

8.1

我们知道如下定理：如果 $R1 \cap R2 \rightarrow R1$ 或 $R1 \cap R2 \rightarrow R2$ ，则分解 $\{R1, R2\}$ 是无损连接分解。

根据这道题目的描述，我们可以设 $R1 = (A, B, C)$ ， $R2 = (A, D, E)$ ， $R1 \cap R2 = A$ 。由于 A 是候选码，因此 $R1 \cap R2 \rightarrow R1$ 。

8.13

不保留依赖关系 $B \rightarrow D$ 。 $F1, F$ 对 (A, B, C) 的限制是 $A \rightarrow ABC$ ， $A \rightarrow AB$ ， $A \rightarrow AC$ ， $A \rightarrow BC$ ， $A \rightarrow B$ ， $A \rightarrow C$ ， $A \rightarrow A$ ， $B \rightarrow B$ ， $C \rightarrow C$ ， $AB \rightarrow AC$ ， $AB \rightarrow ABC$ ， $AB \rightarrow BC$ ， $AB \rightarrow AB$ ， $AB \rightarrow A$ ， $AB \rightarrow B$ ， $AB \rightarrow C$ ， AC （与 AB 相同）， BC （与 AB 相同）， ABC （与 AB 相同）。 $F2, F$ 对 (A, D, E) 的限制是 $A \rightarrow ADE$ ， $A \rightarrow AD$ ， $A \rightarrow AE$ ， $A \rightarrow DE$ ， $A \rightarrow A$ ， $A \rightarrow D$ ， $A \rightarrow E$ ， $D \rightarrow D$ ， $(E$ 与 A 相同)， AD ， AE ， DE ， ADE （与 A 相同）。 $(F1 \cup F2) +$ 很容易看出不包含 $B \rightarrow D$ ，因为 $F1 \cup F2$ 中唯一的 FD ，左边是 B ， $B \rightarrow B$ ，一个平凡的 FD 。

8.19

我们知道 $B \rightarrow D$ 是不平凡的，箭头左侧不是超码。我们得到关系 $\{(A, B, C), (E, A), (B, D)\}$ 。其属于 BC 范式。

8.20

$R' = \{(A, B, C), (C, D, E), (B, D), (E, A)\}$ 。

模式 (A, B, C) 包含候选键。因此， R' 是第三种常规形式依赖性保持无损连接分解。注意，原始数据 $R = (A, B, C, D, E)$ 已经是 $3NF$ 。因此，没有必要像我们上面那样应用算法。单个原始模式通常是无损连接，依赖保持分解。

8.29

- a) $B \rightarrow BD$
 $BD \rightarrow ABD$
 $ABD \rightarrow ABCD$
 $ABCD \rightarrow ABCDE$
因此， $B + = ABCDE$

- b) $A \rightarrow BCD$ (题目给定)
 $A \rightarrow ABCD$ (A 自反律)
 $BC \rightarrow DE$ (题目给定)
 $ABCD \rightarrow ABCDE$ (用 ABCD 增广律)
 $A \rightarrow ABCDE$ (传递性)
 $AF \rightarrow ABCDEF$ (增广律 F)
- c) 我们看到 D 在 dep 中是无关紧要的。1 和 2 是根据 dep.3。删除这两个，我们得到了一套新的规则 $A \rightarrow BC$ $BC \rightarrow E$ $B \rightarrow D$ $D \rightarrow A$ 现在注意 B+ 是 ABCDE，特别是 FD $B \rightarrow E$ 可以从该组中确定。因此，属性 C 在第三依赖性中是无关紧要的。删除此属性，并与第三个 FD 组合，我们得到最终的规范： $A \rightarrow BC$ $B \rightarrow DE$ $D \rightarrow A$ 在这里，任何 FD 中没有属性是无关紧要的。
- d) r1 (A, B, C)
r2 (B, D, E)
r3 (D, A)
r4 (A, F)
- e) r1(A, B, C, D)
r2(A, F)
r3(A, E)
- f) 如果我们直接在前面的规范封面中使用函数依赖，我们就无法得到上面的分解。但是，我们可以从规范的封面推断出原始的依赖关系，如果我们使用它们进行 BCNF 分解，我们就可以得到相同的分解。