实 验 报 告：数据库应用软件实验

课程名称：　　数据库应用软件实验

学　院：　　计算机科学与工程学院

专　业：　　软件工程　　班级：　　2018 级　班

姓　名：　　KAFLE SAMRAT　　学号：　　201801060933

2021 年 06 月 016 日

山东科技大学教务处制

实验一 创建数据库和表实验

1.实验目的与要求

（1）了解 SQL Server 数据库的逻辑结构和物理结构；

（2）了解表的结构特点；

（3）了解 SQL Server 的基本数据类型；

（4）了解空值的概念；

（5）学会在对象资源管理器中创建数据库和表；

（6）学会使用 T-SQL 语句创建数据库和表。

2.实验内容

（1）创建一个新的数据库

创建用于企业管理的员工管理数据库，数据库名为 YGGL。

数据库 YGGL 的逻辑文件初始大小为 10MB，最大大小为 50MB，数据库自动增长，增长方式是按 5%比例增长。日志文件初始为 2MB，最大可增长到 5MB，按 1MB 增长。

数据库的逻辑文件名和物理文件名均采用默认值。事务日志的逻辑文件名和

物理文件名也均采用默认值。要求分别使用对象资源管理器和 T-SQL 命令完成数据库的创建工作。

Create an employee management database for enterprise management. The database is named YGGL, which contains employee information, department information, and employee salary information. The database YGGL contains the following 3 tables.

(1) Employees: employee natural information table.

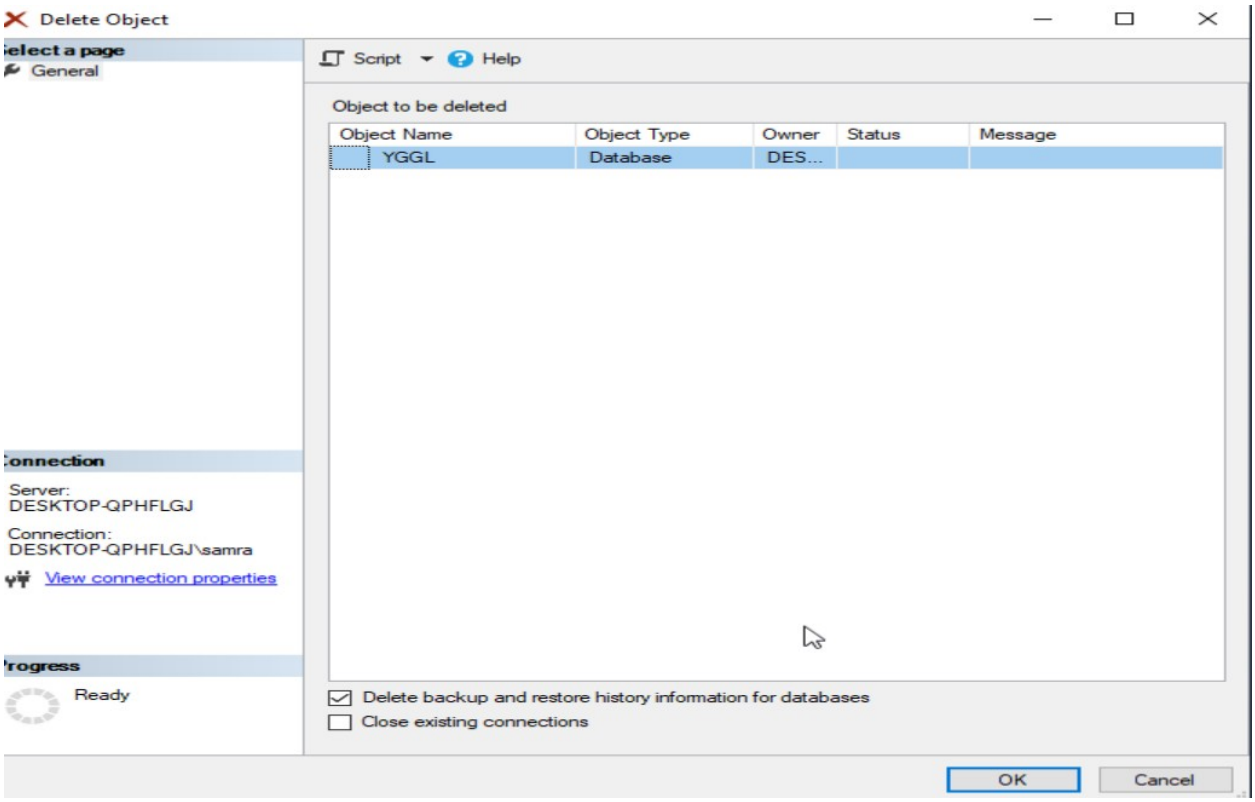
(2) Departments: Department information table.

(3) Salary: employee salary table.

Create database YGGL in Object Explorer

Delete the YGGL database created in the Object Explorer

In the Object Explorer, select the database YGGL.

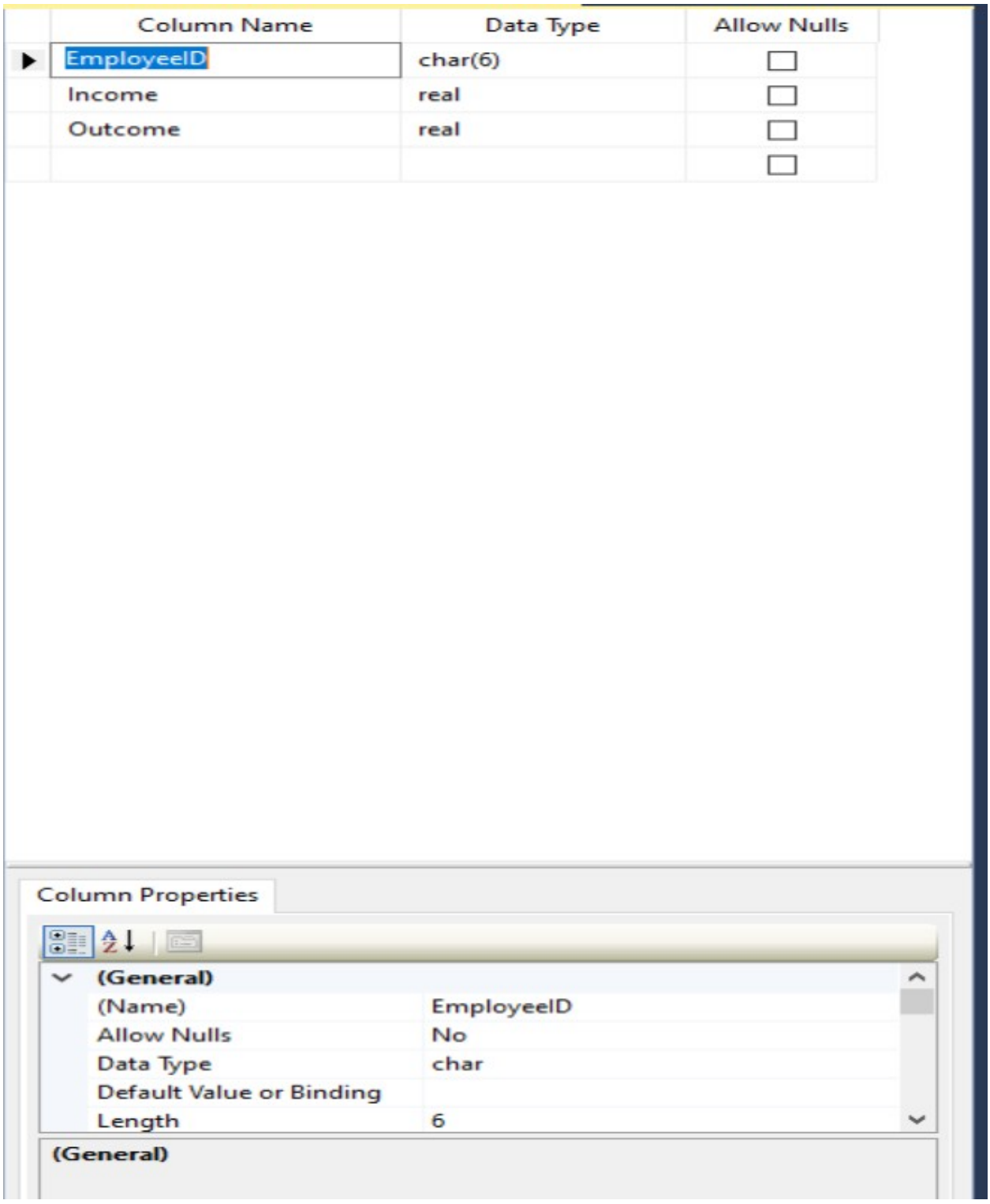


Use T-SQL statement to create database YGGL

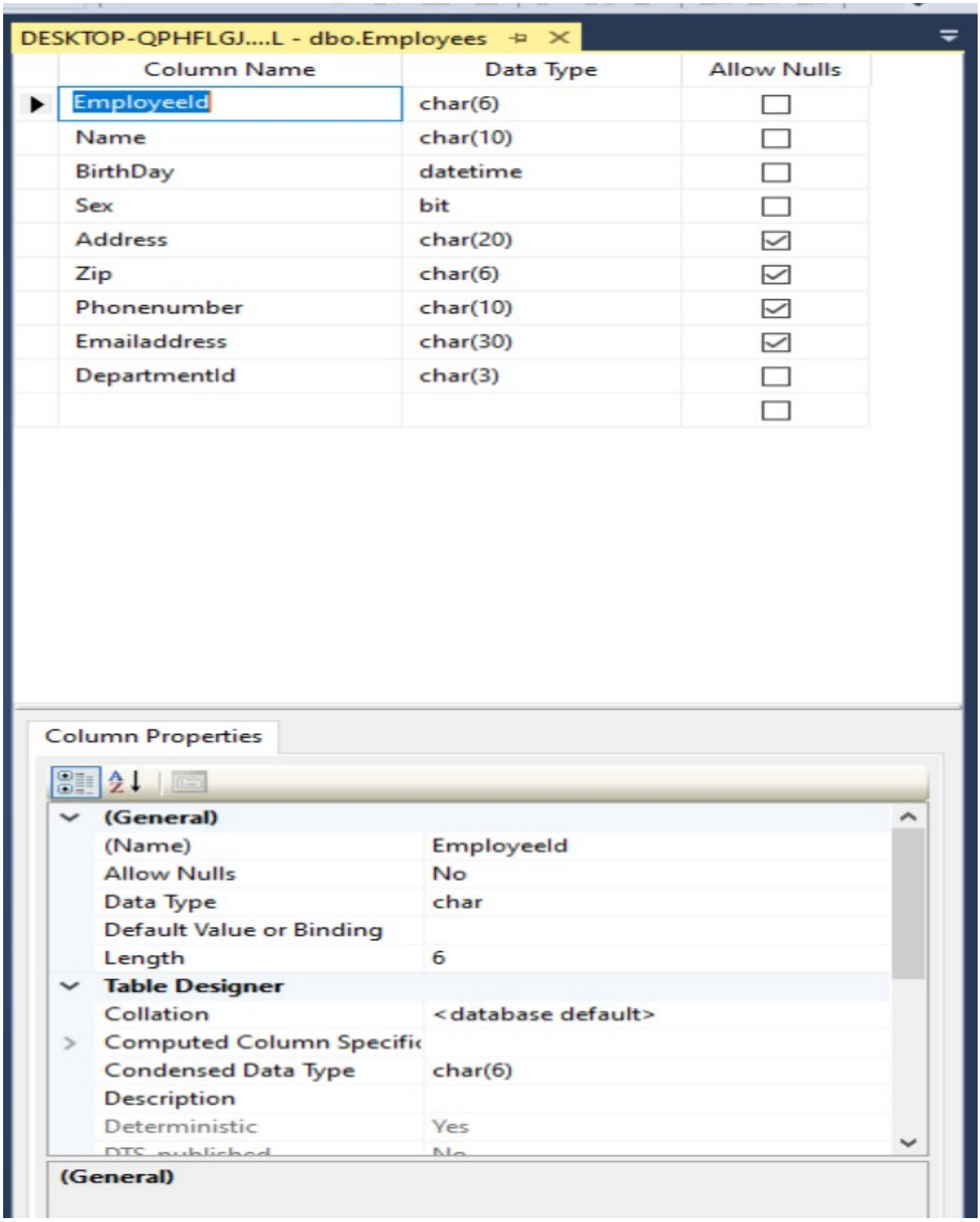
```
CREATE DATABASE YGGL
ON
(
     NAME='YGGL',
     FILENAME='G:\SQL\yggl.mdf',
     SIZE=10MB,
     MAXSIZE=50MB,
     FILEGROWTH=5%
)
LOG ON
(NAME='yggl_log',
FILENAME='G:\SQL\yggl_log.ldf',
SIZE=2MB,
MAXSIZE=5MB,
FILEGROWTH=1MB)
```

.

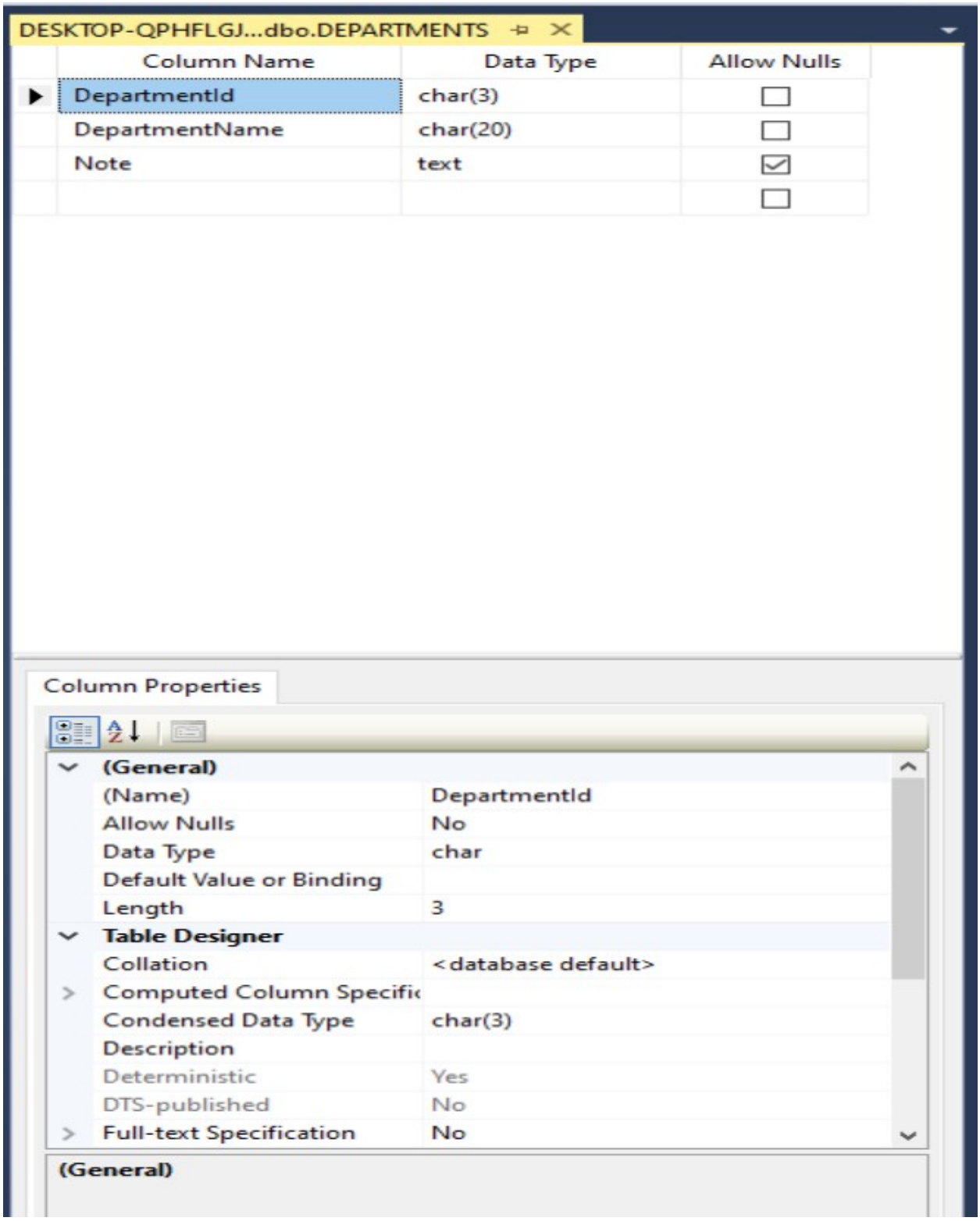
Create the Employees, Departments and Salary tables respectively in the Object Explorer


Column Name
Data Type
Allow Nulls
EmployeeID
char(6)
Income
real
Outcome
real
Column Properties
(General)
(Name)
EmployeeID
Allow Nulls
No
Data Type
char
Default Value or Binding
Length
6
(General)

.


DESKTOP-QPHFLGJ....L - dbo.Employees
Column Name
Data Type
Allow Nulls
EmployeeId
char(6)
Name
char(10)
BirthDay
datetime
Sex
bit
Address
char(20)
Zip
char(6)
Phonenumber
char(10)
Emailaddress
char(30)
DepartmentId
char(3)
Column Properties
(General)
(Name)
EmployeeId
Allow Nulls
No
Data Type
char
Default Value or Binding
Length
6
Table Designer
Collation
<database default>
Computed Column Specifi
Condensed Data Type
char(6)
Description
Deterministic
Yes
(General)

.


DESKTOP-QPHFLGJ...dbo.DEPARTMENTS
Column Name
Data Type
Allow Nulls
DepartmentId
char(3)
DepartmentName
char(20)
Note
text
Column Properties
(General)
(Name)
DepartmentId
Allow Nulls
No
Data Type
char
Default Value or Binding
Length
3
Table Designer
Collation
<database default>
Computed Column Specifi
Condensed Data Type
char(3)
Description
Deterministic
Yes
DTS-published
No
Full-text Specification
No
(General)

（2）在创建好的员工管理数据库（YGGL）中创建数据表

考虑到员工管理数据库 YGGL 要求包含员工信息、部门信息以及员工的薪水信息，所以数据库 YGGL 应包含下列三个表：Employees（员工自然信息）表 、Department（部门信息）表和 Salary（员工薪水情况）表。

要求分别使用对象资源管理器和 T-SQL 语句来完成数据表的创建工作。

Add data to the database YGGL table in the object explorer

EmployeeID	Income	Outcome
1	2100.8	123.09
1008	1582.8	88.09
111006	1987.7	185.09
51	2130.8	103.09
1	2150.8	153.09
1	2170.8	193.09
1	2110.8	173.09
NULL	*NULL*	*NULL*

DepartmentId	DepartmentNa...	Note
123	hello	jhsdjkf
234	hello1	kjsd
125	hello2	sdbfnsd
126	hello3	sdjhfjdk
127	hello4	jhhsfhjsvd
128	hello5	mdfnngkjdf
2	?????	
1	?????	
1	?????	
1	?????	
5	?????	
5	???	
3	???	
5	???	
NULL	*NULL*	*NULL*

.

DESKTOP-QPHFLGJ....L - dbo.Employees | SQLQuery1.sql - DE...PHFLGJ\samra (52))*

EmployeeId	Name	BirthDay	Sex	Address	Zip	Phonenumber	Emailaddress	DepartmentId
121	sam	1999-12-26 00:0...	False	np	dl	8683654387	hsgadhvsdhgv ...	123
000001	ram	1999-12-24 00:0...	True	dsfasfdad	vdafd	9792632	gdsjgfjhsd@chi...	124
011108	ran	1999-12-23 00:0...	False	sdfdsfs	dassdf	1232144	jhsdjs@china.c...	123
132324	scdcds	1999-12-12 00:0...	True	sfdsfds	dsfsd	233255	dsfds@chuna.c...	345
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

.

Modify database YGGL table data in Object Explorer

DepartmentId	DepartmentNa...	Note
123	hello	jhsdjkf
234	hello1	kjsd
126	hello3	sdjhfjdk
127	hello4	jhhsfhjsvd
128	hello5	mdfnngkjdf
2	?????	
1	?????	
1	?????	
1	?????	
5	?????	
5	???	
3	???	
5	???	
NULL	*NULL*	*NULL*

.

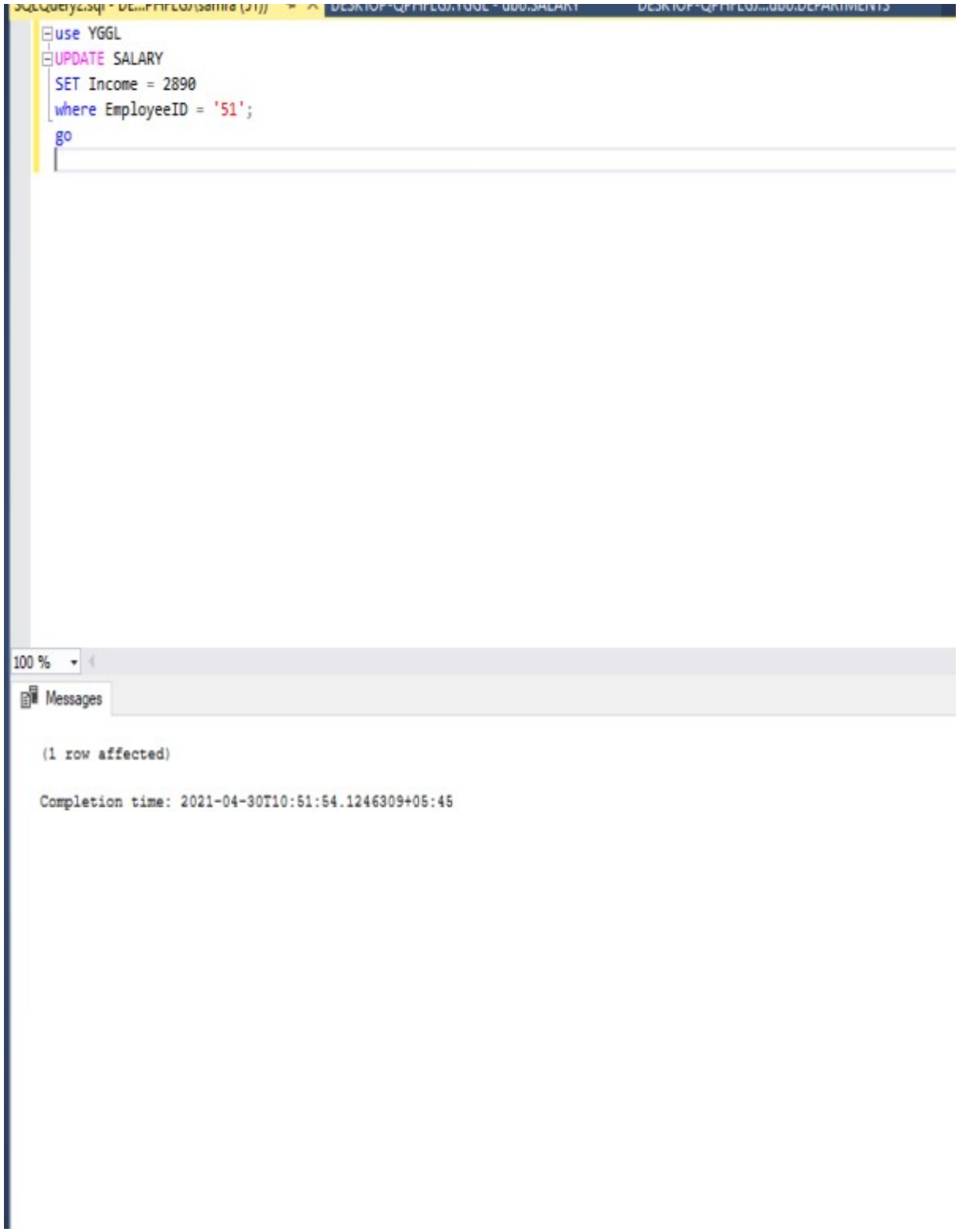
EmployeeId	Name	BirthDay	Sex	Address	Zip	Phonenumber	Emailaddress	DepartmentId
121	sam	1999-12-26 00:0...	False	np	dl	8683654387	hsgadhvsdhgv ...	123
000001	ram	1999-12-24 00:0...	True	dsfasfdad	vdafd	9792632	gdsjgfjhsd@chi...	124
011108	ran	1999-12-23 00:0...	False	sdfdsfs	dassdf	1232144	jhsdjs@china.c...	123
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

.

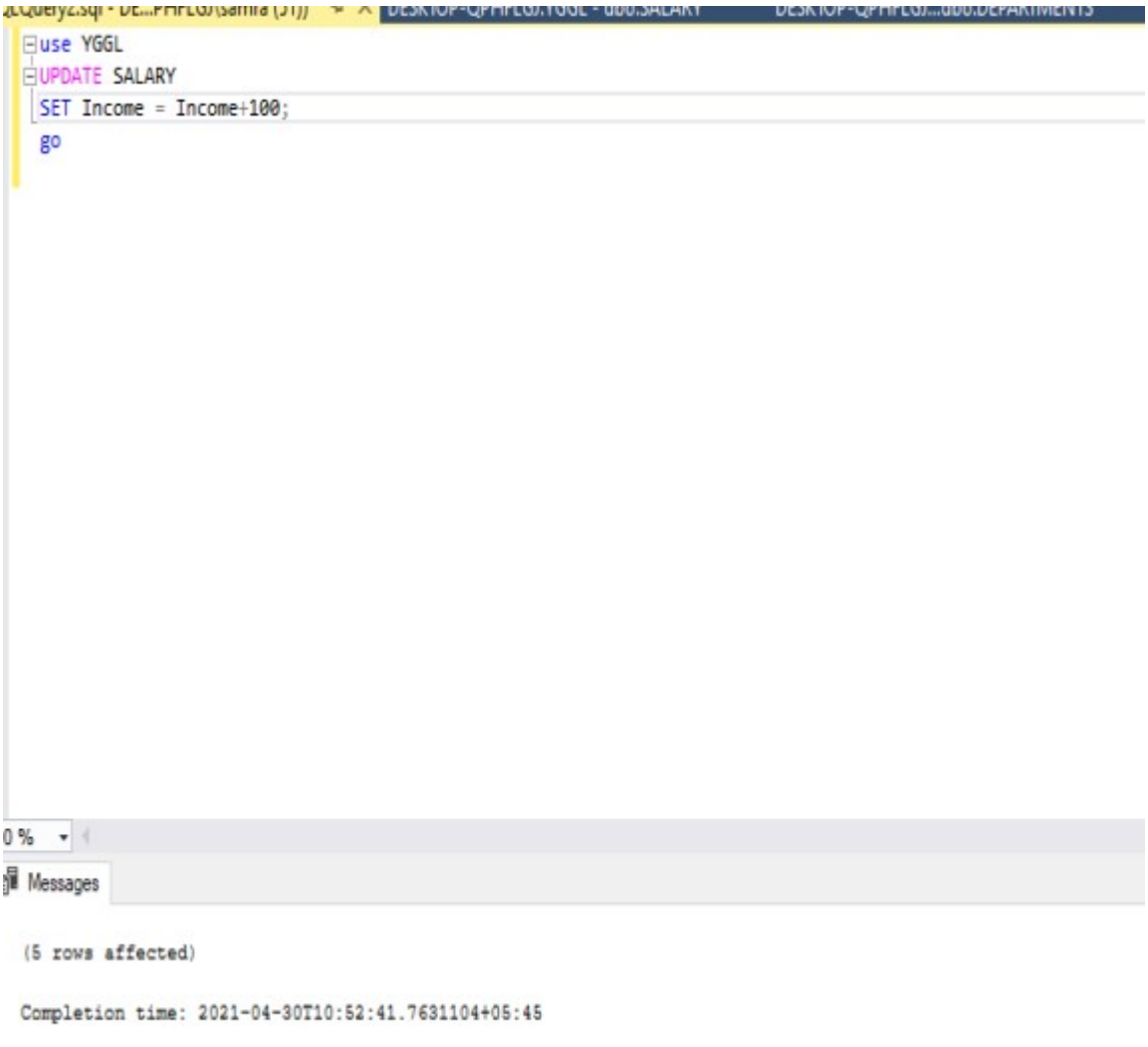
EmployeeID	Income	Outcome
1	2100.8	123.09
51	2130.8	103.09
1	2150.8	153.09
1	2170.8	193.09
1	2110.8	173.09
NULL	NULL	NULL

.

Use the T-SQL command to modify the field value of a record in the Salary table.


use YGGL
UPDATE SALARY
SET Income = 2890
where EmployeeID = '51';
go
100 %
Messages
(1 row affected)
Completion time: 2021-04-30T10:51:54.1246309+05:45

.


use YGGL
UPDATE SALARY
SET Income = Income+100;
go
0 %
Messages
(5 rows affected)
Completion time: 2021-04-30T10:52:41.7631104+05:45

EmployeeID	Income	Outcome
1	2100.8	123.09
51	2130.8	103.09
1	2150.8	153.09
1	2170.8	193.09
1	2110.8	173.09
NULL	NULL	NULL

实验二：表数据插入、删除、修改

一.实验目的与要求

（1）学会在对象资源管理器中对数据库表进行插入、修改和删除数据操作；

（2）学会使用 T-SQL 语句对数据库表进行插入、修改和删除数据操作；

（3）了解数据更新操作时要注意数据完整性。

二.实验内容

（1）按约定方式对数据库、数据表进行改名，建立表间的关联关系；

（2）分别使用对象资源管理器和 T-SQL 语句，向实验一中建立的数据库 YGGL 的三个表 Employees、Department 和 Salary 中插入多行数据记录（Employees 和 Salary 表不低于 50 行，Department 不低于 8 行），数据要具有合理性、规范性，然后修改和删除一些记录。使用 T-SQL 语句进行有限制的修改和删除。

三.实验内容和结果

Modify data to the database using object explorer.

Before adding:

table Employee:

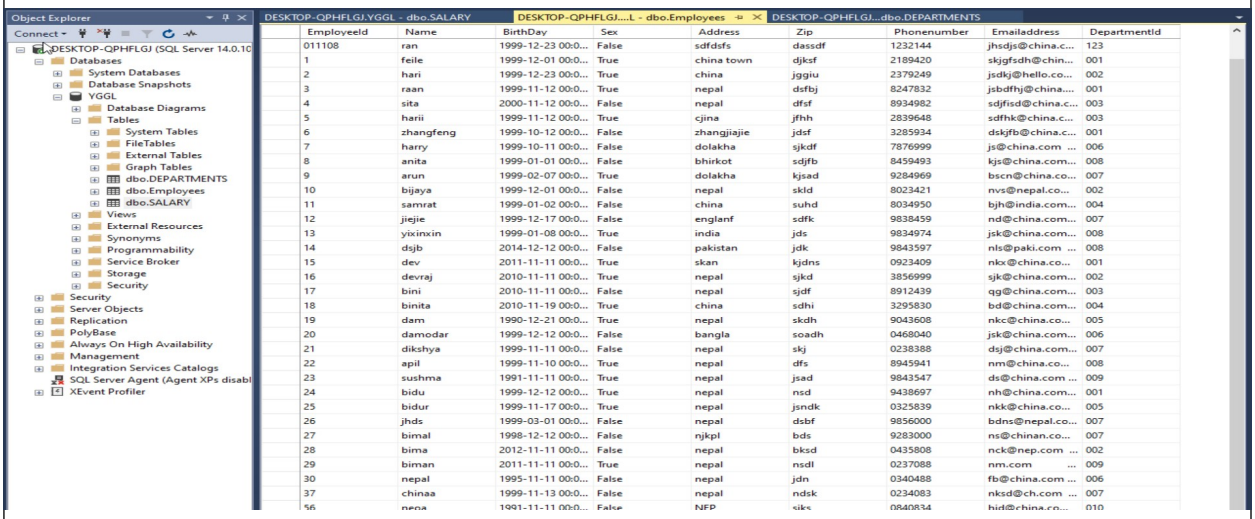
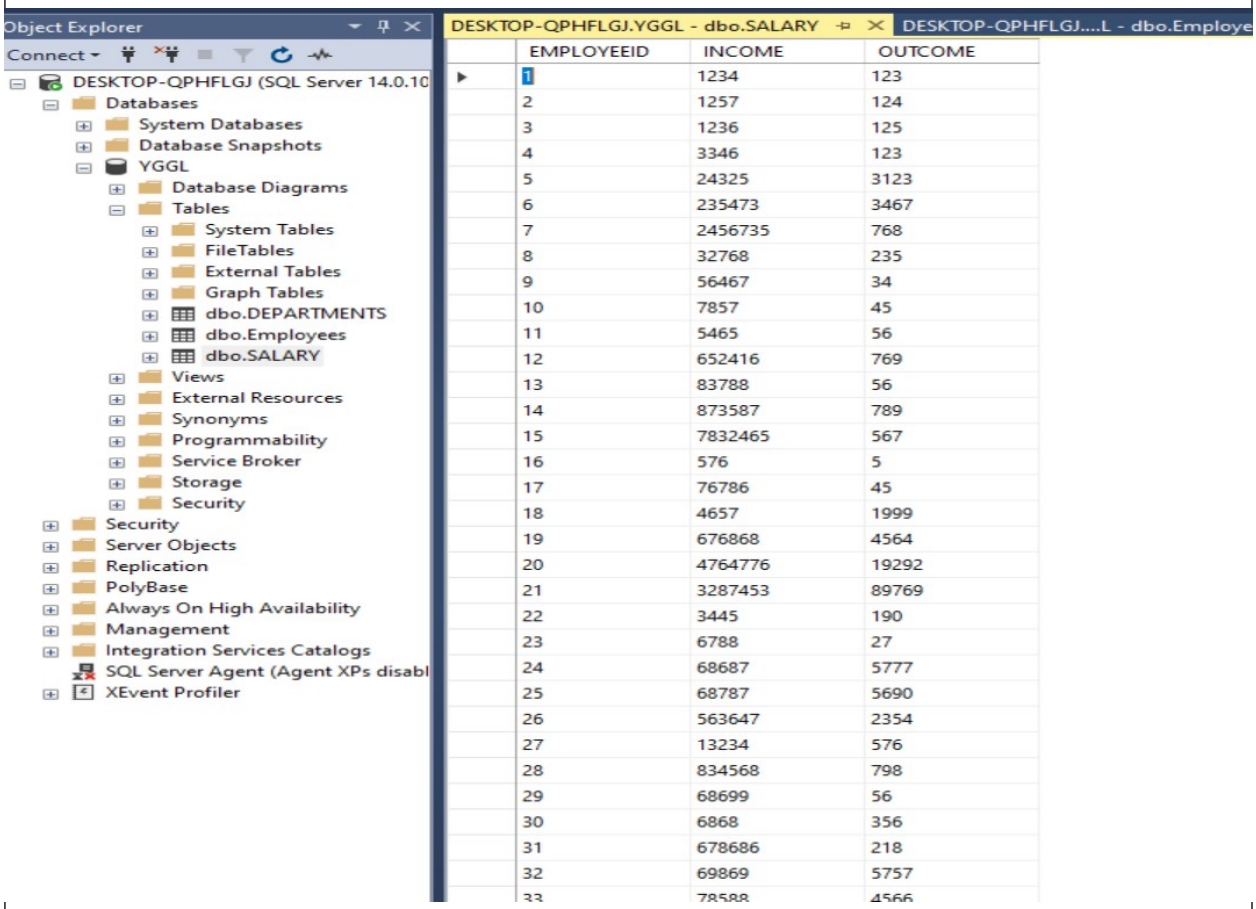
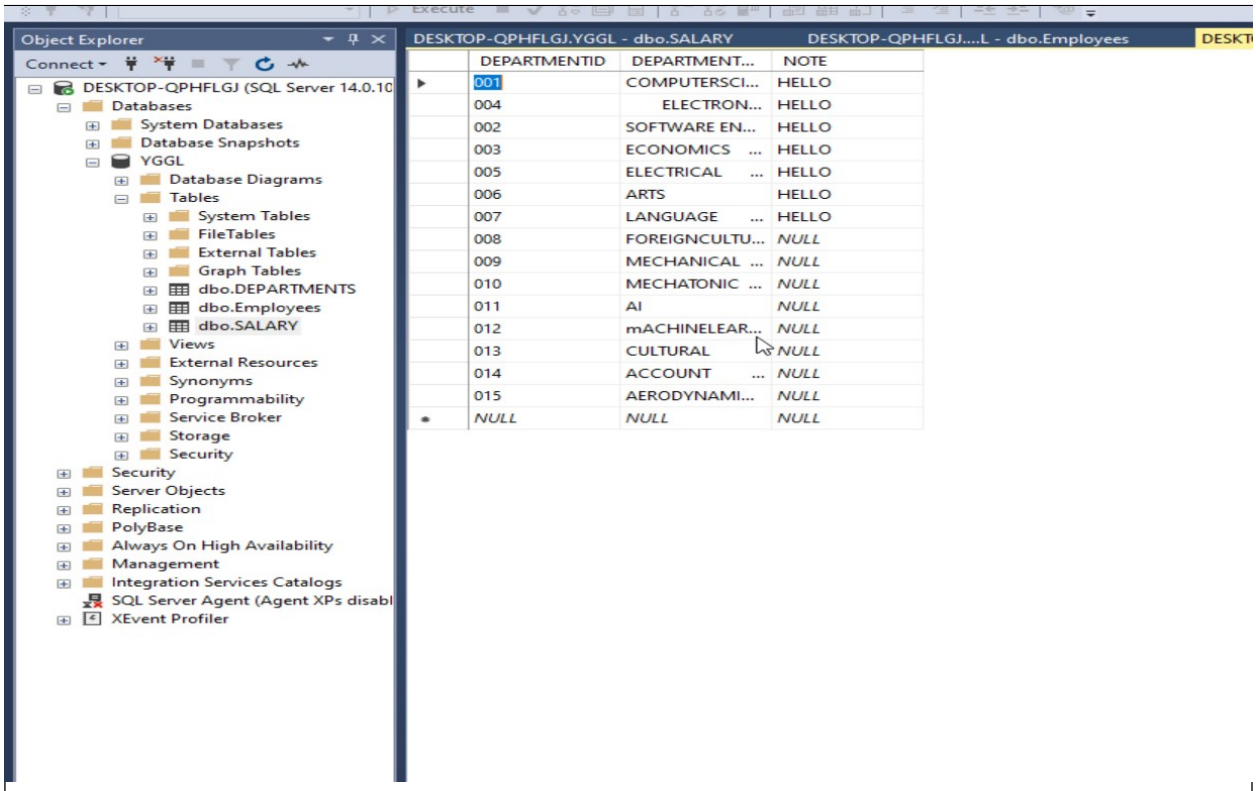


table Salary:



EMPLOYEEID	INCOME	OUTCOME
1	1234	123
2	1257	124
3	1236	125
4	3346	123
5	24325	3123
6	235473	3467
7	2456735	768
8	32768	235
9	56467	34
10	7857	45
11	5465	56
12	652416	769
13	83788	56
14	873587	789
15	7832465	567
16	576	5
17	76786	45
18	4657	1999
19	676868	4564
20	4764776	19292
21	3287453	89769
22	3445	190
23	6788	27
24	68687	5777
25	68787	5690
26	563647	2354
27	13234	576
28	834568	798
29	68699	56
30	6868	356
31	678686	218
32	69869	5757
33	78588	4566

table Departments:



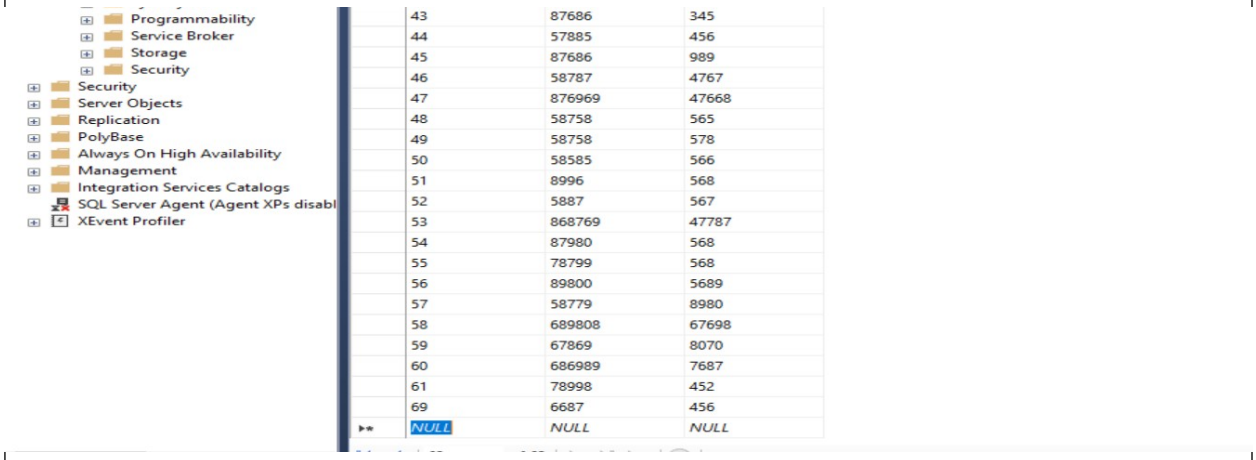
DEPARTMENTID	DEPARTMENT...	NOTE
001	COMPUTERSCI...	HELLO
004	ELECTRON...	HELLO
002	SOFTWARE EN...	HELLO
003	ECONOMICS ...	HELLO
005	ELECTRICAL ...	HELLO
006	ARTS	HELLO
007	LANGUAGE ...	HELLO
008	FOREIGNCULTU...	NULL
009	MECHANICAL ...	NULL
010	MECHATONIC ...	NULL
011	AI	NULL
012	mACHINELEAR...	NULL
013	CULTURAL	NULL
014	ACCOUNT ...	NULL
015	AERODYNAMI...	NULL
NULL	NULL	NULL

After adding:

table Employee:

19	dam	1990-12-21 00:0...	True	nepal	skdh	9043608	nkc@china.co...	005
20	damodar	1999-12-12 00:0...	False	bangla	soadh	0468040	jsk@china.com...	006
21	dikshya	1999-11-11 00:0...	False	nepal	skj	0238388	dsj@china.com...	007
22	apil	1999-11-10 00:0...	True	nepal	dfs	8945941	nm@china.co...	008
23	sushma	1991-11-11 00:0...	True	nepal	jsad	9843547	ds@china.com ...	009
24	bidu	1999-12-12 00:0...	True	nepal	nsd	9438697	nh@china.com...	001
25	bidur	1999-11-17 00:0...	True	nepal	jsndk	0325839	nkk@china.co...	005
26	jhds	1999-03-01 00:0...	False	nepal	dsbf	9856000	bdns@nepal.co...	007
27	bimal	1998-12-12 00:0...	False	njkpl	bds	9283000	ns@chinan.co...	007
28	bima	2012-11-11 00:0...	False	nepal	bksd	0435808	nck@nep.com ...	002
29	biman	2011-11-11 00:0...	True	nepal	nsdl	0237088	nm.com ...	009
30	nepal	1995-11-11 00:0...	False	nepal	jdn	0340488	fb@china.com ...	006
37	chinaa	1999-11-13 00:0...	False	nepal	ndsk	0234083	nksd@ch.com ...	007
56	neoa	1991-11-11 00:0...	False	NEP	sjks	0840834	hjd@china.co...	010
60	neo	1999-02-04 00:0...	False	ndu	uuehr	9374953	hfd@china.co...	011
45	ndhs	1999-01-11 00:0...	False	kjdg	jksdh	9348439	bc@china.com ...	012
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

table Salary:



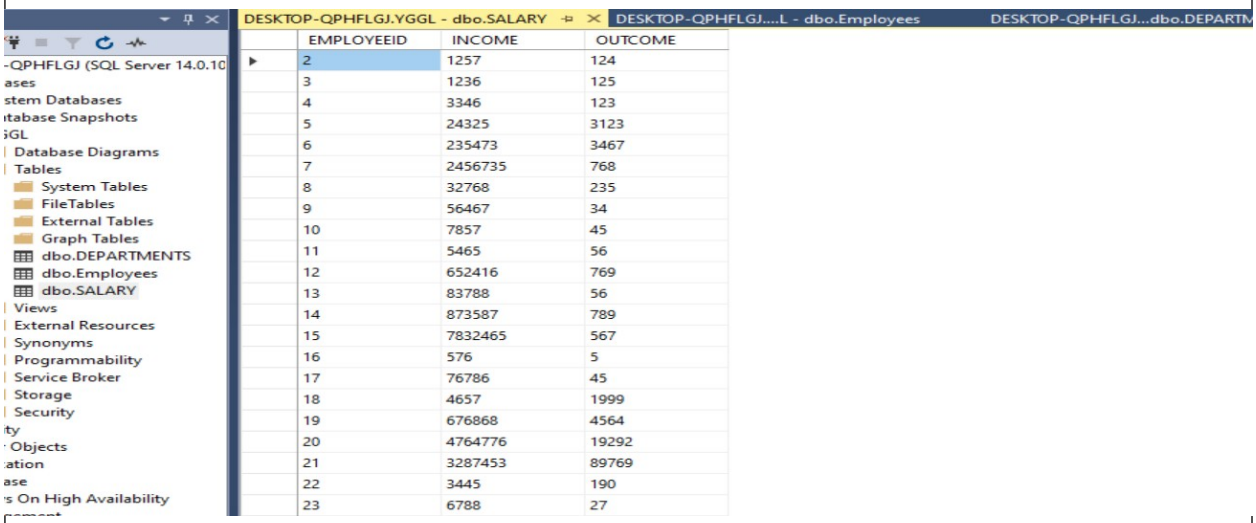
43	87686	345
44	57885	456
45	87686	989
46	58787	4767
47	876969	47668
48	58758	565
49	58758	578
50	58585	566
51	8996	568
52	5887	567
53	868769	47787
54	87980	568
55	78799	568
56	89800	5689
57	58779	8980
58	689808	67698
59	67869	8070
60	686989	7687
61	78998	452
69	6687	456
NULL	NULL	NULL

After Deleting:

table Employee:

Employeeid	Name	BirthDay	Sex	Address	Zip	Phonenumber	Emailaddress	Departmentid
121	sam	1999-12-26 00:0...	False	np	dl	8683654387	hsgadhvsdhgv ...	123
000001	ram	1999-12-24 00:0...	True	dsfasfdad	vdafd	9792632	gdsjgfjhsd@chi...	124
011108	ran	1999-12-23 00:0...	False	sdfdsfs	dassdf	1232144	jhsdjs@china.c...	123
2	hari	1999-12-23 00:0...	True	china	jggiu	2379249	jsdkj@hello.co...	002
3	raan	1999-11-12 00:0...	True	nepal	dsfbj	8247832	jsbdfhj@china....	001
4	sita	2000-11-12 00:0...	False	nepal	dfsf	8934982	sdjfisd@china.c...	003
5	harii	1999-11-12 00:0...	True	cjina	jfhh	2839648	sdfhk@china.c...	003
6	zhangfeng	1999-10-12 00:0...	False	zhangjiajie	jdsf	3285934	dskjfb@china.c...	001
7	harry	1999-10-11 00:0...	False	dolakha	sjkdf	7876999	js@china.com ...	006
8	anita	1999-01-01 00:0...	False	bhirkot	sdjfb	8459493	kjs@china.com...	008
9	arun	1999-02-07 00:0...	True	dolakha	kjsad	9284969	bscn@china.co...	007
10	bijaya	1999-12-01 00:0...	False	nepal	skld	8023421	nvs@nepal.co...	002
11	samrat	1999-01-02 00:0...	False	china	suhd	8034950	bjh@india.com...	004
12	jiejie	1999-12-17 00:0...	False	englanf	sdfk	9838459	nd@china.com...	007
13	yixinxin	1999-01-08 00:0...	True	india	jds	9834974	jsk@china.com...	008
14	dsjb	2014-12-12 00:0...	False	pakistan	jdk	9843597	nls@paki.com ...	008
15	dev	2011-11-11 00:0...	True	skan	kjdns	0923409	nkx@china.co...	001
16	devraj	2010-11-11 00:0...	True	nepal	sjkd	3856999	sjk@china.com...	002
17	bini	2010-11-11 00:0...	False	nepal	sjdf	8912439	qg@china.com...	003
18	binita	2010-11-19 00:0...	True	china	sdhi	3295830	bd@china.com...	004
19	dam	1990-12-21 00:0...	True	nepal	skdh	9043608	nkc@china.co...	005
20	damodar	1999-12-12 00:0...	False	bangla	soadh	0468040	jsk@china.com...	006
21	dikshya	1999-11-11 00:0...	False	nepal	skj	0238388	dsj@china.com...	007
22	apil	1999-11-10 00:0...	True	nepal	dfs	8945941	nm@china.co...	008

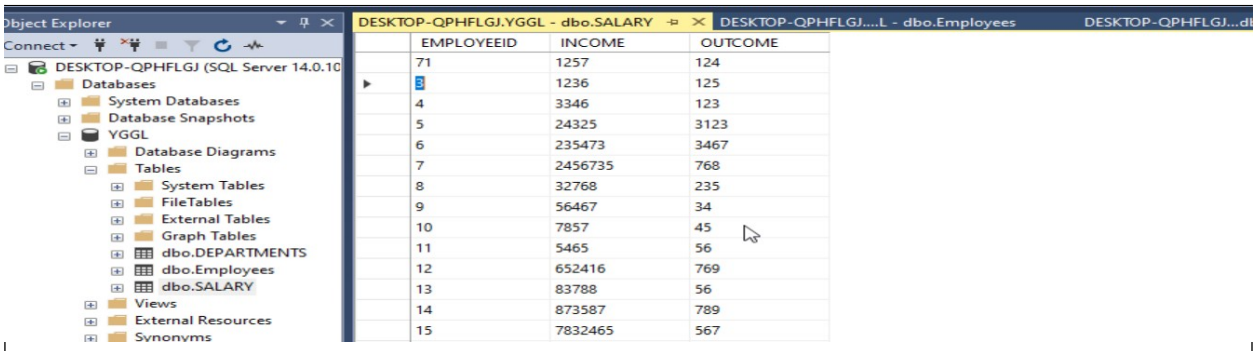
table Salary:



EMPLOYEEID	INCOME	OUTCOME
2	1257	124
3	1236	125
4	3346	123
5	24325	3123
6	235473	3467
7	2456735	768
8	32768	235
9	56467	34
10	7857	45
11	5465	56
12	652416	769
13	83788	56
14	873587	789
15	7832465	567
16	576	5
17	76786	45
18	4657	1999
19	676868	4564
20	4764776	19292
21	3287453	89769
22	3445	190
23	6788	27

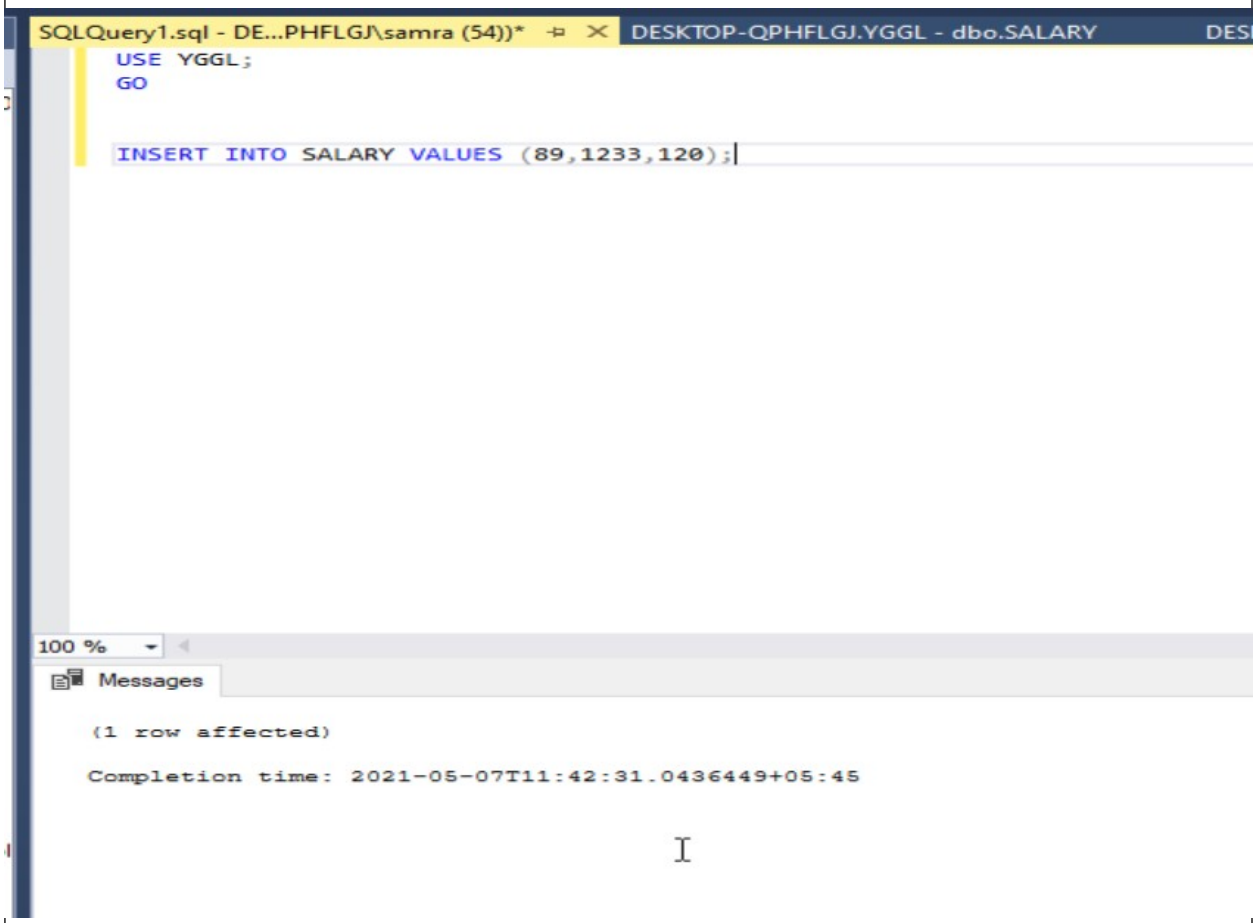
After Editing:

we edit employee id 2 to 71 in the table Salary:

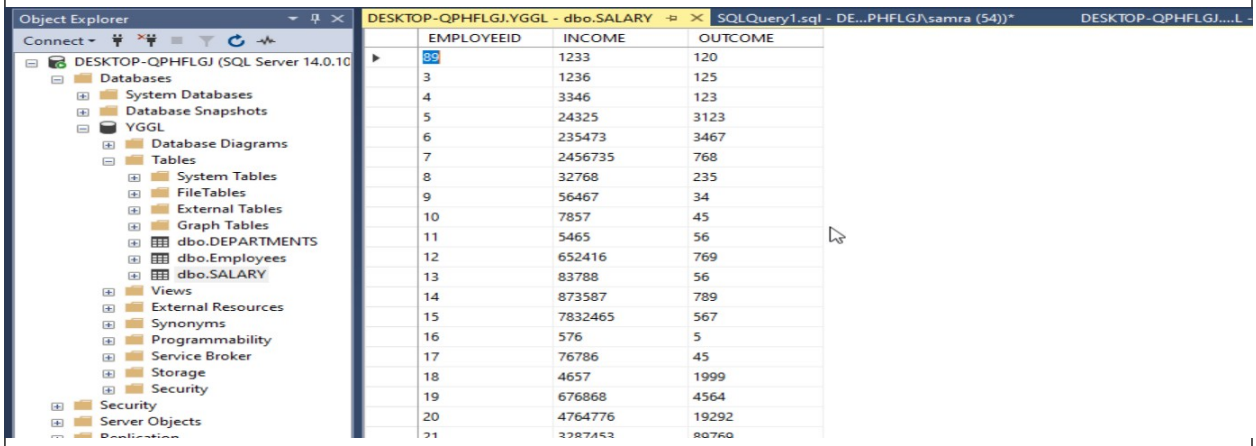


Modify data to the database using object T-SQL statements:

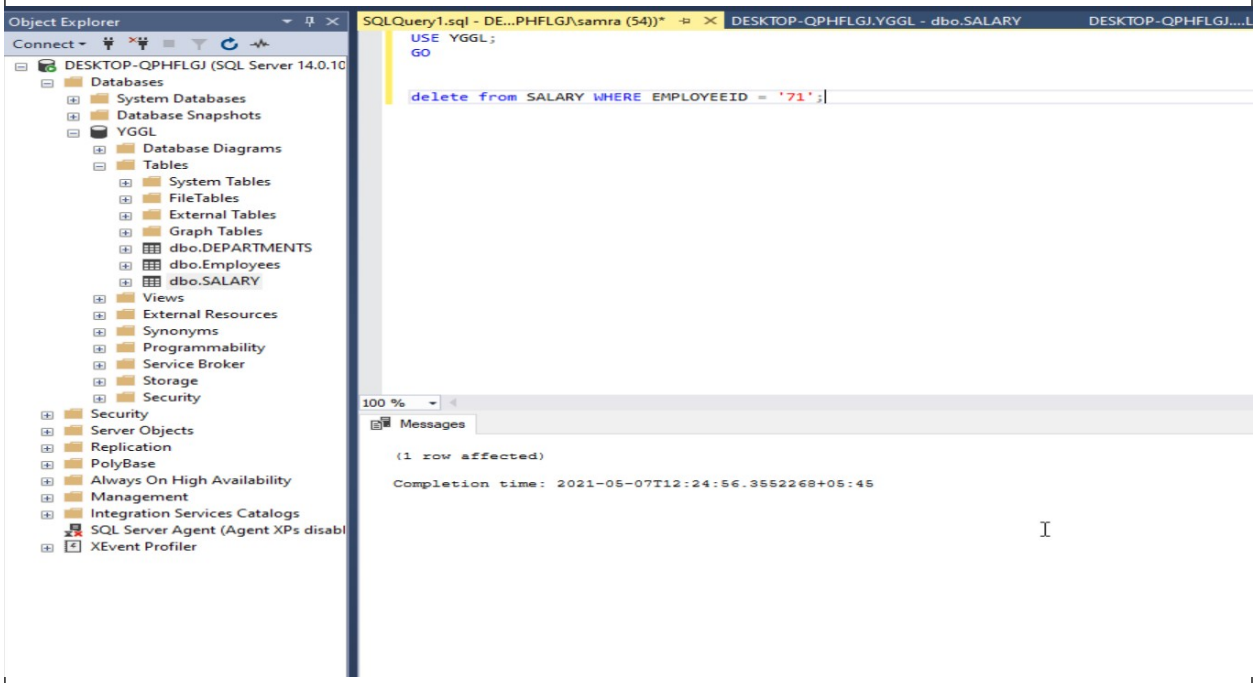
Adding data in table Salary:



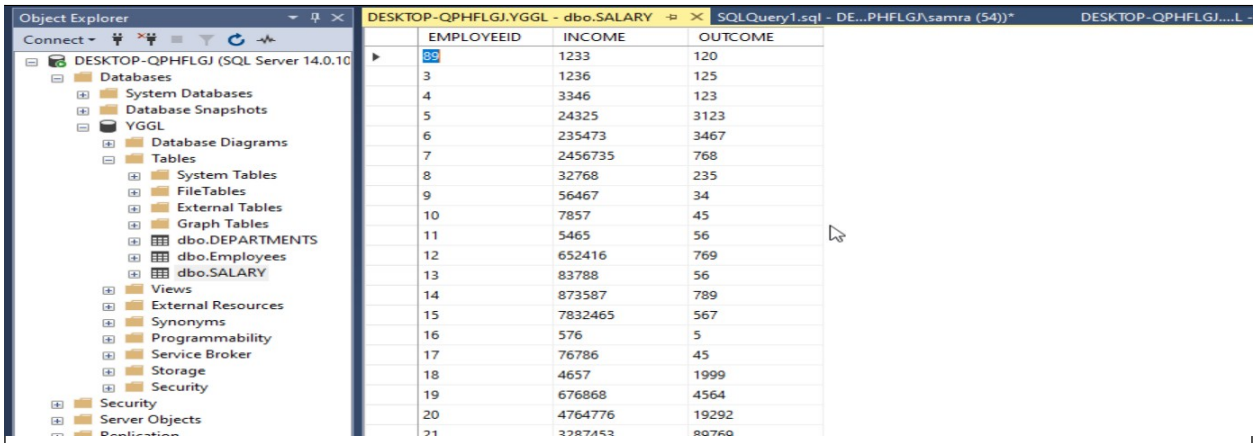
.



Remove data from table Salary:

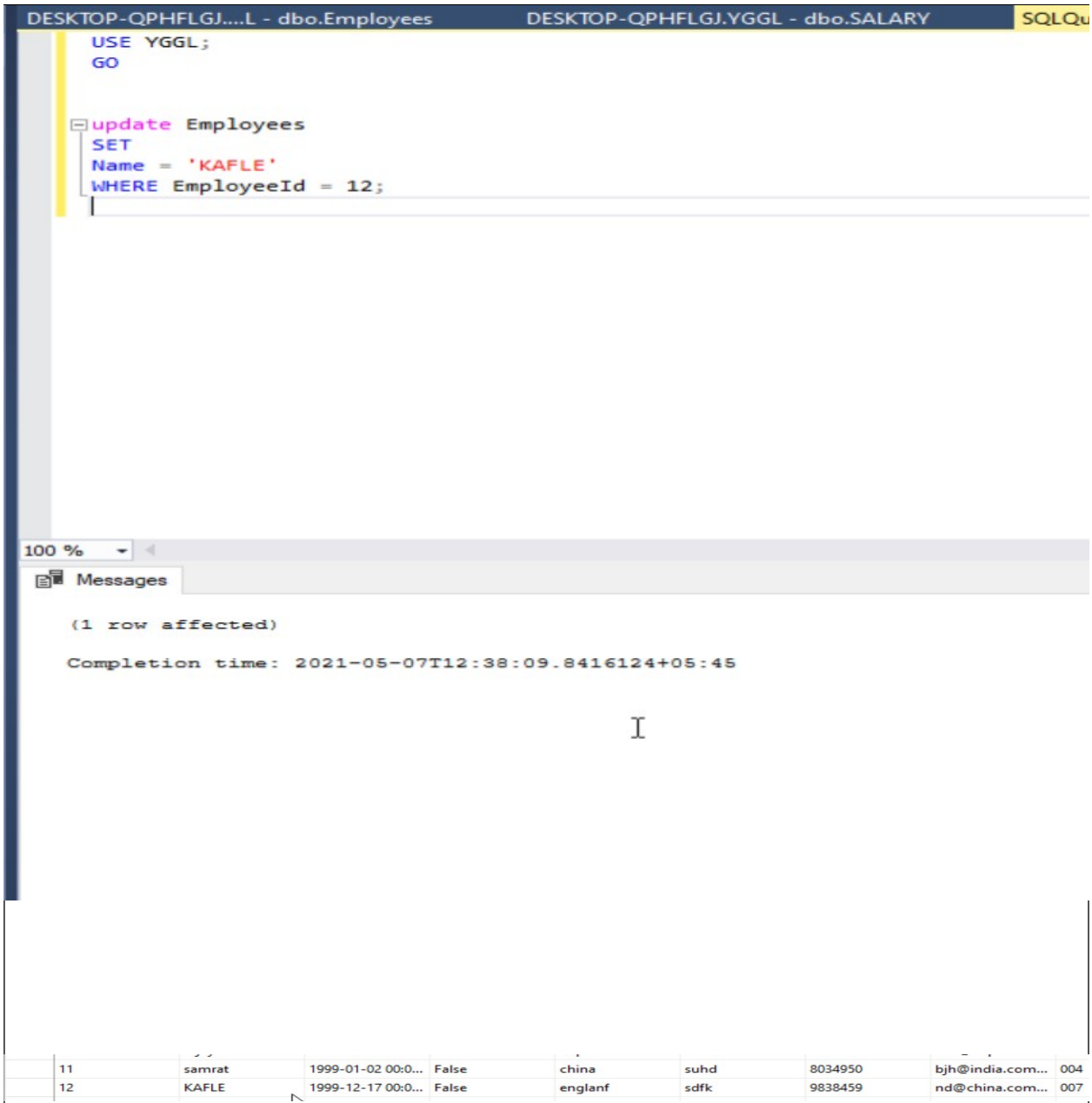


.



.

update data from table Employe:



11	samrat	1999-01-02 00:0...	False	china	suhd	8034950	bjh@india.com...	004
12	KAFLE	1999-12-17 00:0...	False	englanf	sdfk	9838459	nd@china.com...	007

实验总结及体会：

1. When inserting data, it is important to pay attention to the data type and the size of the allocation when the table is originally built.

2. Proficient in the language of inserting table data, deleting delete, modifying alter and other operations, and modifying basic tables The operation of internal properties is differentiated, and the keyword to delete the basic table is drop, and the keyword to delete the data is delete.

3. You cannot delete tables that have been built and inserted into data at will.

实验三：数据库的备份与恢复

一.实验目的与要求

（1）掌握在资源对象管理器中创建命名备份设备的方法。

（2）掌握在对象资源管理器中进行备份操作的步骤。

（3）掌握使用 T-SQL 语句对数据库进行完全备份的方法。

（4）掌握在资源对象管理器中进行数据库恢复的步骤。

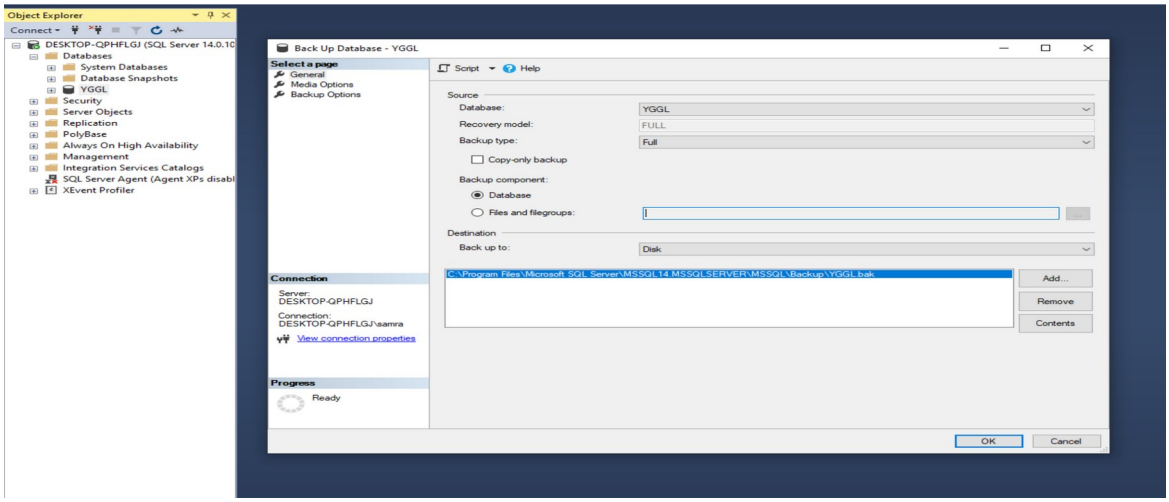
（5）掌握使用 T-SQL 语句进行数据库恢复的方法。

二.实验内容

使用逻辑名 CPYGBAK，并将数据库 YGGL 完全备份到该设备；将数据库 YGGL 完全备份到备份设备 test,并覆盖该设备上原有的内容；创建一个命名的备份设备 YGGLLOGBK，并备份 PXSCJ 数据库的事务日志；恢复整个数据库 YGGL；使用事务日志恢复数据库 YGGL。

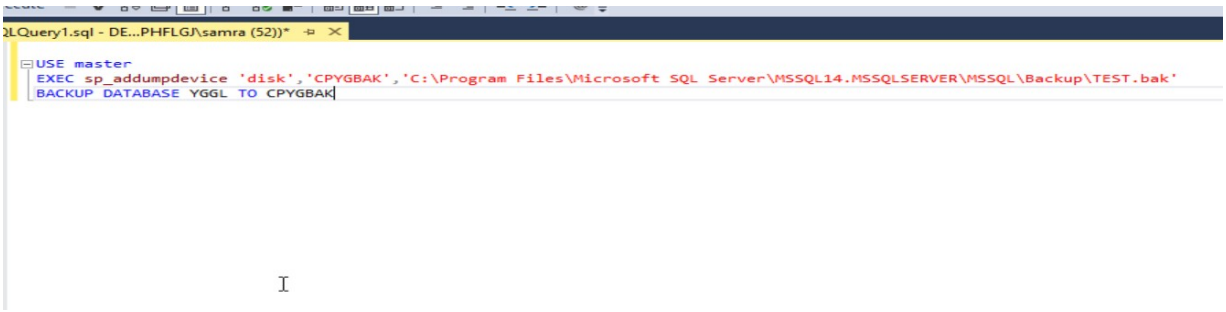
三.实验内容和结果

(1) Make a full database backup in Object Explorer.

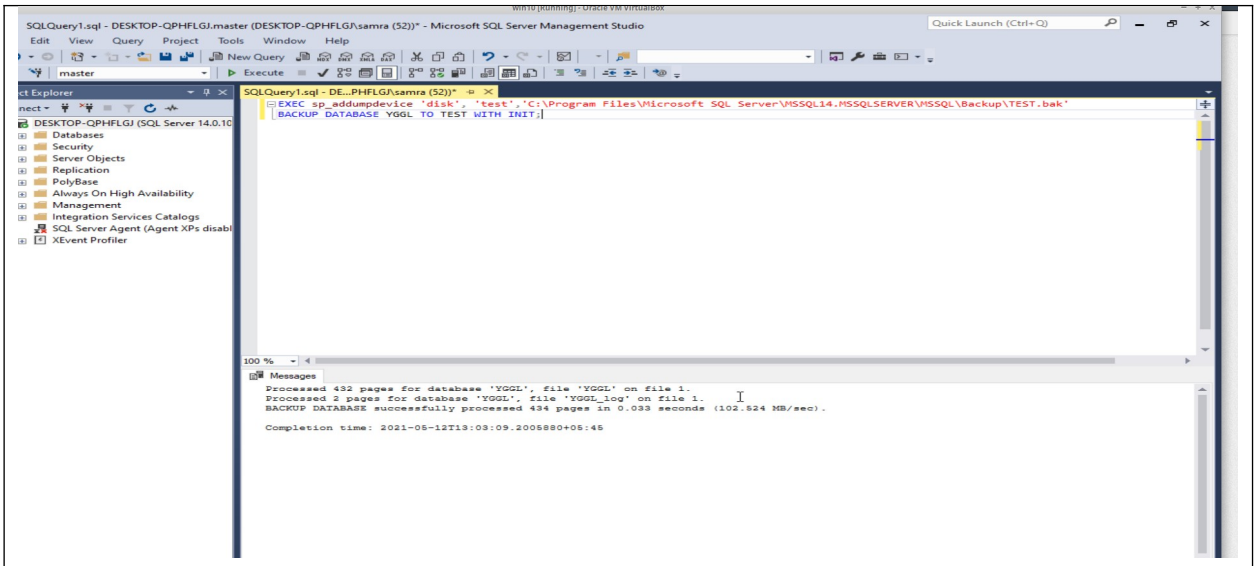


Make a full backup of the database using T-SQL statements:

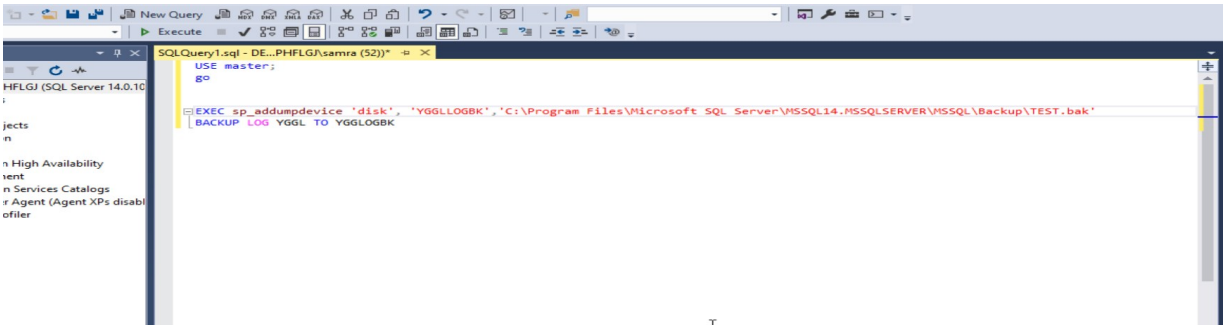
1) Create a named backup device with the logical name CPYGBAK and fully back up the database YGGL to the device:



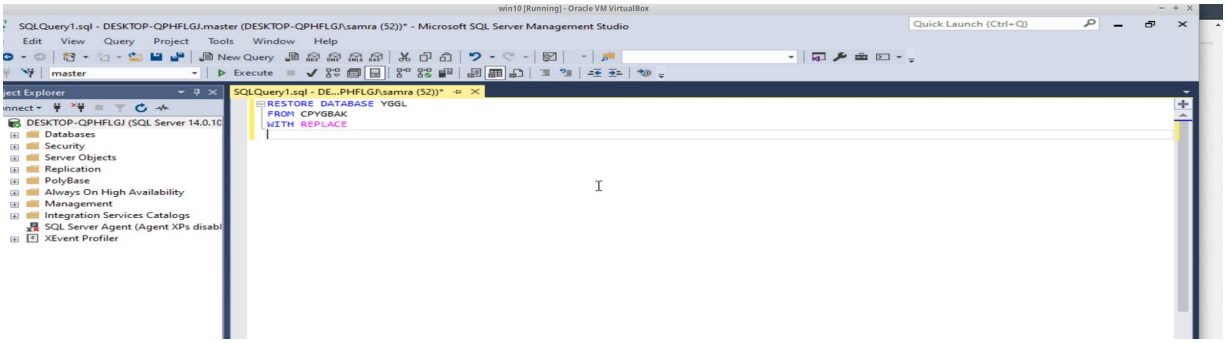
2) Full backup of database YGGL to the device and overwrite the original contents on it.



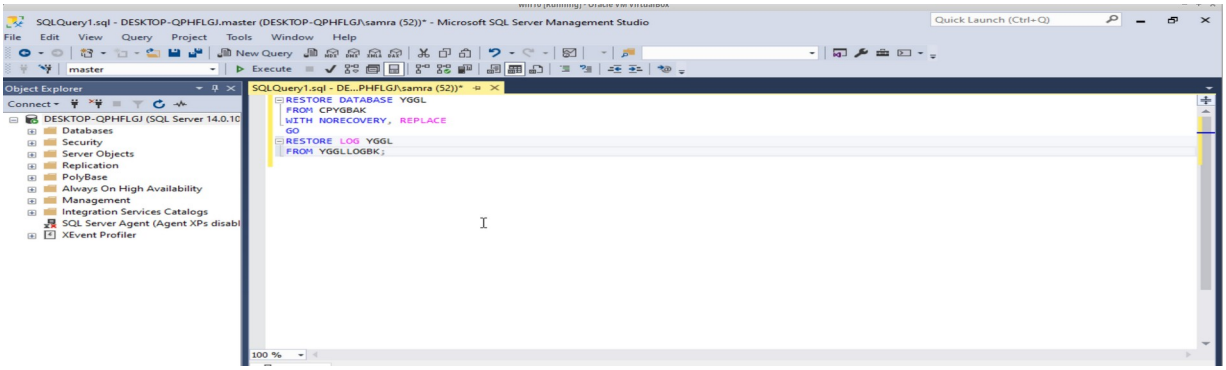
3) Create a named backup device YGGLLOGBAK, and backup transaction log of the PXSCJ database.



4) using T-SQL statements to to recover complete databse YGGL



5) Use transaction log to restore database YGGL



实验总结及体会：

Through this experiment, I learned the basic database backup and

restore, and understood the difference between full backup, differential backup and log backup. Backup and restore is a frequently used function which is a essential part of database system.

实验四：**基本数据库查询**

一.实验目的与要求

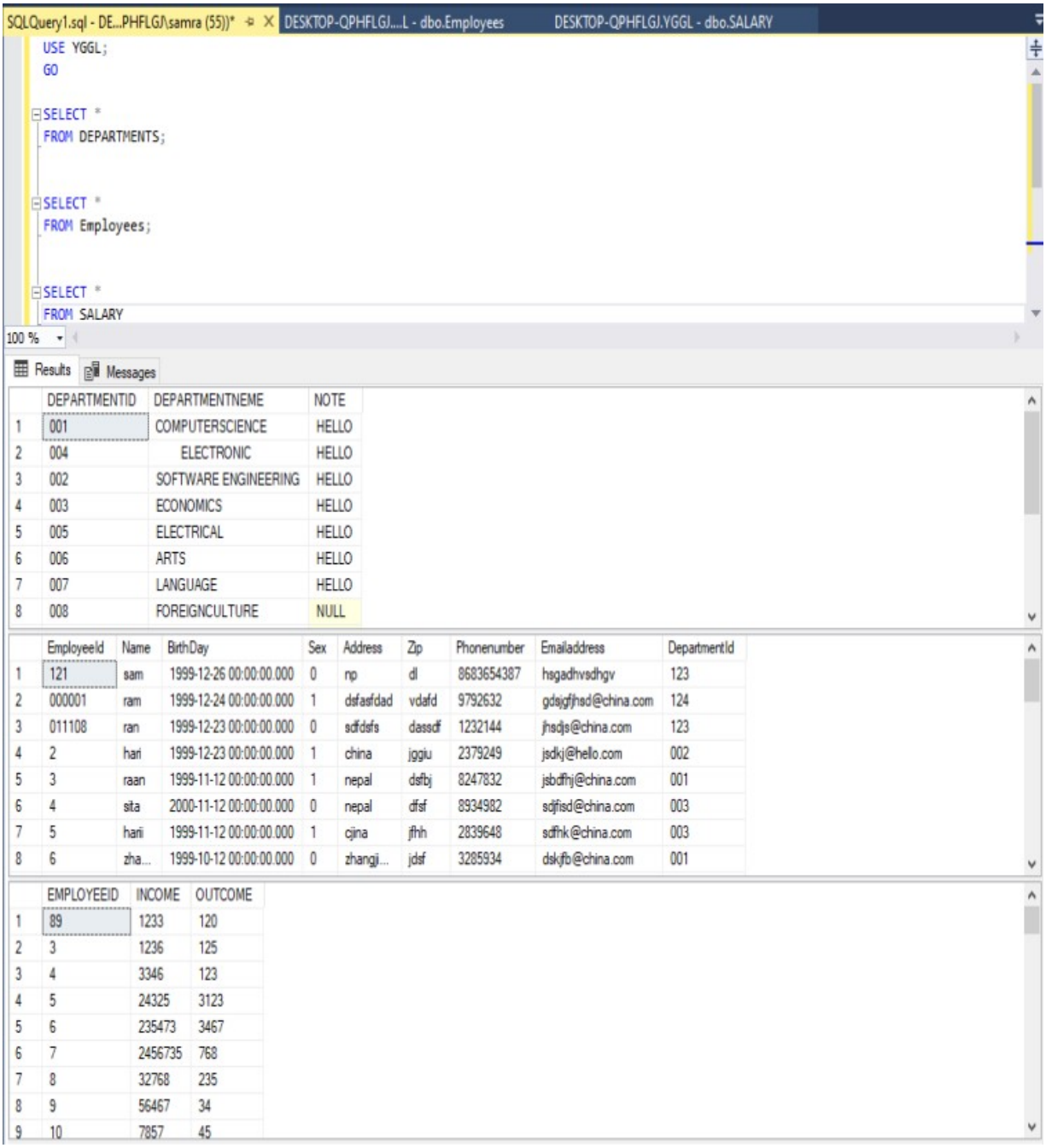
掌握 SELECT 语句的基本语法。

二.实验内容

使用 T-SQL 语句，对向实验一中建立的数据库 YGGL 的三个表 Employees、Department 和 Salary 进行基本的查询，完成实验 4.

三.实验内容和结果

For the database table structure given in Experiment 1, query all the data of each Employees.



	DEPARTMENTID	DEPARTMENTNEME	NOTE
1	001	COMPUTERSCIENCE	HELLO
2	004	ELECTRONIC	HELLO
3	002	SOFTWARE ENGINEERING	HELLO
4	003	ECONOMICS	HELLO
5	005	ELECTRICAL	HELLO
6	006	ARTS	HELLO
7	007	LANGUAGE	HELLO
8	008	FOREIGNCULTURE	NULL

	EmployeeId	Name	BirthDay	Sex	Address	Zip	Phonenumber	Emailaddress	DepartmentId
1	121	sam	1999-12-26 00:00:00.000	0	np	dl	8683654387	hsgadhvsdhgv	123
2	000001	ram	1999-12-24 00:00:00.000	1	dsfasfdad	vdafd	9792632	gdsjgfjhsd@china.com	124
3	011108	ran	1999-12-23 00:00:00.000	0	sdfdsfs	dassdf	1232144	jhsdjs@china.com	123
4	2	hari	1999-12-23 00:00:00.000	1	china	jggiu	2379249	jsdkj@hello.com	002
5	3	raan	1999-11-12 00:00:00.000	1	nepal	dsfbj	8247832	jsbdfhj@china.com	001
6	4	sita	2000-11-12 00:00:00.000	0	nepal	dfsf	8934982	sdjfisd@china.com	003
7	5	harii	1999-11-12 00:00:00.000	1	cjina	jfhh	2839648	sdfhk@china.com	003
8	6	zha...	1999-10-12 00:00:00.000	0	zhangji...	jdsf	3285934	dskjfb@china.com	001

	EMPLOYEEID	INCOME	OUTCOME
1	89	1233	120
2	3	1236	125
3	4	3346	123
4	5	24325	3123
5	6	235473	3467
6	7	2456735	768
7	8	32768	235
8	9	56467	34
9	10	7857	45

.

Query each Employees for Employeeid, Name,Sex, and Departmentid.

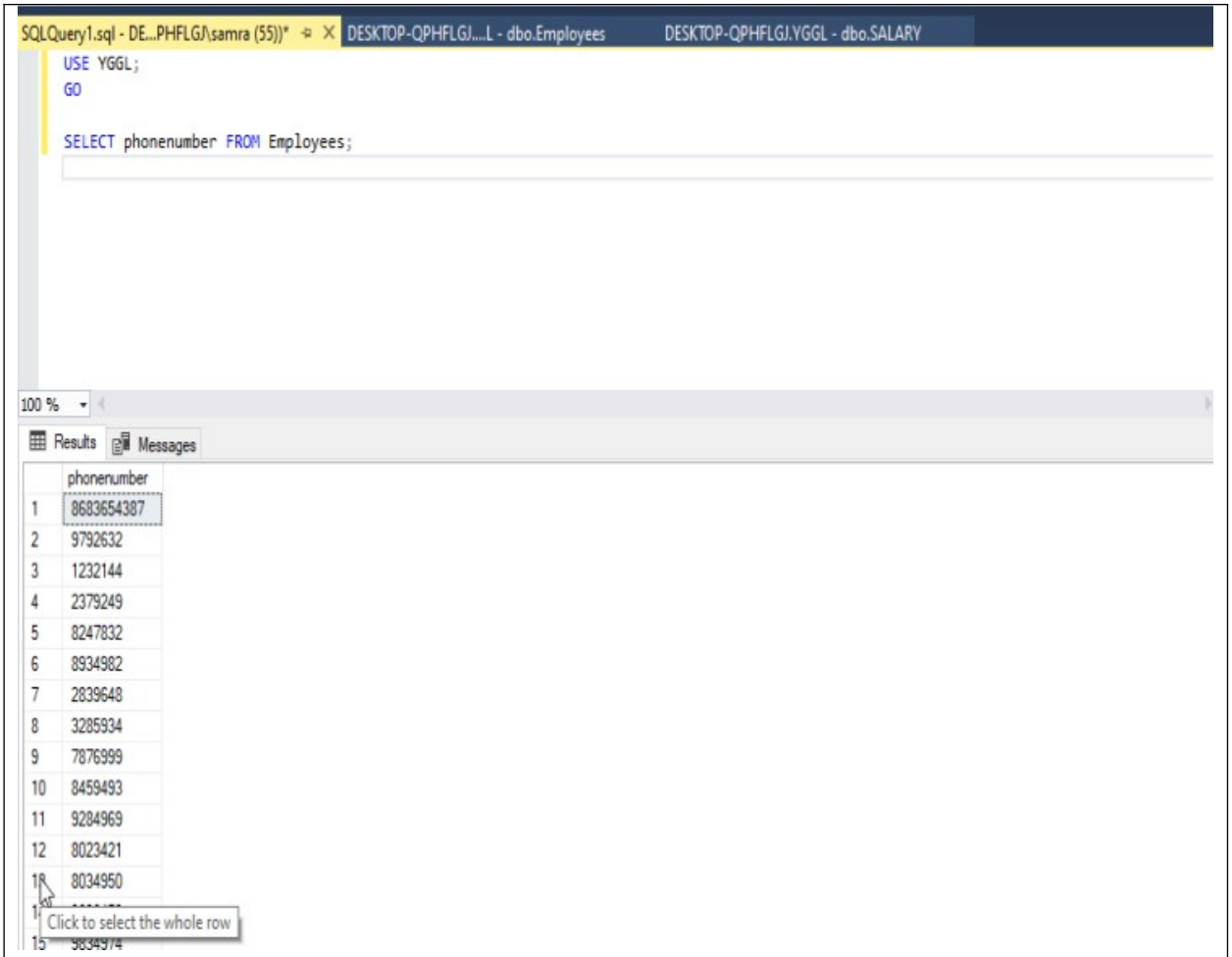
SQLQuery1.sql - DE...PHFLGJ\samra (55))* | DESKTOP-QPHFLGJ....L - dbo.Employees | DESKTOP-QPHFLGJ.YGGL - dbo.SALARY

```
USE YGGL;
GO

SELECT Employeeid,Name,Sex,Departmentid
FROM Employees;
```

100 %

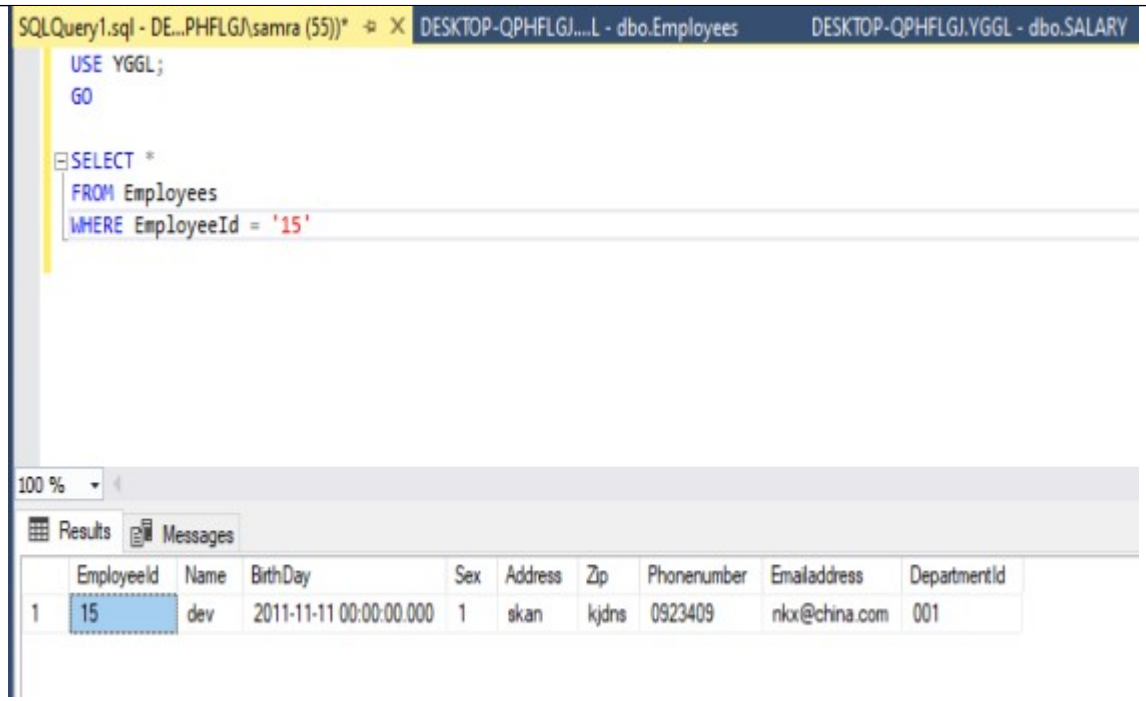
Results | Messages

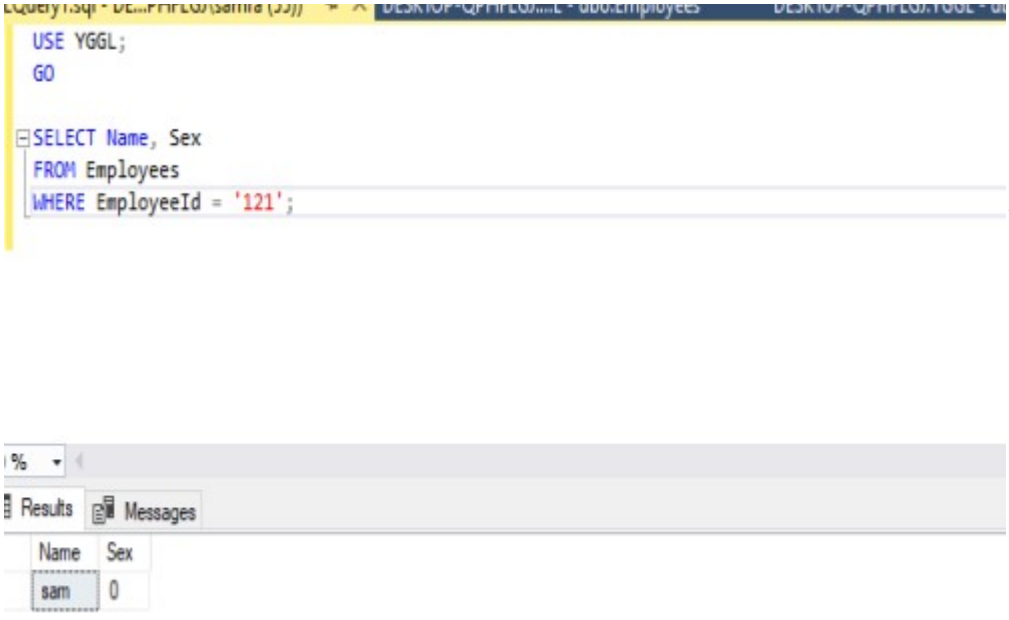
	Employeeid	Name	Sex	Departmentid
1	121	sam	0	123
2	000001	ram	1	124
3	011108	ran	0	123
4	2	hari	1	002
5	3	raan	1	001
6	4	sita	0	003
7	5	harii	1	003
8	6	zhangfeng	0	001
9	7	harry	0	006
10	8	anita	0	008
11	9	arun	1	007
12	10	bijaya	0	002
13	11	samrat	0	004
14	12	KAFLE	0	007
15	13	yixinxin	1	008
16	14	dsjb	0	008
17	15	dev	1	001
18	16	devraj	1	002
19	17	bini	0	003
20	18	binita	1	004
21	19	dam	1	005
22	20	damodar	0	006
23	21	dikshya	0	007
24	22	apil	1	008
25	23	sushma	1	009
26	24	bidu	1	001
27	25	bidur	1	005

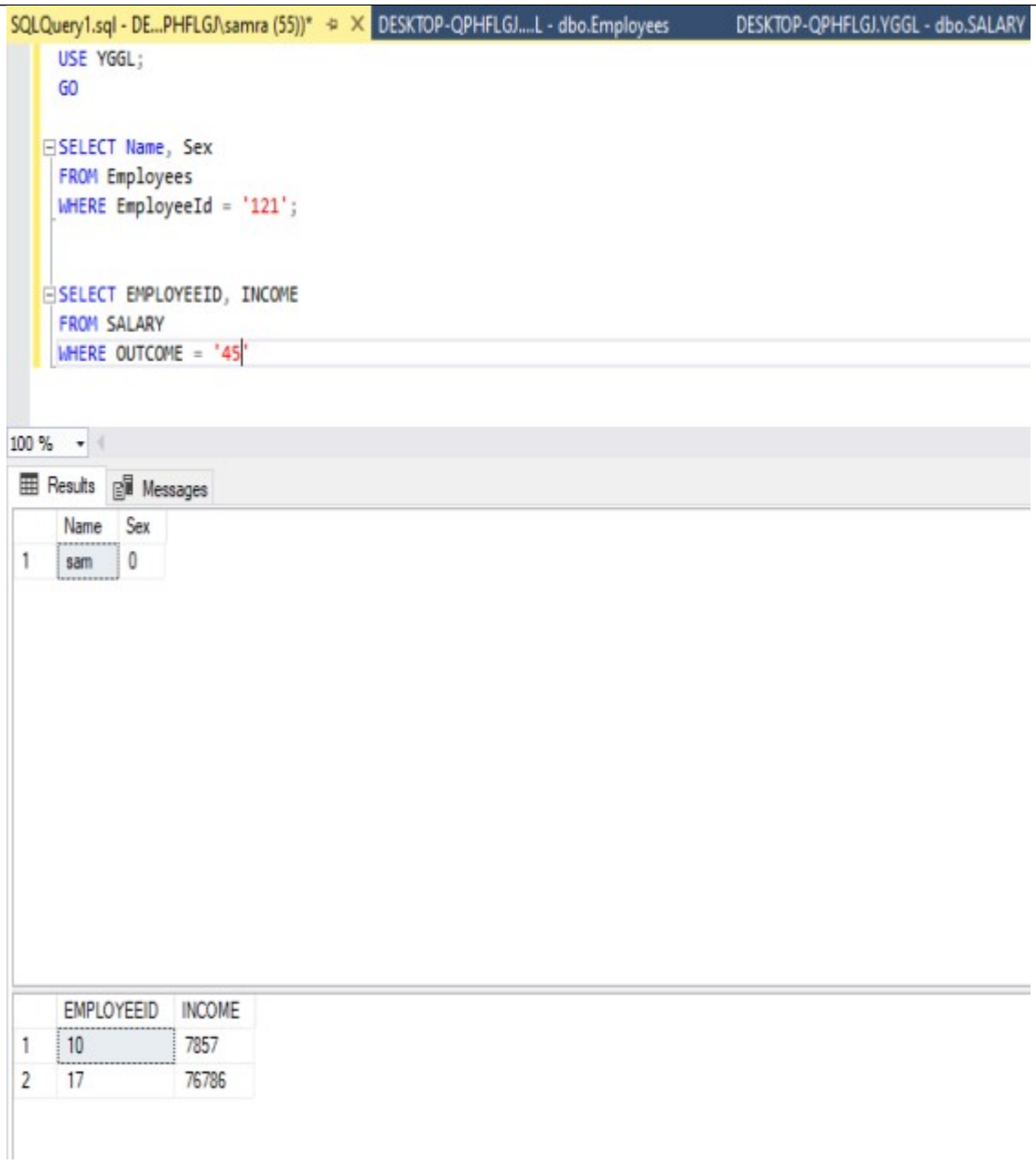


.

Query from table(DEPARTMENT, Employee, SALARY) for columns that meet certain criteria:


SQLQuery1.sql - DE...PHFLGJ\samra (55))*
DESKTOP-QPHFLGJ....L - dbo.Employees
DESKTOP-QPHFLGJ.YGGL - dbo.SALARY
USE YGGL;
GO
SELECT *
FROM Employees
WHERE EmployeeId = '15'
100 %
Results
Messages
EmployeeId Name BirthDay Sex Address Zip Phonenumber Emailaddress DepartmentId
1 15 dev 2011-11-11 00:00:00.000 1 skan kjdns 0923409 nkx@china.com 001


USE YGGL;
GO
SELECT Name, Sex
FROM Employees
WHERE EmployeeId = '121';
%
Results
Messages
Name Sex
sam 0


SQLQuery1.sql - DE...PHFLGJ\samra (55))*
DESKTOP-QPHFLGJ....L - dbo.Employees
DESKTOP-QPHFLGJ.YGGL - dbo.SALARY
USE YGGL;
GO
SELECT Name, Sex
FROM Employees
WHERE EmployeeId = '121';
SELECT EMPLOYEEID, INCOME
FROM SALARY
WHERE OUTCOME = '45'
100 %
Results
Messages
Name
Sex
1
sam
0
EMPLOYEEID
INCOME
1
10
7857
2
17
76786

Using ‘as’ statement to display table:

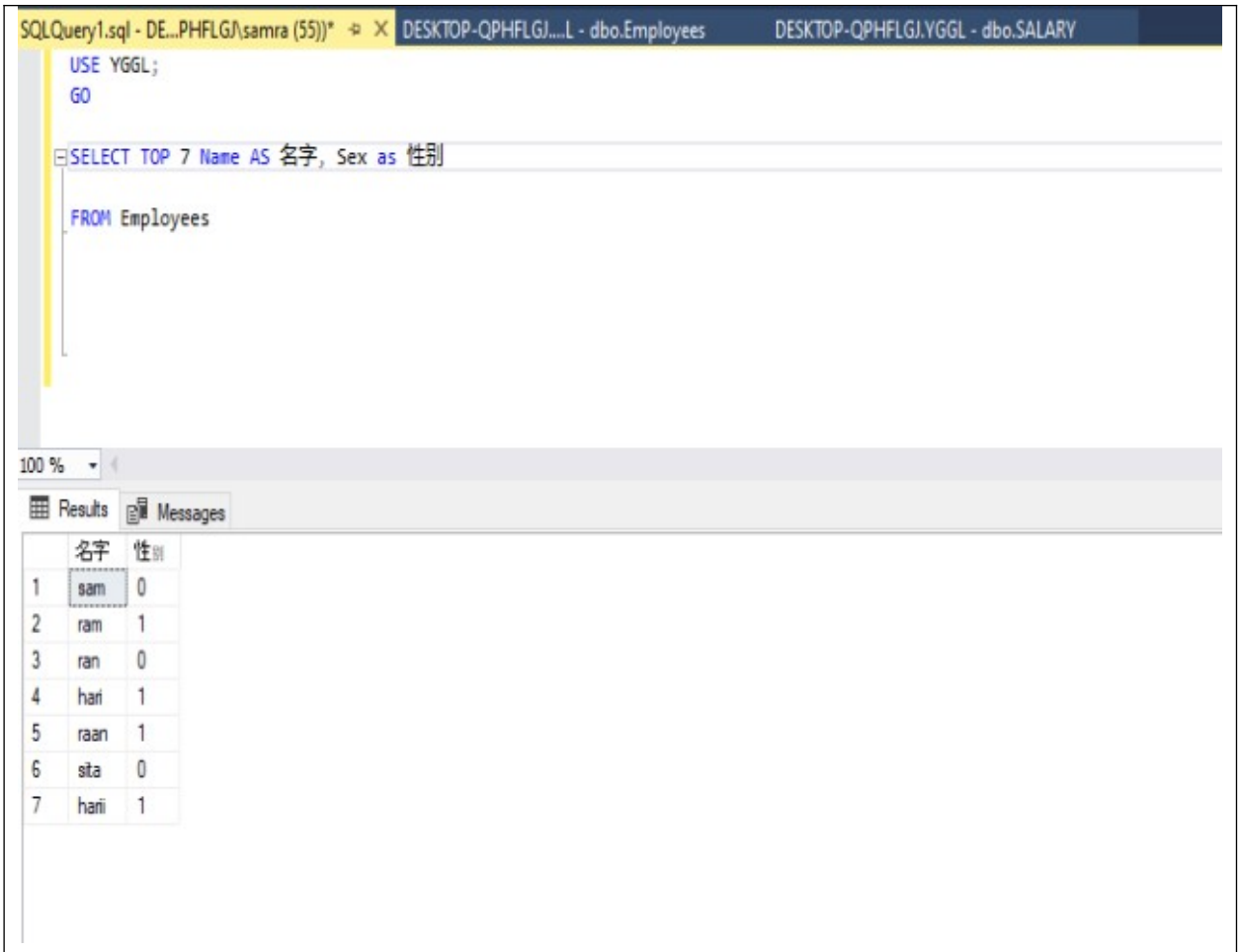
```
USE YGGL;
GO

SELECT Name as 名字, Sex as 性别
FROM Employees


SELECT EMPLOYEEID as EID, INCOME as 输入
FROM SALARY
WHERE OUTCOME = '45'
```

Results | Messages

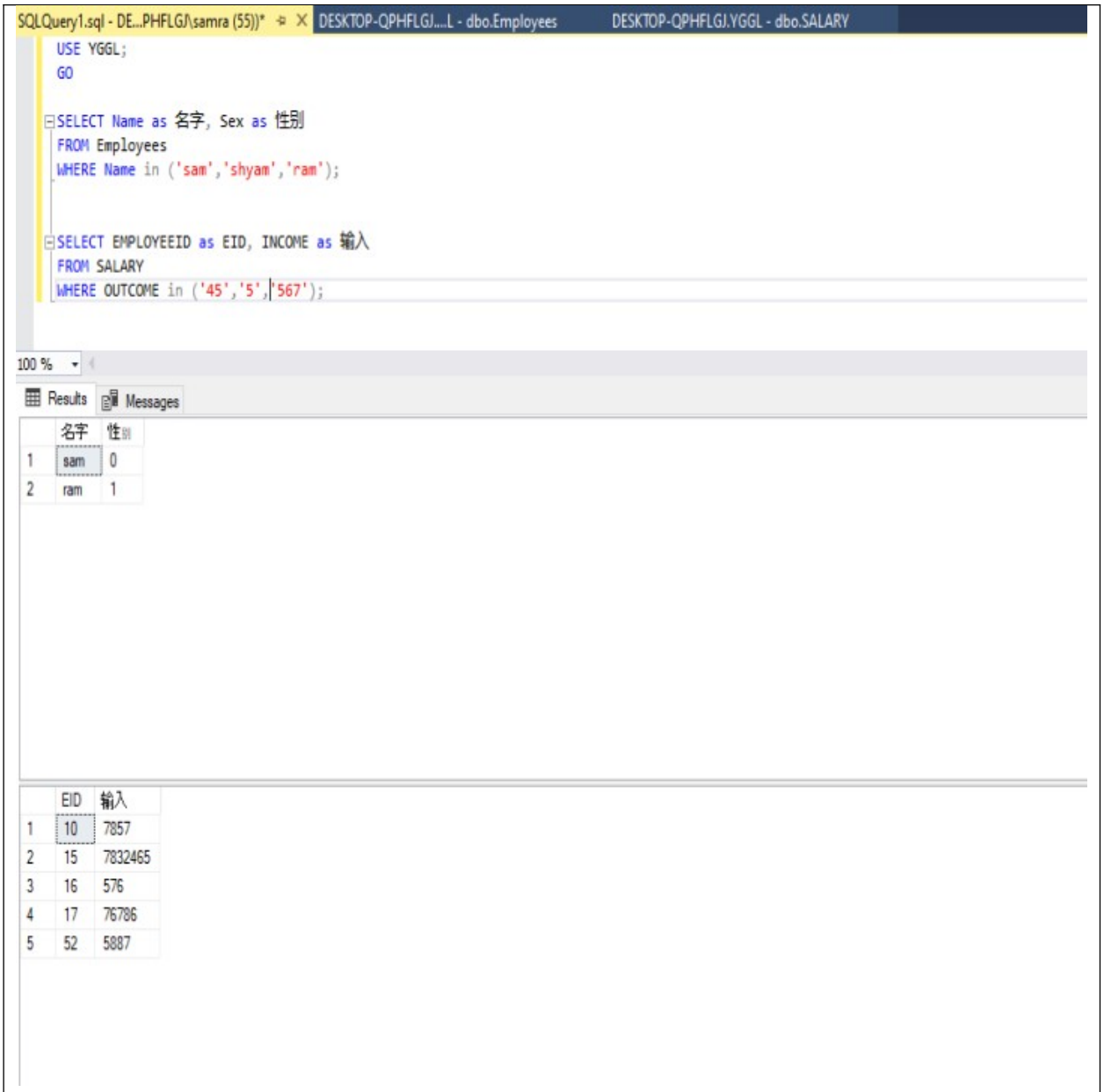
	名字	性别
1	sam	0
2	ram	1
3	ran	0
4	hari	1
5	raan	1
6	sita	0
7	harii	1
8	zhangfeng	0
9	harry	0
10	anita	0
11	arun	1
12	bijaya	0
13	samrat	0
14	KAFLE	0
15	yixinxin	1
16	dsjb	0

	EID	输入
1	10	7857
2	17	76786

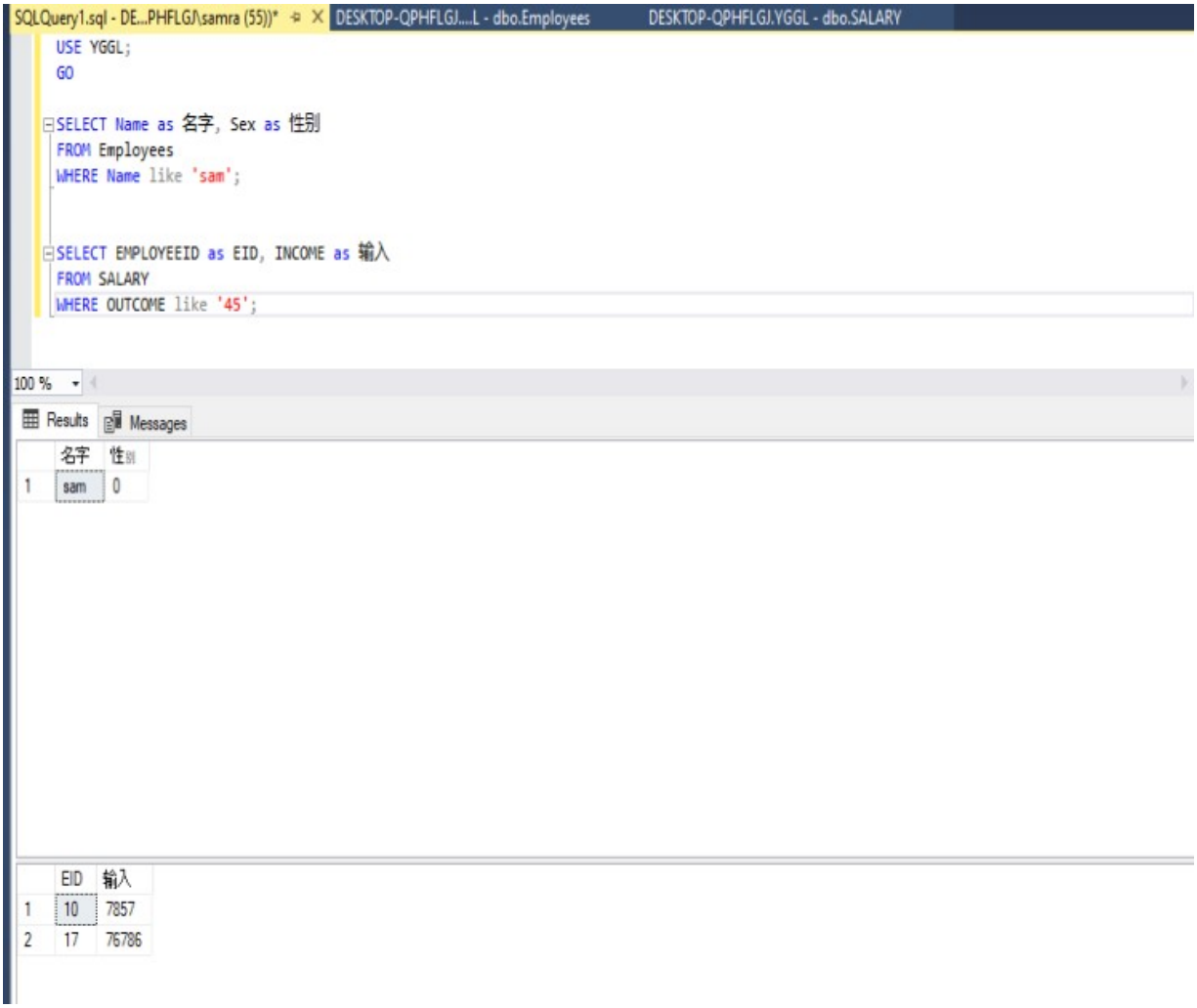
.



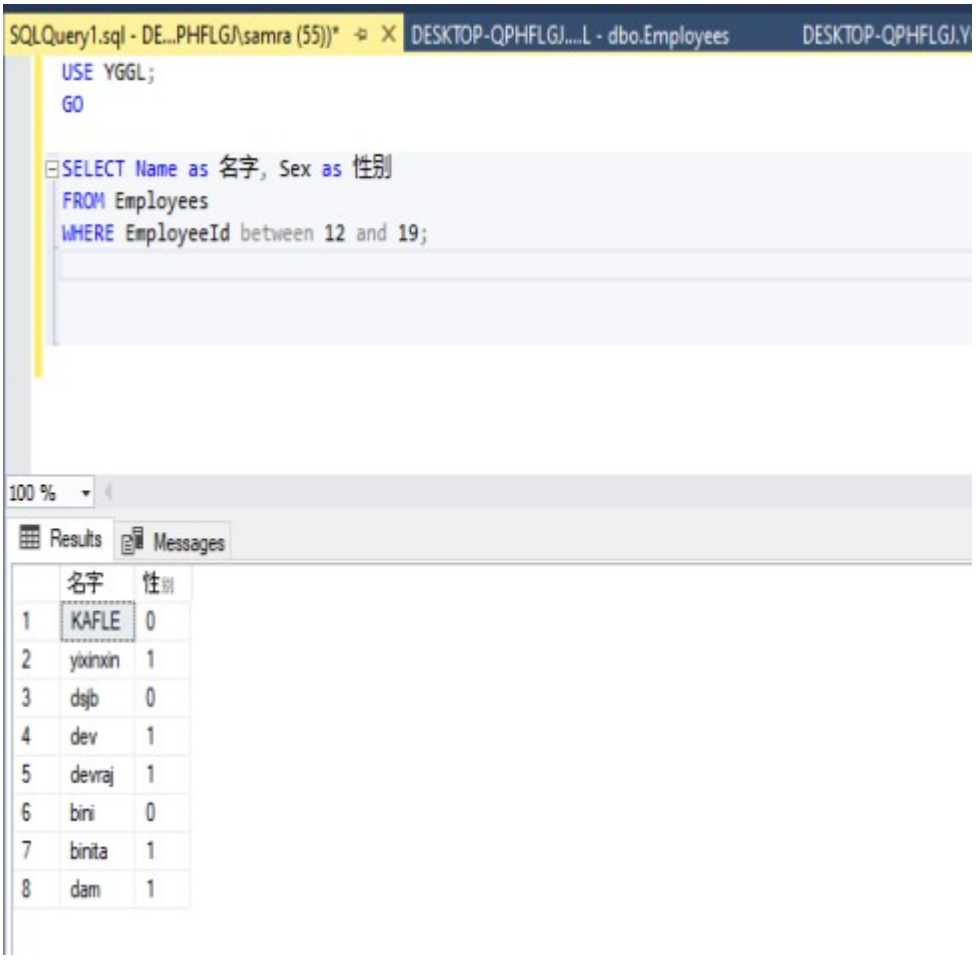
Using ‘in’ statement to display table:



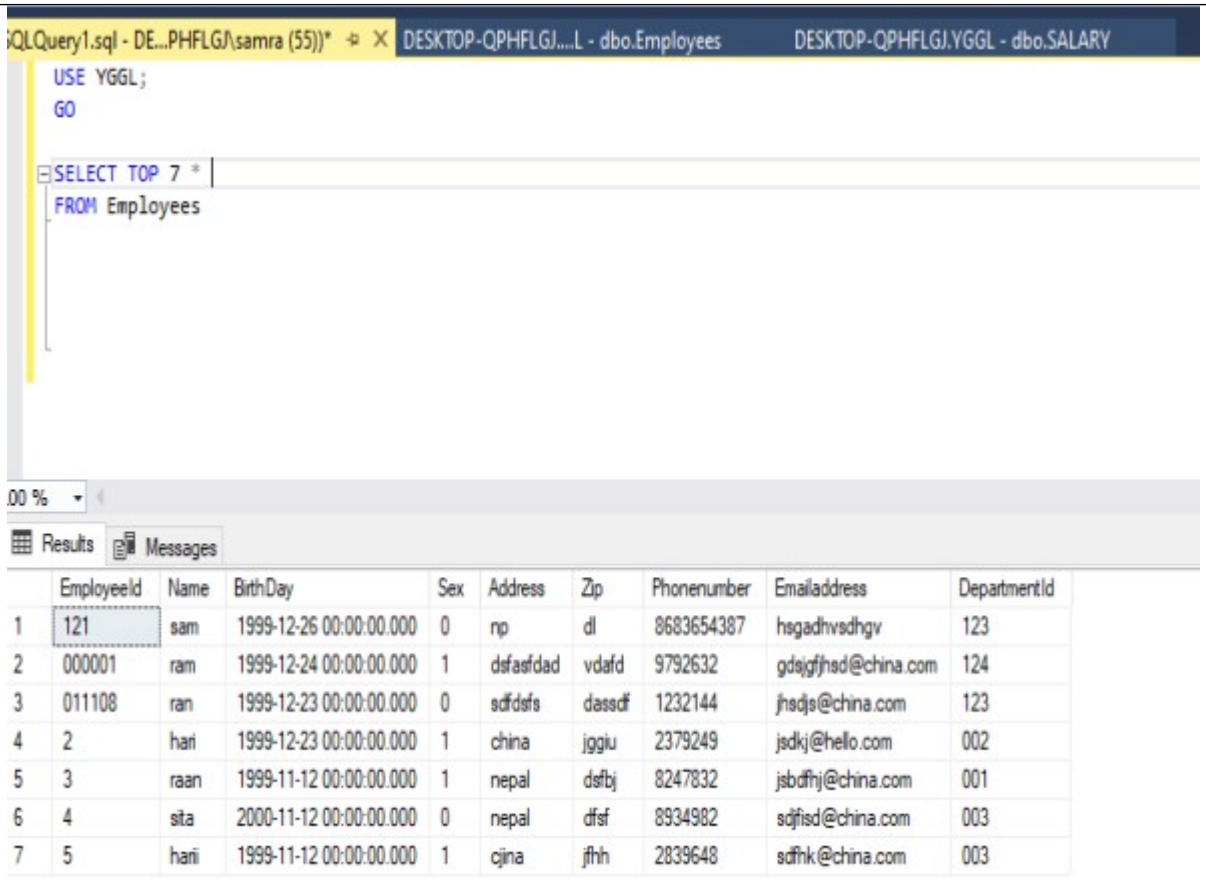
Using ‘like’ statement to display table:



Using ‘between’ statement to display table:


SQLQuery1.sql - DE....PHFLGJ\samra (55))*
DESKTOP-QPHFLGJ....L - dbo.Employees
DESKTOP-QPHFLGJ.Y
USE YGGL;
GO
SELECT Name as 名字, Sex as 性别
FROM Employees
WHERE EmployeeId between 12 and 19;
100 %
Results
Messages
名字
性别
1 KAFLE 0
2 yxinxin 1
3 dsjb 0
4 dev 1
5 devraj 1
6 bini 0
7 binita 1
8 dam 1

Using ‘TOP’ statement to LIMIT the table (rows):



	EmployeeId	Name	BirthDay	Sex	Address	Zip	Phonenumber	Emailaddress	DepartmentId
1	121	sam	1999-12-26 00:00:00.000	0	np	dl	8683654387	hsgadhvsdhgv	123
2	000001	ram	1999-12-24 00:00:00.000	1	dsfasfdad	vdafd	9792632	gdsjgfjhsd@china.com	124
3	011108	ran	1999-12-23 00:00:00.000	0	sdfdsfs	dassdf	1232144	jhsdjs@china.com	123
4	2	hari	1999-12-23 00:00:00.000	1	china	jggiu	2379249	jsdkj@hello.com	002
5	3	raan	1999-11-12 00:00:00.000	1	nepal	dsfbj	8247832	jsbdfhj@china.com	001
6	4	sita	2000-11-12 00:00:00.000	0	nepal	dfsf	8934982	sdjfisd@china.com	003
7	5	hari	1999-11-12 00:00:00.000	1	cjina	jfhh	2839648	sdfhk@china.com	003

实验总结及体会： 通过这个实验，我学习了 AS、LIKE 和 between 子查询的使用，并注意它们各自的作用。LIKE 用于字符串匹配。BETWEEN 用于确定范围，AS 用于修改查询结果中列的别名。

实验五：嵌套查询、连接查询操作

一.实验目的与要求 （1）掌握嵌套查询的表示； （2）掌握连接查询的表示。

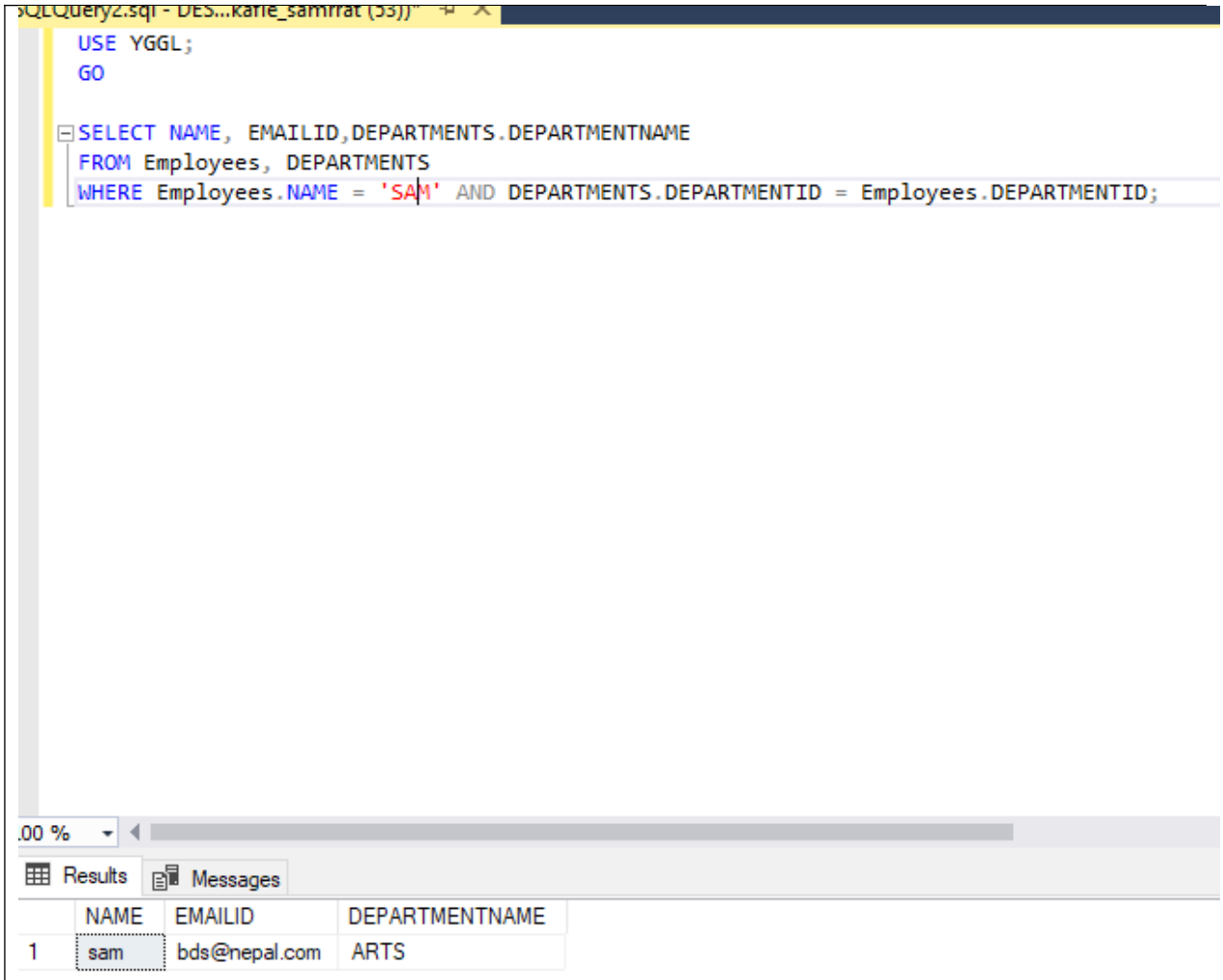
二.实验内容

使用 T-SQL 语句，对向实验一中建立的数据库 YGGL 的三个表 Employees、Department 和 Salary 进行嵌套查询、连接查询操作，完成实验 4.1 中的“2.子查询”、“3.连接查询”两部分的【思考与练习】中的相应查询内容，并给出查询结果。

注：若查询结果集行数超过 7 行时，使用 TOP 选项限制返回行数为 7。

三.实验内容和结果

Find the Employee of name ‘sam’ through DEPARTMENTID from table Employees and table DEPARTMENTS

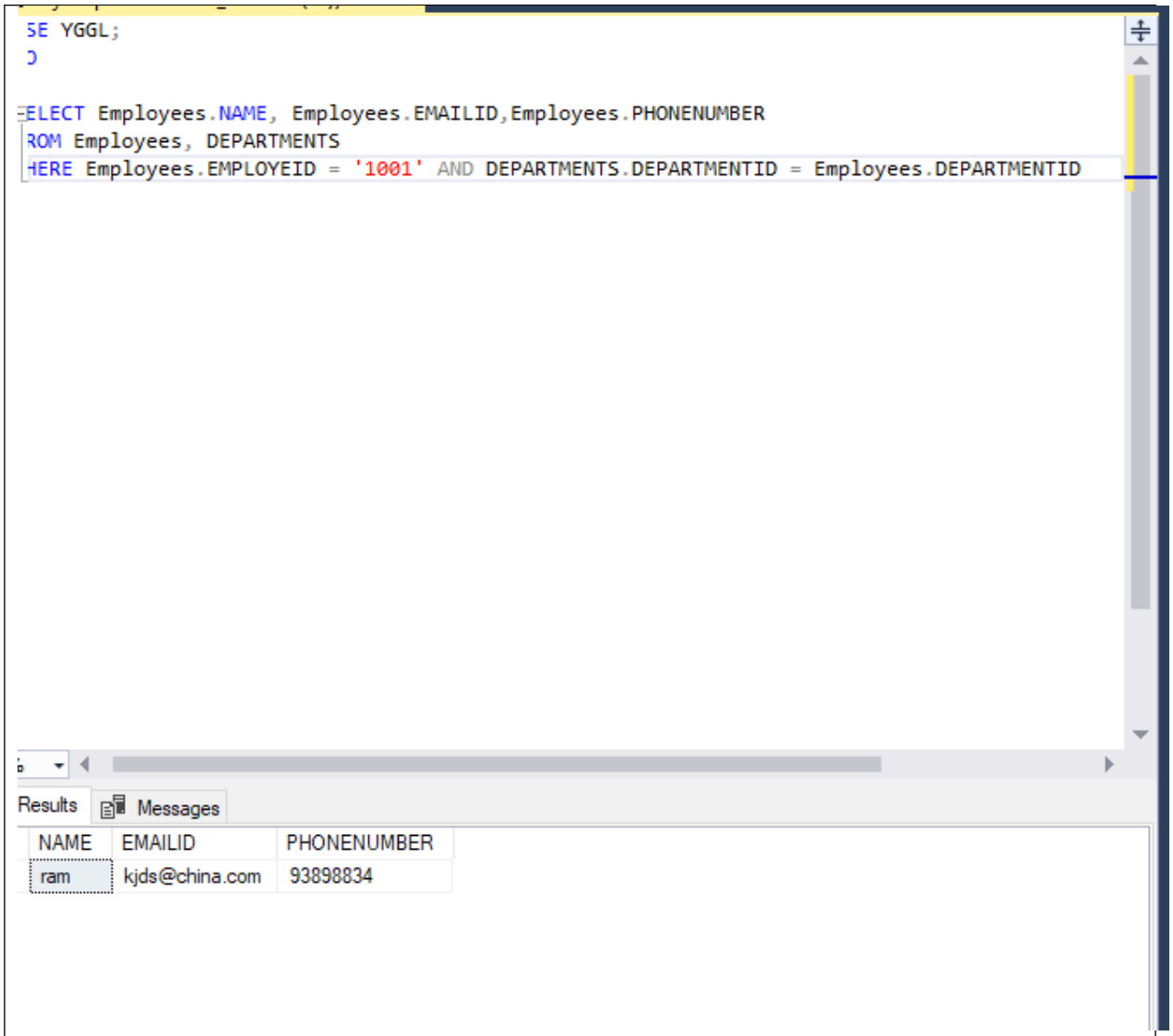


Take out the Emplyee name , email id and phonenumber from table Employee whose Employeid is given:

```
USE YGGL;
GO

SELECT Employees.NAME, Employees.EMAILID,Employees.PHONENUMBER
FROM Employees, DEPARTMENTS
WHERE Employees.EMPLOYEID = '1006' AND DEPARTMENTS.DEPARTMENTID = Employees.DEPARTMENTID;
```

Results | Messages

NAME	EMAILID	PHONENUMBER
amrit	fg@xv.bn	35646464



Name, Emailid, phonenumber and Salary(income) taken by Employees according to Employeeid:

```
USE YGGL;
GO

SELECT Employees.NAME, Employees.EMAILID,Employees.PHONENUMBER,SALARY.INCOME
FROM Employees, SALARY
WHERE SALARY.EMPLOYEID = Employees.EMPLOYEID
```

Results | Messages

	NAME	EMAILID	PHONENUMBER	INCOME
	ram	kjds@china.com	93898834	9385
	sam	bds@nepal.com	89273894	7868
	shyam	dsd@china.com	10290383	78979
	bini	cnsn@china.com	90390322	7899
	mihir	nsbc@china.com	09438590	8799
	amrit	fg@xv.bn	35646464	789999
	kapil	jdks@jinan.com	98347985	78678
	karu	cbc@china.com	90734879	7567
	zhangpwnf	cnsd@china.com	93475987	6577
0	damo	scbb@china.com	98734689	6868
1	dikshya	jsd@cn.com	92374981	6868
2	babu	dsss@cn.com	02734979	8768

List of Employees Name, Emailid, phonenumber and income whose

gender is ‘male’:

QLQuery2.sql - DES...kafle_samrrat (53))*

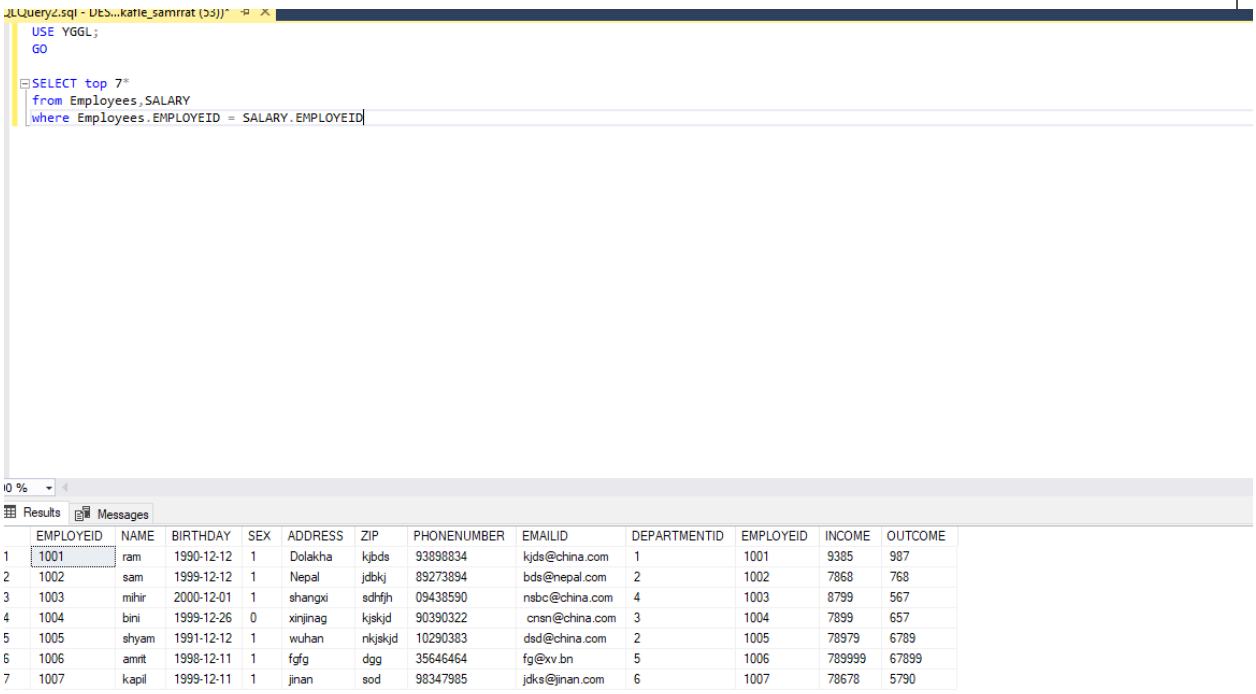
```
USE YGGL;
GO

SELECT Top 7  Employees.EMAILID,Employees.PHONENUMBER,SALARY.INCOME
FROM Employees, SALARY
WHERE Employees.SEX = '1' and SALARY.EMPLOYEID = Employees.EMPLOYEID
```

00 %

Results | Messages

	EMAILID	PHONENUMBER	INCOME
1	kjds@china.com	93898834	9385
2	bds@nepal.com	89273894	7868
3	dsd@china.com	10290383	78979
4	nsbc@china.com	09438590	8799
5	fg@xv.bn	35646464	789999
6	jdks@jinan.com	98347985	78678
7	cnsd@china.com	93475987	6577

Displays all Employes info with income and outcome:



```
USE YGGL;
GO

SELECT top 7*
from Employees,SALARY
where Employees.EMPLOYEID = SALARY.EMPLOYEID
```

Results | Messages

	EMPLOYEID	NAME	BIRTHDAY	SEX	ADDRESS	ZIP	PHONENUMBER	EMAILID	DEPARTMENTID	EMPLOYEID	INCOME	OUTCOME
1	1001	ram	1990-12-12	1	Dolakha	kjbds	93898834	kjds@china.com	1	1001	9385	987
2	1002	sam	1999-12-12	1	Nepal	jdbkj	89273894	bds@nepal.com	2	1002	7868	768
3	1003	mihir	2000-12-01	1	shangxi	sdhfjh	09438590	nsbc@china.com	4	1003	8799	567
4	1004	bini	1999-12-26	0	xinjinag	kjskjd	90390322	cnsn@china.com	3	1004	7899	657
5	1005	shyam	1991-12-12	1	wuhan	nkjskjd	10290383	dsd@china.com	2	1005	78979	6789
6	1006	amrit	1998-12-11	1	fgfg	dgg	35646464	fg@xv.bn	5	1006	789999	67899
7	1007	kapil	1999-12-11	1	jinan	sod	98347985	jdks@jinan.com	6	1007	78678	5790

Display all info from three table :

```
USE YGGL;
GO

SELECT top 7*
from Employees,SALARY,DEPARTMENTS
where Employees.EMPLOYEID = SALARY.EMPLOYEID and Employees.DEPARTMENTID = DEPARTMENTS.DEPARTMENTID;
```

00 %

Results | Messages

	EMPLOYEID	NAME	BIRTHDAY	SEX	ADDRESS	ZIP	PHONENUMBER	EMAILID	DEPARTMENTID	EMPLOYEID	INCOME	OUTCOME	DEPARTMENTID	DEPARTMENTNAME	NOTE
1	1001	ram	1990-12-12	1	Dolakha	kjbds	93898834	kjds@china.com	1	1001	9385	987	1	COMPUTERSCIENCE	HELLO
2	1003	mihir	2000-12-01	1	shangxi	sdhfjh	09438590	nsbc@china.com	4	1003	8799	567	4	ELECTRONIC	hELLO
3	1004	bini	1999-12-26	0	xinjinag	kjskjd	90390322	cnsn@china.com	3	1004	7899	657	3	SOFTWARE ENG	HELLO
4	1008	karu	1999-12-18	0	ksnd	kjnsd	90734879	cbc@china.com	7	1008	7567	565	7	LANGUAGE	NULL
5	1010	damo	1991-12-13	1	ktm	sakjcn	98734689	scbb@china.com	7	1010	6868	678	7	LANGUAGE	NULL
6	1009	zhangpwnf	1999-01-12	1	cksjd	jkd	93475987	cnsd@china.com	8	1009	6577	545	8	FOREIGN CULTURE	NULL
7	1011	dikshya	1992-11-11	0	illam	csa	92374981	jsd@cn.com	9	1011	6868	768	9	FOREIGN LANGUAGE	NULL

List the info of Employee whose income is highest in the row:

SQLQuery2.sql - DES...kafle_samrrat (53))*

```
USE YGGL;
GO

SELECT *

from Employees,DEPARTMENTS,SALARY
where Employees.DEPARTMENTID = '7' and SALARY.INCOME  =  (select max(SALARY.INCOME) from SALARY) and Employees.DEPARTMENTID = DEPARTMENTS.DEPARTMENTID
```

100 %

Results | Messages

	EMPLOYEID	NAME	BIRTHDAY	SEX	ADDRESS	ZIP	PHONENUMBER	EMAILID	DEPARTMENTID	DEPARTMENTID	DEPARTMENTNAME	NOTE	EMPLOYEID	INCOME	OUTCOME
1	1008	karu	1999-12-18	0	ksnd	kjnsd	90734879	cbc@china.com	7	7	LANGUAGE	NULL	1006	789999	67899
2	1010	damo	1991-12-13	1	ktm	sakjcn	98734689	scbb@china.com	7	7	LANGUAGE	NULL	1006	789999	67899

Outer join:

Left join:

SQLQuery2.sql - DES...kafle_samrrat (53))*

```
USE YGGL;
GO


select *
from Employees
left join SALARY on
Employees.EMPLOYEID = SALARY.EMPLOYEID;
```

100 %

Results | Messages

	EMPLOYEID	NAME	BIRTHDAY	SEX	ADDRESS	ZIP	PHONENUMBER	EMAILID	DEPARTMENTID	EMPLOYEID	INCOME	OUTCOME
1	1001	ram	1990-12-12	1	Dolakha	kjbds	93898834	kjds@china.com	1	1001	9385	987
2	1002	sam	1999-12-12	1	Nepal	jdbkj	89273894	bds@nepal.com	2	1002	7868	768
3	1005	shyam	1991-12-12	1	wuhan	nkjskjd	10290383	dsd@china.com	2	1005	78979	6789
4	1004	bini	1999-12-26	0	xinjinag	kjskjd	90390322	cnsn@china.com	3	1004	7899	657
5	1003	mihir	2000-12-01	1	shangxi	sdhfjh	09438590	nsbc@china.com	4	1003	8799	567
6	1006	amrit	1998-12-11	1	fgfg	dgg	35646464	fg@xv.bn	5	1006	789999	67899
7	1007	kapil	1999-12-11	1	jinan	sod	98347985	jdks@jinan.com	6	1007	78678	5790
8	1008	karu	1999-12-18	0	ksnd	kjnsd	90734879	cbc@china.com	7	1008	7567	565
9	1009	zhangpwnf	1999-01-12	1	cksjd	jkd	93475987	cnsd@china.com	8	1009	6577	545
10	1010	damo	1991-12-13	1	ktm	sakjcn	98734689	scbb@china.com	7	1010	6868	678
11	1011	dikshya	1992-11-11	0	illam	csa	92374981	jsd@cn.com	9	1011	6868	768
12	1012	babu	1997-12-19	1	sada	sadsd	02734979	dsss@cn.com	9	1012	8768	768

Right join:

SQLQuery2.sql - DES...kafle_samrrat (53))*

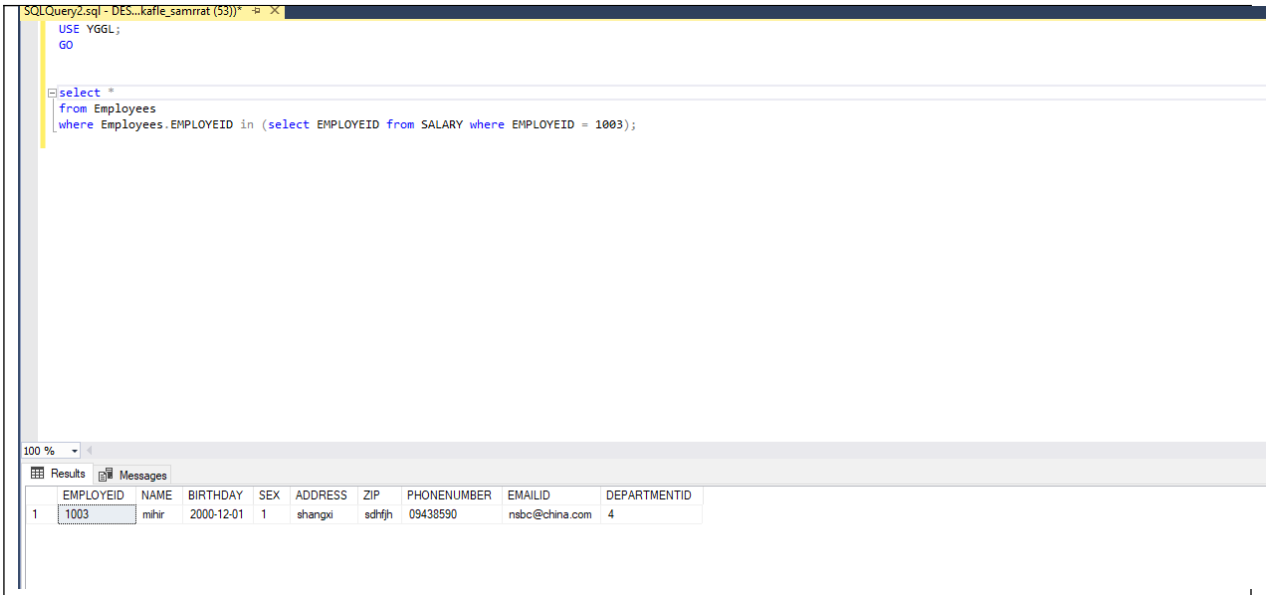
```
USE YGGL;
GO


select top 7 *
from SALARY
right join Employees on
SALARY.EMPLOYEID = Employees.EMPLOYEID;
```

100 %

Results | Messages

	EMPLOYEID	INCOME	OUTCOME	EMPLOYEID	NAME	BIRTHDAY	SEX	ADDRESS	ZIP	PHONENUMBER
1	1001	9385	987	1001	ram	1990-12-12	1	Dolakha	kjbds	93898834
2	1002	7868	768	1002	sam	1999-12-12	1	Nepal	jdbkj	89273894
3	1005	78979	6789	1005	shyam	1991-12-12	1	wuhan	nkjskjd	10290383
4	1004	7899	657	1004	bini	1999-12-26	0	xinjinag	kjskjd	90390322
5	1003	8799	567	1003	mihir	2000-12-01	1	shangxi	sdhfjh	09438590
6	1006	789999	67899	1006	amrit	1998-12-11	1	fgfg	dgg	35646464
7	1007	78678	5790	1007	kapil	1999-12-11	1	jinan	sod	98347985

Nested query:



实验总结及体会：

Through this Experiment I master join query (equivalent and non-equivalent join, self join, external join), nested query (subquery with IN, subquery with comparison operator), and can flexibly use SQL language command to join nested query on data in the table. For my future study to lay a good foundation, accumulated experience.

实验六： **分组排序查询及数据库索引**

一.实验目的与要求

（1）学会在对象资源管理器中对数据库表进行插入、修改和删除数据操作；

（2）学会使用 T-SQL 语句对数据库表进行插入、修改和删除数据操作；

（3）了解数据更新操作时要注意数据完整性。

二.实验内容

（一）分组排序查询

使用 T-SQL 语句，对向实验一中建立的数据库 YGGL 的三个表 Employees、Department 和 Salary 进行分组排序查询，完成实验 4.1 中的 “5.分组排序”的【思考与练习】中的相应查询内容（共 5 个查询），并给出查询结果。

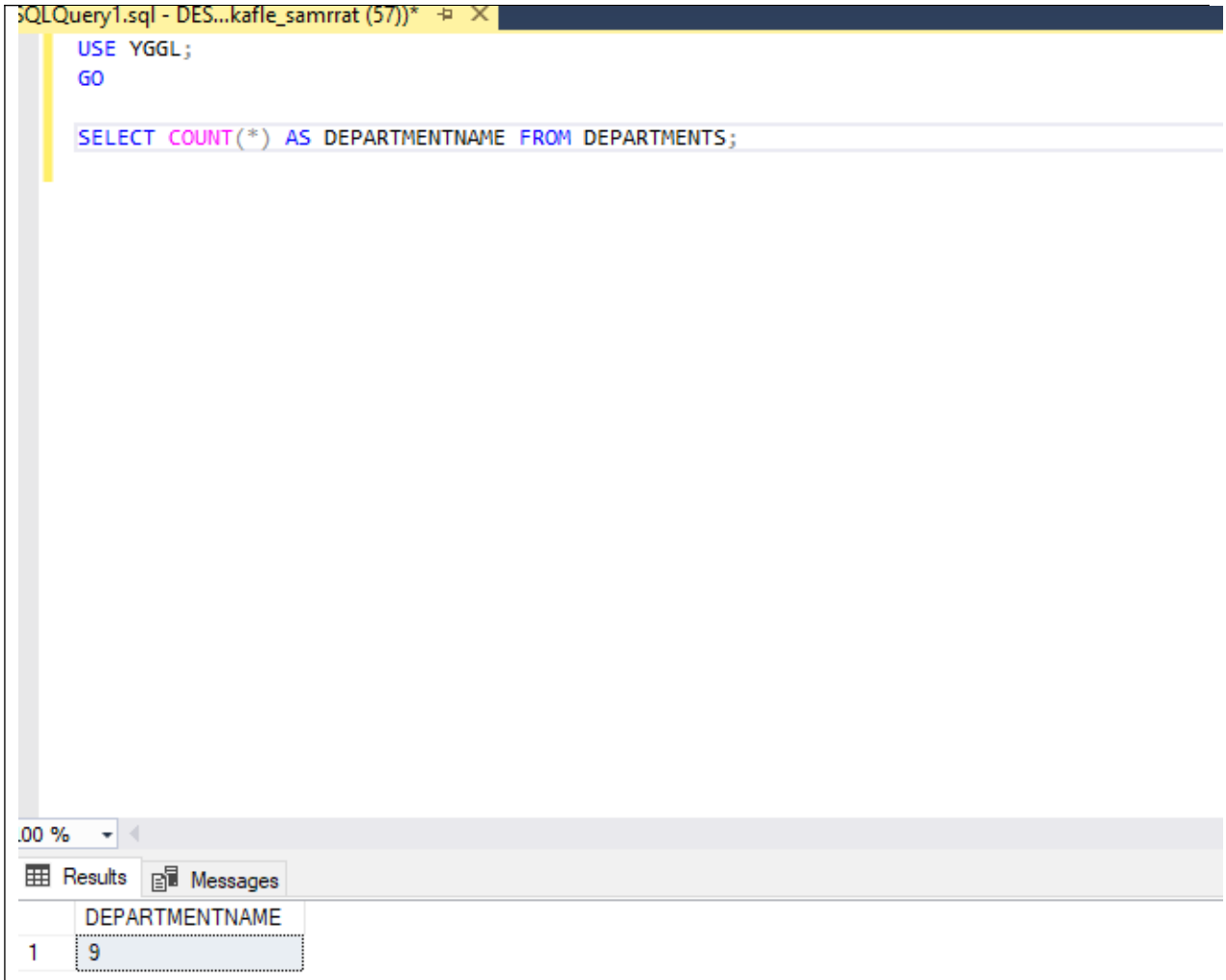
注：若查询结果集行数超过 7 行时，使用 TOP 选项限制返回行数为 7。

（二）数据库索引

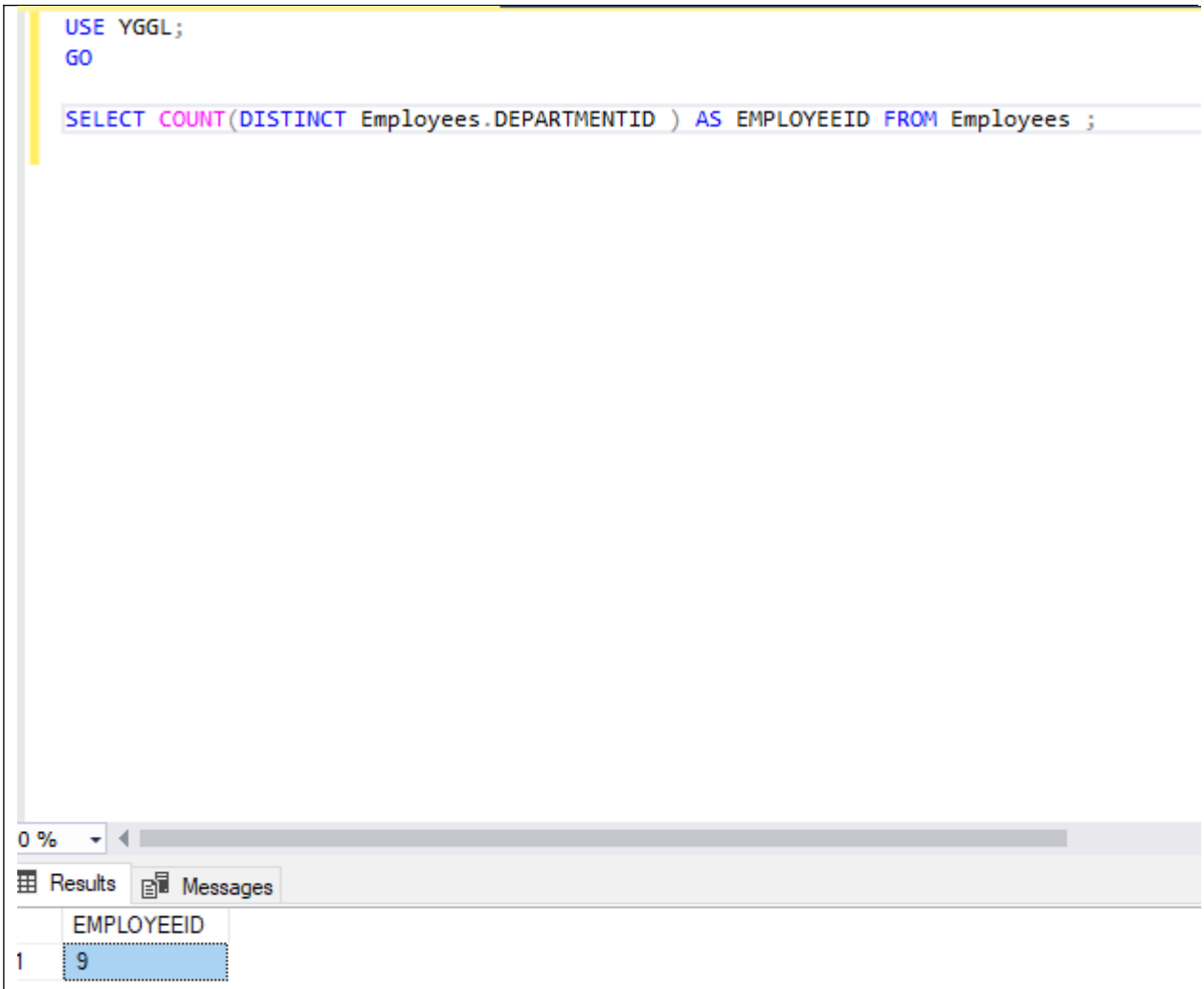
使用 T-SQL 语句，对向实验一中建立的数据库 YGGL 的三个表 Employees、Department 和 Salary 三个表建立相关索引【界面方式不做】。然后对索引进行重建与删除。

三.实验内容和结果

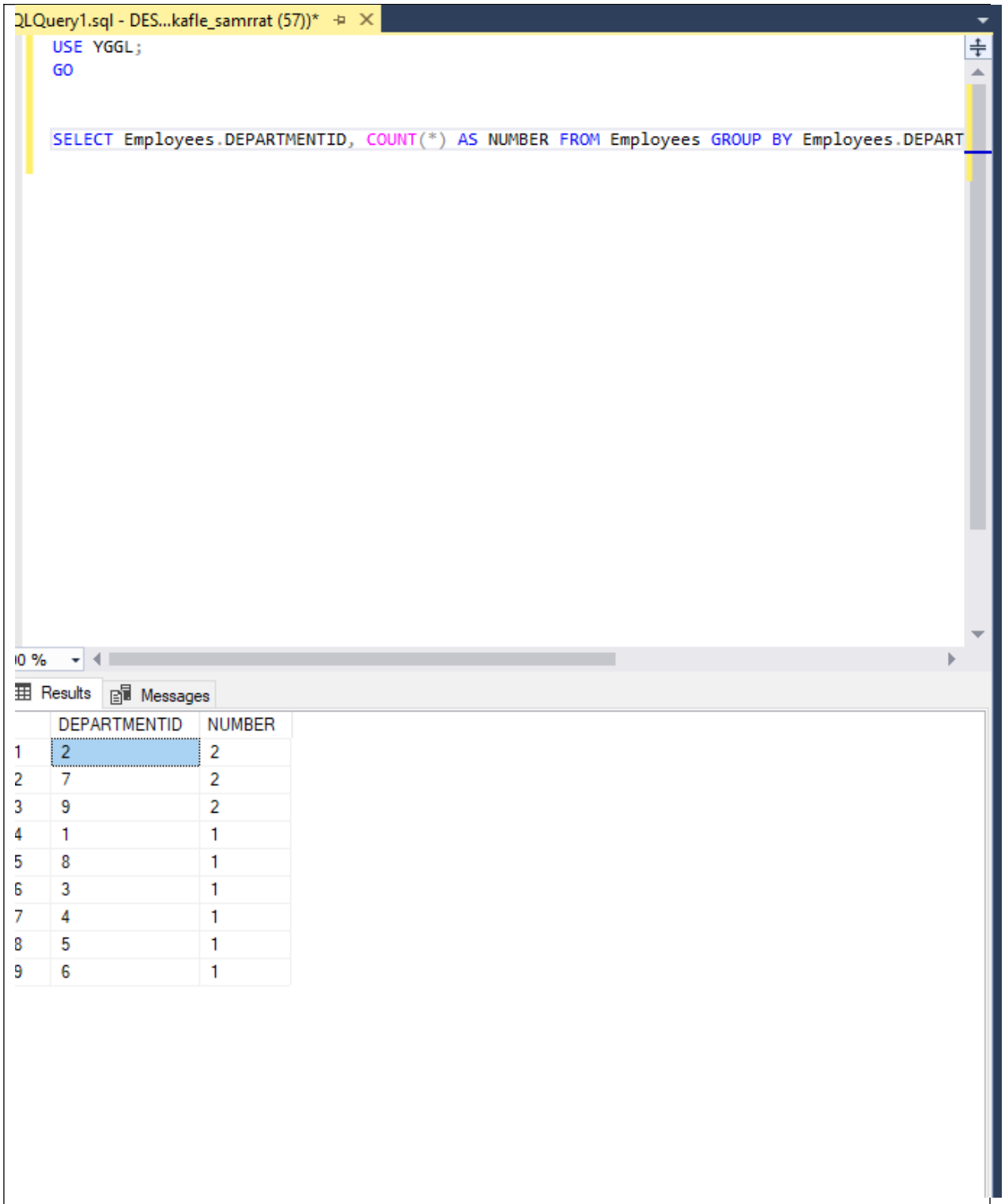
Select the total number of DEPARTMENTS;



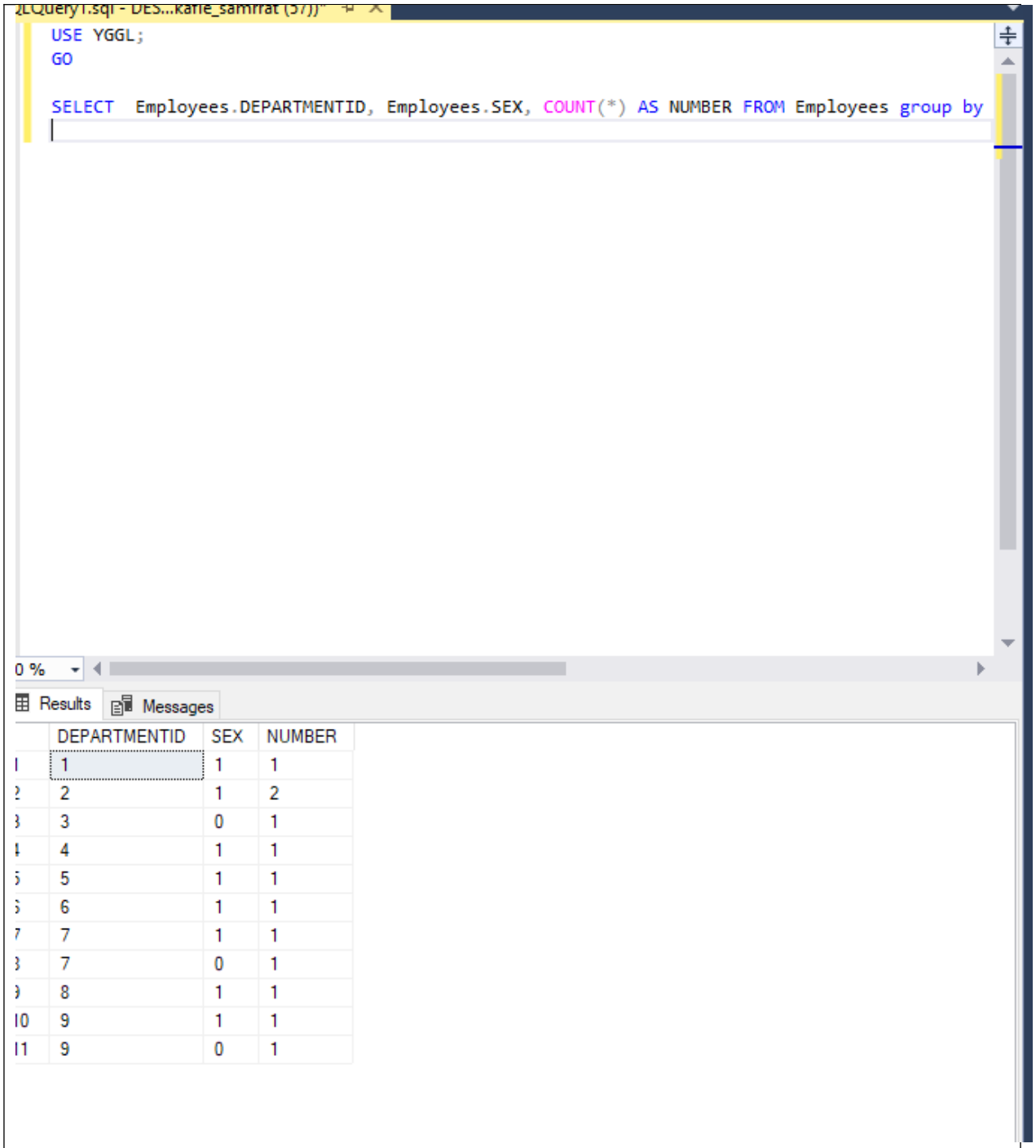
Select distinct departmentsid from Employee:



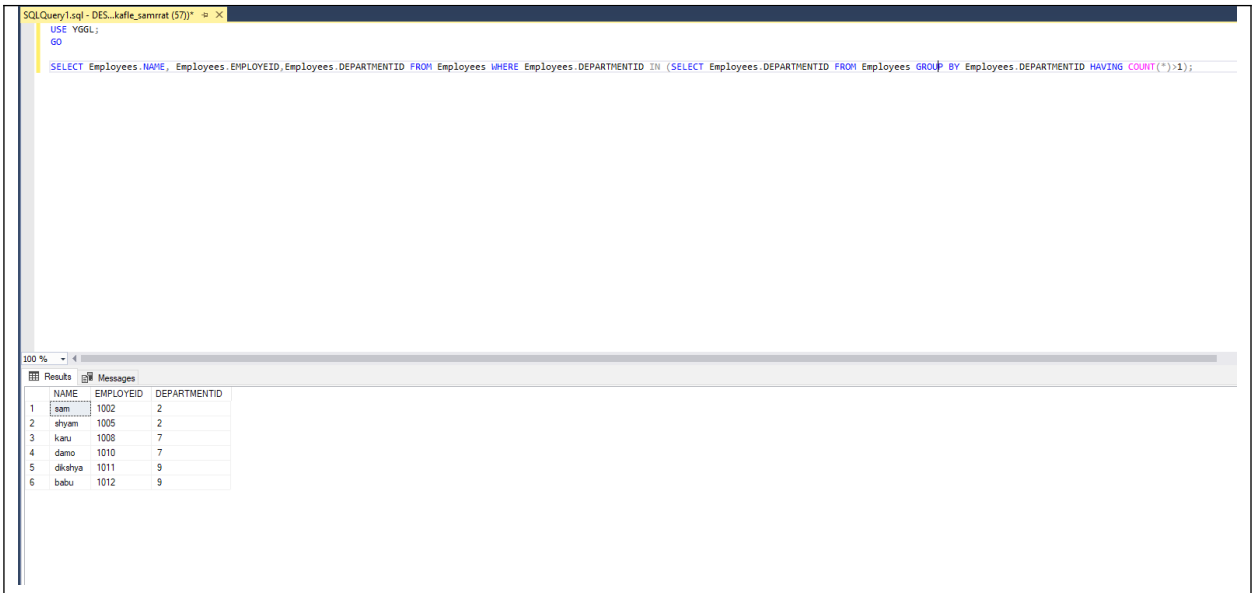
Select number of Employees from each department and sort by number from most to least;



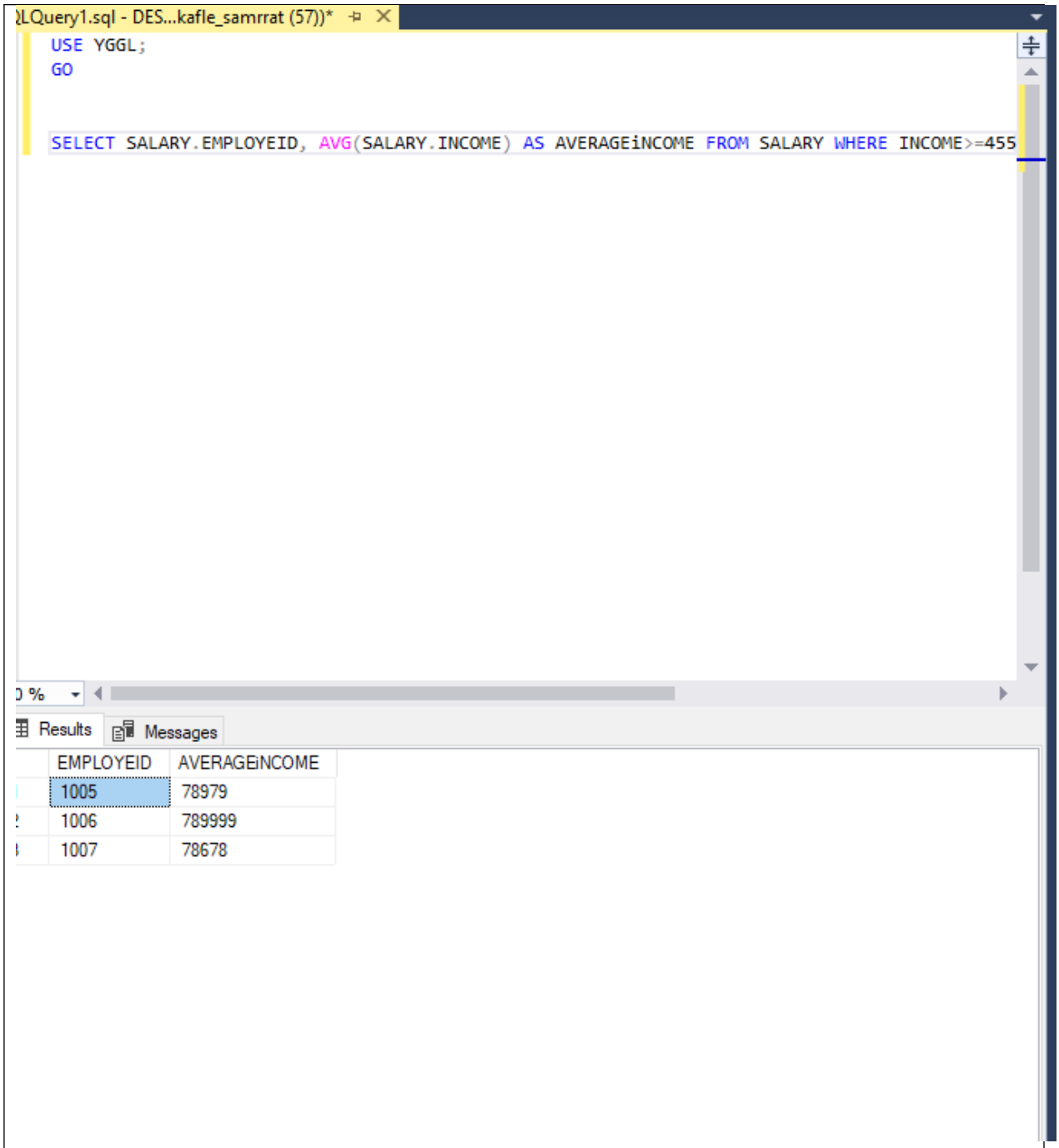
Select from employee and distinguish male and female:



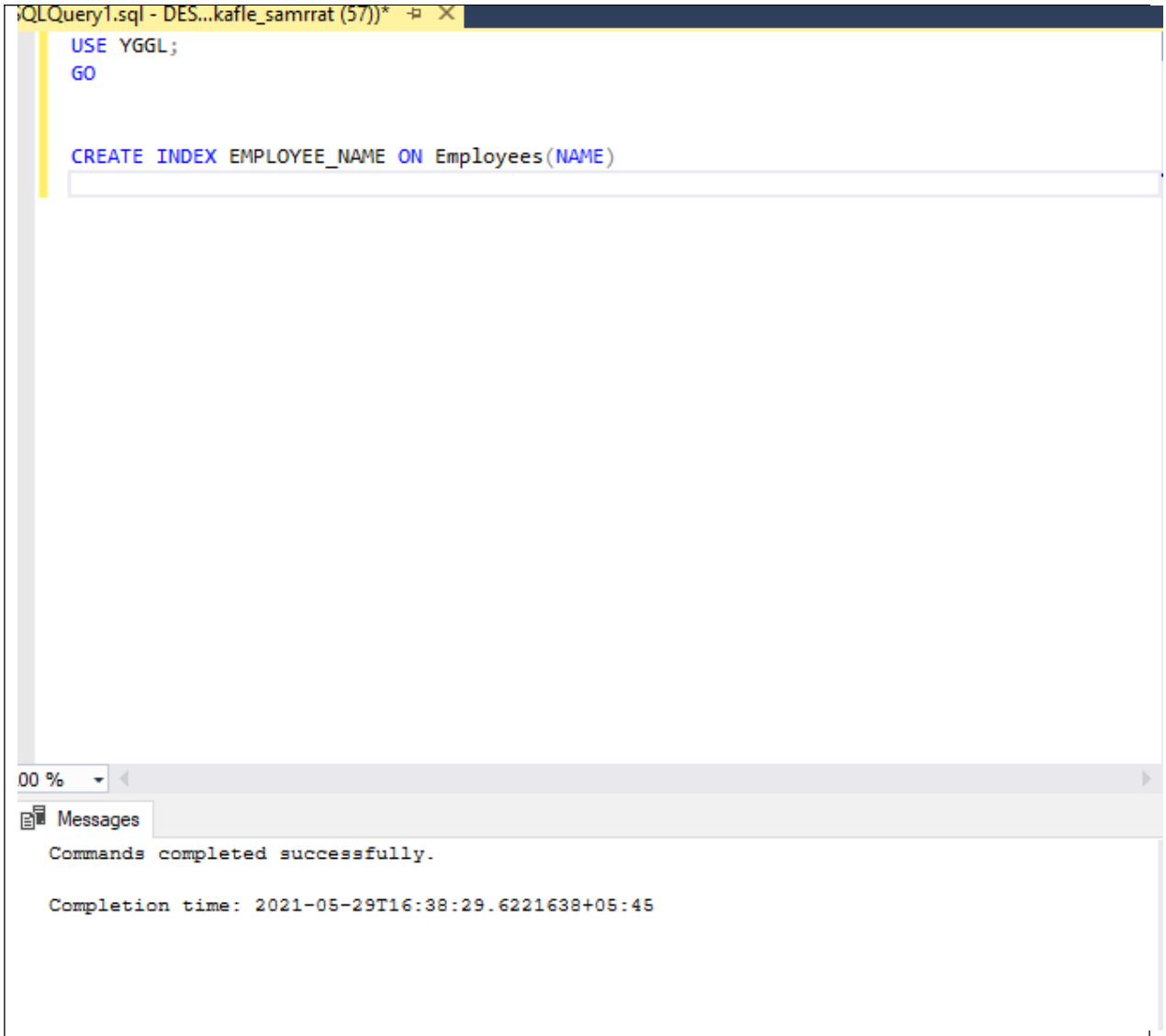
Select from Employee table and count the number of employee in the specific department in which departments contains more than one employee:



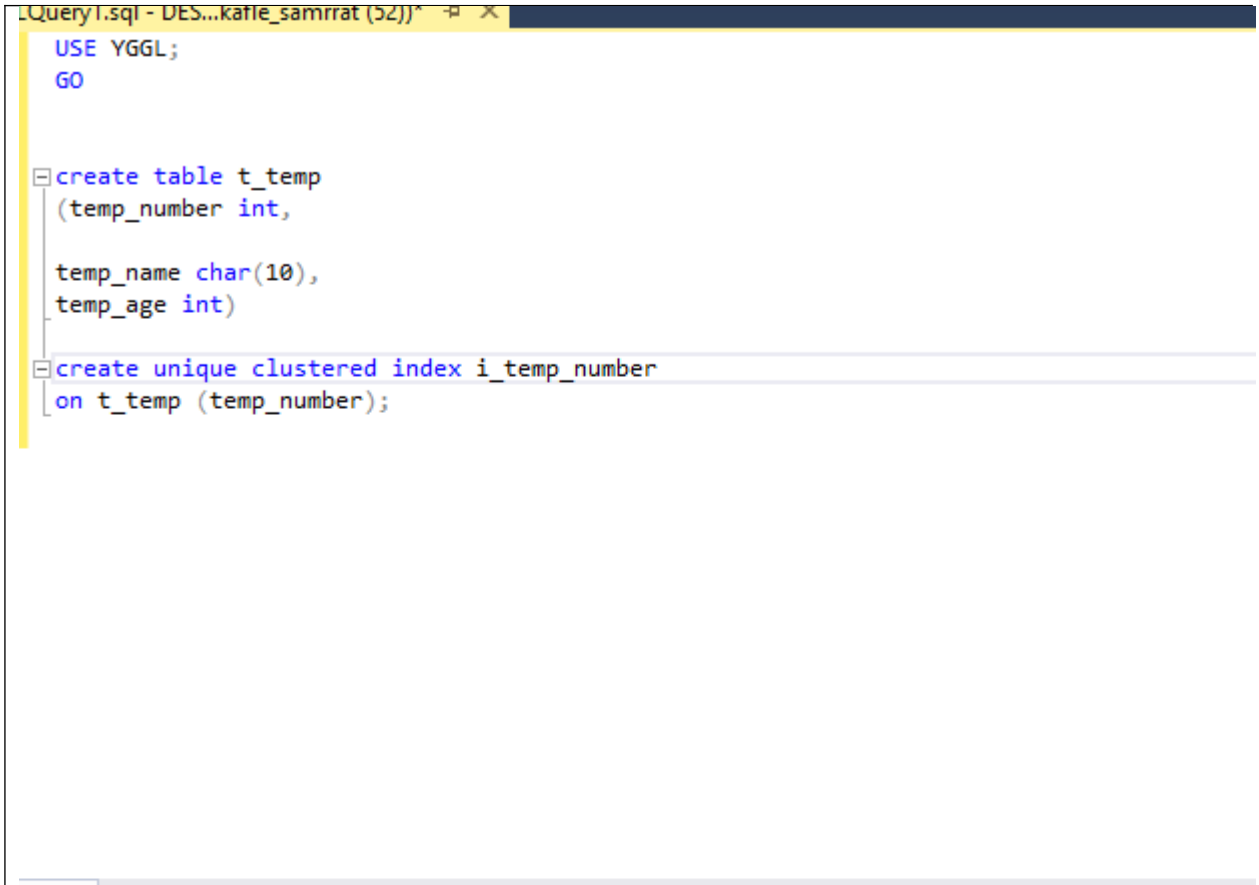
Select the information (Employeeid) from the salary whose averageincome is more than given amount:



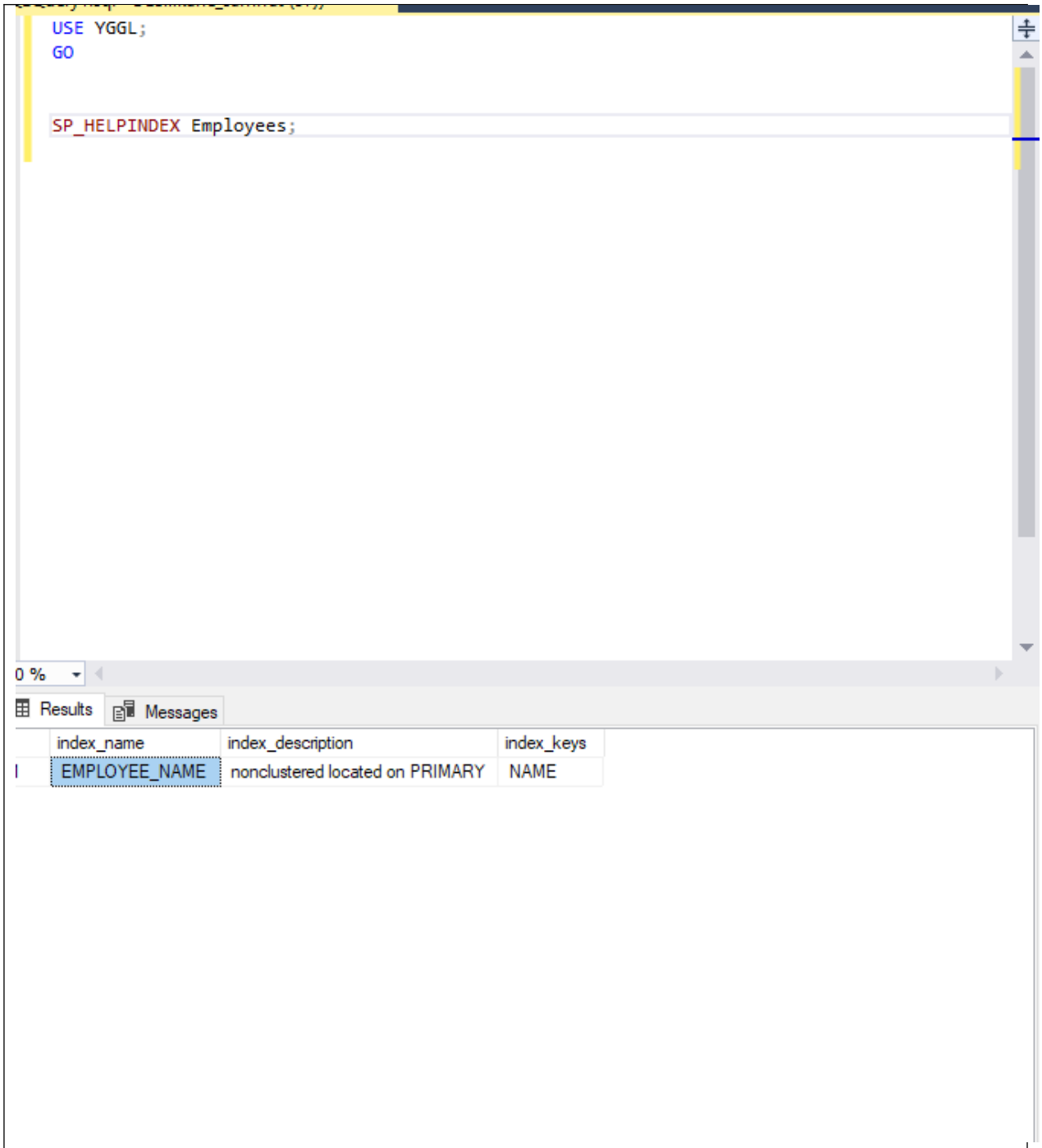
Create a non-clustered index for table EMPLOYEE with index field EMPLOYEE.NAME and index name EMPLOYEE_NAME.



Create a new table named TEMP and create a unique clustered index for this table with the index field temp_number and the index name i_temp_number.



Use the system stored procedure sp_helpindex to view the index information in the following syntax format:

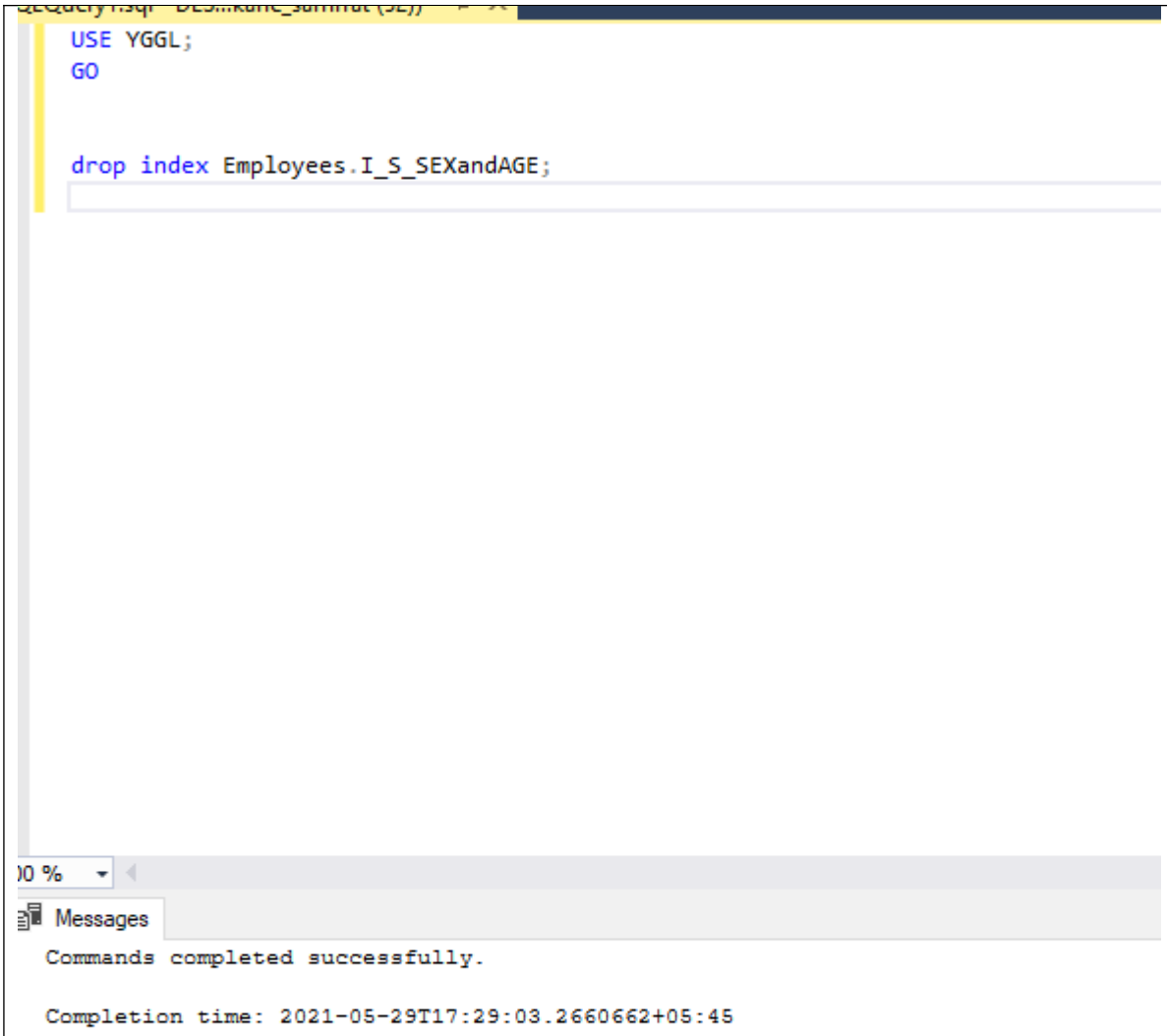


Change the index EMPLOYEE_NAME in table Employees to I_S_SEXANDAGE.

```
USE YGGL;
GO


sp_rename "Employees.EMPLOYEE_NAME ","I_S_SEXandAGE","INDEX";
```

```
Messages
Caution: Changing any part of an object name could break scripts and stored procedures.

Completion time: 2021-05-29T17:24:54.8144895+05:45
```

DROP INDEX I_S_SEXANDAGE FROM TABLE Employees:



实验总结及体会：

Through this Experiment I learn T-SQL statements how to use indexes in database and grouping sort query. When retrieving data using the

ORDER BY and GROUP BY clauses, you can significantly reduce the grouping and sorting time in a query.

实验七：**视图的应用**

一.实验目的与要求

（1）了解和掌握视图的概念、作用等；

（2）掌握视图的创建方法；

（3）掌握如何通过视图做查询、更新等操作。

二.实验内容

使用 T-SQL 语句，通过向实验一中建立的数据库 YGGL 的三个表 Employees、Department 和 Salary 创建视图，然后通过视图做查询、更新

操作等，并给出实验结果。最后，对视图进行删除。【新视图的命名同前约定】

（1）创建视图 Dep_VIEW，视图包含 Department 表的全部列。

（2）创建视图 EMP_VIEW，包含员工号码、姓名、所在部门名称和实际收入这几列。

（3）创建视图 SAL_VIEW，包含部门名称、部门平均收入、部门平均支出这几列。

（4）通过视图 Dep_VIEW 做一个查询操作，并说明查询操作内容。

（5）通过视图 Dep_VIEW 做一个插入操作，并说明插入操作内容。

（6）通过视图 EMP_VIEW 做一个修改操作，并说明修改操作内容，验证是否可行?

（7）通过视图 SAL_VIEW 做一个删除操作，并说明删除操作内容，验证是否可行?

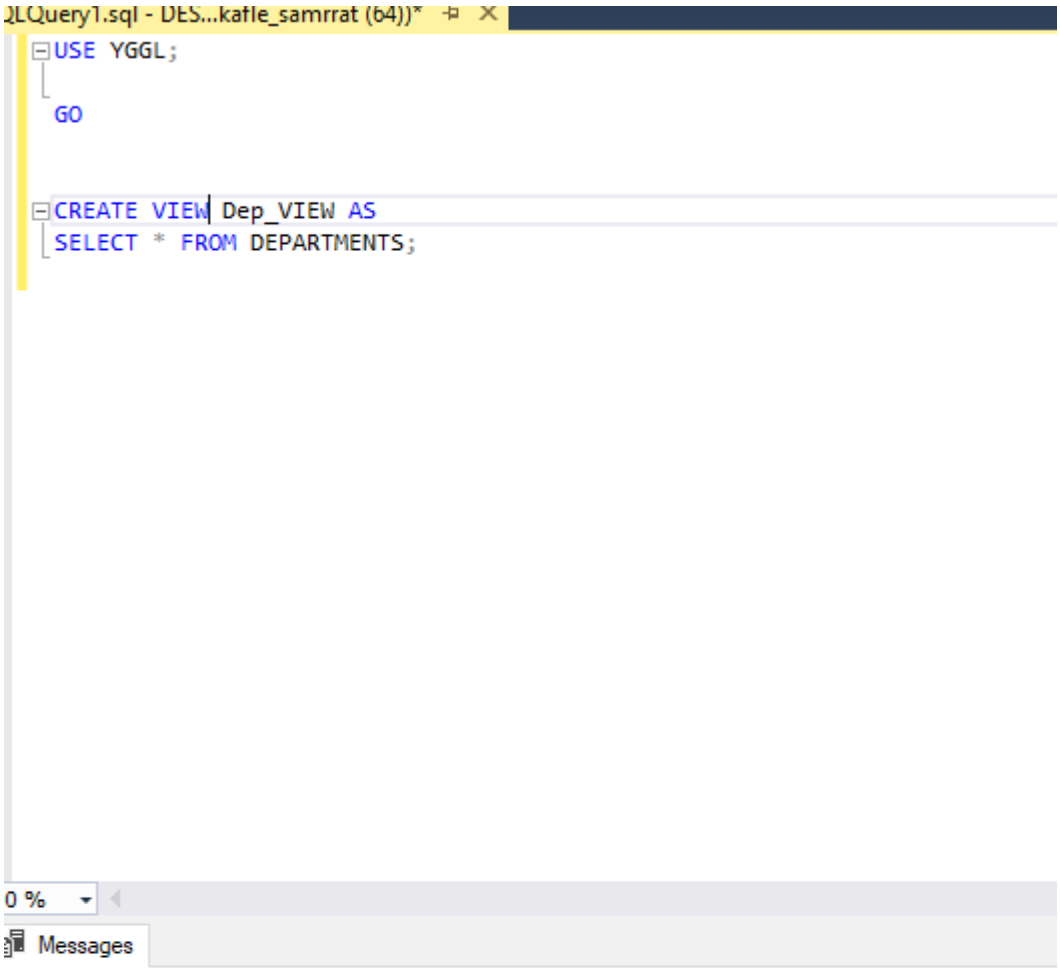
（8）删除视图 SAL_VIEW。

在“总结与体会”中附加回答下列问题：

（1）若视图关联了某表中的所有字段，而此时该表中添加了新的字段，视图中能否查询到该字段?

三.实验内容和结果

1. Create a view DEP_VIEW that contains all columns of the DEPARTMENT table.



2. Create a view EMP_VIEW, which contains the columns of employee number, name, department name and actual income

SQLQuery1.sql - DES...kafle_samrrat (64))*

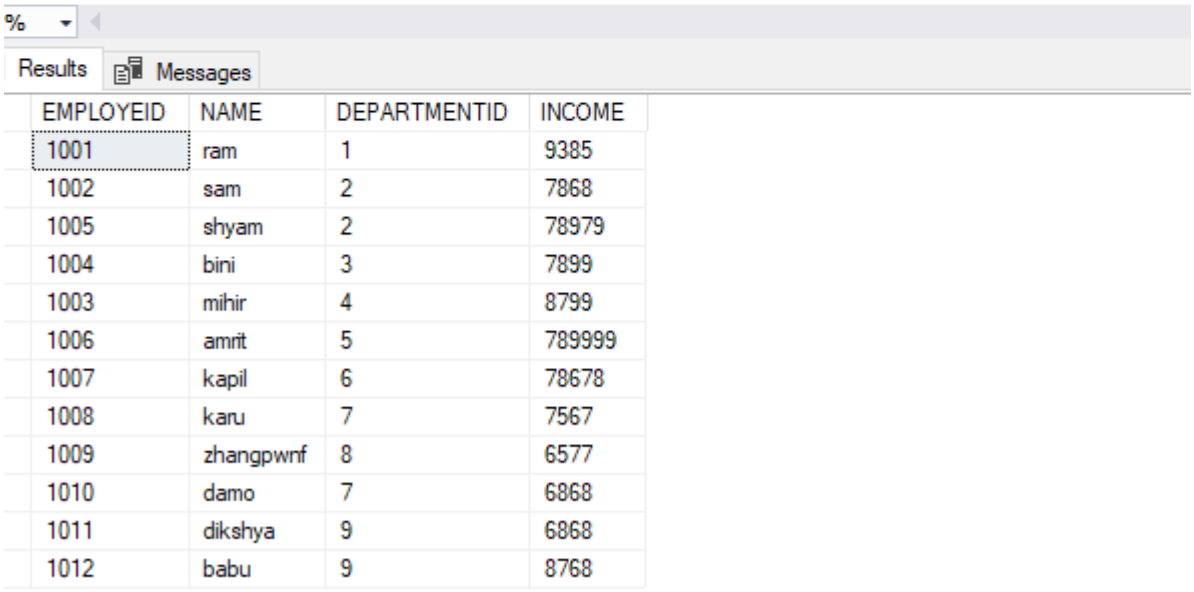
```
USE YGGL;
GO

CREATE VIEW EMP_VIEW (EMPLOYEID,NAME, DEPARTMENTID, INCOME) AS SELECT Employees.EMPLOYEID,Employees.NAME,Employees.DEPARTMENTID,SALARY.INCOME
FROM Employees,SALARY
WHERE Employees.EMPLOYEID = SALARY.EMPLOYEID;
```

100 %

Messages

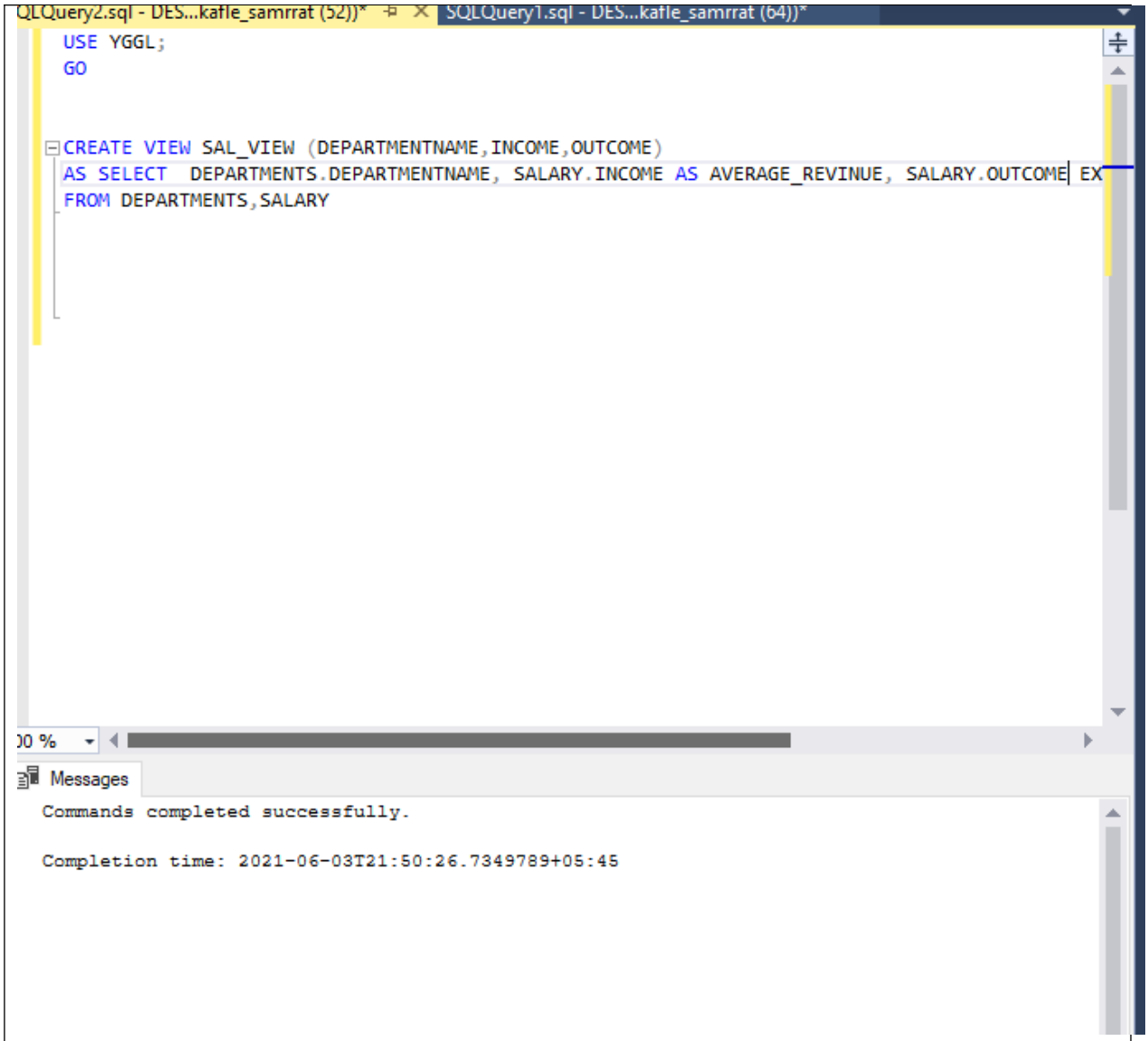
```
Commands completed successfully.

Completion time: 2021-06-03T21:07:12.0278503+05:45
```

```
USE YGGL;
GO

SELECT *
FROM EMP_VIEW;
```



EMPLOYEID	NAME	DEPARTMENTID	INCOME
1001	ram	1	9385
1002	sam	2	7868
1005	shyam	2	78979
1004	bini	3	7899
1003	mihir	4	8799
1006	amrit	5	789999
1007	kapil	6	78678
1008	karu	7	7567
1009	zhangpwnf	8	6577
1010	damo	7	6868
1011	dikshya	9	6868
1012	babu	9	8768

3. Create a view, SAL_VIEW, with columns for department name, department average revenue, and department average expenditure.


QLQuery2.sql - DES...kafle_samrrat (52))*
SQLQuery1.sql - DES...kafle_samrrat (64))*
USE YGGL;
GO
CREATE VIEW SAL_VIEW (DEPARTMENTNAME,INCOME,OUTCOME)
AS SELECT DEPARTMENTS.DEPARTMENTNAME, SALARY.INCOME AS AVERAGE_REVINUE, SALARY.OUTCOME EX
FROM DEPARTMENTS,SALARY
00 %
Messages
Commands completed successfully.
Completion time: 2021-06-03T21:50:26.7349789+05:45

```
USE YGGL;
GO


SELECT * FROM SAL_VIEW;
```

Results | Messages

DEPARTMENTNAME	INCOME	OUTCOME
COMPUTERSCIENCE	9385	987
COMPUTERSCIENCE	7868	768
COMPUTERSCIENCE	8799	567
COMPUTERSCIENCE	7899	657
COMPUTERSCIENCE	78979	6789
COMPUTERSCIENCE	789999	67899
COMPUTERSCIENCE	78678	5790
COMPUTERSCIENCE	7567	565
COMPUTERSCIENCE	6577	545
COMPUTERSCIENCE	6868	678
COMPUTERSCIENCE	6868	768
COMPUTERSCIENCE	8768	768
ELECTRONIC	9385	987
ELECTRONIC	7868	768

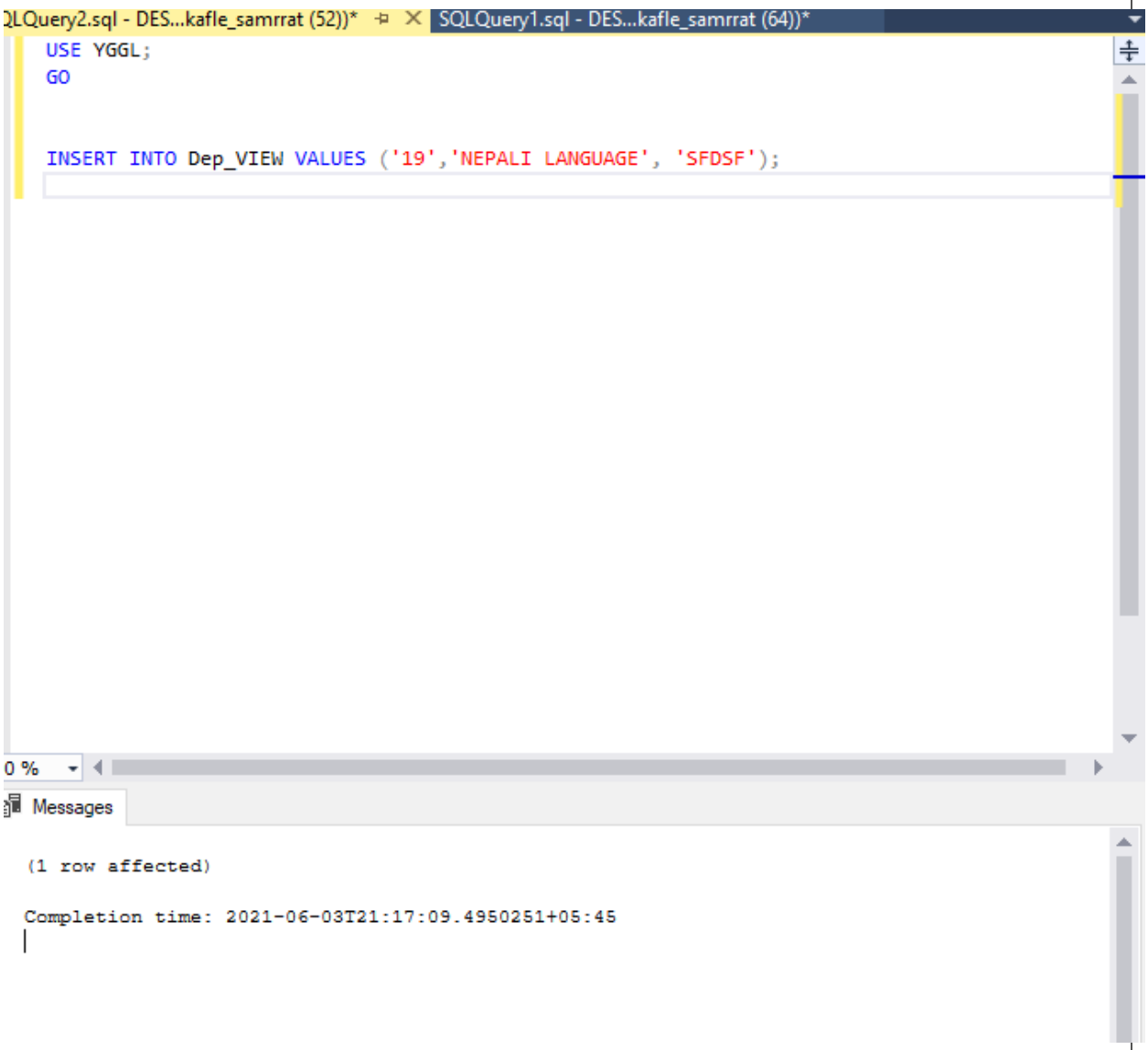
(4) Query the name of the department whose department number is 3 from the view Dep_VIEW.

```
USE YGGL;
GO


SELECT DEPARTMENTNAME, NOTE FROM Dep_VIEW
WHERE DEPARTMENTID = '7';
```

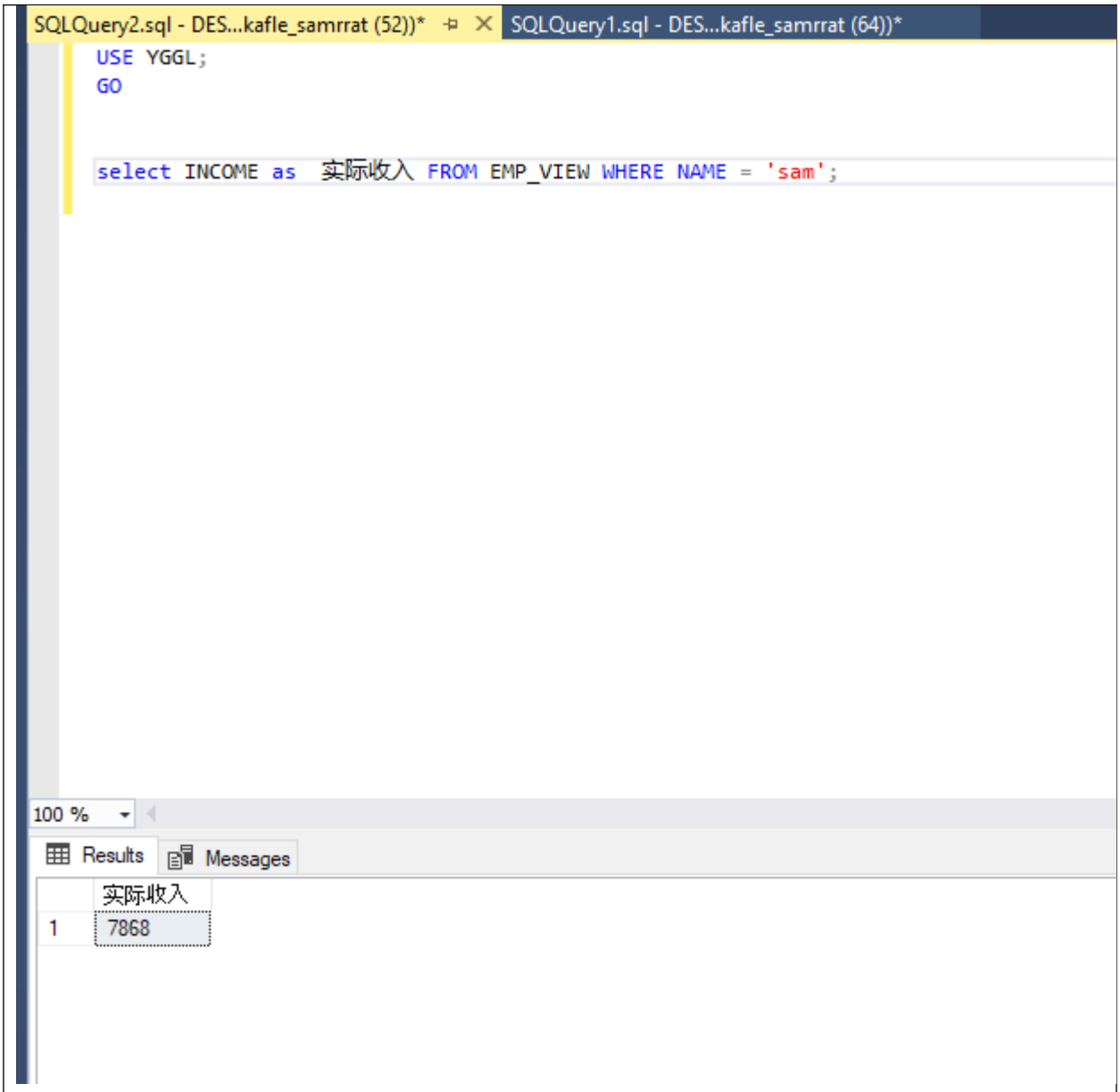
Results | Messages

	DEPARTMENTNAME	NOTE
1	LANGUAGE	NULL

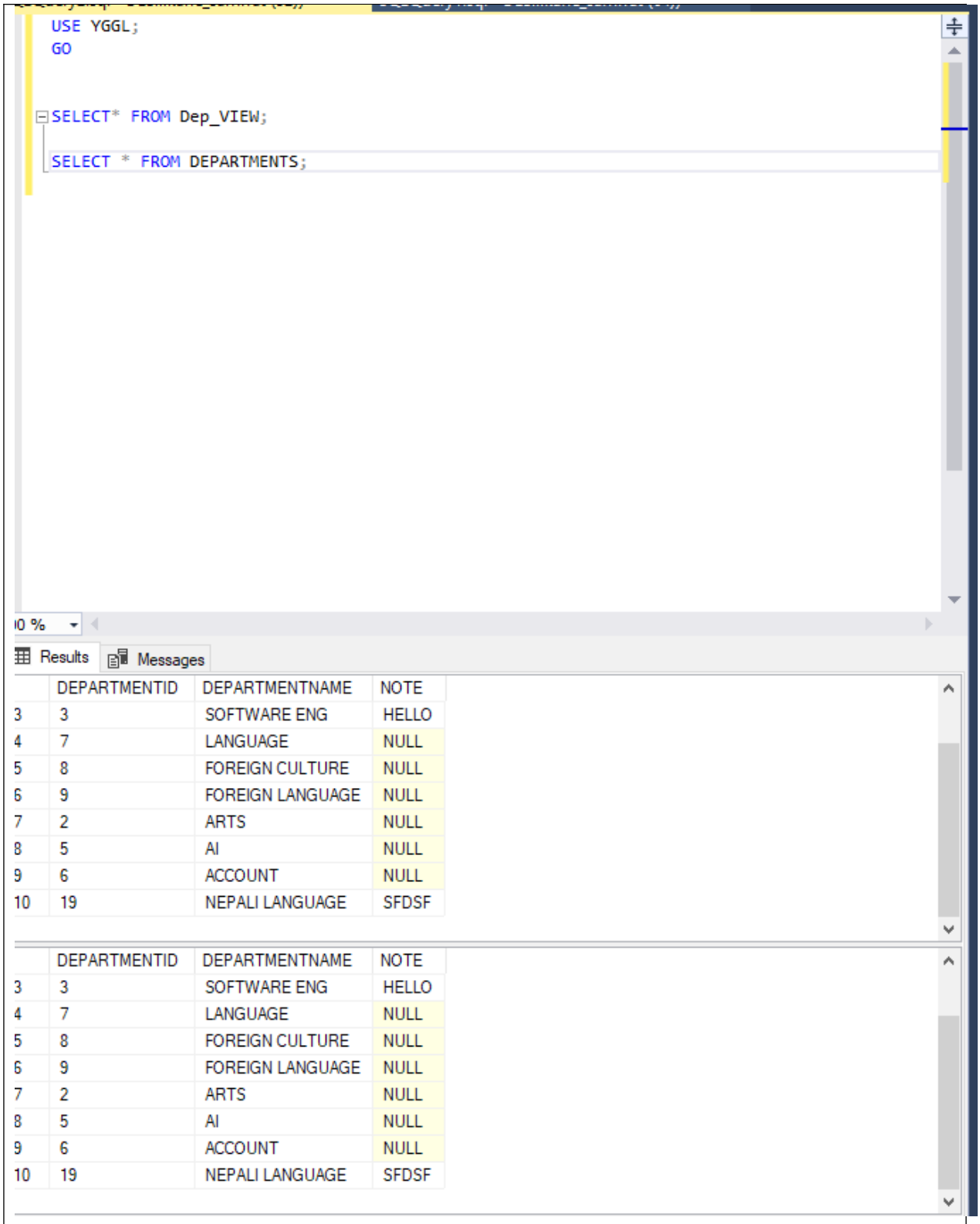
5. inert into Dep_View table :



6. Query the actual income of the employee named "sam" from the view Employees_view.



7. use the SELECT statement to see what has changed in the view Dep_VIEW and the base table Departments, respectively.



	DEPARTMENTID	DEPARTMENTNAME	NOTE
3	3	SOFTWARE ENG	HELLO
4	7	LANGUAGE	NULL
5	8	FOREIGN CULTURE	NULL
6	9	FOREIGN LANGUAGE	NULL
7	2	ARTS	NULL
8	5	AI	NULL
9	6	ACCOUNT	NULL
10	19	NEPALI LANGUAGE	SFDSF

	DEPARTMENTID	DEPARTMENTNAME	NOTE
3	3	SOFTWARE ENG	HELLO
4	7	LANGUAGE	NULL
5	8	FOREIGN CULTURE	NULL
6	9	FOREIGN LANGUAGE	NULL
7	2	ARTS	NULL
8	5	AI	NULL
9	6	ACCOUNT	NULL
10	19	NEPALI LANGUAGE	SFDSF

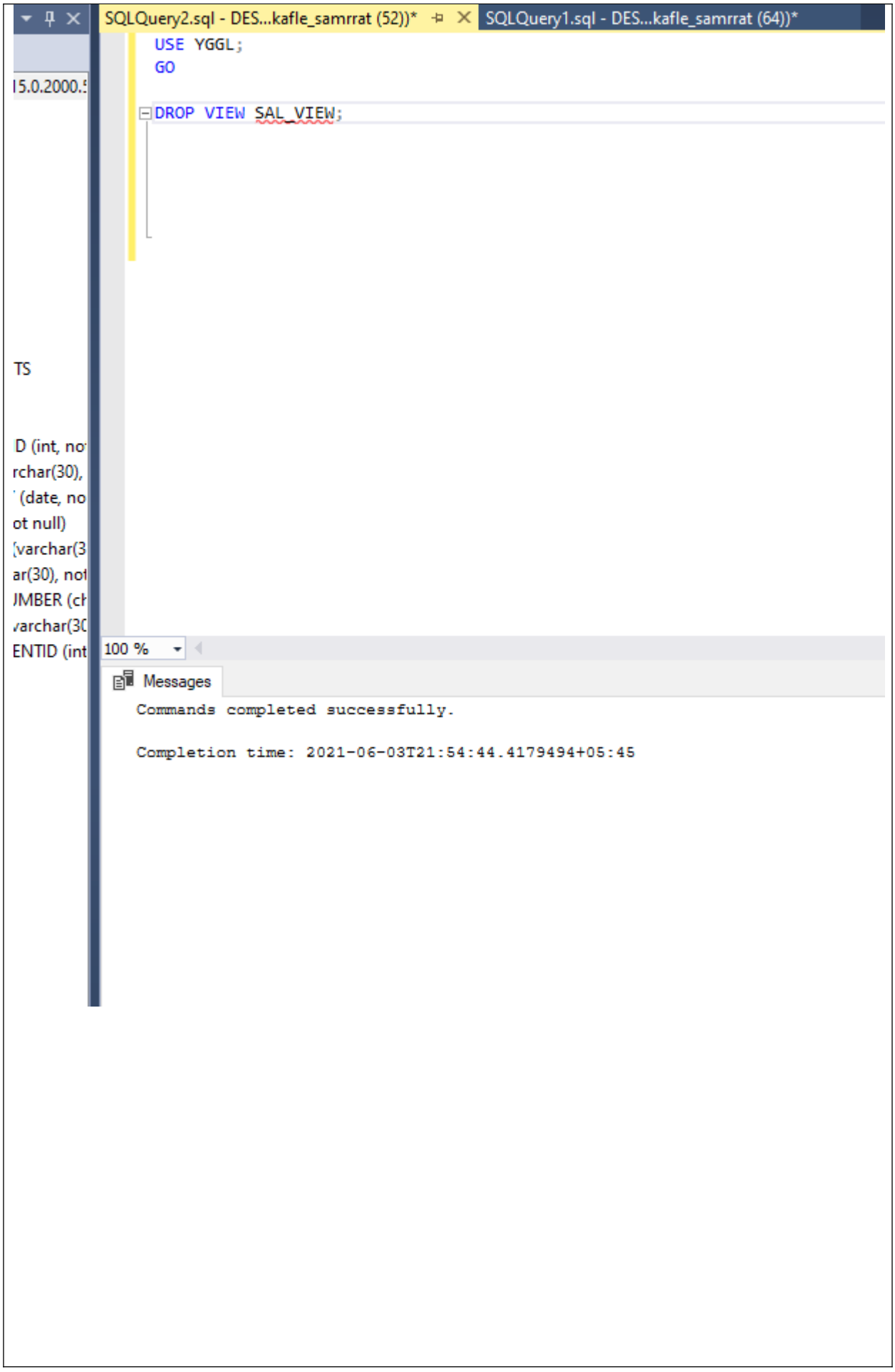
8. Is it feasible to make a change from EMP_VIEW and explain what the change is?

View or function 'EMP_VIEW' is not updatable because the modification affects multiple base tables.

9. Do a delete operation from view SAL_VIEW and explain the contents of the delete operation to verify whether it is feasible?

View or function 'SAL_VIEW' is not updatable because the modification affects multiple base tables.

10, DROP VIEW SAL_VIEW:


SQLQuery2.sql - DES...kafle_samrrat (52))*
SQLQuery1.sql - DES...kafle_samrrat (64))*
USE YGGL;
GO
DROP VIEW SAL_VIEW;
100 %
Messages
Commands completed successfully.
Completion time: 2021-06-03T21:54:44.4179494+05:45

实验总结及体会：

A view is a table derived from one table or multiple tables (or views). A view is different from a table called a virtual table. The data

corresponding to the view is not actually stored. Only the definition of the view is stored in the database. Essentially, the view exists in the form of T-SQL commands. When the user uses the view, the system calls the data in the basic table according to the definition of the view, and what is shown to the user is the required result executed according to the definition of the view.

Basically view is derived from one table or different that's why we can not modify the contents of the VIEW if it is made from more than one table.

实验八： **数据库完整性应用**

一. 实验目的与要求

理解和掌握数据库完整性的含义、作用和实现方法等。

二. 实验内容

使用 T-SQL 语句，对向实验一中建立的数据库 YGGL 的三个表 Employees、Department 和 Salary 三个表进行 UNIQUE 约束、CHECK 约束等的创建和应用。

（1）使用 CREATE TABLE 语句为创建一新表 Department2，包含 DepartmentName 和 Manager 两个属性，并为 Manager 列定义 unique 约束。

（2）创建新表 SC，包含“学号”、“课程号”和“性别”三列，（“学号”、“课程号”）定义为主键，作为表的约束，并为其命名；性别只能包含男或女。

（3）向 SC 表插入数据，“性别”列插入“男”和“女”以外的字符，查看会发生什么情况。

（4）创建新表 Salary2，结构与 Salary 相同，但 Salary2 表不允许 OutCome 列大于 InCome 列。

（5）向 Salary2 表中插入数据，查看 OutCome 值比 InCome 值大会发生什么情况。

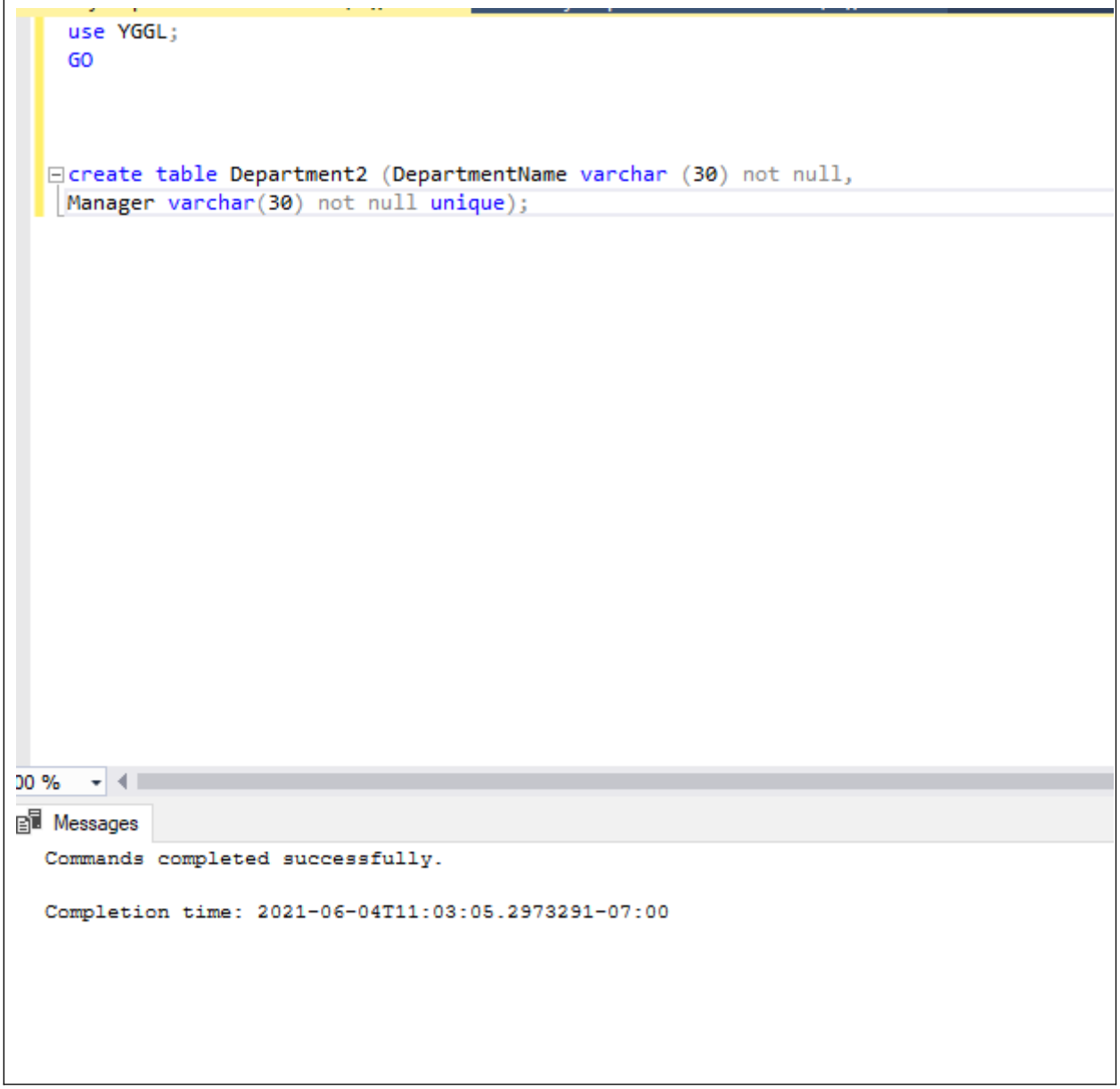
（6）创建一个新表 Employees2，只考虑“员工编号”和“出生日期”两列，出生日期必须晚于 1980 年 1 月 1 日。

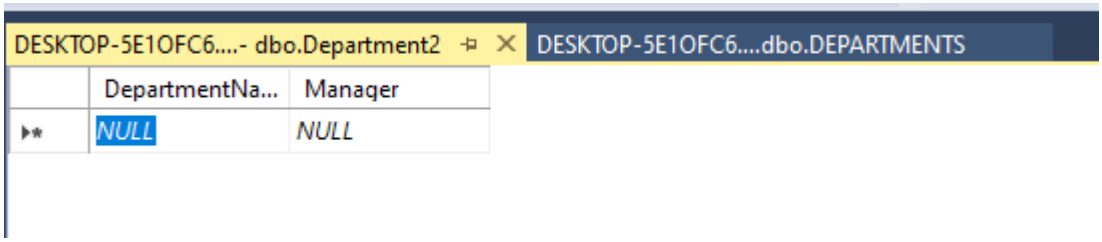
（7）使用 ALTER TABLE 语句向 Salary 表中的 EmployeeID 列上添加一个外键，要求当 Employees 表中要删除或修改与 EmployeeID 值有关的行时，检查 Salary 表中有没有与该 EmployeeID 值相关的记录，如果存在则

拒绝更新 Employees 表。

三. 实验内容和结果

Create a new TABLE Department2 with attributes DepartmentName and Manager, and define UNIQUE constraint for the Manager column.





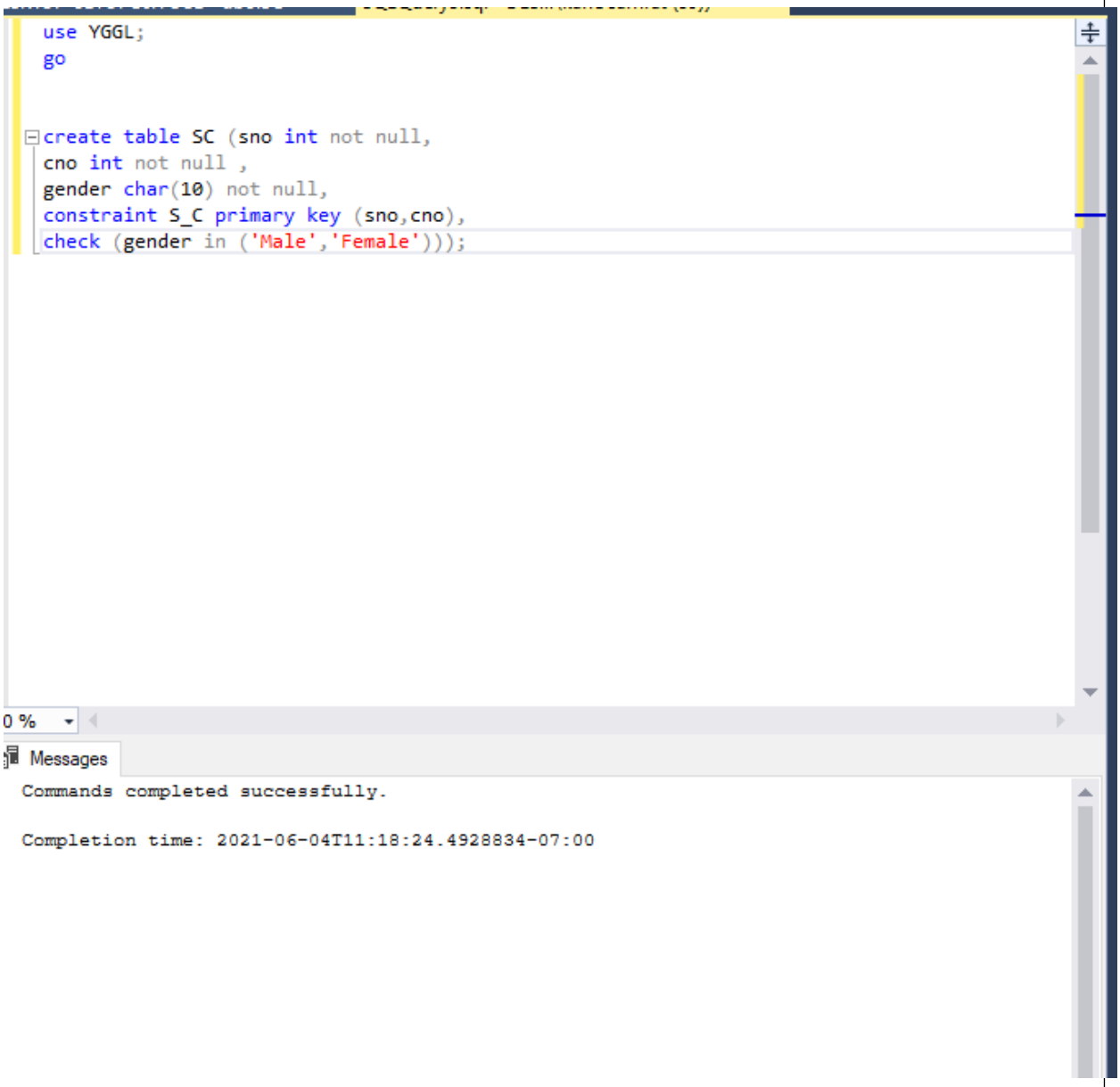
(2) create a new table SC, containing "student number", "course number" and "gender" three columns, (" student number ", "course number") define the primary key, as a constraint on the table, and name it; Gender can only include male or female.

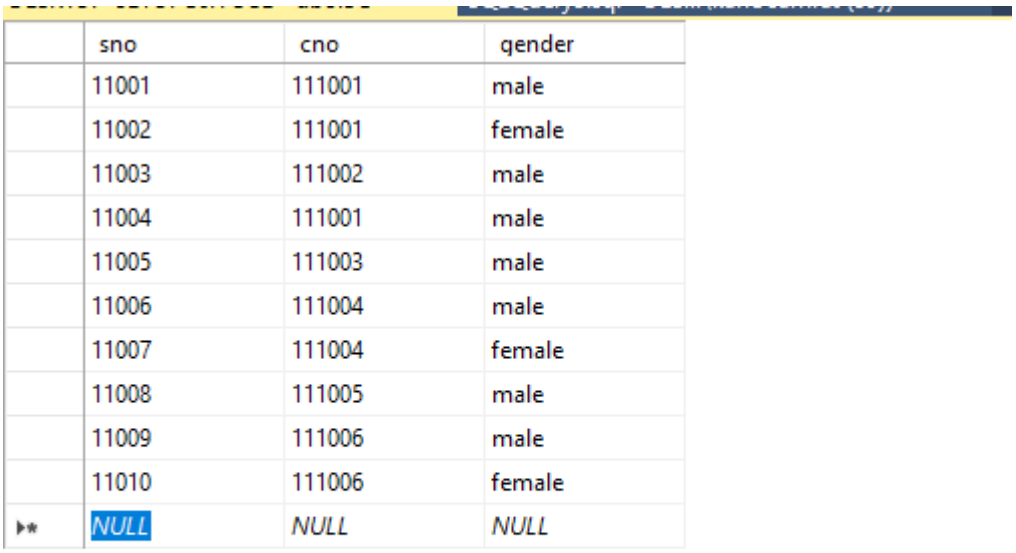
```
use YGGL;
go


create table SC (sno int not null,
cno int not null ,
```

gender char(10) not null,

constraint S_C primary key (sno,cno),

check (gender in ('Male','Female')));

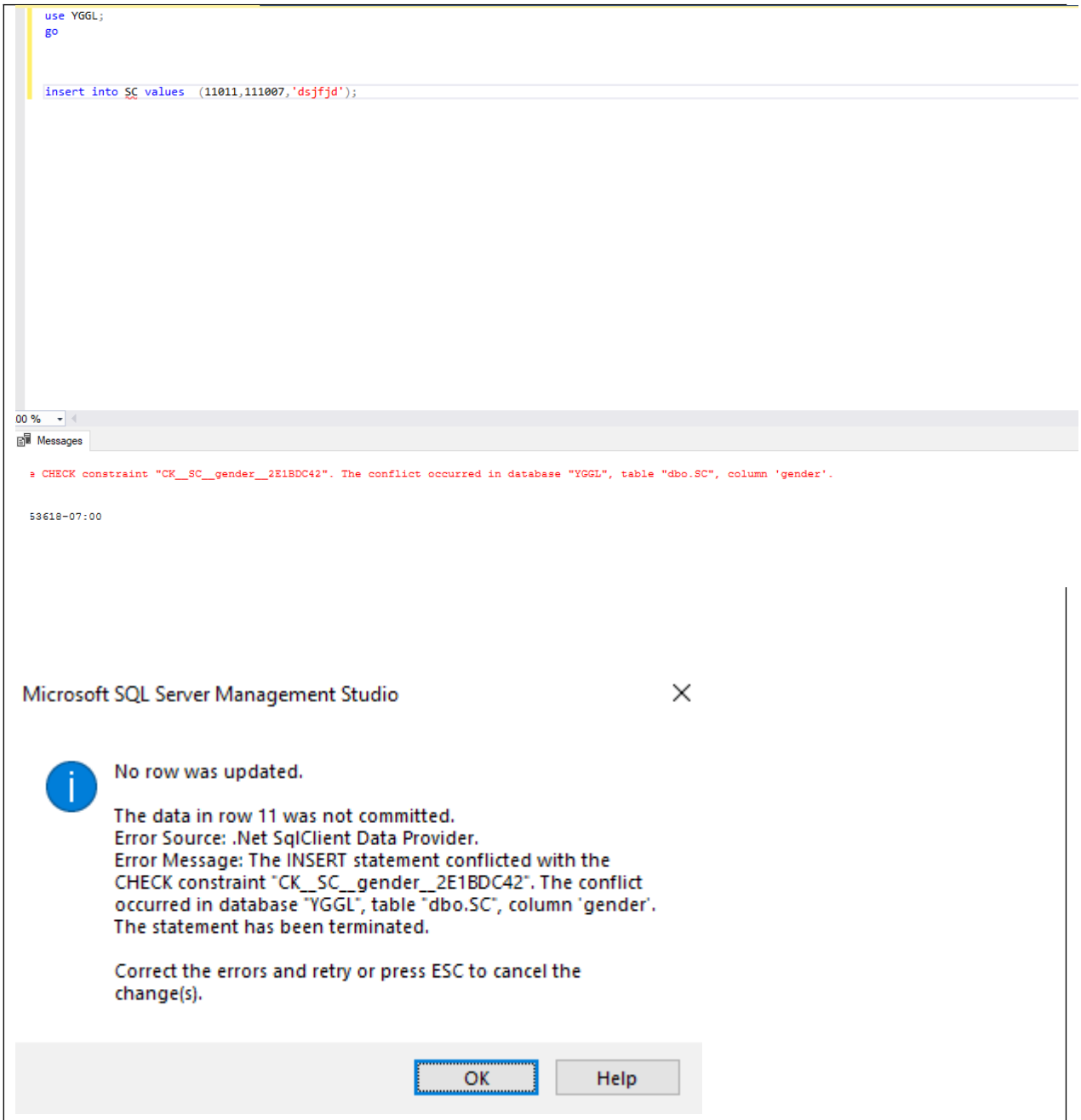




	sno	cno	qender
	11001	111001	male
	11002	111001	female
	11003	111002	male
	11004	111001	male
	11005	111003	male
	11006	111004	male
	11007	111004	female
	11008	111005	male
	11009	111006	male
	11010	111006	female
▶*	NULL	NULL	NULL

(3) Insert data into the SC table, and insert characters other than male and female in the "gender" column to see what happens.

It shows following error:



4. Create table Salary2 with the same structure as Salary, but do not allow the value of OutCome column is to be larger than the the value of

InCome column.

```
use YGGL;
go



create table SALARY2 (

EMPLOYEEID int not null,
INCOME FLOAT NOT NULL,
OUTCOME float not null,
CHECK (INCOME >= OUTCOME));
```

00 %

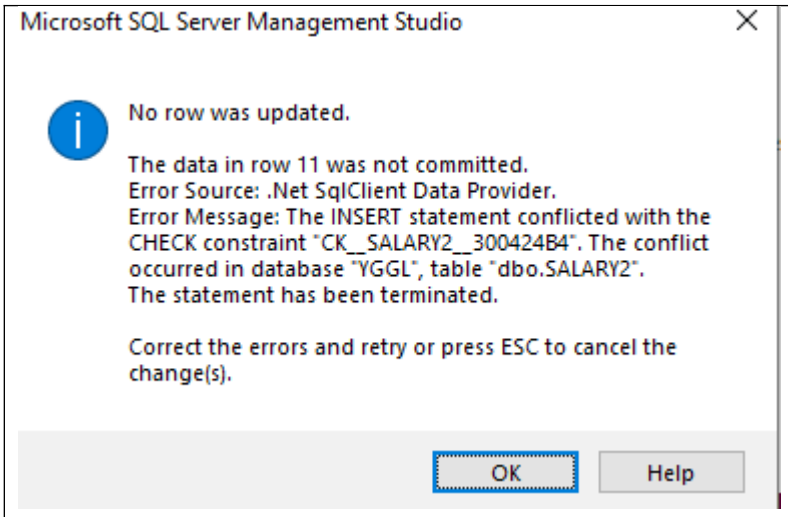
Messages

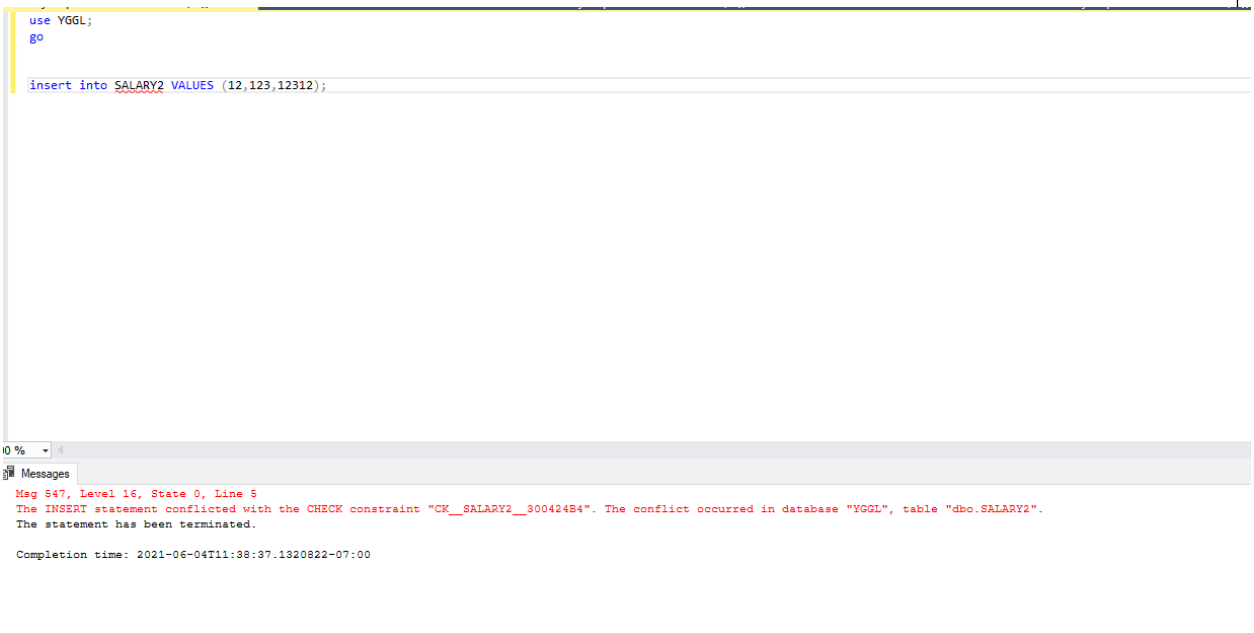
```
Commands completed successfully.

Completion time: 2021-06-04T11:34:40.8613523-07:00
```

	EMPLOYEEID	INCOME	OUTCOME
	1	1008	89
	2	1009	98
	3	8978	900
	4	8798	765
	5	7889	78
	6	889	89
	7	8799	878
	8	989	89
	9	908	98
	10	798989	8987
*	NULL	NULL	NULL

(5) Insert data into the Salary2 table to see what happens when the OutCome value compares with the InCome value.

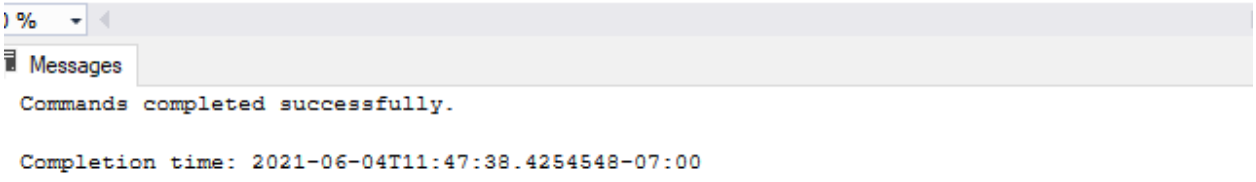
It shows following error:





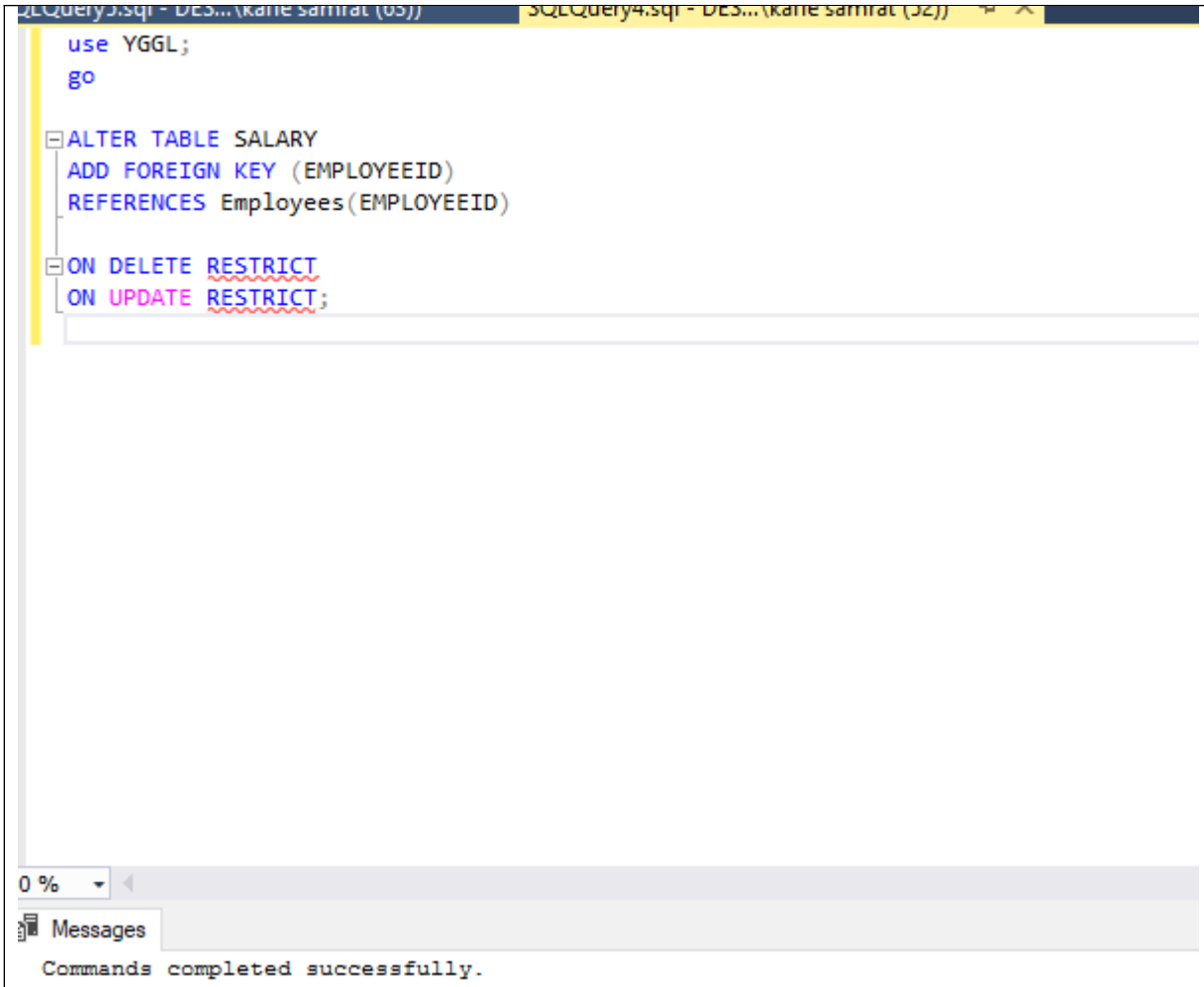
(6) create a new table Employees2, only consider "employee number" and "date of birth" columns, the date of birth must be later than January 1, 1980.

```
use YGGL;
go


CREATE TABLE EMPLOYEES2 (EMPLOYEEID INT NOT NULL,
BIRTHDAY DATE NOT NULL,
CHECK  ( BIRTHDAY >= '1980-01-01'));
```



(7) using the ALTER TABLE statement to the Salary the EmployeeID column in the TABLE to add a foreign key, requirements in the Employees TABLE to delete or modify the row is related to the EmployeeID value, check the Salary have the EmployeeID value related records in the TABLE, Refuse to update the Employees table if it exists.

```
use YGGL;
go

ALTER TABLE SALARY
ADD FOREIGN KEY (EMPLOYEEID)
REFERENCES Employees(EMPLOYEEID)
ON DELETE RESTRICT
ON UPDATE RESTRICT;
```



实验总结及体会：

Through this experiment, I learned the basic database SQL commands, When adding a primary key, the primary key index is automatically created. Ordinary indexes need to be added and specified manually. The most efficient index in the table is the primary key index.

实验九： **自定义函数及存储过程应用**

四. 实验目的与要求

（1）掌握 T-SQL 的变量定义及基本控制语句的使用；

（2）掌握自定义函数、存储过程的定义及使用方法。

五. 实验内容

使用 T-SQL 语句，对向实验一中建立的数据库 YGGL 的三个表 Employees、Department 和 Salary 三个表进行自定义函数、存储过程的创建和调用。

（1）定义一个函数实现如下功能：对于一个给定的 DepartmentID 值，查询该值在 Departments 表中是否存在，若存在则返回 0，不存在返回-1。并写一段 T-SQL 程序调用此函数。

（2）创建存储过程，要求当一个员工的工作年份大于 6 年时，将其转到经理办公室工作。并执行此存储过程。

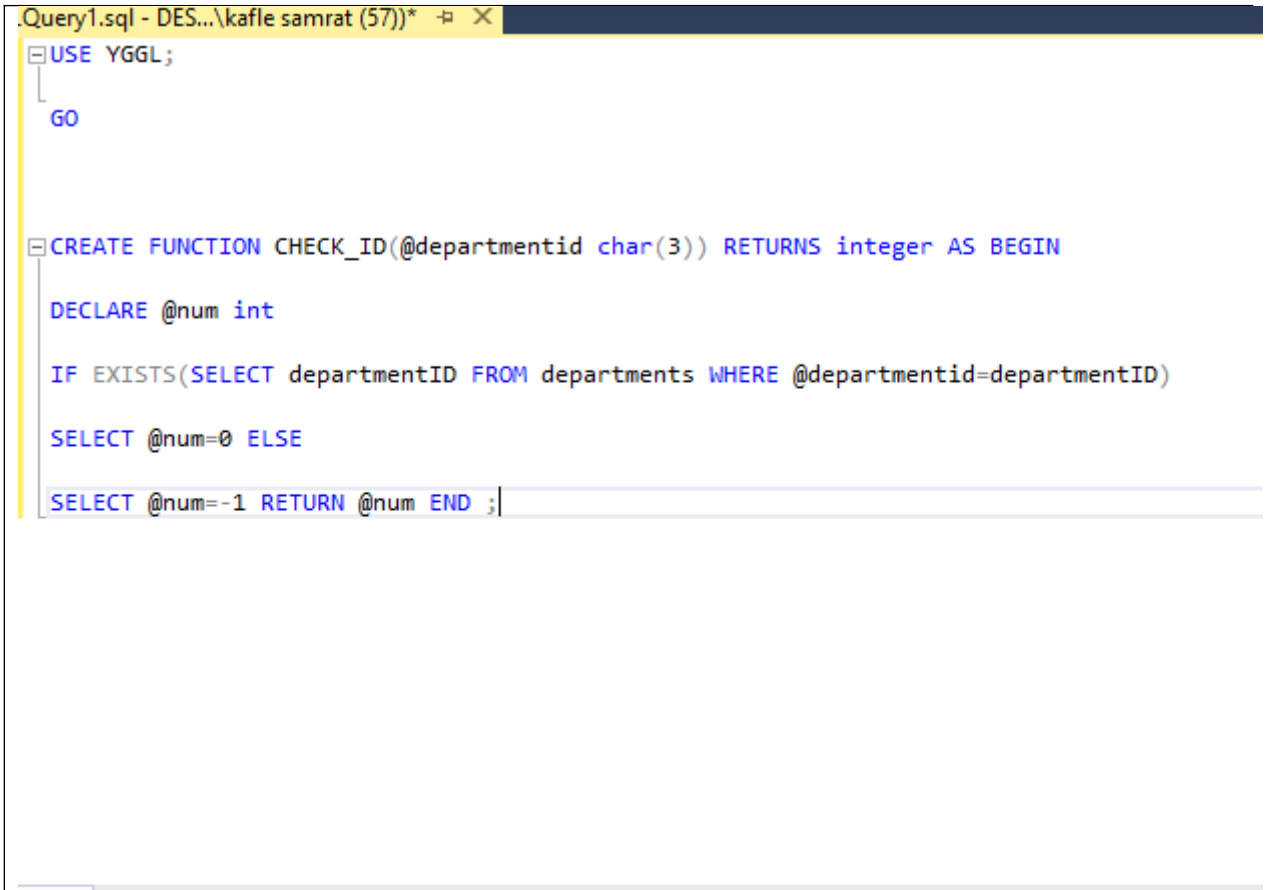
（3）创建存储过程，若员工的学历为硕士或博士，将其收入增加 1000 元。并执行此存储过程。

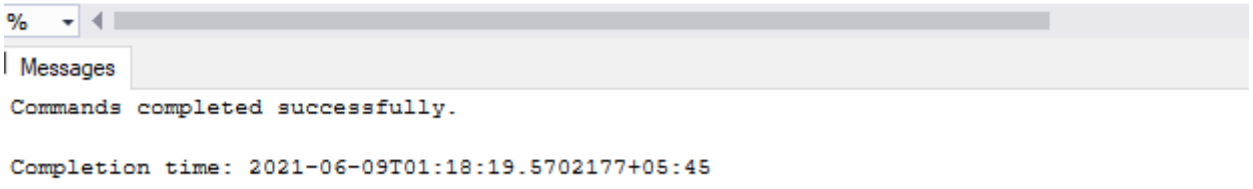
六. 实验内容和结果

(1) Define a function that queries whether a given Departmentid value exists in the Departments table and returns 0 if it does, or -1 if it does not. And write a T-SQL program to call this function.

```
USE YGGL;

GO

CREATE FUNCTION CHECK_ID(@departmentid char(3)) RETURNS integer AS BEGIN
```

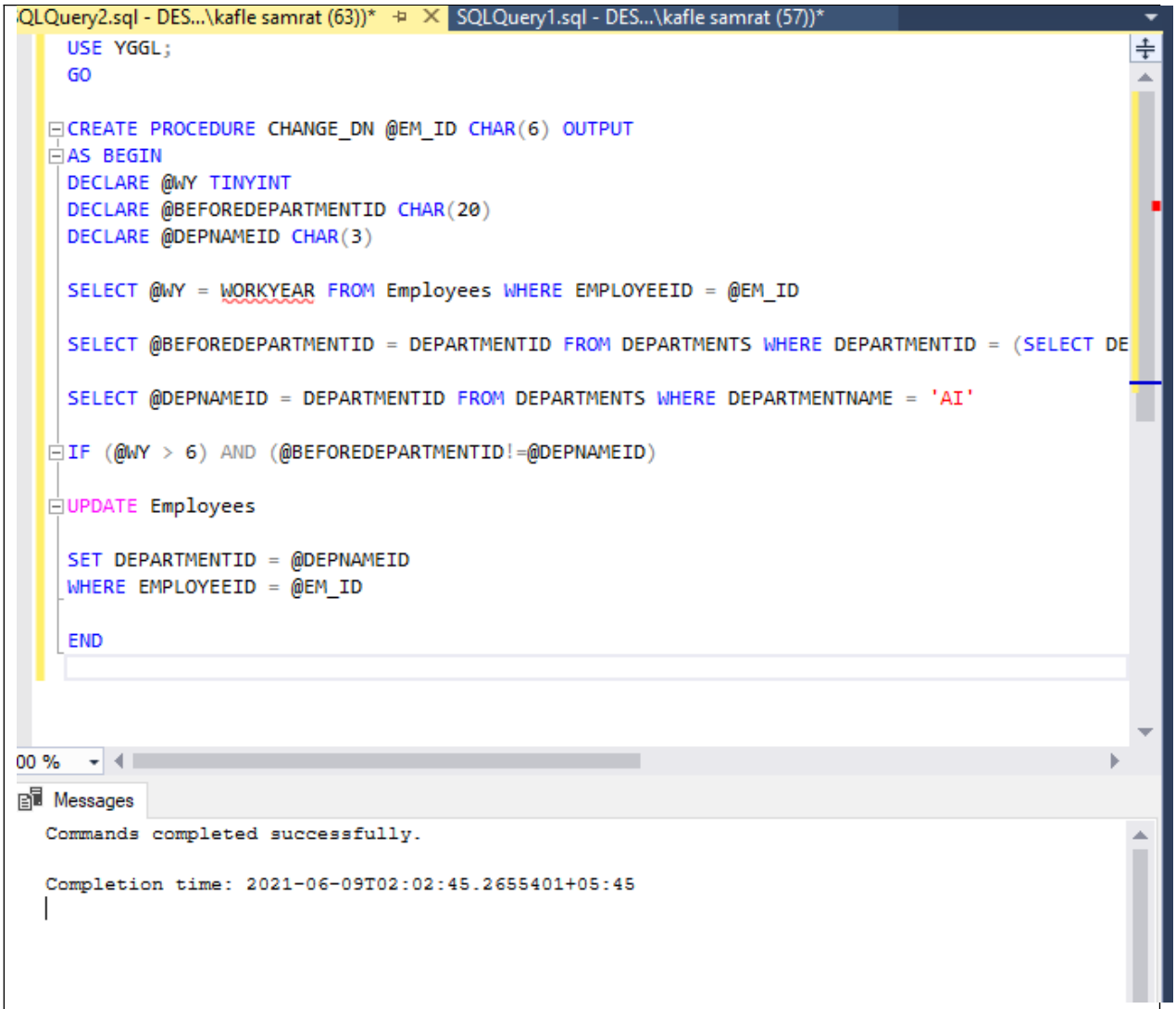
```
DECLARE @num int

IF EXISTS(SELECT departmentID FROM departments WHERE @departmentid=departmentID)

SELECT @num=0 ELSE

SELECT @num=-1 RETURN @num END ;
```





(2) Create a stored procedure that requires an employee to be transferred to the manager's office when his/her working years are greater than 6 years. And execute the stored procedure.

NOTE: Department ‘manager's office’ is replaced with ‘AI’;



USE YGGL;

GO

CREATE PROCEDURE CHANGE_DN @EM_ID CHAR(6) OUTPUT

AS BEGIN

DECLARE @WY TINYINT

DECLARE @BEFOREDEPARTMENTID CHAR(20)

DECLARE @DEPNAMEID CHAR(3)

```
SELECT @WY = WORKYEAR FROM Employees WHERE EMPLOYEEID =
@EM_ID

SELECT @BEFOREDEPARTMENTID = DEPARTMENTID FROM
DEPARTMENTS WHERE DEPARTMENTID = (SELECT DEPARTMENTID
FROM Employees WHERE EMPLOYEEID = @EM_ID)

SELECT @DEPNAMEID = DEPARTMENTID FROM DEPARTMENTS WHERE
DEPARTMENTNAME = 'AI'

IF (@WY > 6) AND (@BEFOREDEPARTMENTID!=@DEPNAMEID)

UPDATE Employees

SET DEPARTMENTID = @DEPNAMEID
WHERE EMPLOYEEID = @EM_ID
END
```

(3) Create a stored procedure. If the employee has a master's degree or a doctor's degree, increase his/her income by 1000 yuan. And execute the stored procedure.

```
USE YGGL;
GO

CREATE PROCEDURE ADDINCOME @EDU CHAR(4) OUTPUT
AS BEGIN

UPDATE SALARY
SET INCOME = INCOME + 1000

WHERE EMPLOYEEID IN (SELECT EMPLOYEEID FROM Employees WHERE EDUCATION = @EDU)
END
```

```
Messages
Commands completed successfully.

Completion time: 2021-06-09T02:07:52.9217044+05:45
```

For doctor degree:

```
DECLARE @EDU CHAR(4)
SET @EDU = 'DOCTORAL'
EXEC ADDINCOME @EDU
```

For masters degree:

```
DECLARE @EDU CHAR(4)
SET @EDU = 'master'
EXEC ADDINCOME @EDU
```

实验总结及体会：

Through this experiment, I have a more familiar understanding of TSOL command and a more profound understanding of T-SQL logic. All in all, the gain was great.

实验十： 触发器应用

七. 实验目的与要求

（1）掌握触发器的创建和使用方法。

（2）了解 Inserted 逻辑表和 Deleted 逻辑表的使用。

八. 实验内容

使用 T-SQL 语句，依据实验一中建立的数据库 YGGL 中的相关数据表，完成下列实验任务。给出实验结果，验证触发器的正确性。

（1）删除 Salary 表和 Employees 表之间建立的外键关系。

（2）创建 INSTEAD OF 触发器，当向 Salary 表中插入记录时，先检查 EmployeeID 列上的值在 Employees 中是否存在，如果存在则执行插入操作；如果不存在则提示“员工不存在”。向 Salary 表中插入数据来验证效果。

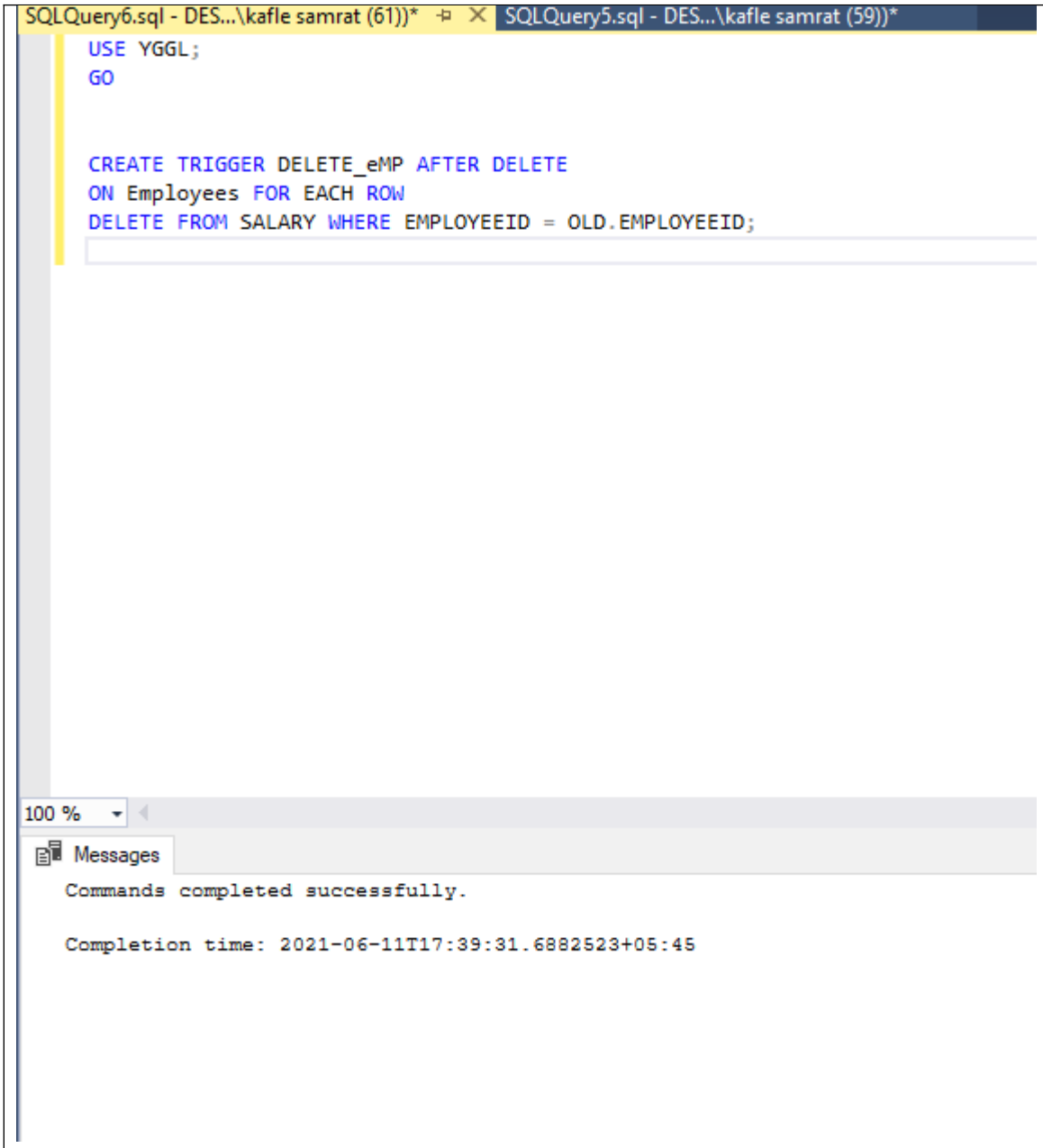
（3）创建触发器，当修改表 Employees 时，若将 Employees 表中员工的工作时间增加 1 年，则将收入增加 500，若增加 2 年则增加 1000，依次增加。若工作时间减少则无变化。

（4）创建 update 触发器，当 Salary 表中 Income 值增加 500

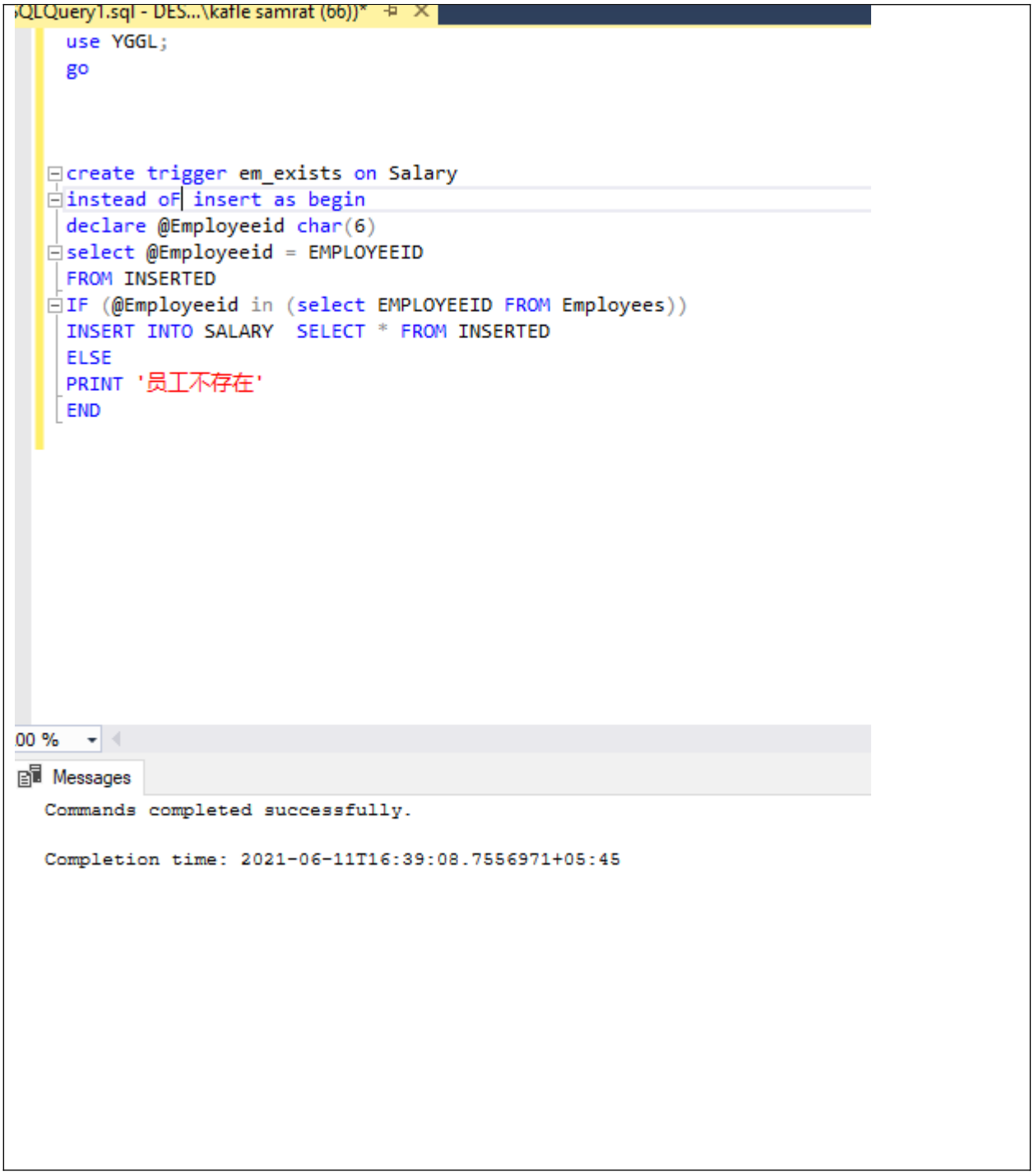
时，Outcome 值则增加 50。

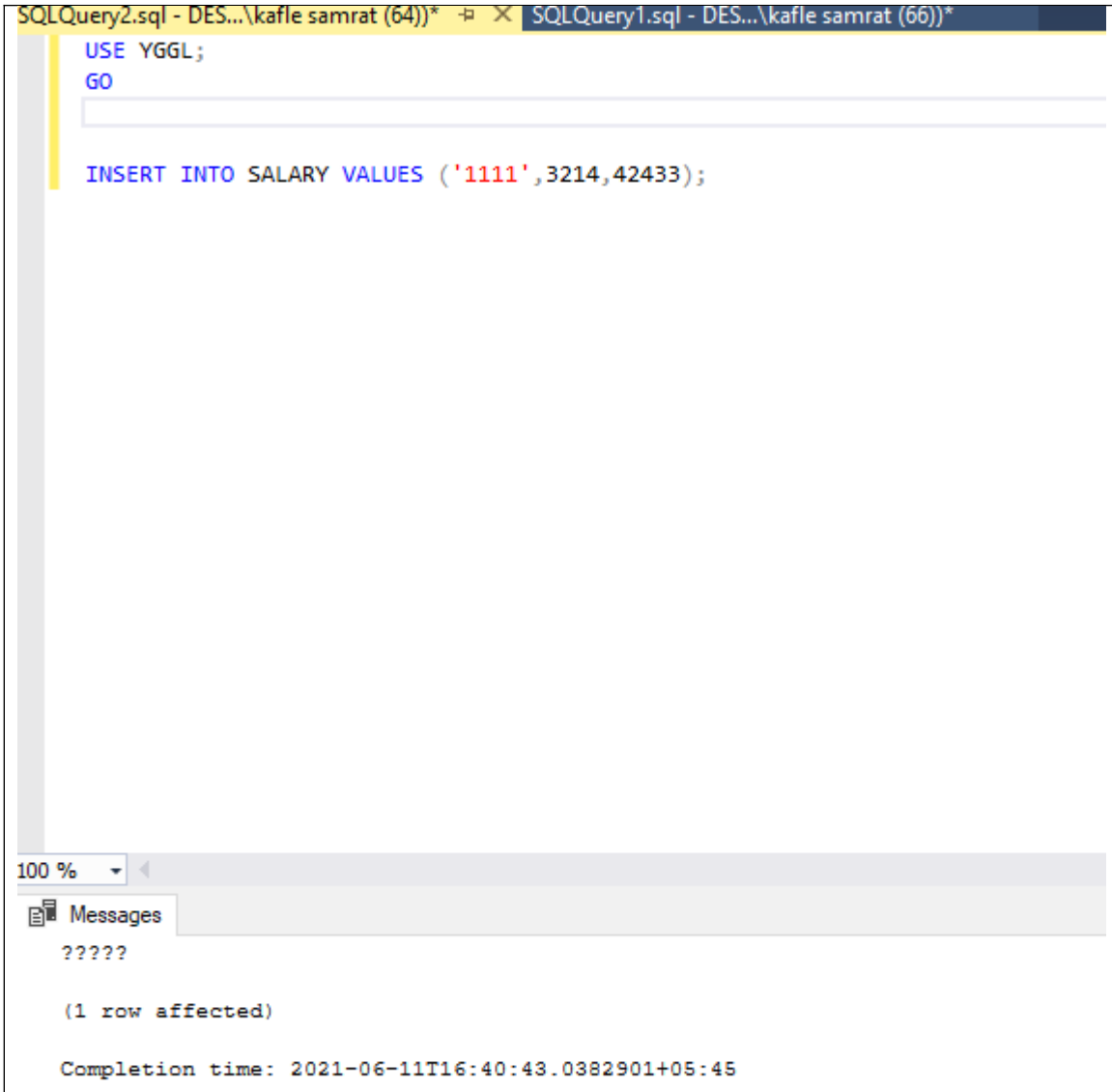
九. 实验内容和结果

（1）删除 Salary 表和 Employees 表之间建立的外键关系。



（2）创建 INSTEAD OF 触发器，当向 Salary 表中插入记录时，先检查 EmployeeID 列上的值在 Employees 中是否存在，如果存在则执行插入操作；如果不存在则提示“员工不存在”。向 Salary 表中插入数据来验证效果。


QLQuery1.sql - DES...\kafle samrat (66))*
use YGGL;
go
create trigger em_exists on Salary
instead oF insert as begin
declare @Employeeid char(6)
select @Employeeid = EMPLOYEEID
FROM INSERTED
IF (@Employeeid in (select EMPLOYEEID FROM Employees))
INSERT INTO SALARY SELECT * FROM INSERTED
ELSE
PRINT '员工不存在'
END
00 %
Messages
Commands completed successfully.
Completion time: 2021-06-11T16:39:08.7556971+05:45



（3）创建触发器，当修改表 Employees 时，若将 Employees 表中员工的工作时间增加 1 年，则将收入增加 500，若增加 2 年则增加 1000，依次增加。若工作时间减少则无变化。

```
USE YGGL;
```

```
GO



CREATE TRIGGER ADD_SALARY AFTER
UPDATE  ON Employees FOR EACH ROW
BEGIN

DECLARE YEARS INT;

SET YEARS=NEW.WorkYear-OlD.WorkYear;
    IF YEARS>0 THEN
UPDATE Salary SET INCOME=INCOME+500*YEARS
WHERE EMPLOYEEID=NEW. EMPLOYRRID;
END IF;
```

SQLQuery5.sql - DES...\kafle samrat (59))* SQLQuery4.sql - DES...\kafle samrat (61))*

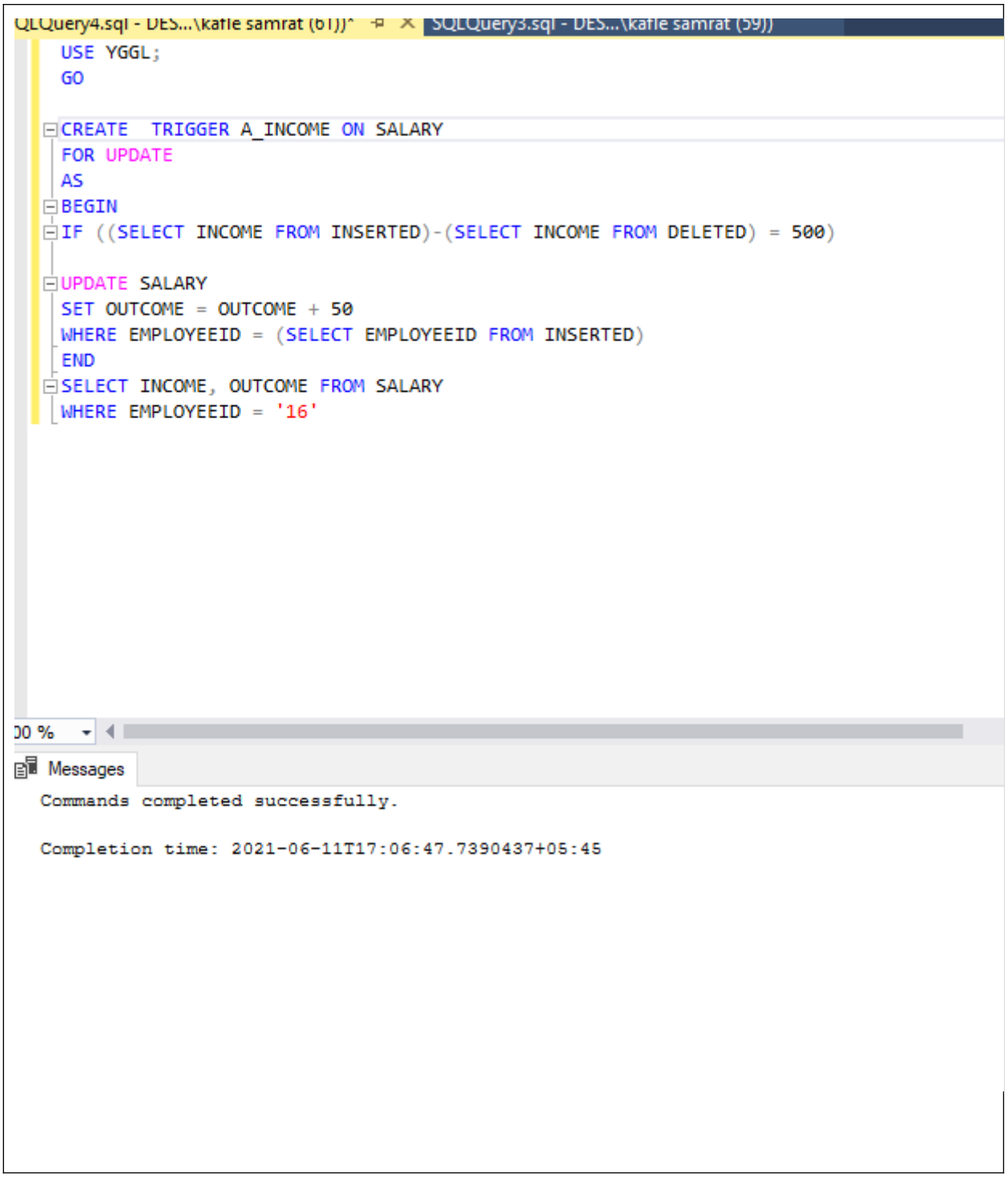
```
USE YGGL;
GO



CREATE TRIGGER ADD_SALARY AFTER
UPDATE  ON Employees FOR EACH ROW
BEGIN

DECLARE YEARS INT;

SET YEARS=NEW.WorkYear-OlD.WorkYear;
     IF YEARS>0 THEN
UPDATE Salary SET INCOME=INCOME+500*YEARS
WHERE EMPLOYEEID=NEW. EMPLOYRRID;
END IF;
```

100 %

Messages

```
Commands completed successfully.

Completion time: 2021-06-11T17:34:57.7882734+05:45
```

（4）创建 update 触发器，当 Salary 表中 Income 值增加 500 时，Outcome 值则增加 50。


QLQuery4.sql - DES...\kafle samrat (61))*
SQLQuery3.sql - DES...\kafle samrat (59))
USE YGGL;
GO
CREATE TRIGGER A_INCOME ON SALARY
FOR UPDATE
AS
BEGIN
IF ((SELECT INCOME FROM INSERTED)-(SELECT INCOME FROM DELETED) = 500)
UPDATE SALARY
SET OUTCOME = OUTCOME + 50
WHERE EMPLOYEEID = (SELECT EMPLOYEEID FROM INSERTED)
END
SELECT INCOME, OUTCOME FROM SALARY
WHERE EMPLOYEEID = '16'
00 %
Messages
Commands completed successfully.
Completion time: 2021-06-11T17:06:47.7390437+05:45

SQLQuery4.sql - DES...\kafle samrat (61))* | SQLQuery3.sql - DES...\kafle samrat (59))

```
USE YGGL;
GO


SELECT INCOME, OUTCOME FROM SALARY
WHERE EMPLOYEEID = '16'
```

100 %

Results | Messages

	INCOME	OUTCOME
1	8989	878

SQLQuery4.sql - DES...\kafle samrat (61))* | SQLQuery3.sql - DES...\kafle samrat (59))

```
USE YGGL;
GO



UPDATE SALARY
SET OUTCOME = OUTCOME + 500
WHERE EMPLOYEEID = '16'



SELECT INCOME, OUTCOME FROM SALARY
WHERE EMPLOYEEID = '16'
```

100 %

Results | Messages

	INCOME	OUTCOME
1	8989	1378

	INCOME	OUTCOME
1	8989	1378

实验总结及体会： Through this experiment, I learned the basic database SQL commands, Know how to create and use triggers. Understand the use of Inserted and Deleted logical tables.