

Lecture 4, 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语句对数据库进行嵌套查询、集合运算。

• 实验内容

1. 嵌套查询

- 通过实验验证对子查询的两个限制条件：不能使用**order by**子句；外层**select**语句变量可以用在子查询中，但反之不行。
- 体会**相关子查询**和**不相关子查询**的不同：相关子查询要使用外层**select**语句提供的数据；不相关子查询即内层子查询不依赖于外层**select**语句。
- 考察**4类谓词**的用法,包括:
 - 第1类,**IN, NOT IN**
 - 第2类,带有**比较运算符**的子查询
 - 第3类,**SOME, ANY**或**ALL**谓词的子查询
 - 第4类,带有 **EXISTS**谓词的子查询

2. 集合运算

- 使用保留字 **UNION、INTERSECT、EXCEPT**进行集合或、交、减运算。

- 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. 嵌套查询

1. 查询学号850955252的学生同年级的所有学生资料

SQLQuery1.sql - (...BRTDQO\dgn (52))*

```
1 SELECT *
2 FROM STUDENTS
3 WHERE GRADE=(
4     SELECT GRADE
5     FROM STUDENTS
6     WHERE SID='850955252')
```

<

结果 消息

	sid	sname	email	grade
1	800028044	ztozk	r369l9m@lmykh.gov	2001
2	800041569	pgmrkdhh	xpqj2wc@hrjtp.edu	2001
3	800070739	nkdnfq	pto7n@sci.com	2001

✓ 查询已成功执行。 (local) (10.0 RTM)

思路：先考虑子查询中查询编号850955252的学生的年级，再外层查询这些学生的基本资料。

实验示例

2. 查询所有的有选课的学生们的详细信息

SQLQuery1.sql - ...NBRTDQO\dgn (52))

```
1 SELECT *
2 FROM STUDENTS
3 WHERE sid IN
4     (SELECT sid
5      FROM CHOICES
6      )
```

结果 消息

	sid	sname	email	grade
1	800001216	gfxrgs	hhce4@qhldj.gov	1992
2	800006682	fiiluommh	ihzd6_k@kzvft.gov	1992
3	800006941	ogvmu	62sfbd@lt.gov	1995

查 (local) (10.0 RTM) | DESKTOP-NBRTDQO\dgn (52)

注：子查询中不能使用order by子句。

SQLQuery1.sql - (...BRTDQO\dgn (52))*

```
1 SELECT *
2 FROM STUDENTS
3 WHERE sid IN
4     (SELECT sid
5      FROM CHOICES
6      ORDER BY sid
7      )
```

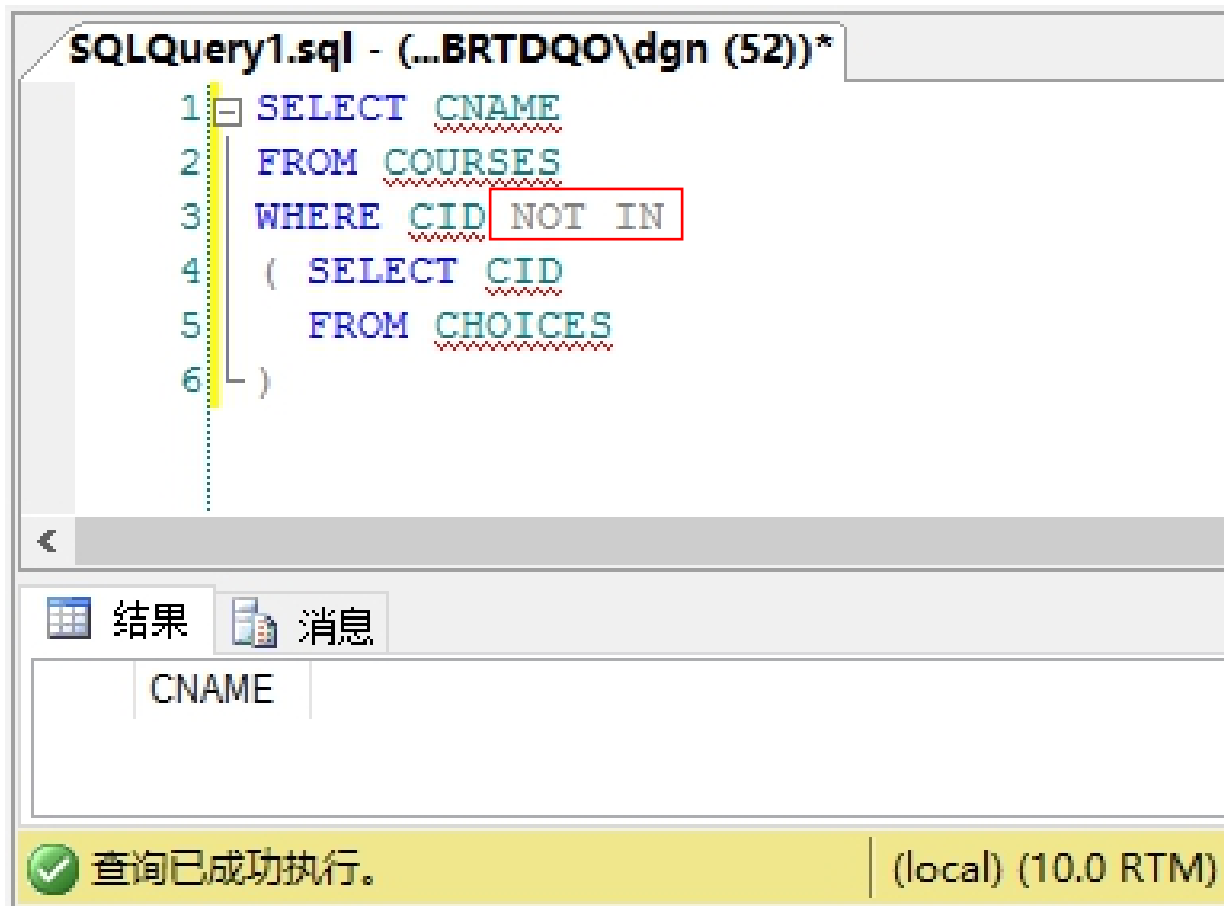
消息

消息 1033, 级别 15, 状态 1, 第 7 行
除非另外还指定了 TOP 或 FOR XML, 否则, ORDER BY 子句在视图、内联函数、派生

查询已... (local) (10.0 RTM) | DESKTOP-NBRTDQO\dgn (52) | School | 00:00:00 | 0 行

- 实验示例

- 3.查询没有学生选的课程编号



The screenshot shows a SQL Server Enterprise Manager interface. At the top, a window titled 'SQLQuery1.sql - (...BRTDQO\dgn (52))*' contains the following SQL query:

```
1 SELECT CNAME
2 FROM COURSES
3 WHERE CID NOT IN
4 ( SELECT CID
5   FROM CHOICES
6 )
```

Below the query window, there are two tabs: '结果' (Results) and '消息' (Messages). The '结果' tab is active, displaying a table with one column header 'CNAME'. The table is currently empty. At the bottom of the interface, a yellow status bar shows a green checkmark icon and the text '查询已成功执行。' (Query executed successfully.), followed by '(local) (10.0 RTM)'.

注：
类似上一题，不过使用的是**NOT IN**

实验示例

4. 查询选修课程成绩中最差的选课记录

SQLQuery1.sql - (...BRTDQO\dgn (52))*

```
1 SELECT *
2 FROM CHOICES
3 WHERE CHOICES.SCORE<=ALL
4 ( SELECT SCORE
5   FROM CHOICES
6   WHERE SCORE IS NOT NULL
7 )
```

查询成绩最差，可用‘<=ALL’来实现

不要漏，否则查询可能为空值

结果 消息

	no	sid	tid	cid	score
1	502212292	867147599	241941534	10026	50
2	502216521	865886581	203770215	10025	50
3	502262872	854282824	203768264	10014	50

查询已成功执行。 (local) (10.0 RTM) DESKTOP-NBRTDQO\dgn (52) School 00:00:00 5961 行

• 实验示例

5.找出和课程UML或课程C++的课时一样的课程名称。（可用‘=SOME’实现）

SQLQuery1.sql - (...BRTDQO\dgn (52))*

```
1 SELECT CNAME
2 FROM COURSES
3 WHERE HOUR=SOME
4 (
5     SELECT HOUR
6     FROM COURSES
7     WHERE CNAME='uml' OR CNAME='c++'
8 )
```

<

结果 消息

	CNAME
1	c++
2	uml
3	data s...
4	comp...

✓ 查询已成功执行。 (local) (10.0 RT)

注：
‘=SOME’相当于‘IN’,也可替换为
‘=ANY’

• 实验示例

6. 查询所有选修编号为10001课程的学生姓名

```
SQLQuery4.sql - (...BRTDQO\dgn (53))* SQLQuery1.sql - (...BRTDQO\dgn (52))*
1 SELECT SNAME
2 FROM STUDENTS
3 WHERE EXISTS
4 (
5 SELECT *
6 FROM CHOICES X
7 WHERE X.CID='10001' AND X.SID=STUDENTS.SID
8 )
```

查找是否存在符合某条件的元祖，可使用**EXISTS**语句

此为相关子查询。内层的**select**语句需要接受外层传递的 **student.sid** 变量。

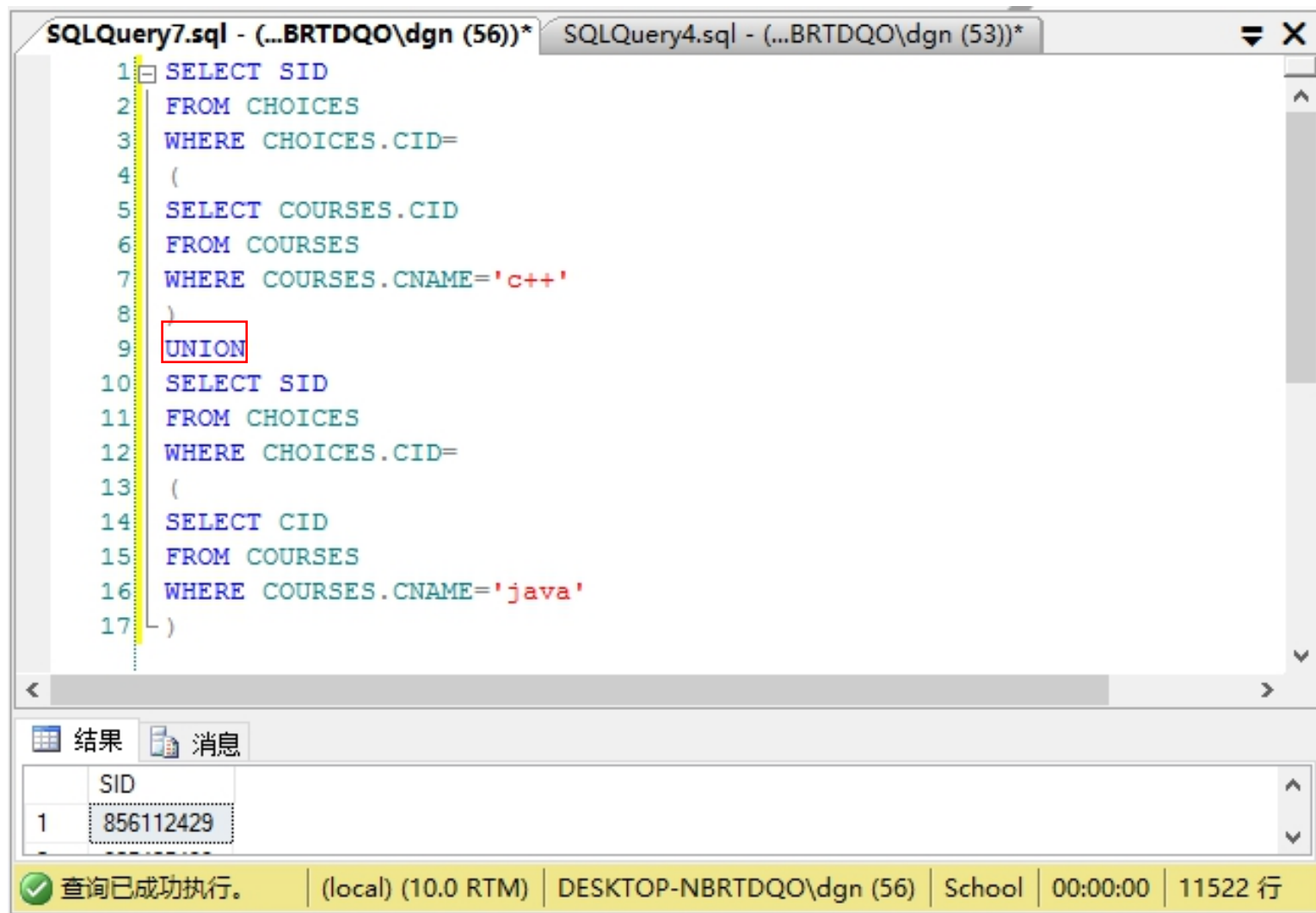
等同于以下查询语句

```
SQLQuery4.sql - (...BRTDQO\dgn (53))* SQLQuery6.sql - (...BRTDQO\dgn (55))*
1 SELECT distinct sname
2 FROM STUDENTS, CHOICES
3 WHERE (CHOICES.cid='10001' and STUDENTS.sid=CHOICES.sid)
4
```

此处**distinct**原因是，可能出现某学生重修一门课，即同一门课可能有两个重复的学生名字，所以需要去重

• 实验示例

7.利用集合并运算，查询选修课程C++或选修JAVA课程的学生编号。（UNION）



The screenshot shows a SQL query editor with two tabs: 'SQLQuery7.sql - (...BRTDQO\dgn (56))*' and 'SQLQuery4.sql - (...BRTDQO\dgn (53))*'. The active tab contains the following SQL code:

```
1 SELECT SID
2 FROM CHOICES
3 WHERE CHOICES.CID=
4 (
5 SELECT COURSES.CID
6 FROM COURSES
7 WHERE COURSES.CNAME='c++'
8 )
9 UNION
10 SELECT SID
11 FROM CHOICES
12 WHERE CHOICES.CID=
13 (
14 SELECT CID
15 FROM COURSES
16 WHERE COURSES.CNAME='java'
17 )
```

The word 'UNION' is highlighted with a red box. Below the query editor, there is a results pane with tabs for '结果' (Results) and '消息' (Messages). The '结果' tab is active, showing a table with the following data:

	SID
1	856112429

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 (56) | School | 00:00:00 | 11522 行

• 实验示例

注：如果上题不使用集合查询方式，而是采用**OR**连接两个判断条件，则会报错。

原因：带有比较运算符的子查询，该子查询必须返回单值，否则引起编译错误。



The screenshot shows a SQL query editor with two tabs. The active tab is 'SQLQuery8.sql - (...BRTDQO\dgn (57))*'. The query text is as follows:

```
1 SELECT S.ID
2 FROM CHOICES
3 WHERE CHOICES.CID=
4 (
5 SELECT COURSES.CID
6 FROM COURSES
7 WHERE COURSES.CNAME='c++' OR COURSES.CNAME='java'
8 )
```

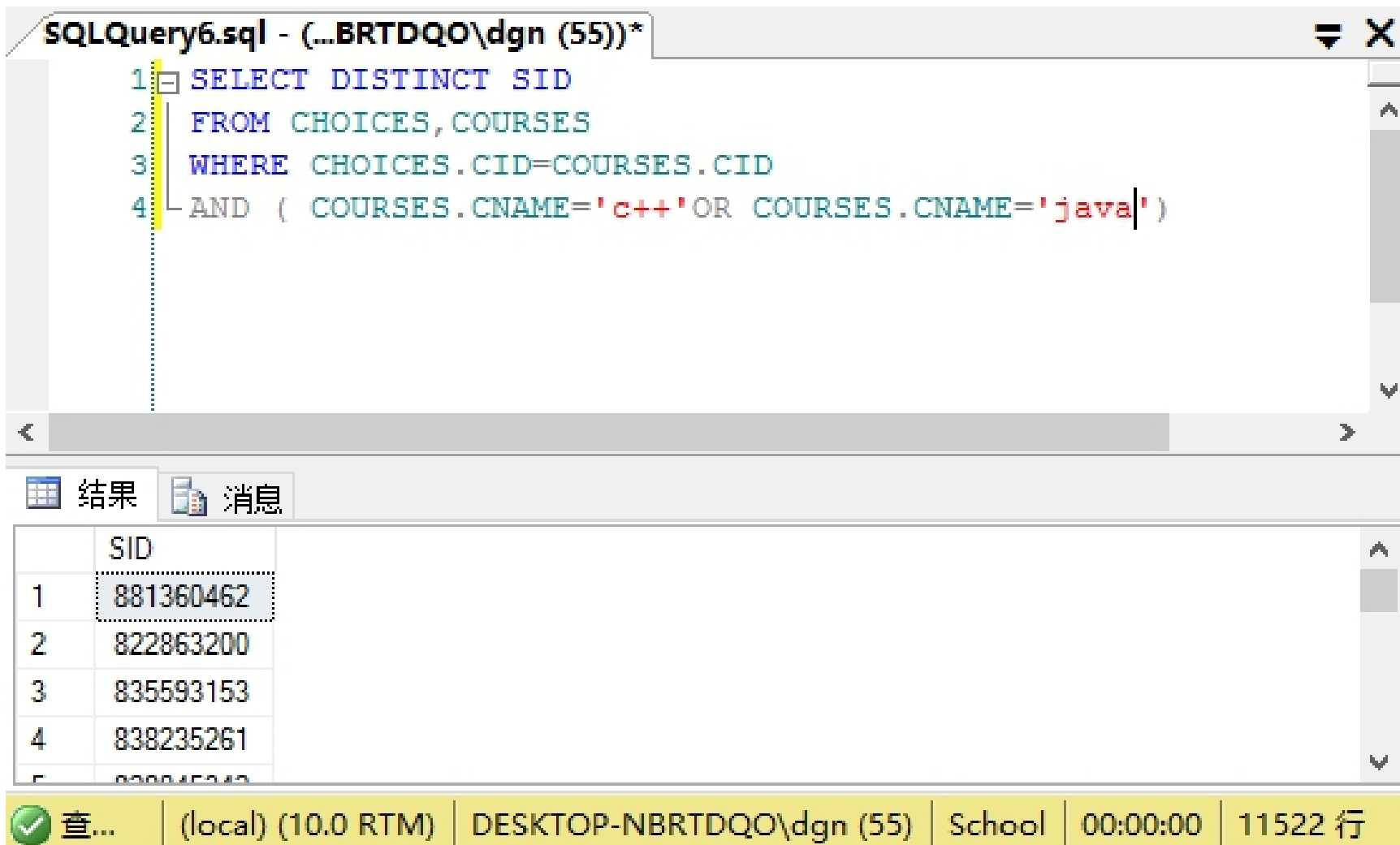
Below the query editor, there is a '消息' (Messages) tab. It displays the following error message:

消息 512, 级别 16, 状态 1, 第 1 行
子查询返回的值不止一个。当子查询跟随在 =、!=、<、<=、>、>= 之后, 或子查询用作表达式时, 这种情况是不允许的。

At the bottom of the window, a status bar indicates: 查询已完成, 但有错误。 (local) (10.0 RTM) | DESKTOP-NBRTDQO\dgn (57) | School | 00:00:00 | 0 行

• 实验示例

注：上题也可采用等值连接查询。但一般情况下，嵌套查询效率更高。



The screenshot shows a SQL query editor window titled "SQLQuery6.sql - (...BRTDQO\dgn (55))*". The query is as follows:

```
1 SELECT DISTINCT SID
2 FROM CHOICES, COURSES
3 WHERE CHOICES.CID=COURSES.CID
4 AND ( COURSES.CNAME='c++'OR COURSES.CNAME='java')
```

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:

	SID
1	881360462
2	822863200
3	835593153
4	838235261
5	838845243

At the bottom of the window, a status bar displays the following information: 查... (Query), (local) (10.0 RTM), DESKTOP-NBRTDQO\dgn (55), School, 00:00:00, and 11522 行 (11522 rows).

• 实验示例

8.利用集合减运算，查询及选修课程C++而没有选修JAVA课程的学生编号。(EXCEPT)

The screenshot shows a SQL query editor window titled "SQLQuery6.sql - (...BRTDQO\dgn (55))*". The query is as follows:

```
1 SELECT SID
2 FROM CHOICES
3 WHERE CHOICES.CID=
4 (
5 SELECT COURSES.CID
6 FROM COURSES
7 WHERE COURSES.CNAME='c++'
8 )
9 EXCEPT
10 SELECT SID
11 FROM CHOICES
12 WHERE CHOICES.CID=
13 (
14 SELECT CID
15 FROM COURSES
16 WHERE COURSES.CNAME='java'
17 )
```

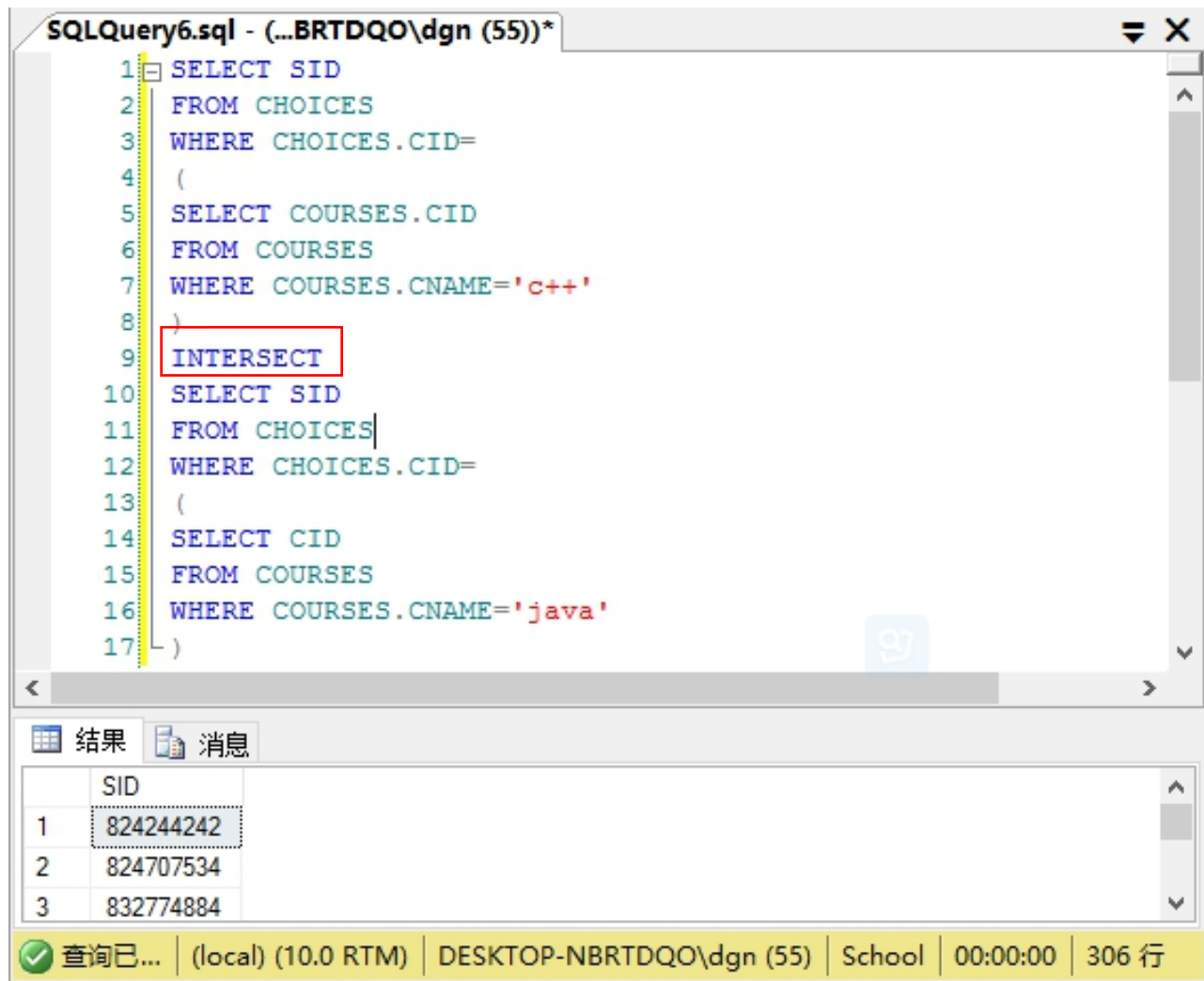
The word "EXCEPT" is highlighted with a red box. Below the query editor, there is a results pane with tabs for "结果" (Results) and "消息" (Messages). The "结果" tab is active, showing a table with the following data:

	SID
1	805425493
2	806857219
3	830629543

At the bottom of the window, a status bar displays: "查询..." (Query...), "(local) (10.0 RTM)", "DESKTOP-NBRTDQO\dgn (55)", "School", "00:00:00", and "5570 行" (5570 rows).

• 实验示例

9.利用集合交运算，查询及选修课程C++也选修JAVA课程的学生编号。(INTERSECT)



The screenshot shows a SQL query window titled "SQLQuery6.sql - (...BRTDQO\dgn (55))*". The query is as follows:

```
1 SELECT SID
2 FROM CHOICES
3 WHERE CHOICES.CID=
4 (
5 SELECT COURSES.CID
6 FROM COURSES
7 WHERE COURSES.CNAME='c++'
8 )
9 INTERSECT
10 SELECT SID
11 FROM CHOICES
12 WHERE CHOICES.CID=
13 (
14 SELECT CID
15 FROM COURSES
16 WHERE COURSES.CNAME='java'
17 )
```

The word "INTERSECT" is highlighted with a red box. Below the query, the results are displayed in a table with the following data:

	SID
1	824244242
2	824707534
3	832774884

The status bar at the bottom indicates: 查询已... | (local) (10.0 RTM) | DESKTOP-NBRTDQO\dgn (55) | School | 00:00:00 | 306 行

- 练习

- (1)查询选修**C++**课程的成绩比姓名为 **ZNKOO**的学生高的所有学生的编号和姓名;
- (2)找出和学生**883794999**或学生**850955252**的年级一样的学生的姓名;
- (3)查询没有选修**Java**的学生名称;
- (4)找出课时最少的课程的详细信息;
- (5)查询工资最高的教师的编号和开设的课程号;
- (6)找出选修课程**ERP**成绩最高的学生编号;
- (7)查询没有学生选修的课程名称;
- (8)查询讲授课程**UML**的教师所讲授的所有课程名称;
- (9)使用集合交运算, 查询既选修了**database**又选修了**UML**课程的学生编号;
- (10)使用集合减运算, 查询选修了**database**却没有选修**UML**课程的学生编号;