**一、设有关系Student(Sno, Sname, Ssex, Sage, Sdept), 其中各属性的含义依次为学号、姓名、性别、年龄、系别，关系Course(Cno, Cname, Cpo, Ccredit), 其中各属性的含义依次为课程号、课程名、先行课、学分，以及关系SC(Sno, Cno, Grade)，其中各属性的含义依次为学号、课程号、成绩。采用SQL语句完成以下操作：**

**（1）查询全体学生（Student）的学号与姓名。**

**（2）查询考试成绩不及格的学生的学号。**

**（3）查询课程名中有“语言”两字的课程的课程号和课程名。**

**（4）查询每个学生超过他自己选修课程平均成绩的课程号。**

**（5）将学生的学号及所选修课程的平均成绩定义为一个视图。**

**解：（1）Select Sno, Sname**

**From Student;**

**（2） Select DISTINCT Sno**

**From SC**

**WHERE Grade<60;**

**（3）Select Cno, Cname**

**From Course**

**WHERE Cname LIKE ‘%语言%’;**

**（4）Select Sno, Cno**

**FROM SC x**

**WHERE Grade>=(Select AVG(Grade)**

**FROM SC y**

**WHERE y.Sno=x.Sno);**

**（5）CREATE VIEW S-G(Sno, Gavg)**

**AS**

**SELECT Sno, AVG(Grade)**

**FROM SC**

**GROUP BY Sno;**

**二、**

**（1）使用SQL语句创建表authors，结构如下表所示：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **列名** | **数据类型** | **大小** | **是否为空** | **说明** |
| **authorID** | **int** |  | **N** | **编号（主键），自动增长** |
| **authorName** | **varchar** | **50** | **N** | **姓名** |
| **phone** | **varchar** | **13** | **Y** | **电话** |
| **address** | **varchar** | **255** | **Y** | **地址** |

**使用SQL语句对表authors进行如下修改：**

1. **添加Asex列char(2)，Acity列varchar(20);**
2. **删除Asex列；**
3. **修改表authors中的address列，将该列的属性长度更改为450，并不能为空。**
4. **解：**Create table authors(

authorID int primary key not null auto\_increment,

authorName varchar(50) not null,

Phone varchar(13),

Address varchar(255)

);

1. **解：**Alter table authors

Add Asex char(2), Acity varchar(20);

1. **解：**Alter table authors drop column Asex;
2. **解：**Alter table authors

Alter column address varchar varchar(450) not null;

**三、设有一关系模式R如下：**

**R（商店编号，商品编号，数量，部门编号，负责人），如果规定：**

**（1）每个商店的每种商品只在一个部门销售；**

**（2）每个商店的每个部门只有一个负责人；**

**（3）每个商店的每种商品只有一个库存数量。**

**试回答下列问题：**

**（1）根据上述规定，写出关系模式R的基本函数依赖；**

**（2）找出关系模式R的候选码；**

**（3）试问关系模式R最高已经达到了第几范式？为什么？**

**（4）如果R不属于3NF，请将R分解成3NF模式集。**

**解：（1）有三个函数依赖：**

**（商店编号，商品编号）—>部门编号**

**（商店编号，部门编号）—>负责人**

**（商店编号，商品编号）—>数量**

**（2）R的候选码是（商店编号，商品编号）。**

**（3）R最高达到2NF。因为R中存在非主属性“负责人”对候选码的传递函数依赖，所以R属于2NF，但R不属于3NF。**

**（4）将R分解成：R1（商店编号，商品编号，数量，部门编号）**

**R2（商店编号，部门编号，负责人）**

**四、设有关系R和函数依赖F：**

**R（X, Y, Z），。**

**试求下列问题：**

**（1）关系R属于第几范式？**

**（2）如果关系R不属于BCNF，请将关系R逐步分解为BCNF。**

**要求：写出达到每一级范式的分解过程，并指明消除什么类型的函数依赖。**

**解：（1）R是3NF。候选码为XY和XZ，R中所有属性都是主属性，不存在非主属性对码的传递函数依赖。**

**（2）R不属于BCNF。将R分解为：**

**R1（X, Y, Z），**

**R2（Y, Z），**

**消除了非主属性对码的传递依赖，F1和F2中的函数依赖都是非平凡的，并且决定因素都是候选码，所以上述关系模式是BCNF。**

**五、关系模式R(A,B,C,D)，写出满足下列函数依赖时R的码，并给出R属于的最高范式是哪种？给出原因。**

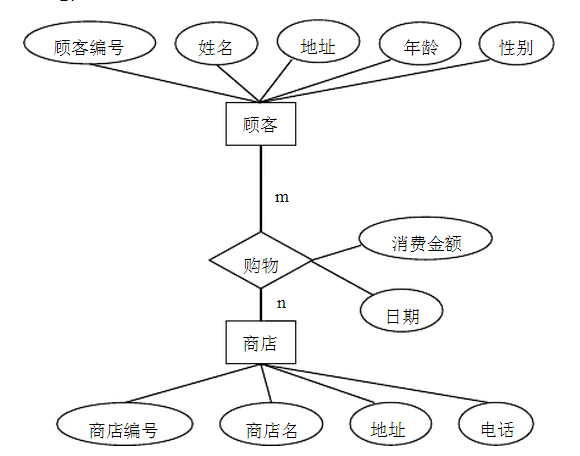
****

**解：**（1）R的码为D，R是2NF。

1. R的码为ABD或CBD，R是3NF。
2. R的码为ABC，R是BCNF。

**六、设有商店和顾客两个实体，“商店”有属性商店编号、商店名、地址、电话，“顾客”有属性顾客编号、姓名、地址、年龄、性别。假设一个商店有多个顾客购物，一个顾客可以到多个商店购物，顾客每次去商店购物有一个消费金额和日期，而且规定每个顾客在每个商店里每天最多消费一次。试画出E-R图，注明属性和联系类型，并将E-R模型转换成关系模式，要求关系模式主码加下划线表示。**

**解：（1）E-R图如下：**

****

**（2）顾客（顾客编号，姓名，地址，年龄，性别），无外码**

**商店（商店编号，商店名，地址，电话），无外码**

**购物（顾客编号，商店名称，日期，消费金额）,假设商店名称是唯一的，外码有顾客编号、商店名称**

**七、**

1. **什么是两段锁协议？（6分）**
2. **设T1、T2、T3是如下三个事务，设A的初值为4。（10分）**

****

**请给出这两个事务都遵守两段锁协议的一个不产生死锁的可串行化调度。**

**解：**（1）所谓两段锁协议是指所有事务必须分两个阶段对数据项加锁和解锁，在对任何数据进行读、写操作之前，首先要申请并获得对该数据的封锁，即获得封锁，在释放一个封锁之后，事务不再申请和获得其他封锁，即使放封锁，其目的是保证并发调度的正确性。

（2）

|  |  |
| --- | --- |
| T1 | T2 |
| Slock A  Y=A=4  Xlock A  A=Y-2  写回A（=2）  Unlock A  Unlock A | Slock A  等待  等待  Y=A=2  Xlock A  等待  A=Y\*2+1  写回A（=5）  Unlock A  Unlock A |

**1、已知某企业各部门有许多职员，一个职员仅属于一个部门；每个职员可在多项工程中做工，职工可负责多个项目管理，一个项目只有一名负责人，每项工程可由多个职员参加。其中，**

**职工有：职工号，职工名；**

**部门有：部门号，部门名；**

**工程有：工程号，工程名；**

**每个职工参加某个工程有参加日期。**

**请完成如下设计或处理：**

1. 根据上述语义设计E-R模型，要求标注联系类型。
2. 将E-R模型转换成关系数据模型，并指出每一个关系的主键和外键（如果存在）。

**2、某企业集团有若干工厂，每个工厂生产多种产品，且每一种产品可以在多个工厂生产，每个工厂按照固定的计划数量生产产品；每个工厂聘用多名职工，且每名职工只能在一个工厂工作，工厂聘用职工有聘期和工资。工厂的属性有工厂编号、厂名、地址，产品的属性有产品编号、产品名、规格，职工的属性有职工号、姓名。（共20分）**

1. 根据上述语义画出E-R图；(8分)
2. 将E-R模型转换为关系模型（要求：1：1和1：n的联系进行合并）；(8分)
3. 指出转换结果中每个关系模式的主码和外码。(4分)

**3、关系模式R(A,B,C,D)，写出满足下列函数依赖时R的码，并给出R属于的最高范式是哪种？给出原因。**

****

**4、设有关系R函数依赖F：（每小题5分，共10分）**

**R（A，B，C，D，E），F={}。**

1. 关系R的候选码是什么？R属于第几范式？并说明理由。
2. 如果关系R不属于BCNF，请将关系R逐步分为了BCNF。

**5、某商场可以为顾客办理会员卡，每个顾客只能办理一张会员卡，顾客信息包括顾客姓名、地址、电话、身份证号，会员卡信息包括号码、等级、积分，顾客具有多个地址，每个地址具有多个电话号码，地址包括地址编号、省、市、区、街道，电话号码包括区号、号码。**

解题要求：

1. 根据上述语义画出E-R图；(13分)

完成数据库的逻辑模型的设计，并指出关系模式中每个关系的主码和外码。(12分)

**1、解：（1）本题的E-R图：**

职工

m

n

n

工程

部门

工作

参加

1

负责

1

n

1. **转换后的关系模式：**

职工（职工号，职工名，部门号）；工程（工程号，工程名，负责人）；

部门（部门号，部门名）； 参加（职工号，工程号，日期）。

工程：主码是工程号，外码负责人； 部门：主码是部门号，无外码；

职工：主码是职工号，外码是部门号；

参加：主码是（职工号，工程号），外码是职工号，工程号。

**2、解：（1）本题的E-R图：（8分）**

工厂

职工

产品

生产

聘用

1

n

m

n

1. **转换后的关系模式：（8分）**

工厂（工厂编号，厂名，地址）；产品（产品编号，产品名，规格）；

职工（职工号，姓名，工厂编号，聘期，工资）； 生产（工厂编号，产品编号，计划数量）。

1. **每个关系的主码、外码如下：（4分）**

工厂：主码是工厂编号，无外码； 产品：主码是产品编号，无外码；

职工：主码是职工号，外码是工厂编号；

生产：主码是（工厂编号，产品编号），外码是工厂编号，产品编号。

1. **解：**
2. R的码为D，R是2NF。
3. R的码为ABD或CBD，R是3NF。
4. R的码为ABC，R是BCNF。

4、 **解：**（1）关系R的候选码是（A，B，C），，因为R中存在非主属性D，E对候选码（A，B，C）的部分函数依赖。

（2）首先消除部分函数依赖，将关系分解为：R1（A，B，C），R2（B，C，D，E），R2的函数依赖集为：F2={}，存在非主属性E对候选码（B，C）的传递函数依赖，所以R2进一步分解：R21（B，C，D），（B，C）为候选码，R22（D，E），D为候选码。

**5、解：（1）本题的E-R图：（13分）**

地址

顾客

会员卡

拥有

拥有地址

1

1

n

1

电话

拥有电话

1

m

顾客

会员卡

电话

顾客

1. **转换后的关系模式：（12分）**

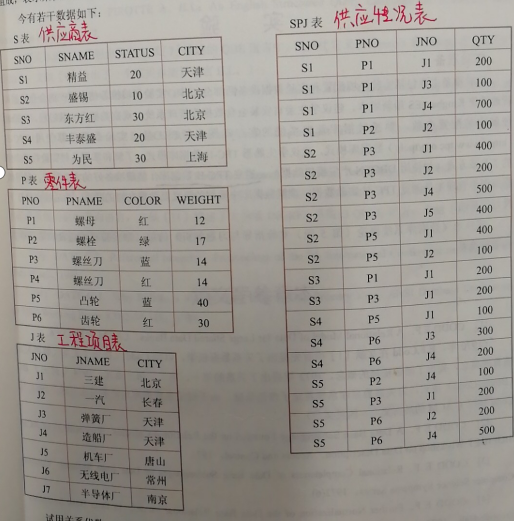
顾客（姓名，身份证号，卡号码）， 主码：身份证号，外码：卡号码；

会员卡（卡号码，等级，积分），主码：卡号码，无外码；

地址（地址编号，省，市，区，街道，卡号码），主码：地址编号，外码：卡号码；

电话（号码，区号，地址编号），主码：（号码，区号），外码：地址编号。

P130-5



1. 找出所有供应商的姓名和所在城市；

Select sname, city from S;

1. 找出所有零件的名称、颜色、重量；

Select pname, color, weight from P;

1. 找出使用供应商S1所供应零件的工程号码；

Select jno from SPJ

Where sno=’S1’;

1. 找出工程项目J2使用的各种零件的名称及其数量；

Select P.pname, SPJ.QTY from P, SPJ

Where P.PNO=SPJ.PNO and SPJ.JNO=’J2’;

1. 找出上海厂商供应的所有零件号码；

Select Distinct PNO from SPJ

Where SNO in (

Select SNO from S where city=’上海’);

1. 找出适用上海产的零件的工程号码；

Select jname from J, SPJ, S

Where J.JNO=SPJ.JNO and SPJ.SNO=S.SNO and S.city=’上海’;

或

Select jname from J

Where JNO in (

Select JNO from SPJ, S where SPJ.SNO=S.SNO and S.city=’上海’);

1. 找出没有使用天津产的零件的工程号码；

Select JNO from J

Where not exists (

Select \* from SPJ

Where SPJ.JNO=J.JNO and SNO in (

Select SNO from S

Where city=’天津’)

);

或

Select JNO from J

Where not exists (

Select \* from SPJ, S

Where SPJ.JNO=J.JNO and SPJ.SNO=S.SNO and S.city=’天津’);

1. 把全部红色零件的颜色改成蓝色；

Update P

Set color=’蓝’

Where color=’红’;

1. 由S5供给J4的零件P6改为由S3供应，请作必要的修改；

Update SPJ

Set SNO=’S3’

Where SNO=’S5’ AND JNO=’J4’ AND PNO=’P6’;

1. 从供应商关系中删除S2的纪录，并从供应情况关系中删除相应的纪录；

Delete from SPJ

Where SNO=’S2’;

Delete from S

Where SNO=’S2’;

1. 请将（S2，J6，P4，200）插入供应情况关系。

Insert into SPJ

Values(S2, P4, J6, 200);

或

Insert into SPJ(SNO, JNO, PNO, QTY)

Values(S2, P4, J6, 200);

P130-9

请为三建工程项目建立一个供应情况的视图，包括供应商代码（SNO）、零件代码（PNO）、供应数量（QTY）。针对该视图完成下列查询：

1. 找出三建工程项目使用的各种零件代码及其数量；

创建视图：

Create view v\_spj AS

Select SNO, PNO, QTY

From SPJ

Where JNO=

( select JNO from J where Jname=’三建’);

Select PNO, QTY

From v\_spj;

1. 找出供应商S1的供应情况。

Select PNO, QTY

From v\_spj

Where SNO=’S1’;

P155-6

有学生表和班级表

1. 授予用户U1对两个表的所有权限，并可给其他用户授权。

Grant all privileges on table 学生，班级 to U1 with grant option;

1. 授予用户U2对学生表具有查看权限，对家庭住址具有更新权限。

Grant select, update(家庭住址) on table 学生 to U2;

1. 将对班级表查看权限授予所有用户。

Grant select on table 班级 to pulic;

1. 将对学生表的查询、更新权限授予角色R1。

Grant select, update on table 学生 to R1;

1. 将角色R1授予用户U1，并且U1可继续授权给其他角色。

Grant R1 to U1 with admin option;

P155-7

有职工表和部门表

1. 用户王明对两个表有SELECT权限。

Grant select on table 职工，部门 to 王明；

1. 用户李勇对两个表有INSERT和DELETE权限。

Grant insert, delete on table 职工，部门 to 李勇；

1. 每个职工只对自己的纪录有SELECT权限。

Grant select on table 职工 when user()=name to all;

1. 用户刘星对职工表有SELECT权限，对工资字段具有更新权限。

Grant select, update(工资) on table 职工 to 刘星；

1. 用户张新具有修改这两个表的结构的权限。

Grant alter table on table 职工，部门 to 张新；

1. 用户周平具有对两个表的所有权限，并具有给其他用户授权的权限。

Grant all privileges on table 职工，部门 to 周平with grant option;

1. 用户杨兰具有从每个部门职工中SELECT最高工资、最低工资、平均工资的权限，他不能查看每个人的工资。

Create view 部门工资 AS

Select 部门.名称，MAX（工资），MIN(工资)，AVG（工资）

From 职工，部门

Where 职工.部门号=部门.部门号

Group by 职工.部门号;

Grant select on table 部门工资 to 杨兰；

P173-6

有职工表和部门表

1. 定义每个模式的主码
2. 定义参照完整性
3. 定义职工年龄不得超过60岁。

Create table dept (

Deptno int primary key,

Deptname varchar(10),

Manager varchar(10),

Phonenumber char(12)

);

Create table emp (

Empno int primary key,

Ename varchar(10),

Age int,

Job varchar(9),

Sal float,

Deptno int,

Constraint c1 check (age<=60),

Constraint fk\_deptno foreign key (deptno) references dept (deptno)

);

P173-8

有male表和female表

建立一断言，将来宾的人数限制在50人以内。

Create table male (

Serialnumber smallint primary key,

Name char(8),

Age smallint,

Occupation char(20)

);

Create table female (

Serialnumber smallint primary key,

Name char(8),

Age smallint,

Occupation char(20)

);

Create assertion party

Check ((select count (\*) from male)+(select count (\*) from female)<=50);

存储过程：

delimiter //  
CREATE PROCEDURE pro\_cust(in rname varchar(20),in pass varchar(30))  
BEGIN  
DECLARE rn VARCHAR(20);  
DECLARE ps varchar(30);  
# DECLARE n int DEFAULT 0;  
select name into rn from `user`;  
SELECT PASSWORD into ps FROM `user`;  
IF (rn=rname and ps=pass) THEN  
SELECT '登录成功！';  
ELSE if (ps!=pass) THEN  
SELECT '密码错误！';  
ELSE   
SELECT '姓名错误！';  
End IF;  
END if;  
END //  
delimiter ;

delimiter //  
CREATE PROCEDURE pro\_cust()  
BEGIN  
DECLARE rid VARCHAR(20);

DECLARE rname VARCHAR(20);  
DECLARE done int DEFAULT FALSE;   
Declare c\_rzb\_select cursor for select 入住单号，顾客姓名 from 入住备份表;  
DECLARE CONTINUE HANDLER for not FOUND SET done=true;  
Open c\_rzb\_select;  
READ\_loop:LOOP  
 FETCH c\_rzb\_select INTO rid,rname;  
 IF done THEN  
  LEAVE read\_loop;  
 END IF;  
Select \* from 入住备份表 where 入住单号=rid;  
End Loop;  
Close c\_rzb\_select;  
End //  
delimiter ;

# 数据库简单笔记

### 第一章 模式 数据库 数据

**1. 外模式**：保证了数据的逻辑独立性

**2. 内模式**：保证了数据的物理独立性

Eg：要保证数据库的逻辑独立性，需要修改的是模式与外模式。

数据的逻辑独立性是指模式改变，外模式和应用程序不变。

1. 设有关系模式 EMP（职工号，姓名，年龄，技能）。假设职工号唯一，每个职工有多项技能，则EMP 表的主码是（职工号，技能）。
2. 在数据库三级模式中，**模式的个数**只有1个。
3. 三级模式是对数据库中数据的三个抽象级别，两级映象是在DBMS内部实现这三个抽象层次的联系和转换。
4. 模式SCHEMA、表TABLE、视图VIEW、检索INDEX
5. 数据定义语言–DDL （建立CREATE、修改ALTER、删除DROP）

数据操纵语言–DML （插入INSERT、更新UPDATE、删除DELETE）

数据控制语言–DCL （授权 GRANT 、拒绝DENY、 撤销REVOKE）

数据查询语言–DQL （查找SELECT）

例子：数据库管理系统能实现对数据库中数据的查询、插入、修改和删除等操作的数据库语言称为数据操纵语言（ DML ）。

**8.** **数据库的特点**是：

A、数据结构化 ,数据之间具有联系,面向整个系统。

B、数据的共享性高,冗余度低,易扩充 。

C、数据独立性高 。

D、数据由DBMS统一管理和控制。

### 第二章 键 关系代数

**1.** 数据库中的键

A、超键：由超键的定义可知，在学生表中含有学号或者身份证号的任意组合都可以唯一标识一个学生，那么它们就是此表的超键。

Eg.（学号）、（身份证号）、（学号，姓名）、（身份证号，性别）等。

B、候选键：候选键属于超键，它是最小的超键，就是说如果再去掉候选键中的任何一个属性它就不再是超键了。

Eg.对于（学号、姓名）来说，去掉姓名后仍是一个超键，那么它就不是候选键。其中，学生表中的候选键为：（学号）、（身份证号），主属性就是学号、身份证号。

C、主键：主键就是候选键里面的一个，用户可以选择。

Eg.那么在这里我们选择（学号）作为学生表的主键。

D、超关键字：二维表中，能够惟一确定记录的一个字段或几个字段的组合被称为“超关键字”。“超关键字”虽然能唯一确定记录，但是它所包含的字段可能是有多余的。

1. **数据库完整性**：是指保护数据库中数据的正确性、有效性和相容性，防止错误数据进入数据库造成无效操作。
2. 关系数据模型的**完整性约束**分为四类：

A、域完整性约束：把属性限制在一个有限的集合中

B、实体完整性约束：主码不能取空值

C、参照完整性约束（或引用完整性约束）：外码存在、取空值

D、用户自定义的完整性约束：制定取值范围

**4.** 并、差、笛卡尔积、选择、投影（∩，—，×，σ，π）为5中基本运算。

**5.** **逻辑独立性**：在关系数据库系统中，当关系的类型改变时，用户程序也可以不变。

**6.** 四种连接：

A、θ连接：在关系代数中，从两个关系的笛卡尔积中选取它们属性间满足一定条件的元组的操作。

Eg.记作S⋈R(AθB)

其中：A为包含R中的属性的表达式，B为包含S中的属性的表达式，θ通常为关系比较符。

B、自然连接：从两个关系的广义笛卡尔积中选取在相同属性列B上取值相等的元组，并去掉一个重复的属性列B 。

C、广义笛卡尔积：属性A+B，元组A\*B。

D、等值连接：两个值相等，但是不一定是同一属性。

**7.** θ在“=”时的连接为等值连接。

### 第三章 SQL语句

**1.** **SQL**语句

**GROUP BY** group\_by\_expression --分组统计

**HAVING** search\_condition --满足条件的组才输出

**ORDER BY** order\_expression [ASC （升）| DESC（降）] --对查询结果排序

1. ESCAPE ‘＼’ 表示“ ＼” 为换码字符

例子：

查询DB\_Design课程的课程号和学分。

**SELECT** Cno，Ccredit

**FROM** Course

**WHERE** Cname **LIKE** 'DB\\_Design' **ESCAPE** ' \ '

1. 外连接：

左外连接 ※=

右外连接 =※

1. 并操作：

**UNION**：将多个查询结果合并起来时，系统自动去掉重复元组。

**UNION ALL**：将多个查询结果合并起来时，保留重复元组。

1. SQL关于基本表的操作

A、定义基本表（**TABLE**）：

**CREATE TABLE** <表名>

（<列名> <数据类型>[ <列级完整性约束条件> ]

[，<列名> <数据类型>[ <列级完整性约束条件>] ]

[，<表级完整性约束条件> ] ）；

B、修改基本表：

**ALTER** **TABLE** <表名>

[ **ADD** <新列名> <数据类型> [ 完整性约束 ] ]

[ **DROP** <完整性约束名> ]

[ **ALTER COLUMN**<列名> <数据类型> ]；--------**COLUMN：列**

C、删除基本表：

**DROP** **TABLE** <表名> [**RESTRICT** | **CASCADE**];

**RESTRICT**：删除表是有限制的。

a.欲删除的基本表不能被其他表的约束所引用

b.如果存在依赖该表的对象，则此表不能被删除

**CASCADE**：删除该表没有限制。

在删除基本表的同时，相关的依赖对象一起删除

D、在xscj库的xsqk表中增加一个“email”列，类型为varchar(30)；

**Alter** **table** xsqk

**Add** email vachar(30)

1. 关于单个元组的操作

A、插入单个元组：

**INSERT** **INTO** <表名>[(<列名1>[,<列名2>…])] **VALUES**(<值>)

B、修改数据：

**UPDATE** <表名>

**SET** <列名>=<表达式> [,<列名>=<表达式>][WHERE <条件>]

C、删除数据：

**DELETE FROM** <表名>

[WHERE <条件>]

1. 创建索引：

**CREATE** [UNIQUE] [CLUSTERED] **INDEX** <索引名> ON <表名> (<列名> [次序] [{,<列名>}] [次序]…)-------UNIQUE:唯一索引 CLUSTERED:聚簇索引

1. 显然在一个基本表上最多只能建立一个聚簇索引。
2. 创建视图：

**CREATE** VIEW view\_name [(column\_name [,...n])]

**AS**

select\_statement

**[WITH CHECK OPTION]**

1. **视图**作用：

A、数据库视图隐藏了数据的复杂性。

B、数据库视图有利于控制用户对表中某些列的访问。

C、数据库视图使用户查询变得简单。

**11.** 在视图上不能完成的操作是在视图上定义新表。

### 第四章 安全性

**1.** 自主存取控制

通过 SQL 的 GRANT 语句和 REVOKE 语句实现

**GRANT** <权限>[,<权限>]...

[ON <对象类型> <对象名>]

**TO** <用户>[,<用户>]...

[WITH GRANT OPTION]; --可传播该权限

1. 回收授权

**REVOKE** <权限>[,<权限>]...

[ON <对象类型> <对象名>]

FROM <用户>[,<用户>]...

[CASCADE]; --级联操作

1. 角色创建

**CREATE** ROLE <角色名>

### 第五章 触发器

**1.** 建立触发器：

**CREATE** **TRIGGER** 触发器名称

**ON** 表名

{ BEFOR | AFTER | INSTEAD OF }

{ [ INSERT ] [ , ] [ DELETE ] [ , ] [UPDATE ] }

AS

SQL 语句 [ ... n ]

### 第六章 数据库完整性

**1.** 数据库完整性

完整性约束的例子：

Eg.建立学生登记表Student，要求学号在900~999之间，姓名不能取空值，年龄小于30，性别只能是“男”或“女”。

**CREATE** **TABLE** Student

( Sno **NUMERIC**(6)

**CONSTRAINT** StudentKey **PRIMARY KEY**

**CONSTRAINT** C1 **CHECK** (Sno **BETWEEN** 900 **AND** 999),

Sname **CHAR**(20)

**CONSTRAINT** C2 **NOT NULL**,

Sage **NUMERIC**(3)

**CONSTRAINT** C3 **CHECK** (Sage < 30),

Ssex **CHAR**(2)

**CONSTRAINT** C4 **CHECK** (Ssex IN ( '男'，'女'))

　　)

1. 修改表中的完整性限制

先删除原来的约束条件，再增加新的约束条件。

Eg.修改表Student中的约束条件，要求学号改为在0000~9999之间，年龄由小于30改为小于40。

**ALTER** **TABLE** Student **DROP** **CONSTRAINT** C1;

**ALTER TABLE** Student **ADD CONSTRAINT** C1 **CHECK** (Sno BETWEEN 0000 AND 9999);

**ALTER** **TABLE** Student **DROP CONSTRAINT** C3;

**ALTER** **TABLE** Student **ADD CONSTRAINT** C3 **CHECK** (Sage < 40);

1. 关于<规则>的SQL语句

A、创建语法格式：

**create** rule 规则名

as 条件表达式

Eg.在student数据库中定义1个规则，限制联系电话的格式为“(区号)电话号码”，其中区号为4位，电话号码至少为6位 。

Use student

Go

Create rule 电话号码格式

As @电话号码 like ' (\_ \_ \_ \_)\_ \_ \_ \_ \_ \_ % '

B、绑定语法规格式：

Exec sp\_bindrule 规则名,‘表名.列名’

Eg：将名为“ro\_电话格式”的规则绑定到student表的“联系电话”列上。

Exec sp\_bindrule ro\_电话格式,‘student.联系电话’

C、用系统存储过程解除绑定格式：

exec sp\_unbindrule‘表名.列名’

D、删除规则语法格式：

drop rule 规则名[,…n]

例3：删除xscj库中的规则。

drop rule ro\_电话格式

总体示例：

CREATE RULE age\_rule as @Sage>=15 and @Sage<=50

EXEC sp\_bindrule 'age\_rule', 'Student.Sage'

EXEC sp\_unbindrule 'Student.Sage'

DROP RULE age\_rule

规范小结：

规则对象

创建规则:create rule 规则名 as 条件表达式

绑定规则:[exec] sp\_bindrule 规则名, ‘表名.列名’

解除规则:[exec] sp\_unbindrule ‘表名.列名’

删除规则:drop rule 规则名[，…n]

### 第七章 范式

**1.** 关系的属性是原子的（不可分的），这样的关系模式就属于1NF。

若R属于1NF，且每个非主属性完全函数依赖于关键字，则R属于2NF。

若R属于2NF，且R的每个非主属性都不传递依赖于关键字，则R属于3NF。

### 第八章 设计数据库

1.需求分析

2.概念结构设计：绘制E-R图

3.逻辑结构设计：将E-R图转换为关系表、设计关系模式。

4.数据库的物理设计：存取方法设计

5.数据库的实施和维护

### 第九章 存储过程

1.创建存储过程

**CREATE** **PROC | PROCEDURE** procedure\_name

[ @parameter ] data\_type

[ = default ] [ OUT | OUTPUT ] ] [ ,...]

[ WITH ENCRYPTION ]

**AS**

**BEGIN**

<sql\_statement> [ ,... ]

**END**

--执行存储过程

**[EXECUTE | EXEC]** proc\_name

Eg.通过某一位用户的姓名的一部分查知该用户的信息

**CREATE PROC** pr\_queryphone @name char(10)

**AS**

**SELECT** \*

**FROM** user

**WHERE** username **like** ‘%’+@name+‘%’

Eexe pr\_queryname ‘张’

### 第十章 数据库恢复

**1.** 事务的定义方式

显式定义方式：

**BEGIN** **TRANSACTION**

SQL 语句1

...

**COMMIT** （ 或 ROLLBACK)

1. 事务的四个特性

**原子性**：原子性是指事务是数据库的逻辑工作单位，事务中包括的诸操作要么都做，要么都不做。

**一致性**：事务执行的结果必须是使数据库从一个一致性状态变到另一个一致性状态。

**隔离性**：一个事务的执行不能被其他事务干扰。

**持久性**：一个事务一旦提交，它对数据库中数据的改变就应该是永久性的。

1. **介质故障**：若系统在运行过程中，由于某种硬件故障，使存储在外存上的数据部分损失 或全部损失，这种情况称为介质故障。
2. 登记的次序严格按并发事务执行的时间次序。

B、 必须先写日志文件,后写数据库。

**4.** 数据库恢复的基础是利用转储的冗余数据。这些转储的冗余数据是指日志文件、数据库后备副本。

第十一章 并发控制

**1.** 问题的产生

A、**丢失修改**：

两事务读出同一数据并修改，先写回的数据修改丢失。

B、**读出脏数据**：

事务T1修改某一数据，事务T2读取同一数据；

事务T1由于某种原因被撤消，则T2读到的就是“脏”数据。

C、**不可重复读**：

事务T1读取某一数据，事务T2读取并修改了同一数据；

事务T1为了对读取值进行校对再读此数据，得到了不同的结果。

1. **封锁**

A、**排它锁**（简记为X锁，又称写锁）：若事务T对数据对象R加上X锁，则只允许T读/写R，禁止其它事务对R加任何锁，相应地其它事务就无法读/写对象R。

B、**共享锁**（简记为S锁，又称读锁）：若事务T对数据对象R加上S锁，则T可以读R，但不可以写R，且其它事务可以对R加S锁、但禁止加X锁。这保证了事务T在释放R的S锁之前，其它事务只可以读R，不可以修改R。

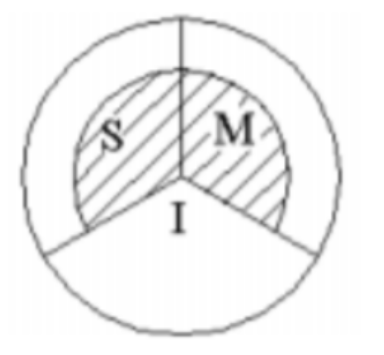
1. **两段锁的含义**：

1）第一阶段（扩展阶段）：所有事务对数据加锁，但不能解锁；

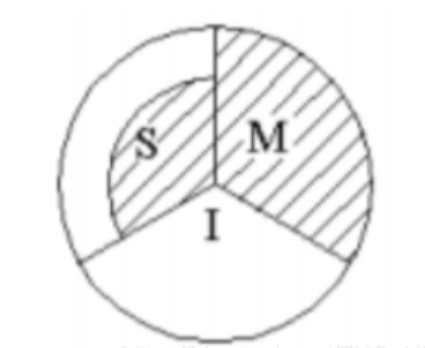
2）第二阶段（收缩阶段）：所有事务对数据解锁，但不能加锁。

补充

1. 最小关系系统：



最完备的关系系统：



1. 数据库的网状模型应满足的条件是（ ）

A．允许一个以上结点无双亲，也允许一个结点有多个双亲

1. 如何构造出一个合适的数据逻辑结构是（关系数据库规范化理论）主要解决的问题。