

## 实验 3 数据库实验

学号	19335015	姓名	陈恩婷
----	----------	----	-----

### 一、 实验目的

熟悉使用 select 语句进行数据库的查询，熟悉 SQL 语言支持的有关视图的操作，能够熟练地使用 SQL 语句来创建需要的视图，对视图进行查询和取消等。

### 二、 实验内容

使用 SELECT 子句设定查询内容

1. 查询图书表 book 的所有列
2. 查询读者表 reader 的读者姓名和所在院系
3. 查询读者表 reader 的读者姓名和所在系，要求查询结果显示为“姓名”和“院系”
4. 查询图书表 book 的所有信息，只显示查询结果的前 5 行数据。
5. 查询图书表 book 的出版社名称，要求出版社名称不重复显示。
6. 查询数据表 book 的借阅次数的总和，要求查询结果显示为“总借阅次数”
7. 在读者表 reader 中统计读者总人数
8. 查询读者表 reader 中所有姓“王”的读者的信息
9. 查询图书表 record 中 notes 列值为空的记录
10. 查询图书表 book 中所有图书信息，并按借阅次数由高到低进行排序

- 11.在图书表 book 中统计各出版社出版图书的总数
- 12.在图书表 book 中查询图书借阅次数在 20 次以上, 而且出版图书总数大于 1 的出版社。
- 13.查询没有借阅编号为 “b0005” 这本书的读者姓名
- 14.查询借阅至少 2 本书的读者姓名
- 15.查询借阅次数最多和最少的图书编号
- 16.查询借阅过图书编号为 “b0002” 图书的读者编号和姓名
- 17.查询编号为 “r0007” 读者的姓名、所在院系以及所借阅图书的书名。

## 视图

1. 创建视图, 用于查看借阅次数大于 30 次的图书信息。
2. 创建视图, 用于查看借阅了图书的读者姓名, 以及他们借阅的图书名称。
3. 创建视图, 用于查询访问次数大于 30 次的图书名称和借阅次数, 并在视图中为列指定别名。
4. 上面的视图可更新吗? 通过 SQL 更新语句来验证。
5. 删除上面视图

## 三、实验报告

### 1. 实验代码

使用 SELECT 子句设定查询内容

1. 查询图书表 book 的所有列

```
SELECT *  
FROM BOOK
```

2. 查询读者表 reader 的读者姓名和所在院系

```
SELECT reader_name, reader_department  
FROM READER
```

3. 查询读者表 reader 的读者姓名和所在系，要求查询结果显示为“姓名”和“院系”

```
SELECT reader_name AS '姓名', reader_department AS '院系'  
FROM READER
```

4. 查询图书表 book 的所有信息，只显示查询结果的前 5 行数据。

```
SELECT TOP 5  
*  
FROM BOOK
```

5. 查询图书表 book 的出版社名称，要求出版社名称不重复显示。

```
SELECT DISTINCT book_publisher  
FROM BOOK
```

6. 查询数据表 book 的借阅次数的总和，要求查询结果显示为“总借阅次数”

```
SELECT SUM(interview_times) AS '总借阅次数'  
FROM BOOK
```

7. 在读者表 reader 中统计读者总人数

```
SELECT COUNT(reader_id)  
FROM READER
```

8. 查询读者表 reader 中所有姓“王”的读者的信息

```
SELECT *  
FROM READER  
WHERE reader_id LIKE '王%'
```

9. 查询图书表 record 中 notes 列值为空的记录

```
SELECT *  
FROM RECORD  
WHERE notes IS NULL
```

10. 查询图书表 book 中所有图书信息，并按借阅次数由高到低进行排序

```
SELECT *  
FROM BOOK  
ORDER BY interview_times DESC
```

11.在图书表 book 中统计各出版社出版图书的总数

```
SELECT book_publisher, COUNT(book_id)
FROM BOOK
GROUP BY book_publisher
```

12.在图书表 book 中查询图书借阅次数在 20 次以上, 而且出版图书总数大于 1 的出版社。

```
SELECT book_publisher
FROM
    (SELECT book_publisher, COUNT(book_id) AS book_num, SUM(int
erview_times) AS interview_times
    FROM BOOK
    GROUP BY book_publisher) AS T
WHERE (book_num > 1 AND interview_times > 20)
```

13.查询没有借阅编号为 “b0005” 这本书的读者姓名

```
SELECT reader_name
FROM READER
WHERE (reader_id NOT IN (SELECT reader_id
FROM RECORD
WHERE book_id = 'b0005'))
```

14.查询借阅至少 2 本书的读者姓名

```
SELECT reader_name
FROM READER
WHERE reader_id IN
(SELECT reader_id
FROM RECORD
GROUP BY reader_id
HAVING COUNT(book_id) >= 2 )
```

15.查询借阅次数最多和最少的图书编号

```
SELECT book_id as book_id_max
FROM BOOK
WHERE interview_times IN (SELECT MAX(interview_times)
FROM BOOK)

SELECT book_id as book_id_min
FROM BOOK
WHERE interview_times IN (SELECT MIN(interview_times)
FROM BOOK)
```

16.查询借阅过图书编号为“b0002” 图书的读者编号和姓名

```
SELECT reader_name, reader_id
FROM READER
WHERE reader_id IN (SELECT reader_id
FROM RECORD
WHERE book_id = 'b0002' )
```

17.查询编号为“r0007” 读者的姓名、所在院系以及所借阅图书的书名。

```
SELECT reader_name, reader_department, book_name
FROM READER JOIN RECORD ON READER.reader_id = RECORD.reader_id
JOIN BOOK ON RECORD.book_id = BOOK.book_id
WHERE READER.reader_id = 'r0007'
```

## 视图

1. 创建视图，用于查看借阅次数大于 30 次的图书信息。

```
CREATE VIEW V1 AS
SELECT * FROM BOOK
WHERE interview_times > 30
```

2. 创建视图，用于查看借阅了图书的读者姓名，以及他们借阅的图书名称。

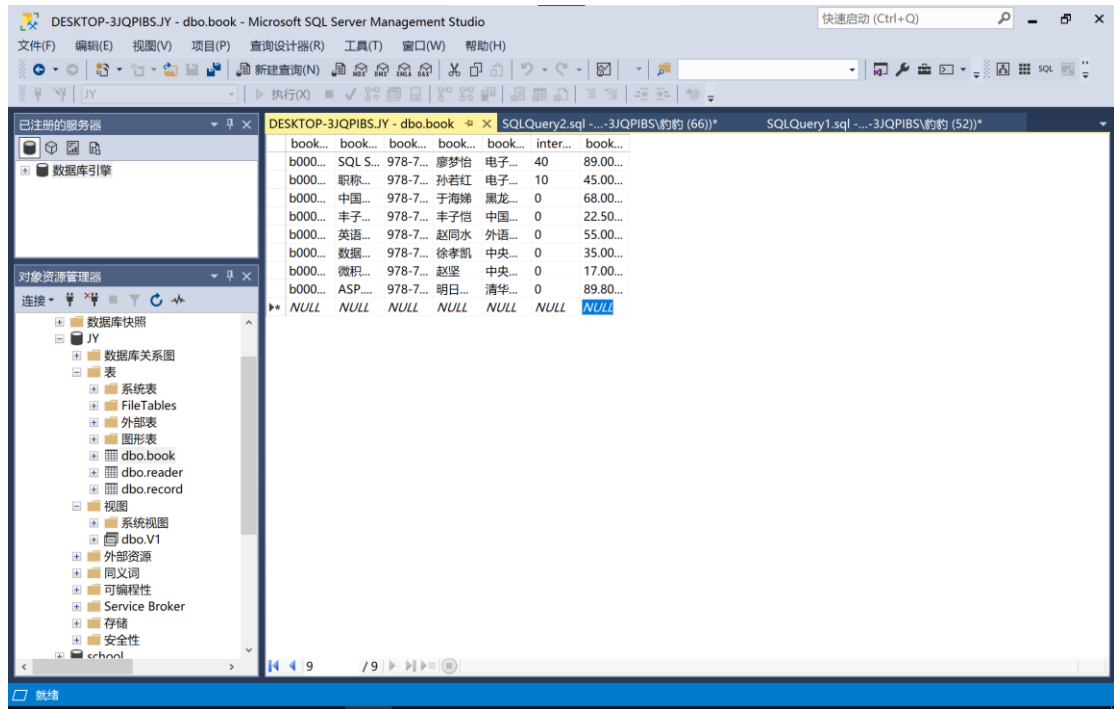
```
CREATE VIEW V2 AS
SELECT reader_name, book_name
FROM READER
JOIN RECORD ON READER.reader_id = RECORD.reader_id
JOIN BOOK ON RECORD.book_id = BOOK.book_id
```

3. 创建视图，用于查询访问次数大于 30 次的图书名称和借阅次数，并在视图中为列指定别名。

```
CREATE VIEW V3
AS
SELECT book_name, interview_times
FROM BOOK
WHERE interview_times > 30
```

4. 上面的视图可更新吗？通过 SQL 更新语句来验证。

先修改 book 表, 将《SQL Server 宝典》一书的借阅次数改为 41, 使得 V3 中有数据:



book...	book...	book...	book...	book...	inter...	book...
b000...	SQL S...	978-7...	廖梦怡	电子...	40	89.00...
b000...	职称...	978-7...	孙若红	电子...	10	45.00...
b000...	中国...	978-7...	于海娣	黑龙江...	0	68.00...
b000...	丰子...	978-7...	丰子恺	中国...	0	22.50...
b000...	英语...	978-7...	赵同水	外语...	0	55.00...
b000...	数据...	978-7...	徐孝凯	中央...	0	35.00...
b000...	微积...	978-7...	赵坚	中央...	0	17.00...
b000...	ASP...	978-7...	明日...	清华...	0	89.80...
NULL	NULL	NULL	NULL	NULL	NULL	NULL

再执行 UPDATE 操作:

```
UPDATE V3 SET interview_times = 41
WHERE book_name = 'SQL Server 2012 宝典'
```

5. 删除上面视图

```
DROP VIEW V3
```

## 2. 实验结果

### 使用 SELECT 子句设定查询内容

#### 1. 查询图书表 book 的所有列

The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor contains the following SQL statement:

```
1 --1
2 SELECT *
3 FROM BOOK
```

The results pane displays the following data:

book_id	book_name	book_isbn	book_author	book_publisher	interview_times	book_price
b0001	SQL Server 2012宝典	978-7-121-22013-5	廖梦怡	电子工业出版社	10	89.00
b0002	职称英语专用教材	978-7-121-14800-2	孙若红	电子工业出版社	10	45.00
b0003	中国通史	978-7-3388-53155	于海娣	黑龙江科学技术出版社	0	68.00
b0004	丰子恺儿童文学选集	978-7-3007-8972-7	丰子恺	中国少年儿童出版社	0	22.50
b0005	英语同义词辨析词典	978-7-5135-2294-6	赵同永	外语教学与研究出版社	0	55.00
b0006	数据库基础与应用	978-7-304-06430-3	徐孝凯	中央广播电视大学出版社	0	35.00
b0007	微积分初步	978-7-304-03742-0	赵圣	中央广播电视大学出版社	0	17.00
b0008	ASP.NET从入门到精通	978-7-302-28753-7	明日科技	清华大学出版社	0	89.80

The status bar at the bottom indicates: 查询已成功执行。 DESKTOP-3JQPIBS (15.0 RTM) DESKTOP-3JQPIBS\约约 (66) JY 00:00:00 8 行

#### 2. 查询读者表 reader 的读者姓名和所在院系

The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor contains the following SQL statement:

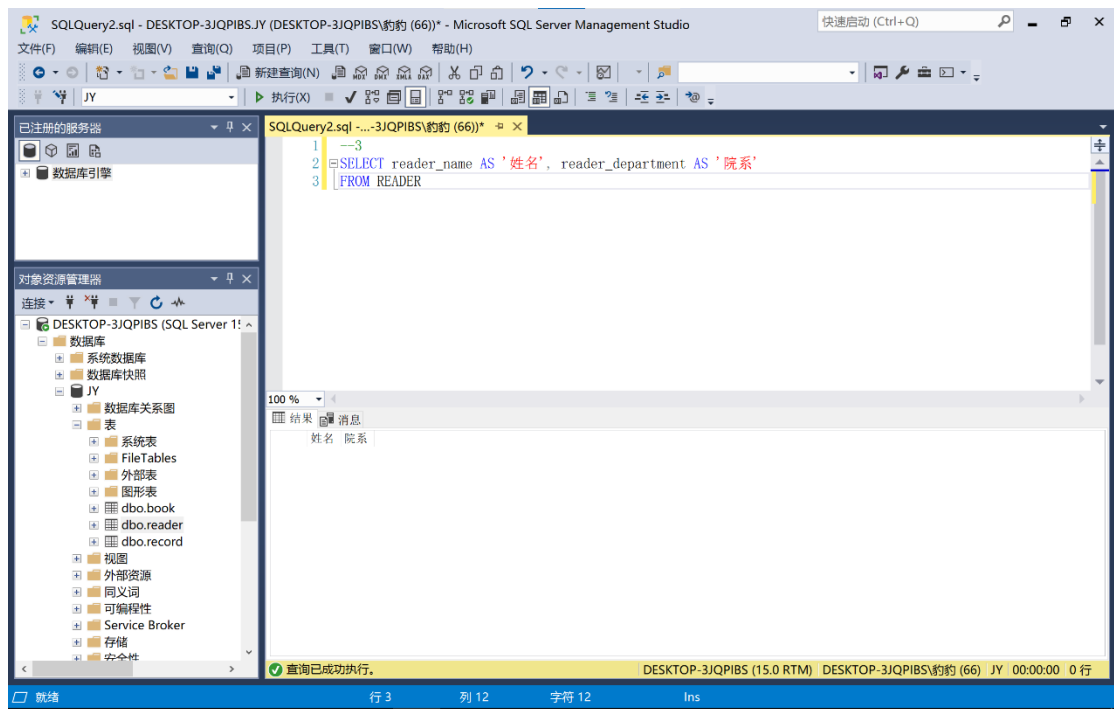
```
1 --2
2 SELECT reader_name, reader_department
3 FROM READER
```

The results pane displays the following data:

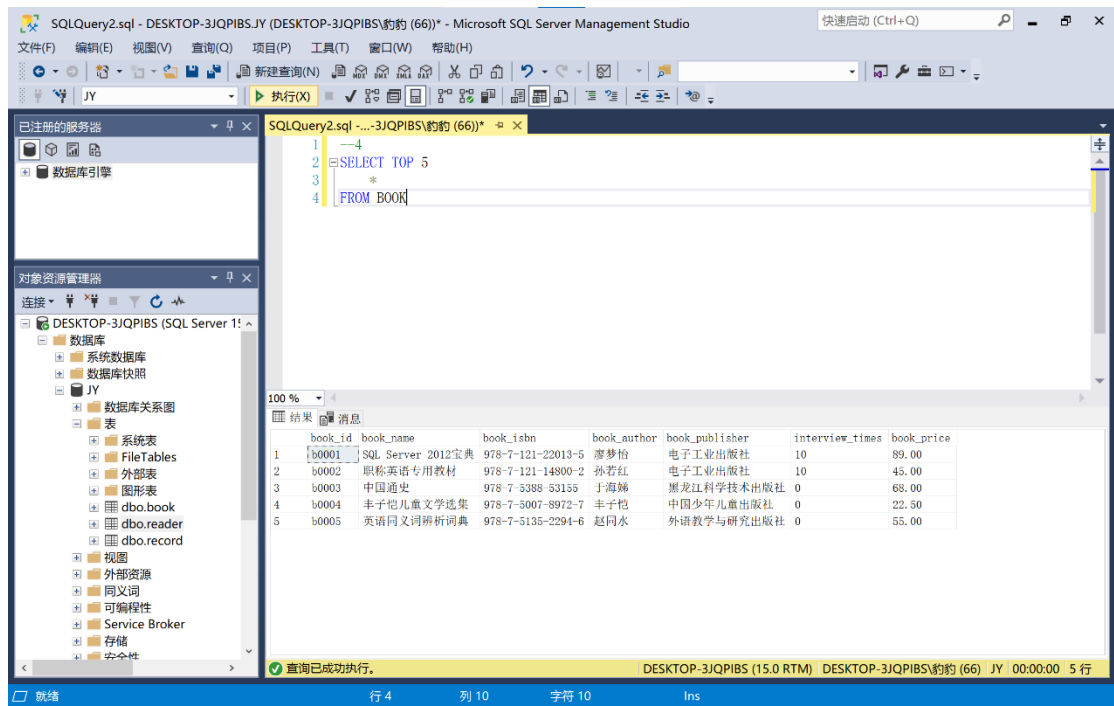
reader_name	reader_department
-------------	-------------------

The status bar at the bottom indicates: 查询已成功执行。 DESKTOP-3JQPIBS (15.0 RTM) DESKTOP-3JQPIBS\约约 (66) JY 00:00:00 0 行

3. 查询读者表 reader 的读者姓名和所在系，要求查询结果显示为“姓名”和“院系”

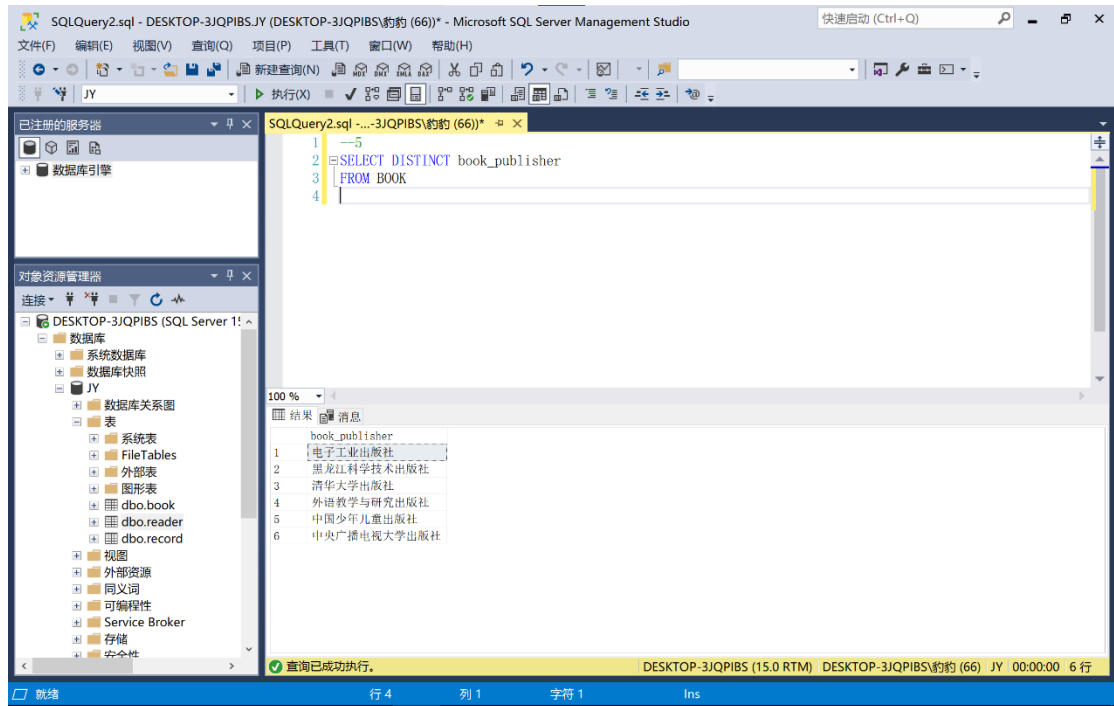


4. 查询图书表 book 的所有信息，只显示查询结果的前 5 行数据。

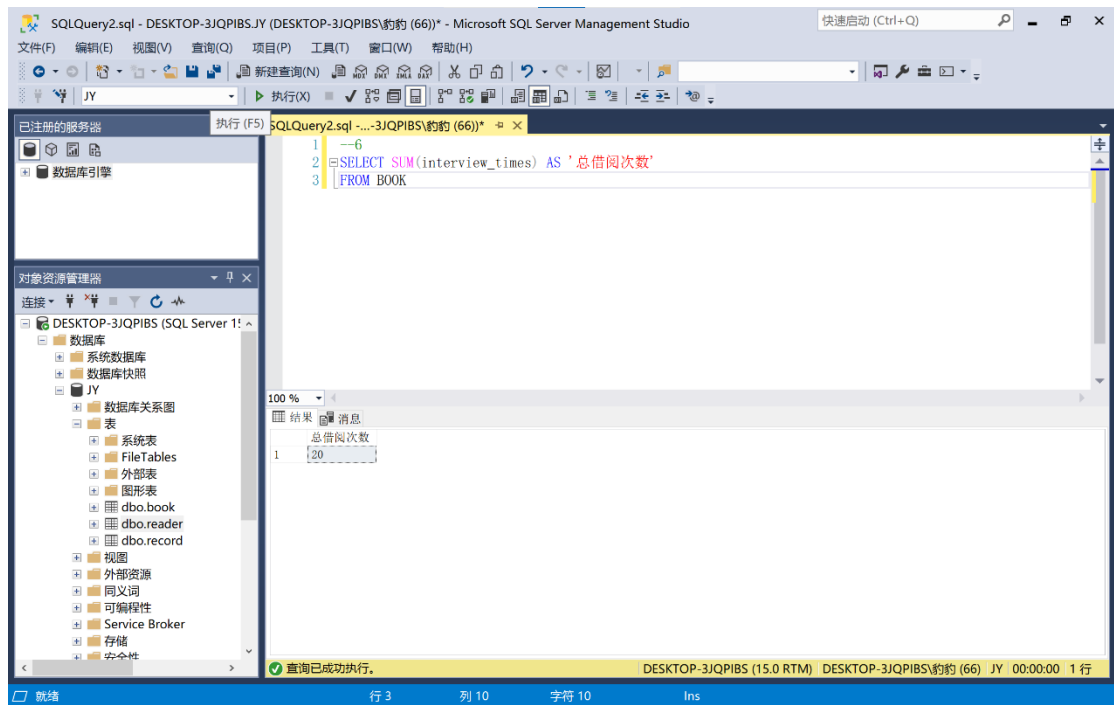




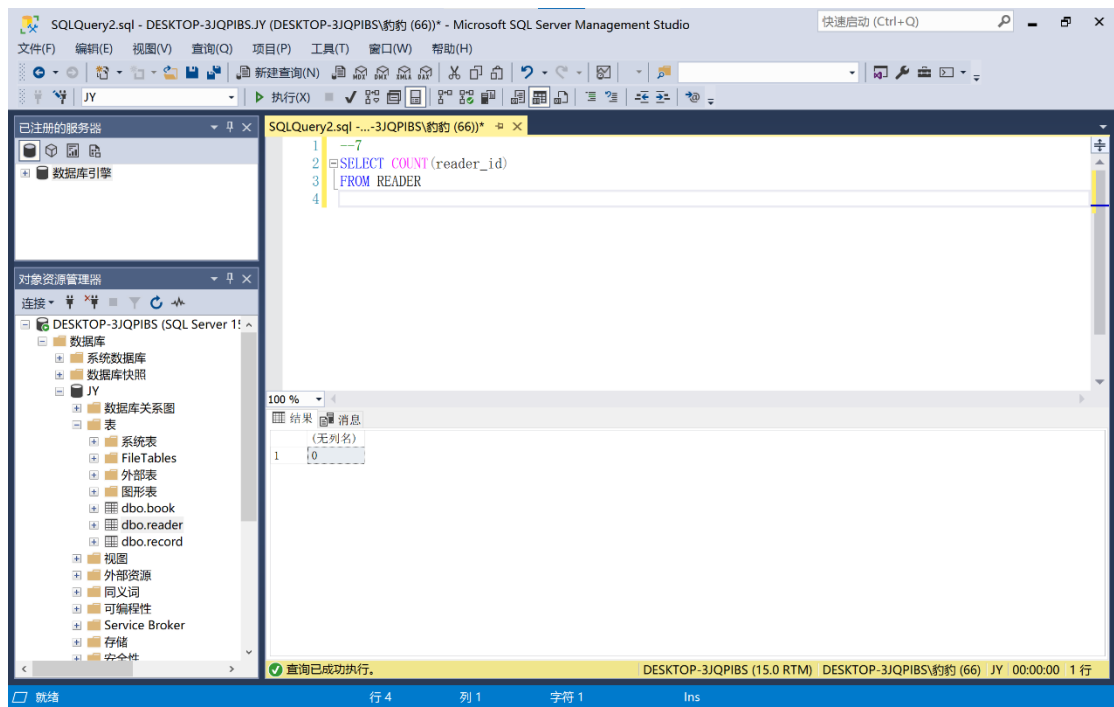
5. 查询图书表 book 的出版社名称，要求出版社名称不重复显示。



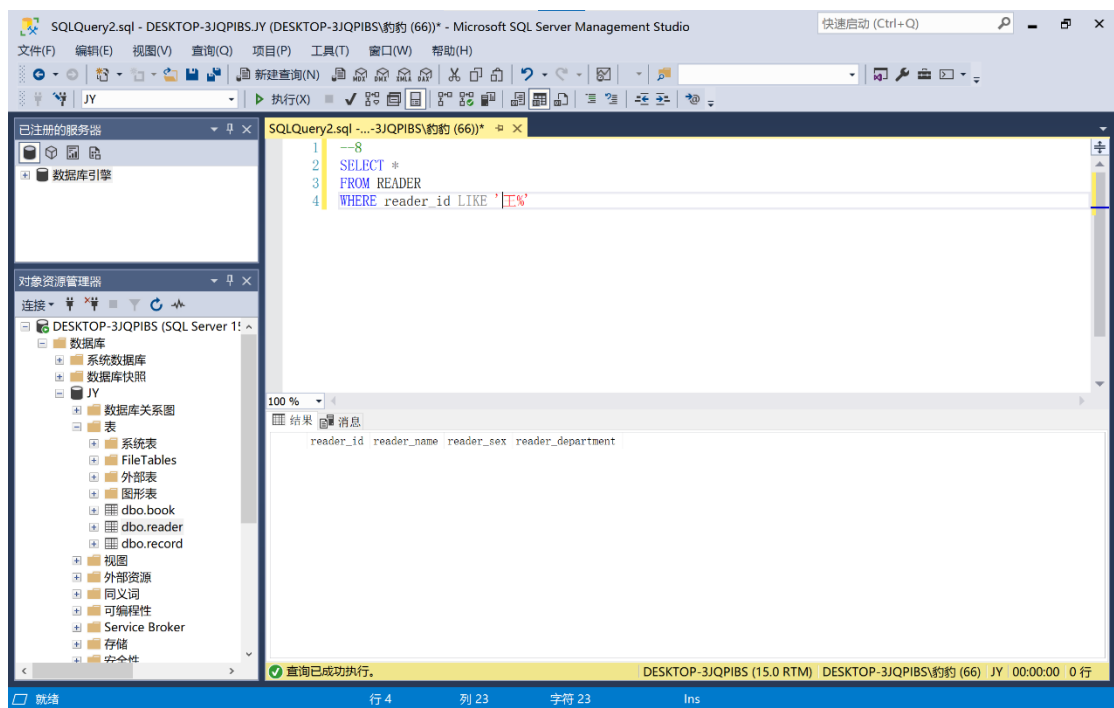
6. 查询数据表 book 的借阅次数的总和，要求查询结果显示为“总借阅次数”



## 7. 在读者表 reader 中统计读者总人数



## 8. 查询读者表 reader 中所有姓“王”的读者的信息



## 9. 查询图书表 record 中 notes 列值为空的记录

The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor contains the following SQL code:

```
1 --9
2 SELECT *
3 FROM RECORD
4 WHERE notes IS NULL
```

The results pane displays the following data:

reader_id	book_id	borrow_date	return_date	notes
r0001	b0003	2014-01-12	2014-01-12	NULL
r0001	b0005	2014-01-26	2014-06-21	NULL
r0004	b0001	2014-03-02	2014-04-20	NULL
r0004	b0008	2014-03-26	2014-05-28	NULL
r0006	b0001	2014-04-16	2014-07-11	NULL
r0007	b0006	2014-05-08	2014-09-17	NULL
r0008	b0007	2014-08-15	2014-10-21	NULL
r0008	b0008	2014-06-29	2014-08-29	NULL

## 10. 查询图书表 book 中所有图书信息，并按借阅次数由高到低进行排序

The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor contains the following SQL code:

```
1 --10
2 SELECT *
3 FROM BOOK
4 ORDER BY interview_times DESC
```

The results pane displays the following data:

book_id	book_name	book_isbn	book_author	book_publisher	interview_times	book_price
b0001	SQL Server 2012宝典	978-7-121-22013-5	廖梦怡	电子工业出版社	10	89.00
b0002	职称英语专用教材	978-7-121-14800-2	孙若红	电子工业出版社	10	45.00
b0003	中国通史	978-7-5388-53155	于海娜	黑龙江科学技术出版社	0	68.00
b0004	丰子恺儿童文学选集	978-7-5007-8972-7	丰子恺	中国少年儿童出版社	0	22.50
b0005	英语同义词辨析词典	978-7-5135-2294-6	赵同水	外语教学与研究出版社	0	55.00
b0006	数据库基础与应用	978-7-304-06430-3	徐承凯	中央广播电视大学出版社	0	35.00
b0007	微积分初步	978-7-304-03742-0	赵坚	中央广播电视大学出版社	0	17.00
b0008	ASP.NET入门到精通	978-7-302-28753-7	明日科技	清华大学出版社	0	89.80

## 11.在图书表 book 中统计各出版社出版图书的总数

The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor contains the following SQL code:

```
1 --11
2 SELECT book_publisher, COUNT(book_id)
3 FROM BOOK
4 GROUP BY book_publisher
```

The results pane displays the following data:

book_publisher	(无列名)
电子工业出版社	2
黑龙江科学技术出版社	1
清华大学出版社	1
外语教学与研究出版社	1
中国少年儿童出版社	1
中央广播电视大学出版社	2

The status bar at the bottom indicates that the query was successfully executed, returning 6 rows.

## 12.在图书表 book 中查询图书借阅次数在 20 次以上, 而且出版图书总数大于 1 的出版社。

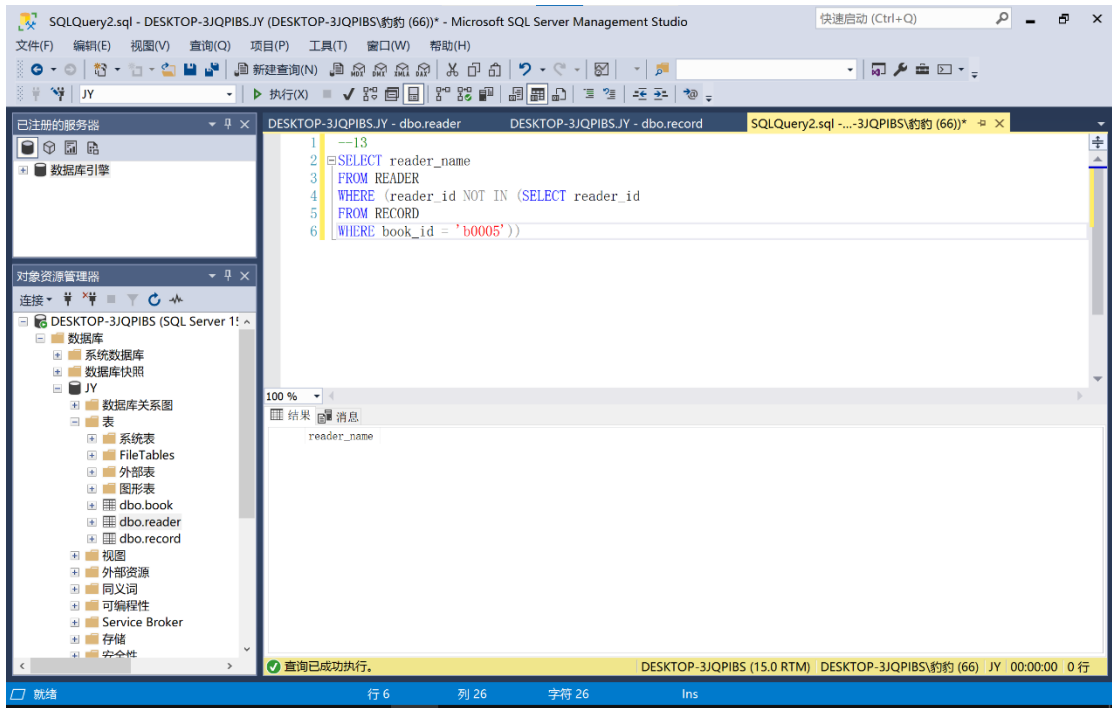
The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor contains the following SQL code:

```
1 --12
2 SELECT book_publisher
3 FROM
4 (SELECT book_publisher, COUNT(book_id) AS book_num, SUM(interview_times) AS interview_times
5 FROM BOOK
6 GROUP BY book_publisher) AS T
7 WHERE (book_num > 1 AND interview_times > 20)
```

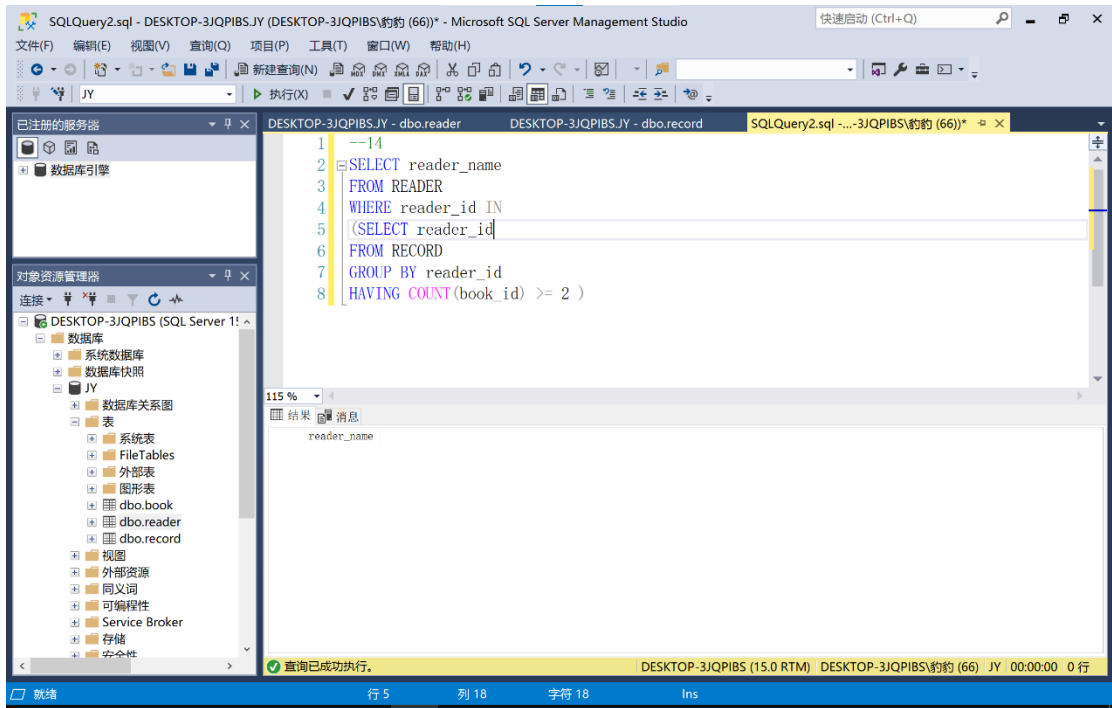
The results pane is currently empty, showing only the column header 'book\_publisher'.

The status bar at the bottom indicates that the query was successfully executed, returning 0 rows.

13.查询没有借阅编号为“b0005”这本书的读者姓名



14.查询借阅至少 2 本书的读者姓名



### 15. 查询借阅次数最多和最少的图书编号

The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor contains the following SQL code:

```
1  --15
2  SELECT book_id as book_id_max
3  FROM BOOK
4  WHERE interview_times IN (SELECT MAX(interview_times) FROM BOOK)
5
6  SELECT book_id as book_id_min
7  FROM BOOK
8  WHERE interview_times IN (SELECT MIN(interview_times) FROM BOOK)
```

The results pane shows two tables:

book_id_max	
1	b0001
2	b0002

book_id_min	
1	b0003
2	b0004
3	b0005
4	b0006
5	b0007
6	b0008

The status bar at the bottom indicates the query was executed successfully, returning 8 rows.

### 16. 查询借阅过图书编号为“b0002”图书的读者编号和姓名

The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor contains the following SQL code:

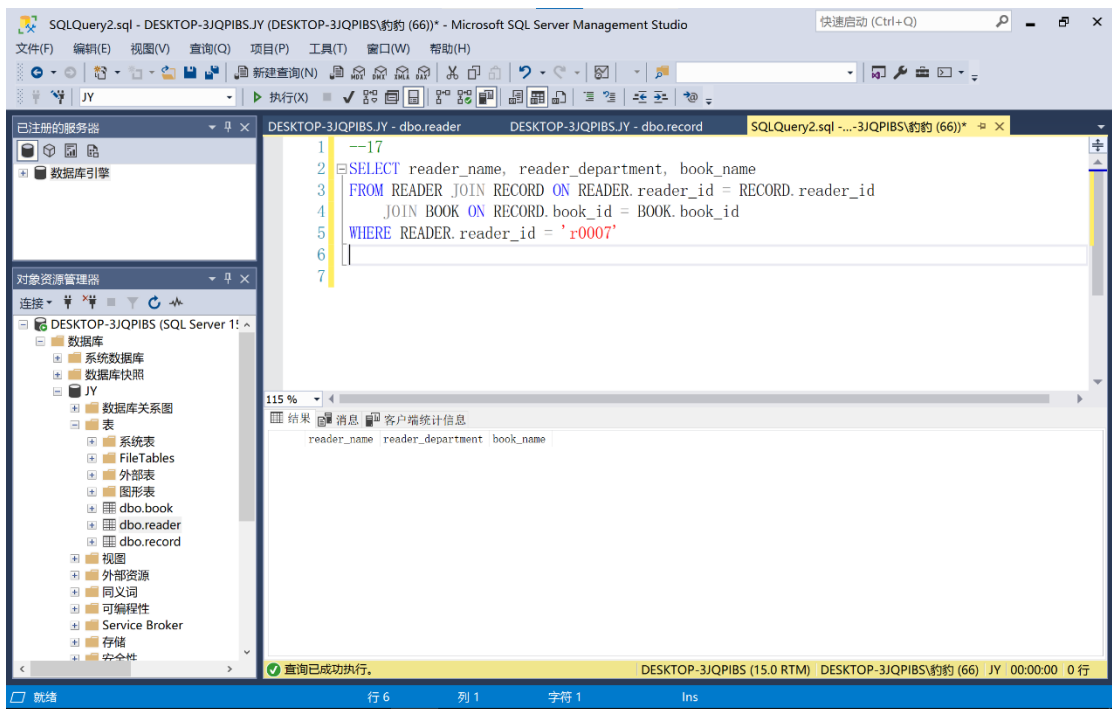
```
1  --16
2  SELECT reader_name, reader_id
3  FROM READER
4  WHERE reader_id IN (SELECT reader_id
5  FROM RECORD
6  WHERE book_id = 'b0002' )
7
```

The results pane shows a table with the following columns:

reader_name	reader_id
-------------	-----------

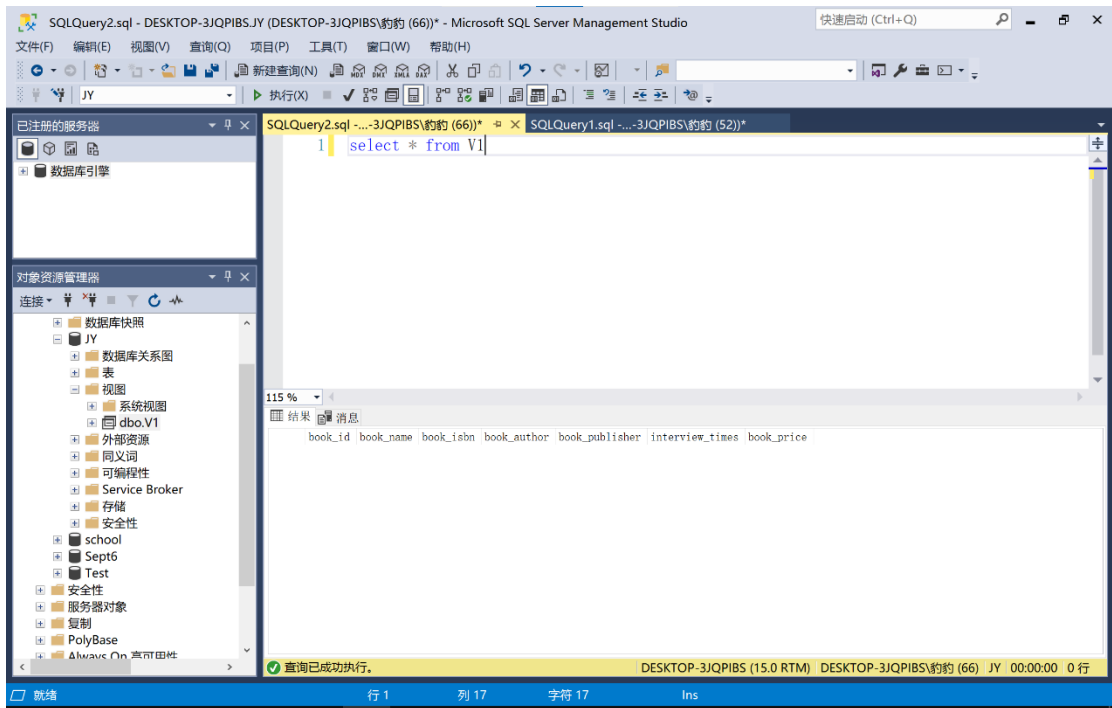
The status bar at the bottom indicates the query was executed successfully, returning 0 rows.

17.查询编号为“r0007”读者的姓名、所在院系以及所借阅图书的书名。

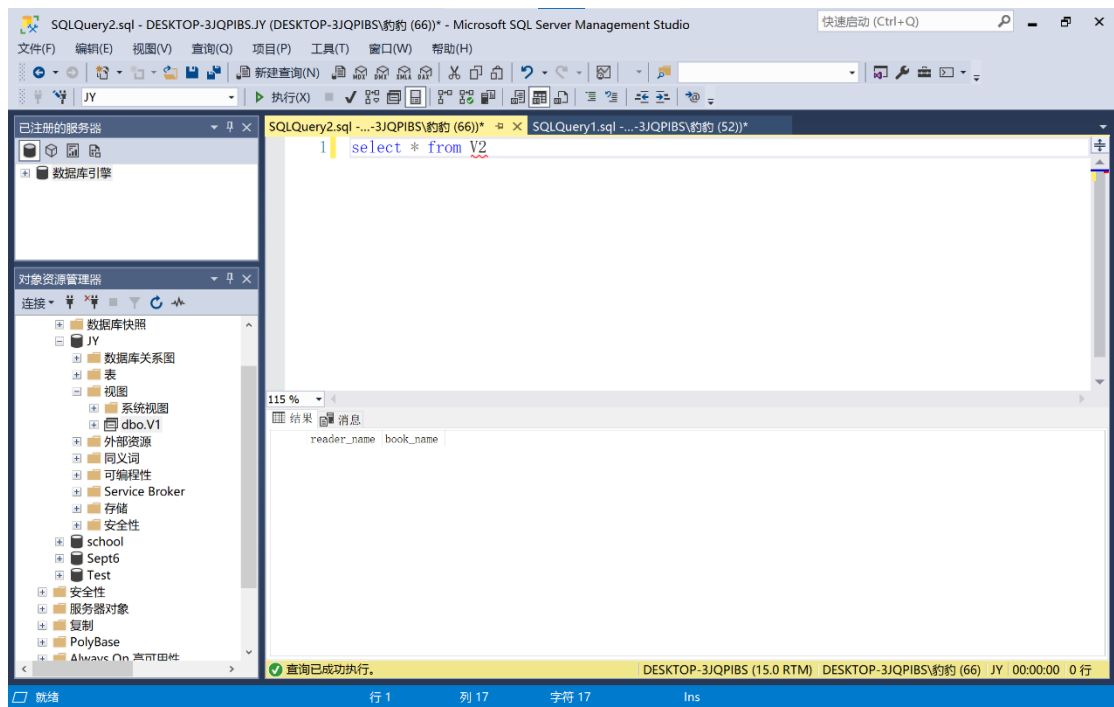


视图

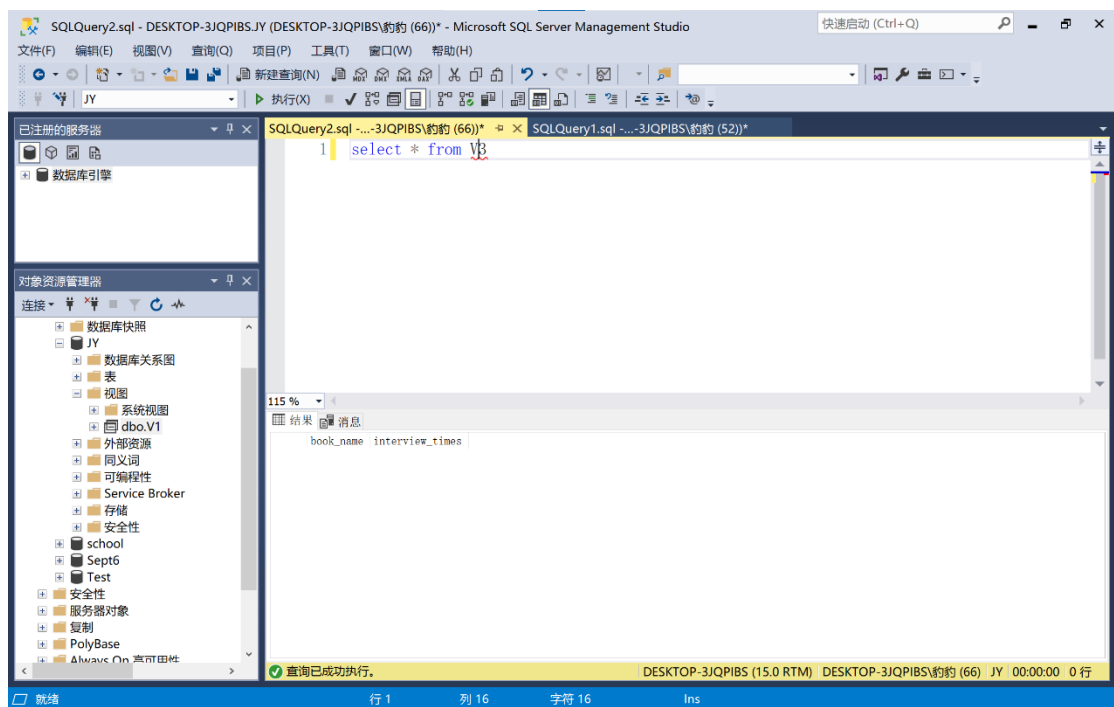
1. 创建视图，用于查看借阅次数大于 30 次的图书信息。



## 2. 创建视图，用于查看借阅了图书的读者姓名，以及他们借阅的图书名称。

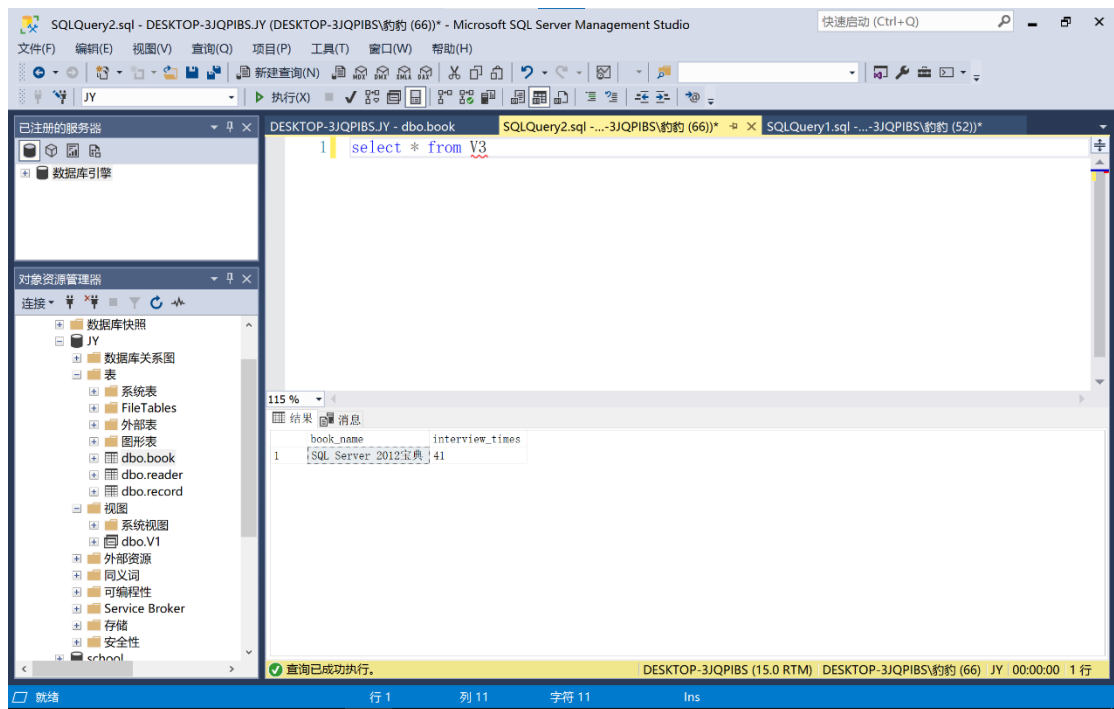


## 3. 创建视图，用于查询访问次数大于 30 次的图书名称和借阅次数，并在视图中为列指定别名。

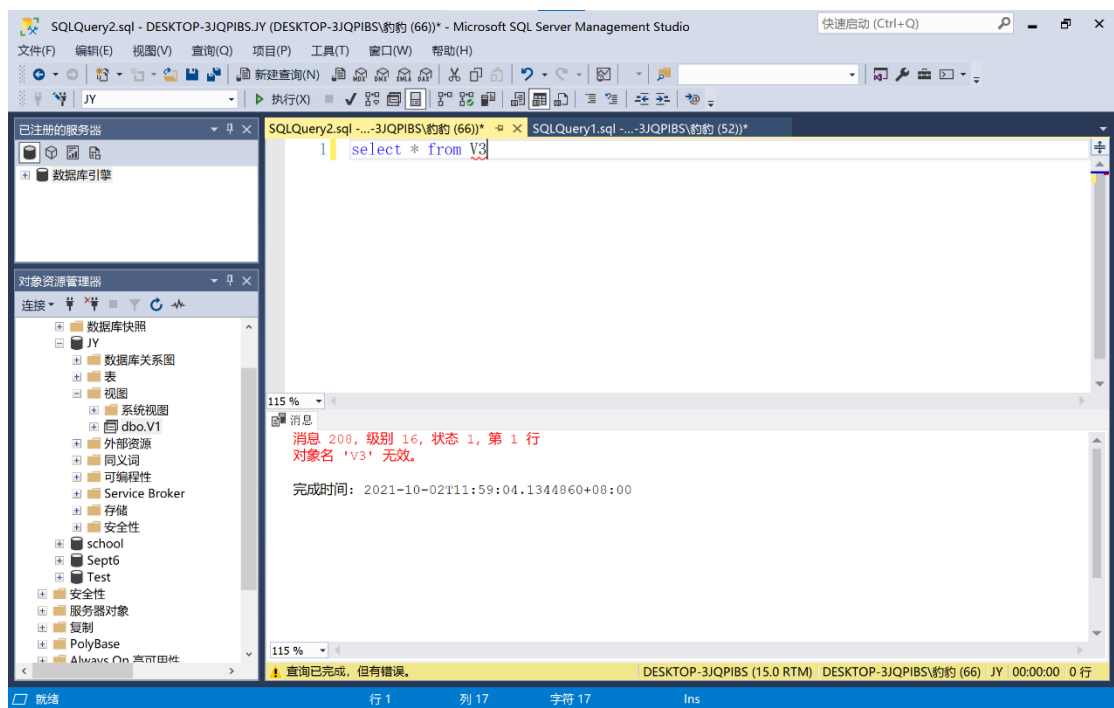




#### 4. 上面的视图可更新吗？通过 SQL 更新语句来验证。



#### 5. 删除上面视图



### 3. 实验总结

在实验课上的练习和这次实验中，我进一步熟悉了 SSMS 的基本操作，了解了 SQL Server 关于嵌套查询和创建与修改视图的一些简单语句和语法，培养了对数据库系统原理的兴趣。希望能再接再厉，在今后的学习中掌握更多知识与技能。