

# 《数据库系统概论》实验指导书

华 中 科 技 大 学  
网络空间安全学院

2019 年 11 月

## 实验前的准备工作

实验的环境为 Windows 操作系统。

数据库使用 Microsoft SQLServer 或者 MySQL 免费开源版。

实验前请自己下载相关的数据库管理系统(DBMS), 可以是最新版或者是任意可运行的版本。

请首先熟练掌握数据库管理系统的安装过程。Microsoft SQLServer 安装时建议选择混合模式的身份验证方式。记住系统管理员的密码。

针对 Microsoft SQLServer, 要熟练掌握服务管理平台和查询分析器的界面操作。针对 MySQL 需要同时安装其可视化客户端管理工具, 例如 navicat 或者 MSQL-Front 等, 并熟练掌握其操作。需要先熟练掌握数据库创建、创建表、创建用户、使用不同的用户进行登录等基本操作。

最后一个实验需要进行数据库应用开发, 需要先熟悉一门编程语言, 例如 VC++、Delphi、PB、VB、Java、Python 等, 或者各种 .net 环境。

如果你准备使用 B/S 模式, 还需要先熟练掌握 Tomcat、WebLogic 或者 IIS 等应用服务器的安装和应用。

如果使用 MySQL 开发 C/S 程序还需要安装其 ODBC 接口。

如果使用 Java 开发应用程序还需要熟悉 Java 的集成开发环境例如 Eclipse 或 MyEclipse 等, 并安装 Java 插件。使用其他语言开发, 也需要熟悉类似的集成开发环境。如果开发安卓类移动应用, 需要安装安卓模拟器。

## 实验一 数据库定义与基本操作（4 学时）

## 1、实验目的

- (1) 掌握 DBMS 的数据定义功能
- (2) 掌握 SQL 语言的数据定义语句
- (3) 掌握 DBMS 的数据单表查询功能
- (4) 掌握 SQL 语言的数据单表查询语句

## 2、实验内容

- (1) 创建数据库
- (2) 创建、删除表
- (3) 查看、修改表的定义
- (4) 理解索引的特点
- (5) 创建和删除索引
- (6) SELECT 语句的基本用法
- (7) 使用 WHERE 子句进行有条件的查询
- (8) 使用 IN, NOT IN, BETWEEN AND 等谓词查询
- (9) 利用 LIKE 子句实现模糊查询
- (10) 利用 ORDER BY 子句为结果排序
- (11) 用 SQL Server/MySQL 的聚集函数进行统计计算
- (12) 用 GROUP BY 子句实现分组查询的方法

## 3、实验要求

- (1) 熟练掌握 SQL 的数据定义语句 CREATE、ALTER、DROP、Select
- (2) 写出实验报告

## 4、实验步骤

## 4.1 安装数据库

- (1) 安装数据库管理系统 DBMS;
- (2) 基于可视化界面或者命令行窗口创建数据库, 命名为 CSEDB\_学号;

## 4.2 基本表操作

设有一个学生-课程数据库, 包括学生关系 Student、课程关系 Course 和选修关系 SC:

学生表: Student(Sno, Sname, Ssex, Sage, Sdept, Scholarship)

课程表: Course(Cno, Cname, Cpno, Ccredit)

学生选课表: SC(Sno, Cno, Grade)

利用 SQLServer 的查询分析器或 MySQL 的查询编辑器进行如下操作, 不同的 DBMS 有少许差别, 如果存在错误, 需要根据错误提示自己排除

- (1) 创建、删除表, 例如:

CREATE TABLE Student

```
(Sno      CHAR(5) NOT NULL  UNIQUE,
 Sname    CHAR(20) UNIQUE,
 Ssex     CHAR(1),
 Sage     INT,
 Sdept    CHAR(15),
 Scholarship CHAR(2))
```

CREATE TABLE SC(

```
Sno CHAR(5),
Cno CHAR(3),
Grade int,
Primary key (Sno, Cno));
```

DROP TABLE Student

- (2) 查看、修改表的定义, 例如:

```
ALTER TABLE Student ADD Scome DATETIME
```

```
ALTER TABLE Student ALTER COLUMN Sage SMALLINT
```

### (3) 创建和删除索引

```
CREATE UNIQUE INDEX Stusno ON Student(Sno);
```

```
CREATE UNIQUE INDEX Coucno ON Course(Cno);
```

```
CREATE UNIQUE INDEX SChno ON SC(Sno ASC, Cno DESC);
```

```
DROP INDEX Stusno
```

## 4.3 删除数据库

### 4.4 创建（定义）示例数据库 S\_T\_学号

### 4.5 在数据库 S\_T\_学号中创建学生表 Student、课程表 Course 和选修表 SC

#### (1) 创建 3 个表

##### ① 利用 SQL 语句中的 Create Table 命令/或者可视化环境创建表

```
create table Student
(Sno CHAR(9) PRIMARY KEY,
 Sname CHAR(20) UNIQUE,
 Ssex CHAR(2),
 Sage SMALLINT,
 Sdept CHAR(20),
 Scholarship char(2)
);
go
/*表 Student 的主码为 Sno，属性列 Sname 取唯一值*/
create table Course
(Cno CHAR(4) PRIMARY KEY,
 Cname CHAR(40),
 Cpno CHAR(4),
 Ccredit SMALLINT,
 FOREIGN KEY (Cpno) REFERENCES Course(Cno)
);
go
/*表 Course 的主码为 Cno，属性列 Cpno(先修课)为外码，被参照表为 Course，被参照列是 Cno*/
create table SC
(Sno CHAR(9),
 Cno CHAR(4),
 Grade SMALLINT,
 primary key (Sno, Cno),
 FOREIGN KEY (Sno) REFERENCES Student(Sno),
 FOREIGN KEY (Cno) REFERENCES Course(Cno)
);
go
/*表 SC 的主码为(Sno, Cno), Sno 和 Cno 均为外码，被参照表分别为 Student 和 Course，被参照列分别为 Student.Sno 和 Course.Cno*/
```

#### (2) 在 3 个表中添加示例数据（任选一种数据添加方法）

##### 表 Student

学号 Sno	姓名 Sname	性别 Ssex	年龄 Sage	所在系 Sdept	奖学金 Scholarship
200215121	李勇	男	20	CS	否
200215122	刘晨	女	19	CS	否
200215123	王敏	女	18	MA	否
200215125	张立	男	19	IS	否

表 Course

课程号 Cno	课程名 Cname	现行课 Cpno	学分 Ccredit
1	数据库	5	4
2	数学		2
3	信息系统	5	4
4	操作系统	6	3
5	数据结构	7	4
6	数据处理		2
7	PASCAL 语言	6	4

表 SC

学号 Sno	课程号 Cno	成绩 Grade
200215121	1	92
200215121	2	85
200215121	3	88
200215122	2	90
200215122	3	80

## ① 用 SQL 语句中的更新语句（Insert 语句、Update 语句和 Delete 语句）往 3 个表输入示例数据。

```

use S_T; /*将 S_T 设为当前数据库*/
insert into student values('200215121','李勇','男',20,'CS', '否');
insert into student values('200215122','刘晨','女',19,'CS', '否');
insert into student values('200215123','王敏','女',18,'MA', '否');
insert into student values('200215125','张立','男',19,'IS', '否');
go
/*为表 Student 添加数据*/
insert into course values('1', '数据库', NULL,4);
insert into course values('2', '数学', NULL,2);
insert into course values('3', '信息系统', NULL,4);
insert into course values('4', '操作系统', NULL,3);
insert into course values('5', '数据结构', NULL,4);
insert into course values('6', '数据处理', NULL, 2);
insert into course values('7', 'java', NULL,4);
go
update Course set Cpno = '5' where Cno = '1';
update Course set Cpno = '1' where Cno = '3';
update Course set Cpno = '6' where Cno = '4';
update Course set Cpno = '7' where Cno = '5';
update Course set Cpno = '6' where Cno = '7';
/*为表 Course 添加数据*/
go
insert into SC values('200215121', '1',92);
insert into SC values('200215121', '2',85);
insert into SC values('200215121', '3',88);
insert into SC values('200215122', '2',90);
insert into SC values('200215122', '3',80);
/*为表 SC 添加数据*/
go

```

## ② 利用可视化环境交互式输入数据。

## 4.6 对学生关系 Student、课程关系 Course 和选修关系 SC 进行查询。

### 4.6.1 基本练习

(1) SELECT 语句的基本用法

例如：查询全体学生的详细记录。

```
SELECT Sno,Sname,Ssex,Sage,Sdept FROM Student
```

(2)使用 WHERE 子句进行有条件的查询

例如：查询选修 2 号课程且成绩在 90 分以上的所有学生的学号、姓名

```
SELECT Student.Sno, student.Sname
```

```
FROM Student, SC
```

```
WHERE Student.Sno = SC.Sno AND SC.Cno= '2' AND SC.Grade > 90
```

(3)使用 IN, NOT IN, BETWEEN 等谓词查询

例如：查询信息系（IS）、数学系（MA）和计算机科学系（CS）学生的姓名和性别。

```
SELECT Sname,Ssex
```

```
FROM Student
```

```
WHERE Sdept IN ( 'IS','MA','CS' )
```

例如：查询年龄在 20~23 岁（包括 20 岁和 23 岁）之间的学生的姓名、系别和年龄。

```
SELECT Sname,Sdept,Sage
```

```
FROM Student
```

```
WHERE Sage BETWEEN 20 AND 23
```

(4)利用 LIKE 子句实现模糊查询

例如：查询所有姓刘学生的姓名、学号和性别。

```
SELECT Sname,Sno,Ssex
```

```
FROM Student
```

```
WHERE Sname LIKE '刘%'
```

(5)利用 ORDER 子句为结果排序

例如：查询选修了 3 号课程的学生的学号及其成绩，查询结果按分数降序排列。

```
SELECT Sno,Grade
```

```
FROM SC
```

```
WHERE Cno= '3'
```

```
ORDER BY Grade DESC
```

(6)用 SQL Server 的统计函数进行统计计算

例如：计算 1 号课程的学生平均成绩。

```
SELECT AVG(Grade)
```

```
FROM SC
```

```
WHERE Cno= '1'
```

(7)用 GROUP BY 子句实现分组查询的方法

例如：查询选修了 3 门以上课程的学生学号。

```
SELECT Sno
```

```
FROM SC
```

```
GROUP BY Sno
```

HAVING COUNT(\*)>3

**4.4.2 扩展练习（要求写出并执行 SQL 语句来完成以下各种操作，记录查询结果）**

- (1) 查询全体学生的学号、姓名和年龄；
- (2) 查询所有计算机系学生的详细记录；
- (3) 找出考试成绩为优秀（90 分及以上）或不及格的学生的学号、课程号及成绩；
- (4) 查询年龄不在 19~20 岁之间的学生姓名、性别和年龄；
- (5) 查询数学系（MA）、信息系（IS）的学生的姓名和所在系；
- (6) 查询名称中包含“数据”的所有课程的课程号、课程名及其学分；
- (7) 找出所有没有选修课成绩的学生学号和课程号；
- (8) 查询学生 200215121 选修课的最高分、最低分以及平均成绩；
- (9) 查询选修了 2 号课程的学生的学号及其成绩，查询结果按成绩升序排列；
- (10) 查询每个系名及其学生的平均年龄。  
(思考：如何查询学生平均年龄在 19 岁以下（含 19 岁）的系别及其学生的平均年龄？)