### 1.数据库：

##### 数据库是以一定的组织方式将相关数据组织在一起存放在计算机外存储器上，并能为多个用户共享的与应用程序彼此独立的一组相关数据的集合。

##### 1.数据库是信息的基础

##### 2.信息通过数据处理产生

##### 3.信息用于反映数据的意义，用做决策

### 2.数据库管理系统(DBMS)的主要功能:

##### 1.数据定义对象

##### 2.数据操纵功能

##### 3.数据库的运行管理

##### 4.数据库的建立和维护

### 3.数据库系统（DBS）的特点：

##### 1.数据结构化

##### 2.数据的共享性高，冗余度低，易于扩充

##### 3.数据独立性强（物理独立性、逻辑独立性）

##### 4.数据由DBMS统一管理和控制

### 4.现实世界—>信息世界—>数据世界 P5：

##### 1.现实世界到信息世界是：概念模型（E->R图来表示）

##### 2.信息世界到数据世界是关系模型(逻辑模型)

### 5.实体间的联系:

##### 1.一对一联系

##### 2.一对多联系

##### 3.多对多联系

| 现实世界 | 信息世界 | 数据世界 |
| --- | --- | --- |
| 事物总体 | 实体集 | 文件 |
| 个体 | 实体 | 记录（元组）（行） |
| 特征 | 属性（列） | 字段（列） |
| 联系 | 概念模型【E-R图】 | 数据模型 |

1.由数据结构、数据操作、完整性约束组成

2.概念模型必须转换成逻辑数据模型才能在DBMS中实现。

3.关系表是元祖的集合

4.关系模式是关系模型中的一个表

5.属性是列，元组是行，域是一个属性的范围

### 6.E-R图表示：

##### 圆圈表示属性

##### 矩形表示实体

##### 菱形表示联系

### 7.数据库设计:

##### 1.用户需求分析

##### 2.概念结构设计

##### 3.逻辑结构设计和物理设计

### 8.E-R图关系模型的转换

##### 步骤：

实体中的各属性作为关系模式中的各属性

1:1将A/B主键和联系加入B或A

1:n将A(1)中的主键和联系属性加入B

m:n将A主键和B的外键共同构成新模式的主键联系属性

### 9.范式

1.第一范式：元组中的每一个分量都必须是不可分割的数据项

2.第二范式：在满足第一范式的基础上，所有非主属性完全依赖于其主关键字

3.第三范式：在满足第二范式的基础上，它的任何一个非主属性都不依赖于任何主关键字

若关系函数存在传递函数依赖，该关系最高为2NF

### 10.专门的关系运算 P11

1.选择（行）

2.投影（列）

3.连接（左连接、右连接、自然连接）

补：笛卡尔积 RxS行相乘，列相加





### 11.数据库系统的三级模式结构

外模式、模式（概念模式或逻辑模式）、内模式

### 12.二级映像

1.外模式/模式映像：保证了数据与程序的逻辑独立性

2.模式/内模式映像：保证了数据与程序的物理独立性

##### 三级模式和二级映像是保证数据独立性

### 13.数据库设计过程

需求分析->概念结构设计->逻辑结构设计->物理结构设计->数据库实施->数据库运行和维护

### P21页

（1）关系数据库中每个关系的形式是（二维表）。
（2）数据库技术研究在(计算机）环境下如何合理组织数据有效管理数据和高效处理数据。
（3）在实体间的联系中，学校和校长两个实体型之间存在(1:1）联系，而老师和同学两个实体型之间存在（n:m)联系。
（4）在关系数据模型中。二维表的列称为（属性/字段）二维表的行称为（元组/记录）。
（5）数据模型不仅表示反映事物本身的数据,而且表示事物之间的（联系）。
（6）从表中取出满足条件元组的操作称为（选择）

（7）把两个关系中相同属性值的元组连接到一起形成新的二维表的操作称为（连接）
（8）从表中抽取属性值满足条件列的操作称为（投影）。
（9）为了把多对多的联系分解成两个一对多联系所建立的“纽带表”中应包含（主键）。
（10) 用二维表数据来表示实体及实体之间联系的数据的数据模型为（关系模型）。

### 14.管理数据库

创建数据库：create database/schema [if not exists] 数据库名 ；

删除数据库：drop database if exists 数据库名称；

修改数据库：alter database 数据库名 选项；

P55

1. MySQL的系统数据库为：sys，mysql，information\_schema和performance\_schema。
2. MySQL数据库对象有表、视图、索引、触发器和储存过程等。
3. MySQL中使用（use）语句来打开或者切换至指定的数据库。
4. （数据库）是表、视图、存储过程，触发器对象的集合，是数据库管理系统的核心内容。

### 15.表的概念P57

1.表通常代表一个实体

2.表由行（元组/记录）和列(字段/属性)组成。

3.行值在同一个表中具有唯一性（不能有完全重复的一行）。

4.字段名在同一个表中具有唯一性。

5.行和列的无序性。

### 16.表P61-P71

##### 创建表：

CREATE TABLE IF NOT EXISTS 表名 (

 ID VARCHAR(20) PRIMARY KEY,
 userName VARCHAR(50) NOT NULL,
 userPassword VARCHAR(50) NOT NULL,
 firstTime VARCHAR(50) NOT NULL,
 idNumber VARCHAR(18) NOT NULL UNIQUE,
 accountBalance DECIMAL(15,2) NOT NULL DEFAULT 0.00,
 is\_admin BOOLEAN NOT NULL DEFAULT FALSE
);

##### 查看表： show tables；

##### 查看表结构：desc 表名；

##### 修改表结构：

1.添加字段： alter table 表名 add 字段名 字段类型 ；

2.修改字段数据类型：alter table 表名 modify 要修改的字段名 修改后的字段类型；

3.删除字段名：alter table 表名 drop column 要删除的字段名；

4.删除主键/外键约束: alter table 表名 drop primary/primary key;

例如:

（1）查看dept表的表结构。
desc dept;
（2）为dept表添加dp\_count字段,数据类型为varchar,长度为3,允许为空。
alter table dept add dp\_count varchar(3) ;
（3）修改dept表的dp\_count字段数据类型为int
alter table dept modify do\_count int;
（4）将dp\_count字段的位置调整到dp\_id字段的后面。
alter table dept modify dp\_count int after dp\_id;
（5）将dp\_count字段的字段名改为dp\_cnt。
alter table dept change dp\_count dp\_cnt int;
 （6）删除dept表的dp\_cnt字段。
alter table dept drop dp\_cnt;
 （7）再次查看dept表的表结构。
desc dept;
 （8）将dept表改名为department。
alter table dept rename department;
（9）将slt\_course表的字段id删除。
alter table slt\_course drop id;

##### 删除表： drop table 表名；

##### 插入数据：

1、insert into 表名 （字段名）values（‘ 字段值’）；

2、insert into 表名 values（’字段值‘）；

例如：

1、INSERT INTO bank\_user (ID, userName, userPassword, firstTime, idNumber, accountBalance, is\_admin)
VALUES ('admin', '系统管理员', 'admin123', NOW(), '000000000000000000', 0.00, TRUE);

2、INSERT INTO bank\_user VALUES ('admin', '系统管理员', 'admin123', NOW(), '000000000000000000', 0.00, TRUE);

##### 更新数据：

update 表名 set 字段名=字段值 where 条件的字段名=字段值；

（1）修改student表的记录,将马丽的入学成绩改为422。
update student set st\_score =422 where st\_id=1902105;

（2）将course表中所有课程的学时改为64,学分改为4。
update course set cs\_tm=64,cs\_sc=4;

（3）将slt\_course表中学号为1901001的学生的101号课程的成绩改为87。
update slt\_course set score=87 where st\_id=1901001 and cs\_id=101;

（4）修改管理系主任的联系电话为86677223。
update dept set dp\_tel='86677223' where dp\_nm='管理系';

##### 删除数据

delete from 表名 where 条件；

示例：

删除“张飞”的相关记录。
delete from student where st\_nm="张飞";

删除表中名称为zs的记录。
delete from employee where username="zs";

删除表中所有记录。
delete from employee;

删除某一种动物的信息 ,并查看视图
delete from ani\_view where kinds="哺乳动物";

### 17.数据定义语言与数据操作语言P71

1.数据定义语言：create 、alter 、drop

2.数据操作语言：update、delete、insert、select

### 18.数据完整性P74

1.实体完整性（唯一性约束、唯一性索引、主键约束）

2.域完整性（默认值、检查约束、外键约束、数据类型）

3.参照完整性（外键、检查约束、触发器、存储过程）

4.用户自定义完整性（检查约束、触发器、存储过程）

### P81页选择

（1）对一个已创建的表（D）操作是不可以的。
A.更改表名
B.增加或删除列
C.修改已有列的属性
D.将已有te xt数据类型修改为image数据类型
(2)SQL中创建基本表应使用(B）语句。
A. CREATE SCHEMA

B. CREATE TABLE

C. CREATE VIEW

D. CREATE DATEBASE
（3) SQL中删除表中数据的命令是（A）。
A. DELETE B. DROP C. CLEAR D. REMOVE
（4)下面关于数据库中表的行和列的叙述正确的是（B）
A.表中的行是有序的列是无序的

B.表中的列是有序的行是无序的

C.表中的行和列都是有序的

D.表中的行和列都是无序的
（5） SQL的数据操作语句包括 SELECT、INSERT 、UPDATE 和 DELETE 等。其中最重要的也是使用最频繁的语句是（ A）。
A. SELECT B.INSERT C.UPDATE D. DELETE
（6）在下列SQL语句中修改表结构的语句是（A）
A. ALTER B. CREATE C.UPDATE D.INSERT
（8）下列儿种情况下，不适合创建索引的是（A）。
A.列的取值范围很少 B.用作查询条件的列
C.频繁搜索范围的列 D.连接中频繁使用的列
(9) CREATE UNIQUE INDEX writer\_ index ON作者信息（作者编号）语句创建了一个（A）
A.唯一索引 B.全文索引 C.主键索引 D.普通索引
（10）建立索引的目的是（D）
A.降低MySQL数据检索的速度

B.与MySQL数据检索的速度无关

C.加快数据库的打开速度

D.提高MySQL数据检索的速度

(11)创建索引的命令是（D）。
A. CREATE TRIGGER

B. CREATE PROCEDURE

C. CREATE FUNCTION

D. CREATE INDEX
（12）索引只能创建在CHAR、 VARCHAR或者TEXT类型的字段上，称之为（C）

A.主键索引

B.唯一性索引

C.全文索引

D.哈希索引

（13）下面关于唯一性索引描述不正确的是(A）
A.某列创建了唯一性索引，则这一列为主键
B.不允许插入重复的列值
C.某列创建为主键,则该列会自动创建唯一性索引
D.一个表中可以有多个唯一性索引
（14）某数据表已经将列F定义为主关键字。则以下说法中错误的是（D）
A.列F的数据是有序排列的
B.列F的数据在整个数据表中是唯一存在的
C.不能再给数据表其他列建立主键
D.当为其他列建立普通索引时。将导致此数据表的记录重新排列
（15）以下关于数据库完整性描述不正确的是（A)。
A.数据应随时可以被更新。
B.表中的主键的值不能为空。
C.数据的取值应在有效范围内。
D.一个表的值若引用其他表的值，应使用外键进行关联。

（16)下面关于默认值的描述，正确的是（A）。

A.表中添加新列时，如果没有指明值，可以使用默认值。

B.可以绑定到表列，也可以绑定到数据类型。

C.可以响应特定事件的操作。

D.以上描述都正确。

（17）下列SQL语句中，能够实现实体完整性控制的语句是（B）。
A.FOREIGN KEY
B. PRIMARY KEY
C. REFERENCES
D. FOREIGN KEY 和 REFERENCES
（18)关于FOREIGN KEY约束的描述不正确的是（D）。
A.体现数据库中表之间的关系
B.实现参照完整性
C.以其他表PRIMARY KEY约束和 UNIQUE约束为前提
D.每个表中都必须定义

（19）下列SQL语句中，能够实现参照完整性控制的语句是（D）
A. FOREIGN KEY

B. PRIMARY KEY决
C. REFERENCES
D. FOREIGN KEY 和 REFERENCES
（20）限制输入到列的值的范围，应使用（A）约束。
A. CHECK
B. PRIMARY KEY
C. FOREIGN KEY
D. UNIQUE

### P83填空

1.完整性约束包括(实体)完整性、(参照)完整性、（默认）完整性、参照完整性和用户定义完整性P75
2.索引的类型主要有（普通索引）、（唯一性索引）、（主键索引）、（全文类索引）、（空间索引）和（哈希索引）。P72
3（域完整性）是指保证指定列的数据具有正确的数据类型,格式和有效的数据范围。
4（实体完整性）用于保证数据库中数据表的每一一个特定实体的记录都是唯一的。
5.创建修改和删除表命令分别是（create）table、（alter）table、（drop）table。
6.如果表的某一列被指定具有 NOT NULL属性，则表示（非空）。
7.建立和使用（约束）的目的是保证数据的完整性。
8.当在一个表中已存在PRIMARY KEY约束时，不能再创建（主键）索引。用CREATE IN DEX ID\_Index ON Students (身份证）建立的索引为(唯一）索引
9.创建唯一性索引时，应保证创建索引的列不包括重复的数据,并且没有两个或两个以上的空值。如果有这种数据，必须先将其 （删除） 否则索引不能成功创建。

10.两个表的主关键字和外关键字的数据对应一致，这是属于（参照）完整性，通常可以通过（主键）和（外键）来实现。

### 19.select查询语句

##### 单表查询：

查询student表中所有学生的信息。
select \* from student;

查询所有学生的学号、姓名和来源。
select st\_id as '学号' , st\_nm as '姓名', st\_from as '来源' from student;

查询所有学生的成绩信息，成绩以score*0.7显示出来。*
*select st\_id, st\_nm, st\_sex, st\_birth, st\_score*0.7, st\_date, st\_from, st\_dpid, st\_mnt from student;

使用as关键字为dept表中如下属性指定字段名：系号、系名、系主任、联系电话。
select dp\_id as '系号', dp\_nm as '系别', dp\_drt as '系主任', dp\_tel as "联系电话" from dept;

查询所有学生的学号、姓名、性别、班级（学号前两位为班级号）。
select st\_id "学号", st\_nm '姓名', st\_sex '性别', left(st\_id,2) '班级' from student;

查询所有男生的信息。
select \* from student where st\_sex="男";

查询学分为3的课程的信息。
select \* from course where cs\_sc=3;

查询来自四川达州和四川巴中的女生的信息。
select \* from student where st\_from="四川达州" or st\_from="四川巴中" and st\_sex="女";

##### like

查询“计算机”开头的课程的信息。
select \* from course where cs\_nm like '计算机%';

查询所有姓“张”的学生的信息。
select \* from student where st\_nm like '张%';

查询所有名字为三个字且姓“张”的学生的信息。
select \* from student where st\_nm like '张\_\_';

查询所有课程中包含“数据”的课程的信息。
select \* from course where cs\_nm like '%数据%';

询11系、12系、13系学生的信息。
select \* from student where st\_dpid in ('11','12','13');

查询所有姓“张”、姓“王”、姓“李”、姓“赵”的学生的信息。（答案不唯一)
select \* from student where st\_nm like ('张%')or st\_nm like ('李%') or st\_nm like ('王%') or st\_nm like('赵%');

##### between and

查询成绩为60~80的学生的学号。
select st\_id as "学号" from slt\_course where (score between 60 and 80);

查询选修了101课程，成绩为60~80的学生的学号。
select st\_id '学号' from slt\_course where cs\_id=101 and (score between 60 and 80);

查询在1999.7.1至2000.7.1出生的学生的信息。
select \* from student where st\_birth between '1999-7-1' and '2000-7-1';

##### order by

查询课程信息，按照课程名称降序显示。
select \* from course order by cs\_nm desc ;

查询选修了101课程的学生的学号，按成绩降序显示。
select st\_id '学号' from slt\_course where cs\_id=101 order by score desc ;

查询11系学生的学号、姓名、年龄，按照年龄升序显示。
select st\_id '学号', st\_nm '姓名',st\_birth '年龄' from student where st\_dpid=11 order by st\_birth desc ;

查询课程信息，按照课程的学分降序排列，再按照课程名称升序排列。
select \* from course order by cs\_sc desc ,cs\_nm asc ;

##### 多表查询

查询学生的学号、姓名、系别及所选课程编号。
select \* from student;
select \* from dept;
select \* from course;
select \* from slt\_course;
select s.st\_id '学号',s.st\_nm '姓名',s.st\_dpid '系部' , sc.cs\_id "学号" from student s join slt\_course sc on s.st\_id=sc.st\_id;

查询学生的学号、姓名、所选课程名称。
select s.st\_id '学号',s.st\_nm '姓名', sc.cs\_id "学号",c.cs\_nm "课程名称" from student s , slt\_course sc ,course c where s.st\_id=sc.st\_id and sc.cs\_id=c.cs\_id;

查询学生的学号 姓名 所选课程名称及成绩。
select s.st\_id ,s.st\_nm,sc.cs\_id, c.cs\_nm,sc.score from student s,slt\_course sc,course c where s.st\_id=sc.st\_id and sc.cs\_id=c.cs\_id;

查询学生的学号、姓名、所在系别的名称。
select s.st\_id '学号',s.st\_nm '姓名',d.dp\_nm '系部' from student s ,dept d where s.st\_dpid=d.dp\_id;

查询学生的学号、姓名、所在系别的名称及联系电话。
select s.st\_id,s.st\_nm,d.dp\_nm,d.dp\_tel from student s ,dept d where s.st\_dpid=d.dp\_id;

查询选修了101号课程的学生的学号、姓名、成绩。
select student.st\_id,st\_nm,score from student,slt\_course where cs\_id=101 and student.st\_id=slt\_course.st\_id;

查询选修了“数据结构”课程的学生的姓名、学号、成绩。
select student.st\_id,student.st\_nm,slt\_course.score from student,slt\_course,course where course.cs\_nm='数据结构' and student.st\_id =slt\_course.st\_id and course.cs\_id=slt\_course.cs\_id;

查询没有选修任何课程的学生的学号、姓名。
select student.st\_nm ,student.st\_id from student,slt\_course where slt\_course.cs\_id is null;

查询不及格的学生的姓名、学号。
select student.st\_nm, student.st\_id from student,slt\_course where slt\_course.score<60;

查询入学成绩在420分以上的学生的学号、姓名、所在系的名称及联系电话
select student.st\_id,student.st\_nm,dept.dp\_nm,dept.dp\_tel from student ,dept where student.st\_score>420 and student.st\_dpid=dept.dp\_id;

查询计科系学生的信息。
select \* from student where st\_dpid=11;

查询计科系学生的选修课成绩。
select student.st\_id,student.st\_nm,slt\_course.cs\_id,slt\_course.score from student,slt\_course where student.st\_id=slt\_course.st\_id and student.st\_dpid=11;

##### 子查询

查询计科系学生的信息。
select \* from student where st\_dpid = (select dp\_id from dept where dp\_nm = '计科系');

查询不及格的学生的姓名、学号。
select st\_nm, st\_id from student where st\_id = (select st\_id from slt\_course where score < 60);

查询选修了课程的学生的姓名、学号、性别。
select st\_id, st\_nm, st\_sex from student where st\_id in (select st\_id from slt\_course);

查询“张飞”同学所在班级的学生的姓名、学号。
select st\_nm, st\_id from student where st\_id = (select st\_id from student where st\_nm = '张飞');

查询101号课程选课成绩比所有科目的平均成绩高的学生的学号、姓名。
select slt\_course.st\_id, student.st\_nm from slt\_course,student where slt\_course.st\_id = student.st\_id and slt\_course.score > (select avg(score) from slt\_course where cs\_id = '101');

查询姓“张”的学生所在的系的名称和联系电话。
select dp\_nm, dp\_tel from dept where dp\_id in (select st\_dpid from student where st\_nm like "张%");

查询女生的选修课的平均成绩。
select avg(score) from slt\_course where st\_id in (select st\_id from student where st\_sex = "女");

查询班长的选修课成绩。
select cs\_id, score from slt\_course where st\_id = (select st\_id from student where st\_mnt = "班长");

查询班长的选修课名称、学分和成绩。
select course.cs\_nm, course.cs\_sc, student.st\_score from slt\_course,course,student where student.st\_id = slt\_course.st\_id and course.cs\_id = slt\_course.cs\_id and student.st\_mnt = "班长";

查询入学成绩在420分以上的学生所在系的名称及联系电话。
select dp\_nm, dp\_tel from dept where dp\_id in (select st\_dpid from student where st\_score > 420);

查询入学成绩最高的学生所在系的名称及联系电话。
select dp\_nm, dp\_tel from dept where dp\_id = (select st\_dpid from student where st\_score = (select max(st\_score) from student));

查询计科系学生选修课的平均成绩。
select avg(score) from slt\_course where st\_id in (select st\_id from student where st\_dpid = (select dp\_id from dept where dp\_nm = "计科系"));

查询student表中的所有记录的sname、ssex和class列。
select sname, ssex, class from student;

查询教师所在的单位,即查询不重复的depart列。
-- distinct 删除重复行 select distinct 字段名 from 表名；
select distinct depart from teacher;

查询student表的所有记录。
select \* from student;

查询score表中成绩为60~80的所有记录。
select \* from score where degree >= 60 && degree <= 80;

查询score表中成绩为85、86或88的记录。
select \* from score where degree = 85 or degree = 86 or degree = 88;

查询student表中95031班或性别为“女”的学生的记录。
select \* from student where class = 95031 or ssex = '女';

以class降序查询student表的所有记录。
-- 对查询结果排序 select \* from 表名 order by 字段名+[asc|desc]; 默认是升序
select \* from student order by class desc;

以cno升序、degree降序查询score表的所有记录。
select \* from score order by cno asc, degree desc;

查询95031班的学生人数。
select count(sname) as "班级人数" from student where class = 95031;

查询score表中最高分的学生学号和课程号。
select sno as "学号", sno as "课程号" from score where degree = (select max(degree) from score);

查询3\_105号课程的平均分。
select avg(degree) as "平均分", cno as "课程号" from score where score.cno = '3\_105';

查询score表中至少有5名学生选修并以3开头的课程的平均分数。
select avg(sno) '平均分数' from score group by cno having avg(sno) > 5;

查询最低分大于70日最高分小于90的sno列。
select sno, degree from score where degree > 70 and degree < 90;

查询所有学生的sname、cno和degree列。
select sname, cno, degree from student, score where student.sno = score.sno;

查询所有学生的sno、cname和degree列。
select distinct student.sno, course.cname, score.degree from student, score, course where student.sno = score.sno and score.cno = course.cno order by student.sno asc;

查询95033班所选课程的平均分。
select avg(score.degree) from student,score where student.sno = score.sno and student.class = "95033";

查询选修3\_105课程且成绩高于109号学生成绩的所有学生的记录。
select distinct student.\* from student, score where score.cno = '3\_105' and degree > (select degree from score where sno = '109' and cno = '3\_105');

查询score中选学一门以上课程的学生中分数为非最高分成绩的记录
select distinct \* from score where degree < (select max(degree) from score) group by cno having count(sno) > 1;

查询成绩高于学号为109、课程号为3\_105的课程成绩的所有记录。
select distinct score.\* from score,course where score.cno = '3\_105' and degree > (select degree from score where sno = '109' and cno = '3\_105');

查询和学号为108的学生同年出生的所有学生的sno、sname和sbirthday列。
set @birthday = (select sbirthday from student where sno = '108');
select sno, sname, sbirthday from student where year(sbirthday) = year(@birthday);

查询“张旭”教师任课的学生成绩。
select teacher.tname, course.*, score.* from teacher, score, course where teacher.tname = "张旭" and course.cno = score.cno;

查询选修某课程的学生人数多于5人的教师姓名。
select teacher.tname
from teacher,course where teacher.tno = course.cno group by cno having count(cno) > 5;

查询95033班和95031班全体学生的记录。
select \* from student where class = '95033' or class = '95031';

查询存在有85分以上成绩的课程cno。
select distinct score.cno from score where score.degree > 85 order by cno;

查询出“计算机系”教师所教课程的成绩表。
select degree, course.cname from teacher,score,course where teacher.depart = "计算机系";

查询“计算机系”与“电子工程系”不同职称的教师的tname和prof。
select distinct tname, prof from teacher where teacher.depart = "计算机系" and prof not in (select prof from teacher where depart = "电子工程系");

查询选修编号为3\_105课程且成绩高于选修编号为3\_245的学生的cno、sno和degree,并按degree列从高到低次序排序。
select cno, sno, degree from score where cno = '3\_105' and degree > (select max(degree) from score where cno = '3\_245') order by degree desc;

查询选修编号为3\_105且成绩高于选修编号为3\_245课程的学生的cno、sno和degree。
select cno, sno, degree from score where cno = '3\_105' and degree > (select max(degree) from score where cno = '3\_245');

查询所有教师和学生的name、sex和birthday。
select sname, ssex, sbirthday, tname, tsex, tbirthday from student,teacher;

查询所有“女”教师和“女”学生的name、sex和birthday
select tname, tsex, tbirthday from teacher where tsex = "女";

查询成绩比该课程平均成绩低的学生的成绩表
select distinct sc.\* from score sc where sc.degree<(select avg(ss.degree) from score ss where ss.cno=sc.cno)order by sc.cno;

查询所有任课教师的tname和depart。
select tname,depart from teacher join course using (tno);

查询所有未讲课的教师的tname和depart。
select tname,depart from teacher te where not exists(select sc.tno from course sc where sc.tno=te.tno);

查询至少有2名男生的班号。
select class from student where ssex='男' group by class having count(ssex)>1;

查询student表中不姓“王”的学生记录。
select student.\* from student where sname not like '王%';

查询student表中每个学生的姓名和年龄。
select sname,sbirthday,round(datediff (now(),sbirthday)/365)age from student;

查询student表中最大和最小的sbirthday日期值。
select min(st.sbirthday) ,max(st.sbirthday) from student st;

以班号和年龄从大到小的顺序查询student表中的全部记录。
select student.\* from student order by class desc,sbirthday asc;

查询“男”教师及其所上的课程。
select tname,cname from teacher join course using (tno) where tsex='男';

查询最高分学生的sno、cno和degree列。
select sno,cno,degree from score where degree=(select max(sc.degree) from score sc);

查询和“李军”同性别的所有学生的sname。
select sname from student where ssex=(select ssex from student where sname='李军');

查询所有选修“计算机导论”课程的“男”学生的成绩表。
select score.\* from score join (student,course) using(sno,cno) where ssex='男' and cname='计算机导论';

##### having 过滤分组后数据使用子句

与having一起使用 select 字段名,count(字段名) from 表名 group by字段名 having count(字段名)(判断条件)
[例如 按照性别对student分组，显示每个组中记录大于或等于2的数：
select sex,count(sex) from student group by sex having count(sex)<2;]
select cs\_id '课程号',avg(score) '平均分' from slt\_course group by cs\_id having avg(score)>75;

查询选修了两门以上（含两门）课程的学生的学号。
select st\_id '学号',count(cs\_id) '课程号' from slt\_course group by st\_id having count(cs\_id)>1;

查询score表中至少有5名学生选修并以3开头的课程的平均分数。
select avg(sno) '平均分数' from score group by cno having avg(sno) > 5;

查询score中选学一门以上课程的学生中分数为非最高分成绩的记录
select distinct \* from score where degree < (select max(degree) from score) group by cno having count(sno) > 1;

查询选修某课程的学生人数多于5人的教师姓名。
select teacher.tname
from teacher,course where teacher.tno = course.cno group by cno having count(cno) > 5;

对于emp表中低于自己工资至少5人的员工，列出其部门号、姓名、工资，以及工资少于自己的人数。
select a.deptno,a.ename,a.sal,(select count(b.ename) from emp b where b.sal < a.sal) num from emp a having num >= 5;

使用标准SQL嵌套语句查询选修全部课程的学员姓名和所属单位。
select sn,sd from s where sid in (select sid from sc group by sid having count(*)=(select count(*) from c));

查询选修课程超过2门的学员学号、姓名、单位。
select sid as '学号',sn as '姓名',sd as '单位' from s where sid in(select sid from sc group by sid having count(distinct cid)>2);

### P127

1.用select进行模式匹配时，可以使用like或者not like 匹配符，但要在条件值中使用（%）或（\_）等通配符来配合查询。并且模式匹配只能针对（字符）类型字段查询。

2.检索姓名字段中含有娟的表达式为:姓名like （%娟%）。
3.HAVING 子句与 WHERE 子句很相似，其区别在于: WHERE子句作用的对象是（表、视图），HAVING子句作用的对象是（组）。

4.视图是从（表、视图)中导出的表，数据库中实际存放的是视图的（定义），而不是（数据）。
5.当对视图进行UPDATE、 INSERT和 DELETE操作时，为了保证被操作的行满足视图定义中子查询语句的谓词条件,应在视图定义语句中使用可选择项（ with check option）
6.如果在视图中删除或修改一条记录，则相应的也随着视图更新

### P188

事物的特性：

1.原子性

2.一致性

3.隔离性

4.持久性