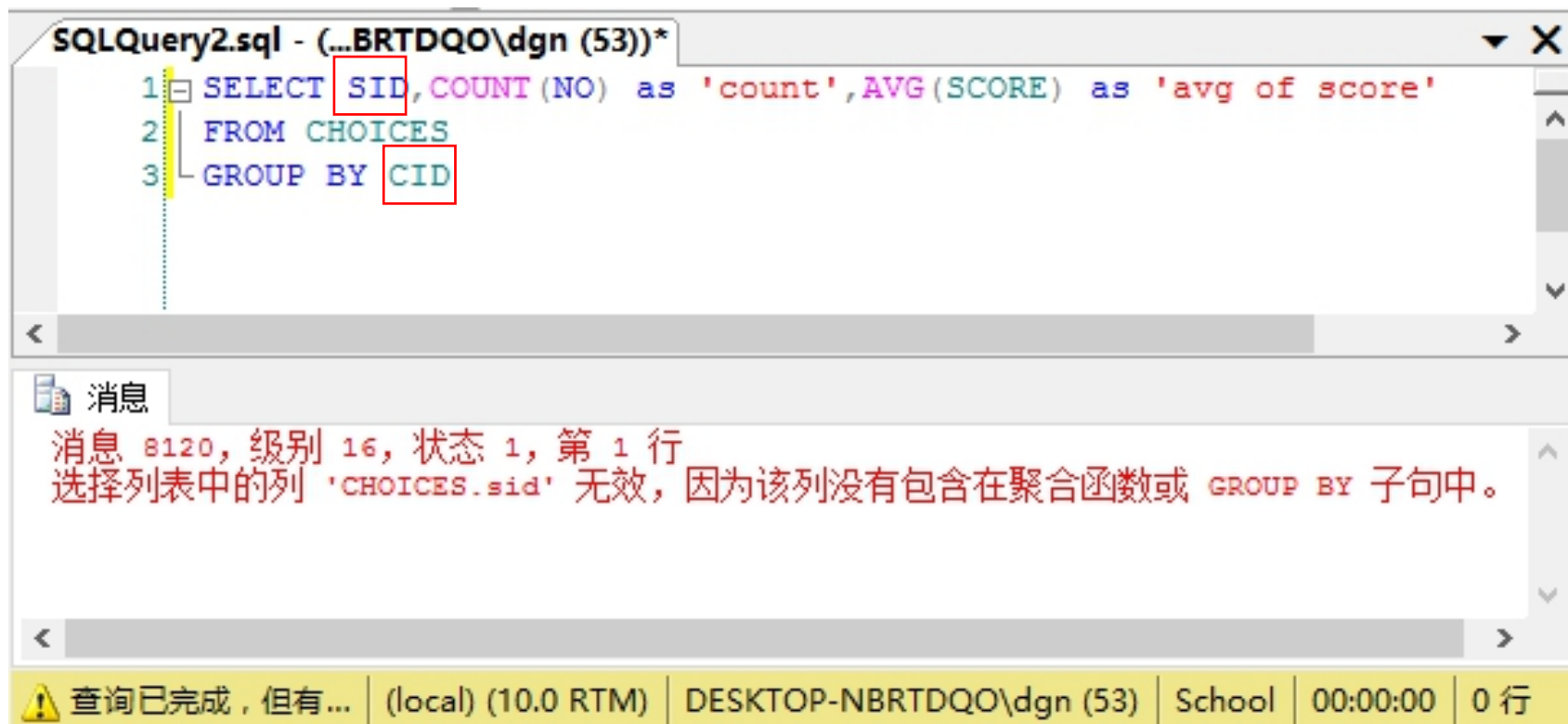
	CID	count	avg of score
1	10008	5985	75
2	10019	6074	75
3	10018	5969	76
4	10040	6102	75

The status bar at the bottom indicates: "查询已成功执行。 (local) (10.0 RTM) DESKTOP-NBRTDQO\dgn (53) School 00:00:00 50 行"

• 实验示例

注意：如果有group by语句，则select语句允许**groupby**字句中出现的字段和集合函数表达式，不允许出现包含其他字段的表达式。

（比如将上题中select语句中 CID 替换成 SID则报错，因为groupby 是CID，select语句中出现SID会报错）



The screenshot shows a SQL query editor window titled "SQLQuery2.sql - (...BRTDQO\dgn (53))*". The query text is:

```
1 SELECT SID, COUNT(NO) as 'count', AVG(SCORE) as 'avg of score'
2 FROM CHOICES
3 GROUP BY CID
```

The words "SID" and "CID" are highlighted with red boxes. Below the query editor, a message pane displays the following error:

消息 8120, 级别 16, 状态 1, 第 1 行
选择列表中的列 'CHOICES.sid' 无效, 因为该列没有包含在聚合函数或 GROUP BY 子句中。

The status bar at the bottom indicates: 查询已完成, 但有... (local) (10.0 RTM) | DESKTOP-NBRTDQO\dgn (53) | School | 00:00:00 | 0 行

• 实验示例

8. 查询至少选修了三门课程的学生编号

(注：分组后还要求按照一定条件对这些组进行筛选，则使用**having**语句指定筛选条件)

SQLQuery2.sql - (...BRTDQO\dgn (53))*

```
1 SELECT SID
2 FROM CHOICES
3 GROUP BY SID
4 HAVING COUNT(*) > 3
```

结果 消息

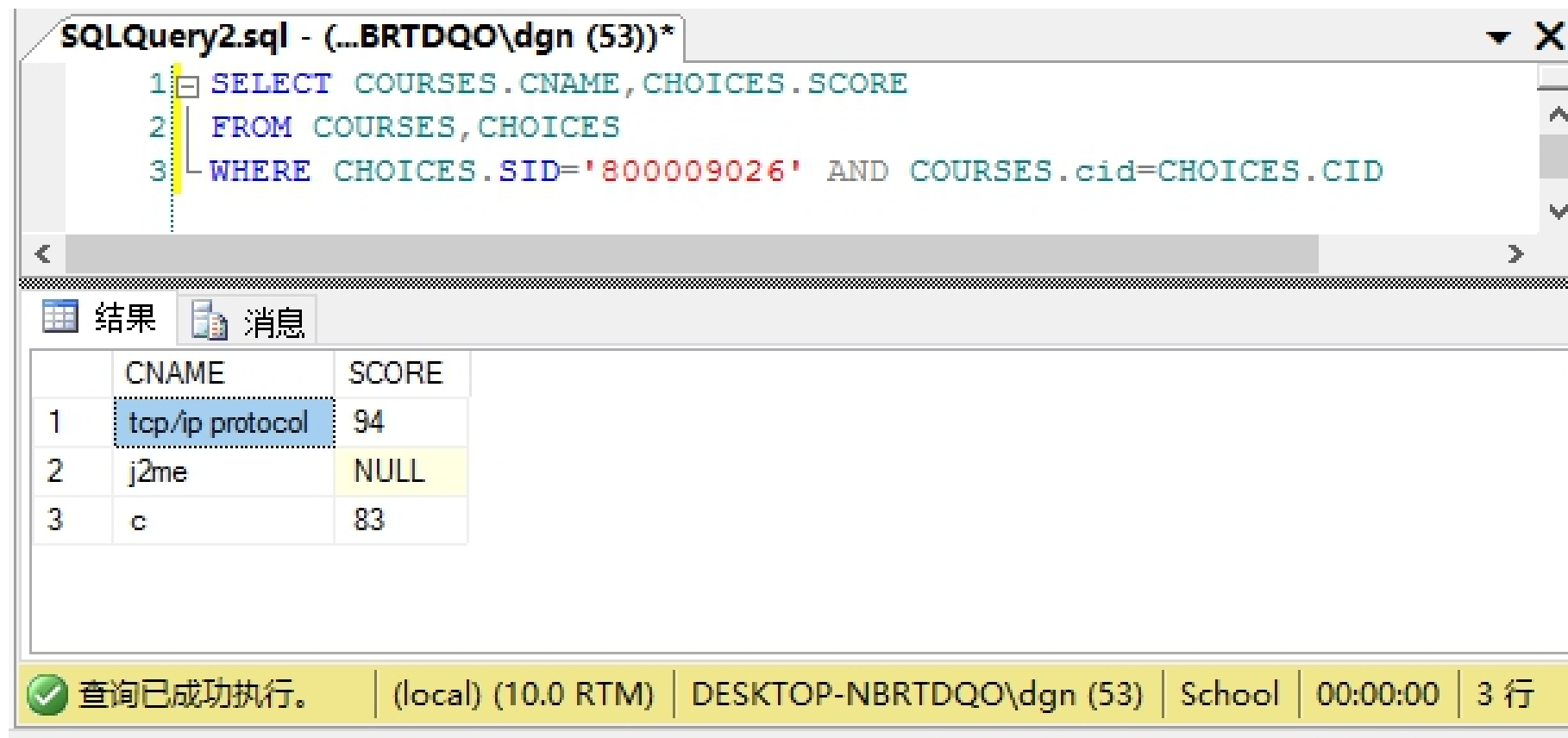
	SID
1	801855166
2	812917218
3	888277410
4	807064377

查询已成功... | (local) (10.0 RTM) | DESKTOP-NBRTDQO\dgn (53) | School | 00:00:00 | 39912 行

• 实验示例

二. 连接查询

1. 查询编号800009026的学生所选的全部课程的课程名和成绩。



The screenshot shows a SQL query window titled "SQLQuery2.sql - (...BRTDQO\dgn (53))*". The query is as follows:

```
1 SELECT COURSES.CNAME, CHOICES.SCORE
2 FROM COURSES, CHOICES
3 WHERE CHOICES.SID='800009026' AND COURSES.cid=CHOICES.CID
```

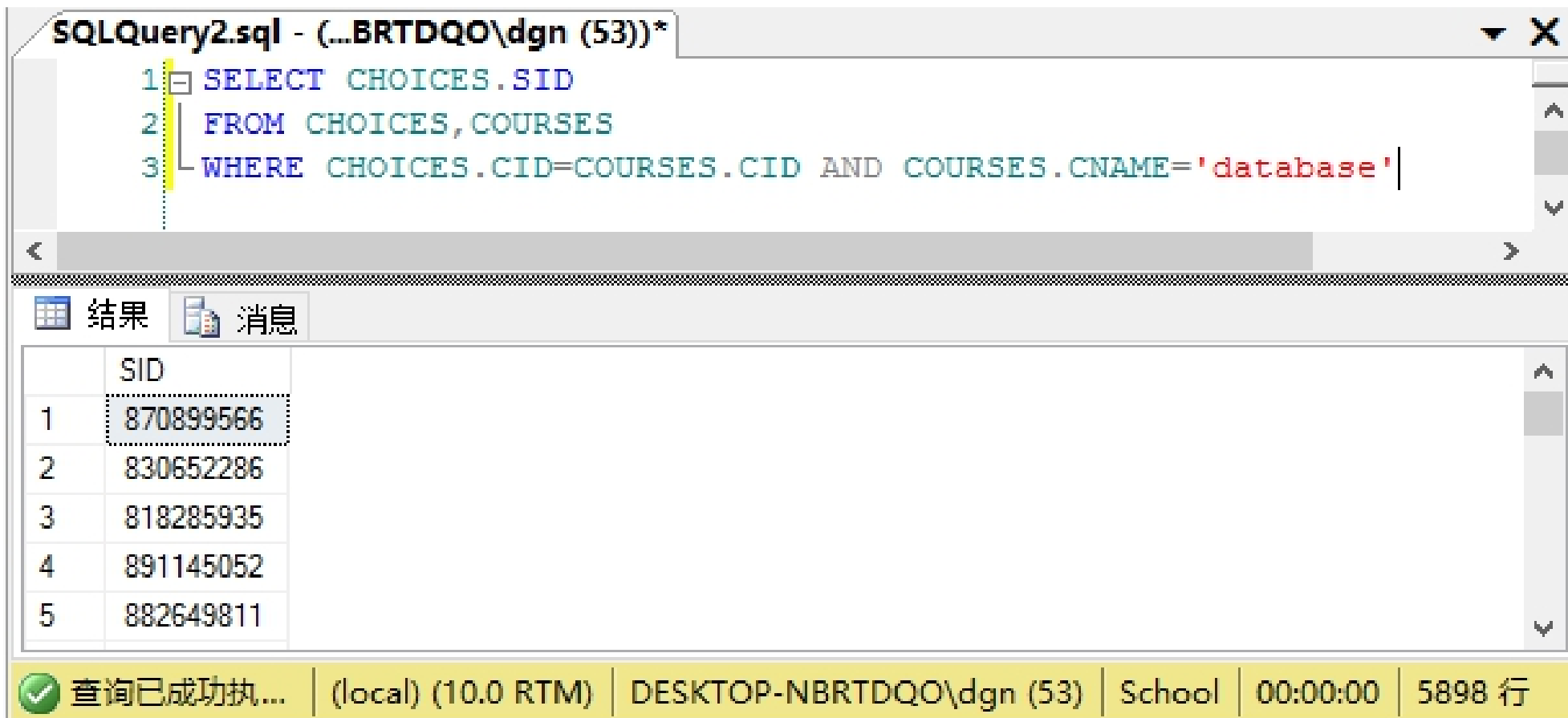
Below the query, there are two tabs: "结果" (Results) and "消息" (Messages). The "结果" tab is active, displaying a table with the following data:

	CNAME	SCORE
1	tcp/ip protocol	94
2	j2me	NULL
3	c	83

At the bottom of the window, a status bar indicates: "查询已成功执行。" (Query executed successfully.) | (local) (10.0 RTM) | DESKTOP-NBRTDQO\dgn (53) | School | 00:00:00 | 3 行

- 实验示例

2.查询所有选择了database课程的学生的编号。



SQLQuery2.sql - (...BRTDQO\dgn (53))*

```
1 SELECT CHOICES.SID
2 FROM CHOICES, COURSES
3 WHERE CHOICES.CID=COURSES.CID AND COURSES.CNAME='database'
```

结果 消息

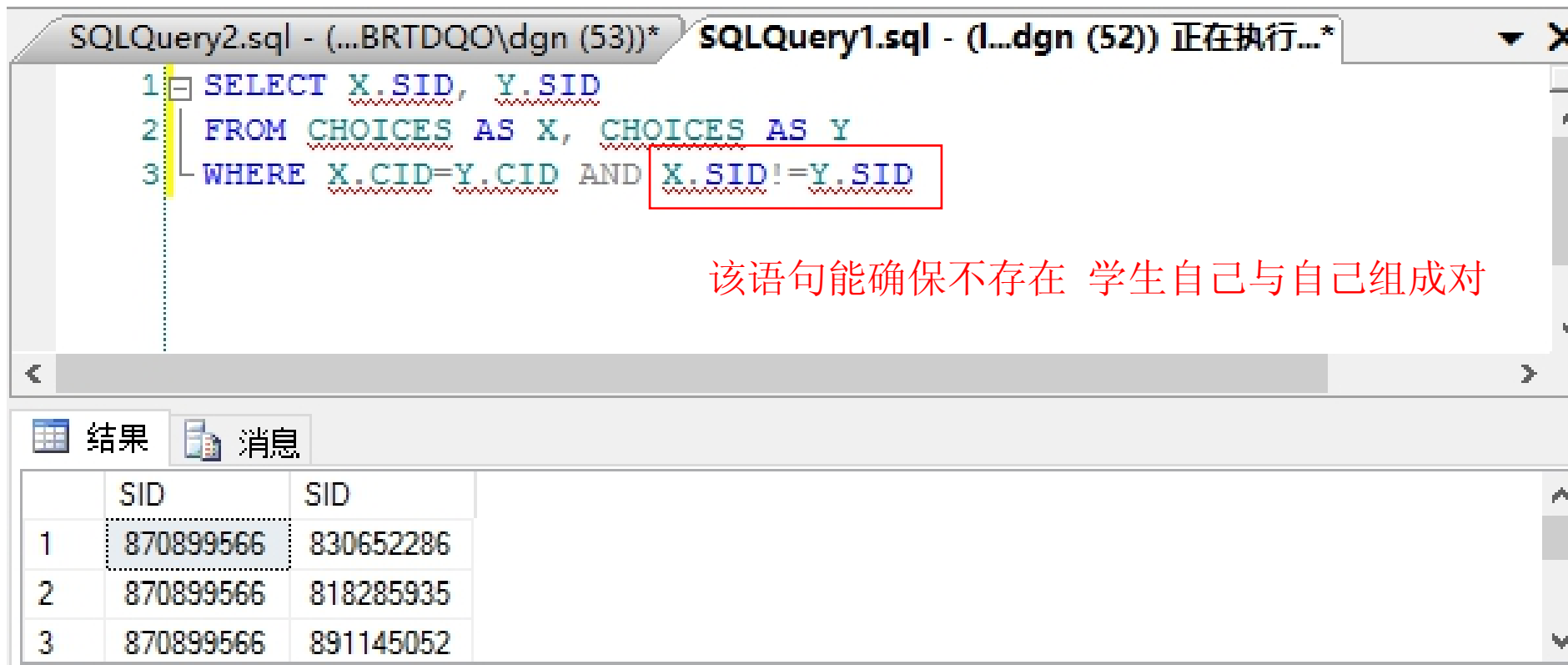
	SID
1	870899566
2	830652286
3	818285935
4	891145052
5	882649811

查询已成功执... | (local) (10.0 RTM) | DESKTOP-NBRTDQO\dgn (53) | School | 00:00:00 | 5898 行

• 实验示例

3. 查询选择了同一个课程的学生对。

（注：因为要查询同一课程的所有学生对，需要对表**CHOICES**进行自身连接，需要两个不同的名称来标志同一个表**CHOICES**，可以通过给表取不同别名实现）



```
1 SELECT X.SID, Y.SID
2 FROM CHOICES AS X, CHOICES AS Y
3 WHERE X.CID=Y.CID AND X.SID!=Y.SID
```

该语句能确保不存在 学生自己与自己组成对

	SID	SID
1	870899566	830652286
2	870899566	818285935
3	870899566	891145052

• 实验示例

4.查询与编号850955252的学生选修至少一门相同课程的学生们的编号。

The screenshot shows a SQL query window titled "SQLQuery1.sql - (...BRTDQO\dgn (52))*". The query is as follows:

```
1 SELECT Y.SID
2 FROM CHOICES AS X, CHOICES AS Y
3 WHERE X.CID=Y.CID AND X.SID='850955252'
```

Below the query editor, there are two tabs: "结果" (Results) and "消息" (Messages). The "结果" tab is active, displaying a table with the following data:

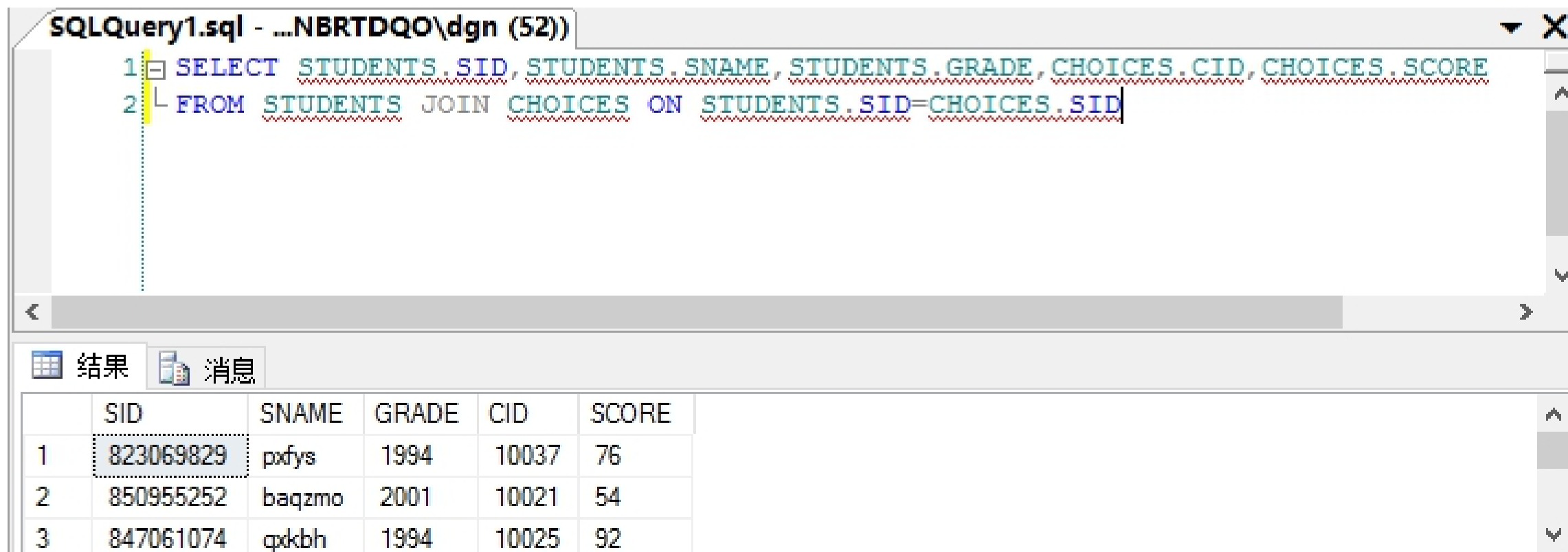
	SID
1	866594050
2	835316300
3	890098848

At the bottom of the window, a status bar indicates: "查询已成功..." (Query successful...), "(local) (10.0 RTM)", "DESKTOP-NBRTDQO\dgn (52)", "School", "00:00:00", and "23683 行" (23683 rows).

• 实验示例

5. 查询学生的基本信息以及选修课程编号和成绩（外连接）

（注：若某同学没有选课，则只输出其基本情况，其选课信息为空值即可，即用外连接）

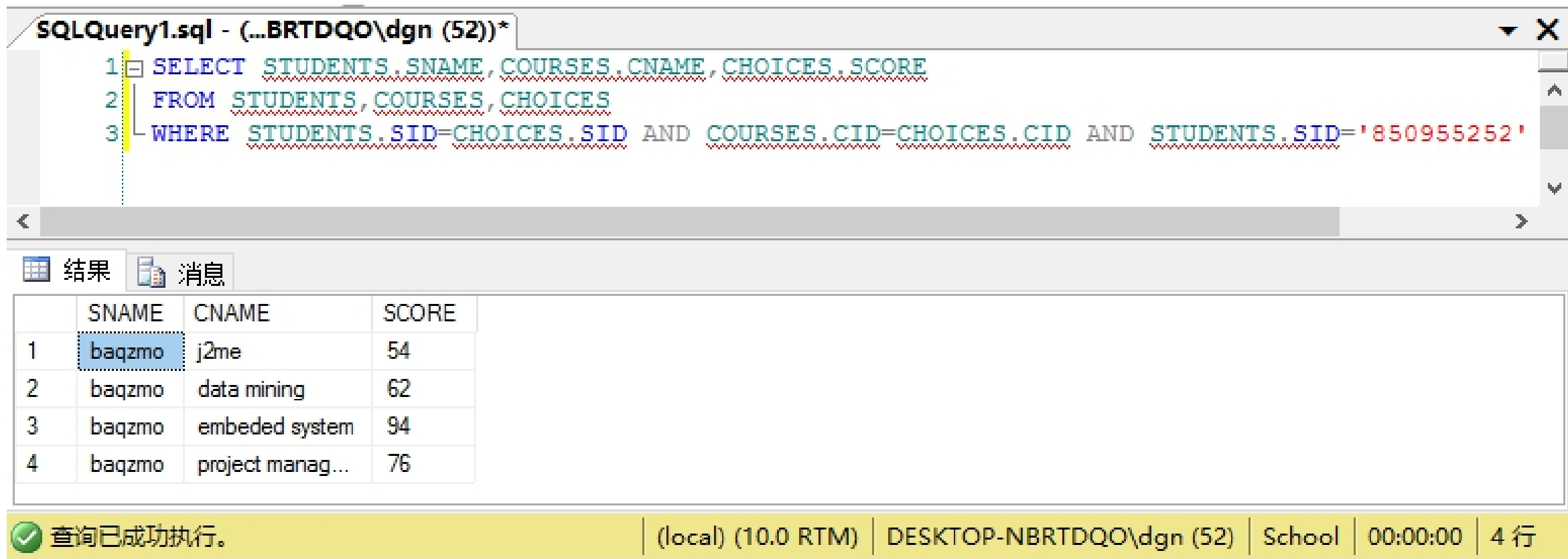


```
SQLQuery1.sql - ...NBRTDQO\dgn (52)
1 SELECT STUDENTS.SID, STUDENTS.SNAME, STUDENTS.GRADE, CHOICES.CID, CHOICES.SCORE
2 FROM STUDENTS JOIN CHOICES ON STUDENTS.SID=CHOICES.SID
```

	SID	SNAME	GRADE	CID	SCORE	
1	823069829	pxfys	1994	10037	76	
2	850955252	baqzmo	2001	10021	54	
3	847061074	qxkbh	1994	10025	92	

• 实验示例

6. 查询编号为850955252学生的姓名和选修课程名称以及成绩。
(注：多表连接)



The screenshot shows a SQL query window titled "SQLQuery1.sql - (...BRTDQO\dgn (52))*". The query is as follows:

```
1 SELECT STUDENTS.SNAME, COURSES.CNAME, CHOICES.SCORE
2 FROM STUDENTS, COURSES, CHOICES
3 WHERE STUDENTS.SID=CHOICES.SID AND COURSES.CID=CHOICES.CID AND STUDENTS.SID='850955252'
```

Below the query, there are two tabs: "结果" (Results) and "消息" (Messages). The "结果" tab is active, displaying a table with the following data:

	SNAME	CNAME	SCORE
1	baqzmo	j2me	54
2	baqzmo	data mining	62
3	baqzmo	embeded system	94
4	baqzmo	project manag...	76

At the bottom of the window, a status bar indicates: "查询已成功执行。" (Query executed successfully.) followed by "(local) (10.0 RTM) | DESKTOP-NBRTDQO\dgn (52) | School | 00:00:00 | 4 行" (4 rows).

• 练习

- (1) 查询全部课程的详细记录;
- (2) 查询所有有选修课的学生编号;
- (3) 查询课时<88(小时)的课程编号;
- (4) 请找出总分超过400分的学生;
- (5) 查询课程的总数;
- (6) 查询所有课程和选修该课程的学生总数;
- (7) 查询选修成绩超过60的课程超过两门的学生编号;
- (8) 统计各个学生的选修课程数目和平均成绩;
- (9) 查询选修Java的所有学生的编号及姓名;
- (10) 查询姓名为ssSht的学生所选的课程编号和成绩;
- (11) 查询其他课时比课程C++多的课程名称;