数据库理论作业week14/P208

曹广杰

15352015 数据科学与计算机

授课教师：刘玉葆

Content

[8.19]给出以下模式R的一个无损连接的BCNF分解

$$r(A, B, C, D, E)$$

有如下函数依赖：

$$\begin{gathered} A \rightarrow BC \\ CD \rightarrow E \\ B \rightarrow D \\ E \rightarrow A \end{gathered}$$

无损分解表示将一个关系集合分解为多个关系集合的时候，没有信息的损失，即分解之后的子关系的自然连接（natural join）恰好为原来的数据集。

确定关系是否符合BCNF

首先需要确定当前的关系是否符合BCNF。

BCNF的设计主要是为了去除冗余，要求每一个非平凡的函数依赖都是超码。这种限定使得每一条函数依赖都是与超码紧密相连的，是最简单最直接的去除冗余的方法。

下面分析函数依赖，可以看到以上的4个非平凡函数依赖中，$B \rightarrow D$并不能通过Amstrong公理推导出所有的属性，因此该函数依赖并不符合BCNF。接下来进行无损分解算法的讨论：

无损的BCNF分解：

满足**BCNF**分解

为了使每一个函数依赖都符合BCNF，这里将非平凡的函数依赖$B \rightarrow D$分离出来作为一个新的关系集合，那么该函数集合的左值就变成超码，该子关系集符合BCNF。

满足无损特征

在拆分出来之后，由于属性D是原子的，在从原有的属性集合中删除该属性的时候，不会同时删除其子属性，此时分离出的属性集合之间由超码B关联，而同时B又是属性集{B, D}的候选码，则在自然连接的时候，可以实现准确的意义对应，于是不会造成信息的损失。

故最终的分解结果为：

$$\{A, B, C, E\}$$
$$\{B, D\}$$

[8.26]用Amstrong公理证明分解率的正确有效性

分解率：

$$\{\alpha \rightarrow \beta\gamma\} \Rightarrow \{\alpha \rightarrow \beta, \alpha \rightarrow \gamma\}$$

推导过程如下：

$$\begin{aligned} \alpha &\rightarrow \beta\gamma \\ \beta\gamma &\rightarrow \beta \\ \beta\gamma &\rightarrow \gamma \\ \Rightarrow \alpha &\rightarrow \beta \\ \alpha &\rightarrow \gamma \end{aligned}$$

[8.27]用以下函数依赖集计算函数依赖B^+

函数依赖集合：

$$\begin{aligned} A &\rightarrow BC \\ CD &\rightarrow E \\ B &\rightarrow D \\ E &\rightarrow A \end{aligned}$$

需要在函数依赖的集合中根据包含属性B的函数依赖进行拓展，通过拓展得到，B的闭包是：

$$\{B, D\}$$

[8.28]证明对于属性集$r(A,B,C,D,E)$的分解$(A,B,C)\&(C,D,E)$不是无损分解。

因为C并不是属性集r的超码，所以在原来的属性集中在属性C下是允许有重复的数据的。

设若存在元组c_1和c_2，二者有相同的属性C，则在查询的时候，两个子集就会混淆，信息就会丢失。

[8.29]考虑关系模式$r(A,B,C,D,E,F)$上的函数依赖集：

$$\begin{gathered}A\rightarrow BCD\\BC\rightarrow DE\\B\rightarrow D\\D\rightarrow A\end{gathered}$$

1. 计算a_+：

$$\begin{gathered}B\rightarrow D\\BD\rightarrow ABD\\ABD\rightarrow ABCD\\ABCD\rightarrow ABCDE\end{gathered}$$

2. 使用Amstrong公理证明AF是超码

   AF是超码，则AF的闭包AF_+是所有的属性集合ABCDE：

$$\begin{gathered}AF\rightarrow ABCDF\\BCDF\rightarrow ABCDEF\end{gathered}$$

   包含了所有的属性，故而是超码。

3. 计算函数依赖集的正则覆盖

   所谓正则覆盖是原有函数依赖的集合的计算结果，使得包含其所有的函数依赖同时又不包含任何无关属性。

   首先合并函数依赖的左半部：

$$\begin{gathered}A\rightarrow BCD\\BC\rightarrow DE\\B\rightarrow D\\D\rightarrow A\end{gathered}$$

   接下来逐个排除无关属性：

   $B\rightarrow D$和$D\rightarrow A$中没有无关属性，基于二者进行化简操作。

   1. 无关属性化简

      考虑$BC\rightarrow DE$中左值的闭包，B的闭包为$\{D,A,C,E\}$，所以在这个函数依赖中C为无关属性，可以化简。得到$B\rightarrow DE$;

   2. 相同左值合并

      此时得到的$B\rightarrow DE$与已有的函数依赖$B\rightarrow D$可以合并——因为左值相同，于是得到新的函数依赖集合：

$$\begin{gathered}A\rightarrow BCD\\B\rightarrow DE\\D\rightarrow A\end{gathered}$$

   3. 无关属性化简

考虑$A \to BCD$中的右值部分，由于B的闭包为$\{A, B, C, D, E\}$，所以可以在该函数依赖的右侧部分进行化简。

$$\begin{aligned} A \to BCD \Rightarrow A \to B \\ \&A \to C \\ \&A \to D \\ \because B \to D \& B \to E \end{aligned}$$

因此可以省略掉B原本推导出的内容，而不必在意B的闭包原来是什么。此时的结果已经没有相同的左值需要进行化简，于是我们获得了最后的函数依赖集合：

$$A \to BC$$
$$B \to DE$$
$$D \to A$$

4. 基于正则覆盖实现r的3NF分解

已经获得以上函数依赖的正则覆盖，此时进行第3范式的分解时，第三范式要求在每一个属性都绑定到超码的条件上添加了一个最小放宽条件——即便当前的非平凡函数依赖不能满足与超码进行绑定，那么对右值中的所有的属性：

1. 在左值中未出现
2. 仅在右值中出现

也都应该属于候选码。属于候选码就意味着对于这些属性出现次数的限定，因为对于候选码而言，在每一个元组中是各自不同的，具有很高的区分度。所以将属性限定为是候选码（超码的最小子集）就可以限定其出现的次数。

综上，在使用正则覆盖实现第三范式的分解时候，需要首先考虑是否满足BCNF的条件，由于以上的函数依赖都没有包含属性F，所以都左值都不是超码。因此需要将F单独分出成为一个子集，剩下的部分其实是符合BCNF限定的。

因此，基于正则覆盖的函数依赖集合第三范式的分解结果如下：

$$\begin{aligned} \{A, B, C\} \\ \{B, D, E\} \\ \{D, A\} \\ \{A, F\} \end{aligned}$$

由于之前的推导过程显示B是属性集的候选码，而又存在$A \to B \& D \to A$可以得到AD也均为超码，所以以上的函数依赖都是符合BCNF范式的，可以直接作为分解的子集。

5. 利用原始的函数依赖集，给出r的BCNF分解

根据第一个函数依赖关系可以将属性集划分为两个部分：

$$r_1(A, B, C, D) \& r_2(A, E, F)$$

由于r_2是不符合BCNF范式的求解的，因此根据Amstrong传递律得到新的划分方式：

$$\begin{aligned} r_1(A, B, C, D) \\ r_2(A, E) \\ r_3(A, F) \end{aligned}$$

6. 利用正则覆盖得到以上与r相同的BCNF分解

根据正则覆盖函数依赖集合可以得到原始的函数依赖集合：

$$
\begin{aligned}
&A \to BC(canonial)\\
&B \to DE(canonial)\\
\Rightarrow\ &A \to BCD\\
&BC \to DE\\
&B \to D\\
&D \to A
\end{aligned}
$$

因此可以推导出与上一题相同的BCNF分解。

End