

数据库

张创琦组卷

关系的完整性、关系代数

1. 设有如下关系：

图书（书号，书名，作者，出版社）；

读者（借书证号，读者名，读者地址）；

借阅（读者名，书号，借书日期，归还日期）。

① 指出每个关系模式的候选码，主码，外码，主属性。

② 试用关系代数表达式查询 2010 年 12 月 31 日以前已经还书和借书未还的读者姓名和图书书名。

2. 设关系 R、W 和 D 如下，计算下列关系代数：

关系 R

P	Q	T	Y
2	b	c	d
9	a	e	f
2	b	e	f
9	a	d	e
7	g	e	f
7	g	c	d

关系 W

T	Y	B
c	d	m
c	d	n
d	f	n

关系 D

T	Y
c	d
e	f

① $R1 = \Pi_{Y,T}(R)$

② $R2 = \sigma_{P>5 \wedge T=c}(R)$

③ $R3 = R \bowtie W$

④ $R4 = \Pi_{2,1,6}(\sigma_{3=5}(R \times D))$

⑤ $R5 = R \div D$

二 已知表格如下，回答下列问题。

表 2-1 关系数据库实例 *Education* 中的学生关系 *Students*

<i>Sno</i> 学号	<i>Sname</i> 姓名	<i>Gender</i> 性别	<i>BirthDate</i> 出生日期	<i>PID</i> 身份证号	<i>Major</i> 所属专业
S1	李红	女	1990-12-21	330602199012211234	信息管理
S2	张雷	男	1991-01-15	110103199101156789	计算机科学
S3	胡斌	男	1989-11-09	321001198911091234	电子商务
S4	陈龙	男	1991-04-12	360101199104123456	信息管理
S5	赵敏	女	1990-12-06	440605199012067891	计算机科学
S6	王涛	男	1990-10-17	330101199010178901	信息管理

表 2-2 关系数据库实例 *Education* 中的课程关系 *Courses*

<i>Cno</i> 课程号	<i>Cname</i> 课程名	<i>Pcno</i> 前修课程	<i>Type</i> 类别	<i>Credit</i> 学分
C1	数据库	C2	自然科学	4
C2	C 程序设计		自然科学	3
C3	电子商务		经济管理	2
C4	数据结构	C2	自然科学	3
C5	编译原理	C4	自然科学	3
C6	操作系统	C5	自然科学	4
C7	Web 技术	C1	自然科学	3

表 2-3 关系数据库实例 *Education* 中的教师关系 *Teachers*

<i>Tno</i> 教师号	<i>Tname</i> 姓名	<i>Gender</i> 性别	<i>BirthDate</i> 出生日期	<i>Title</i> 职称	<i>Major</i> 所属专业
T1	赵宏民	男	1967-12-15	教授	信息管理
T2	钱敏霞	女	1976-04-16	副教授	电子商务
T3	孙一维	男	1980-10-23	讲师	计算机科学
T4	李晓明	男	1978-11-06	副教授	信息管理

表 2-4 关系数据库实例 *Education* 中的选课关系 *StudCourses*

<i>Sno</i> 学号	<i>Cno</i> 课程号	<i>Grade</i> 成绩		<i>Sno</i> 学号	<i>Cno</i> 课程号	<i>Grade</i> 成绩
S1	C1	90		S2	C5	82
S1	C2	86		S3	C1	65
S1	C3	75		S3	C3	50
S1	C4	70		S4	C1	69
S2	C1	82		S4	C6	55
S2	C4	88		S5	C7	76

表 2-7 关系数据库实例 *Education* 中的授课关系 *Instructions*

<i>Tno</i> 教师号	<i>Cno</i> 课程号	<i>Period</i> 开课学期		<i>Tno</i> 教师号	<i>Cno</i> 课程号	<i>Period</i> 开课学期
T1	C1	2010-1		T3	C5	2010-1

T1	C2	2010-1		T3	C7	2010-1
T1	C1	2010-2		T3	C5	2010-2
T2	C3	2010-1		T3	C7	2010-2
T2	C1	2010-1		T4	C6	2010-1
T2	C3	2010-2		T4	C6	2010-2

用关系代数完成下列查询：

- 1 查询选修了 C2 课程的学生学号和姓名。
- 2 查询没有选修 C3 课程的学生学号。
- 3 查询既选修了 C1 又选修了 C3 课程的学生学号。
- 4 查询选修了 C1 或 C3 课程的学生姓名。
- 5 查询选修了课程名称为“数据库”的学生姓名。
- 6 查询至少选修 C1 和 C3 课程的学生号码。
- 7 查询哪些学生至少选修了学号为“S1”的这个学生所选修过的所有课程，并得到这些学生的学号和姓名。
- 8 查询至少选修了两门自然科学类课程的学生姓名。
- 9 查询哪些学生至少选修了一门学号为“S1”的这个学生所选修过的课程，求这些学生的学号和姓名。

三 给定一个有关学生选课数据库 Education，它包含学生、课程、教师、选课和授课等 5 个关系模式，分别用 Students, Courses, Teachers, StudCourses, Instructions 表示。各个关系模式表示如下：

Students (Sno, Sname, Gender, Major) = 学生 (学号, 姓名, 性别, 所属专业)

Courses (Cno, Cname, Pno, Credit) = 课程 (课程编号, 课程名称, 前修课程, 学分)

Teachers (Tno, Tname, Title, Major) = 教师 (教师编号, 姓名, 职称, 所属专业)

StudCourses (Cno, Sno, Period, Grade) = 选课 (学号, 课程编号, 选课学期, 成绩)

Instructions (Tno, Cno, Period) = 授课 (教师编号, 课程编号, 授课学期)

用关系代数完成下列查询：

- 1 查询选修过“数据库”和“数据结构”这两门课程的学生姓名。
- 2 查询姓名为“李平”这个学生所选修的全部课程的名称。
- 3 查询没有选修过“数据库”这门课程的学生姓名。
- 4 查询选修过“数据库”但没有选修其先行课的学生学号。
- 5 查询所有课程成绩全部及格的学生姓名。
- 6 查询选修过教师“达尔文”所授的全部课程的学生姓名。
- 7 查询哪些学生选修的课程中其前修课程还没有选修过。
- 8 查询哪些学生与学号为“S1”的学生选修了完全相同的课程。
- 9 查询哪些学生没有选修过教师“达尔文”所授的任何一门课程。
- 10 查询哪些学生至少选修了教师“达尔文”所授的两门不同的课程。

SQL 基础语句练习 (用 SQL 语句写出下列要求)

练习 1:

Manager(管理员表):

mid 编号 int (主键)

mname 名字 varchar(20)

age 年龄 int

sex 性别 char(2)

password 密码 varchar(20)

address 地址 varchar(20)

phone 电话 varchar(20)

数据:

1 王子 18 男 123 北京 110

2 公主 20 女 456 上海 220

3 太子 23 男 789 南京 330

(1)查询公主的所有信息

(2)查询年龄在 18-30 之间的管理员姓名

(3)查询表中所有的用户名和电话

(4)查询性别是男, 名字是王子的个人信息

(5)查询出地址在北京和上海的员工信息

练习 2:

scores

stuid int 学生 id

java int java 成绩

mysql int mysql 成绩

stuname varchar(20) 学生姓名

数据:

limit (pageindex-1)*pagesize,pagesize; 1 67 78 张三

2 87 55 李四

3 66 90 王五

4 98 78 赵六

5 80 88 田七

需求:

(1)对 java 成绩进行降序排序

(2)得到 mysql 成绩前三名

(3)得到 java 学生中最后一名学生的信息

(4)查询出两门成绩都优秀(≥ 80)的学生姓名

(5)查询出成绩在 90 分以上(≥ 90)的学生信息

(6)查询出每名学员的 java,mysql,总成绩

(7)显示出每名学生的总分以及姓名

练习 3:

测试数据:

郭敬明 1371234567 北京 java S1101 89 1979-04-05

张三丰 1372839201 上海 数据库 S1102 67 1967-09-07

赵敏 1387839201 山东 mysql S1103 99 1987-09-07

Student2

stuname 姓名 varchar(20)

telephone 电话 varchar(20)

address 住址 varchar(20)

subject 科目 varchar(20)

stuNo 学号 varchar(20)

score 成绩 int

birthday 出生日期 date

//1.要查询列 2.条件

a.查询住址为“山东”的学生姓名、电话、住址

b.查询名称中含有“数据库”字样科目信息

c.查询电话中以“1387”开头的学生信息

d.查询姓姜的, 三个字的学生信息

欢迎加入 QQ 群数据库试卷探讨: 790586125 试题由张创琦拼凑而成, 很多题目没有答案, 欢迎加群探讨

e.查询学号为 S1101 的指定 java, mysql 科目考试成绩

f.查询出 80 后学员信息

g.查询出家庭住址在北上广的学生名字 h.显示成绩在第 5-10 名的学生名字和电话

i.查询分数在 80-90 之间并且在北京的学生

练习 4:聚合函数练习

表: scores2

年级 grade varchar(10)

学号 stuno varchar(20)

考试时间 examDate date

科目 subject varchar(20)

成绩 score int

学期 xueqi int

数据:

S1 S1101 2015-02-03 C 89 1

S2 S1103 2015-03-03 JAVA 90 2

S3 S1102 2015-07-03 C 100 1

1.查询学生总人数

2.学号为 S1101 的学生第一学期考试总成绩,平均分

3.查询 2013 年 3 月 22 日科目“C”的最高分、最低分、平均分

4.查询 2013 年 3 月 22 日科目“C”及格学生的平均分

5.查询所有参加“C”科目考试的平均分

6.查看考 java 的人数

练习 5:分组练习

表名: student

年级(grade) varchar(10)

学生姓名(name) varchar(10)

学时(xueshi) int --每人单个学时

参加考试(isexam) char(1) 是/否、

课程(subject) varchar(10)

分数(score) int

数据:

1 张三 10 是 java 99

1 李四 10 否 java 0

2 王五 20 是 mysql 88

2 赵六 20 是 mysql 77 2 王五 20 是 java 99

2 赵六 20 否 java 0

1 张三 10 是 mysql 88

练习:

a:查询每个年级的总学时数, 并按照升序排列

b:查询每个参加考试的学员的平均分

c:查询每门课程的平均分, 并按照降序排列

练习 6:综合练习

Student

科目名称 subjectName varchar(20)

学生姓名 stuname varchar(20)

学生地址 address varchar(20)

学生性别 sex char(2)

电子邮件 email varchar(30)

年级 grade varchar(10)

出生日期 birthday date

考试日期 examDate date

成绩 scores int

数据:

JAVA 张三 北京 男 123@qq.com S1 1990-03-04 2013-5-6 89

html 李四 上海 男 S2 1993-08-04 2014-5-6 87

html 王五 北京 男 123@qq.com S2 1990-03-04 2015-4-6 90

1.查询 S2 的科目名称

2.查询 S2 男同学的姓名和住址

欢迎加入 QQ 群数据库试卷探讨：790586125 试题由张创琦拼凑而成，很多题目没有答案，欢迎加群探讨

3.查询无电子邮件的学生姓名和年级信息

4.查询出生日期在 1993 年之后的 S2 的学生姓名和年级信息

5.查询参加了日期为 2013 年 2 月 15 日的“HTML” 科目考试的成绩信息

练习 7:综合练习

某网上商城数据库如下图所示



#一对多的实现

#创建分类表

```
create table category(
cid varchar(32) PRIMARY KEY ,# 分类 id
cname varchar(100) #分类名称
);
```

商品表

```
CREATE TABLE `products` (
`pid` varchar(32) PRIMARY KEY ,#商品 id
`name` VARCHAR(40) ,#商品名称
`price` DOUBLE, # 价格
category_id varchar(32) # 分类 id,外键列
);
```

#多对多的实现

#用户表

```
create table users(
userid int,
username varchar(20),
upass varchar(20)
)
```

#订单表

```
create table `orders`(
`oid` varchar(32) PRIMARY KEY ,#订单 id
`totalprice` double, #总计
uid int # 用户 id
);
```

```
# 订单项表
create table orderitem(
oid varchar(50), #订单 id
pid varchar(50) #商品 id
);
#初始化数据
#给分类表初始化数据
insert into category values('c001','电器');
insert into category values('c002','服饰');
insert into category values('c003','化妆品');
insert into category values('c004','书籍');
#给商品表初始化数据
insert into products(pid,name,price,category_id) values('p001','联想',5000,'c001');
insert into products(pid,name,price,category_id) values('p002','海尔',3000,'c001');
insert into products(pid,name,price,category_id) values('p003','雷神',5000,'c001');
insert into products(pid,name,price,category_id) values('p004','JACK JONES',800,'c002');
insert into products(pid,name,price,category_id) values('p005','真维斯',200,'c002');
insert into products(pid,name,price,category_id) values('p006','花花公子',440,'c002');
insert into products(pid,name,price,category_id) values('p007','劲霸',2000,'c002');
insert into products(pid,name,price,category_id) values('p008','香奈儿',800,'c003');
insert into products(pid,name,price,category_id) values('p009','相宜本草',200,'c003');
insert into products(pid,name,price,category_id) values('p010','梅明子',200,null);
```

3.1 综合练习-【多表查询】

1>查询用户的订单,没有订单的用户不显示

2>查询所有用户的订单详情

3>查询所有订单的用户详情

3.2 综合练习 2-【子查询】

1>查看用户为张三的订单详情

2>查询出订单的价格大于 300 的所有用户信息。

3>查询订单价格大于 300 的订单信息及相用户的用户信息。

3.2 综合练习 3-【分页查询】

1>查询所有订单信息，每页显示 5 条数据

综合练习

判断数据库 myGrade 是否存在，如果不存在，则创建之。打开该数据库，创建学生、课程、教师、选课和授课等 5 个数据表，各表及其列名含义如下（主键已用下画线标记）：

学生（学号，姓名，性别，出生日期，班级）

课程（课程编号，课程名称，前修课程，课程性质，学分）

教师（教师编号，姓名，性别，出生日期，职称）

选课（学号，课程编号，选课学期，成绩）

授课（教师编号，课程编号，授课学期）

试创建上述 5 个表，要求使用非中文的表名和列名，并在各表中插入模拟数据。在建表时必须定义各表的主键、外键、CHECK 等约束条件。具体要求如下：

1) 学生表和教师表中的性别取值“M”或“F”分别表示“男”或“女”；学号长度为 8 位，第一位以字母开头，最后一位为性别（即 F 或 M），其他 6 位为数字。

2) 课程表中的前修课程为外键，它参照自己所在表中的主键列（即“课程编码”）；课程性质分为“必修课”和“选修课”两类；学分取值 0.5~10 之间。

3) 选课学期和授课学期都为 11 位字符串，例如“2010-2011-1”。其中前 9 位表示学年（年份之间用横杆分隔），最后一位表示某个学年中的学期序号，取值 1 或 2。

4) 其他列的类型、长度、外键及 CHECK 等约束条件根据选课数据库语义自行定义。

5) 在插入模拟数据之后，为各个外键中的每一列创建非聚簇索引（二级索引）。

SQL 语句综合练习

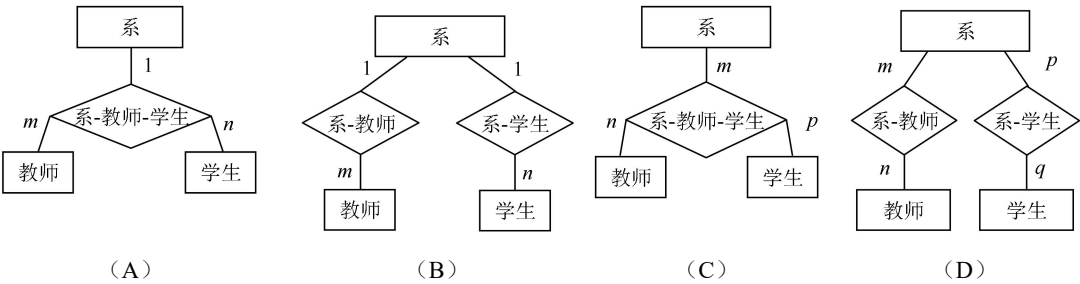
试为 A 工厂的生产制造系统设计一个数据库。该数据库包含车间、员工、原材料、供应商等对象，用于生产不同的产品。假设该数据库的语义如下：

1) 每个车间包含车间编号和车间名称等属性；每个车间每天生产一定数量的产品，每个产品包含产品编码、名称、规格型号和计量单位等属性。2) 每个车间设有一个车间主任，车间与员工定期签订劳动合同；每个员工包含编号、姓名、性别、出生日期、身份证号和家庭住址等属性；一个员工只能担任一个车间的主任。3) 每个产品在生产制造中消耗多种原材料，每个原材料包含编码、名称、规格型号、计量单位、成本单价等属性；工厂每个月需要统计每个产品耗用原材料的数量和金额。4) 每个供应商包含编码、名称、地址等属性，每个原材料只有一个供应商提供。

试创建满足该数据库需求的各个数据表，并用 INSERT 语句插入模拟数据。

规范化及 E-R 图

1 设系、学生、教师3个实体之间存在约束：一个系可以有多名教师，一名教师只属于一个系，一个系可以有多名学生，一名学生只属于一个系。下列E-R图中能准确表达以上约束的是（ ）。



2. 已知某教务管理系统的各个关系模式如下（带下画线的属性是主码）：

- 系(系号，系名)
- 学生(学号，姓名，性别，入学日期，正常毕业日期，所在系号)
- 课程(课程号，课程名，学分，开课系号，教师号)
- 选课(学号，课程号，成绩)
- 教师(教师号，姓名，职称，所在系号)

① 请根据给出的关系模式，画出该关系的 E-R 图，图中忽略实体集的属性，但如果实体集之间的联系有属性则需要给出联系的属性。

② 假设该系统的业务需求发生变化，需要满足下列要求：为课程增加先修课程信息（一门课程可有多门先修课程）；一门课程可由多名教师讲授，一名教师可以讲授多门课程。试根据上述需求修改关系模式，仅列出有改动的关系模式，并使每个关系模式满足 3NF。

3 某旅游公司向游客提供一系列旅游服务项目，游客以项目（即团队）的形式参加旅游活动。已知相关实体及其属性如下：1）旅游项目/团队：项目编码、名称、起止日期、简介等；2）游客：游客姓名、性别、出生日期、联系电话、身份证号；3）导游：导游编码、姓名、性别、出生日期、联系电话、简介；4）酒店：酒店编码、名称、地址、联系电话；5）景点：景点编码、名称、地址、简介。

假设数据库的语义如下：1）一个旅游项目只有一个导游，一个导游可以负责多个旅游项目；2）一个游客可以参加多个旅游项目，但同一个游客在同一时间内只能参加一个旅游项目，同一旅游项目对不同游客的收费可能不同；3）内容相同但时间不同的旅游项目按不同项目处理，例如 2018 年 3 月 1 日~5 日与 2018 年 5 月 1 日~5 日的“丽江 5 日游”按两个不同的旅游项目处理；4）一个旅游项目包含多个游览景点及居住旅馆等信息，不同时期同一旅游项目的酒店或景点的收费标准（价格）可能不同。试根据上述实体属性与语义完成下列各项数据库设计任务：

1）设计满足上述语义要求的 E-R 图。

2）根据上述 E-R 图，导出该数据库的各个关系模式，列出各个关系模式中的主码和外码及其参照的属性对象。

3）使用关系代数，检索游客“赵敏”2018 年游览过哪些景点。

欢迎加入 QQ 群数据库试卷探讨：790586125 试题由张创琦拼凑而成，很多题目没有答案，欢迎加群探讨

4) 使用关系代数，检索哪些游客参加过该公司的两个或两个以上的旅游项目，要求列出游客的姓名和身份证号。

5) 使用 CREATE 语句将“旅游项目”关系模式转成 MySQL 数据表（列名以英文表示用中文标注），要求注明主键、外键和其他约束条件，其中旅游项目编码规则为：前 10 位为日期，后 3 位为字母，最后一位为序号（例如 2018-05-01LJY1）。

6) MySQL 中如何实现“同一个游客在同一时间内只能参加一个旅游项目”这个约束条件。（可以不编程描述方法，也可以编程实现）

并发控制

1 设 T_1 , T_2 , T_3 是如下的三个事务, 设 A 的初始值为 0.

$T_1: A:=A+2;$

$T_2: A:=A*2;$

$T_3: A:=A**2;$ (即 $A \leftarrow A^2$)

(1) 若这三个事务允许并发执行, 则有多少种可能的正确结果? 请一一列举出来;

(2) 请给出一个可串行化的调度, 并给出执行结果;

(3) 请给出一个非串行化的调度, 并给出执行结果;

(4) 若这三个事务都遵守两段锁协议, 请给出一个不产生死锁的可串行化调度;

(5) 若这三个事务都遵守两段锁协议, 请给出一个产生死锁的调度。

2 今有三个事务的一个调度 $r_3(B)r_1(A)w_3(B)r_2(B)r_2(A)w_2(B)r_1(B)w_1(A)$, 该调度是冲突可串行化的调度吗? 为什么?

3 考虑 T_1 和 T_2 两个事务。

$T_1: R(A);R(B);B=A+B;W(B)$

$T_2: R(B);R(A);A=A+B;W(A)$

(1) 改写 T_1 和 T_2 , 增加加锁操作和解锁操作, 并要求遵循两段封锁协议;

(2) 说明 T_1 和 T_2 的执行是否会引起死锁, 给出 T_1 和 T_2 的一个调度并说明之。