第13章_约束

讲师：尚硅谷-宋红康（江湖人称：康师傅）

官网： http://www.atguigu.com

基础练习：

练习1

已经存在数据库test04_emp，两张表emp2和dept2

```
CREATE DATABASE test04_emp;

use test04_emp;

CREATE TABLE emp2(
id INT,
emp_name VARCHAR(15)
);

CREATE TABLE dept2(
id INT,
dept_name VARCHAR(15)
);
```

题目：

```
#1.向表emp2的id列中添加PRIMARY KEY约束

#2. 向表dept2的id列中添加PRIMARY KEY约束

#3. 向表emp2中添加列dept_id，并在其中定义FOREIGN KEY约束，与之相关联的列是dept2表中的id列。
```

答案：

```
#1.向表emp2的id列中添加PRIMARY KEY约束

ALTER TABLE emp2 MODIFY COLUMN id INT PRIMARY KEY;
ALTER TABLE emp2 ADD PRIMARY KEY(id);

#2. 向表dept2的id列中添加PRIMARY KEY约束
ALTER TABLE dept2 MODIFY COLUMN id INT PRIMARY KEY;
ALTER TABLE dept2 ADD PRIMARY KEY(id);

#3. 向表emp2中添加列dept_id，并在其中定义FOREIGN KEY约束，与之相关联的列是dept2表中的id列。
ALTER TABLE emp2 ADD COLUMN dept_id INT;
ALTER TABLE emp2 ADD CONSTRAINT fk_emp2_deptid FOREIGN KEY(dept_id) REFERENCES
dept2(id);
```

练习2

承接《第11章_数据处理之增删改》的综合案例。

```
# 1、创建数据库test01_library

# 2、创建表 books，表结构如下：
```

字段名	字段说明	数据类型
id	书编号	INT
name	书名	VARCHAR(50)
authors	作者	VARCHAR(100)
price	价格	FLOAT
pubdate	出版日期	YEAR
note	说明	VARCHAR(100)
num	库存	INT

```
# 3、使用ALTER语句给books按如下要求增加相应的约束
```

字段名	字段说明	数据类型	主键	外键	非空	唯一	自增
id	书编号	INT(11)	是	否	是	是	是
name	书名	VARCHAR(50)	否	否	是	否	否
authors	作者	VARCHAR(100)	否	否	是	否	否
price	价格	FLOAT	否	否	是	否	否
pubdate	出版日期	YEAR	否	否	是	否	否
note	说明	VARCHAR(100)	否	否	否	否	否
num	库存	INT(11)	否	否	是	否	否

答案：

```
# 1、2、略

# 3、使用ALTER语句给books按如下要求增加相应的约束
#给id增加主键约束
ALTER TABLE books ADD PRIMARY KEY(id);

#给id字段增加自增约束
ALTER TABLE books MODIFY id INT AUTO_INCREMENT;

#给name等字段增加非空约束
ALTER TABLE books name VARCHAR(50) NOT NULL;
ALTER TABLE books `authors` VARCHAR(100) NOT NULL;
ALTER TABLE books price FLOAT NOT NULL;
ALTER TABLE books pubdate DATE NOT NULL;
```

```
ALTER TABLE books num INT NOT NULL;
```

练习3

题目：

```
#1. 创建数据库test04_company

#2. 按照下表给出的表结构在test04_company数据库中创建两个数据表offices和employees
```

- offices表：

字段名	数据类型	主键	外键	非空	唯一	自增
officeCode	INT(10)	是	否	是	是	否
city	VARCHAR(50)	否	否	是	否	否
address	VARCHAR(50)	否	否	否	否	否
country	VARCHAR(50)	否	否	是	否	否
postalCode	VARCHAR(15)	否	否	否	是	否

- employees表：

字段名	数据类型	主键	外键	非空	唯一	自增
employeeNumber	INT(11)	是	否	是	是	是
lastName	VARCHAR(50)	否	否	是	否	否
firstName	VARCHAR(50)	否	否	是	否	否
mobile	VARCHAR(25)	否	否	否	是	否
officeCode	INT(10)	否	是	是	否	否
jobTitle	VARCHAR(50)	否	否	是	否	否
birth	DATETIME	否	否	是	否	否
note	VARCHAR(255)	否	否	否	否	否
sex	VARCHAR(5)	否	否	否	否	否

```
#3. 将表employees的mobile字段修改到officeCode字段后面

#4. 将表employees的birth字段改名为employee_birth

#5. 修改sex字段，数据类型为CHAR(1)，非空约束

#6. 删除字段note

#7. 增加字段名favoriate_activity，数据类型为VARCHAR(100)

#8. 将表employees名称修改为employees_info
```

答案：

```
#1. 创建数据库test04_company
CREATE DATABASE test04_company;

#2. 按照下表给出的表结构在test04_company数据库中创建两个数据表offices和employees
USE test04_company;

CREATE TABLE offices(
officeCode INT(10) ,
city VARCHAR(50) NOT NULL,
```

```
address VARCHAR(50),
country VARCHAR(50) NOT NULL,
postalCode VARCHAR(15) UNIQUE,
PRIMARY KEY(officeCode)
);

CREATE TABLE employees(
employeeNumber INT(11) PRIMARY KEY AUTO_INCREMENT,
lastName VARCHAR(50) NOT NULL,
firstName VARCHAR(50) NOT NULL,
mobile VARCHAR(25) UNIQUE,
officeCode INT(10) NOT NULL,
jobTitle VARCHAR(50) NOT NULL,
birth DATETIME NOT NULL,
note VARCHAR(255),
sex VARCHAR(5),
CONSTRAINT fk_emp_ofCode FOREIGN KEY(officeCode) REFERENCES offices(officeCode)
);

#3. 将表employees的mobile字段修改到officeCode字段后面
ALTER TABLE employees MODIFY mobile VARCHAR(25) AFTER officeCode;

#4. 将表employees的birth字段改名为employee_birth
ALTER TABLE employees CHANGE birth employee_birth DATETIME;

#5. 修改sex字段，数据类型为CHAR(1)，非空约束
ALTER TABLE employees MODIFY sex CHAR(1) NOT NULL;

#6. 删除字段note
ALTER TABLE employees DROP COLUMN note;

#7. 增加字段名favoriate_activity，数据类型为VARCHAR(100)
ALTER TABLE employees ADD favoriate_activity VARCHAR(100);

#8. 将表employees名称修改为employees_info
ALTER TABLE employees RENAME employees_info;
```

拓展练习：

练习1

创建数据库test04_Market，在test04_Market中创建数据表customers。customers表结构如下所示，按以下要求进行操作。

字段名	数据类型	主键	外键	非空	唯一	自增
c_num	INT(11)	是	否	是	是	是
c_name	VARCHAR(50)	否	否	否	否	否
c_contact	VARCHAR(50)	否	否	否	否	否
c_city	VARCHAR(50)	否	否	否	否	否
c_birth	DATETIME	否	否	是	否	否

（1）创建数据库test04_Market。 （2）创建数据表customers，在c_num字段上添加主键约束和自增约束，在c_birth字段上添加非空约束。 （3）将c_contact字段插入c_birth字段后面。 （4）将c_name字段数据类型改为VARCHAR(70)。 （5）将c_contact字段改名为c_phone。 （6）增加c_gender字段，数据类型为CHAR(1)。 （7）将表名修改为customers_info。 （8）删除字段c_city。

在test04_Market中创建数据表orders。orders表结构如下所示，按以下要求进行操作。

字段名	数据类型	主键	外键	非空	唯一	自增
o_num	INT(11)	是	否	是	是	是
o_date	DATE	否	否	否	否	否
c_id	INT(11)	否	是	否	否	否

（1）创建数据表orders，在o_num字段上添加主键约束和自增约束，在c_id字段上添加外键约束，关联customers表中的主键c_num。 （2）删除orders表的外键约束，然后删除表customers。

练习2

创建数据表pet，并对表进行插入、更新与删除操作。pet表结构如下表所示。 （1）首先创建数据表pet，使用不同的方法将表记录插入到pet表中。 （2）使用UPDATE语句将名称为Fang的狗的主人改为Kevin。 （3）将没有主人的宠物的owner字段值都改为Duck。 （4）删除已经死亡的宠物记录。 （5）删除所有表中的记录。

pet表结构：

字段名	字段说明	数据类型	主键	外键	非空	唯一	自增
name	宠物名称	VARCHAR(20)	否	否	是	否	否
owner	宠物主人	VARCHAR(20)	否	否	否	否	否
species	种类	VARCHAR(20)	否	否	是	否	否
sex	性别	CHAR(1)	否	否	是	否	否
birth	出生日期	YEAR	否	否	是	否	否
death	死亡日期	YEAR	否	否	否	否	否

pet表中记录：

name	owner	species	sex	birth	death
Fluffy	Harold	cat	f	2003	2010
Claws	Gwen	cat	m	2004	NULL
Buffy	NULL	dog	f	2009	NULL
Fang	Benny	dog	m	2000	NULL
Bowser	Diane	dog	m	2003	2009
Chirpy	NULL	bird	f	2008	NULL

练习3

1、创建数据库：test_company

2、在此数据库下创建如下3表，数据类型，宽度，是否为空根据实际情况自己定义。

A. `部门表（department）`：部门编号（depid），部门名称（depname），部门简介（deinfo）；其中部门编号为主键。

B. `雇员表（emoloyee）`：雇员编号（empid），姓名（name），性别（sex），职称（title），出生日期（birthday），所在部门编号（depid）；其中

- 雇员编号为主键；
- 部门编号为外键，外键约束等级为（on update cascade 和on delete set null）；

- 性别默认为男；

C. `工资表（salary）`：雇员编号（empid），基本工资（basesalary），职务工资（titlesalary），扣除（deduction）。其中雇员编号为主键。

3、给工资表（salary）的雇员编号（empid）增加外键约束，外键约束等级为（on update cascade 和on delete cascade）

4、添加数据如下：

部门表：

部门编号	部门名称	部门简介
111	生产部	Null
222	销售部	Null
333	人事部	人力资源管理

雇员表：

雇员编号	姓名	性别	职称	出生日期	所在部门编号
1001	张三	男	高级工程师	1975-1-1	111
1002	李四	女	助工	1985-1-1	111
1003	王五	男	工程师	1978-11-11	222
1004	张六	男	工程师	1999-1-1	222

工资表：

雇员编号	基本工资	职务工资	扣除
1001	2200	1100	200
1002	1200	200	NULL
1003	2900	700	200
1004	1950	700	150

5、查询出每个雇员的雇员编号，姓名，职称，所在部门名称，应发工资（基本工资+职务工资），实发工资（基本工资+职务工资-扣除）。

6、查询销售部门的雇员姓名及其基本工资

7、查询姓“张”且年龄小于40的员工的全部信息和年龄

8、查询所有男员工的基本工资和职务工资

9、查询基本工资低于2000的员工姓名和职称、所在部门名称

10、查询员工总数

11、查询部门总数

12、查询应发工资的平均工资和最高工资、最低工资

13、按照部门统计应发工资的平均工资

14、找出部门基本工资的平均工资低于2000的

15、按照员工编号、姓名、基本工资、职务工资、扣除，并按照职务升序排列，如果职务工资相同，再按照基本工资升序排列

16、查询员工编号、姓名，出生日期，及年龄段。其中，如果80年之前出生的，定为”老年“；80后定为”中年“，90后定为”青壮年“

17、查询所有的员工信息，和他所在的部门名称

18、查询所有部门信息，和该部门的员工信息

19、查询所有职位中含“工程师”的男员工的人数

20、查询每个部门的男生和女生的人数和平均基本工资

```
#创建数据库：test_company
CREATE DATABASE test_company;

#使用数据库test_company
USE test_company;

#创建部门表（department）
CREATE TABLE department(
    depid INT PRIMARY KEY,
    depname VARCHAR(20) NOT NULL,
    deinfo VARCHAR(50)
);

#创建雇员表（emoloyee）
CREATE TABLE employee(
    empid INT PRIMARY KEY,
    `name` VARCHAR(20) NOT NULL,
    sex CHAR NOT NULL DEFAULT '男',
    title VARCHAR(20) NOT NULL,
    birthday DATE,
    depid INT,
    FOREIGN KEY(depid) REFERENCES department(depid) ON UPDATE CASCADE ON DELETE SET
NULL
);

#创建工资表（salary）
CREATE TABLE salary(
    empid INT PRIMARY KEY,
    basesalary DOUBLE,
    titlesalary DOUBLE,
    deduction DOUBLE
);

#给工资表（salary）的雇员编号（empid）增加外键约束，外键约束等级为（on update cascade 和on
delete cascade）
ALTER TABLE salary ADD FOREIGN KEY empid REFERENCES employee(empid) ON UPDATE CASCADE
ON DELETE CASCADE;

#添加部门表数据
INSERT INTO department VALUES
(111,'生产部',NULL),
(222,'销售部',NULL),
```

```
(333,'人事部','人力资源管理');

#添加雇员表
INSERT INTO employee VALUES
(1001,'张三',DEFAULT,'高级工程师','1975-1-1',111),
(1002,'李四','女','助工','1985-1-1',111),
(1003,'王五','男','工程师','1978-11-11',222),
(1004,'张六',DEFAULT,'工程师','1999-1-1',222);

#添加工资表
INSERT INTO salary VALUES
(1001,2200,1100,200),
(1002,1200,200,NULL),
(1003,2900,700,200),
(1004,1950,700,150);


/*
查询出每个雇员的雇员编号，姓名，职称，所在部门名称，
应发工资（基本工资+职务工资），
实发工资（基本工资+职务工资-扣除）。
*/
SELECT employee.empid,`name`,title,depname,
basesalary+titlesalary AS "应发工资",
basesalary+titlesalary-IFNULL(deduction,0) AS "实发工资"
FROM department INNER JOIN employee INNER JOIN salary
ON department.depid = employee.depid AND employee.empid = salary.empid;

#查询销售部门的雇员姓名及其基本工资
SELECT `name`,basesalary
FROM department INNER JOIN employee INNER JOIN salary
ON department.depid = employee.depid AND employee.empid = salary.empid
WHERE department.depname = '销售部';

#查询姓“张”且年龄小于40的员工的全部信息和年龄
SELECT *,YEAR(CURRENT_DATE())-YEAR(birthday) AS "年龄"
FROM employee
WHERE `name` LIKE '张%' AND YEAR(CURRENT_DATE())-YEAR(birthday)<40;

#查询所有男员工的基本工资和职务工资
SELECT basesalary,titlesalary
FROM employee INNER JOIN salary
ON employee.empid = salary.empid
WHERE employee.sex = '男';

#查询基本工资低于2000的员工姓名和职称、所在部门名称
SELECT `name`,title,depname
FROM department INNER JOIN employee INNER JOIN salary
ON department.depid = employee.depid AND employee.empid = salary.empid
WHERE basesalary < 2000;

#查询员工总数
SELECT COUNT(*) FROM employee;

#查询部门总数
SELECT COUNT(*) FROM department;

#查询应发工资的平均工资和最高应发工资、最低应发工资
```

```sql
SELECT AVG(basesalary+titlesalary) AS "平均应发工资",
    MAX(basesalary+titlesalary) AS "最高应发工资",
    MIN(basesalary+titlesalary) AS "最低应发工资"
FROM salary;

#按照部门统计应发工资的平均工资
SELECT depid,AVG(basesalary+titlesalary)
FROM employee INNER JOIN salary
ON employee.`empid` = salary.`empid`
GROUP BY employee.`depid`;

#找出部门基本工资的平均工资低于2000的
SELECT depid,AVG(basesalary)
FROM employee INNER JOIN salary
ON employee.`empid` = salary.`empid`
GROUP BY employee.`depid`
HAVING AVG(basesalary)<2000;


#按照员工编号、姓名、基本工资、职务工资、扣除，
#并按照职务升序排列，如果职务工资相同，再按照基本工资升序排列
SELECT emp.empid,`name`,basesalary,titlesalary,deduction
FROM employee emp INNER JOIN salary
ON emp.`empid` = salary.`empid`
ORDER BY emp.`title` ASC , basesalary ASC;

#查询员工编号、姓名，出生日期，及年龄段，其中
·#如果80年之前出生的，定为”老年“；80后定为”中年“，90后定为”青壮年“
SELECT empid,`name`,birthday,
 CASE WHEN YEAR(birthday)<1980 THEN '老年'
      WHEN YEAR(birthday)<1990 THEN '中年'
 ELSE '青壮年' END "年龄段"
FROM employee;

#查询所有的员工信息，和他所在的部门名称
SELECT emp.*,depname
FROM employee emp LEFT JOIN department dep
ON emp.`depid` = dep.`depid`;

#查询所有部门信息，和该部门的员工信息
SELECT dep.*,emp.*
FROM employee emp RIGHT JOIN department dep
ON emp.`depid` = dep.`depid`;


#查询所有职位中含“工程师”的男员工的人数
SELECT COUNT(*) FROM employee WHERE sex='男' AND title LIKE '%工程师%';

#查询每个部门的男生和女生的人数和平均基本工资
SELECT dep.depid,sex,COUNT(*),AVG(basesalary)
FROM department dep INNER JOIN employee INNER JOIN salary
ON dep.depid = employee.depid AND employee.empid = salary.empid
GROUP BY dep.depid,sex;
```

练习4

1、创建一个数据库：test_school

2、创建如下表格

表1 Department表的定义

字段名	字段描述	数据类型	主键	外键	非空	唯一
DepNo	部门号	int(10)	是	否	是	是
DepName	部门名称	varchar(20)	否	否	是	否
DepNote	部门备注	Varchar(50)	否	否	否	否

表2 Teacher表的定义

字段名	字段描述	数据类型	主键	外键	非空	唯一
Number	教工号	int	是	否	是	是
Name	姓名	varchar(30)	否	否	是	否
Sex	性别	varchar(4)	否	否	否	否
Birth	出生日期	date	否	否	否	否
DepNo	部门号	int	否	是	否	否
Salary	工资	float	否	否	否	否
Address	家庭住址	varchar(100)	否	否	否	否

3、添加记录

DepNo	DepName	DepNote
601	软件技术系	软件技术等专业
602	网络技术系	多媒体技术等专业
603	艺术设计系	广告艺术设计等专业
604	管理工程系	连锁经营管理等专业

Number	Name	Sex	Birth	DepNo	Salary	Address
2001	Tom	女	1970-01-10	602	4500	四川省绵阳市
2002	Lucy	男	1983-12-18	601	2500	北京市昌平区
2003	Mike	男	1990-06-01	604	1500	重庆市渝中区
2004	James	女	1980-10-20	602	3500	四川省成都市
2005	Jack	男	1975-05-30	603	1200	重庆市南岸区

4、用SELECT语句查询Teacher表的所有记录。

5、找出所有其家庭地址中含有“北京”的教师的教工号及部门名称，要求显示结果中各列标题用中文别名表示。

6、获得Teacher表中工资最高的教工号和姓名。

7、找出所有收入在2500～4000之间的教工号。

8、查找在网络技术系工作的教师的姓名、性别和工资。

```
#创建一个数据库：test_school
CREATE DATABASE test_school;

#使用数据库
USE test_school;

#创建表格
-- 部门信息表Department
CREATE TABLE Department(
    DepNo INT(10) PRIMARY KEY,
    DepName VARCHAR(20) NOT NULL,
    DepNote VARCHAR(50)
);
-- 创建数据表Teacher
CREATE TABLE Teacher(
    Number INT PRIMARY KEY,
    `Name` VARCHAR(30) UNIQUE,
    Sex VARCHAR(4),
    Birth DATE,
    DepNo INT,
    Salary FLOAT,
    Address VARCHAR(100),
    FOREIGN KEY (DepNo) REFERENCES Department(DepNo)
);
-- 将表4的内容插入Department表中
INSERT INTO Department VALUES (601,'软件技术系','软件技术等专业');
INSERT INTO Department VALUES (602,'网络技术系','多媒体技术等专业');
INSERT INTO Department VALUES (603,'艺术设计系','广告艺术设计等专业');
INSERT INTO Department VALUES (604,'管理工程系','连锁经营管理等专业');
-- 将表3的内容插入Teacher表中。
INSERT INTO Teacher VALUES(2001,'Tom','女','1970-01-10',602,4500,'四川省绵阳市');
INSERT INTO Teacher VALUES(2002,'Lucy','男','1983-12-18',601,2500,'北京市昌平区');
INSERT INTO Teacher VALUES(2003,'Mike','男','1990-06-01',604,1500,'重庆市渝中区');
INSERT INTO Teacher VALUES(2004,'James','女','1980-10-20',602,3500,'四川省成都市');
INSERT INTO Teacher VALUES(2005,'Jack','男','1975-05-30',603,1200,'重庆市南岸区');

#用SELECT语句查询Teacher表的所有记录。
SELECT * FROM teacher;

#找出所有其家庭地址中含有“北京”的教师的教工号及部门名称，要求显示结果中各列标题用中文表示。
SELECT number AS 教工号,Teacher.depno AS 部门名称
FROM Teacher INNER JOIN Department
ON Teacher.DepNo = Department.DepNo
WHERE address LIKE '%北京%';

#获得Teacher表中工资最高的教工号和姓名。
SELECT number,`name` FROM teacher WHERE salary = (SELECT MAX(salary) FROM teacher);
SELECT number,`name` FROM teacher ORDER BY salary DESC LIMIT 0,1;
```

```
#找出所有收入在2500～4000之间的教工号。
SELECT number FROM teacher WHERE salary BETWEEN 2500 AND 4000;

#查找在网络技术系工作的教师的姓名、性别和工资。
SELECT `name`,sex,salary FROM teacher
WHERE depno=(SELECT depno FROM department WHERE depname='网络技术系');

SELECT `name`,sex,salary
FROM teacher INNER JOIN department
ON teacher.depno = department.depno
WHERE depname ='网络技术系';
```

练习5

1、建立数据库test_student

2、建立以下三张表，并插入记录

Table:Classes

专业	班级	姓名	姓别	座位
计算机网络	1班	张三	男	8
软件工程	2班	李四	男	12
计算机维护	1班	王五	男	9
计算机网络	2班	LILY	女	15
软件工程	1班	小强	男	20
计算机维护	1班	CoCo	女	18

Table:Score

姓名	英语	数学	语文
张三	65	75	98
李四	87	45	86
王五	98	85	65
LILY	75	86	87
小强	85	60	58
CoCo	96	87	70

Table: Records

姓名	记录
小强	迟到
小强	事假
李四	旷课
李四	旷课
李四	迟到
CoCo	病假
LILY	事假

3、写出将张三的语文成绩修改为88的SQL语句。

4、搜索出计算机维护1班各门课程的平均成绩。

5、搜索科目有不及格的人的名单。

6、查询记录2次以上的学生的姓名和各科成绩。

```
#1、建立数据库test_student
CREATE DATABASE test_student;

#使用数据库
USE test_student;

#2、创建表格并添加记录
CREATE TABLE Classes(
    Pro_name VARCHAR(20) NOT NULL,
    Grade VARCHAR(10) NOT NULL,
    `name` VARCHAR(10) NOT NULL,
    sex VARCHAR(4) NOT NULL,
    seat INT(10) NOT NULL UNIQUE
);
CREATE TABLE Score(
    `name` VARCHAR(10) NOT NULL,
    En_score INT(10) NOT NULL,
    Ma_score INT(10) NOT NULL,
    Ch_score INT(10) NOT NULL
);
CREATE TABLE Records(
    `name` VARCHAR(10) NOT NULL,
    record VARCHAR(10)
);

-- 向classes中添加数据
INSERT INTO classes VALUES('计算机网络','1班','张三','男',8);
INSERT INTO classes VALUES('软件工程','2班','李四','男',12);
INSERT INTO classes VALUES('计算机维护','1班','王五','男',9);
INSERT INTO classes VALUES('计算机网络','2班','LILY','女',15);
INSERT INTO classes VALUES('软件工程','1班','小强','男',20);
INSERT INTO classes VALUES('计算机维护','1班','CoCo','女',18);

-- 向score中添加数据
INSERT INTO Score VALUES('张三',65,75,98);
```

```
INSERT INTO Score VALUES('李四',87,45,86);
INSERT INTO Score VALUES('王五',98,85,65);
INSERT INTO Score VALUES('LILY',75,86,87);
INSERT INTO Score VALUES('小强',85,60,58);
INSERT INTO Score VALUES('CoCo',96,87,70);

-- 向records中添加数据
INSERT INTO records VALUES('小强','迟到');
INSERT INTO records VALUES('小强','事假');
INSERT INTO records VALUES('李四','旷课');
INSERT INTO records VALUES('李四','旷课');
INSERT INTO records VALUES('李四','迟到');
INSERT INTO records VALUES('CoCo','病假');
INSERT INTO records VALUES('LILY','事假');

#3、写出将张三的语文成绩修改为88的SQL语句。
UPDATE score SET ch_score=88 WHERE `name`='张三';

#4、搜索出计算机维护1班各门课程的平均成绩。
SELECT AVG(en_score),AVG(ma_score),AVG(ch_score) FROM score
WHERE `name` IN (SELECT `name` FROM classes WHERE Pro_name='计算机维护' AND grade='1
班');

#5、搜索科目有不及格的人的名单。
SELECT `name` FROM score WHERE en_score<60 OR ma_score<60 OR ch_score<60;

#6、查询记录2次以上的学生的姓名和各科成绩。
SELECT *
FROM score INNER JOIN
(SELECT `name`,COUNT(*) FROM Records GROUP BY `name` HAVING COUNT(*)>2) temp
ON score.name = temp.name;
```

练习6

1、建立数据库：test_xuankedb

2、建立如下三张表：

学生表Student由学号(Sno)、姓名(Sname)、性别(Ssex)、年龄(Sage)、所在系(Sdept)五个字段，Sno 为关键字。

课程表Course由课程号(Cno)、课程名(Cname)、选修课号(Cpno)、学分(Ccredit)四个字段，Cno为关键字。

成绩表SG由学号(Sno)、课程号(Cno)、成绩(Grade)三个字段，(SNO, CNO)为关键字。

3、向Student表增加“入学时间(Scome)”列，其数据类型为日期型。

4、查询选修了3号课程的学生的学号及其成绩，查询结果按分数的降序排列。

5、查询学习1号课程的学生最高分数、平均成绩。

6、查询与“李洋”在同一个系学习的学生。

7、将计算机系全体学生的成绩置零。

8、删除学生表中学号为05019的学生记录。

9、删除计算机系所有学生的成绩记录。

```sql
-- 1、创建一个数据库：test_xuankedb
CREATE DATABASE test_xuankedb;

-- 使用数据库
USE test_xuankedb;

-- 2、创建学生表
CREATE TABLE student(
    sno INT(10) PRIMARY KEY,
    sname VARCHAR(10),
    ssex VARCHAR(10),
    sage INT(10),
    sdept VARCHAR(40)
);

-- 创建课程表
CREATE TABLE course(
    cno INT(10) PRIMARY KEY,
    cname VARCHAR(20),
    cpno VARCHAR(40),
    ccredit INT(20)
);

-- 创建成绩表
CREATE TABLE sg(
    sno INT(10),
    cno INT(10),
    grade INT(3),
    PRIMARY KEY(sno,cno),
    CONSTRAINT stu_s_sno_fk FOREIGN KEY (sno) REFERENCES student(sno),
    CONSTRAINT cou_s_sno_fk FOREIGN KEY (cno) REFERENCES course(cno)
);

#3、向Student表增加“入学时间(Scome)”列，其数据类型为日期型。
ALTER TABLE student ADD COLUMN scome DATE;

#4、查询选修了3号课程的学生的学号及其成绩，查询结果按分数的降序排列。
SELECT sno,grade FROM sg WHERE cno=3 ORDER BY grade DESC;

#5、查询学习1号课程的学生最高分数、平均成绩。
SELECT MAX(grade),AVG(grade) FROM sg WHERE cno=1;

#6、查询与“李洋”在同一个系学习的学生。
SELECT * FROM student WHERE sdept=(SELECT sdept FROM student WHERE sname='李洋');

#7、将计算机系全体学生的成绩置零。
UPDATE sg SET grade=0 WHERE sno IN (SELECT sno FROM student WHERE sdept='计算机系')

#8、删除学生表中学号为05019的学生记录。
DELETE FROM student WHERE sno=05019;

#9、删除计算机系所有学生的成绩记录。
DELETE FROM sg WHERE sno IN (SELECT sno FROM student WHERE sdept='计算机系');
```

练习7

1、建立数据库：test_library

2、建立如下三个表： 表一：press 出版社 属性：编号pressid(int)、名称pressname(varchar)、地址address(varchar)

表二：sort 种类 属性：编号sortno(int)、数量scount(int)

表二：book图书 属性：编号bid(int)、名称 bname(varchar)、种类bsortno(int)、出版社编号pressid(int)

3、给sort表中添加一列属性：描述describes(varchar)

4、向三个表中各插入几条数据

```
mysql> select * from press;
+---------+------------+---------+
| pressid | pressname  | address |
+---------+------------+---------+
|     100 | 外研社     | 上海    |
|     101 | 北大出版社 | 北京    |
|     102 | 教育出版社 | 北京    |
+---------+------------+---------+
```

```
mysql> select * from sort;
+--------+--------+-----------+
| sortno | scount | describes |
+--------+--------+-----------+
|     11 |     50 | 小说      |
|     12 |    300 | 科幻      |
|     13 |    100 | 神话      |
+--------+--------+-----------+
```

```
mysql> select * from book;
+------+----------+---------+---------+
| bid  | bname    | bsortno | pressid |
+------+----------+---------+---------+
|    1 | 红与黑   |      11 |     100 |
|    2 | 幻城     |      12 |     102 |
|    3 | 希腊神话 |      13 |     102 |
+------+----------+---------+---------+
```

5、查询出版社id为100的书的全部信息

6、查询出版社为外研社的书的全部信息

7、查询图书数量（scount）大于100的种类

8、查询图书种类最多的出版社信息

```sql
-- 1、建立数据库：test_library
CREATE DATABASE test_library;

-- 使用数据库
USE test_library;

-- 2、创建出版社表
CREATE TABLE press(
    pressid INT(10) PRIMARY KEY,
    pressname VARCHAR(30),
    address VARCHAR(50)
);
```

```sql
-- 创建一个种类表
CREATE TABLE sort(
    sortno INT(10) PRIMARY KEY,
    scount INT(10)
);

-- 创建图书表
CREATE TABLE book(
    bid INT(10) PRIMARY KEY,
    bname VARCHAR(40),
    bsortno INT(10),
    pressid INT(10),
    CONSTRAINT p_b_pid_fk FOREIGN KEY (pressid) REFERENCES press(pressid),
    CONSTRAINT s_b_sno_fk FOREIGN KEY (bsortno) REFERENCES sort(sortno)
);

-- 3、添加一列属性
ALTER TABLE sort ADD COLUMN describes VARCHAR(30);

-- 4、添加数据
INSERT INTO press VALUES(100,'外研社','上海');
INSERT INTO press VALUES(101,'北大出版社','北京');
INSERT INTO press VALUES(102,'教育出版社','北京');

-- 添加数据
INSERT INTO sort(sortno,scount,describes)VALUES(11,50,'小说');
INSERT INTO sort(sortno,scount,describes)VALUES(12,300,'科幻');
INSERT INTO sort(sortno,scount,describes)VALUES(13,100,'神话');

-- 添加数据
INSERT INTO book VALUES(1,'红与黑',11,100);
INSERT INTO book VALUES(2,'幻城',12,102);
INSERT INTO book VALUES(3,'希腊神话',13,102);
INSERT INTO book VALUES(4,'一千零一夜',13,102);

#5、查询出版社id为100的书的全部信息
SELECT * FROM book WHERE pressid=100;

#6、查询出版社为外研社的书的全部信息
SELECT * FROM book WHERE pressid=(SELECT pressid FROM press WHERE pressname='外研社');

#7、查询图书数量（scount）大于100的种类
SELECT * FROM sort WHERE scount>100;

#8、查询图书种类最多的出版社信息
SELECT * FROM press WHERE pressid=(
    SELECT temp.pressid FROM
 (SELECT pressid,MAX(t.c) FROM (SELECT pressid,COUNT(*) AS c FROM book GROUP BY
pressid ORDER BY c DESC ) AS t) AS temp);


SELECT * FROM press WHERE pressid=(
SELECT pressid
FROM (SELECT pressid,bsortno FROM book GROUP BY pressid,bsortno) temp
GROUP BY pressid
ORDER BY COUNT(*) DESC
LIMIT 0,1)
```

练习8

1、建立数据库：test_tour

2、建立如下两个表：

agency旅行社表：

列名（英文名）	列名（中文名）	数据类型	允许空值	说明
Id	旅行社编号	int	no	主键
Name	旅行社名	varchar	no	
Address	旅行社地址	varchar	no	
Areaid	所属区域Id	Int	yes	

travel旅行线路表：

列名（英文名）	列名（中文名）	数据类型	允许空值	说明
Tid	旅行线路编号	int	no	主键
Time	所需时间	varchar	no	
Position	目的地	varchar	no	
Money	花费	Float	yes	
Aid	所属旅行社id	Int	no	外键
Count	报名人数	Int	yes	

3、添加记录

agency表数据

id	name	address
101	青年旅行社	北京海淀
102	天天旅行社	天津海院

travel表数据

tid	time	position	money	aid	rcount
1	5天	八达岭	3000	101	10
2	7天	水长城	5000	101	14
3	8天	水长城	6000	102	11

4、查出旅行线路最多的旅社

5、查出最热门的旅行线路(也就是查询出报名人数最多的线路)

6、查询花费少于5000的旅行线路

7、找到一次旅行花费最昂贵的旅行社名

8、查出青年旅社所有的旅行线路都玩一遍需要多少时间。

```
#1、建立数据库：test_tour
CREATE DATABASE test_tour;

#使用数据库
USE test_tour;

#2、
CREATE TABLE agency(
    id INT PRIMARY KEY NOT NULL,
    NAME VARCHAR(20) NOT NULL,
    address VARCHAR(100) NOT NULL,
    areaid INT
);

CREATE TABLE trval(
    tid INT PRIMARY KEY NOT NULL,
    TIME VARCHAR(50) NOT NULL,
    POSITION VARCHAR(100) NOT NULL,
    money FLOAT,
    aid INT NOT NULL,
    rcount INT,
    CONSTRAINT bk_aid FOREIGN KEY trval(aid) REFERENCES agency(id)
);

#3、
INSERT INTO agency(id,NAME,address) VALUES (101,'青年旅行社','北京海淀');
INSERT INTO agency(id,NAME,address) VALUES (102,'天天旅行社','天津海院');

INSERT INTO trval(tid,TIME,POSITION,money,aid,rcount) VALUES (1,'5天','八达
岭',3000,101,10);
INSERT INTO trval(tid,TIME,POSITION,money,aid,rcount) VALUES (2,'7天','水长
城',5000,101,14);
INSERT INTO trval(tid,TIME,POSITION,money,aid,rcount) VALUES (3,'8天','水长
城',6000,102,11);

SELECT * FROM agency;
SELECT * FROM trval;

#4、查出旅行线路最多的旅社
SELECT *
FROM agency INNER JOIN
(SELECT t.aid,MAX(t.c) FROM (SELECT aid,COUNT(*) AS c FROM trval GROUP BY aid)  AS
t)temp
ON agency.id = temp.aid

#5、查出最热门的旅行线路(也就是查询出报名人数最多的线路)
SELECT * FROM trval WHERE rcount=(SELECT MAX(rcount) FROM trval);

#6、查询花费少于5000的旅行线路
SELECT * FROM trval WHERE money<5000;
```

```sql
#7、找到一次旅行花费最昂贵的旅行社名
SELECT NAME FROM agency WHERE id =
(SELECT aid FROM trval WHERE money =(SELECT MAX(money) FROM trval ));

#8、查出青年旅社所有的旅行线路都玩一遍需要多少时间。
SELECT SUM(TIME) FROM trval WHERE aid=(SELECT id FROM agency WHERE NAME='青年旅行社');
```