

实验 2、SQL 数据查询

1.1 实验目的

熟悉 SQL 语句的数据查询语言，能够 SQL 语句对数据库进行单表查询、连接查询、嵌套查询、集合查询和统计查询。

1.2 实验内容

实验内容主要是对数据库的数据进行查询操作，包括如下四类查询方式：

(1) 单表查询

- 查询的目标表达式为所有列、指定的列或指定的列的运算三种不同。
- 使用 **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。

(3) 嵌套查询

- 通过实验验证对子查询的两个限制条件。
- 体会相关子查询和不相关子查询的不同。
- 考察四类谓词的用法，包括：
 - 第一类，**IN**、**NOT IN**；
 - 第二类，带有比较运算符的子查询；
 - 第三类，**SOME**、**ANY** 或 **ALL** 谓词的子查询，查询最大值和最小值；
 - 第四类，带有 **EXISTS** 谓词的子查询，实现“所有”等情况（如王宏的“所有”课程，“所有”女生选修的课程）

(4) 集合运算

- 使用保留字 **UNION** 进行集合或运算。
- 采用逻辑运算符 **AND** 或 **OR** 来实现集合交和减运算。

1.3 实验步骤

以 **Chenht_University_Mis** 数据库为例，该数据库中有四张如实验 1，其中 **cht_Score** 是每门课的考试成绩，**cht_Scredit** 是学生所有考试合格课程所获得的积分总数，**cht_Ccredit** 每门课程的学分数。

在数据库中,存在这样的联系:学生可以选择课程,一个课程对应一个教师。
在表 Reports 中保存学生的选课记录和考试成绩。

请先输入如下符合条件的元组后,再对数据库进行有关的查询操作:

	Sno	Sname	Semail	Scredit	Ssex
►	S01	王建平	WJP@zjut.edu.cn	23.1	男
	S02	刘华	LH@zjut.edu.cn	34.6	女
	S03	范林军	FLJ@zjut.edu.cn	16.6	女
	S04	李伟	LW@zjut.edu.cn	15.8	男
	S26	黄河	Huanghe@zjut.edu.cn	13.4	男
	S52	长江	Changjiang@zjut.edu.cn	12.4	男
*					

图 1.1、chenht_Students 表

	Tno	Tname	Temail	Tsalary
►	T01	刘涛	LT@zjut.edu.cn	4300
	T02	吴碧燕	WBY@zjut.edu.cn	2500
	T03	张莹	ZY@zjut.edu.cn	3000
	T04	张宁亚	ZNY@zjut.edu.cn	5500
	T05	叶烨	Yeye@zjut.edu.cn	3800
	T06	杨兰	YL@zjut.edu.cn	3500
	T07	程媛媛	ChengYY@zjut.edu.cn	5000
*				

图 1.2、chenht_Teachers 表

	Cno	Cname	Ccredit
►	C01	C++	4
	C02	UML	3
	C03	JAVA	3
	C04	算法分析与设计	2
	C05	数据库原理及其应用	2
	C06	数据结构与算法分析	2
	C07	英语	4
*			

图 1.3、chenht_Courses 表

	Sno	Tno	Cno	Score
►	S01	T01	C01	83
	S01	T03	C03	83
	S02	T01	C01	75
	S02	T02	C02	59
	S02	T03	C03	<NULL>
	S02	T04	C04	<NULL>
	S02	T05	C05	77
	S02	T06	C06	86
	S02	T07	C07	80
	S03	T02	C02	88
	S03	T04	C04	76
	S04	T05	C05	90
	S04	T06	C06	89
	S26	T04	C04	72
	S26	T05	C05	95
	S26	T07	C07	55
	S52	T01	C01	61
	S52	T05	C05	90
	S52	T07	C07	<NULL>
*				

图 1.4、chenht_Reports 表

- (1) 查询性别为“男”的所有学生的名称并按学号升序排列。
- (2) 查询学生的选课成绩合格的课程成绩,并把成绩换算为积分。积分的计算公式为: $[1+(考试成绩-60)*0.1]*cht_Ccredit$ 。考试成绩 ≥ 60

否则=0

- (3) 查询学分是 3 或 4 的课程名称。
- (4) 查询所有课程名称中含有“算法”的课程编号。
- (5) 查询所有选课记录的课程号（不重复显示）。
- (6) 统计所有老师的平均工资。
- (7) 查询所有教师的编号及选修其课程的学生们的平均成绩，按平均成绩降序排列。
- (8) 统计各个课程的选课人数和平均成绩。
- (9) 查询至少选修了三门课程的学生编号和姓名。
- (10) 查询编号 S26 的学生所选的全部课程的课程名和成绩。
- (11) 查询所有选了“数据库原理及其应用”课程的学生编号和姓名。
- (12) 求出选择了同一个课程的学生对。
- (13) 求出至少被两名学生选修的课程编号。
- (14) 查询选修了编号 S26 的学生所选的某个课程的学生编号。
- (15) 查询学生的基本信息及选修课程编号和成绩。
- (16) 查询学号 S52 的学生的姓名和选修的课程名称及成绩。
- (17) 查询和学号 S52 的学生同性别的所有学生资料。
- (18) 查询所有选课的学生们的详细信息。
- (19) 查询没有学生选的课程编号和名称。
- (20) 查询选修了课程名为 C++ 的学生学号和姓名。
- (21) 找出选修课程 UML 或者课程 C++ 的学生学号和姓名。
- (22) 找出和课程 UML 或课程 C++ 的学分一样课程名称。
- (23) 查询所有选修编号 C01 的课程的学生们的姓名。
- (24) 查询选修了所有课程的学生姓名。
- (25) 利用集合查询方式，查询选修课程 C++ 或选择课程 JAVA 的学生的编号、姓名和积分。
- (26) 实现集合交运算，查询既选修课程 C++ 又选修课程 JAVA 的学生的编号、姓名和积分。
- (27) 实现集合减运算，查询选修课程 C++ 而没有选修课程 JAVA 的学生的编号。

1.4 实验结果

- 2) 在之间创建的数据库 chenht_University_Mis 右键，新建查询，输入查询语句，并点击执行，获得结果。之后的所有查询操作与这个类似。

查询题目：查询学生的选课成绩合格的课程成绩，并把成绩换算为积分。积分的计算公式为： $[1+(考试成绩-60)*0.1]*cht_Ccredit$ 。考试成绩 ≥ 60 否则=0

SQL 语句为：

```
select cht_sno, chenht_Reports.cht_cno, cht_score,
(1+(cht_score-60)*0.1*cht_ccredit)cht_p
from chenht_Reports, chenht_Courses
where cht_score>=60 and chenht_Reports.cht_cno =
chenht_Courses.cht_cno
union select cht_sno, chenht_Reports.cht_cno, cht_score,
(0)cht_p
from chenht_Reports
```

```
where cht_score<60
```

实验结果如图 2-1.

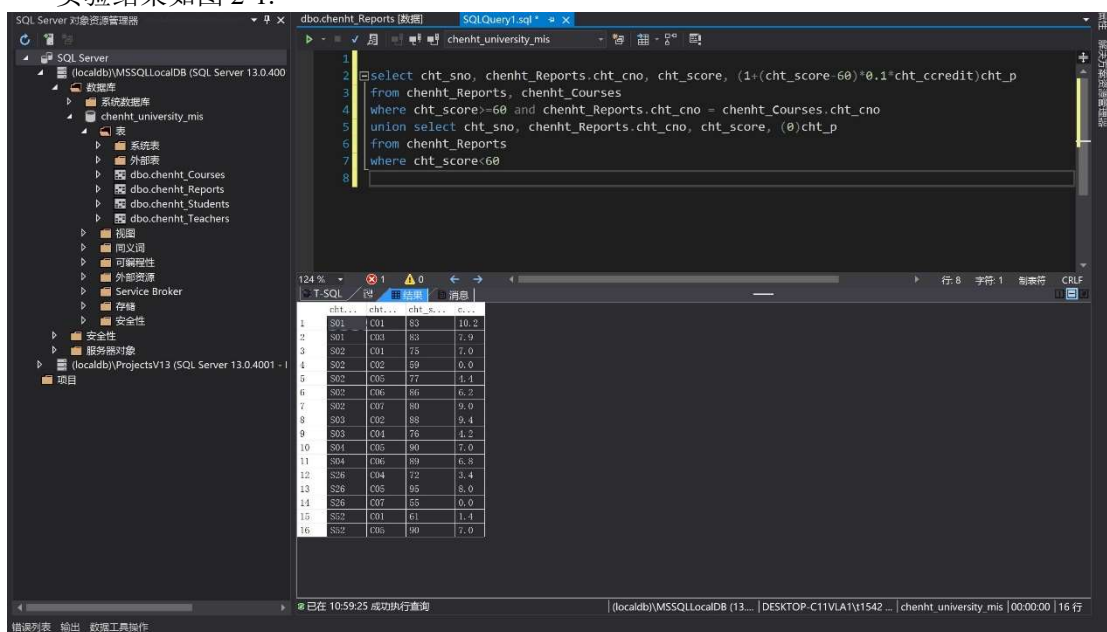


图 2-1 查询（2）的命令和结果截图

7) 进行查询操作

查询题目：查询所有教师的编号及选修其课程的学生的平均成绩，按平均成绩降序排列。

SQL 语句为：

```
select chenht_Teachers.cht_tno, avg(cht_score)avg_grade
from chenht_Teachers, chenht_Reports
where chenht_Teachers.cht_tno = chenht_Reports.cht_tno
group by chenht_Teachers.cht_tno
order by avg_grade desc
```

实验结果如图 2-2

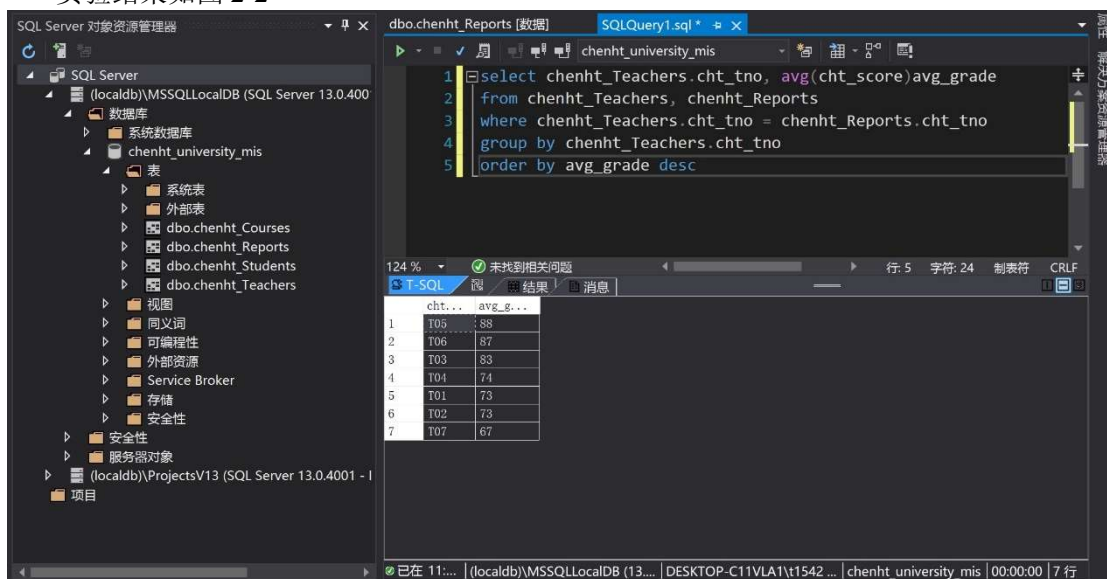


图 2-2 查询（7）的命令和结果截图

11) 进行查询操作

查询题目：查询所有选了“数据库原理及其应用”课程的学生编号和姓名。

SQL 语句为:

```
select chenht_Students.cht_sno, cht_sname
from chenht_Students, chenht_Reports
where chenht_Students.cht_sno = chenht_Reports.cht_sno
    and chenht_Reports.cht_cno = (
        select cht_cno from chenht_Courses
        where cht_cname like '数据库原理及其应用'
    )
```

实验结果如图 2-3

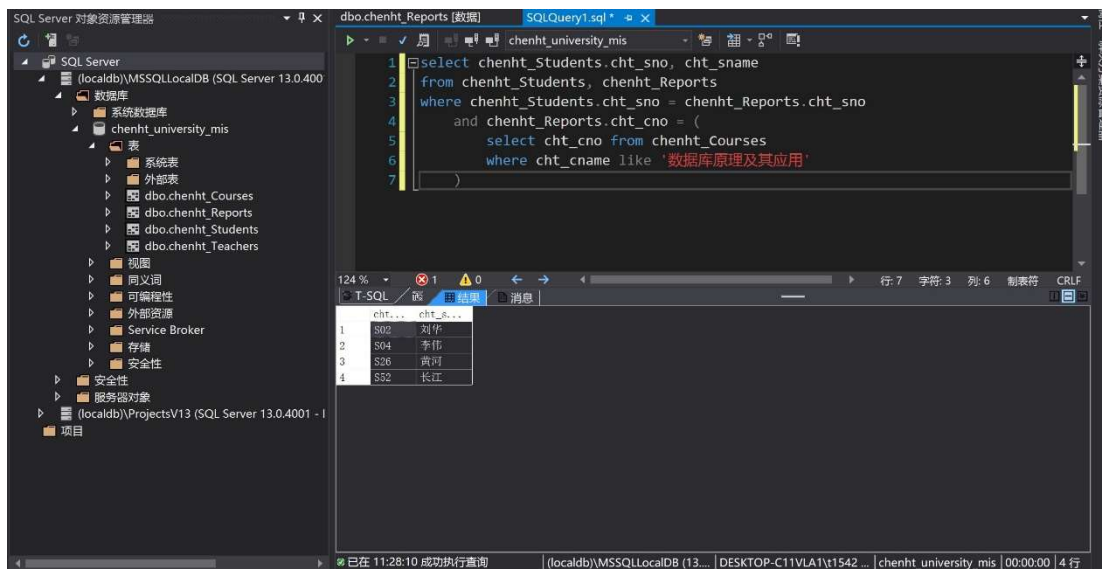


图 2-3 查询 (11) 的命令和结果截图

12) 进行查询操作

查询题目: 求出选择了同一个课程的学生对。

SQL 语句为

```
select R1.cht_cno, R1.cht_sno pair_1, R2.cht_sno pair_2
from chenht_Reports R1, chenht_Reports R2
where R1.cht_cno = R2.cht_cno and R1.cht_sno < R2.cht_sno
```

实验结果如图 2-4

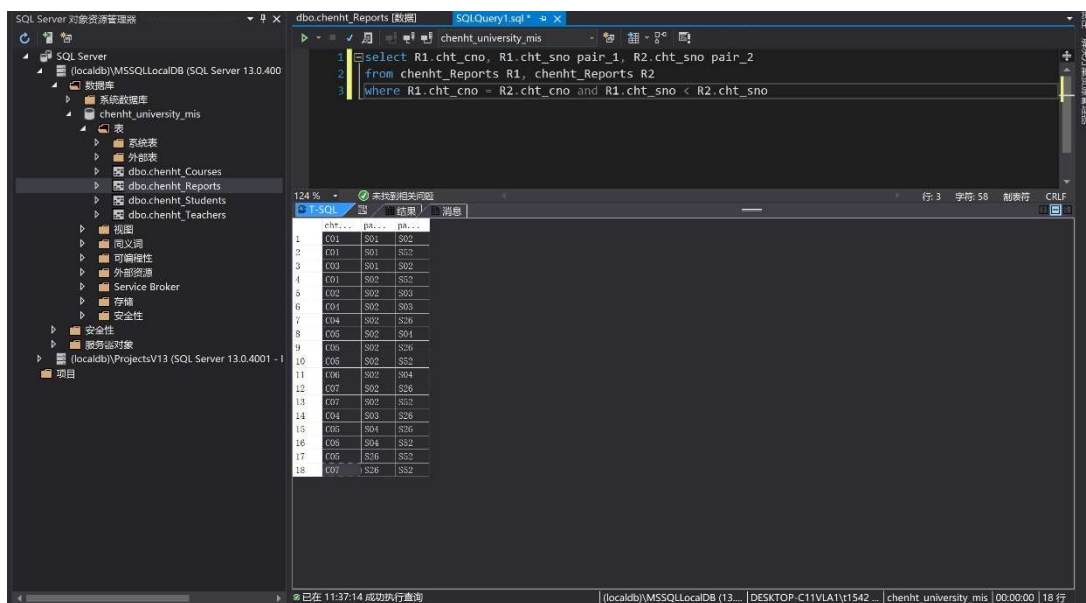


图 2-4 查询（12）的命令和结果截图

16) 进行查询操作

查询题目：查询学号 S52 的学生的姓名和选修的课程名称及成绩。

SQL 语句为：

```
select chenht_Students.cht_sname, chenht_Courses.cht_cname,
chenht_Reports.cht_score
from chenht_Reports, chenht_Courses, chenht_Students
where chenht_Reports.cht_cno = chenht_Courses.cht_cno
and chenht_Students.cht_sno = chenht_Reports.cht_sno
and chenht_Students.cht_sno = 'S52'
```

实验结果如图 2-5

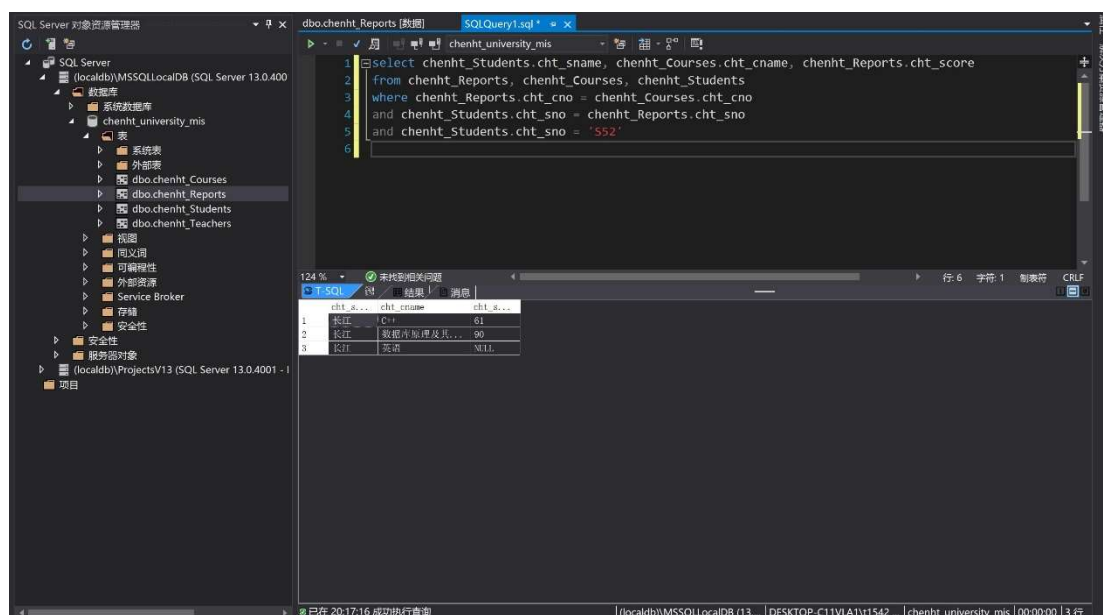


图 2-5 查询（16）的命令和结果截图

17) 进行查询操作

查询题目：查询和学号 S52 的学生同性别的所有学生资料。

SQL 代码为

```

select chenht_Students.*
from chenht_Students
where chenht_Students.cht_ssex = (
    select cht_ssex
    from chenht_Students
    where cht_sno = 'S52'
)

```

实验结果如图 2-6

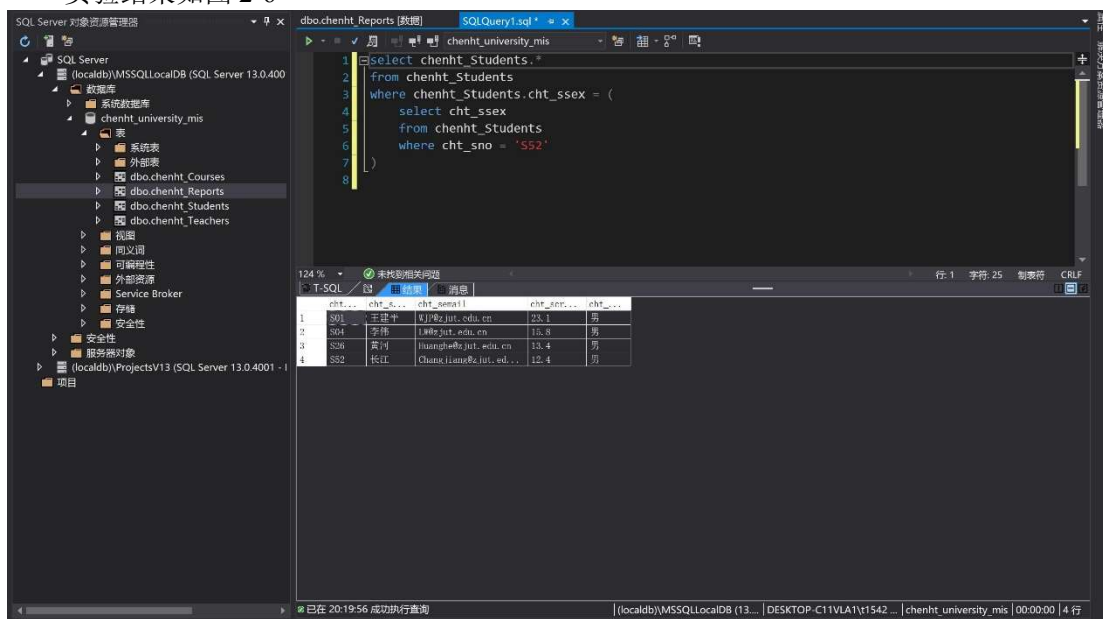


图 2-6 查询（17）的命令和结果截图

19) 进行查询操作

查询题目：查询没有学生选的课程的编号和名称。

SQL 代码为：

```

select chenht_Courses.cht_cno, chenht_Courses.cht_cname
from chenht_Courses
where cht_cno not in (
    select distinct(cht_cno)
    from chenht_Reports
)

```

实验结果如图 2-7

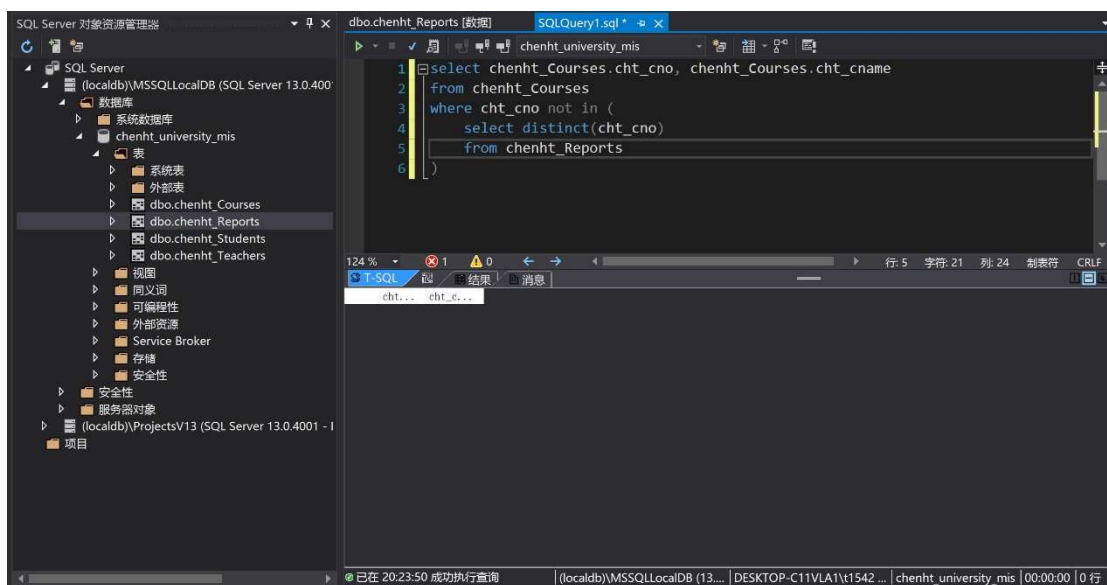


图 2-7 查询（19）的命令和结果截图

24) 进行查询操作

查询题目：查询选修了所有课程的学生姓名。

SQL 代码为

```
select chenht_Students.cht_sname
from chenht_Students
where not exists (
    select *
    from chenht_Courses
    where not exists(
        select *
        from chenht_Reports
        where chenht_Courses.cht_cno =
chenht_Reports.cht_cno
        and chenht_Students.cht_sno =
chenht_Reports.cht_sno
    )
)
```

实验结果如图 2-8

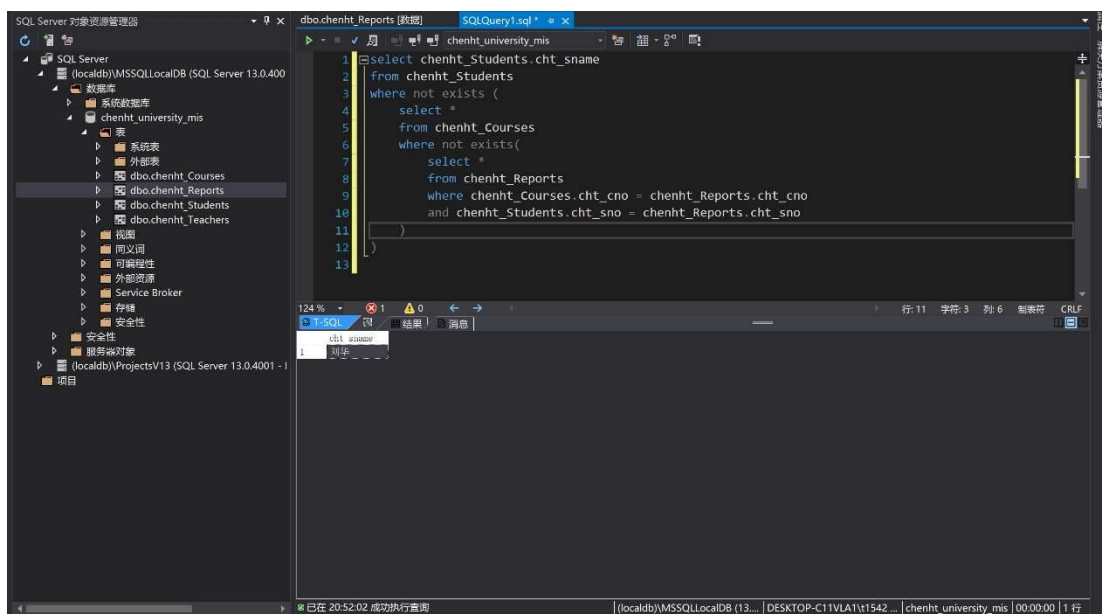


图 2-8 查询（24）的命令和结果截图

1.5 实验体会

1) 实验反思

1、在进行 group by 语句进行分组时，在 select 和 having 语句块只能使用 group by 约束列或者其他列的集函数。其在逻辑上可以理解为在进行分组操作后，select 为分组后的各个块。因为块是一个集合，只能使用集函数和 group by 约束列。

在一种情况下，以内连接的主表作为约束列进行分组时。虽然结果中有些字段在每一个被分组的块内是唯一的，但是只要不出现在约束列中，仍然不能被 select 选择，在这里典型的例子中为 9 题。

错误的 SQL 代码为：（因为 select 了非约束键，虽然 cht_sname 在每一组中是唯一的）

```
select chenht_Students.cht_sno, cht_sname
from chenht_Students, chenht_Reports
where chenht_Students.cht_sno = chenht_Reports.cht_sno
group by chenht_Students.cht_sno having
count(chenht_Students.cht_sno)>=3
```

正确的代码应该使用 in 关键词将分组作为内部集合映射出去。

正确的 SQL 代码为：

```
select cht_sno, cht_sname
from chenht_Students
where cht_sno in (
    select cht_sno from chenht_Reports
    group by cht_sno having count(cht_cno)>=3
)
```

2、在很多情况下 in 关键词比 = 关键词更通用，= 的右半部分嵌套 select 查询时，其返回的结果只能为 1 行 1 列的结果，如果查询得出多列的结果，则会查询失败。一般情况下，为了防止出现错误，常常更多使用 in 关键词。

3、使用 distinct 关键词，当下一次出现重复的元素时，会自动跳过。

当结果集是多列的时候，其他列的信息只会返回一条。一般只会用于其他列是一対一的情况（常常在连接查询时会用到）。

2) 实验收获

通过实验中各种出现的错误，对查询时需要注意的地方有一个更加深入的了解。可以减少后续实验中出现的错误。对查询的各种模式有了更加深刻的理解。