4.1、下表给出的关系R为第几范式？是否存在操作异常？若存在，则将其分解为高一级范式。分解完成的高级范式中是否可以避免分解前关系中存在的操作异常？

解：它为1NF。因为该关系的候选码为（工程号，材料号），而非主属性“开工日期”和“完工日期”部分函数依赖于候选码的子集“工程号”，即：



∴它不是2NF。

它存在操作异常，如果工程项目确定后，若暂时未用到材料，则该工程的数据因缺少码的一部分（材料号）而不能进入到数据库中，出现插入异常。若某工程下马，则删去该工程的操作也可能丢失材料方面的信息。

将其中的部分函数依赖分解为一个独立的关系，则产生如下所示的两个2NF关系模式：

R1（工程号，材料号，数量，价格）

R2(工程号，开工日期，完工日期)

分解后，新工程确定后，尽管还未用到材料，该工程数据可在关系R2中插入。某工程数据删除时，仅对关系R2操作，也不会丢失材料方面的信息。

4.2 D+={DHG}

4.3 F中A+=ABCDE CD+=CDE

G中 A+=ABCDE CD+=CDE据此，可以得知F中每个函数依赖均在G+中，G中每个函数依赖均在F+中（具体描述略），所以F=G

4.4 ACD+=ABCD=U并且对于ACD的任一真子集的闭包均不等于U所以ACD为候选码

4.5 (1)AB+=ABCD

AC+=ACD

DE+=ABCDE

⑵ 求R的所有候选码BE，CE，DE

⑶ 求F的最小覆盖

Fm={B→C，DE→B，C→D，D→A}

4.6 Fm={A→C，C→A，B→C，D→C}

设：

GA={C→A,B→A,B→C,D→A,D→C,BD→A}，得(A) GA+=A[\*]

C不属于(A) GA+ [\*]

A→C不冗余

判断C→A是否冗余

设：GB={A→C,B→A,B→C,D→A,D→C,BD→A}，得(A) GB+=C

[\*]A不属于(C) GA+ [\*]C→A不冗余

判断B→A是否冗余

设：GC={A→C,C→A,B→C,D→A,D→C,BD→A}，得(B) GC+=BCA

[\*]A属于(B) GC+ [\*]B→A冗余

判断B→C是否冗余

设：GD={A→C,C→A,D→A,D→C,BD→A}，得(B) GD+=B

[\*]C不属于(B) GD+ [\*]B→C不冗余

判断D→A是否冗余

设：GE={A→C,C→A,B→C,D→C,BD→A}，得(D) CE+=DCA

[\*]A属于(D) GE+ [\*]D→A冗余

判断A→C是否冗余

设：GF={A→C,C→A,B→C,BD→A}，得(D) GF+=D

[\*]C不属于(D) GF+ [\*]D→C不冗余

判断BD→A是否冗余

设：GG={A→C,C→A,B→C,D→C}，得(BD) GG+=BDCA

[\*]A属于(BD) GG+ [\*]BD→A冗余F={A→C,C→A,B→C,D→C}

由于各函数依赖左部都为单属性，故：

Fm={A→C,C→A,B→C,D→C}

B)τ={AC，BC，DC，BD}

4.7 不具有

4.8 均不具有

4.9 F={B🡪C,D🡪A}ρ={AD,BC}

KEY=BD,有损分解，保持依赖，均属于BCNF

F={ AB🡪C,C🡪A, C🡪D }ρ={ACD,BC}

KEY=AB,BC,无损分解，依赖不保持，均属于BCNF

F={ A🡪BC,C🡪AD }ρ={ABC, AD}

KEY=A,C,无损分解，依赖保持，均属于BCNF

F={ A🡪B, B🡪C, C🡪D }ρ={ACD,AB}

KEY=A,无损分解，依赖不保持，AB属于BCNF，ACD属于2NF

F={ A🡪B, B🡪C, C🡪D }ρ={AD,AB,CD}

KEY=A,有损分解，依赖不保持，均属于BC范式

F={ AB🡪C,C🡪A, D🡪A }ρ={ACD,BC}

KEY=AB,BC,无损分解，依赖不保持，均属于BCNF

4.10 具有无损连接性

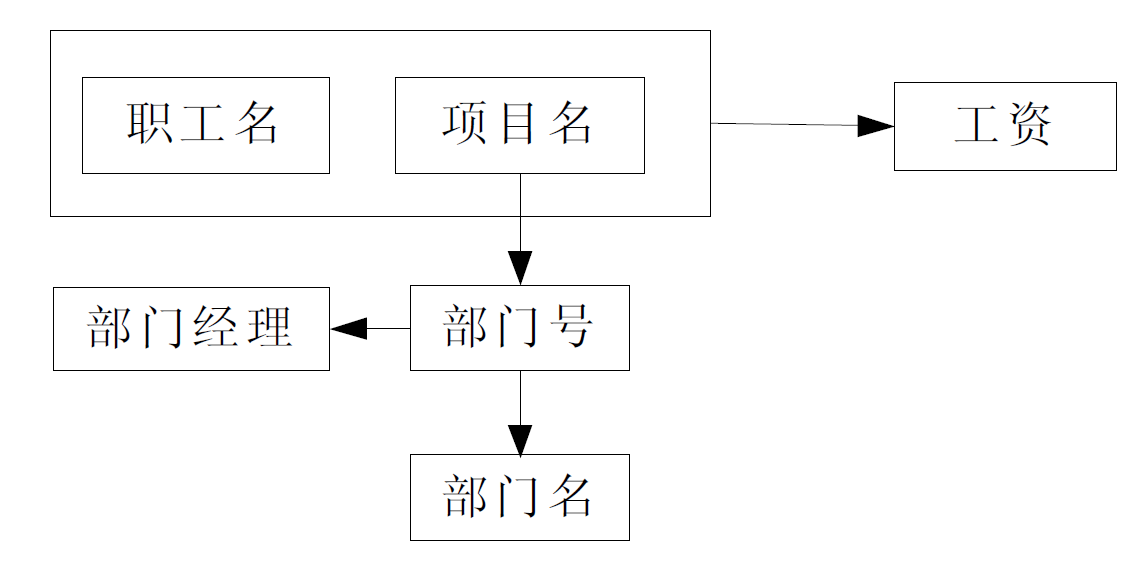
（1）候选码：AB

（2）Fmin={ AB→C，C→D，D→E }

(3) *ρ*＝{R1（ABC），R2（CD），R3（DE）}

4.11 Fm={AB🡪C BE🡪C C🡪ABDEG}

4.12 （1）函数依赖



候选码：（职工名，项目名）

由于部门名，部门经理部分函数依赖于码，因此该模式只满足第一范式，所以采用投影分解法将一个 1NF 的关系分解为多个 2NF 的关系，即：SLC 分解 为两个关系模式，以消除这些部分函数依赖

R1（职工名，项目名，工资） ∈ 2NF

R2（项目名，部门号，部门名，部门经理） ∈ 2NF

但是 2NF 关系模式 R2 中，部门名和部门经理传递函数依赖于项目名，即 R2 中 存在非主属性对码的传递函数依赖。所以进一步把 SL 分解为两个关系模式，以 消除传递函数依赖：

R2（项目名，部门号） ∈ 3NF

R3（部门号，部门名，部门经理） ∈ 3NF

4.13

解：因为关系模式至少是1NF关系，即不包含重复组并且不存在嵌套结构，给出的数据集显然不可直接作为关系数据库中的关系，改造为1NF的关系如下。

|  |  |  |
| --- | --- | --- |
| 系名 | 课程名 | 教师名 |
| 计算机系 | DB | 张伟 |
| 计算机系 | DB | 王强 |
| 信控系 | AI | 孙兰 |
| 信控系 | AI | 宋灵 |
| 信息安全系 | DS | 张燕 |
| 信息安全系 | DS | 李杰 |
| 数学系 | CM | 刘海 |

4.14 解：各关系模式如下：

学生(学号,姓名,出生年月,系名,班级号,宿舍区)

班级(班级号,专业名,系名,人数,入校年份)

系(系名,系号,系办公地点,人数)

社团(社团名,成立年份,地点,人数)

加入社团（社团名，学号，学生参加社团的年份）

学生(学号,姓名,出生年月,系名,班级号,宿舍区)

“学生”关系的最小函数依赖集为：

Fmin={学号→姓名,学号→班级号,学号→出生年月,学号→系名，系名→宿舍区}

以上关系模式中存在传递函数依赖，如:学号→系名，系名→宿舍区

候选键是学号,外部键是班级号,系名。

班级(班级号,专业名,系名,人数,入校年份)

“班级”关系的最小函数依赖集为：

Fmin={(系名,专业名)→班级号,班级号→人数，班级号→入校年份，班级号→系名，班级号→专业名}

(假设没有相同的系，不同系中专业名可以相同)

以上关系模式中不存在传递函数依赖。

“(系名,专业名)→班级号”是完全函数依赖。

候选键是(系名,专业名),班级号，外部键是系名。

系(系名,系号,系办公地点,人数)

“系”关系的最小函数依赖集为： Fmin={系号→系名,系名→系办公地点,系名→人数，系名→系号}

以上关系模式中不存在传递函数依赖

候选键是系名,系号

社团(社团名,成立年份,地点,人数)

“社团”关系的最小函数依赖集为： Fmin={社团名→成立年份,社团名→地点,社团名→人数}

以上关系模式中不存在传递函数依赖。

候选键是社团名

加入社团（社团名，学号，学生参加社团的年份）

“加入社团”关系的最小函数依赖集为： Fmin={（社团名，学号）→学生参加社团的年份}

“（社团名，学号）→学生参加社团的年份”是完全函数依赖。

以上关系模式中不存在传递函数依赖。

候选键是（社团名，学号）。