

[示例]考虑一图书馆的关系数据库。

BOOKS (TITLE , AUTHOR , PNAME , LC_NO)

注：PNAME为出版社名，LC_NO为图书馆图书编目号

PUBLISHERS (PNAME , PADDR , PCITY)

注：PADDR为出版社地址，PCITY为出版社所在地

BORROWERS (NAME , ADDR , CITY , CARD_NO)

注：NAME为读者名，ADDR为读者所在地址，CITY为读者所在城市，CARD_NO为图书证号。

LOANS (CARD_NO , LC_NO , DATE)

注：DATE为借出日期

➤为方使用户使用，定义了视图**XLOANS**:

$$XLOANS = \Pi_S(\sigma_F(LOANS \times BORROWERS \times BOOKS))$$

S = TITLE , AUTHOR , PNAME , LC_NO , NAME , ADDR , CITY , CARD_NO , DATE

F = (BORROWERS.CARD_NO = LOANS.CARD_NO) \wedge (BOOKS.LC_NO = LOANS.LC_NO)

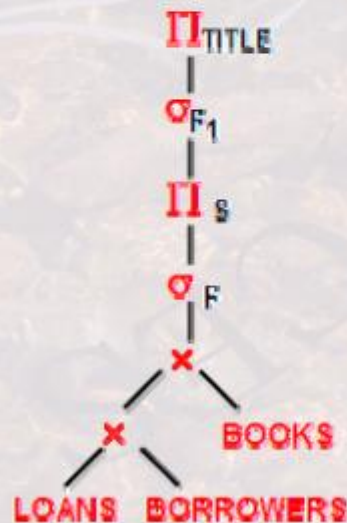
查出1978年1月1日前被借出的所有书的书名

查询：“查出1978年1月1日前被借出的所有书的书名”

“SELECT Title FROM XLOANS WHERE Date ≤ 1/1/78”

$\Pi_{\text{TITLE}}(\sigma_{\text{DATE} \leq \text{"1/1/78"}}(\text{XLOANS}))$

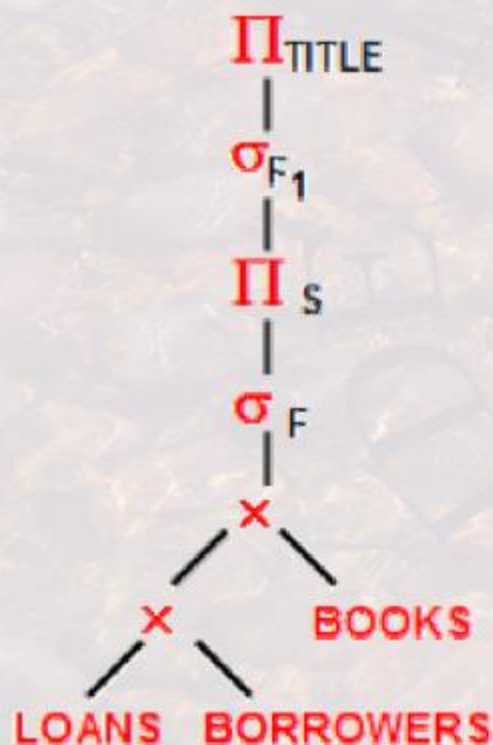
语法树



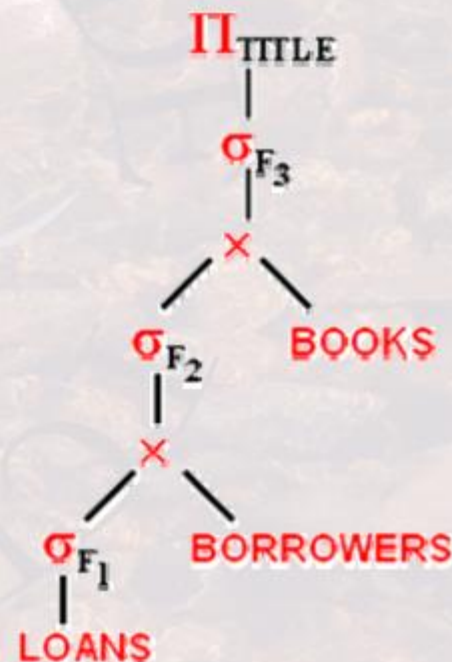
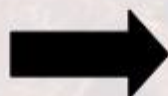
由树叶向树根反映了操作先后次序

其中： $F = F_2 \wedge F_3$ ； $F_1 = \text{"DATE} \leq 1/1/78\text{"}$ ；
 $F_2 = \text{"BORROWERS.CARD_NO} = \text{LOANS.CARD_NO"}$ ；
 $F_3 = \text{"BOOKS.LC_NO} = \text{LOANS.LC_NO"}$ 。

- (1) 依据定理L4，把形如 $\sigma_{F_1 \wedge F_2 \wedge \dots \wedge F_n}(E)$ 的选择表达式变成串接形式 $\sigma_{F_1}(\sigma_{F_2}(\dots(\sigma_{F_n}(E))))$ 。
- (2) 对每个选择，依据定理L4至L9，尽可能把它移至树的底部。



- $\sigma_{F_2 \wedge F_3} = \sigma_{F_2}(\sigma_{F_3})$
- σ_{F_1} 通过交换移动到底部
- σ_{F_2} 通过交换移动到底部
- $\Pi_{TITLE}(\Pi_S) = \Pi_{TITLE}$



其中: $F = F_2 \wedge F_3$; $F_1 = \text{"DATE} \leq 1/1/78\text{"}$;
 $F_2 = \text{"BORROWERS.CARD_NO} = \text{LOANS.CARD_NO"}$;
 $F_3 = \text{"BOOKS.LC_NO} = \text{LOANS.LC_NO"}$ 。

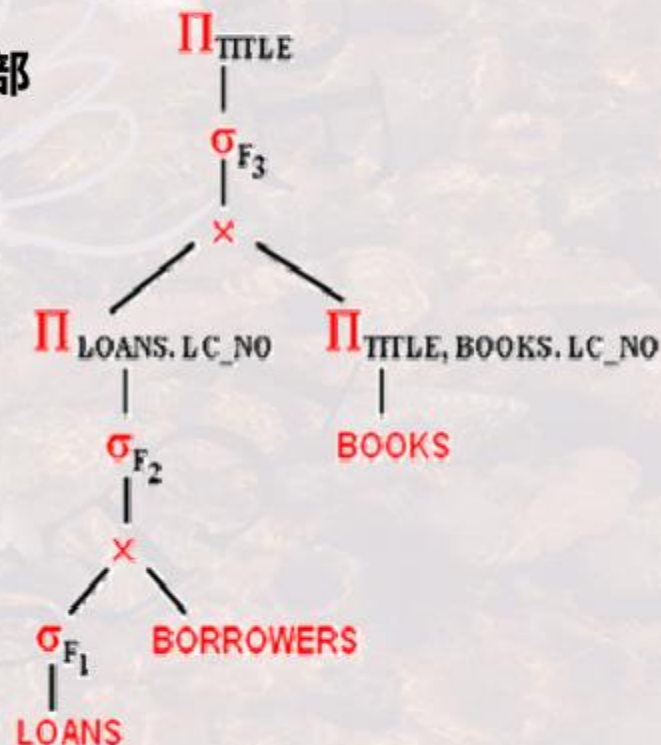
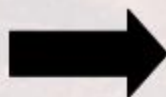
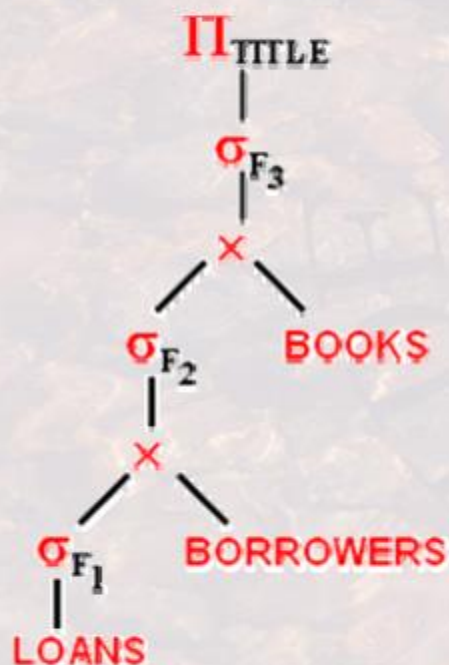
(S3)对每个投影，依据定理L3，L7，L10和L5，尽可能把它移至树的底部。
如果一个投影是对某表达式所有属性进行的，则去掉之。

• $\Pi_{\text{TITLE}} = \Pi_{\text{TITLE}} (\Pi_{\text{TITLE}, \text{BOOKS.LC_NO}, \text{LOANS.LC_NO}})$

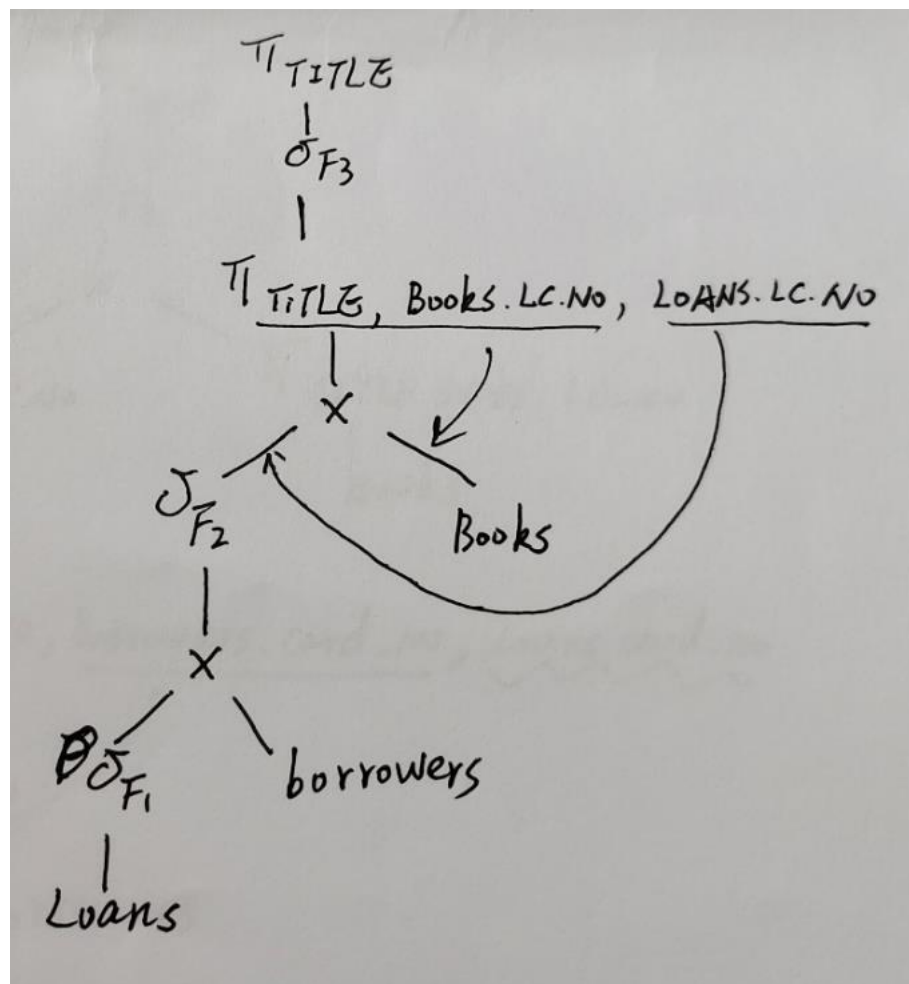
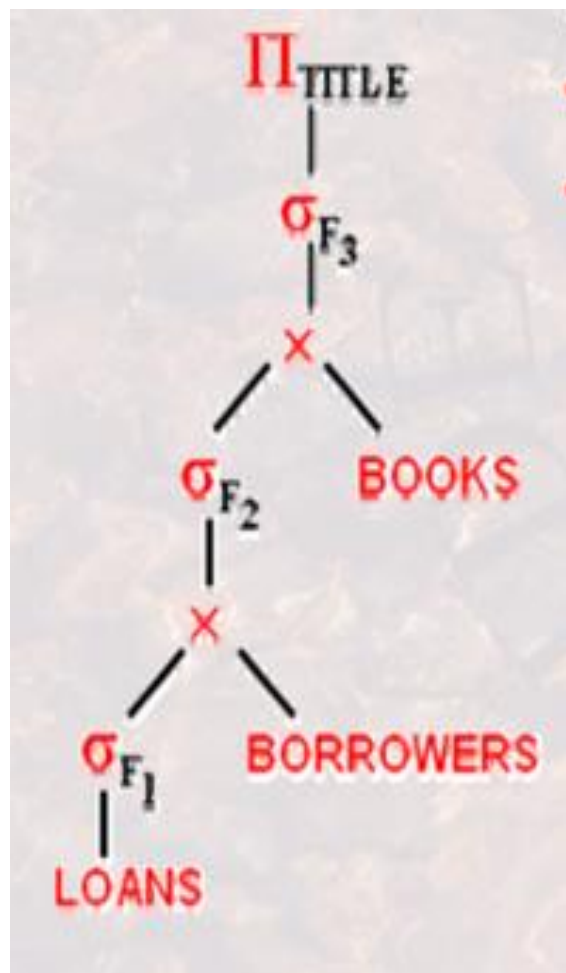
使其包含 F_3 涉及的属性

• $\Pi_{\text{TITLE}, \text{BOOKS.LC_NO}}$ 移动到底部

• $\Pi_{\text{LOANS.LC.NO}}$ 移动到底部

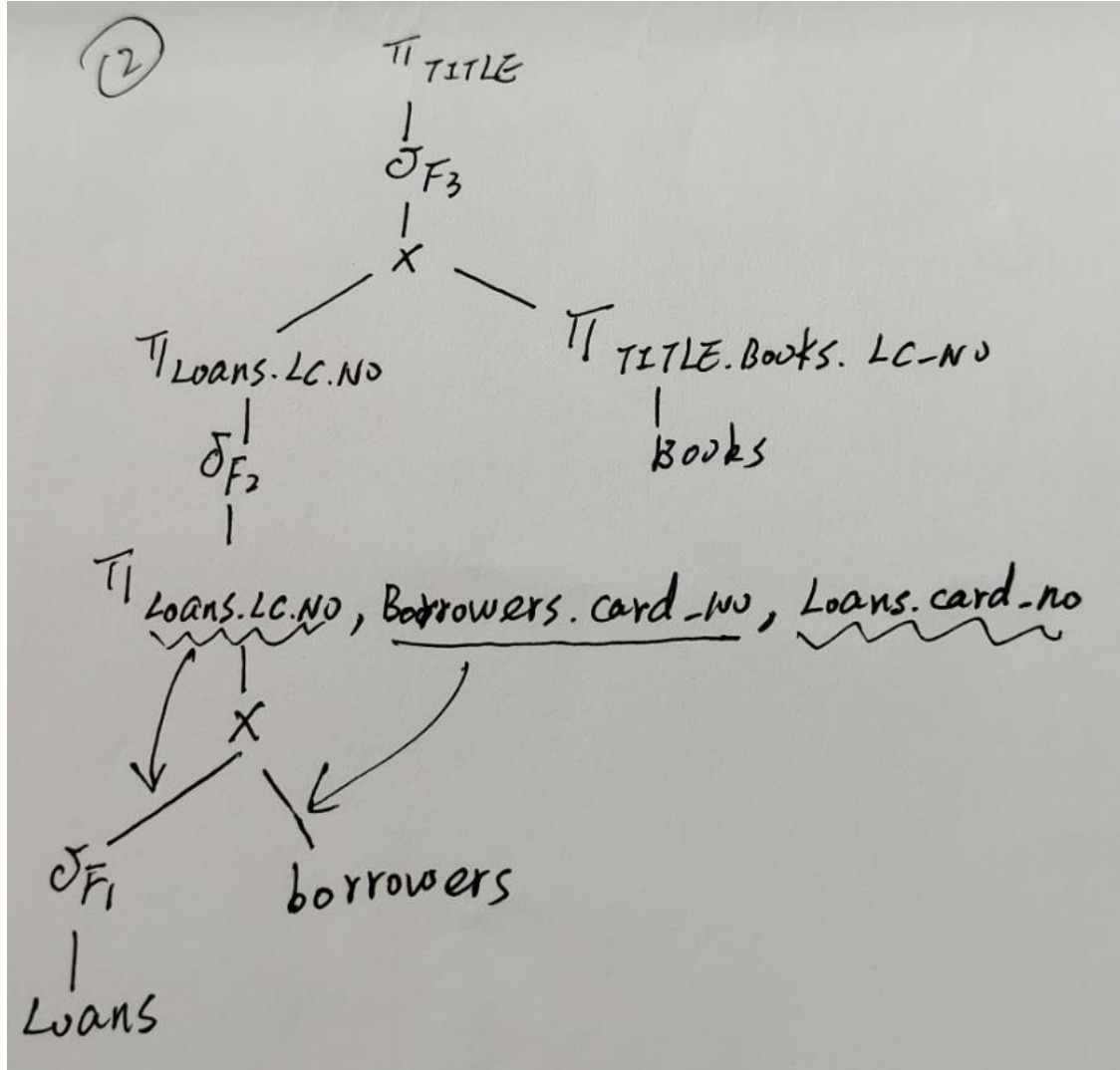
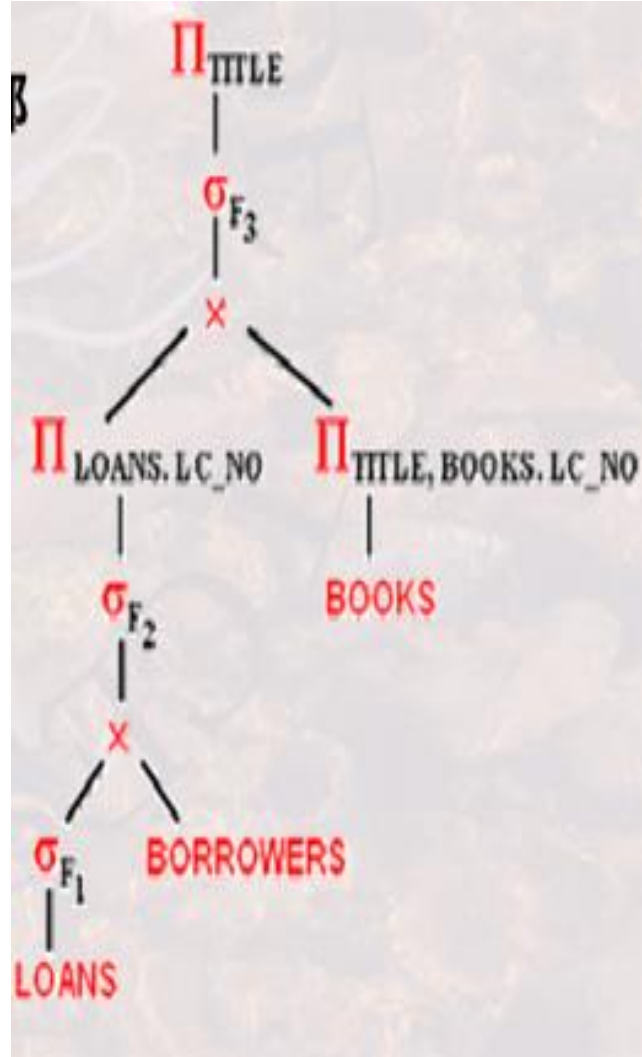


其中: $F = F_2 \wedge F_3$; $F_1 = \text{"DATE} \leq 1/1/78\text{"}$;
 $F_2 = \text{"BORROWERS.CARD_NO} = \text{LOANS.CARD_NO"}$;
 $F_3 = \text{"BOOKS.LC_NO} = \text{LOANS.LC_NO"}$.



其中: $F = F_2 \wedge F_3$; $F_1 = \text{"DATE} \leq 1/1/78\text{"}$;
 $F_2 = \text{"BORROWERS.CARD_NO} = \text{LOANS.CARD_NO"}$;
 $F_3 = \text{"BOOKS.LC_NO} = \text{LOANS.LC_NO"}$ 。

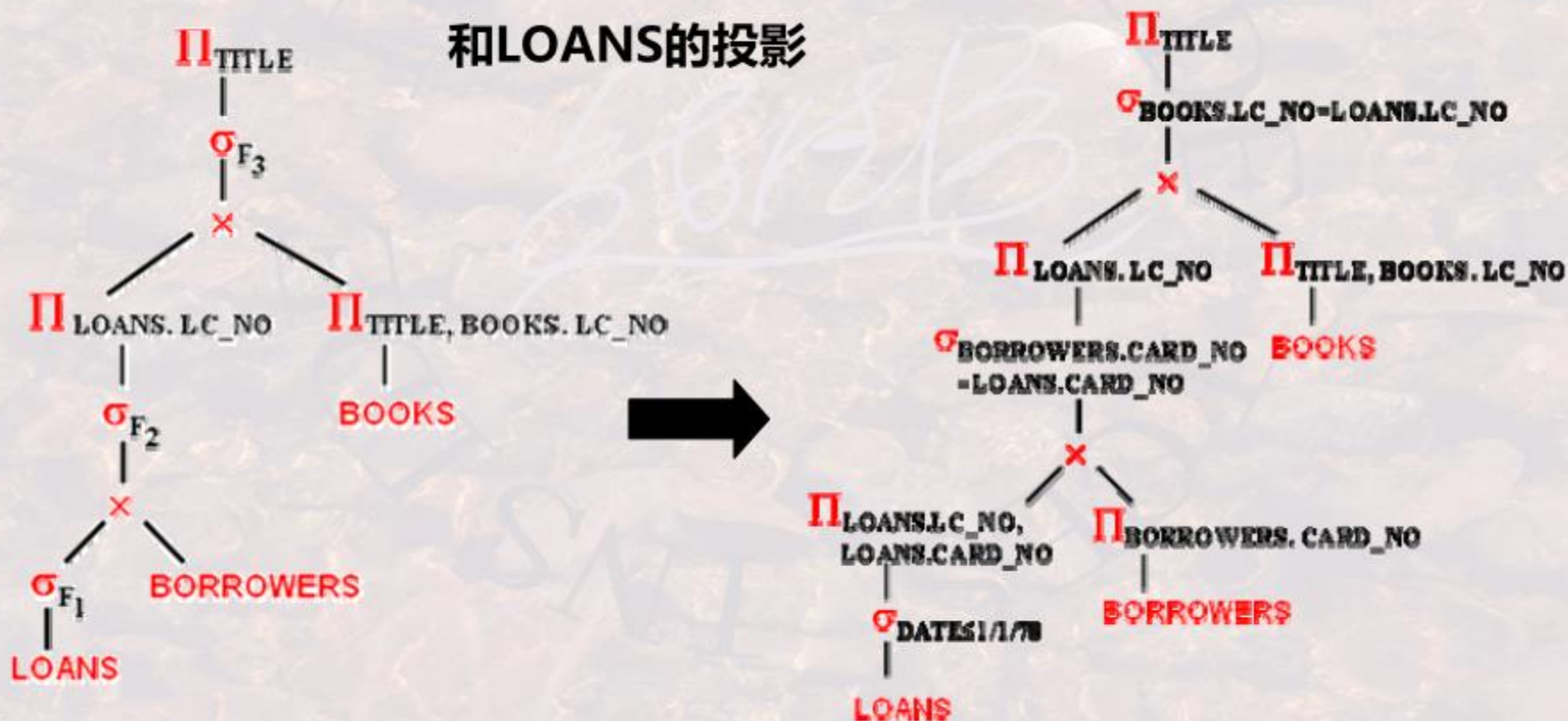




其中: $F = F_2 \wedge F_3$; $F_1 = \text{"DATE} \leq 1/1/78\text{"}$;
 $F_2 = \text{"BORROWERS.CARD_NO} = \text{LOANS.CARD_NO"}$;
 $F_3 = \text{"BOOKS.LC_NO} = \text{LOANS.LC_NO"}$.



•类似处理BORROWERS 和LOANS的投影



其中: $F = F_2 \wedge F_3$; $F_1 = \text{"DATE} \leq 1/1/78\text{"}$;
 $F_2 = \text{"BORROWERS.CARD_NO} = \text{LOANS.CARD_NO"}$;
 $F_3 = \text{"BOOKS.LC_NO} = \text{LOANS.LC_NO"}$ 。

(S5)按以下方式分组：
每个二元运算结点
(积、并、差、连接等)
和其所有一元运算直接
祖先结点放在一组；对
于其后代结点，若后代
结点是一串一元运算且
以树叶为终点，则将这
些一元运算结点放在该
组中；若该二元运算结
点是笛卡儿积，且其后
代结点不能和它组合成
等连接，则不能将后代
结点归入该组。
(S6)产生一个程序：它
以每组结点为一步，但
后代组先执行。

