

# 西安邮电大学

## 网络空间安全学院

### 实验报告

(2022/ 2023 学年第 1 学期)

实验名称: 安全数据库 A

学生姓名: 王璐

专 业: 网安 2101 班

学 号: 26213035

指导教师: 俞惠芳

考核表

学生姓名	王璐	班级/学号	网安 2101/26213035
承担任务实验室 (单位)	网络安全技术实验室 531	所在部门	网络空间安全学院
实施时间	2023 年 05 月 24 日 — 2023 年 06 月 15 日		
实验内容	掌握一种常见的关系数据库管理系统的安装和配置，实现 SQL 语言的基本使用，包括查询、更新、视图、索引的基本使用方法，数据库安全性操作语言和数据库完整性操作语言。		
指导教师 (师傅) 姓名	俞惠芳	职务或职称	教授
指导教师 (师傅) 对学生的评价	评价点	评价内容	评价得分
	平时成绩 (20 分)	课堂出勤、课堂纪律及学习态度 遵守实验室的规章制度等情况	
	实验能力 (30 分)	文件检索或软硬件设计能力 分析、解决实际问题能力	
	实验验收 (20 分)	实验最终完成质量情况 实验演示与答辩情况	
	实验报告 (30 分)	语言及文字表达能力 实验报告结构及书写格式规范	
总评成绩		五级制	
指导教师 (师傅) 评语	指导教师 (师傅) 签字  年 月 日		

# 实验一 MySQL 的安装及使用

## 一、实验目的

1. 掌握 MySQL（或其他关系数据库）的安装；
2. 掌握 MySQL（或其他关系数据库）的配置；
3. 掌握启动和停止 MySQL（或其他关系数据库）的方法；
4. 掌握登录 MySQL（或其他关系数据库）的方法。

## 二、实验要求

1. 了解 MySQL（或其他关系数据库）的安装过程；
2. 熟练掌握 MySQL（或其他关系数据库）的配置操作；
3. 熟练掌握 MySQL（或其他关系数据库）的启动和停止方法；
4. 熟练掌握 MySQL（或其他关系数据库）的登录方式；
5. 独立完成实验内容，并提交实验报告。

## 三、实验内容

下载并安装 MySQL（或其他关系数据库），进行相关配置操作。启动或停止 MySQL（或其他关系数据库）服务，登录 MySQL（或其他关系数据库），设置 root 用户密码，进行简单操作。

## 四、实验步骤

MySQL 的安装

打开浏览器，在地址栏输入：<https://dev.mysql.com/>，注意这是 MySQL 的官方网站。

在下面找到这个按钮

A blue rectangular button with rounded corners containing the text "MySQL Downloads »" in a light blue font.

进去之后点击这个按钮

A blue rectangular button with rounded corners containing the MySQL logo (a yellow circle with a red 'M') and the text "MySQL Installer for Windows" in a light blue font.

选择电脑的系统



选择下载

<b>Windows (x86, 32-bit), MSI Installer</b> (mysql-installer-web-community-8.0.25.0.msi)	8.0.25	2.4M	<a href="#">Download</a>
MD5: 7f64b7ca2c526c359919d9db1bc8d609   <a href="#">Signature</a>			
<b>Windows (x86, 32-bit), MSI Installer</b> (mysql-installer-community-8.0.25.0.msi)	8.0.25	435.7M	<a href="#">Download</a>
MD5: 0c64075a9cc4ec00cce35806761aafc9   <a href="#">Signature</a>			

We suggest that you use the [MD5 checksums](#) and [GnuPG signatures](#) to verify the integrity of the packages you download.

安装好之后在 windows 命令行窗口输入 MySQL

```
C:\Users\jzh>MySQL
ERROR 1045 (28000): Access denied for user 'ODBC'@'localhost' (using password: NO)
C:\Users\jzh>
```

出现这个就代表安装成功，可以运行

MySQL 的使用

- 1 进入 mysql，在命令行中输入: `mysql -uroot -p####` (其中: ####表示密码)
- 2 查看 mysql 中有哪些个数据库: `show databases;`
- 3 使用一个数据库: `use 数据库名称;`
- 4 新建一个数据库: `create database 数据库名`
- 5 查看指定的数据库中有哪些数据表: `show tables;`
- 6 查看表的结构: `desc 表名`
- 7 删除表: `drop table 表名`

创建数据表

在数据库中，数据表是数据库中最重要、最基本的操作对象，是数据存储的基本单位。数据表被定义为列的集合，数据在表中是按照行和列的格式来存储的。每一行代表一条唯一的记录。每一列代表记录的一个域。

创建数据表的语法:

`create table 表名`

(

列名数据类型[列级别约束条件][默认值],

列名数据类型[列级别约束条件] [默认值],  
 列名数据类型[列级别约束条件] [默认值],  
 列名数据类型[列级别约束条件] [默认值],  
 ....  
 [表级别约束条件]  
 )

注意:

由于数据表属于数据库，在创建数据表之前，应该使用语句“use 数据库名”指定操作是在哪个数据库中进行，如果没有选择数据库，会抛出“No database selected”的错误。在使用该命令创建数据表时，必须指定以下信息：

- 1: 要创建的表的名称，不区分大小写，不能使用 SQL 语言中的关键字，如 DROP、ALTER、INSERT、和 DESC 等。
- 2: 数据表中的每一列（字段）的名称和数据类型。
- 3: 多个列之间使用都好进行分割。

#### MySQL 的数据类型

MySQL 支持多种数据类型，主要有数值类型、日期/时间类型、字符串类型和二进制数据类型。

##### 数值类型:

数值类型时现实生活中经常遇到的数据类型之一，例如：公司的员工数、销售额、利润、工资、学生的考试分数以及年龄等。只有使用了数值类型的列，才能进行汇总运算、平均值运算等数学统计或者数学计算。

类型	大小	范围（有符号）	范围（无符号）	用
TINYINT	1 byte	(-128, 127)	(0, 255)	小值
SMALLINT	2 bytes	(-32 768, 32 767)	(0, 65 535)	大值
MEDIUMINT	3 bytes	(-8 388 608, 8 388 607)	(0, 16 777 215)	大值
INT 或	4 bytes	(-2 147 483 648, 2 147 483 647)	(0, 4 294 967 295)	大值

类型	大小	范围 (有符号)	范围 (无符号) 用途
INTEGER			值
BIGINT	8 bytes	(-9,223,372,036,854,775,808, 9 223 372 036 854 775 807)	(0, 18 446 744 073 709 551 615) 极大数
FLOAT	4 bytes	(-3.402 823 466 E+38, -1.175 494 351 E-38) (1.175 494 351 E-38, 3.402 823 466 351 E+38)	0, (1.175 494 351 E-38, 3.402 823 466 E+38) 单精度浮点值
DOUBLE	8 bytes	(-1.797 693 134 862 315 7 E+308, -2.225 073 858 507 201 4 E-308) (1.797 693 134 862 315 7 E+308, 2.225 073 858 507 201 4 E-308)	0, (2.225 073 858 507 201 4 E-308, 2.225 073 858 507 201 4 E+308) 双精度浮点值

对 DECIMAL(M,D) ,  
DECIMAL 果 M>D, 为 M+2 否 依赖于 M 和 D 的值  
D+2 依赖于 M 和 D 小

字符串类型:

字符串类型也是数据表中数据存储的重要类型之一，字符串类型主要时用来存储字符串或文本信息的。

类型	大小	用途
CHAR	0-255 bytes	定长字符串
VARCHAR	0-65535 bytes	变长字符串
TINYBLOB	0-255 bytes	不超过 255 个字符的二进制字符串
TINYTEXT	0-255 bytes	短文本字符串
BLOB	0-65 535 bytes	二进制形式的长文本数据
TEXT	0-65 535 bytes	长文本数据
MEDIUMBLOB	0-16 777 215 bytes	二进制形式的中等长度文本数据
MEDIUMTEXT	0-16 777 215 bytes	中等长度文本数据
LONGBLOB	0-4 294 967 295 bytes	二进制形式的极大文本数据

类型	大小	用途
LONGTEXT	0-4 294 967 295	极大文本数据

### 日期和时间类型

MySQL 中有多种表示日期和时间的数据类型。其中 YEAR 表示年份，DATE 表示日期，TIME 表示时间

类型	范围	格式	用途
DATE	'1000-01-01/9999-12-31'	YYYY-MM-DD	日期值
TIME	'-838:59:59'/838:59:59'	HH:MM:SS	时间值 持续时间
YEAR	1901/2155	YYYY	年份值
DATETIME	'1000-01-01 00:00:00/9999-12-31 23:59:59'	YYYY-MM-DD HH:MM:SS	混合日期 时间值
TIMESTAMP	'1970-01-01 00:00:00/2038-12-31 23:59:59'	YYYYMMDD HHMMSS	混合日期 时间值，戳

### 二进制数据类型:

二进制类型是在数据库中存储二进制数据的数据类型，二进制类型包括 BINARY、VARBINARY、BIT、TINYBLOB、BLOB、MEDIUMBLOB、LONGBLOB。

## 五、实验总结及心得体会

安装 MySQL 可以参考官方网站提供的下载安装程序，安装过程中需要设置密码和其他管理参数。在 MySQL 中，我们首先要创建一个数据库，然后可以创建表格和添加数据。

MySQL 具有强大的查询语言，例如 SELECT 语句可以从表格中读取数据，JOIN 语句可以联接多个表格。

可以使用 UPDATE 和 DELETE 语句更新和删除数据。

数据库和表格的命名应该具有表现力和可读性，这样可以方便后期的维护和管理。

在创建表格时，应该选择合适的数据类型和约束条件，以确保数据的完整性和准确性。

在编写 SQL 语句时，应该注意使用标准的数据库设计原则，例如第一范式、第二范式和第三范式等。

在使用 MySQL 进行开发时，建议使用编辑器或 IDE 等工具，可以提高效率和代码的可维护性。总的来说，学习和使用 MySQL 需要掌握一些基础知识和技巧，例如 SQL 语言、数据库设计原则、数据类型、约束条件等等。通过不断实践和总结，我们可以更加熟练地使用 MySQL，并且能够更好地应用它来解决实际问题。



## 实验二 SQL 语言的基本使用

### 一、实验目的

5. 掌握 SQL 语言中数据库的相关操作语句;
6. 掌握 SQL 语言中基本表的相关操作语句;
7. 掌握 SQL 语言中数据的相关操作语句;
8. 掌握 SQL 语言中视图的相关操作语句;
9. 掌握 SQL 语言中索引的相关操作语句。

### 二、实验要求

6. 熟练掌握 SQL 语言中数据库的查看、创建、修改、删除、选择语句;
7. 熟练掌握 SQL 语言中基本表的创建、修改、删除语句;
8. 熟练掌握 SQL 语言中数据的查询、插入、修改、删除语句;
9. 熟练掌握 SQL 语言中视图的创建、删除、查询、更新语句;
10. 熟练掌握 SQL 语言中索引的创建、修改、删除语句;
11. 独立完成实验内容, 并提交实验报告。

### 三、实验内容

以教材中学生-课程数据库为例, 使用 SQL 语言实现数据库的查看、创建、修改、删除、选择等操作; 实现基本表的创建、修改、删除等操作; 实现数据的查询、插入、修改、删除等操作; 实现视图的创建、删除、查询、更新等操作; 实现索引的创建、修改、删除等操作。

### 四、实验步骤

1. 导入与导出表数据如图 1.1.1, 图 1.1.2, 图 1.1.3, 字段值如果是字符就用双引号标注, 字段值之间用逗号隔开, 每行以“?”为结束标志;

```
mysql> select * from xc
-> into outfile 'e:/myfile1.txt'
->         fields terminated by ','
->         optionally enclosed by '"'
->         lines terminated by '?';
```

图 1.1.1

myfile1.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

学号	姓名	专业名	性别
0001	张三	计算机	1
0002	李四	计算机	0
0003	王五	通讯工程	1
0004	吴六	软件工程	1
0005	赵七	计算机	0
0006	钱八	计算机	0
0007	孙九	计算机	0

图 1.1.2

```
mysql> load data infile 'e:/myfile1.txt'
-> into table course
->      fields terminated by ','
->      optionally enclosed by '"'
->      lines terminated by '\n';
```

图 1.1.3

2.备份数据库如图 2.1.1，图 2.1.2，假如数据库损坏，可用如图 2.1.3 指令，恢复数据则用如图 2.1.4；

```
C:\Users\iiLincolnii>cd D:\mysql\mysql80\bin
```

图 2.1.1

```
C:\Users\iiLincolnii>mysqldump -u root -p xscj xc > twotables.sql
Enter password: ***

C:\Users\iiLincolnii>
```

图 2.1.2

```
C:\Users\iiLincolnii>mysql -uroot -p620102 xscj < c:/xc.sql
```

图 2.1.3

```
C:\Users\iiLincolnii>mysqldump -uroot -p620102 xscj > xscj.sql
```

图 2.1.4

3.创建一个新用户如图 3.1.1，从 user 表中查到刚刚添加的两个用户记录图 3.1.2，图 3.1.3；

```
mysql> create user
-> 'WIN2000 (NT)' '@'localhost' identified by '1234',
-> 'WIN2000 (NT)' '@'localhost' identified by '1234';
Query OK, 0 rows affected
```

图 3.1.1

```
mysql> use mysql;
Database changed
mysql> show tables;
+-----+
| Tables_in_mysql |
+-----+
| columns_priv     |
| component        |
| db               |
| default_roles    |
| engine_cost      |
| func             |
| general_log      |
| global_grants    |
| gtid_executed     |
| help_category    |
| help_keyword     |
| help_relation    |
| help_topic       |
| innodb_index_stats |
| innodb_table_stats |
| password_history  |
| plugin           |
| procs_priv       |
| proxies_priv     |
| role_edges       |
| server_cost      |
| servers          |
| slave_master_info |
| slave_relay_log_info |
| slave_worker_info |
| slow_log         |
| tables_priv      |
| time_zone        |
| time_zone_leap_second |
| time_zone_name   |
| time_zone_transition |
| time_zone_transition_type |
| user             |
+-----+
33 rows in set
```

图 3.1.2



图 4.1

```
mysql> grant all
-> on xc
-> to WIN2001 (NT) @localhost;
Query OK, 0 rows affected
```

图 4.2

如图 4.3，就是限制用户每小时只能执行一条 select 命令；

```
mysql> grant select
-> on xc
-> to WIN2001 (NT) @localhost
-> with max_queries_per_hour 1;
```

图 4.3

如图 4.4，将用户的权限回收；

```
mysql> revoke select
-> on xc
-> from WIN2001 (NT) @localhost;
Query OK, 0 rows affected
```

图 4.4

5.更新表 xc 的索引的可压缩性如图 5.1.1，图 5.1.2，变量如图 5.2；

```
mysql> analyze table xc;
```

Table	Op	Msg_type	Msg_text
xscj.xc	analyze	status	OK

```
1 row in set

mysql> show index from xc;
```

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Pa
ed	Null	Index_type	Comment	Index_comment	Visible	Expression		
xc	0	PRIMARY	1	学号	A	7	NULL	NU
		BTREE		YES	NULL			
xc	1	xh_xc	1	学号	A	7	5	NU
		BTREE		YES	NULL			

```
2 rows in set
```

图 5.1

检查表是否正确如图 5.2;

```
mysql> check table xc;
```

Table	Op	Msg_type	Msg_text
xscj.xc	check	status	OK

```
1 row in set
```

5.2

获得表 xc 的校验和的值如图 5.3;

```
mysql> checksum table xc;
```

Table	Checksum
xscj.xc	1991839049

```
1 row in set
```

图 5.3



优化表 xc, 如图 5.4;

```
mysql> optimize table xc;
```

Table	Op	Msg_type	Msg_text
xscj.xc	optimize	note	Table does not support optimize, doing recreate + analyze instead
xscj.xc	optimize	status	OK

```
2 rows in set
```

图 5.4

6.事务处理如图 6.1;

```
mysql> set @@autocommit=0;
Query OK, 0 rows affected

mysql> delete from xc
      -> where 学号='0008';
Query OK, 0 rows affected

mysql> select*from xc;
```

学号	姓名	专业名	性别
0001	张三	计算机	1
0002	李四	计算机	0
0003	王五	通讯工程	1
0004	吴六	软件工程	1
0005	赵七	计算机	0
0006	钱八	计算机	0
0007	孙九	计算机	0

```
7 rows in set
```

图 6.1

开始事务如图 6.2.1, begin work 可代替 start transaction, 而结束事务如图 6.2.2, 撤销事务如图 6.2.3, 回滚事务则为图 6.2.4;

```
mysql> begin work;
Query OK, 0 rows affected
```

图 6.2.1

```
mysql> commit work;  
Query OK, 0 rows affected
```

图 6.2.2

```
mysql> rollback work;  
Query OK, 0 rows affected
```

图 6.2.3

```
mysql> savepoint identifier;  
Query OK, 0 rows affected
```

图 6.2.4

## 五、实验总结及心得体会

事务属性有三种，1.原子性意味着每个事务都必须被看作是一个不可分割的单元。假设一个事务由两个或者多个任务组成，其中的语句必须同时成功才能认为整个事务是成功的。如果事务失败，系统将会返回到该事务以前的状态。2.不管事务是完全成功完成还是中途失败，当事务使系统处于一致的状态时存在一致性。3.隔离性是指每个事务在它自己的空间发生，和其他发生在系统中的事务隔离，而且事务的结果只有在它完全被执行时才能看到。4.持久性是指即使系统崩溃，一个提交的事务仍然存在。当一个事务完成，数据库的日志已经被更新时，持久性就开始发生作用。

几次实验完成下来，对事务的处理，数据库备份，创建用户等操作更加熟悉了，从起初的不了解数据库，到现在可以较为熟练地在 MySQL 中进行系列操作，都是在练习中一点一点进步，甚至有些命令可以脱离课本和课件，因为自己再实验室学习的方向也主要是 Web 前端，所以之后也会继续对数据库进行基础的学习，多掌握一种技能。



## 实验三 数据库安全性

### 一、实验目的

10. 掌握 SQL 语言中用户的相关操作语句;
11. 掌握 SQL 语言中角色的相关操作语句;
12. 掌握 SQL 语言中视图的相关操作语句;
13. 掌握 SQL 语言中审计的相关操作语句。

### 二、实验要求

12. 熟练掌握 SQL 语言中用户的创建、修改、删除及用户权限的查看、授予、撤销语句;
13. 熟练掌握 SQL 语言中角色的创建、删除及角色权限的授予、收回语句;
14. 熟练掌握 SQL 语言中视图的创建及用户对视图的权限的授予、收回语句;
15. 熟练掌握 SQL 语言中审计的设置、取消语句;
16. 独立完成实验内容, 并提交实验报告。

### 三、实验内容

以教材中学生-课程数据库为例, 使用 SQL 语言实现用户的创建、修改、删除及用户权限的查看、授予、撤销等操作; 实现角色的创建、删除及角色权限的授予、收回等操作; 实现视图的创建及用户对视图的权限的授予、收回等操作; 实现审计的设置、取消等操作。

### 四、实验步骤

#### 1. 准备实验数据

创建数据库 s\_t, 创建学生表 student、课程表 course、选修表 sc, 向表中添加适量数据。

#### 2. 在 s\_t 数据库中, 完成以下操作

##### (1) 创建用户 zhangsan 和 wangwu

```
create user zhangsan identified by '123456' ;
```

```
CREATE USER wangwu IDENTIFIED BY '123456';
```

##### (2) 授予用户 zhangsan 拥有学生表的所有权限, 并可以给其他用户授权。

```
grant all
```

```
on s_t.student
```

```
to zhangsan
```

```
with grant option;
```

##### (3) 授予用户 zhangsan 在 s\_t 数据库中创建视图的权限(create view)。

```
grant create view
```

```
on s_t.*
```

```
to zhangsan;
```

##### (4) 授予用户 wangwu 对学生表具有查看权限, 对姓名具有更新权限。

```
GRANT SELECT,UPDATE(Sname)
```

```
ON TABLE student
```

TO wangwu;

(5) 使用用户 zhangsan 登录数据库，并将学生表的删除权限授予用户 wangwu。

grant delete on shujuku.student to wangwu;

```
C:\Users\>mysql -h localhost -P 3306 -u zhangsan -p123456
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 22
Server version: 8.0.18 MySQL Community Server - GPL

Copyright (c) 2000, 2019, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> grant delete on table student to wangwu;
ERROR 1046 (3D000): No database selected
mysql> grant delete on shujuku.student to wangwu;
Query OK, 0 rows affected (0.01 sec) https://blog.csdn.net/Happy\_change
```

(6) 用户 zhangsan 创建视图 student\_info，视图包含信息系学生的学号、姓名和年龄。

```
mysql> use shujuku;
Database changed
mysql> create view student_info
-> as
-> select Sno, Sname, Sage
-> from student;
Query OK, 0 rows affected (0.01 sec)
```

(7) 将视图 student\_info 的查询权限授予用户 wangwu。

```
mysql> grant select
-> on shujuku.student_info
-> to wangwu;
ERROR 1142 (42000): SELECT, GRANT command denied to user '
```

网上查 1142 (42000) 错误，发现是权限不够

(8) 分别使用用户 zhangsan 和 wangwu 登录数据库，设计 SQL 语句验证其是否具有相应权限。

show grants for zhangsan;

```
mysql> show grants for zhangsan;
+-----+
| Grants for zhangsan@% |
+-----+
| GRANT USAGE ON *.* TO `zhangsan`@`%` |
| GRANT CREATE VIEW ON `shujuku`.* TO `zhangsan`@`%` |
| GRANT ALL PRIVILEGES ON `shujuku`.`student` TO `zhangsan`@`%` WITH GRANT OPTION |
+-----+
3 rows in set (0.00 sec)
```

show grants for wangwu;

```
mysql> show grants for wangwu;
+-----+
| Grants for wangwu@% |
+-----+
| GRANT USAGE ON *.* TO `wangwu`@`%` |
| GRANT SELECT, UPDATE (`Sname`), DELETE ON `shujuku`.`student` TO `wangwu`@`%` |
+-----+
2 rows in set (0.00 sec)
```

(9) 收回授予 zhangsan 和 wangwu 的所有权限

REVOKE ALL

ON shujuku.\*

FROM zhangsan;

REVOKE ALL

ON shujuku.student

FROM zhangsan;

REVOKE ALL

ON shujuku.student

FROM zhangsan;

(10) 为教务管理创建管理人员角色 manager 和教师角色 teacher，通过对角色授权，使得管理人员拥有学生表和课程表的所有权限，对选课表可以进行查询、删除操作，并可以添加选课记录（不含成绩）；使得教师能够查询所有表，并可以修改选课表中的成绩。

CREATE role manager;

CREATE role teacher;

GRANT ALL

ON shujuku.student

TO manager;

GRANT ALL

ON shujuku.course

TO manager;

```
GRANT INSERT(Sno,Cno),SELECT,DELETE
ON shujuku.sc
to manager;
```

```
GRANT SELECT
ON shujuku.*
TO teacher;
```

```
GRANT ALL
ON shujuku.sc
TO teacher;
```

```
grant insert(sno,cno)
On sc
To manager;
```

(11) 将角色 manager 授予管理员 zhangsan, 角色 teacher 授予教师 wangwu。

```
GRANT manager
TO zhangsan;
```

```
GRANT teacher
TO wangwu;
```

(12) 使用用户 zhangsan、wangwu 分别登陆数据库, 设计 SQL 语句验证其是否获得相应权限。

```
select current_role();
set role manager;
```

(13) 删除角色 teacher 和用户 wangwu。

```
DROP role teacher;
DROP role manager;
```

## 五、实验总结及心得体会

使用 grant 和 revoke 语句向用户授予或收回对数据的操作权限。

Mysql 创建用户，例：create user zhangsan identified by '123456' ;

在使用 mysql 可视化工具 Navicat for mysql，这些功能不好验证，没有找到使用用户登录数据库的方法，于是在创建好用户对用户功能进行验证时大多使用的命令行，在命令行输入相关语句。

查看用户权限

在命令行使用语句 show grants for zhangsan;可查看用户张三的权限

也可在可视化工具中查看图形化的用户权限界面。

## 实验四 数据库完整性

### 一、实验目的

14. 掌握 SQL 语言中实体完整性的相关操作语句;
15. 掌握 SQL 语言中参照完整性的相关操作语句;
16. 掌握 SQL 语言中用户定义完整性的相关操作语句;
17. 掌握 SQL 语言中完整性约束的相关操作语句;
18. 掌握 SQL 语言中触发器的相关操作语句。

### 二、实验要求

17. 熟练掌握 SQL 语言中实体完整性（码）的定义语句;
18. 熟练掌握 SQL 语言中参照完整性（外码）的定义语句;
19. 熟练掌握 SQL 语言中用户定义完整性在属性上和元组上的定义语句;
20. 熟练掌握 SQL 语言中完整性约束的定义、修改语句;
21. 熟练掌握 SQL 语言中触发器的定义语句;
22. 独立完成实验内容，并提交实验报告。

### 三、实验内容

以教材中学生-课程数据库为例，使用 SQL 语言实现实体完整性（码）的定义等操作；实现参照完整性（外码）的定义等操作；实现用户定义完整性在属性上和元组上的定义等操作；实现完整性约束的定义、修改等操作；实现触发器的定义等操作。

### 四、实验步骤

#### 1. 完整性约束的添加、删除

约束是强制数据完整性的首选方法。约束有 6 种类型：非空约束、默认值约束、Check 约束、主键约束、外键约束、唯一性约束。

##### （1）使用企业管理器实施约束

1) 为表 Student 的 Sdept 字段即所在系字段，设置默认值约束，默认值取‘CS’。

添加默认约束的操作步骤：

删除默认约束的操作步骤：

2) (若原有约束, 请在企业管理器中删除 Student 表的主键约束) 在企业管理器中将 Sno 重设为主键。(会操作即可, 操作步骤可略)

3) 为 Student 表的 Sname 字段添加唯一性约束。

添加唯一性约束操作步骤:

删除唯一性约束操作步骤:

4) (若原有约束, 请在企业管理器中删除原有的外键约束) 在企业管理器中删除 SC 原有的外键约束, 在企业管理器中对 Sno, Cno 字段设置外键约束, 约束名自己取。并允许级联删除与级联更新。

添加外键约束的操作步骤:

删除外键约束的操作步骤:

(2) 使用 T-SQL 语句添加、删除约束 (约束名自取, 并自行添加或修改某些字段值检验有效性)

1) 为 Student 表的 Sno 字段添加一个 check 约束, 使学号满足如下条件:

- a. 学号前四位为 2002。
- b. 学号后五位为数字字符。

T-SQL 语句为:

```
create table Student
```

```
(Sno char(9) check(Sno like ' [2002][0-9][0-9][0-9][0-9]' ));
```

2) (若原有约束, 请在企业管理器中删除 SC 表主键约束及外键约束) 将 SC 表中的 Sno, Cno 字段设置为主键。

T-SQL 语句为:

```
alter table SC
```

```
add constraint PK_SC primary key(Sno,Cno)
```

3) 为 SC 表中的 Sno, Cno 添加外键约束, 约束名自取; 并允许级联删除与级联更新。  
添加外键约束 T-SQL 语句为:

```
alter table SC
add constraint FK_Sno foreign key(Sno) references Student(Sno)
on delete cascade on update cascade;
alter table SC
add constraint FK_Cno foreign key(Cno) references Course(Cno)
on delete cascade on update cascade
```

验证级联删除与级联更新 T-SQL 语句:

(测试实例自编)

```
**select from Student
where Sno= '200200002'
select from SC
where Sno= '200200001'
delete from Student
where Sno= '200200001'
select from Student
where Sno= '200200001'
selectfrom SC
where Sno= '200200001'
```

## 2. 通用默认值的实施

(1) 使用企业管理器实现

1) 使用企业管理器中为数据库 Stu\_Course 创建一个默认的年龄, 名称自取, 值为: 19。  
具体步骤如下:

2) 将该默认值绑定到 Student 表中的 Sage 列。

具体步骤如下:



3) 将 Sage 列上的绑定解除，并在企业管理器中删除该默认值。

具体步骤如下：

若未解除绑定，能否删除默认值？

(2) 使用 T-SQL 语句实施默认值

1) 为数据库 Stu\_Course 创建一个关于性别的默认值，名称自取，值为“女”。

T-SQL 语句为：

```
create default about_sex as '女'
```

2) 将该默认值绑定到 Student 表中的 Ssex 字段（原有的约束请删除）

T-SQL 语句：

```
sp_bindefault about_sex , "Student.Ssex"
```

3) 解除表 Student 的 Ssex 列的默认值绑定，并删除该默认值。

T-SQL 语句：

```
sp_unbindefault 'Student.Ssex'
```

```
drop default about_sex
```

3. 规则的实施

(1) 使用企业管理器实现

1) 使用企业管理器为数据库 Stu\_Course 创建一个关于性别的取值规则，规则名自取，要求字段的取值仅能为‘男’或‘女’。

具体步骤如下：

2) 将该规则绑定到 Student 表的 Ssex 字段上。

具体步骤如下：

3) 解除 Student 表的 Ssex 列上的绑定, 并删除该规则。

具体步骤如下:

(2) 使用 T-SQL 实施规则

1) 为数据库 Stu\_Course 创建一个关于学号的规则, 名称自取, 学号的具体取值规则为:

a.学号前四位为 2002。

b.学号后五位为数字字符。

T-SQL 语句为:

```
create rule rule_Sno
```

```
as @range like ' 2002[0-9][0-9][0-9][0-9]
```

2) 将该规则绑定到表 Student 表、SC 表的 Sno 列上。

T-SQL 语句为:

```
exec sp_bindrule 'rule_Sno' , 'Student.Sno'
```

```
exec sp_bindrule 'rule_Sno' , 'SC.Sno'
```

3) 解除所有的绑定, 并删除该规则。

T-SQL 语句为:

```
exec sp_unbindrule 'Student.Sno'
```

```
exec sp_unbindrule 'SC.Sno'
```

## 五、实验总结及心得体会

主键: 主键是必须唯一且非空的字段, 用于唯一地标识表格中的每个行记录, 使用主键可以有效地保证数据的唯一性和完整性。

外键: 外键是用来建立两个表之间关系的字段, 一个表的外键指向另一个表的主键。使用外键可以保证数据的一致性和完整性, 确保不会出现数据冗余和不一致的情况。

唯一约束: 通过唯一约束可以保证数据的唯一性, 即某个字段的值不能在表格中重复出现, 做到了数据的严格验证。

非空约束：非空约束可以确保某些字段不为空，如必填字段，保证数据的完整性和有意义性。

默认约束：默认约束通过为某个字段提供默认值，确保数据的一致性，有效地避免了数据项空值引起的问题。

总之，在数据库设计的过程中，必须对表格中的数据项设置以上类型的约束条件，同时，通过触发器进行触发，实现数据完整性的验证和保证。这样才能让数据库中的数据高效、准确和可信，更好的维护并促进应用的正常运行。

通过完成此实验，我更加深入了解数据库完整性的概念和原理，并通过在实验中的操作来掌握了如何在 MySQL 数据库中实现数据完整性的约束。同时，我认识到数据完整性是非常重要的，特别是在一些重要应用数据库中。在实际开发中，数据库完整性必须得到重视，着重于对数据约束、完整性等方面的应用，尤其是在数据存储与处理方面必须严格按照实际要求进行验证。