

# 数据库全总结

By wln, 2022.12

Notice: 仅适用于sjs老师的《数据库》课程期末考试！

## 数据库全总结

- 一、基本概念填空
- 二、关系代数与QBE
  - 1、基础知识
    - 1.1 关系代数
    - 1.2 QBE
  - 2、作业和往年题
- 三、SQL查询
- 四、关系模式与规范化
  - 1、基础知识
  - 2、往年题
- 五、概念模型（E-R图）

## 一、基本概念填空

### 数据模型相关

- 数据库系统中所支持的主要数据模型是：**[非关系模型]**；**[关系模型]**；**[面向对象模型]**和**[对象关系模型]**（和**[半结构化数据模型]**）。现在流行的实际系统用的数据模型是**[关系模型]**。  
其中：**[层次模型]**、**[网状模型]**统称为非关系模型（视给的空数决定写它们两个还是一个非关系模型）  
（最常用的逻辑模型：层次模型、网状模型、关系模型）
- 数据库系统分三代，第一代是**[层次数据库系统]**和**[网状数据库系统]**，第二代是**[关系数据库系统]**，第三代是**[面向对象和对象关系数据库系统]**。目前商用的是第**[二]**代。
- 数据模型的三要素是指：**[数据结构]**、**[数据操作]**和**[数据完整性约束条件]**。分别描述系统的**[静态特性]**、**[动态特性]**和**[完整性约束条件]**。
- 数据模型分为**[概念模型]**和**[逻辑模型]**与**[物理模型]**，其中**[概念模型]**也叫**[信息模型]**，是按用户的观点来对数据和信息建模，用**[E-R图]**实现；而**[逻辑模型]**是按计算机系统的观点对数据建模，用于**[DBMS(数据库管理系统)]**实现。

### 关系与视图相关

- 关系是一张**[二]**维表，**关系**是其列（属性）的**[笛卡儿积]**的**[子集]**。关系的列是**[同质的]**，列的次序是**[任意的]**，行的次序是**[任意的]**
- 表的每一行对应一个**[元组]**，表的每一列对应一个**[域]**，给每一列起一个名字，称为**[属性]**。
- 关系模型的三类**完整性**是**[实体完整性]**、**[参照完整性]**和**[用户定义的完整性]**。其中**[实体完整性]**要求表示实体的**[主码中的属性]**不能为空。**[实体完整性]**和**[参照完整性]**是关系模型必须满足的约束性条件。
- **[视图]**是从**[一个或多个基本表或视图]**导出的表，数据库中只存放视图的**[定义]**，存放于**[数据字典]**中。因此，视图是一个**[虚表]**。行列子集视图是**[单]**个**[基本表]**导出的。**[可以]**在视图上再建立视

图。

### 规范化相关

- 一般情况下，如果关系模式R是[1NF]，若消除了非主属性对码的[部分]依赖就可以使之成为[2NF]；若再消除了非主属性对码的[传递]依赖就可以使之成为[3NF]。

BCNF是消除了3NF中，主属性对码的[部分函数]依赖和[传递函数]依赖

- [函数依赖]范围内关系模式最高可分解到[BC]NF。若在消除了非平凡非函数的[多值]依赖就可以使之成为[4NF]，若再消除了非平凡的[连接]依赖就可以使之成为[5NF]。

### 数据库安全性与设计等相关

- [数据库保护]也称为[数据控制]，主要包括数据的[安全性]、[完整性]、[并发控制]和[数据恢复]。

其中[并发控制]常用技术有[封锁]和[时间印]（Time Stamps）。

[数据恢复技术]包括数据[转储]和[登录日志文件]。

在数据[安全性]控制中主要技术有[自主存取控制DAC]和[强制存取控制MAC]。

- 按规范化设计的方法可将数据库设计分为六个阶段，分别是：[需求分析]、[概念模型设计]、[逻辑模型设计]、[物理模型设计]、[数据库实施]和[数据库运行与维护]。

其中在[需求分析]阶段实现[形成数据字典和数据流图]

在[概念模型设计]阶段画出[E-R图]。

在[逻辑模型设计]阶段[将E-R图转化为关系模型]

### 名词解释

- DBMS: Database Management System, 数据库管理系统
- DBA: Database Administrator, 数据库管理员
- OODBS: Object-Oriented Database System, 面向对象数据库系统
- E-R Diagram: Entity Relationship Diagram, ER图, 实体-联系图

## 二、关系代数与QBE

### 1、基础知识

以下示例使用“学生-课程数据库”：

Student

学号 Sno	姓名 Sname	性别 Ssex	年龄 Sage	所在系 Sdept
201215121	李勇	男	20	CS
201215122	刘晨	女	19	CS
201215123	王敏	女	18	MA
201215125	张立	男	19	IS

(a)

Course

课程号 Cno	课程名 Cname	先行课 Cpno	学分 Ccredit
1	数据库	5	4
2	数学		2
3	信息系统	1	4
4	操作系统	6	3
5	数据结构	7	4
6	数据处理		2
7	PASCAL语言	6	4

(b)

SC

学号 Sno	课程号 Cno	成绩 Grade
201215121	1	92
201215121	2	85
201215121	3	88
201215122	2	90
201215122	3	80

(c)

## 1.1 关系代数

运算符		含义
集合 运算符	$\cup$	并
	$-$	差
	$\cap$	交
	$\times$	笛卡儿积
专门的关系 运算符	$\sigma$	选择
	$\Pi$	投影
	$\bowtie$	连接
	$\div$	除

- 选择:**  $\sigma_F(R)$

从关系（表）R中选取使逻辑表达式（条件）F为真的元组。

（也即依照某些条件选出一些元组）

eg:

- 查询IS系的全体学生:  $\sigma_{Sdept='IS'}(Student)$
- 查询年龄小于20岁的学生:  $\sigma_{Sage<20}(Student)$

- 投影:**  $\Pi_A(R)$

关系R上的投影是从R中选择出若干属性列组成新的关系。其中A为R中的属性列。

（也即从关系中选出特定几列，组成一个新关系）

投影之后不仅取消了原关系中的某些列，而且还可能取消某些元组。因为取消某些属性列后，就可能出现重复行，应取消这些完全相同的行（比如原Student关系中有不同学生属于同一个系，因此原表中系别这一列上会有重复系名出现。若投影到系别上则应把重复的系名取消）

eg:

- 查询学生姓名和所在系，即求Student关系上学生姓名和所在系两个属性上的投影：  
 $\Pi_{Sname, Sdept}(Student)$

- 查询学生关系Student中有哪些系，即查询关系Student上所在系属性上的投影：

$$\Pi_{Sdept}(Student)$$

### • 连接： $R \bowtie_{A\theta B} S$

连接也称为 $\theta$ 连接。它是从两个关系的笛卡尔积中选取属性间满足一定条件的元组。

A和B分别为关系R和S上列数相等且可比的属性组， $\theta$ 是比较运算符。连接运算从R和S的笛卡尔积 $R \times S$ 中选取R关系在A属性组上的值与S关系在B属性组上的值满足比较关系 $\theta$ 的元组。

(也即把两个表通过满足某些条件的元组——对应，来连接称为一个新表)

连接运算中有两种最为重要也最为常用的连接，一种是**等值连接**，一种是**自然连接**。

$\theta$ 为“=”的连接运算称为**等值连接**。它是从关系R与S的广义笛卡尔积中选取**A, B属性值相等的那些元素**，然后将两个表对应地连接在一起。即等值连接为： $R \bowtie_{A=B} S$

**自然连接**是一种特殊的等值连接，它要求两个关系中进行比较的分量必须是同名属性组，并且在结果中把重复的属性列去掉。(也即在连接得到的新表中合并用于比较的同名属性列)

eg:

R			S		$R \bowtie_{C < E} S$				
A	B	C	B	E	A	R.B	C	S.B	E
a <sub>1</sub>	b <sub>1</sub>	5	b <sub>1</sub>	3	a <sub>1</sub>	b <sub>1</sub>	5	b <sub>2</sub>	7
a <sub>1</sub>	b <sub>2</sub>	6	b <sub>2</sub>	7	a <sub>1</sub>	b <sub>1</sub>	5	b <sub>3</sub>	10
a <sub>2</sub>	b <sub>3</sub>	8	b <sub>3</sub>	10	a <sub>1</sub>	b <sub>2</sub>	6	b <sub>2</sub>	7
a <sub>2</sub>	b <sub>4</sub>	12	b <sub>3</sub>	2	a <sub>1</sub>	b <sub>2</sub>	6	b <sub>3</sub>	10
			b <sub>5</sub>	2	a <sub>2</sub>	b <sub>3</sub>	8	b <sub>3</sub>	10

(a) 关系R			(b) 关系S		(c) 非等值连接				
A	B	C	B	E	A	B	C	E	
a <sub>1</sub>	b <sub>1</sub>	5	b <sub>1</sub>	3	a <sub>1</sub>	b <sub>1</sub>	5	3	
a <sub>1</sub>	b <sub>2</sub>	6	b <sub>2</sub>	7	a <sub>1</sub>	b <sub>2</sub>	6	7	
a <sub>2</sub>	b <sub>3</sub>	8	b <sub>3</sub>	10	a <sub>2</sub>	b <sub>3</sub>	8	10	
a <sub>2</sub>	b <sub>3</sub>	8	b <sub>3</sub>	2	a <sub>2</sub>	b <sub>3</sub>	8	2	

(d) 等值连接					(e) 自然连接			
A	R.B	C	S.B	E	A	B	C	E
a <sub>1</sub>	b <sub>1</sub>	5	b <sub>1</sub>	3	a <sub>1</sub>	b <sub>1</sub>	5	3
a <sub>1</sub>	b <sub>2</sub>	6	b <sub>2</sub>	7	a <sub>1</sub>	b <sub>2</sub>	6	7
a <sub>2</sub>	b <sub>3</sub>	8	b <sub>3</sub>	10	a <sub>2</sub>	b <sub>3</sub>	8	10
a <sub>2</sub>	b <sub>3</sub>	8	b <sub>3</sub>	2	a <sub>2</sub>	b <sub>3</sub>	8	2

上图中：(c) 图是 $R \bowtie_{C < E} S$ 得到的结果，也即把两表中所有满足C值小于B值的元组——对应，然后拼成一个新表。(d) 图为等值连接 $R \bowtie_{R.B=S.B} S$ 得到的结果，也即把两表中每个B值相等的元组——对应，然后拼成一个新表。(e) 图为自然连接 $R \bowtie S$ 得到的结果，与等值连接的区别在于把两表的B列在新表中合并成一列了。

### • 除运算： $R \div S$

设关系R除以关系S的结果为关系T，则T包含了所有在R但不在S中的属性及其值，且T的元组与S的元组的所有组合都在R中。除操作是同时从行和列角度进行运算。

(也即：首先需要把同时存在于R和S中的列从R中去除，保留R中剩下的列，然后再看这些属性(列)的值应该保留哪些：保留那些在R中属于的元组覆盖了S中所有与R重合列的属性值的值)

eg1:

R			S		
A	B	C	B	C	D
a <sub>1</sub>	b <sub>1</sub>	c <sub>2</sub>	b <sub>1</sub>	c <sub>2</sub>	d <sub>1</sub>
a <sub>2</sub>	b <sub>3</sub>	c <sub>7</sub>	b <sub>2</sub>	c <sub>1</sub>	d <sub>1</sub>
a <sub>3</sub>	b <sub>4</sub>	c <sub>6</sub>	b <sub>2</sub>	c <sub>3</sub>	d <sub>2</sub>
a <sub>1</sub>	b <sub>2</sub>	c <sub>3</sub>			
a <sub>4</sub>	b <sub>6</sub>	c <sub>6</sub>			
a <sub>2</sub>	b <sub>2</sub>	c <sub>3</sub>			
a <sub>1</sub>	b <sub>2</sub>	c <sub>1</sub>			

(a)

(b)

R ÷ S

A
a <sub>1</sub>

(c)

可以发现， $R \div S$ 后只剩下A这一列。并且A这一列的属性值中，仅有a<sub>1</sub>对应的A中的元组覆盖了S中所有B、C的属性值组合：b<sub>1</sub>c<sub>2</sub>、b<sub>2</sub>c<sub>1</sub>、b<sub>2</sub>c<sub>3</sub>。因此最终得到的结果就是A列的a<sub>1</sub>。

**除运算常用于“至少选择多个”、“全部”问题的解决。一般方法就是除以“至少”或“全部”的内容**

eg2:

**[例 2.10]** 查询至少选修 1 号课程和 3 号课程的学生号码。

首先建立一个临时关系 K:

K
Cno
1
3

然后求

$$\Pi_{Sno, Cno}(SC) \div K$$

结果为{201215121}。

先从SC表提取出Sno（学号）和Cno（课程号）两列，然后用其除含有Cno=1、3的K，这样留下的就只有Sno，并且保留的Sno数据在原表中对应的Cno至少覆盖了Cno=1、3。

## • 综合例题

- 查询选修了2号课程的学生的学号： $\Pi_{Sno}(\sigma_{Cno='2'}(SC))$

思路：先在同时含有学号、课程号属性的SC表中选择出Cno=2的元组，然后再投影到Sno上，得到满足条件的学号。

- 查询至少选修了一门其直接先行课为5号课程的学生姓名：

$$\Pi_{Sname}(\sigma_{Cpno='5'}(Course) \bowtie SC \bowtie \Pi_{Sno, Sname}(Student))$$

思路：首先在Course表（含有课程号和先修课程号）中选取符合先修课程号条件的元组们，然后再与SC表（含有课程号和学号）通过课程号进行自然连接（相当于扩展到了学号列），再与Student表（含有学号和学生姓名）通过学号进行自然连接（相当于扩展到了学生姓名列）；最终把连接得到的表向学生姓名（Sname）列进行投影，得到满足条件的学生姓名。

- 查询选修了全部课程的学生号码和姓名：

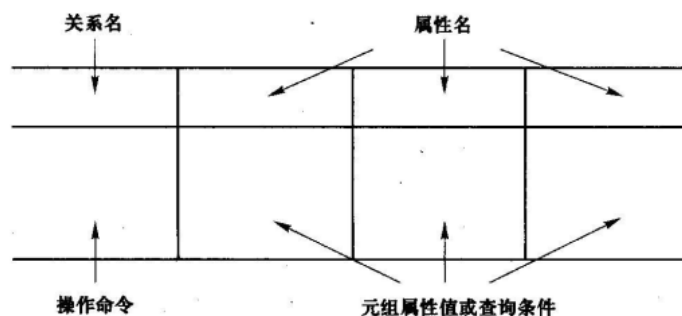
$$\Pi_{Sno, Cno}(SC) \div \Pi_{Cno}(Course) \bowtie \Pi_{Sno, Sname}(Student)$$

思路：“全部”问题也可用除法。先在SC表中选取所有的学号和课程号信息，再除以所有的课程号，即得到了“至少选取所有课程的学号”，再将其与同时含有学号和学生姓名的Student表在学号、学生姓名上投影的结果连接，即得到了符合条件的学号和学生姓名。（注意“全部”类查询也可用除法）

## 1.2 QBE

QBE (Query By Example, 域关系演算语言)。QBE用**示例元素**来表示查询结果可能的情况。示例元素实际上就是域变量。

操作框架：



仍以上边的学生-课程数据库为例：

- eg1: 查询信息系 (Sdept=IS) 的全体学生姓名

Student	Sno	Sname	Ssex	Sage	Sdept
		P. 李勇			IS

首先把关系（表）名写在表格左上角，然后在第一行列出所有的属性名。接下来在相应的属性下写出题设条件（这里是Sdept=IS，等号可省略）。然后再用P.XXX填充需要查询的属性，“P”是打印（Print）的意思，也即输出；而后边下划线上的东西只是一个示例，这个示例不一定真是该属性的一个观测值（比如这里的“李勇”就不是真正的一个Sname），它只是一个示例，可以写任何东西。

查询后屏幕上即显示查询结果：

Student	Sno	Sname	Ssex	Sage	Sdept
		张立			IS

- eg2: 查询全体学生的全部数据

Student	Sno	Sname	Ssex	Sage	Sdept
	P. 201215121	P. 李勇	P. 男	P. 20	P. CS

显示全部数据也可以简单地把 P.操作符作用在关系名上。因此本查询也可以简单地表示如下：

Student	Sno	Sname	Ssex	Sage	Sdept
P.					

- eg3: 查询年龄大于19岁的学生的学号

Student	Sno	Sname	Ssex	Sage	Sdept
	P. 201215121			>19	

- eg4: 查询在计算机科学系且年龄大于19岁的学生的学号

这里涉及了“与”操作。在QBE中表示两个条件的“与”有两种写法：

① 把两个条件写在同一行上；

Student	Sno	Sname	Ssex	Sage	Sdept
	P. 201215121			>19	CS

② 把两个条件写在不同行上，但使用相同的示例元素值。

Student	Sno	Sname	Ssex	Sage	Sdept
	P. 201215121				CS
	P. 201215121			>19	

- eg6: 将上边的“与”改成“或”

此时只能写成两行，并且需要使用不同的示例元素值：

Student	Sno	Sname	Ssex	Sage	Sdept
	P. 201215121				CS
	P. 201215122			>19	

- eg7: 查询既选修了1号课程又选修了2号课程的学生的学号

此时是在一个属性中的“与”关系，它只能用“与”条件的多行表示，也即写两行且示例元素相同。

SC	Sno	Cno	Grade
	P. 201215121	1	
	P. 201215121	2	

- eg8: 查询选修了1号课程的学生姓名

该查询涉及两个关系：SC（包含了学号和课程号信息）和Student（包含了学号和姓名信息）。QBE中实现的方法为通过相同的连接属性（XXX，同样只需填写任意的示例值）把多个关系连接起来：

Student	Sno	Sname	Ssex	Sage	Sdept
	201215121	P.李勇			

SC	Sno	Cno	Grade
	201215121	1	

这里示例元素Sno是连接属性，其示例值要在两个表中相同。

- eg9: 查询未选修1号课程的学生姓名

这里查询条件用到逻辑“非”，在QBE中表示逻辑“非”的方法是将逻辑“非”写在关系名下边。

Student	Sno	Sname	Ssex	Sage	Sdept
	201215121	P.李勇			
	201215122	P.王勇			

SC	Sno	Cno	Grade
¬	201215121	1	
¬	201215122		

本查询的条件是，该学生选修1号课程为假或者学生什么课程都没有选修（也即该学生由于没选修任何课导致其根本就不在SC）。（注意“未...”的可能需要考虑到这种全空情况）

- eg10: 查询有两个人以上选修的课程号

这里查询的是“至少”/“几个以上”，一般最多到“两个”，只需在表内用如下方式罗列即可。通过把“非”作用在一个实例值上表示与其不同的另一个。

本查询是在一个表内连接。这个查询就是要显示这样的课程 1，它不仅被 201215121 选修，而且也被另一个学生 (¬ 201215121) 选修了。

SC	Sno	Cno	Grade
	201215121	P. 1	
	¬ 201215121	1	

集函数：

表 2.7 QBE 中的聚集函数

函数名	功能
CNT	对元组计数
SUM	求总和
AVG	求平均值
MAX	求最大值
MIN	求最小值

- eg11: 查询信息系学生的平均年龄

Student	Sno	Sname	Ssex	Sage	Sdept
				P.AVGALL.	IS

## 2、作业和往年题

### (1) 作业

6. 设有一个 SPJ 数据库,包括 S、P、J、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	50

用关系代数和QBE语言完成下列查询：

注：以下QBE的答案中，查询属性和连接属性在表中的值均应加下划线，只有条件属性值不用加下划线。

- 求供应工程J1零件的供应商号SNO

思路：JNO和SNO同时存在于表SPJ中，因此只需单表查询即可。

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

QBE：

SPJ	SNO	PNO	JNO	QTY
	<u>P.S1</u>		<u>J1</u>	

- 求供应工程J1零件P1的供应商号SNO

思路：JNO、PNO和SNO同时存在于表SPJ中，因此只需单表查询即可。注意需要用到“交”关系。

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

QBE：

SPJ	SNO	PNO	JNO	QTY
	<u>P.S1</u>	<u>P1</u>	<u>J1</u>	

- 求供应工程J1红色零件的供应商号SNO

思路：SNO、JNO、PNO在表SPJ中，PNO、COLOR在表P中，因此应分别在SPJ、P表中选取JNO=J1和COLOR='红'的元组，然后通过PNO把两表进行自然连接，最后再映射到SNO上。

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

QBE:

SPJ	SNO	PNO	JNO	QTY
	P.S1	P1	J1	

P	PNO	PNAME	COLOR	WEIGHT
	P1		红	

- 求没有使用天津供应商生产的红色零件的工程师JNO

思路：求“没有”则使用减法，从J表（包括所有的工程）的JNO中减去“使用了天津供应商生产的红色零件的工程师JNO”。其中CITY（供应商城市）和SNO（供应商号）存在于表S中，SNO、PNO、JNO存在于表SPJ中，PNO和COLOR（零件颜色）存在于表P中。因此主要思路即以SPJ表为桥梁，连接分别带上相应查询条件的另外两个表。

关系代数：

$$\Pi_{JNO}(J) - \Pi_{JNO}(\Pi_{SNO}(\sigma_{CITY='天津'}(S)) \bowtie \Pi_{SNO,PNO,JNO}(SPJ) \bowtie \Pi_{PNO}(\sigma_{COLOR='红'}(P)))$$

QBE:

若不考虑没有使用任何零件的工程：

S	SNO	SNAME	STATUS	CITY
	S1			天津

P	PNO	PNAME	COLOR	WEIGHT
	P1		红	

SPJ	SNO	PNO	JNO	QTY
¬	S1	P1	P.J1	

也即先正着选，然后最终在SPJ表中取反。

但注意QBE中查询“未.....”时可能要考虑“一个都没有”的情况。这里还需要考虑没有使用任何零件的工程（其不会出现在SPJ表上），因此需写为：

J	JNO	JNAME	CITY
¬	P.J1		

S	SNO	SNAME	STATUS	CITY
	S1			天津

P	PNO	PNAME	COLOR	WEIGHT
	P1		红	

SPJ	SNO	PNO	JNO	QTY
	S1	P1	J1	

这里相当于最终是映射到J表（包含了所有工程）的JNO，并在中进行取反。

- 求至少使用了S1供应商所供应的全部零件的工程师JNO

思路：“至少”、“全部”问题即使用除法，除以“至少”的内容。

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

## (2) 往年题

设有供应商—零件关系数据库，其中的供应商关系S（供应商号SID，姓名SN，住址ADDR），零件关系P（零件号PID，零件名PN，颜色COLOR）。供应目录关系SP（供应商号SID，零件号PID，价格COST），分别如下表所示：

S:

SID	SN	ADDR
S1	A	BJ
S2	C	BJ
S3	E	SH
S4	B	TJ
S5	D	SH

SP:

SID	PID	COST
S1	P1	10
S1	P2	20
S1	P4	40
S2	P2	25
S2	P5	50
S3	P1	15
S3	P2	22
S3	P4	42
S3	P6	60
S4	P3	30
S4	P4	38
S5	P5	51
S5	P6	58

P:

PID	PN	COLOR
P1	PD	红
P2	PA	绿
P3	PC	蓝
P4	PB	红
P5	PF	蓝
P6	PE	绿

关系代数：

本部分自然连接的做法多为直接将几个表相连，得到一个包括所有涉及属性的新大表，再进行选择，这种思路也是可以的。

- 求供应红色或蓝色零件的供应商号

思路：可直接将SP和P通过PID属性自然连接后，将COLOR（零件颜色）和供应商号置于一张表中，直接进行选择与投影。

关系代数： $\Pi_{SID}(\sigma_{COLOR='红'\vee COLOR='蓝'}(SP\bowtie P))$

- 求供应绿色零件且住址是SH的供应商号

思路：直接将三个表连接，得到含有所有属性的表，再进行选择。

关系代数： $\Pi_{SID}(\sigma_{COLOR='绿'\wedge ADDR='SH'}(SP\bowtie S\bowtie P))$

- 求零件价格高于30的零件名和供应商名

思路：仍为将所有表直接连接，然后再进行选择。

关系代数： $\Pi_{PN,SN}(\sigma_{COST>30}(SP\bowtie S\bowtie P))$

- 求供应全部红色零件的供应商号

思路：“全部”问题的思路也是使用除法，除以“全部”的内容，剩下需要的内容。

关系代数： $\Pi_{SID,PID}(SP) \div \Pi_{PID}(\sigma_{COLOR='红'}(P))$

- 求没有供应蓝色零件的供应商名

思路：“没有”则使用减法即可，用所有供应商名减去供应蓝色零件的供应商名。

关系代数： $\Pi_{SN}(S) - \Pi_{SN}(\sigma_{COLOR='蓝'}(SP\bowtie S\bowtie P))$

QBE：

- 求供应零件P1的供应商名

S	SID	SN	ADDR
	<u>S0</u>	P. <u>SN0</u>	

SP	SID	PID	COST
	<u>S0</u>	P1	

- 求供应红色零件且住址是BJ的供应商名

S	SID	SN	ADDR
	<u>SID0</u>	P. <u>SN0</u>	BJ

SP	SID	PID	COST
	<u>SID0</u>	<u>PID0</u>	

P	PID	PN	COLOR
	<u>PID0</u>		红

- 求至少供应两种零件的供应商号

参考上边对“至少”问题在QBE中方法的解释。 [链接](#)

SP	SID	PID	COST
	P. <u>SID0</u>	<u>PID0</u>	
	<u>SID0</u>	$\neg$ <u>PID0</u>	

### 三、SQL查询

基础知识略。只涉及查询操作。

#### (1) 作业

6. 设有一个 SPJ 数据库,包括 S、P、J、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	50

- 找出所有供应商的姓名和所在城市

```
SELECT SNAME,CITY FROM S
```

- 找出所有零件的名称、颜色、重量

```
SELECT PNAME,COLOR,WEIGHT FROM P
```

- 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'
```

- 找出工程项目J2使用的各种零件的名称及其数量

思路：涉及多表连接查询。这里通过 P.PNO=SPJ.PNO 来连接两表

```
SELECT P.PNAME,SPJ.QTY
FROM P,SPJ
WHERE P.PNO=SPJ.PNO
AND SPJ.JNO='J2'
```

- 找出上海厂商供应的所有零件号码

思路：使用嵌套查询 IN 子句，内层查询所有上海的厂商，再使用IN语句作为外层查询的条件。注意这里需去重。

“IN”子句只能返回一个字段。

```
SELECT DISTINCT PNO
FROM SPJ
WHERE SNO IN
(SELECT SNO
FROM S
WHERE CITY='上海')
```

- 找出使用上海产的零件的工程名称

思路：涉及到三个表：S（包含供应商号SNO和供应商城市CITY）、SPJ（包含供应商号SNO和工程号JNO）、J（包含工程号JNO和工程名JNAME），分别通过SNO、JNO连接并进行条件查询即可。

```
SELECT JNAME
FROM J,SPJ,S
WHERE J.JNO=SPJ.JNO
AND SPJ.SNO=S.SNO
AND S.CITY='上海'
```

- 找出没有使用天津产的零件的工程号码

思路：“没有”的查询使用 NOT EXISTS 子句，转化为相反的查询语句：连接三个表，先查询所有使用了天津产零件的元组（select \* 即可），然后取 NOT EXISTS 的工程号码即可

“EXISTS”强调的是是否返回结果集，不要求知道返回什么。“EXISTS”、“NOT EXISTS”的返回值为真或假。因此“EXISTS”、“NOT EXISTS”子句内部一般使用 SELECT \* 即可。

注：这里内层查询实际上是把三个表都连起来了，没有写J表的原因是J表在外层查询处已经声明了。

```

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'
    )

```

## (2) 往年题

设有供应商—零件关系数据库，其中的供应商关系S（供应商号SID，姓名SN，住址ADDR），零件关系P（零件号PID，零件名PN，颜色COLOR）。供应目录关系SP（供应商号SID，零件号PID，价格COST），分别如下表所示：

S:

SID	SN	ADDR
S1	A	BJ
S2	C	BJ
S3	E	SH
S4	B	TJ
S5	D	SH

SP:

SID	PID	COST
S1	P1	10
S1	P2	20
S1	P4	40
S2	P2	25
S2	P5	50
S3	P1	15
S3	P2	22
S3	P4	42
S3	P6	60
S4	P3	30
S4	P4	38
S5	P5	51
S5	P6	58

P:

PID	PN	COLOR
P1	PD	红
P2	PA	绿
P3	PC	蓝
P4	PB	红
P5	PF	蓝
P6	PE	绿

- 求供应商S1所供应零件的平均价格

```

SELECT AVG(COST)
FROM SP
WHERE SID='S1'

```

- 求供应零件P2且价格不高于22的供应商名

```

SELECT S.SN
FROM SP,S
WHERE SP.SID=S.SID
AND SP.PID='P2'
AND SP.COST<=22

```

- 求未供应绿色零件的供应商名

思路：参照上边“NOT EXISTS”例子的解释。

```
SELECT SN
FROM S
WHERE NOT EXISTS
    (SELECT *
     FROM SP,P
     WHERE SP.SID=S.SID
     AND SP.PID=P.PID
     AND COLOR='绿'
    )
```

- 求同时供应红色和蓝色零件的供应商名

“同时”的查询方法可以在外层先写一个条件，再利用嵌套查询，在内层查询写另一个条件。

```
SELECT SN
FROM S,SP,P
WHERE SP.SID=S.SID
AND SP.PID=P.PID
AND COLOR='红'
AND SID IN
    (SELECT SID
     FROM SP,P
     WHERE SP.PID=P.PID
     AND COLOR='蓝'
    )
```

- 求这样的供应商号，其至少供应了供应商S1和S2所供应的全部零件

思路：“至少”也可用“NOT EXISTS”子句，其内部是不满足“至少”条件的查询。这里也即选取没供应S1和S2所供应全部零件的元组，具体方法为：选择含有S1或S2的元组，并且这些元组对应的零件号PID不在那些“S1或S2供应商供应的零件”中

```
SELECT SID
FROM S
WHERE NOT EXISTS
    (SELECT *
     FROM SP
     WHERE (SID='S1' OR SID='S2')
     AND PID NOT IN
        (SELECT PID
         FROM SP
         WHERE SID=S.SID
        )
    )
```



## 四、关系模式与规范化

### 1、基础知识

题目：

四。（15分）先给出要建立的数据库的两个关系模式，并给出语义描述。画出函数依赖图，分别说明上述两个关系模式是否为第三范式。若不是第三范式，则通过关系模式的分解（注意：不要过度分解！），使之成为第三范式，并写出分解后的各个关系模式。

- 关系模式

关系的描述称为关系模式

eg:

关系模式:学生  $S(SNO, SN, SB, DN, CNO, SA)$   
班级  $C(CNO, CS, DN, CNUM, CDATE)$   
系  $D(DNO, DN, DA, DNUM)$   
学会  $P(PN, DATE1, PA, PNUM)$   
学生-学会  $SP(SNO, PN, DATE2)$

(S、C、D、P、SP分别为一个关系模式)

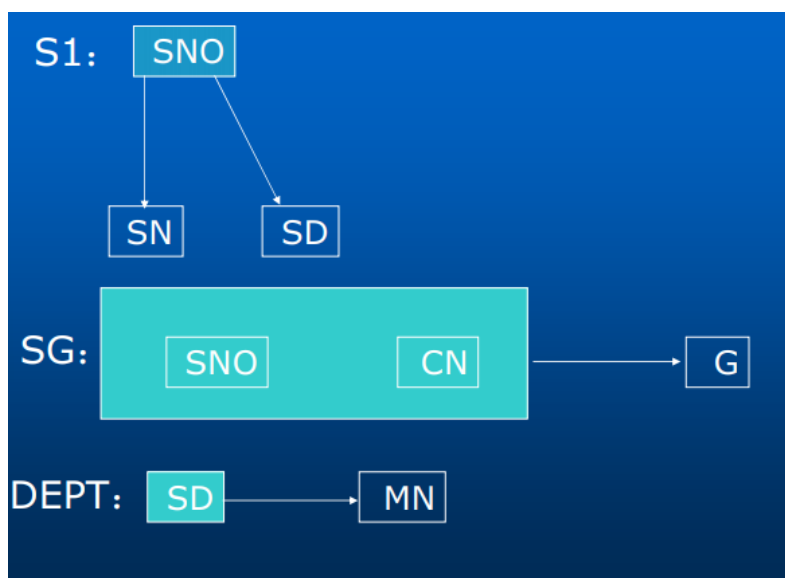
- 语义描述

eg:

有关语义如下：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，每个学会会有若干学生。学生参加某学会会有一个入会年份。

- 函数依赖图

eg:



每个关系模式都对应一个函数依赖图。

以下设 $R(U)$ 为属性集 $U$ 上的关系模式， $X, Y$ 是 $U$ 的子集， $F$ 为数据依赖。

- 函数依赖

函数依赖是属性间的一种联系，体现在属性值是否相等。如果 $Y$ 函数依赖于 $X$ （记作 $X \rightarrow Y$ ），则对于 $R(U)$ 中任意一个可能关系 $r$ ，对于 $r$ 中的任意两个元组，若在 $X$ 上的属性值相同，则在 $Y$ 上的属性值一定也相同。

- **完全函数依赖、部分函数依赖**

在 $R(U)$ 中，如果 $X \rightarrow Y$ ，且对于 $X$ 的任何一个真子集 $X'$ ，都有 $X' \not\rightarrow Y$ ，则称 $Y$ 对 $X$ 完全函数依赖，记作： $X \xrightarrow{F} Y$

若 $X \rightarrow Y$ ，但 $Y$ 不完全函数依赖于 $X$ ，则称 $Y$ 对 $X$ 部分函数依赖，记作： $X \xrightarrow{P} Y$

- **传递依赖**

在 $R(U)$ 中，如果 $X \rightarrow Y$ ， $Y \not\subseteq X$ ， $Y \rightarrow X$ ， $Y \rightarrow Z$ ， $Z \not\subseteq Y$ ，则称 $Z$ 对 $X$ 传递函数依赖

- **候选码/主码/主属性/非主属性/外码/全码**

- 若关系中的某一属性组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**。简称“码”。
- 若一个关系有多个候选码，则选定其中一个为**主码**。
- 候选码的诸属性称为**主属性**。
- 不包含在任何候选码中的属性为**非主属性或非码属性**。
- 关系模式 $R$ 中，属性或属性组 $X$ 并非 $R$ 的码，但 $X$ 是另一个关系模式的码，则称 $X$ 是 $R$ 的**外码**
- 在最简单的情况下，候选码只包含一个属性。在最极端的情况下，关系模式的所有属性是这个关系模式的候选码，称为**全码**。

- **1NF（第一范式）**

如果一个关系模式 $R$ 的所有属性都是不可分的基本数据项，则 $R \in 1NF$

第一范式是对关系模式最基本的要求

- **2NF(第二范式)**

若关系模式 $R \in 1NF$ ，且每一个非主属性都完全函数依赖于 $R$ 的码，则 $R \in 2NF$

通俗说法：消除了非主属性对码的**[部分]**依赖

eg:

SNO: 学生姓名, SN: 学号, SD: 学生所在院系, MN: 系主任, CN: 课程名, G: 学生课程成绩

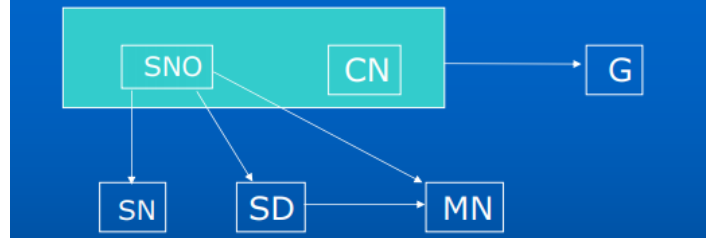
例如：关系模式 $S(SNO, SN, SD, MN, CN, G)$  的码为  $(SNO, CN)$ 。

函数依赖有：

$$\begin{aligned} SNO &\rightarrow SN, (SNO, CN) \xrightarrow{P} SN \\ SNO &\rightarrow SD, (SNO, CN) \xrightarrow{P} SD \\ SNO &\rightarrow MN, (SNO, CN) \xrightarrow{P} MN \\ (SNO, CN) &\xrightarrow{F} G \end{aligned}$$

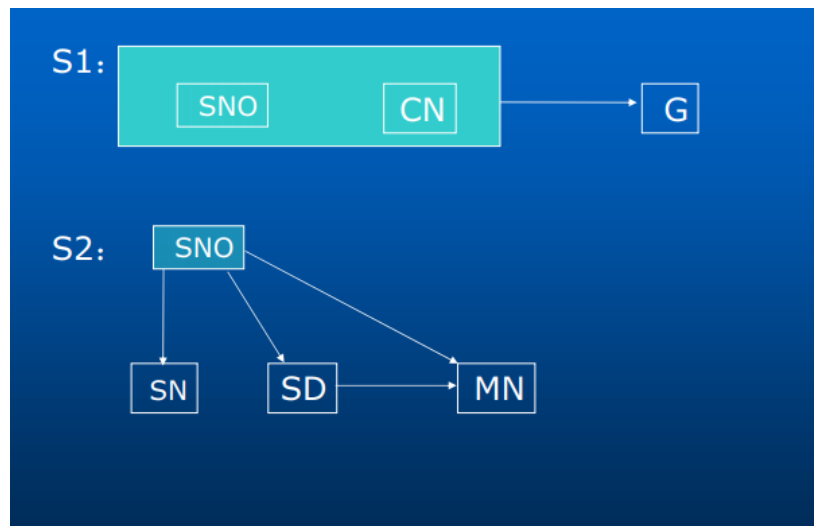
其中有部分依赖，不符合**2NF**的定义，因此 $S \notin 2NF$ 。

函数依赖图为：



解决办法：**投影分解**，将关系模式S分解成两个关系模式：

- $S1(\underline{SNO}, CN, G)$  码是(SNO, CN)
- $S2(\underline{SNO}, SN, SD, MN)$  码是SNO



这样非主属性对每个关系模式的码都是完全函数依赖，符合2NF的定义，也即  $S1 \in 2NF, S2 \in 2NF$

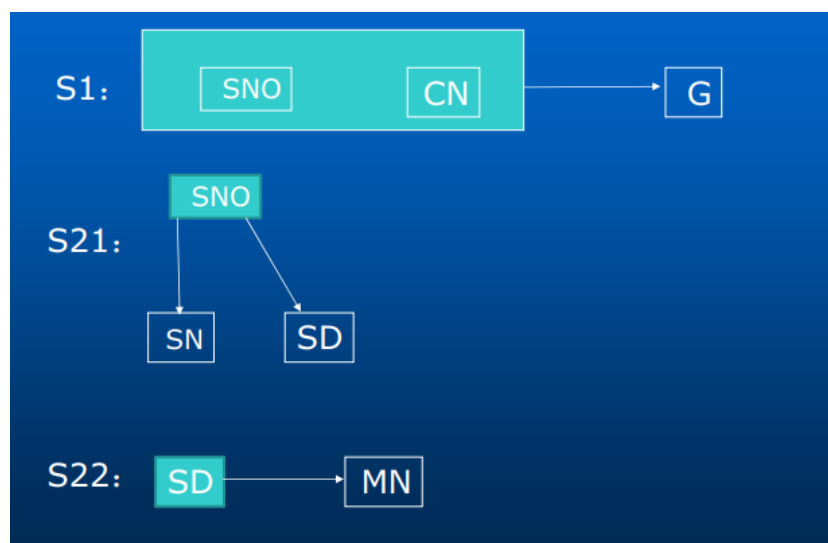
### • 3NF (第三范式)

关系模式  $R\langle U, F \rangle$  中若不存在这样的码  $X$ ，属性组  $Y$  及非主属性  $Z (Z \notin Y)$  使得  $X \rightarrow Y, (Y \not\rightarrow X), Y \rightarrow Z$  成立，则称  $R\langle U, F \rangle \in 3NF$

通俗说法：消除了非主属性对码的[传递]依赖

eg:

接上例，分解成3NF后函数的依赖图为：



综上：首先选出候选码（即可以决定所有其他非主属性的一组属性）。判断3NF，则看是否都是完全函数依赖（2NF条件），以及是否没有传递依赖（3NF条件）

## 2、往年题

我们要建立一个数据库来描述学生—班级—系—宿舍—课程的情况，对象有：学生、学号（SID）、学生姓名（SN）、班级号（CID）、班级名（CLN）、系号（DID）、系名（DN）、宿舍（DORM）、课程名（CN）、成绩（G）。

首先构造了两个关系模式：

学生关系：S <SID,SN,CLN,CN,G>

班级关系：CL <CID,CLN,DID,DN,DORM>

现实世界的已知事实是（没提到的不要臆断）：固有：Sid->Sn, Cid->Cln, Did->Dn

(1) 一个班有若干学生，但一个学生只属于一个班； Sid->Cid

(2) 一个系有若干班，但一个班只属于一个系； Cid->Did

(3) 一个系住在同一个宿舍楼； Did->Dorm

(4) 一个学生可以选修多门课程，每门课程有若干学生选修；

(5) 每个学生所修的每门课程有一个成绩。 (Sid,Cn)->G

试画出依赖图，说明上述关系模式不是第三范式，并通过关系模式的分解，使之成为第三范式。

## 五、概念模型（E-R图）

五。（15分）用文字给出某组织的组织结构和活动描述。

要求按数据库设计的要求完成以下工作：

（1）用E-R图画出这个组织的组织结构和活动的概念模型（标明每个联系是哪种性质的联系）；

（2）将E-R图转换成关系数据模型，写出相应关系模式；并说明在满足第三范式的条件下至少需要几个关系模式，并写出这些关系模式。

注意：关系模式和属性用汉字表示即可，不要自己定义符号。

- E-R图（实体-联系图）

- 实体型用矩形表示，矩形框内写明实体名
- 属性用椭圆表示，并用无向边将其与相应的实体型连接起来
- 联系用菱形表示，菱形框内写明联系名，并用无向边分别与有关实体连接起来，同时，在无向边旁标上联系的类型（1:1 或 1:n 或 m:n）

- E-R图向关系模型转换

- 每个实体型转换为一个关系模式

(实体的属性 -> 关系的属性, 实体的码 -> 关系的码)

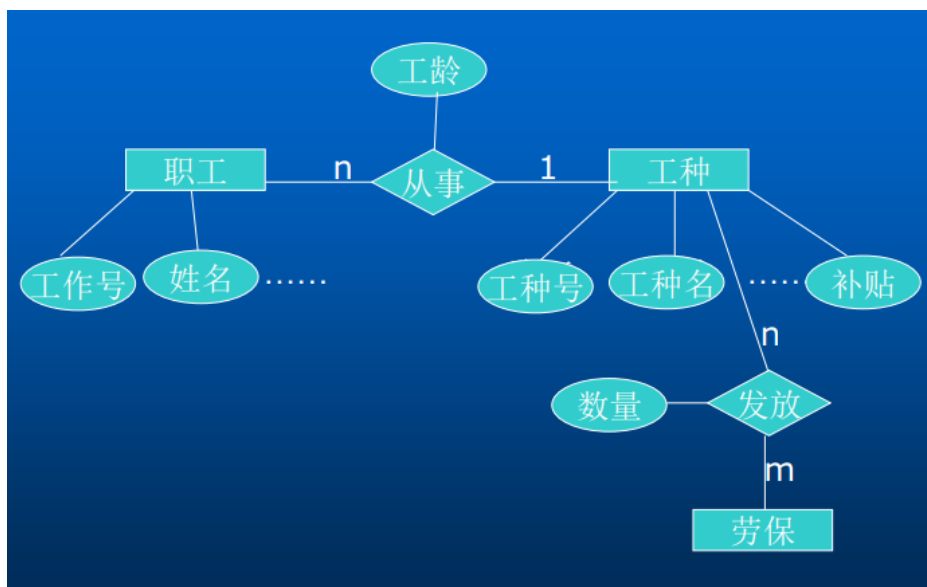
- 每个m:n的联系转换为一个关系模式, 与该联系相连的各实体的码以及联系的属性转换为关系的属性, 关系的码是各实体的码的组合
- 一个1:1的联系可以转换为一个关系模式, 也可以与该联系相连的任意一端的实体合并。此关系的码可以是与该联系相连的任意一端的实体的码。联系的属性转换为关系的属性。
- 一个1:n的联系可以转换为一个关系模式, 也可以与该联系相连的n端的实体合并。此关系的码是与该联系相连的n端的实体的码。联系的属性转换为关系的属性
- 具有相同码的关系模式可以合并

**总结来看: 每个实体必须转为一个关系模式, 每个m:n的联系也必须转为一个关系模式 (其主码为它联系的两个实体的主码的结合), 1:n、1:1的联系由于其主码必然与某些实体重合因此不必单独转为一个关系模式。**

• eg:

(1)

E-R图 (概念模型):

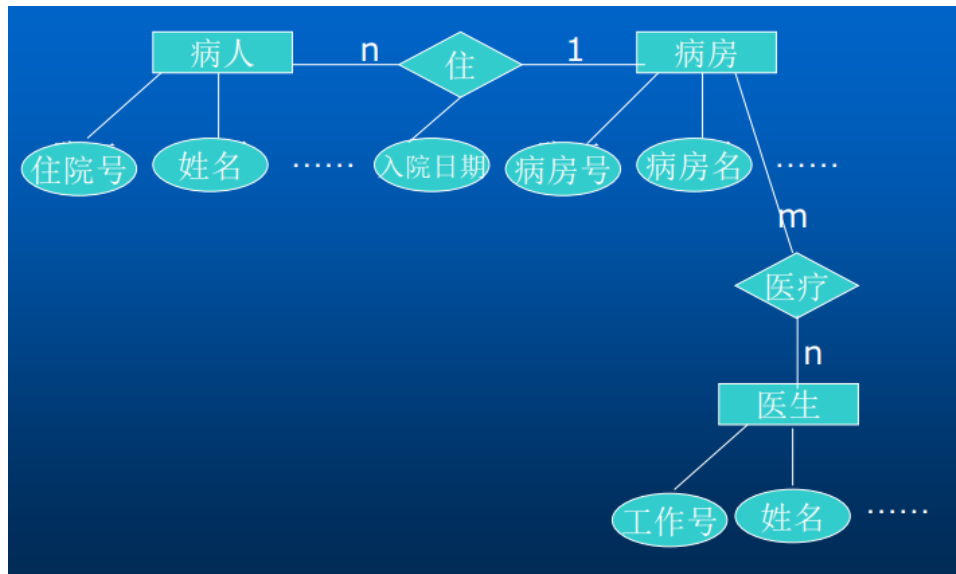


转为关系模型:

- 职工(工作号, 姓名, .....)
  - 工种(工种号, 工种名, 补贴)
  - 从事(工作号, 工种号, 工龄)
  - 劳保(劳保名, .....)
  - 发放(工种号, 劳保名, 数量)
- 职工和从事两个关系模式由于主码相同可以合并为一个:  
职工(工作号, 姓名, ....., 工种号, 工龄)
- 最终的数据库有4个关系模式, 这是最少的数目。

(2)

E-R图 (概念模型) :



转化为关系模型:

- 病人(住院号, 姓名, ....)
  - 病房(病房号, 病房名, ....)
  - 住(住院号, 病房号, 入院日期)
  - 医生(工作号, 姓名, .....)
  - 医疗(工作号, 病房号, ....)
- 病人和住两个关系模式由于主码相同可以合并为一个:  
病人(住院号, 姓名, ....., 病房号, 入院日期)
- 最终的数据库有4个关系模式, 这是最少的数目。

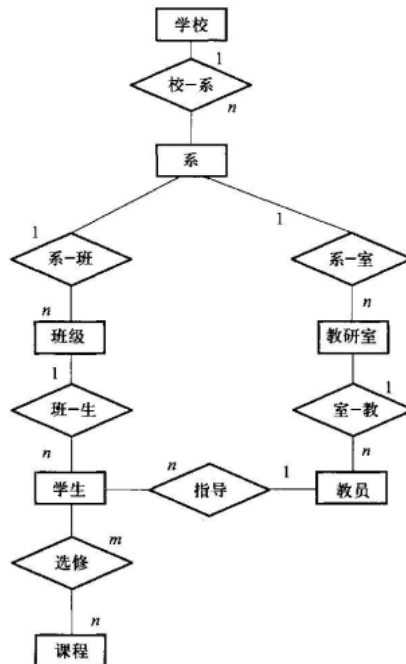
40

### 作业题

(1)

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。

答:



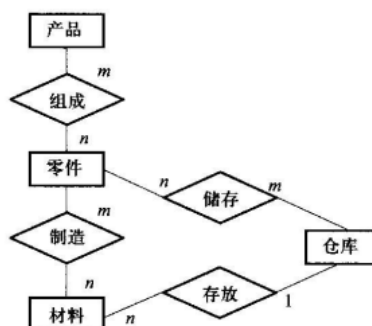
转化为关系模型:

系(系编号,系名,学校名)  
 班级(班级编号,班级名,系编号)  
 教研室(教研室编号,教研室,系编号)  
 学生(学号,姓名,学历,班级编号,导师职工号)  
 课程(课程编号,课程名)  
 教员(职工号,姓名,职称,教研室编号)  
 选课(学号,课程编号,成绩)

(2)

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。

答:



转为关系模型:

产品(产品号,产品名,仓库号)

零件(零件号,零件名)

原材料(原材料号,原材料名,类别,仓库号,存放量)

仓库(仓库号,仓库名)

产品组成(产品号,零件号,使用零件量)

零件组成(零件号,原材料号,使用原材料量)

零件储存(零件号,仓库号,存储量)