

作业集合

4. 举出适合用文件系统而不是数据库系统的应用例子，以及适合用数据库系统的应用例子。

文件系统适合数据量小、用户单一、共享需求低、逻辑简单的场景（无需复杂管理，直接存储即可）：

1. 个人文档管理：比如电脑里的 Word 文档、Excel 表格（自己用，数据独立，无需多人共享 / 复杂查询）；
2. 小型单机程序的配置存储：比如手机 App 的本地设置文件（仅存储少量键值对，无需多用户访问）；
3. 临时数据存储：比如程序运行中生成的日志文件（按行追加，无需修改历史记录，查询需求简单）；
4. 简单的个人记账工具：比如本地 Excel 版的记账表（只有自己维护，数据关系简单，无需多用户协作）。

数据库系统适合数据量大、多用户共享、逻辑复杂、需保证一致性 / 安全性的场景（需要统一管理、高效查询、事务支持）：

1. 电商平台：比如淘宝的商品、订单、用户数据（多用户同时访问，需保证订单的原子性、库存一致性）；
2. 银行系统：比如账户、转账记录（必须保证事务安全，避免“转账成功但余额未扣”的问题）；
3. 学校教务系统：比如学生信息、课程表、成绩（多部门共享数据，需支持复杂查询，如“查询某系所有学生的必修课成绩”）；
4. 社交平台：比如用户动态、好友关系

● 数据库系统的三级模式结构是什么，这种结构的优点是什么？

数据库的三级模式结构：

内模式 → 模式 → 外模式

这种结构的优点为：

- 1、通过将应用，存储，逻辑相分离来实现数据的独立性
- 2、支持“按需访问”，防止在应用层暴露太多非必要信息
- 3、便于数据库维护和升级，便于数据库的灵活扩展

15. 试述数据库系统三级模式结构,并说明这种结构的优点是什么?

答:

数据库系统的三级模式结构由外模式、模式和内模式组成(参考《概论》图 1.16)。

外模式,亦称子模式或用户模式,是数据库用户能够看见和使用的局部数据的逻辑结构和特征的描述,是数据库用户的数据视图。

模式,亦称逻辑模式,是数据库中全体数据的逻辑结构和特性的描述,是所有用户的公共数据视图。模式描述的是数据的全局逻辑结构。外模式通常是模式的子集。

内模式,亦称存储模式,是数据在数据库系统内部的表示,即对数据的物理结构和存储方式的描述。

为了能够在内部实现这三个抽象层次的联系和转换,数据库系统在这三级模式之间提供了两级映像:外模式/模式映像和模式/内模式映像。正是这两级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。

● 定义并理解下列术语,说明它们之间的联系与区别:

(1) 域、笛卡儿积、关系、元组、属性

(2) 主码、候选码、外码

(3) 关系模式、关系、关系数据库

(1) 域:表示某一元素所有的可能取值

笛卡尔积:是两个集合的元素的一对一的所有可能的组合的集合

关系:表示事物之间的联系

元组:表示描述一个事物的所有属性取值的集合

属性:对象的某一个特点被成为该对象的一个属性

(2) 主码:可以唯一确定一个元组的属性或者属性集合

候选码:除了主码之外可以唯一确定某个元组的属性或者属性集合

外码:为参考关系当中的某个属性或者属性集合,可以唯一标定被参考关系当中的某一个元组,即成为外码

(3) 关系模式:是抽象的数据结构,描述一张表的结构,但是不包含具体数据

关系:是关系模式的实例化,是包含实际数据的表

关系数据库:是管理多个关系的软件系统

(1) 域,笛卡儿积,关系,元组,属性

答:

域:域是一组具有相同数据类型的值的集合。

笛卡儿积:给定一组域 D_1, D_2, \dots, D_n , 允许其中某些域是有相同的。这组域的笛卡儿积为

$$D_1 \times D_2 \times \dots \times D_n = \{ (d_1, d_2, \dots, d_n) \mid d_i \in D_i, i=1, 2, \dots, n \}$$

关系:在域 D_1, D_2, \dots, D_n 上笛卡儿积 $D_1 \times D_2 \times \dots \times D_n$ 的子集称为关系,表示为

$$R(D_1, D_2, \dots, D_n)。$$

注意,这里是用较为形式化的方法来定义关系。在第1章中则是用通俗的语言来说明什么是关系,是一种不严格的定义。

元组:关系中的每个元素是关系中的元组。

属性:关系也是一个二维表,表的每行对应一个元组,表的每列对应一个域。由于域可以相同,为了加以区分,必须对每列起一个名字,称为属性(attribute)。

(2) 候选码,主码,外码

答:

候选码:若关系中的某一属性组的值能唯一地标识一个元组,而其子集不能,则称该属性组为候选码(candidate key)。

主码:若一个关系有多个候选码,则选定其中一个为主码(primary key)。

外码:设 F 是基本关系 R 的一个或一组属性,但不是关系 R 的码,如果 F 与基本关系 S 的主码 K 相对应,则称 F 是基本关系 R 的外部码(foreign key),简称外码。

(3) 关系模式,关系,关系数据库

关系模式:关系的描述称为关系模式(relation schema)。它可以形式化地表示为

$$R(U, D, \text{DOM}, F)$$

其中 R 为关系名, U 为组成该关系的属性名集合, D 为属性组 U 中属性所来自的域, DOM 为属性向域的映像集合, F 为属性间数据的依赖关系集合。

关系:见(1),关系是关系模式在某一时刻的状态或内容。关系模式是静态的、稳定的,而关系是动态的、随时间不断变化的,因为关系操作在不断地更新着数据库中的数据。

关系数据库:关系数据库也有型和值之分。关系数据库的型称为关系数据库模式,是对关系数据库的描述,它包括若干域的定义以及在这些域上定义的若干关系模式。关系数据库的值是这些关系模式在某一时刻对应的关系的集合,通常就称为关系数据库。

● 题目由下图给出

6. 设有一个 SPJ 数据库, 包括 S、P、J 及 SPJ 4 个关系模式:

S(SNO,SNAME,STATUS,CITY);

P(PNO,PNAME,COLOR,WEIGHT);

J(JNO,JNAME,CITY);

SPJ(SNO,PNO,JNO,QTY)。

供应商表 S 由供应商代码 (SNO)、供应商姓名 (SNAME)、供应商状态 (STATUS)、供应商所

在城市 (CITY) 组成。

零件表 P 由零件代码 (PNO)、零件名 (PNAME)、颜色 (COLOR)、重量 (WEIGHT) 组成。

工程项目表 J 由工程项目代码 (JNO)、工程项目名 (JNAME)、工程项目所在城市 (CITY) 组成。

供应情况表 SPJ 由供应商代码 (SNO)、零件代码 (PNO)、工程项目代码 (JNO)、供应数量 (QTY) 组成, 表示某供应商供应某种零件给某工程项目的数量为 QTY。

今有若干数据如下:

S 表

SNO	SNAME	STATUS	CITY
S1	精益	20	天津
S2	盛锡	10	北京
S3	东方红	30	北京
S4	丰泰盛	20	天津
S5	为民	30	上海

P 表

PNO	PNAME	COLOR	WEIGHT
P1	螺母	红	12
P2	螺栓	绿	17
P3	螺丝刀	蓝	14
P4	螺丝刀	红	14
P5	凸轮	蓝	40
P6	齿轮	红	30

J 表

JNO	JNAME	CITY
J1	三建	北京
J2	一汽	长春
J3	弹簧厂	天津
J4	造船厂	天津
J5	机车厂	唐山
J6	无线电厂	常州
J7	半导体厂	南京

SPJ 表

SNO	PNO	JNO	QTY
S1	P1	J1	200
S1	P1	J3	100
S1	P1	J4	700
S1	P2	J2	100
S2	P3	J1	400
S2	P3	J2	200
S2	P3	J4	500
S2	P3	J5	400
S2	P5	J1	400
S2	P5	J2	100
S3	P1	J1	200
S3	P3	J1	200
S4	P5	J1	100
S4	P6	J3	300
S4	P6	J4	200
S5	P2	J4	100
S5	P3	J1	200
S5	P6	J2	200
S5	P6	J4	500

试用关系代数、ALPHA 语言、QBE 语言完成如下查询：

(1) 求供应工程 J1 零件的供应商号码 SNO；

(2) 求供应工程 J1 零件 P1 的供应商号码 SNO；

(3) 求供应工程 J1 零件为红色的供应商号码 SNO；

(4) 求没有使用天津供应商生产的红色零件的工程号 JNO；

(5) 求至少用了供应商 S1 所供应的全部零件的工程号 JNO。

$$(1) \sigma_{(JNO='J1')}(\pi_{(SNO, JNO)}(SPJ))$$

$$(2) \text{令 } k = (PNO=P1, JNO=J1) \text{ 则为 } \pi_{SNO}(SPJ \div k)$$

$$(3) \pi_{SNO}(SPJ \div (\sigma_{(PNO='P1', COLOR='红')}(P)))$$

$$(4) \pi_{JNO}(\sigma_{(CITY='天津' \wedge COLOR='红')}(P \bowtie_{(PNO)} SPJ \bowtie_{(JNO)} J))$$

$$(5) \pi_{(JNO, PNO)}(SPJ) \div \pi_{PNO}(\sigma_{(SNO='S1')}(SPJ))$$

ANSWER

(1) 求供应工程 J1 零件的供应商号 SNO。

答：

关系代数： $\pi_{SNO}(\sigma_{JNO='J1'}(SPJ))$

(2) 求供应工程 J1 零件 P1 的供应商号 SNO。

答：

关系代数： $\pi_{SNO}(\sigma_{JNO='J1' \wedge PNO='P1'}(SPJ))$

(3) 求供应工程 J1 红色零件的供应商号 SNO。

答:

关系代数: $\Pi_{SNO}(\Pi_{SNO,PNO}(\sigma_{JNO='J1'}(SPJ)) \bowtie \Pi_{PNO}(\sigma_{COLOR='红'}(P)))$

(4) 求没有使用天津供应商生产的红色零件的工程号 JNO。

答:

关系代数: $\Pi_{JNO}(J) - \Pi_{JNO}(\Pi_{SNO}(\sigma_{CITY='天津'}(S)) \bowtie \Pi_{SNO,PNO,JNO}(SPJ))$

$$\bowtie \Pi_{PNO}(\sigma_{COLOR='红'}(P)))$$

解析:

减法运算中,被减的部分是使用了天津供应商生产的红色零件的所有工程号, $\Pi_{JNO}(J)$ 是全部工程的工程号,两者相减就是没有使用天津供应商生产的红色零件的工程号,包括没有使用任何零件的工程号。

(5) 求至少用了 S1 供应商所供应的全部零件的工程号 JNO。

答:

关系代数: $\Pi_{JNO,PNO}(SPJ) \div \Pi_{PNO}(\sigma_{SNO='S1'}(SPJ))$

解析:

第一部分是所有工程与该工程所用的零件,第二部分是 S1 所供应的全部零件号,对于 SPJ 表中的某一个 JNO,如果该工程使用的所有零件的集合包含了 S1 所供应的全部零件号,则该 JNO 符合本题条件,在除法运算的结果集中。

可以看到,使用关系代数的除法运算概念清晰,语言表达也很简洁。

3. 有两个关系 S(A, B, C, D) 和 T(C, D, E, F), 写出与下列查询等价的 SQL 表达式:

(1) $\sigma_{A=10}(S)$; (2) $\Pi_{A,B}(S)$; (3) $S \bowtie T$; (4) $S \bowtie_{S.C=T.C} T$; (5) $S \bowtie_{A<E} T$; (6) $\Pi_{C,D}(S) \times T$ 。

4. 用 SQL 语句建立第 2 章习题 6 中的 4 个表; 针对建立的 4 个表用 SQL 完成第 2 章习题 6 中的查询。

3. 有两个关系 S(A, B, C, D) 和 T(C, D, E, F), 写出与下列查询等价的 SQL 表达式:

(1) $\sigma_{A=10}(S)$; (2) $\Pi_{A,B}(S)$; (3) $S \bowtie T$; (4) $S \bowtie_{S.C=T.C} T$; (5) $S \bowtie_{A<E} T$; (6) $\Pi_{C,D}(S) \times T$ 。

答:

(1) SELECT * FROM S WHERE A = 10

(2) SELECT DISTINCT A, B FROM S

(3) SELECT A, B, S.C, S.D, E, F FROM S, T WHERE S.C = T.C AND S.D = T.D

(4) SELECT A, B, S.C, S.D, T.C, T.D, E, F FROM S, T WHERE S.C = T.C

(5) SELECT A, B, S.C, S.D, T.C, T.D, E, F FROM S, T WHERE A < E

(6) SELECT S1.C, S1.D, T.C, T.D, E, F FROM T, (select DISTINCT C, D FROM S) AS S1

4. 用 SQL 语句建立第 2 章习题 6 中的 4 个表;针对建立的 4 个表用 SQL 语言完成第 2 章习题 6 中的查询。

答:

建 S 表:

```
S(SNO,SNAME,STATUS,CITY);  
CREATE TABLE S
```

```
(SNO CHAR(3),  
SNAME CHAR(10),  
STATUS CHAR(2),  
CITY CHAR(10));
```

建 P 表:

```
P(PNO,PNAME,COLOR,WEIGHT);  
CREATE TABLE P  
(PNO CHAR(3),  
PNAME CHAR(10),  
COLOR CHAR(4),  
WEIGHT INT);
```

建 J 表:

```
J(JNO,JNAME,CITY);  
CREATE TABLE J  
(JNO CHAR(3),  
JNAME CHAR(10),  
CITY CHAR(10));
```

建 SPJ 表:

```
SPJ(SNO,PNO,JNO,QTY);  
CREATE TABLE SPJ  
(SNO CHAR(3),  
PNO CHAR(3),  
JNO CHAR(3),  
QTY INT);
```

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;
- (2) 找出供应商 S1 的供应情况。

S 表

SNO	SNAME	STATUS	CITY
S1	精益	20	天津
S2	盛锡	10	北京
S3	东方红	30	北京
S4	丰泰盛	20	天津
S5	为民	30	上海

P 表

PNO	PNAME	COLOR	WEIGHT
P1	螺母	红	12
P2	螺栓	绿	17
P3	螺丝刀	蓝	14
P4	螺丝刀	红	14
P5	凸轮	蓝	40
P6	齿轮	红	30

J 表

JNO	JNAME	CITY
J1	三建	北京
J2	一汽	长春
J3	弹簧厂	天津
J4	造船厂	天津
J5	机车厂	唐山
J6	无线电厂	常州
J7	半导体厂	南京

SPJ 表

SNO	PNO	JNO	QTY
S1	P1	J1	200
S1	P1	J3	100
S1	P1	J4	700
S1	P2	J2	100
S2	P3	J1	400
S2	P3	J2	200
S2	P3	J4	500
S2	P3	J5	400
S2	P5	J1	400
S2	P5	J2	100
S3	P1	J1	200
S3	P3	J1	200
S4	P5	J1	100
S4	P6	J3	300
S4	P6	J4	200
S5	P2	J4	100
S5	P3	J1	200
S5	P6	J2	200
S5	P6	J4	500

视图定义: CREATE VIEW J1 AS

SELECT SPJ.SNO, SPJ.PNO, SPJ.QTY
FROM SPJ
WHERE SPJ.JNO = "J1";

(1) SELECT PNO, SUM(QTY) AS TOTAL
FROM J1
GROUP BY PNO;

(2) SELECT * FROM J1 WHERE SNO = "J1";

7. 今有以下两个关系模式:

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:

- (1) 用户王明对两个表有 SELECT 权限。
- (2) 用户李勇对两个表有 INSERT 和 DELETE 权限。
- (3) 每个职工只对自己的记录有 SELECT 权限。
- (4) 用户刘星对职工表有 SELECT 权限, 对工资字段具有更新权限。
- (5) 用户张新具有修改这两个表的结构权限。
- (6) 用户周平具有对两个表的所有权限 (读、插、改、删数据), 并具有给其他用户授权的权限。
- (7) 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。

Schema: 定义 worker (Wno, Wname, Wage, Wrank, Wwage, Dno)
department (Dno, Dname, Dceo, Dadress, Dphone)

1) GRANT SELECT ON worker, department TO 王明

2) GRANT INSERT, SELECT ON worker, department TO 李勇

3) GRANT SELECT ON worker, Wwage WHERE (Wname = NAME)

4) GRANT SELECT ON worker TO 刘星;
GRANT UPDATE ON department TO 刘星;

5) GRANT * ON worker, department TO 张新;

6) GRANT * ON worker, department TO 周平, With GRANT OPTION;

7) GRANT SELECT AVG(*), MAX(*), MIN(*) ON worker, Wwage TO 杨兰;

参考答案:

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:

① 用户王明对两个表有 SELECT 权限。

GRANT SELECT ON TABLE 职工, 部门 TO 王明;

② 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;

③ * 每个职工只对自己的记录有 SELECT 权限;

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

这里假定系统的 GRANT 语句支持 WHEN 子句和 USER() 的使用。用户将自己的名字作为 ID。注意,不同的系统这些扩展语句可能是不同的。读者应该了解所使用的 DBMS 产品的扩展语句。

④ 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;
```

⑤ 用户张新具有修改这两个表的结构权限。

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

⑥ 用户周平具有对两个表所有权(读,插,改,删数据),并具有给其他用户授权的权限。

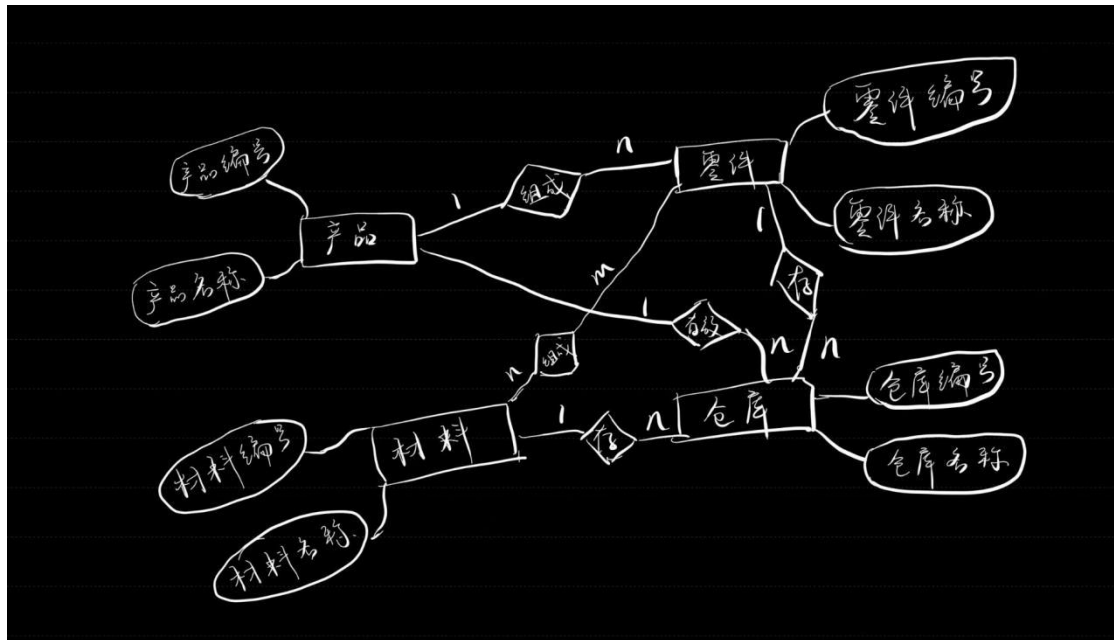
```
GRANT ALL PRIVILEGES ON TABLE 职工,部门 TO 周平 WITH GRANT OPTION;
```

⑦ 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。

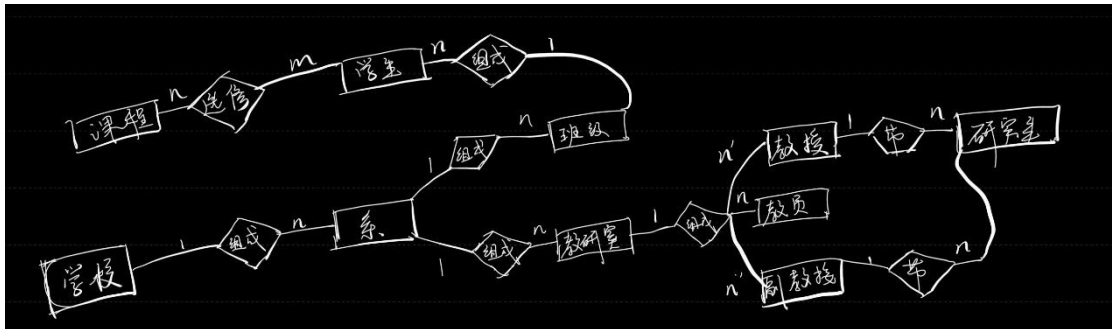
首先建立一个视图。然后对这个视图定义杨兰的存取权限。

```
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;
```

8. 某工厂生产若干产品,每种产品由不同的零件组成,有的零件可用在不同的产品上。这些零件由不同的原材料制成,不同零件所用的材料可以相同。这些零件按所属的不同产品分别放在仓库中,原材料按照类别放在若干仓库中。请用 E-R 图画出此工厂产品、零件、材料、仓库的概念模型。



7. 学校中有若干系，每个系有若干班级和教研室，每个教研室有若干教员，其中有的教授和副教授每人各带若干研究生，每个班有若干学生，每个学生选修若干课程，每门课可由若干学生选修。请用 E-R 图画出此学校的概念模型。



6. 有关系模式 $R(A, B, C, D, E)$ ，回答下面各个问题：

- (1) 若 A 是 R 的候选码，具有函数依赖 $BC \rightarrow DE$ ，那么在什么条件下 R 是 BCNF?
- (2) 如果存在函数依赖 $A \rightarrow B$, $BC \rightarrow D$, $DE \rightarrow A$ ，列出 R 的所有码。
- (3) 如果存在函数依赖 $A \rightarrow B$, $BC \rightarrow D$, $DE \rightarrow A$ ， R 属于 3NF 还是 BCNF。

(1) 要使 R 是 BCNF，需满足每个非平凡函数依赖的决定因素都是候选码。已知 A 是候选码，函数依赖 $BC \rightarrow DE$ 是非平凡函数依赖。所以需要 BC 也是候选码，此时 R 的候选码为 A 和 BC ，所有非平凡函数依赖的决定因素都是候选码， R 属于 BCNF

(2) R 的所有候选码 ACE , BCE , CDE

(3) R 属于 3NF，不属于 BCNF

7. 学校中有若干系，每个系有若干班级和教研室，每个教研室有若干教员，其中有的教授和副教授每人各带若干研究生，每个班有若干学生，每个学生选修若干课程，每门课可由若干学生选修。请用 E-R 图画出此学校的概念模型。

8. 某工厂生产若干产品，每种产品由不同的零件组成，有的零件可用在不同的产品上。这些零件由不同的原材料制成，不同零件所用的材料可以相同。这些零件按所属的不同产品分别放在仓库中，原材料按照类别放在若干仓库中。请用 E-R 图画出此工厂产品、零件、材料、仓库的概念模型。

9. 什么是数据库的逻辑结构设计？试述其设计步骤。

10. 试把习题 7 和习题 8 中的 E-R 图转换为关系模型。

第七题

系（系号，系名，系主任）

班级（班级号，班级名，所属系号）

教研室（教研室号，教研室名，所属系号）

教员（教员号，姓名，职称，所属教研室号）

学生（学号，姓名，班级号）

课程（课程号，课程名，学分）

研究生（研究生号，姓名，导师教员号）

选修（学号，课程号，成绩）

第八题

产品（产品号，产品名，规格）

零件（零件号，零件名，规格）

原材料（材料号，材料名，类别）

仓库（仓库号，仓库名，地址）

产品零件（产品号，零件号，数量）

零件材料（零件号，材料号，用量）

产品仓库（产品号，仓库号，存放量）

材料仓库（材料号，仓库号，存储量）

4. 考虑下图所示的日志记录：

序号	日志
1	T ₁ : 开始
2	T ₁ : 写 A, A=10
3	T ₂ : 开始
4	T ₂ : 写 B, B=9
5	T ₁ : 写 C, C=11
6	T ₁ : 提交
7	T ₂ : 写 C, C=13
8	T ₃ : 开始
9	T ₃ : 写 A, A=8
10	T ₂ : 回滚
11	T ₃ : 写 B, B=7
12	T ₄ : 开始
13	T ₃ : 提交
14	T ₄ : 写 C, C=12

- (1) 如果系统故障发生在 14 之后，说明哪些事务需要重做，哪些事务需要回滚。
- (2) 如果系统故障发生在 10 之后，说明哪些事务需要重做，哪些事务需要回滚。
- (3) 如果系统故障发生在 9 之后，说明哪些事务需要重做，哪些事务需要回滚。
- (4) 如果系统故障发生在 7 之后，说明哪些事务需要重做，哪些事务需要回滚。

故障恢复分析总结（Redo / Undo）

根据 事务的提交（Commit）与回滚（Rollback）状态：

- 已提交事务 → Redo（重做）
- 未提交事务 → Undo（回滚）
- 已回滚事务 → 无需处理

① 故障发生在 序号 14 之后

✓ 重做（Redo）

- T₁（序号 6 已提交）
- T₃（序号 13 已提交）
- T₄（已开始且故障发生在其写操作之后）

✗ 回滚（Undo）

- 无
（T₂ 在序号 10 已回滚）

② 故障发生在 序号 10 之后

✓ 重做 (Redo)

- T₁ (已提交)
- T₃ (未提交, 但其写入已落盘)

✗ 回滚 (Undo)

- T₄ (已开始但未提交)

说明: T₂ 已回滚, 无需处理。

③ 故障发生在 序号 9 之后

✓ 重做 (Redo)

- T₁

✗ 回滚 (Undo)

- T₃ (仅开始、未提交)

说明:

T₂ 已回滚; T₄ 尚未开始, 无需处理。

④ 故障发生在 序号 7 之后

✓ 重做 (Redo)

- T₁

✗ 回滚 (Undo)

- T₂ (执行了写 B、C, 但未提交)

说明:

T₃、T₄ 尚未开始, 无需处理。

故障发生位置	需重做事务 (Redo)	需回滚事务 (Undo)
14 之后	T ₁ 、T ₃ 、T ₄	无
10 之后	T ₁ 、T ₃	T ₄
9 之后	T ₁	T ₃
7 之后	T ₁	T ₂

（1）为什么要引入意向锁？

数据库存在表 → 页 → 行等层级锁。如果事务想申请表级锁，而不引入意向锁，系统必须遍历整张表检查是否存在行锁，效率极低。意向锁用于快速判断表内部是否被行锁占用，从而避免全表扫描，提高锁冲突判断效率。

（2）意向锁的含义是什么？

意向锁是一种加在表级别的指示性锁，用来表示表的内部是否存在行级锁（如 S/X 锁）。它本身不锁数据，只是为多级锁管理提供快速冲突检测

考试相关总结

一、说明关系数据库系统中视图（view）的定义及其优点

视图是建立在基本表或者是其他视图上，通过查询定义出来的虚拟表

二、什么是日志文件，为什么要设置日志文件？

日志文件

- 日志文件(log file)是用来记录事务对数据库的更新操作的文件
- 日志文件的格式
 - 以记录为单位的日志文件
 - 以数据块为单位的日志文件
- 用途
 - 进行事务故障恢复
 - 进行系统故障恢复
 - 协助后备副本进行介质故障恢复