**第一章 绪论**

**6 题略**

**（1）数据定义功能**

**（2）数据组织、存储和管理  
 （3）数据操纵**

**（4）数据库的事务管理和运行管理**

**并发控制，故障恢复，完整性，安全性**

**（5）数据库的建立和维护**

**初始数据的输入、故障恢复、数据转换、转储、数据重组、性能监视和分析**

**（6）其他功能**

**通信，同构或异构数据库之间数据转换、数据库互访和互操作等功能。**

**17 题略**

**答：数据与程序的逻辑独立性：当模式改变时（例如增加新的关系、新的属性、改变属性的数据类型等），由数据库管理员对各个外模式／模式的映像作相应改变，可以使外模式保持不变。应用程序是依据数据的外模式编写的，从而应用程序不必修改，保证了数据与程序的逻辑独立性，简称数据的逻辑独立性。数据与程序的物理独立性：当数据库的存储结构改变了，由数据库管理员对模式／内模式映像作相应改变，可以使模式保持不变，从而应用程序也不必改变。保证了数据与程序的物理独立性，简称数据的物理独立性。数据库管理系统在三级模式之间提供的两层映像保证了数据库系统中的数据能够具**

**有较高的逻辑独立性和物理独立性。**

**第二章 关系数据库**

6 题略，

(1) 

(2) 

(3)  

(4) －  

(5)

1) 

(2) 

(3)  





(4) －  



域关系演算省略。

(5) 

域关系演算省略

**第三章 关系数据库标准语言SQL**

**参见第三章练习.ppt**

**第六章 关系数据理论**

**2．建立一个关于系、学生、班级、学会等诸信息的关系数据库。**

**学生：学号、姓名、出生年月、系名、班号、宿舍区。**

**班级：班号、专业名、系名、人数、入校年份。**

**系：系名、系号、系办公地点、人数。**

**学会：学会名、成立年份、办公地点、人数。**

**语义如下：一个系有若干专业，每个专业每年只招一个班，每个班有若干学生。一个系的学生住在同一宿舍区。每个学生可参加若干学会，每个学会有若干学生。学生参加某学会有一个入会年份。**

**请给出关系模式，写出每个关系模式的极小函数依赖集，指出是否存在传递函数依赖，对于函数依赖左部是多属性的情况讨论函数依赖是完全函数依赖，还是部分函数依赖。指出各关系模式的候选码、外部码，有没有全码存在?**

**解：(1)关系模式如下：**

**学生：S(Sno，Sname，Sbirth，Dept，Class，Rno)**

**学号，学生名，出生日期，系名，班号，宿舍号**

**班级：C(Class，Pname，Dept，Cnum，Cyear)**

**班号，专业名，系名，班级人数，入校年份**

**系：D(Dept，Dno，Office，Dnum)**

**系名，系号，办公地点，人数**

**学会：M(Mname，Myear，Maddr，Mnum)**

**学会名、成立年份、办公地点、人数**

**学生-学会SP（Sno，Mname，SPYear）**

**学号，学会名，入会年份**

**(2)每个关系模式的最小函数依赖集如下：**

**A、学生S (Sno，Sname，Sbirth，Dept，Class，Rno)**

**传递依赖如下：**

**由于Sno🡪Dept，而Dept→＼ Sno，Dept🡪Rno**

**所以Sno与Rno之间存在着传递函数依赖**

**由于Class🡪Dept，而Dept→＼ Class，Dept 🡪Rno**

**所以Class与Rno之间存在着传递函数依赖。**

**由于Sno🡪Class，而Class→＼ Sno， Class🡪Dept**

**所以Sno与Dept之间存在着传递函数依赖。**

**最小函数依赖集如下:{Sno🡪Sname，Sno🡪Sbirth，Sno🡪Class，Class🡪Dept，DEPT🡪Rno}**

**B、班级C(Class，Pname，Dept，Cnum，Cyear)的最小函数依赖集如下:**

**Class🡪Pname，Class🡪Cnum，Class🡪Cyear，Pname🡪Dept，**

**（Pname，Cyear）🡪 Class**

**由于Class🡪Pname，而Pname→＼ Class，Pname🡪Dept**

**所以C1ass与Dept之间存在着传递函数依赖。**

**C、系D(Dept，Dno，Office，Dnum)的最小函数依赖集如下：**

**Dept🡪Dno，Dno🡪Dept，Dno🡪Office，Dno🡪Dnum**

**根据上述函数依赖可知，Dept与Office，Dept与Dnum之间不存在传递依赖。**

**D、学会M(Mname，Myear，Maddr，Mnum)的最小函数依赖集如下：**

**Mname🡪Myear，Mname🡪Maddr，Mname🡪Mnum**

**该模式不存在传递依赖。**

**学生-学会SP（Sno，Mname，SPYear）的最小函数依赖集如下：**

**（Sno，Mname）🡪SPYear**

**对于（Sno，Mname）🡪SPYear，（Pname，Cyear）🡪 Class**

**不存在部分函数依赖**

**(3)各关系模式的候选码、外部码，全码如下：**

**A、学生S候选码：Sno；外部码：Dept、Class；无全码**

**B、班级C候选码：Class和（Pname，Cyear）；外部码：Dept；无全码**

**C、系D候选码：Dept和Dno；无外部码；无全码**

**D、学会M候选码：Mname；无外部码；无全码**

**学生-学会SP候选码：（Sno，Mname）；外部码: Sno，Mname；无全码**

**7．下面的结论哪些是正确的? 哪些是错误的? 对于错误的请给一个反例说明之。**

**（1）任何一个二目关系是属于3NF。**

**答：正确。因为关系模式中只有两个属性，所以无传递。**

**（2）任何一个二目关系是属于BCNF.**

**答:正确。按BCNF的定义，若X🡪Y,且Y不是X的子集时，每个决定因素都包含码，对于二目关系决定因素必然包含码。详细证明如下：（任何二元关系模式必定是BCNF）。**

**证明：设R为一个二目关系R(A1，A2)，则属性A1和A2之间可能存在以下几种依赖关系：**

**A、A1🡪A2，但A2🡪A1，则关系R的码为A1，决定因素都包含码，所以，R是BCNF。**

**B、A1🡪A2，A2🡪A1，则关系R的码为A2，所以决定因素都包含码，R是BCNF。**

**包含码。R是BCNF。C、R的码为(A1，A2)（即A1 🡪A2，A2 🡪A1），决定因素都**

**（3）任何一个二目关系是属于4NF.**

**答:正确。因为只有两个属性，所以无非平凡的多值依赖。**

**(4 )当且仅当函数依赖A🡪B在R上成立，关系R(A,B,C)等于其投影R1(A,B)和R2(A,C)的连接。**

**答：错误。根据定理6.5，当函数依赖A🡪C时，也是无损连接。**

**(5)若R.A🡪R.B，R.B🡪R.C，则R.A🡪R.C**

**答：正确。传递律**

**(6)若R.A🡪R.B, R.A🡪R.C,则R.A🡪R.(B,C)**

**答：正确。合并规则**

**(7) 若R.B🡪R.A，R.C🡪R.A，则R.(B,C)🡪R.A**

**答：正确。因为R.(B,C)🡪R.B,又R.B🡪R.A，根据传递律，有R.(B,C)🡪R.A**

**(8)若R.(B,C)🡪R.A，则R.B🡪R.A，R.C🡪R.A**

**答：错误。反例，如(SNO,CNO) 🡪GRADE，但是没有SNO 🡪GRADE，CNO 🡪GRADE**

**第七章 数据库设计**

8 题略

产品

储存

零件

仓库

材料

制造

存放

存储

m

n

m

n

m

n

m

1

以上E－R图省略了实体和联系的属性！

**第十章 数据库恢复技术**

4 题略

（1）REDO的事务:T1, T3 UNDO的事务：T2,T4

(2) REDO的事务：T1 UNDO的事务: T2，T3

（3）REDO的事务：T1 UNDO的事务: T2,T3

(4) REDO的事务：T1 UNDO的事务：T2

5 题略：

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 前象 | | |  |  | 后象 | | |
|  | A | B | C |  |  | A | B | C |
|  | 0 | 0 | 0 | 1 | T1开始 | 0 | 0 | 0 |
| T1 | 0 |  |  | 2 | T1写A，A=10 | 10 |  |  |
| T2 |  |  |  | 3 | T2开始 |  |  |  |
|  |  | 0 |  | 4 | T2写B,B=9 |  | 9 |  |
| T1 |  |  | 0 | 5 | T1写C,C=11 |  |  | 11 |
| T1 |  |  |  | 6 | T1提交 | 10 |  | 11 |
| T2 |  |  | 11 | 7 | T2写C，C=13 |  |  | 13 |
| T3 | 10 | 9 |  | 8 | T3开始 | 10 | 9 |  |
| T3 | 10 |  |  | 9 | T3写A,A=8 | 8 |  |  |
|  |  |  |  | 10 | T2：回滚 |  | 0 | 11 |
| T3 |  | 0 |  | 11 | T3:写B,B=7 |  | 7 |  |
|  |  |  |  | 12 | T4开始 |  |  |  |
|  |  |  |  | 13 | T3提交 | 8 | 7 |  |
|  |  |  | 11 | 14 | T4:写C,C=12 |  |  | 12 |

每个故障点的恢复步骤：

|  |  |  |  |
| --- | --- | --- | --- |
| 习题 | 故障点 | 恢复步骤及结果 | |
| 5(1) | 14后 | UNDO:T4,T2 | 14.C回到前象,C=11 |
|  |  |  | 7. C回到前象C=11 |
|  |  |  | 4.B回到前象B=0 |
|  |  | REDO:T1 | 2.A赋值时刻2的后象A=10 |
|  |  |  | 5.C赋值时刻5的后象C=11 |
|  |  | REDO:T3 | 9.A赋值时刻9的后象A=8 |
|  |  |  | 11.B赋值时刻11的后象B=7 |
|  |  | 结果 | A=8,B=7,C=11 |
| 5(2) | 12后 | UNDO:T4，T3，T2 |  |
|  |  |  | 11.B赋值到时刻11的前象B=0 |
|  |  |  | 9.A回到时刻8的前象,A=10 |
|  |  |  | 7.C赋值到时刻7的前象C=11 |
|  |  |  | 4.B回到时刻4的前象,B=0 |
|  |  | REDO:T1 | 2.A赋值时刻2的后象A=10 |
|  |  |  | 5.C赋值时刻5的后象C=11 |
|  |  | 结果 | A=10,B=0,C=11 |
| 5(3) | 10后 | UNDO:T3,T2 |  |
|  |  |  | 9.A回到时刻9的前象,A=10 |
|  |  |  | 7.C赋值到时刻7的前象C=11 |
|  |  |  | 4.B回到时刻4的前象,B=0 |
|  |  | REDO:T1 | 2.A赋值时刻2的后象A=10 |
|  |  |  | 5.C赋值时刻5的后象C=11 |
|  |  | 结果 | A=10,B=0,C=11 |
| 5（4） | 9后 | UNDO:T3,T2 |  |
|  |  |  | 9.A回到时刻9的前象,A=10 |
|  |  |  | 7.C赋值到时刻7的前象C=11 |
|  |  |  | 4.B回到时刻4的前象,B=0 |
|  |  | REDO:T1 | 2.A赋值时刻2的后象A=10 |
|  |  |  | 5.C赋值时刻5的后象C=11 |
|  |  | 结果 | A=10,B=0,C=11 |
| 5(5) | 7后 | UNDO:T2 |  |
|  |  |  |  |
|  |  |  | 7.C赋值到时刻7的前象C=11 |
|  |  |  | 4.B回到时刻4的前象,B=0 |
|  |  | REDO:T1 | 2.A赋值时刻2的后象A=10 |
|  |  |  | 5.C赋值时刻5的后象C=11 |
|  |  | 结果 | A=10,B=0,C=11 |
| 5(6) | 5后 | UNDO:T2,T1 | 5.C赋值到时刻5的前象C=0 |
|  |  |  | 4.B赋值到时刻4的前象B=0 |
|  |  |  | 2.A赋值时刻2的前象A=0 |
|  |  | 结果 | A=0,B=0,C=0 |

**第十一章 并发控制**

9题略

（1）正确的可能结果有2，4，8，16，

T1T2T3 结果A=16

T1T3T2 结果A=8

T2T1T3 结果A=4

T2T3T1 结果A=2

T3T2T1 结果A=2

T3T1T2 结果A=4

(2)

T1 T2 T3

Slock A

Y = A=0

Unlock A

Xlock A

Slock A

A=Y+2 等待

写回A(=2) 等待

Unlock A 等待

Y=A=2

Unlock A

Xlock A

Slock A

A＝Y\*2 等待

写回A(=4) 等待

Unlock A 等待

Y＝A＝4

Unlock A

Xlock A

A＝Y\*\*2

**写回A(=16)**

Unlock A

(3)

T1 T2 T3

Slock A

Y = A=0

Unlock A

Slock A

Y=A=0

Xlock A

等待 Unlock A

A=Y+2

写回A(=2) Slock A

Unlock A 等待

Y=A=2

Unlock A

Xlock A

Xlock A

等待

等待 A=Y\*\*2

等待 写回A（＝4）

等待 Unlcok A

A＝Y\*2

写回A(=0)

Unlock A

（4）

T1 T2 T3

Slock A

Y = A=0

Xlock A

A=Y+2 Slock A

写回A(=2) 等待

Unlock A 等待

Y=A=2

Xlock A

Unlock A 等待 Slock A

A=Y\*2 等待

写回A(=4) 等待

Unlock A 等待

Y＝A＝4

Unlock A

Xlock A

A＝Y\*\*2

**写回A(=16)**

Unlock A

Unlock A

5)

T1 T2 T3

Slock A

Y = A=0

Slock A

Y=A=0

Xlock A

等待

Xlock A

等待

Slock A

Y=A=0

Xlock A

等待

**第四章 数据库安全性**

**6**

**① grant all privileges on TABLE 学生，班级 to U1 with grant option;**

**② GRANT SELECT,UPDATE(家庭住址) ON Table 学生 TO U2;**

**③ GRANT SELECT ON TABLE 班级 TO PUBLIC**

**④ CREATE ROLE R1;**

**GRANT SELECT,UPDATE ON 学生 TO R1;**

**⑤ grant R1 to U1 with ADMIN option;**

**7．职工（职工号，姓名，年龄，职务，工资，部门号）**

**部门（部门号，名称，经理名，地址，电话号）**

**请用SQL的GRANT 和REVOKE语句(加上视图机制)完成以下授权定义或存取控制功能:**

**( a ）用户王明对两个表有SELECT 权力。**

**GRANT SELECT ON TABLE 职工,部门 TO 王明**

**( b ）用户李勇对两个表有INSERT 和DELETE 权力。**

**GRANT INSERT,DELETE ON TABLE 职工,部门 TO 李勇**

**( c ) 每个职工只对自己的记录有SELECT 权力。**

**GRANT SELECT ON TABLE 职工 WHEN USER()=NAME TO ALL;**

**需要DBMS支持**

**( d ）用户刘星对职工表有SELECT 权力，对工资字段具有更新权力。**

**GRANT SELECT,UPDATE(工资) ON TABLE职工 TO 刘星**

**( e ）用户张新具有修改这两个表的结构的权力。**

**GRANT ALTER TABLE ON TABLE 职工,部门 TO 张新;**

**( f ）用户周平具有对两个表所有权力（读，插，改，删数据），并具有给其他用户授权的权力。**

**GRANT ALL PRIVILIGES ON TABLE 职工,部门**

**TO 周平**

**WITH GRANT OPTION;**

**( g ）用户杨兰具有从每个部门职工中SELECT 最高工资、最低工资、平均工资的权力，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;**

**第五章 数据库完整性**

7 答： 对于违反实体完整性和参照完整的的操作，通常采取拒绝执行；对于违反参照完整的的操作，参照课本P161 表5.1

|  |  |  |
| --- | --- | --- |
| 被参照表（如student） | 参照表（如SC） | 违约处理 |
| 可能破坏参照完整性 | 插入元组 | 拒绝 |
| 可能破坏参照完整性 | 修改外码 | 拒绝 |
| 删除元组 | 可能破坏参照完整性 | 拒绝/级联删除/置空 |
| 修改主码 | 可能破坏参照完整性 | 拒绝/级联删除/置空 |

8

**CREATE TABLE Female**

**(**

**SerialNumber smallint PRIMARY KEY,**

**Name CHAR(10) NOT NULL,**

**Age smallint NOT NULL,**

**RegisterDate DateTime NOT NULL**

**);**

**CREATE TABLE Male**

**(**

**SerialNumber smallint PRIMARY KEY,**

**Name CHAR(10) NOT NULL,**

**Age smallint NOT NULL,**

**RegisterDate DateTime NOT NULL**

**);**

**CREATE ASSERTION PARTY**

**CHECK( (**

**(SELECT COUNT(\*) from Male) +**

**(Select COUNT(\*) FROM Female)**

**)<=50);**

6 题略。

**CREATE TABLE DEPT**

**(Deptno NUMBER(2),**

**DeptnameVARCHAR(10),**

**ManagerVARCHAR(10),**

**PhoneNumberChar(12),**

**CONSTRAINT PK\_SC PRIMARY KEY(Deptno));**

**CREATE TABLE EMP**

**(Empno NUMBER(4),**

**Ename VARCHAR(10),**

**Age NUMBER(2),**

**CONSTRAINT C1 CHECK(Aage<=60),**

**Job VARCHAR(9),**

**Sal NUMBER(7,2),**

**DeptnoNUMBER(2),**

**CONSTRAINT FK\_DEPTNO FOREIGNKEY(Deptno) REFERENCES DEPT(Deptno))；**