

**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： | 翟彬彬 |

|  |  |
| --- | --- |
| 分数 |  |
| 教师签名 |  |

20 年 月 日

**任务书**

。。。。。。

。

。

。

。

。

**目 录**

[1 课程任务概述 1](#_Toc8399293)

[2 软件功能学习部分 3](#_Toc8399294)

[2.1 任务要求 3](#_Toc8399295)

[2.2 完成过程 3](#_Toc8399296)

[2.3任务总结 5](#_Toc8399297)

[3 Sql练习部分 8](#_Toc8399298)

[3.1 任务要求 8](#_Toc8399299)

[3.2 完成过程 8](#_Toc8399300)

[3.3 任务总结 23](#_Toc8399301)

[4 综合实践任务 24](#_Toc8399302)

[4.1系统设计目标 24](#_Toc8399303)

[4.2 需求分析 24](#_Toc8399304)

[4.3 总体设计 24](#_Toc8399305)

[4.4 数据库设计 25](#_Toc8399306)

[4.5 详细设计与实现 25](#_Toc8399307)

[4.6 系统测试 25](#_Toc8399308)

[4.7 系统设计与实现总结 25](#_Toc8399309)

[5 课程总结 26](#_Toc8399310)

[附录 27](#_Toc8399311)

# 1 课程任务概述

**一级标题，单独起页（下同）**

居中 小三号 宋体加粗

简要陈述介绍本实践课程的各项任务要求。

**关于正文排版要求：**

正文统一采用小四号宋体/Times New Roman和1.25倍行距，段前、段后均0磅间隔。

正文应论述清楚，文字简练通顺，插图简明，书写整洁。文中图、表按制图要求绘制。

**关于正文章节标题的排版要求：**

报告章标题称为一级标题，章内小节标题依次分为二级标题、三级标题等。一级标题的编号用数字1，2，…编制；二级标题的编号用1.1，1.2，…编制；三级标题的编号用1.1.1，1.2.1，… 编制；四级及以后各级标题可依此类推。建议标题不超过3级（如1.1.1），超出部分可根据需要使用(1)，①，A，a)，…等形式描述。

标题编号与标题文字之间均用空格隔开，如：“1 引言”、“2.1 需求分析”。报告正文的一级标题（章）须另起一页居中排版。

1级标题宋体小三号字加黑，单倍行距，段前、段后均12磅间隔。

2级标题宋体四号字加黑，1.25倍行距，段前6磅，段后0磅间隔。

3级标题宋体小四号字加黑，单倍行距，段前3磅、段后8磅间隔。

**关于正文中的图的排版要求：**

正文中所有插图要求图面整洁，布局合理，线条粗细均匀，圆弧连接光滑，尺寸标注规范。所有曲线、图表、线路图、流程图、程序框图、示意图等不准徒手画，必须按国家规定标准或工程要求采用计算机或手工绘制。

所有插图均应有图号和图名。图号按章编，如第2章的图为图2.1、图2.2、…，第3章的图为图3.1、图3.2、…等。图名是插图的名称，扼要概括图的内容，字数不宜太多。图号和图名写在图下方，并相对于图居中排版。少数图有图注，图注写在图下面且字号应比图号、图名的字小一号，图名和图注后面均不加标点符号。

所有插图均应在正文中予以引用。引用某插图时，一般写为“…见图x.y”或“图x.y是…”。正文中的插图均须安排在文中第一次引用到该图的正文下面，要求先见文，后见插图，且图一般不跨页绘制。

图中文字、图号和图名，统一采用小五号宋体。

**关于正文中的表的排版要求：**

表格由表号、表名、表头、表身等组成。表号按章编，如第2章的表为表2.1、表2.2、…，第3章的表为表3.1、表3.2、…等。表名是表格的名称，扼要概括表的内容，字数不宜太多。表号、表名放在表的正上方，相对于表体居中排版。表号及表名后不加标点。表头包括栏头、行头，与表身一起构成表格的主体。表中的竖称为栏，横格称为行。表身的内容，一般包括：数据、文字、公式和表图等。表内的数据对应位要对齐。少数表有表注，表注写在表下面且字号应比表号、表名的字小一号。

所有表格均应在正文中予以引用。引用某表格时，一般写为“…见表x.y”或“表x.y是…”。表格应尽量靠近正文的叙述，一般应先见文，后见表，表不跨节。表格允许转页。表格转页部分可以不写表号和表名，但要重复书写表头，并在表头右上角写“（续）”字标注。

表中文字、图号和图名，统一采用小五号宋体。

**关于正文中公式的排版要求：**

公式一般另行居中写，公式末不加标点。若公式前有文字，如例、解、证、假定等，文字顶格写，公式仍居中写。一行写不下时，公式允许转行。公式转行需处理得当，做到既意义正确，又使版面美观匀称。

公式要有编号，公式编号按章编，如第2章的公式为(2.1)、(2.2)、…，第3章的公式为(3.1)、(3.2)、…等。公式编号写在公式右侧行末顶边线，并加圆括号。

公式一般应在正文中予以引用，引用时以公式编号指示公式。正文中常有公式中表示量的符号说明，采用“式中”二字作为标志。一般可写成接排形式，如“式中，A指……；B指……”。

**关于实践报告的目录格式要求：**

目录是课程实验报告的纲要。正文的各级标题（一般最多取三级）、附录应编入目录，但目录本身不出现在其中。

目录要求层次清晰，含标题及对应的起始页号。目录的最后 “附录”无章节号。

课程实验报告正文、参考文献和附录页面，使用“1，2，3，… ”编连续页码。页码应标在页面的右下角。

目录中正文的各级标题名称、参考文献和附录及其对应的起始页号，务必与报告中正文的各级标题名称、附录及其对应的起始页号保持一致。

正文（下同）

小四号 宋体/Times New Roman 1.25倍行距

# 2 软件功能学习部分

## 2.1 任务要求

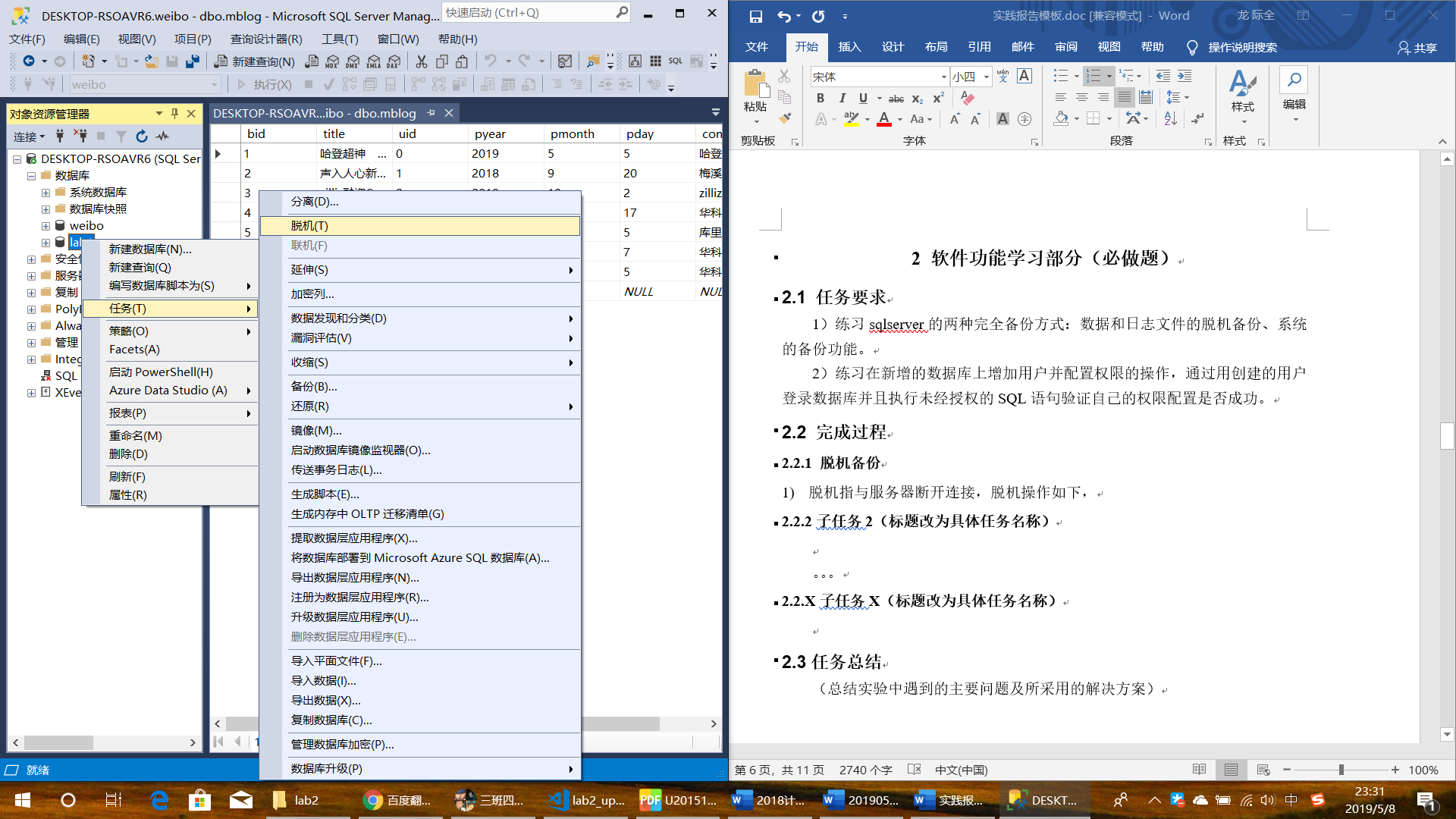
1）练习sqlserver的两种完全备份方式：数据和日志文件的脱机备份、系统的备份功能。

2）练习在新增的数据库上增加用户并配置权限的操作，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。

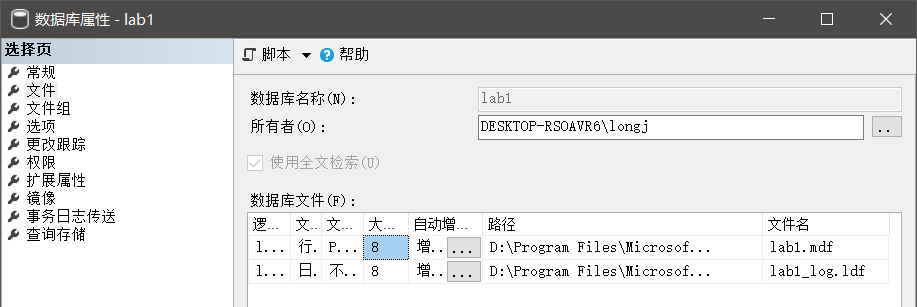
## 2.2 完成过程

### 2.2.1 脱机备份

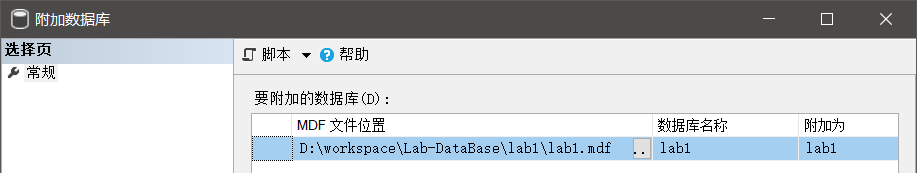
1. 脱机指与服务器断开连接，脱机操作为“**右键-任务-脱机**”，如下：



1. 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，拷贝到想要拷贝的地方即实现了脱机备份。

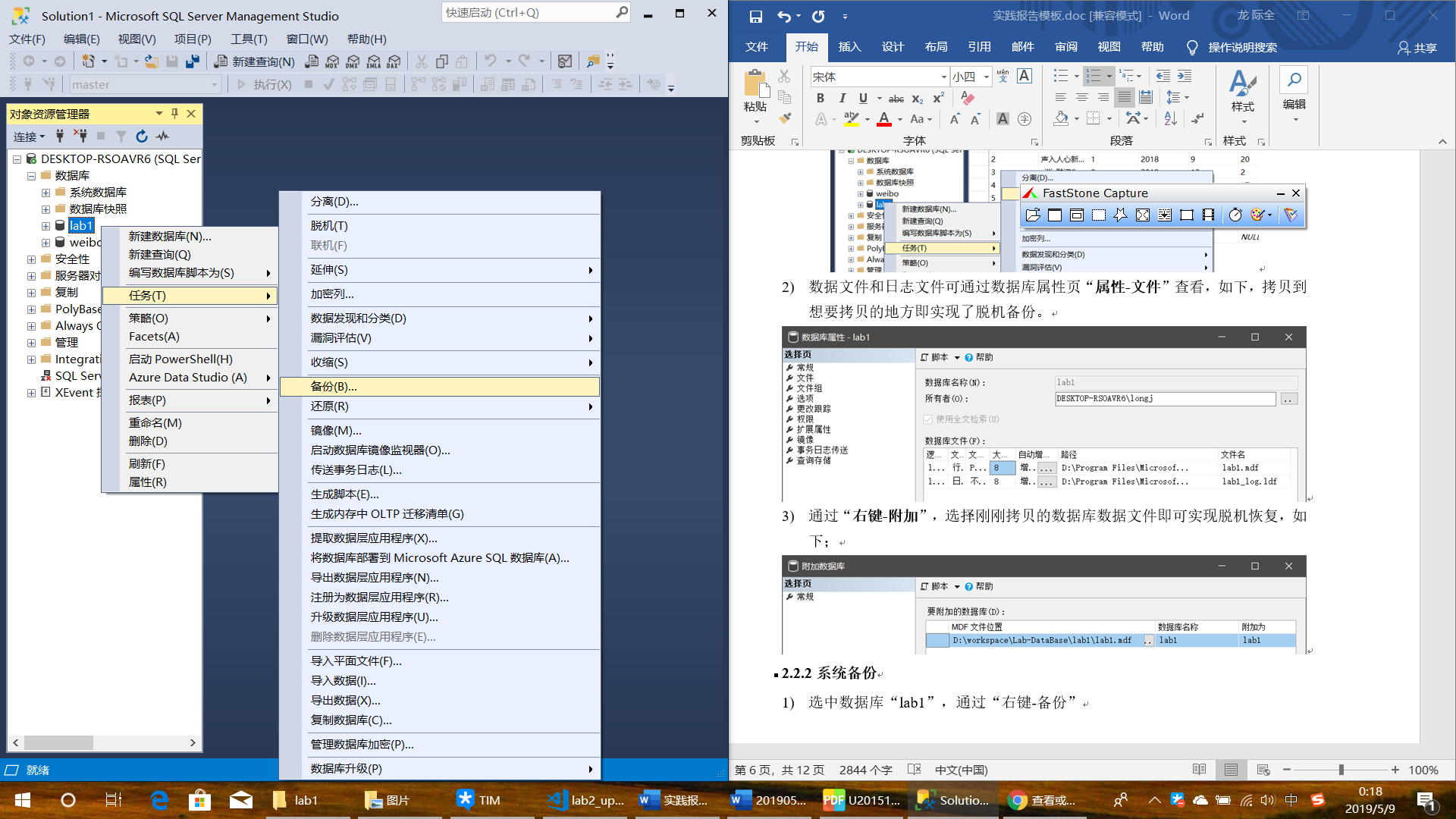


1. 通过“**右键-附加**”，选择刚刚拷贝的数据库数据文件即可实现脱机恢复，如下；

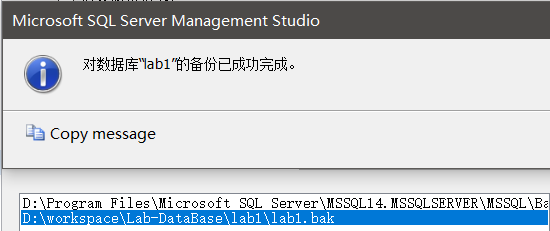


### 2.2.2系统备份

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



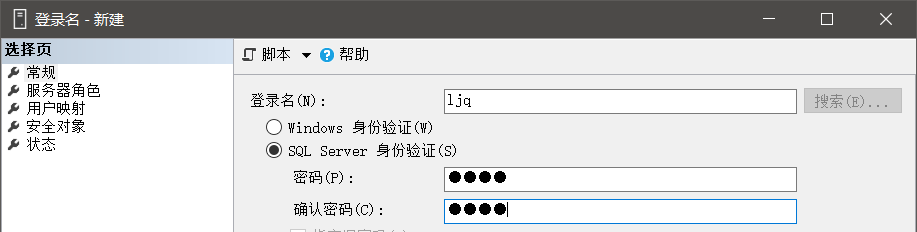
1. 备份到如下文件夹，如下：



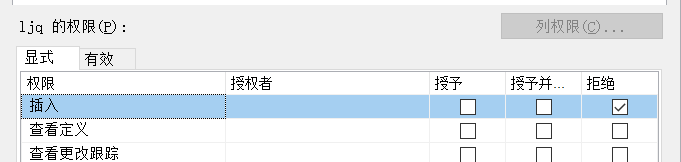
1. 选中数据库“**lab1**”，通过“**右键-任务-还原-数据库**”即可打开还原选项卡，选择刚才生成的.bak文件即可还原。

### 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；



1. 给用户“ljq”授权如下；

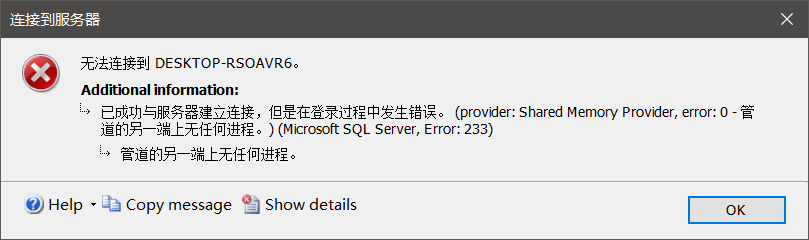


1. 登录数据库，如下；

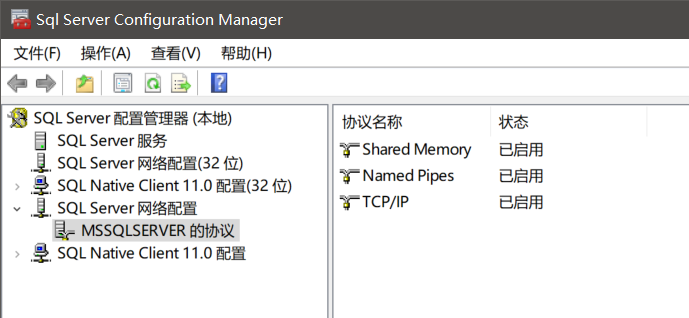


## 2.3任务总结

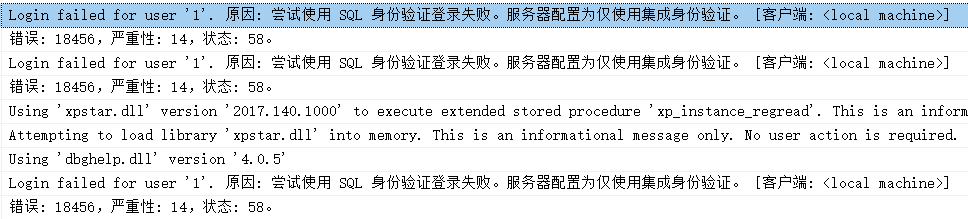
1. 脱机备份
2. 现象：数据库lab1没有脱机时无法将数据库的数据文件lab1.mdf和日志文件lab1\_log.mdf拷贝到其他位置；
3. 原因：数据库未进行脱机操作时，lab1.mdf和lab1\_log.mdf被SSMS通过操作系统进行保护，无法进行拷贝；
4. 解决方法：备份lab1.mdf和lab1\_log.mdf时先对数据库进行脱机操作即可。
5. 用户登录
6. 现象：采用SQL Server身份验证登录时出现运行时错误，如下所示：



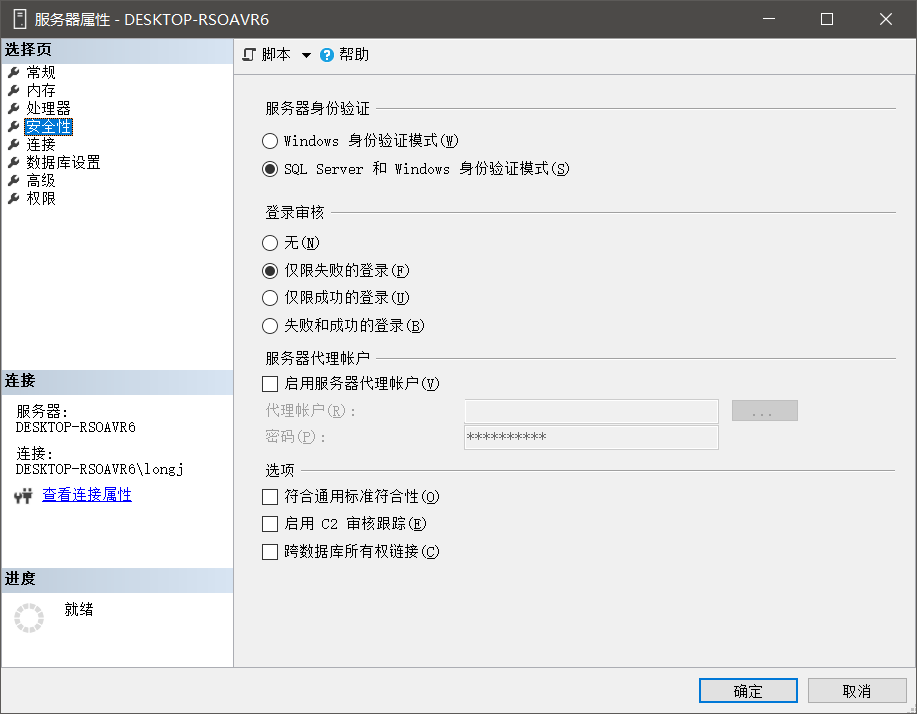
1. 原因：未开启SQL Server的TCP/IP协议和Named Pipes协议；
2. 解决方法：可通过Sql Server Configuration Manager启用，如下。



1. 身份验证
2. 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；
3. 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：



1. 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：



# 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

1. 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)

);

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)

);

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)

);

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)

);

drop table if exists topday;

create table topday(

tyear int not null,

tmonth int not null,

tday int not null,

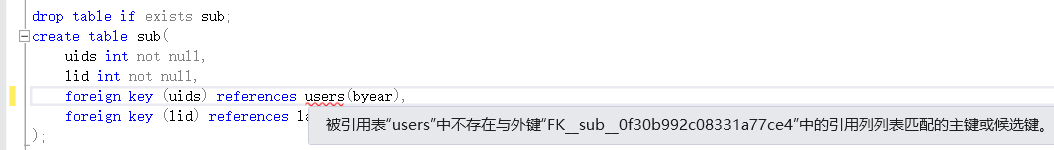
bid int not null,

tno int not null,

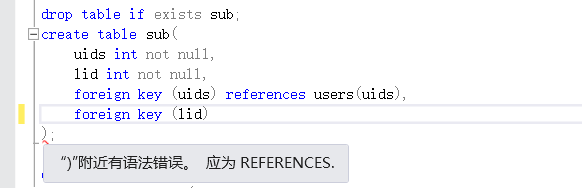
foreign key (bid) references mblog(bid)

);

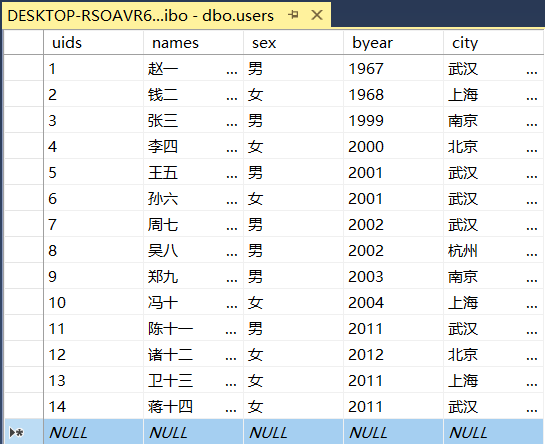
1. 观察性实验
2. 用户在订阅分类时一定要参考被参照关系的主码，如下byear并不是users表的主码，SSMS红线标错；



1. 缺少参照，SSMS仍然红线标错如下。



1. 数据准备（只展示了三组，皆为在SSMS中操作，其他类似）
2. Users



1. Label



1. Mblog



### 3.2.2 数据更新

1. 博文表增删改
2. 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’)

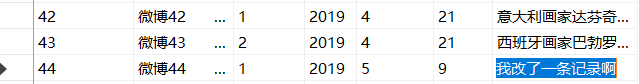
查看mblog表，如下：



1. 改

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

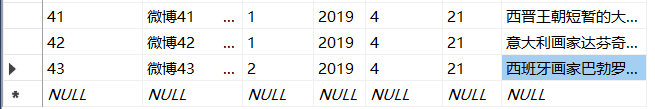
查看mblo表，如下：



1. 删

Delete from mblog where bid = ‘44’

查看mblo表，如下：



1. 批处理操作
2. 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

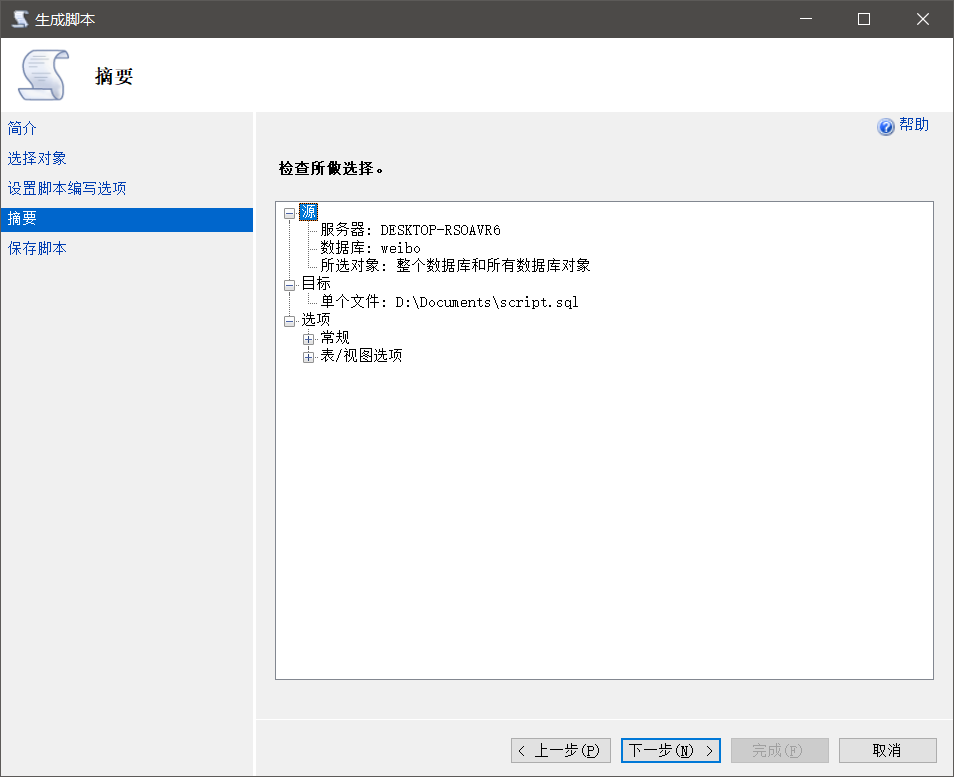
where users.uids = follow.uids and follow.uidfled = 3

1. 操作后的fans\_3表如下：



1. 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：



1. 观察性实验
2. 代码如下：

use weibo;

go

create table observe(

obj1 int,

obj2 text

);

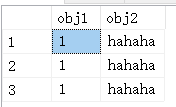
insert into observe values

(1, 'hahaha'),

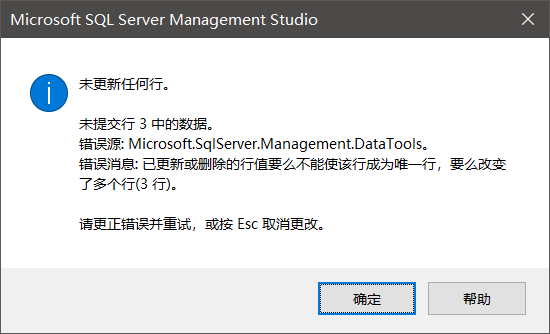
(1, 'hahaha'),

(1, 'hahaha')

1. 更新后的表如下：



1. 对重复元组进行修改或者删除报错，如下：



1. 触发器实验
2. 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

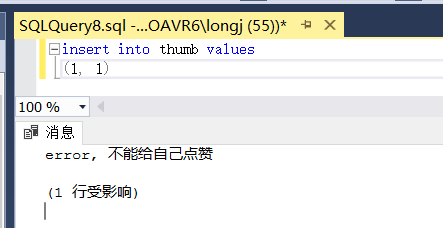
from inserted

end

end

go

1. 测试代码如下：



### 3.2.2 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。
2. 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

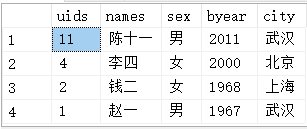
from users

where users.names = '张三'

)

order by users.byear desc, users.uids asc

1. 查询结果：



1. 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。
2. 代码：

select mblog.bid, mblog.title, users.names

from mblog, users

where users.uids = mblog.uids and mblog.bid not in

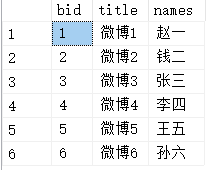
(

select bid from thumb

)

order by mblog.title asc

1. 查询结果：



1. 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；
2. 代码：

select mblog.bid

from mblog, users, topday

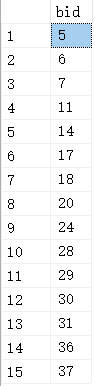
where mblog.uids = users.uids and

topday.bid = mblog.bid and

users.byear > 2000 and

users.city = '武汉'

1. 查询结果：



1. 查找订阅了所有分类的用户ID；
2. 代码：

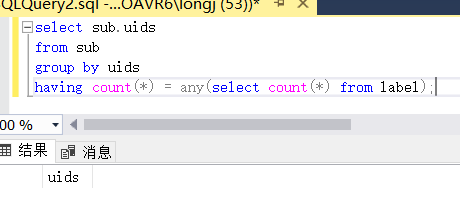
select sub.uids

from sub

group by uids

having count(\*) = any(select count(\*) from label);

1. 查询结果：



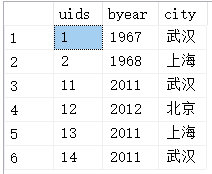
1. 查找出生年份小于1970年或者大于2010年的用户ID、出生年份、所在城市，要求where子句中只能有一个条件表达式；
2. 代码：

select uids, byear, city

from users

where byear not between '1970' and '2010'

1. 查询结果：



1. 统计每个城市的用户数；
2. 代码：

select city, count(uids) '用户数'

from users

group by city

1. 查询结果：



1. 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；
2. 代码：

select city, byear, count(uids)

from users

group by byear,city

order by city, byear desc;

1. 查询结果：



1. 查找被点赞数超过10的博文ID号；
2. 代码：

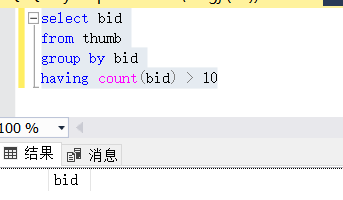
select bid

from thumb

group by bid

having count(bid) > 10

1. 查询结果：



1. 查找被2000年后出生的用户点赞数超过10的博文ID号；
2. 代码：

select bid

from thumb, users

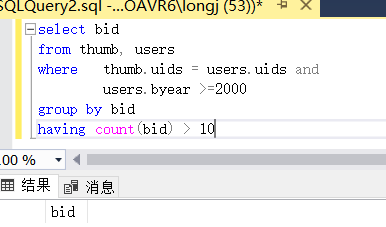
where thumb.uids = users.uids and

users.byear >=2000

group by bid

having count(bid) > 10

1. 查询结果：



1. 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；
2. 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

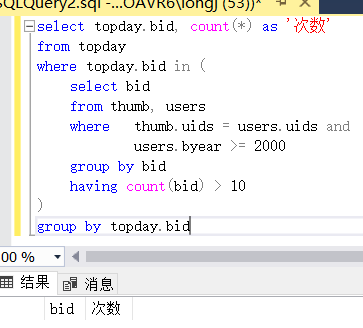
group by bid

having count(bid) > 10

)

group by topday.bid

1. 查询结果：



1. 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；
2. 代码：

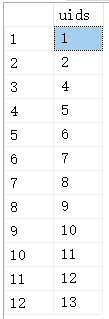
select distinct uids

from sub, label

where sub.lid = label.lid and

label.lname in ('文学','艺术','哲学','音乐')

1. 查询结果：



1. 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；
2. 代码：

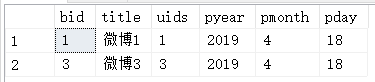
select bid, title, uids, pyear, pmonth, pday

from mblog

where cont like '%最多地铁站%' or

cont like '%\\_华中科技大学%'

1. 查询结果：



1. 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；
2. 代码：

select one.uids, two.uids

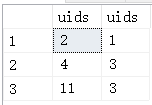
from follow one, follow two

where one.uids = two.uidfled and

one.uidfled = two.uids and

one.uids > two.uids

1. 查询结果：



1. 查找好友圈包含了5号用户好友圈的用户ID；
2. 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

)

)

1. 查询结果：



1. 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；
2. 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'

1. 查询结果：



1. 查找至少有3名共同好友的所有用户对的两个ID号。
2. 代码：

select one.uids, two.uids

from friends one, friends two

where one.uids <> two.uids and

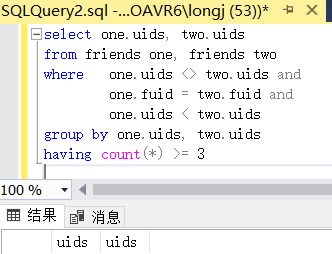
one.fuid = two.fuid and

one.uids < two.uids

group by one.uids, two.uids

having count(\*) >= 3

1. 查询结果：



1. 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. 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

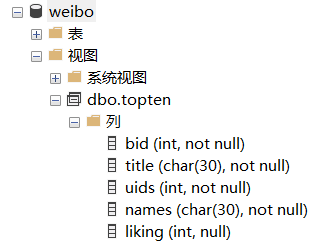
and mblog.uids = users.uids

and thumb.bid = mblog.bid

group by topday.bid, mblog.title, users.uids, users.names

go

1. 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。

# 4 综合实践任务

## 4.1系统设计目标

陈述系统的应用背景和总体目标（300字以内）。

## 4.2 需求分析

展开需求分析，给出需求分析的主要内容的阐述，包括具体的功能需求、性能需求、数据完整性需求、数据流图、数据字典等。

传统的数据库系统多位C/S结构，如图4.1。

。。。

对图的引用



图中文字

小五号 宋体

空1行

图4.1 异构数据库统一检索系统

数据库系统。。。。。

图号 图名

小五号 宋体

置于图的下方

。

。

。

。

。

对表的引用

系统总体需求列表如表4.1所示。

表号 表名 小五号 宋体

置于表的上方

空1行

表4.1 系统总体需求

|  |  |  |  |
| --- | --- | --- | --- |
| 功能编号 | 功能名称 | 功能描述 | 。。。 |
| 1 | 功能1 |  |  |
| 2 | 功能2 |  |  |

注：每漏/多报一个故障算1次；完全一致的情况即所报第1个故障为实际故障。

空1行

表内文字

小五号 宋体

。。。。。

## 4.3 总体设计

阐述系统的总体设计方案，包括系统的C/S或者B/S架构图、功能模块组成及其说明、系统的总体业务流程图及其说明等等总体设计的主要内容。

## 4.4 数据库设计

阐述数据库设计的主要内容，依次是ER图设计及其说明、数据库逻辑结构设计（包括表、视图的逻辑结构说明，含主码、外码说明、用户定义完整性）、数据库物理设计（例如索引）。

## 4.5 详细设计与实现

阐述各主干功能的实现过程，包括主干功能的业务流程图、关键技术和算法说明、数据库事务的定义与实现、数据库函数和触发器的定义与实现等（不允许大段引用源码，如有必要引用必须加详细注释）。

公式的引用

。。。。。的统计如公式4.1所示。

Sum=x+y+z （4.1）

## 4.6 系统测试

包括对测试数据的说明、测试过程阐述、测试结果分析。

## 4.7 系统设计与实现总结

分条目概括、总结应用系统设计的主要工作。

# 5 课程总结

逐条概括、总结此次课程实践的主要工作，阐述此次课程实践的心得体会，展望此次课程实践的有待改进和完善的工作。

# 附录