华中科技大学

数据库系统原理实践报告

综合设计题目：

姓　　名：　龙际全

学　　院：　计算机科学与技术学院

专　　业：　计算机科学与技术

班　　级：　CS1603

学　　号：　U201614577

指 导 教 师：　瞿彬彬

分数	
教师签名	

2019　年 05 月 10 日

任务书

1 软件功能学习部分（必做题）

完成下列 1~2 题，并在实践报告中叙述过程，可适当辅以插图（控制在 A4 三页篇幅以内）

1）练习 sqlserver 的两种完全备份方式：数据和日志文件的脱机备份、系统的备份功能。

2）练习在新增的数据库上增加用户并配置权限的操作，通过用创建的用户登录数据库并且执行未经授权的 SQL 语句验证自己的权限配置是否成功。

2 Sql 练习部分（必做题）

2.1 建表

1）假设某微博平台的数据库中有下列关系，请在 DBMS 中创建这些关系，包括主码和外码的说明，并写出指定关系的建表 SQL 语句：

用户【用户 ID，姓名，性别，出生年份，所在城市】记录所有注册用户的基本信息，英文表名和字段名如下：

USER(UID 整型,NAME 字符串,SEX 一位汉字,BYEAR 整型,CITY 字符串)；

分类【分类 ID，分类名称】记录微博平台中所有可能涉及的微博的类型，例如文学、艺术、军事等，英文表名和字段名如下：

LABEL(LID 整型，LNAME 字符串)；

博文【博文 ID，标题，用户 ID，年，月，日，正文】记录每一篇微博的基本信息，英文表名和字段名如下：

MBLOG(BID 整型，TITLE 字符串，UID 整型，PYEAR 整型，PMONTH 整型，PDAY 整型，CONT 字符串)，

写出该关系的建表 SQL 语句；

博文标注【博文 ID，分类 ID】记录每一篇微博的作者给该微博贴上的分类标签，一篇微博可以涉及不止一种分类，英文表名和字段名如下：

B_L(BID 整型，LID 整型)；

关注【用户 ID，被关注用户 ID】记录每位用户关注的其他用户，每位用户可关注多人，英文表名和字段名如下：

FOLLOW(UID 整型，UIDFLED 整型)；

好友【用户 ID， 好友 ID】记录每位用户的好友（可多个），英文表名和字段名如下：

FRIENDS(UID 整型，FUID 整型);

订阅【用户 ID, 订阅分类 ID】记录用户订阅的每一种分类，英文表名和字段名如下：
SUB(UID 整型，LID 整型);

点赞【用户 ID，博文 ID】记录用户点赞的每一篇微博，英文表名和字段名如下：
THUMB(UID 整型，BID 整型)，
写出该关系的建表 SQL 语句；

头条【年，月，日，博文 ID，顺序号】记录每一天的热度排名前十的博文 ID 号以及该博文在热度前十名中的排名，英文表名和字段名如下：
TOPDAY(TYEAR 整型，TMONTH 整型，TDAY 整型，BID 整型，TNO 整型)。

2）观察性实验

用户在订阅分类时是否一定要参考被参照关系的主码，并在实验报告中简述过程和结果。

3）数据准备

依据后续实验的要求，向上述表格中录入适当数量的实验数据，从而对相关的实验任务能够起到验证的作用。

2.2 数据更新

1）分别用一条 sql 语句完成对博文表基本的增、删、改的操作；

2）批处理操作

将关注 3 号用户的用户信息插入到一个自定义的新表 FANS_3 中。

3）数据导入导出

通过查阅 DBMS 资料学习数据导入导出功能，并将任务 2.1 所建表格的数据导出到操作系统文件，然后再将这些文件的数据导入到相应空表。
在后续的上机实验环节，通过导入导出或者备份机制实现前次上机环节的数据恢复。

4）观察性实验

建立一个关系，但是不设置主码，然后向该关系中插入重复元组，然后观察在图形化交互界面中对已有数据进行删除和修改时所发生的现象。

5）触发器实验

编写一个触发器，用于实现对点赞表的完整性控制规则：当插入或者更新点赞关系时，如果博文作者就是点赞者本人，则拒绝执行。

2.3 查询

请分别用一条 SQL 语句完成下列各个小题的需求：

1）查询“张三”用户关注的所有用户的 ID 号、姓名、性别、出生年份，所在城市，并且按照出生年份的降序排列，同一个年份的则按照用户 ID 号升序排列。

2）查找没有被任何人点赞的博文 ID、标题以及发表者姓名，并将结果按照标题字符顺序排列。

3）查找 2000 年以后出生的武汉市用户发表的进入过头条的博文 ID；

4）查找订阅了所有分类的用户 ID；

5）查找出生年份小于 1970 年或者大于 2010 年的用户 ID、出生年份、所在城市，要求 where 子句中只能有一个条件表达式；

6）统计每个城市的用户数；

7）统计每个城市的每个出生年份的用户数，并将结果按照城市的升序排列，同一个城市按照出生用户数的降序排列其相应的年份；

8）查找被点赞数超过 10 的博文 ID 号；

9）查找被 2000 年后出生的用户点赞数超过 10 的博文 ID 号；

10）查找被 2000 年后出生的用户点赞数超过 10 的每篇博文的进入头条的次数；

11）查找订阅了文学、艺术、哲学、音乐中至少一种分类的用户 ID，要求不能使用嵌套查询，且 where 子句中最多只能包含两个条件；

12）查找标题中包含了“最多地铁站”和“_华中科技大学”两个词的博文基本信息；

13）查找所有相互关注的用户对的两个 ID 号，要求不能使用嵌套查询；

14）查找好友圈包含了 5 号用户好友圈的用户 ID；

15）查找 2019 年 4 月 20 日每一篇头条博文的 ID 号、标题以及该博文的每一个分类 ID，要求即使该博文没有任何分类 ID 也要输出其 ID 号、标题；

16）查找至少有 3 名共同好友的所有用户对的两个 ID 号。

17）创建视图：查阅 DBMS 内部函数，创建一个显示当日热度排名前十的微博信息的视图，其中的属性包括：博文 ID、博文标题、发表者 ID、发表者姓名、被点赞数。

2.4 了解系统的查询性能分析功能（选做，各班指导教师可适当调整）

选择上述 2.3 任务中某些较为复杂的 SQL 语句，查看其执行之前系统给出的分析计划和实际的执行计划，记录观察的结果，并对其进行简单的分析。

2.5 DBMS 函数及存储过程和事务（选做，各班指导教师可适当调整）

1）编写一个依据用户 ID 号计算其发表的博文进入头条的累计天数的 DBMS 自定义函数，并利用其查询 2000 年后出生的上述头条累计天数达到 100 天的所有用户 ID。

2）建立关系“点赞排行榜【博文 ID，当天点赞人数】”，里面存储系统当天点赞数前十名的博文 ID 及其点赞人数，尝试编写一个 DBMS 的存储过程，通过该存储过程更新该表。

3）尝试在 DBMS 的交互式界面中验证事务机制的执行效果。

目录

1　课程任务概述

实验主要分成三部分：软件功能学习部分、SQL 联系部分和综合实践部分。

1） 软件功能学习部分：实验指定使用 SQL server 数据库，该部分主要是熟悉该数据库的操作，包括创建用户，设置权限，备份数据和恢复数据等。

2） SQL 练习部分：用于熟悉 sql 语句的使用，所有的查询都基于与电影相关的关系，共 17 条查询语句，涵盖了大部分 sql 语法。

3） 综合实践部分：提供若干个题目供选择，如工资管理系统、航班管理系统等，每个系统都需要根据实际情况设计并创建若干个表，选用一种编程语言实现一个客户端/网页，要求数据库关系设计合理，客户端要有良好的人机交互界面。

2　软件功能学习部分

2.1　任务要求

1）练习 sqlserver 的两种完全备份方式：数据和日志文件的脱机备份、系统的备份功能。

2）练习在新增的数据库上增加用户并配置权限的操作。

2.2　完成过程

2.2.1　脱机备份

1)　脱机指与服务器断开连接，脱机操作为“**右键-任务-脱机**”，如下图 2-1：

图 2-1 数据库脱机

2)　数据文件和日志文件可通过数据库属性页“**属性-文件**”查看，如下图 2-2，拷贝到想要拷贝的地方即实现了脱机备份。

图 2-2 查看数据库文件位置

3)　通过“**右键-附加**”，选择刚刚拷贝的数据库数据文件即可实现脱机恢复，如下图 2-3；

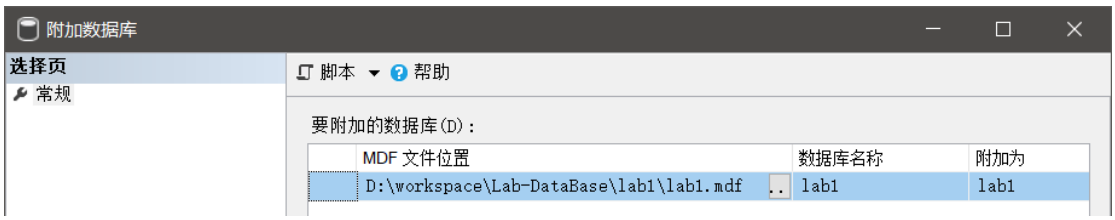

图 2-3 脱机恢复

2.2.2　系统备份

1)　选中数据库“**lab1**”，通过“**右键-备份**”打开备份选项卡，如下图 2-4：

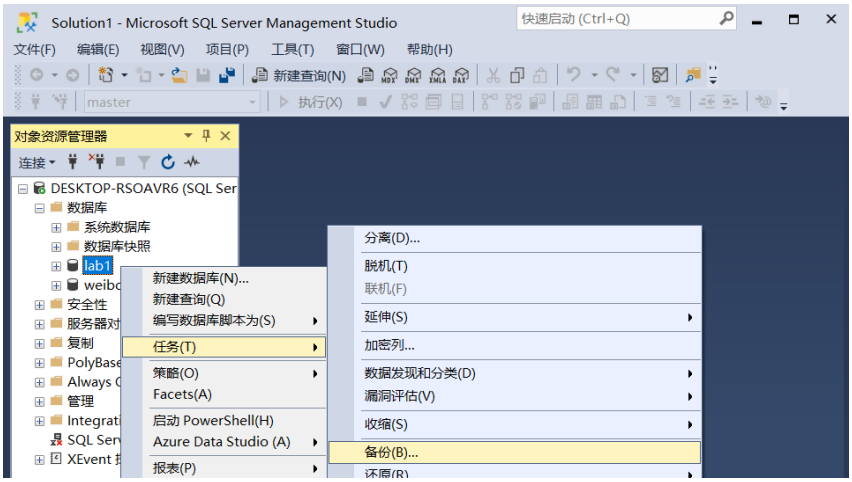

图 2-4 数据库系统备份

2) 备份到如下文件夹，如下图 2-5：

图 2-5 数据库备份位置

3) 选中数据库“**lab1**”，通过“**右键-任务-还原-数据库**”即可打开还原选项卡，选择刚才生成的.bak 文件即可还原。

2.2.3 新增用户

1) 选中“安全性”，“右键-新建-登录名”即可通过 SQL Server 身份验证创建一个新的用户，如下图 2-6；

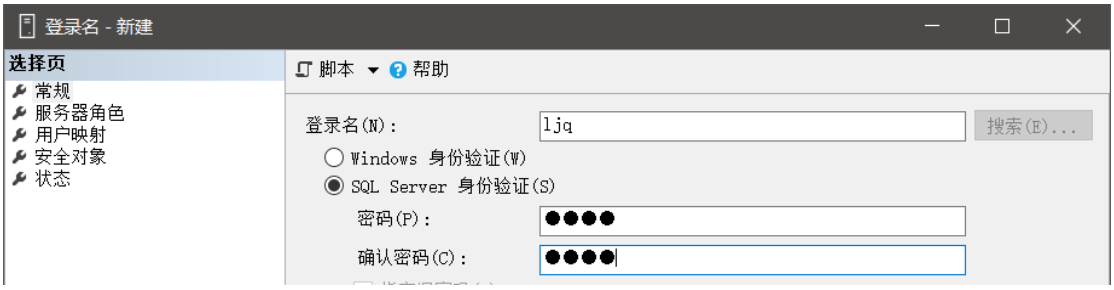

图 2-6 新增用户

2) 给用户“ljq”授权如下图 2-7；

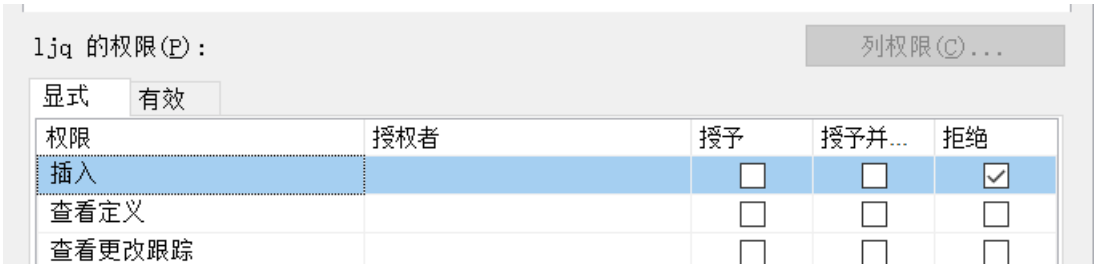

图 2-7 用户授权

3) 登录数据库，如下图 2-8；

图 2-8 登录数据库

2.3 任务总结

1) 脱机备份

a) 现象：数据库 lab1 没有脱机时无法将数据库的数据文件 lab1.mdf 和日志文件 lab1_log.mdf 拷贝到其他位置；

b) 原因：数据库未进行脱机操作时，lab1.mdf 和 lab1_log.mdf 被 SSMS 通过操作系统进行保护，无法进行拷贝；

c) 解决方法：备份 lab1.mdf 和 lab1_log.mdf 时先对数据库进行脱机操作即可。

2) 用户登录

a) 现象：采用 SQL Server 身份验证登录时出现运行时错误，如图 2-9 下所示：

图 2-9 用户登录 Error233

b) 原因：未开启 SQL Server 的 TCP/IP 协议和 Named Pipes 协议；

c) 解决方法：可通过 Sql Server Configuration Manager 启用，如下图 2-10；

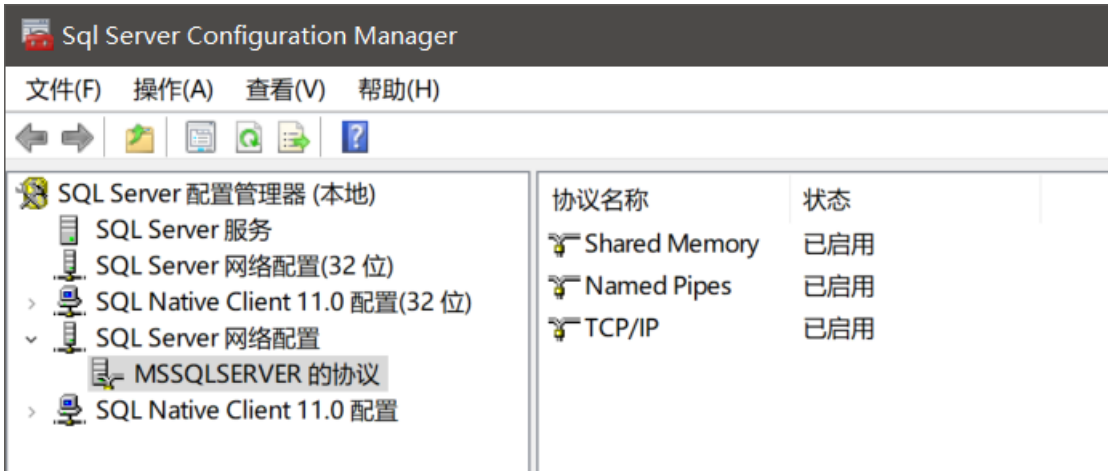

图 2-10 开启 MSSQLSERVER 协议

3) 身份验证

a) 现象：修改 SQL Server 登录协议后仍然无法连接到数据库，错误码提示 18456；

b) 原因：服务器配置为仅使用集成身份验证，SQL Server 运行日志如下图 2-11：

```
Login failed for user 'l'. 原因: 尝试使用 SQL 身份验证登录失败。服务器配置为仅使用集成身份验证。 [客户端: <local machine>]
错误: 18456, 严重性: 14, 状态: 58。
Login failed for user 'l'. 原因: 尝试使用 SQL 身份验证登录失败。服务器配置为仅使用集成身份验证。 [客户端: <local machine>]
错误: 18456, 严重性: 14, 状态: 58。
Using 'xpstar.dll' version '2017.140.1000' to execute extended stored procedure 'xp_instance_regread'. This is an inform
Attempting to load library 'xpstar.dll' into memory. This is an informational message only. No user action is required.
Using 'dbghelp.dll' version '4.0.5'
Login failed for user 'l'. 原因: 尝试使用 SQL 身份验证登录失败。服务器配置为仅使用集成身份验证。 [客户端: <local machine>]
错误: 18456, 严重性: 14, 状态: 58。
```

图 2-11 用户登录错误日志

c) 解决方案：开启 SQL Server 用户验证，右键服务器，选择“**属性-安全性**”，选中“**SQL Server 和 Windows 身份验证模式**”即可，如下图 2-12：

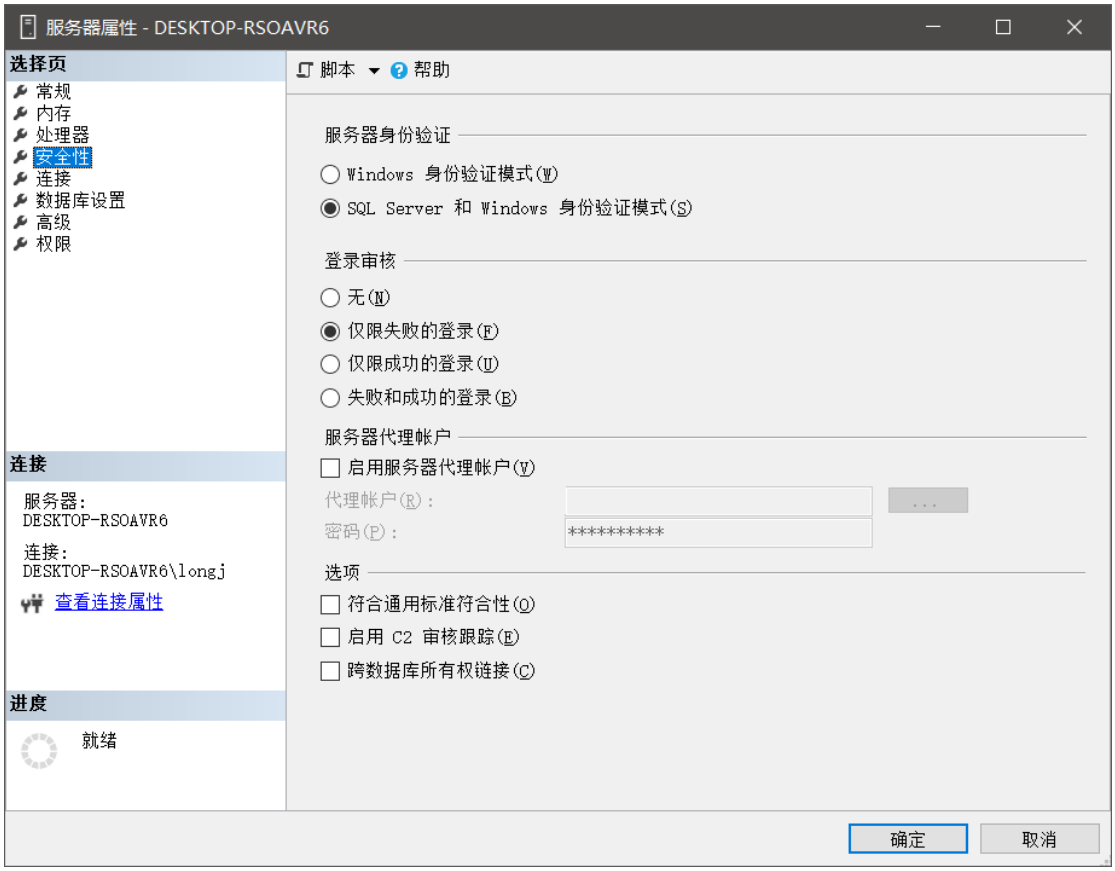

图 2-12 开启 SQL Server 身份验证

3 SQL 练习部分

3.1 任务要求

1. 建表：在 DBMS 中创建指定的关系表，包括主码和外码的说明，写出指定关系的建表 SQL 语句；
2. 数据更新：使用 SQL 语句对博文表增删改查、数据导入导出、编写触发器；
3. 查询：使用 SQL 语句完成相应的小题。

3.2 完成过程

3.2.1 建表

1) 创建并使用数据库[weibo]：

```
create database weibo;
go
use weibo;
go
```

2) 其他关系表建表 SQL 语句类似，如下：

```
drop table if exists users;
create table users(
    uids int primary key,
    names char(30) not null,
    sex char(2) not null,
    byear int not null,
    city char(30) not null
);

drop table if exists label;
create table label(
    lid int primary key,
    lname char(30) not null
);

drop table if exists mblog;
create table mblog(
    bid int primary key,
    title char(30) not null,
    uids int not null,
    pyear int not null,
    pmonth int not null,
    pday int not null,
```

```
    cont text not null,
    foreign key (uids) references users(uids)
);

drop table if exists b_l;
create table b_l(
    bid int not null,
    lid int not null,
    foreign key (bid) references mblog(bid),
    foreign key (lid) references label(lid)
);

drop table if exists follow;
create table follow(
    uids int not null,
    uidfled int not null,
    foreign key (uids) references users(uids),
    foreign key (uidfled) references users(uids),
    primary key (uids, uidfled)
);

drop table if exists friends;
create table friends(
    uids int,
    fuid int not null,
    foreign key (uids) references users(uids),
    foreign key (fuid) references users(fuid),
    primary key (uids, fuid)
);

drop table if exists sub;
create table sub(
    uids int not null,
    lid int not null,
    foreign key (uids) references users(uids),
    foreign key (lid) references label(lid),
    primary key (uids, lid)
);

drop table if exists thumb;
```

```
create table thumb(
    uids int not null,
    bid int not null,
    foreign key (uids) references users(uids),
    foreign key (bid) references mblog(bid),
    primary key (uids, bid)
);

drop table if exists topday;
create table topday(
    tyear int not null,
    tmonth int not null,
    tday int not null,
    bid int not null,
    tno int not null,
    foreign key (bid) references mblog(bid)
);
```

3) 观察性实验

a) 用户在订阅分类时一定要参考被参照关系的主码，如下图 3-1 byear 并不是 users 表的主码，SSMS 红线标错；

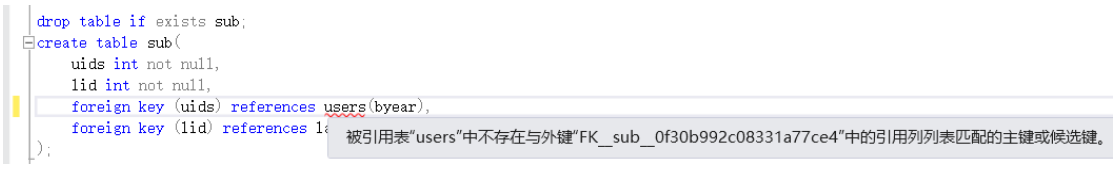

图 3-1 参照非主码

b) 缺少参照，SSMS 仍然红线标错如下图 3-2。

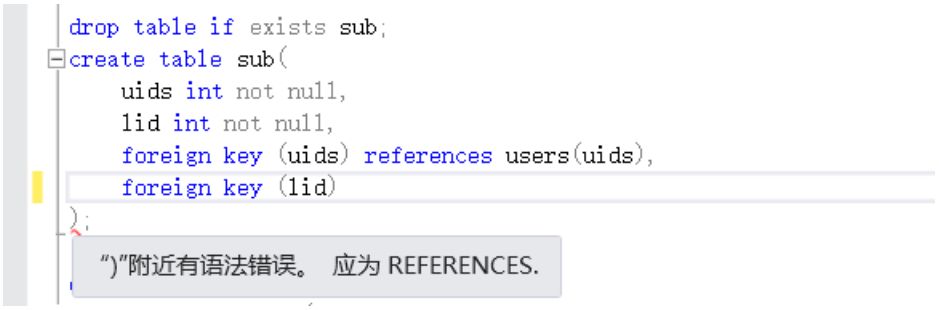

图 3-2 缺少参照

4) 数据准备（只展示了三组，皆为在 SSMS 中操作，其他类似）

a) Users

DESKTOP-RSOAVR6...ibo - dbo.users

uids	names	sex	byear	city
1	赵一 ...	男	1967	武汉 ...
2	钱二 ...	女	1968	上海 ...
3	张三 ...	男	1999	南京 ...
4	李四 ...	女	2000	北京 ...
5	王五 ...	男	2001	武汉 ...
6	孙六 ...	女	2001	武汉 ...
7	周七 ...	男	2002	武汉 ...
8	吴八 ...	男	2002	杭州 ...
9	郑九 ...	男	2003	南京 ...
10	冯十 ...	女	2004	上海 ...
11	陈十一 ...	男	2011	武汉 ...
12	诸十二 ...	女	2012	北京 ...
13	卫十三 ...	女	2011	上海 ...
14	蒋十四 ...	女	2011	武汉 ...
NULL	*NULL*	*NULL*	*NULL*	*NULL*

图 3-3 user 数据

b) Label

DESKTOP-RSOAVR6...ibo - dbo.label

lid	lname
1	文学 ...
2	艺术 ...
3	军事 ...
4	历史 ...
5	地理 ...
6	自然科学 ...
7	工程技术 ...
8	经济 ...
9	教育 ...
10	哲学 ...
11	音乐 ...
NULL	*NULL*

图 3-4 label 数据

c) Mblog

DESKTOP-RSOAVR...ibo - dbo.mblog

bid	title	uids	pyear	pmonth	pday	cont
1	微博1 ...	1	2019	4	18	"最多地铁站"和"_华中科技大学"两个词
2	微博2 ...	2	2019	4	18	华中科技大学是国家教育部直属重点综合性大学
3	微博3 ...	3	2019	4	18	"最多地铁站"和"_华中科技大学"两个词
4	微博4 ...	4	2019	4	19	学校校园占地7000余亩
5	微博5 ...	5	2019	4	19	校学科齐全、结构合理
6	微博6 ...	6	2019	4	19	学校实施"人才兴校"战略
7	微博7 ...	7	2019	4	19	"学生"的教育思想
8	微博8 ...	8	2019	4	19	按照"应用领先"的科技发展方略
9	微博9 ...	9	2019	4	19	学校坚持"贡献即发展"的办学思路
10	微博10 ...	10	2019	4	19	学校坚持开放式办学理念
11	微博11 ...	11	2019	4	19	《巢》："东欧文学的良心"流亡美国的坎坷人生
12	微博12 ...	12	2019	4	19	派特·巴克代表作《重生三部曲》
13	微博13 ...	13	2019	4	19	文学巨匠翻译的经典名著
14	微博14 ...	14	2019	4	19	中国某新型核潜艇接受检阅
15	微博15 ...	1	2019	4	20	中国核电装机容量升至全球第三
16	微博16 ...	3	2019	4	20	海上阅兵活动开始 习近平检阅海上编队
17	微博17 ...	5	2019	4	20	斯里兰卡警方：爆炸袭击案死亡人数升至321人
18	微博18 ...	7	2019	4	20	辽宁舰升级改造 崭新亮相
19	微博19 ...	9	2019	4	20	若是干掉慈禧荣禄，当年康党可赢
20	微博20 ...	11	2019	4	20	三省制：唐代中书门下尚书职权之争

图 3-5 mblog 数据

3.2.2 数据更新

1) 博文表增删改

a) 增

Insert into mblog values

('44', '微博 44', '1', '2019', '5', '9', '水花回暖勇士险胜火箭取赛点')

查看 mblog 表，如下图 3-6：

	43	微博43 ...	2	2019	4	21	西班牙画家...
▶	44	微博44 ...	1	2019	5	9	险胜火箭取赛点

图 3-6 博文表增

b) 改

Update mblog set cont = '我改了一条记录啊' where bid = '44'

查看 mblo 表，如下图 3-7：

	42	微博42 ...	1	2019	4	21	意大利画家达芬奇...
	43	微博43 ...	2	2019	4	21	西班牙画家巴勃罗...
▶	44	微博44 ...	1	2019	5	9	我改了一条记录啊

图 3-7 博文表改

c) 删

Delete from mblog where bid = '44'

查看 mblo 表，如下图 3-8：

	41	微博41	...	1	2019	4	21	西晋王朝短暂的大...
	42	微博42	...	1	2019	4	21	意大利画家达芬奇...
▶	43	微博43	...	2	2019	4	21	西班牙画家巴勃罗...
*	NULL	NULL		NULL	NULL	NULL	NULL	NULL

图 3-8 博文表删

2) 批处理操作

a) 代码如下：

```
use weibo;
go
drop table if exists fans_3;
create table fans_3(
    uids int primary key,
    names char(30) not null,
    sex char(2) not null,
    byear int not null,
    city char(30) not null
);
insert into fans_3
(uids, names, sex, byear, city)
select users.uids, users.names, users.sex, users.byear, users.city
from users, follow
where users.uids = follow.uids and follow.uidfled = 3
```

b) 操作后的 fans_3 表如下图 3-9：

	uids	names	sex	byear	city
1	4	李四	女	2000	北京
2	6	孙六	女	2001	武汉
3	8	吴八	男	2002	杭州
4	10	冯十	女	2004	上海
5	11	陈十一	男	2011	武汉

图 3-9 批处理后的 fans_3 表

3) 数据导入导出

除第一部分“脱机备份”以及“系统备份”外，还可以选择数据库“weibo”，通过“右键-任务-生成脚本”生成数据库的架构（DDL）和数据（DML），如下图 3-10：

图 3-10 导出数据库脚本

4) 观察性实验

a) 代码如下：

```
use weibo;
go
create table observe(
    obj1 int,
    obj2 text
);
insert into observe values
(1, 'hahaha'),
(1, 'hahaha'),
(1, 'hahaha')
```

b) 更新后的表如下图 3-11：

	obj1	obj2
1	1	hahaha
2	1	hahaha
3	1	hahaha

图 3-11 没有主键的表

c) 对重复元组进行修改或者删除报错，如下图 3-12：

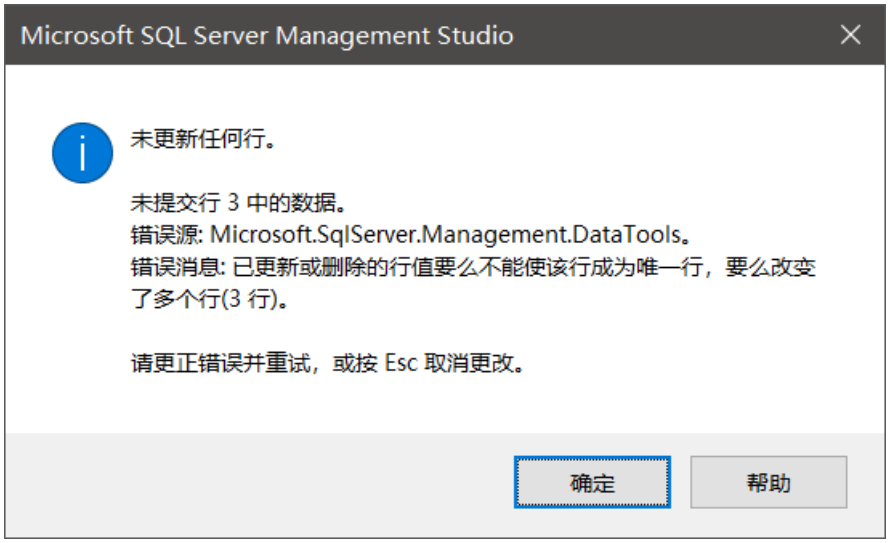

图 3-12 重复元组无法删除

5) 触发器实验

a) 代码如下：

```
use weibo;
go
drop trigger if exists refuse_self_thumb
create trigger refuse_self_thumb on thumb
instead of insert, update
as
begin
    if exists (
        select *
        from inserted, mblog
        where inserted.bid = mblog.bid
        and inserted.uids = mblog.uids
    )
    begin
        print 'error, 不能给自己点赞'
    end
    else
    begin
        insert into thumb
        (uids, bid)
        select uids, bid
        from inserted
    end
end
go
```

b) 测试代码如下：

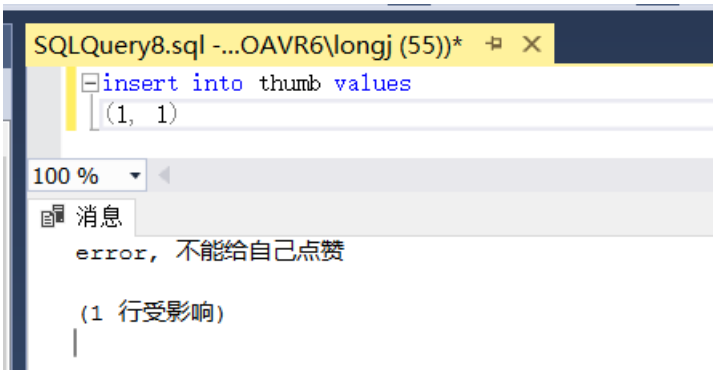

图 3-13 插入后触发器触发输出提示

3.2.3 查询

1) 查询“张三”用户关注的所有用户的 ID 号、姓名、性别、出生年份，所在城市，并且按照出生年份的降序排列，同一个年份的则按照用户 ID 号升序排列。

a) 代码：

```
select users.uids, users.names, users.sex, users.byear, users.city
from follow, users
where follow.uidfled = users.uids and follow.uids in(
    select users.uids
    from users
    where users.names = '张三'
    )
order by users.byear desc, users.uids asc
```

b) 查询结果：

	uids	names	sex	byear	city
1	11	陈十一	男	2011	武汉
2	4	李四	女	2000	北京
3	2	钱二	女	1968	上海
4	1	赵一	男	1967	武汉

图 3-14

2) 查找没有被任何人点赞的博文 ID、标题以及发表者姓名，并将结果按照标题字符顺序排列。

a) 代码：

```
select mblog.bid, mblog.title, users.names
from mblog, users
where users.uids = mblog.uids and    mblog.bid not in
    (
        select bid from thumb
    )
```

order by mblog.title asc

b) 查询结果：

	bid	title	names
1	1	微博1	赵一
2	2	微博2	钱二
3	3	微博3	张三
4	4	微博4	李四
5	5	微博5	王五
6	6	微博6	孙六

图 3-15

3) 查找 2000 年以后出生的武汉市用户发表的进入过头条的博文 ID；

a) 代码：

```
select mblog.bid
from mblog, users, topday
where   mblog.uids = users.uids and
        topday.bid = mblog.bid and
        users.byear > 2000 and
        users.city = '武汉'
```

b) 查询结果：

	bid
1	5
2	6
3	7
4	11
5	14
6	17
7	18
8	20
9	24
10	28
11	29
12	30
13	31
14	36
15	37

图 3-16

4) 查找订阅了所有分类的用户 ID；

a) 代码：

```
select sub.uids
from sub
group by uids
having count(*) = any(select count(*) from label);
```

b) 查询结果：

图 3-17

5) 查找出生年份小于 1970 年或者大于 2010 年的用户 ID、出生年份、所在城市，要求 where 子句中只能有一个条件表达式；

a) 代码：

```
select uids, byear, city
from users
where byear not between '1970' and '2010'
```

b) 查询结果：

	uids	byear	city
1	1	1967	武汉
2	2	1968	上海
3	11	2011	武汉
4	12	2012	北京
5	13	2011	上海
6	14	2011	武汉

图 3-18

6) 统计每个城市的用户数；

a) 代码：

```
select city, count(uids) '用户数'
from users
group by city
```

b) 查询结果：

	city	用户数
1	北京	2
2	杭州	1
3	南京	2
4	上海	3
5	武汉	6

图 3-19

7) 统计每个城市的每个出生年份的用户数，并将结果按照城市的升序排列，同一个城市按照出生用户数的降序排列其相应的年份；

a) 代码：

```
select city, byear, count(uids)
from users
group by    byear,city
order by city, byear desc;
```

b) 查询结果：

	city	byear	用户数
1	北京	2012	1
2	北京	2000	1
3	杭州	2002	1
4	南京	2003	1
5	南京	1999	1
6	上海	2011	1
7	上海	2004	1
8	上海	1968	1
9	武汉	2011	2
10	武汉	2002	1
11	武汉	2001	2
12	武汉	1967	1

图 3-20

8) 查找被点赞数超过 10 的博文 ID 号；

a) 代码：

```
select bid
from thumb
group by bid
having count(bid) > 1 – 数据里只有超过 1 的
```

b) 查询结果：

	bid
1	9
2	10
3	11
4	12
5	13

图 3-21

9) 查找被 2000 年后出生的用户点赞数超过 10 的博文 ID 号；

a) 代码：

```
select bid
from thumb, users
where   thumb.uids = users.uids and
        users.byear >=2000
group by bid
having count(bid) > 1
```

b) 查询结果：

	bid
1	10
2	11
3	12
4	13

图 3-22

10) 查找被 2000 年后出生的用户点赞数超过 10 的每篇博文的进入头条的次数；

a) 代码：

```
select topday.bid, count(*) as '次数'
from topday
where topday.bid in (
    select bid
    from thumb, users
    where   thumb.uids = users.uids and
            users.byear >= 2000
    group by bid
    having count(bid) > 1
)
group by topday.bid
```

b) 查询结果：

	bid	次数
1	10	1
2	11	1
3	12	1
4	13	1

图 3-23

11) 查找订阅了文学、艺术、哲学、音乐中至少一种分类的用户 ID，要求不能使用嵌套查询，且 where 子句中最多只能包含两个条件；

a) 代码：

```
select distinct uids
from sub, label
where   sub.lid = label.lid and
        label.lname in ('文学','艺术','哲学','音乐')
```

b) 查询结果：

	uids
1	1
2	2
3	4
4	5
5	6
6	7
7	8
8	9
9	10
10	11
11	12
12	13

图 3-24

12) 查找标题中包含了“最多地铁站”和“_华中科技大学”两个词的博文基本信息；

a) 代码：

```
select bid, title, uids, pyear, pmonth, pday
from mblog
where   cont like '%最多地铁站%' or
        cont like '%\_华中科技大学%'
```

b) 查询结果：

	bid	title	uids	pyear	pmonth	pday
1	1	微博1	1	2019	4	18
2	3	微博3	3	2019	4	18

图 3-25

13) 查找所有相互关注的用户对的两个 ID 号，要求不能使用嵌套查询；

a) 代码：

```
select one.uids, two.uids
from follow one, follow two
where   one.uids = two.uidfled and
        one.uidfled = two.uids and
        one.uids > two.uids
```

b) 查询结果：

	uids	uids
1	2	1
2	4	3
3	11	3

图 3-26

14) 查找好友圈包含了 5 号用户好友圈的用户 ID；

a) 代码：

```
select uids
from users
where not exists(
    select *
    from friends one
    where    one.uids='5'and
             not exists(
                 select *
                 from friends two
                 where    two.uids = users.uids and
                          one.fuid = two.fuid
             )
    )
```

b) 查询结果：

	uids
1	5

图 3-27

15) 查找 2019 年 4 月 20 日每一篇头条博文的 ID 号、标题以及该博文的每一个分类 ID，要求即使该博文没有任何分类 ID 也要输出其 ID 号、标题；

a) 代码：

```
select mblog.bid, mblog.title, b_l.lid
from topday, mblog, b_l
where   topday.bid = mblog.bid and
        mblog.bid = b_l.bid and
        pyear = '2019' and
        pmonth = '4' and
        pday = '20'
```

b) 查询结果：

	bid	title	lid
1	15	微博15	3
2	16	微博16	3
3	17	微博17	3
4	18	微博18	3
5	19	微博19	4
6	20	微博20	4
7	21	微博21	6
8	22	微博22	6
9	23	微博23	6

图 3-28

16) 查找至少有 3 名共同好友的所有用户对的两个 ID 号。

a) 代码：

```
select one.uids, two.uids
from friends one, friends two
where   one.uids <> two.uids and
        one.fuid = two.fuid and
        one.uids < two.uids
group by one.uids, two.uids
having count(*) > 1
```

b) 查询结果：

	uids	uids
1	4	12
2	5	13

图 3-29

17) 创建视图：查阅 DBMS 内部函数，创建一个显示当日热度排名前十的微博信息的视图，其中的属性包括：博文 ID、博文标题、发表者 ID、发表者姓名、被点赞数。

a) 代码：

```
drop view if exists topten;
go
create view topten as
select distinct topday.bid, mblog.title, users.uids, users.names, count(*)
as liking
from topday, mblog, users, thumb
where topday.tyear = (select DATENAME(YYYY,GETDATE()))
        and topday.tmonth = (select DATENAME(MM,GETDATE()))
        and topday.tday = (select DATENAME(DAY,GETDATE()))
        and topday.bid = mblog.bid
        and mblog.uids = users.uids
        and thumb.bid = mblog.bid
group by topday.bid, mblog.title, users.uids, users.names
go
```

b) 结果如下图 3-30：

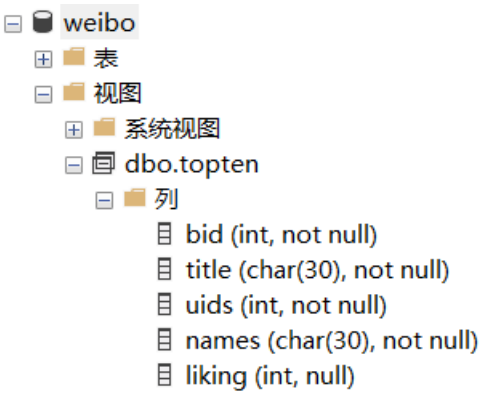

图 3-30 视图界面

c) 视图前 1000 行如下图 3-31：

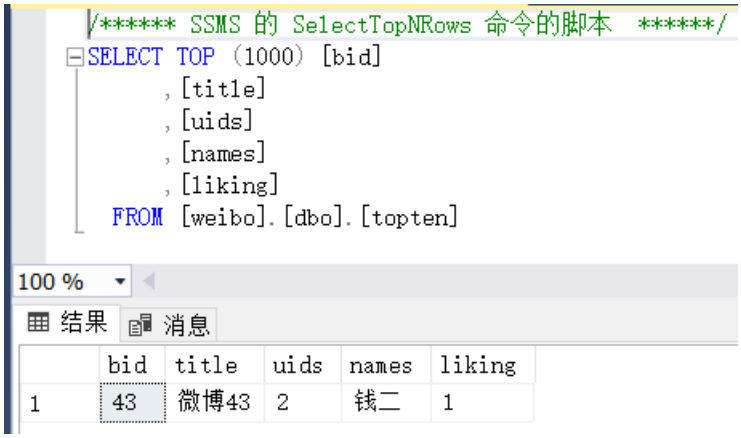

图 3-31 视图结果

3.3 任务总结

1. 我学会了如何在 SQL Server 中创建数据库和数据表；
2. 我学会了如何创建触发器来保证数据表之间的约束完整性；
3. 基本掌握了在数据库中检索数据的方法。