

1、在教学管理数据库中，有如下三个关系表

学生信息表：S(S#, SNAME, AGE, SEX)

课程表：C(C#, CNAME, TEACHER)

选课表：SC(S#, C#, GRADE)

其中 S#、C#为 S、C 表的主码，(S#, C#)是 SC 表的主码，也分别是参照 S、C 表的外码，

请使用关系代数表达式回答下列问题：

1) 查询同时学习过课程号为 001 和 002 课程的学生的学号和成绩

$$\Pi_{S\#,GRADE}(\sigma_{C\#="001" \text{ and } C\#="002"}(SC))$$

错了，应取交集

2) 查询学习过课程号为 001 的学生学号、姓名、及这门课的成绩

$$\Pi_{S\#,SNAME,GRADE}(S \times (\sigma_{C\#="001"}(SC)))$$

笛卡尔积该为自然连接，下同

3) 查询没有学习过课程号为 002 的学生姓名和年龄

$$\Pi_{SNAME,AGE}(S) - \Pi_{SNAME,AGE}(\sigma_{C\#="002"}(S \times SC))$$

4) 查询老师 cheng 所教过的学生中成绩为 90 分以上(包括 90 分)的学生姓名

$$\Pi_{SNAME}(S \times (\sigma_{TEACHER="cheng" \text{ and } GRADE \geq 90}(SC \times C)))$$

5) 查询选修了全部课程的学生姓名

$$\Pi_{SNAME}(\Pi_{S\#,SNAME}(S) \bowtie (\Pi_{S\#,C\#}(SC) \div \Pi_{C\#}(C)))$$

2、在工程管理数据库中，包括 S, P, J, SPJ 四个关系模式：

S(S#, SNAME, SCITY)

P(P#, PNAME, COLOR)

J(J#, JNAME, JCITY)

SPJ(S#, P#, J#, QTY)

供应商表 S 由供应商代码 (S#)、供应商姓名 (SNAME)、供应商所在城市 (SCITY) 组成；

零件表 P 由零件代码 (P#)、零件名 (PNAME)、颜色 (COLOR) 组成；

工程表 J 由工程代码 (J#)、工程名 (JNAME)、工程所在城市 (JCITY) 组成；供

应情况表由 SPJ 由供应商代码 (S#)、零件代码 (P#)、工程代码 (J#)、零件供应数量 (QTY) 组成。

其中 S#、P#、J# 分别是 S、P、J 表的主码，(S#, P#, J#) 是 SPJ 的主码，分别是参照 S、P、J 表的外码。

请使用关系代数表达式回答下列问题：

1 检索使用了在北京且供应商名字为 “S1” 的供应商生产的蓝色零件的工程代码。

$$\Pi_{J\#}(\sigma_{SCITY=Beijing \wedge SNAME="S1" \wedge COLOR="blue"}(S \bowtie SPJ \bowtie P))$$

2 检索供应商与工程所在城市相同的工程代码和工程名。

$$\Pi_{J\#, JNAME}(\sigma_{SCITY=JCITY}(S \bowtie SPJ \bowtie J))$$

3 检索长春的任何工程都不使用的零件代码；

$$\Pi_{P\#}(P) - \Pi_{P\#}(\sigma_{JCITY="Changchun"}(SPJ \bowtie J))$$

4 检索使用了零件号= “P2” 的零件的工程代码及工程名

$$\Pi_{J\#, JNAME}(J \bowtie SPJ \bowtie (\sigma_{P\#="P2"}(P)))$$

5 检索为工程代码= “J5” 的工程供应绿色零件的供应商代码和供应商姓名

$$\Pi_{S\#, SNAME}(S \bowtie (\sigma_{J\#="J5"}(SPJ)) \bowtie (\sigma_{COLOR="green"}(P)))$$

1、判断下列命题是否成立。若不成立，请给出反例

- (a) $\sigma_{\theta_1}(\sigma_{\theta_2}(R)) = \sigma_{\theta_2}(\sigma_{\theta_1}(R)) = \sigma_{\theta_1 \wedge \theta_2}(R)$
- (b) $\Pi_{L_1}(\Pi_{L_2}(R)) = \Pi_{L_2}(\Pi_{L_1}(R))$
- (c) $\Pi_L(\sigma_\theta(R)) = \sigma_\theta(\Pi_L(R))$
- (d) $\Pi_L(R \cup S) = \Pi_L(R) \cup \Pi_L(S)$ // 假定 R 和 S 满足求并的条件
- (e) $\Pi_L(R \cap S) = \Pi_L(R) \cap \Pi_L(S)$ // 假定 R 和 S 满足求交的条件
- (f) $\sigma_\theta(R \cap S) = \sigma_\theta(R) \cap S = R \cap \sigma_\theta(S)$
- (g) $\sigma_\theta(R - S) = \sigma_\theta(R) - S = R - \sigma_\theta(S)$
- (h) $(R \bowtie_{\theta_1} S) \bowtie_{\theta_2} T = R \bowtie_{\theta_1} (S \bowtie_{\theta_2} T)$
- (i) $(R \bowtie S) \bowtie T = R \bowtie (S \bowtie T)$
- (j) $R \bowtie R = R \cap R$

答案：

1. (a) 成立。

(b) 不成立。 $\Pi_a(\Pi_{a,b}(R)) \neq \Pi_{a,b}(\Pi_a(R))$

(c) 不成立。 $\Pi_a(\sigma_{b>0}(R)) = \sigma_{b>0}(\Pi_a(R))$

(d) 成立。

(e) 不成立。设 $R(a,b) = \{(1,2)\}$, $S(a,b) = \{(1,3)\}$, 则 $\Pi_a(R \cap S) = \emptyset$, $\Pi_a(R) \cap \Pi_a(S) = \{(1)\}$

(f) 成立。

(g) 不成立。 $\sigma_\theta(R - S) = \sigma_\theta(R) - S$, 但 $\sigma_\theta(R) - S \neq R - \sigma_\theta(S)$ 。设 $R(a,b) = \{(1,2)\}$, $S(a,b) = \{(3,4)\}$, 则 $\sigma_{a=3}(R) - S = \emptyset$, $R - \sigma_{a=3}(S) = \{(1,2)\}$

(h) 不成立。 $R(a,b)$, $S(b,c)$, $T(a,c)$, $(R \bowtie_{R.b=S.b} S) \bowtie_{R.a=T.a} T \neq R \bowtie_{R.b=S.b} (S \bowtie_{R.a=T.a} T)$

(i) 成立。

(j) 成立。

2、在教学管理数据库中，有如下四个关系表

学生信息表：S(S#, SNAME, AGE, SEX)

课程表：C(C#, CNAME, T#)

选课表：SC(S#, C#, GRADE)

教师表：T(T#, TNAME, ADDR)

请使用SQL语句完成如下的功能操作：

1) 查询未讲授“数据库系统”课程的教师号和教室名。

```
select T#, TNAME
from T
where T# not in ( select T#
                  from C
                  where CNAME= '数据库系统' );
```

2) 查询既讲授了“C1”号又讲授了“C4”号课程的教师姓名。

```
select TNAME
from T
where T# in ( select T#
```

```

from C X, C Y
where X.C#='C1' and Y.C#='C4' );

```

- 3) 查询至少讲授两门课程的教师号、教师姓名和地址。

```

select *
from T
where T# in ( select T#
               from C
               group by T#
               having count(*)>=2 );

```

- 4) 查询年龄大于20岁男、女同学各有多少人。

```

select sex, count(*)
from S
where age>20
group by sex;

```

- 5) 查询年龄大于所有男同学年龄的女同学的学号、姓名和年龄。

```

select S#, SNAME, AGE
from S
where sex='female' and age> ( select max(age)
                              from S
                              where sex='male' );
或者age>all( select age
              from S
              where sex='male' );

```

- 6) 查询只选修了两门课程的男同学的学号和姓名。

```

select S#, SNAME
from S
where sex='female' and S# in ( select S#
                                from SC
                                group by S#
                                having count(*)=2 );

```


11) 将“liu”同学的信息从基本表S和SC中删除。

```
delete from SC
where S# in ( select S#
              from S
              where SNAME= 'liu' );
delete from SC where SNAME= 'liu' ;
```

1、设有如下实体：

图书：书号、书名、出版日期、出版社

读者：借书证号、姓名、年龄、性别、家庭住址

书架：书架号、房间号

上述实体中存在如下联系：

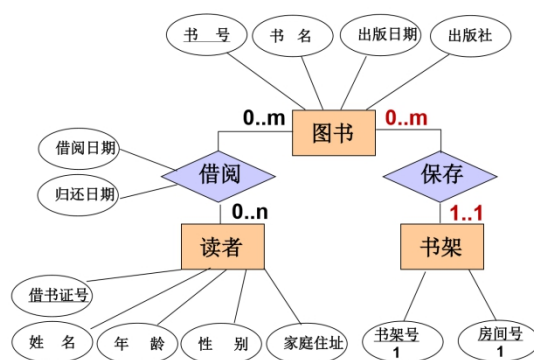
a) 一本书只能放在一个书架上，一个书架可以放多本书；

b) 一位读者可以借多本书，一本书只能被一位读者借阅

试完成如下工作：

(1) 设计该图书管理系统的 E-R 图；

(2) 将该 E-R 图转换为等价的关系模式表示的数据库逻辑结构。



关系模式：

图书（书号、书名、出版日期、出版社）

读者（借书证号、姓名、年龄、性别、家庭住址）

借阅（借书证号、书号）

书架（书架号、房间号）

借阅（书架号、书号）

2、设有如下实体：

学生：学号、单位名称、姓名、性别、年龄、选修课名

课程：编号、课程名、开课单位、任课教师号

教师：教师号、姓名、性别、职称、讲授课程编号

单位：单位名称、电话、教师号、教师姓名

上述实体存在如下联系：

(1) 一个学生可以选多门课程，一门课程可被多名学生选修

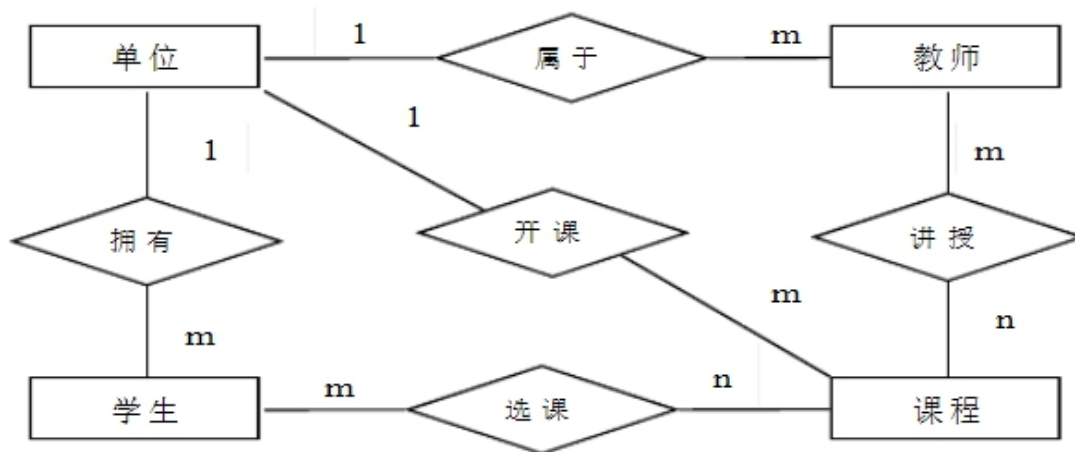
(2) 一个教师可讲授多门课程，一门课程可由多个教师讲授

(3) 一个单位可有多名教师或学生，一个教师或学生只属于一个单位

(4) 一个单位可开设多门课程，一门课程只能属于一个单位

完成如下工作：

设计该系统的 ER 图，并写出对应的关系模式，标明主码



图中各实体属性：

单位：单位名称，电话

学生：学号、姓名、性别、年龄

教师：教师号、姓名、性别、职称

课程：编号、课程名

关系模式：

单位（单位名称，电话）

课程（编号，课程名，单位名称）

教师（教师号，姓名，性别，职称，单位名称）

学生（学号，姓名，性别，年龄，单位名称）

讲授（教师号，课程编号）

选课（学号，课程编号）

3、工厂需建立一个管理数据库存储以下信息：

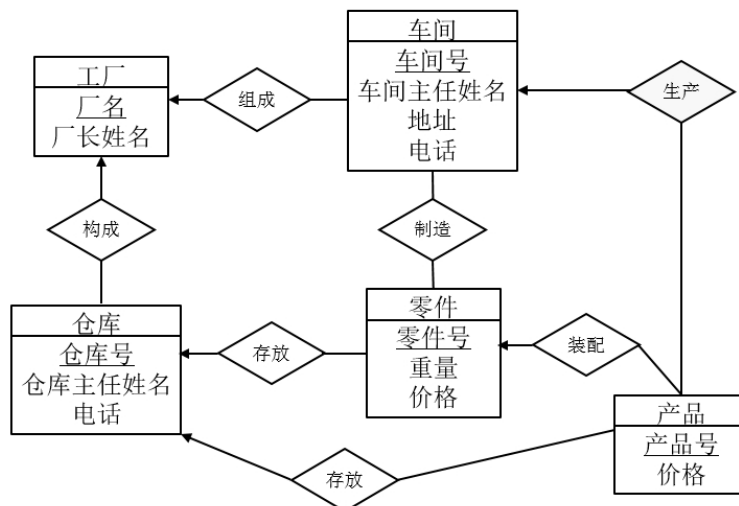
- 1) 工厂：厂名、厂长姓名；
- 2) 车间：车间号、车间主任姓名、地址、电话；
- 3) 仓库：仓库号、仓库主任姓名、电话；
- 4) 零件：零件号、重量、价格；
- 5) 产品：产品号、价格；

上述实体存在如下联系：

- a) 一个工厂内有多个车间和多个仓库，一个车间或一个仓库都只能属于一个工厂；
- b) 一个车间生产多种产品，每种产品只能产自一个车间；
- c) 一个车间生产多种零件，一种零件也可能为多个车间所制造；
- d) 一个产品由多种零件组成，一种零件也可装配出多种产品；
- e) 产品和零件均存入仓库。

根据上述要求，完成如下工作：

画出该系统的 E-R 图，并写出对应的关系模式，标明主码：



工厂（厂名、厂长姓名）

车间（车间号、车间主任姓名、地址、电话、厂名）

仓库（仓库号、仓库主任姓名、电话、厂名）

产品（产品号、价格、车间号、仓库号）

零件（零件号、重量、价格、仓库号）

制造（车间号、零件号）

4. 举例说明 3 个实体型之间的三元联系型与这 3 个实体型之间的 3 个二元联系型表示的意义通常是不同的。在什么情况下 3 个实体型之间的三元联系型与这 3 个实体型之间的 3 个二元联系型表示的意义会是相同的？

如果满足下面的条件，则 3 个实体型之间的三元联系型与这 3 个实体型之间的 3 个二元联系型表示的含义相同：对于任意两个三元联系 (a, b, c) 和 (a', b', c') ，

- 如果 $a = a'$ 且 $b = b'$ ，则 $c = c'$ 。
- 如果 $a = a'$ 且 $c = c'$ ，则 $b = b'$ 。
- 如果 $c = c'$ 且 $b = b'$ ，则 $a = a'$ 。

1. 已知学生关系模式：

$S(Sno, Sname, SD, Sdname, Course, Grade)$

其中：Sno 学号、Sname 姓名、SD 系名、Sdname 系主任名、Course 课程、Grade 成绩。

(1) 写出关系模式 S 的基本函数依赖和主码。

答：关系模式 S 的基本函数依赖如下：

$Sno \rightarrow Sname, SD \rightarrow Sdname, Sno \rightarrow SD, (Sno, Course) \rightarrow Grade$

关系模式 S 的码为：(Sno, Course)。

(2) 原关系模式 S 为几范式？为什么？分解成高一级范式，并说明为什么？

答：原关系模式 S 是属于 1NF 的，码为 (Sno, Course)，非主属性中的成绩完全依赖于码，而其它非主属性对码的函数依赖为部分函数依赖，所以不属于 2NF。

消除非主属性对码的函数依赖为部分函数依赖，将关系模式分解成 2NF 如下：

$S1(Sno, Sname, SD, Sdname)$ 、 $S2(Sno, Course, Grade)$

(3) 将关系模式分解成 3NF，并说明为什么？

答：将上述关系模式分解成 3NF 如下：

关系模式 S1 中存在 $Sno \rightarrow SD, SD \rightarrow Sdname$ ，即非主属性 Sdname 传递依赖于 Sno，所以 S1 不是 3NF。进一步分解如下：

$S11(Sno, Sname, SD)$ $S12(SD, Sdname)$

分解后的关系模式 S11、S12 满足 3NF。

对关系模式 S2 不存在非主属性对码的传递依赖，故属于 3NF。所以，原模式 $S(Sno, Sname, SD, Sdname, Course, Grade)$ 按如下分解满足 3NF。

$S11(Sno, Sname, SD)$ $S12(SD, Sdname)$

$S2(Sno, Course, Grade)$

2. 设关系模式 $R(ABCD)$ ，F 是 R 上成立的 FD 集， $F = \{AB \rightarrow CD, A \rightarrow D\}$ 。

(1) 试说明 R 不是 2NF 模式的理由；

(2) 试把 R 分解成 2NF 模式集；

解：(1) 从已知 FD 集 F 可知，R 的候选键是 AB。另外 $AB \rightarrow D$ 是一个局部依赖，因此 R 不是 2NF 模式。

(2) R 应该分解成 $\rho = \{AD, ABC\}$ 。

3. 设关系模式 $R(ABC)$ ，F 是 R 上成立的 FD 集， $F = \{C \rightarrow B, B \rightarrow A\}$ 。

(1) 试说明 R 不是 3NF 模式的理由；

(2) 试把 R 分解成 3NF 模式集；

解：(1) 从已知 FD 集 F 可知，R 的候选键是 C。从 $C \rightarrow B$ 和 $B \rightarrow A$ 可知， $C \rightarrow A$ 是一个传递依赖，因此 R 不是 3NF 模式。

(2) R 应该分解成 $\rho = \{CB, BA\}$ 。

4. 设关系模式 $R(A, B, C, D)$ ，其函数依赖集为：

$F = \{A \rightarrow C, C \rightarrow A, B \rightarrow AC, D \rightarrow AC\}$

(1) 求 R 的极小函数依赖集（最小覆盖）

(2) 求 R 的候选码

解：(1)

$F = \{A \rightarrow C, C \rightarrow A, B \rightarrow A, B \rightarrow C, D \rightarrow A, D \rightarrow C\}$

$\because B \rightarrow A, A \rightarrow C, \therefore B \rightarrow C$

$\because D \rightarrow A, A \rightarrow C, \therefore D \rightarrow C$

$\therefore F_{\min} = \{A \rightarrow C, C \rightarrow A, B \rightarrow A, D \rightarrow A\}$ 为极小函数依赖集

或

$\because B \rightarrow C, C \rightarrow A, \therefore B \rightarrow A$

$\because D \rightarrow C, C \rightarrow A, \therefore D \rightarrow A$

$\therefore F_{\min} = \{A \rightarrow C, C \rightarrow A, B \rightarrow C, D \rightarrow C\}$ 为极小函数依赖集

(2)

\because BD 在 F 中所有函数依赖的右部均未出现， \therefore 候选码中一定要包含 BD，而 $(BD)^+ = ABCD$ ，因此，BD 是 R 唯一的候选码

5. 设有关系模式 $R\{A, B, C, D, E\}$ ，其上有函数依赖集：

$F = \{A \rightarrow C, C \rightarrow D, B \rightarrow C, DE \rightarrow C, CE \rightarrow A\}$

(1) 求所有候选键。

(2) 判断 $\rho = \{AD, AB, BC, CDE, AE\}$ 是否是无损连接分解？

(3) 把 R 分解为 BCNF，并具有无损连接性。

(1) 从 F 中看，候选关键字至少包含 BE (因为它们不依赖于谁)，而 $(BE)^+ = ABCDE$ ， \therefore BE 是 R 的惟一候选关键字。

(2) ρ 的无损连接性判断结果如图 4.17 所示，由此判定 ρ 不具有无损连接性。

(3) 考虑 $A \rightarrow C$ ， \therefore AC 不是 BCNF (AC 不包含候选关键字 BE)，将 ABCDE 分解为 AC 和 ABDE，AC 已是 BCNF。

进一步分解 ABDE，选择 $B \rightarrow D$ ，把 ABDE 分解为 BD 和 ABE，此时 BD 和 ABE 均为 BCNF。

$\therefore \rho = \{AC, BD, ABE\}$

R_i	A	B	C	D	E
AD	a_1		a_3	a_4	
AB	a_1	a_2	a_3	A_4	
BC		a_2	a_3	A_4	
CDE	a_1		a_3	A_4	a_5
AE	a_1		a_3	A_4	a_5

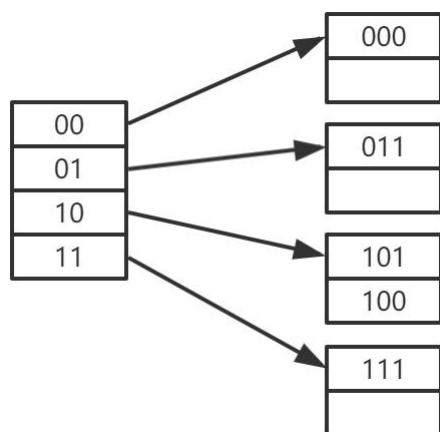
图 4.17 无损连接判断表

1、利用可扩展 hash 方法对以下记录进行 hash 存储：

3, 5, 7, 12, 16

设 hash 函数 $h(x) = x \bmod 8$ ，其中散列函数 $h(k)$ 是一个 b (足够大) 位二进制序列，序列的前 d 位用作索引，来区分每个元素属于哪个桶。

现要求每个桶至多包含 2 个元素，以上元素按从左往右的顺序依次添加。开始时只使用序列的前 1 位作索引（即 $d=1$ ），当桶满时进行分裂， d 相应增大。请画出添加完以上所有元素后，最终的索引结构。

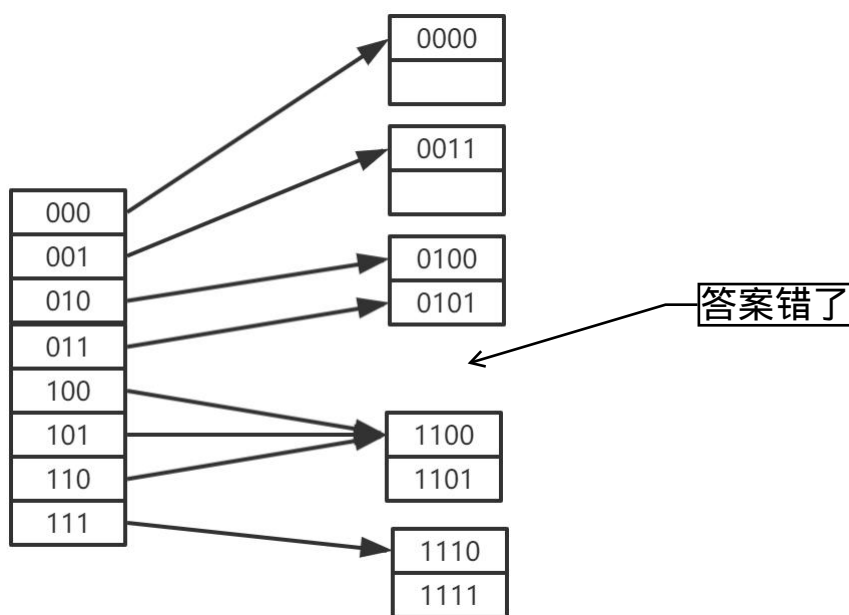


2、利用可扩展 hash 方法对以下记录进行 hash 存储：

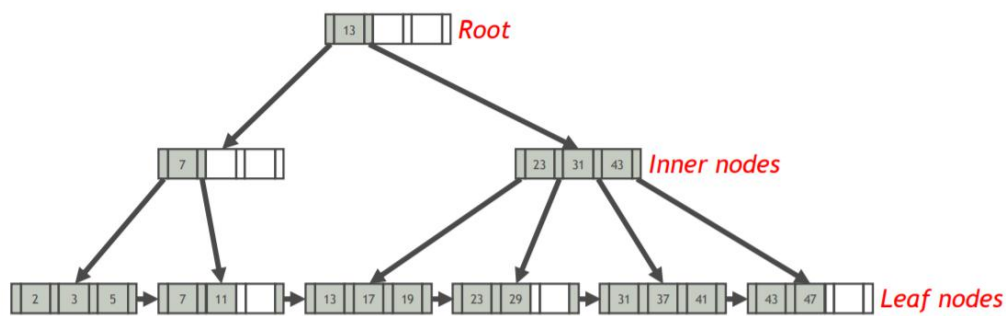
3, 16, 20, 21, 30, 44, 47, 61

设 hash 函数 $h(x) = x \bmod 16$ ，其中散列函数 $h(k)$ 是一个 b (足够大) 位二进制序列，序列的前 d 位用作索引，来区分每个元素属于哪个桶。

现要求每个桶至多包含 2 个元素，以上元素按从左往右的顺序依次添加。开始时只使用序列的前 1 位作索引（即 $d=1$ ），当桶满时进行分裂， d 相应增大。请画出添加完以上所有元素后，最终的索引结构。



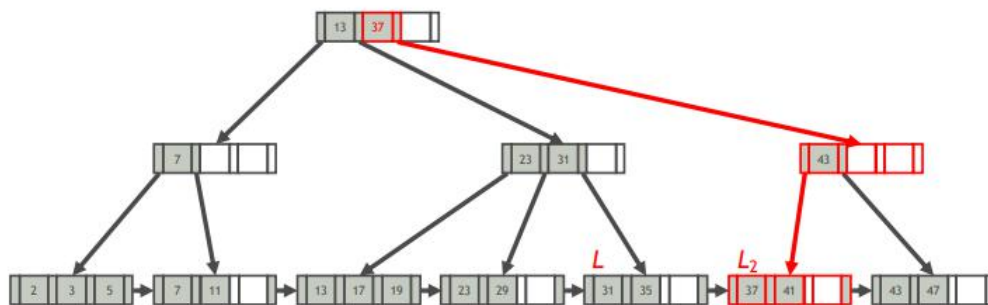
3、已知有如下 b+树：



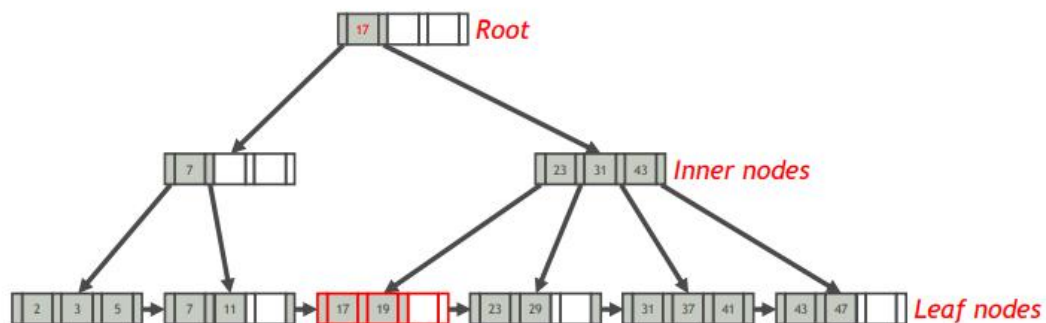
回答下列问题：

- (1) 插入键值为 35 的索引项(index entry)后，该 B+树变成什么样？请绘制出来；
- (2) 删除键值为 13 的索引项(index entry)后，该 B+树变成什么样？请绘制出来；

(1)



(2)



4.设教学管理数据库有如下 3 个关系模式：

S(S#, SNAME, AGE, SEX)

C(C#, CNAME, TEACHER)

SC(S#, C#, GRADE)

其中 S 为学生信息表、SC 为选课表、C 为课程信息表；S#、C#分别为 S、C 表的主码，(S#, C#)是 SC 表的主码，也分别是参照 S、C 表的外码

用户有一查询语句：

Select SNAME

From S, SC, C

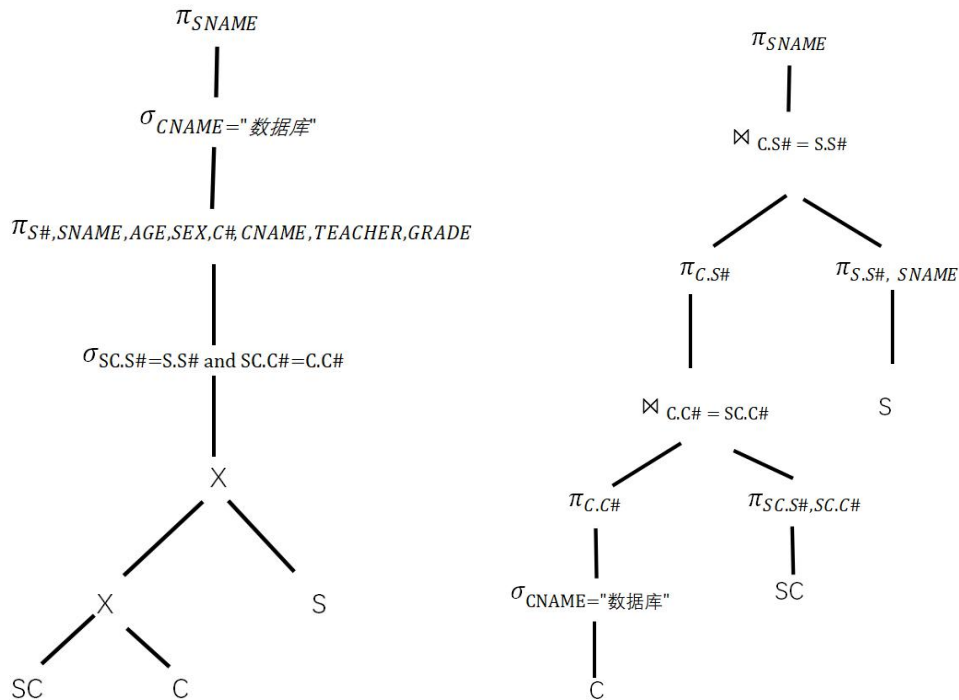
Where SC.S#=S.S# and SC.C#=C.C# and CNAME= “数据库”

检索选学“数据库”课程的学生姓名。

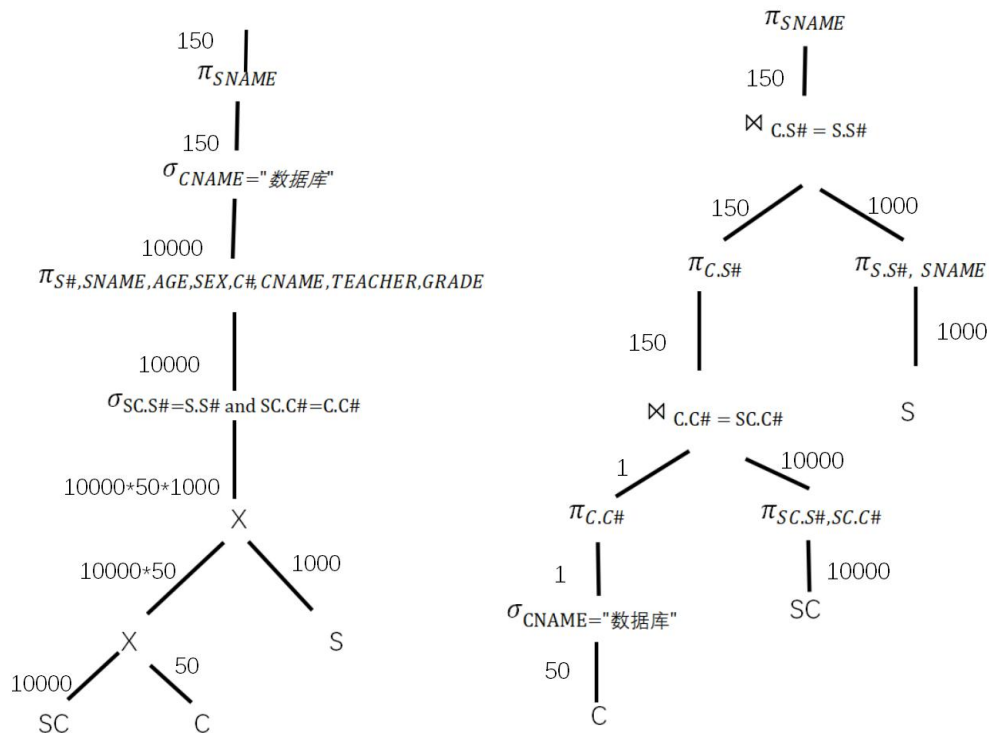
(1) 写出以上 SQL 语句所对应的关系代数表达式。

$$\pi_{SNAME}(\sigma_{CNAME=\text{数据库} \wedge SC.S\# = S.S\# \text{ and } SC.C\# = C.C\#}(SC \times C \times S))$$

(2) 画出上述关系代数表达式所对应的查询计划树。使用启发式查询优化算法，对以上查询计划树进行优化，并画出优化后的查询计划树。



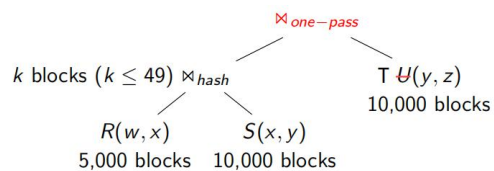
(3) 设 SC 表有 10000 条元组，C 表有 50 条元组，S 表中有 1000 条元组，SC 中满足选修数据库课程的元组数为 150，计算优化前与优化后的查询计划中每一步所产生的中间结果大小



5. 已知关系 $R(w,x), S(x,y), T(y,z)$ 的块数分别为 5000, 10000, 10000。我们准备执行关系代数查询 $(R \bowtie S) \bowtie T$ 。假设缓冲池中有 $M = 101$ 个页可用, R, S, T 上均无索引且未按连接属性排序。请回答下列问题。

- 使用什么算法执行 $R \bowtie S$ 最适合? 说明理由。
- 使用(a)中选择的算法执行 $R \bowtie S$ 的 I/O 代价是多少?
- 如果 $R \bowtie S$ 的结果不超过 49 块, 那么在使用(a)中选择的算法执行 $R \bowtie S$ 时, $R \bowtie S$ 的结果是否需要物化(materialize)到文件中? 说明理由。
- 如果 $R \bowtie S$ 的结果不超过 49 块, 那么使用什么算法将 $R \bowtie S$ 的结果与 T 进行自然连接最合适? 说明理由。
- 使用(d)中选择的算法计算连接结果的 I/O 代价是多少?
- 如果 $R \bowtie S$ 的结果大于 49 块, 那么使用什么算法将 $R \bowtie S$ 的结果与 T 进行自然连接最合适? 说明理由。
- 使用(f)中选择的算法计算连接结果的 I/O 代价是多少?

- Grace 哈希连接算法最合适, 因为 R 和 S 的块数都超过了 M , 一趟算法不可用; R 和 S 无索引, 索引连接不可用; R 和 S 未排序, 排序归并连接不可用。
- $3B(R) + 3B(S) = 45000$ 。
- 不需要。使用 Grace 哈希连接算法执行 $R \bowtie S$, R 和 S 都被分到 100 个桶中, 因此 R 的每个桶大约 50 块, S 的每个桶大约 100 块。在执行 $R_i \bowtie S_i$ 时, 可以使用一趟连接算法, 需要使用内存缓冲区 51 个页面, 还剩 50 个, 能够存放 $R \bowtie S$ 的结果。
- 如果 $R \bowtie S$ 的结果不超过 49 块, 那么在执行 $R \bowtie S$ 时, 结果可以存放在剩余可用缓冲区中, 因此使用一趟连接算法执行 $(R \bowtie S) \bowtie T$ 最合适。



使用一趟连接(one-pass join)执行 $(R \bowtie S) \bowtie U \bowtie T$

- 一趟连接使用51页内存(不计输出缓冲)

$T \bowtie U$ 的缓冲 1页

$R \bowtie S$ 的结果 k 页

输出缓冲 $100 - k$ 页

- I/O代价: $B(U) = 10000$ ($R \bowtie S$ 的结果已在内存中, 无需I/O)

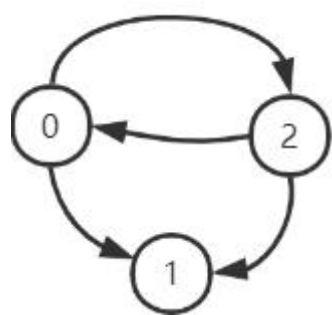
- 策略如下:

- 如果 $k \leq 49$, one-pass join + pipelining
- 如果 $50 < k \leq 300$, nested-loop join + materialization
- 如果 $300 < k \leq 5000$, Grace hash join + pipelining
- 如果 $k > 5000$, Grace hash join + materialization

1、考虑下面的三个事务和它们的一个调度 S（时间从上往下依次增大）。判断 S 是否是冲突可串行化的调度？要求画出优先图并给出判断依据。

T0	T1	T2
r0(A)		
w0(A)		
		r2(A)
		w2(A)
	r1(A)	
r0(B)		
		r2(B)
w0(B)		
		w2(B)
	r1(B)	

解：
不是。



因为优先图中存在环，存在 $r0(B)r2(B)w0(B)w2(B)$ ， $r2(B)$ 与 $w0(B)$ 冲突， $w0(B)w2(B)$ 冲突，无论怎么调整，都不能转化为一个串行调度，所以这不是一个冲突可串行化调度。

2、设 T1、T2、T3 是如下三个事务：

T1: $A:=A+4$

T2: $A:=A*3$

T3: $A:=A^2$

初始 $A=2$

(1) 设三个事务都遵守两段锁协议，按 T2-T3-T1 的顺序执行，请给出一个不产生死锁的可串行化调度，并给出最终 A 的结果

时间	T1	T2	T3
1		LOCK-X(A)	
2		READ(A)	
3		$A:=A*3$	
4		WRITE(A)	
5		UNLOCK(A)	
6			LOCK-X(A)
7			READ(A)
8			$A:=A^2$
9			WRITE(A)

10			UNLOCK(A)
11	LOCK-X(A)		
12	READ(A)		
13	A:=A+4		
14	WRITE(A)		
15	UNLOCK(A)		

A=40

(2) 若这三个事务都遵循两段锁协议，请给出一个产生死锁的调度。

时间	T1	T2	T3
1	LOCK-S(A)		
2	READ(A)		
3		LOCK-S(A)	
4		READ(A)	
5	LOCK-X(A)		
6	等待		
7		LOCK-X(A)	
8		等待	
9			LOCK-S(A)
10			READ(A)
11			LOCK-X(A)
12			等待

3、考虑两个事务 T1, T2。其中，T1 显示账户 A 与 B 的内容：

T1: Read(B);
Read(A);
Display(A+B).

T2 表示从账户 B 转 50 美元到账户 A, 然后显示两个账户的内容：

T2: Read(B);
B := B-50;
Write(B);
Read(A);
A := A+50;
Write(A);
Display(A+B).

请给出一个满足时间戳协议的一个可能的调度。（注：Display(A+B)仅表示显示账户 A 和账户 B 的内容）

解：令 TS (T1) , TS (T2) 分别是事务 T1 和 T2 开始的时间戳，并且 TS (T2) < TS (T1)

T1	T2	W-ts(A)	R-ts(A)	W-ts(B)	R-ts(B)
	Read(B);				TS(T2)
	B := B-50;				
	Write(B);			TS(T2)	
Read(B);					TS(T1)
	Read(A);		TS(T2)		
	A := A+50;				
Read(A);			TS(T1)		
	Write(A);	TS(T2)			
Display(A+B).					
	Display(A+B).				

4、一个带检查点的日志内容如下，结束处发生了故障，请简述恢复算法的过程，并给出 Undo-List 和 Redo-List，以及数据库系统恢复后的 A,B,C 的值。

```

Start of the logs
...
<T4 start>
<T4, B, 1000, 1700>
<T5 start>
<checkpoint {T4, T5}>
<T5, C, 300, 100>
<T5 commit>
<T6 start>
<T6, A, 700, 600>
<T4, B, 1000>
<T4 abort>
← System crash, start recovery

```

解：找到最后的 checkpoint，令 Undo-List=L，本题中为{T4,T5}，从此处向下扫描，若是 start 的事务则加入 Undo-List，若是 commit 或 abort 的事务则从 Undo-List 移除，并添加到 Redo-List。将 Undo-List 中事务涉及的 ABC 赋旧值，将 Redo-List 中的事务重新执行，即对 ABC 赋新值。

Undo-List: T6

Redo-List: T4, T5

A=700, B=1000, C=100

5、设一个数据库系统启动后中，执行 4 个事务 T0、T1、T2 和 T3。四个事务的内容如下：

T0: A := A + 20 (读入数据库元素 A 的值，加上 20 后，再写回 A 的值)

T1: B := B - 10 (读入数据库元素 B 的值，减去 10 后，再写回 B 的值)

T2: C := C * 2 (读入数据库元素 C 的值，乘以 2 后，再写回 C 的值)

T3: D := D + 15 (读入数据库元素 D 的值，加上 15 后，再写回 D 的值)

除了这四个事务外，系统中无其他事务执行。设四个事务开始前，数据库元素 A、

B、C、D 的值分别为 A = 50, B = 30, C = 35, D = 15。在执行这四个事务的过程中，系统发生了故障。系统重启后，经故障恢复，数据库元素 A、B、C、D 的值被恢复为 A = 50, B = 20, C = 70, D = 15。故障恢复时，数据库系统日志文件中包含如下 12 条日志记录，这里只给出部分日志记录。已知该数据库管理系统使用基于 undo-redo 日志的故障恢复技术，这段日志中仅有 1 个不停机检查点(又称模糊检查点)。

1	
2	
3	
4	<start checkpoint (T0, T2)>
5	<end checkpoint>
6	
7	
8	
9	
10	
11	
12	

请根据上述信息，回答下列问题：

a.将日志文件补充完整，直接在上面的日志文件中填写。一个事务 T 启动时向日志文件中写入日志记录<T, start>;提交时向日志文件中写入日志记录<T, commit>;中止时向日志文件中写入日志记录<T, abort>;对数据库元素 X 进行修改时向日志文件中写入日志记录<T, X,X 的旧值, X 的新值>。

1	<T0, start>
2	<T0, A, 50, 70>
3	<T2, start>
4	<start checkpoint (T0, T2)>
5	<end checkpoint>
6	<T1, start>
7	<T1, B, 30, 20>
8	<T1, commit>
9	<T2, C, 35, 70>
10	<T3, start>
11	<T3, D, 15, 30>
12	<T2, commit>

b.在故障恢复过程中，哪些事务需要 redo，哪些事务需要 undo。说明理由。

解：由日志文件知，T1、T2 事务已提交，所以需要 redo

T0、T3 事务未提交，所以需要 undo

c.在故障恢复过程中，还会向日志文件添加什么日志记录？说明理由。

解：检查点记录，系统周期性的执行检查点，内存中所有的日志记录输出到日志

文件，将内存中所有修改了的数据块输出到数据文件，将一个日志记录输出到日志文件，该日志记录称为检查点记录，检查点记录的内容：建立检查点时刻所正在执行的事务清单，这些事务最近一个日志记录地址。