# 第十一章 并发控制

# 封锁粒度

- 封锁对象的大小称为封锁粒度(Granularity)

- 封锁的对象:逻辑单元，物理单元

  例：关系数据库中的封锁对象

  - 逻辑单元: 属性值、属性值的集合、元组、关系、索引项、整个索引、整个数据库等
  - 物理单元：页（数据页或索引页）、物理记录等

# 选择封锁粒度原则

- 封锁粒度与系统的并发度和并发控制的开销密切相关。
  - 封锁的粒度越大，数据库所能够封锁的数据单元就越少，并发度就越小，系统开销也越小；
  - 封锁的粒度越小，并发度较高，但系统开销也就越大

# 选择封锁粒度的原则（续）

例1：事务$T_1$需要修改元组$L_1$，事务$T_2$需要修改元组$L_2$，$L_1$和$L_2$位于同一个数据页面A。

- 若封锁粒度是数据页，事务$T_1$需要对数据页A加锁，$T_2$被迫等待，直到$T_1$释放A。
- 如果封锁粒度是元组，则$T_1$和$T_2$可以同时对$L_1$和$L_2$加锁，不需要互相等待，提高了系统的并行度。

封锁粒度越小，并发度越高。

# 选择封锁粒度的原则（续）

例2：事务$T_3$需要读取整个表

- 若封锁粒度是元组，$T_3$必须对表中的每一个元组加锁，开销极大
- 若封锁粒度是关系，$T_3$只需要一次加锁
- 若锁粒度是数据页呢？

封锁粒度越小，封锁开销越大。

# 选择封锁粒度的原则（续）

- 多粒度封锁(Multiple Granularity Locking)

  在一个系统中同时支持多种封锁粒度供不同的事务选择

- 选择封锁粒度

  同时考虑封锁开销和并发度两个因素，适当选择封锁粒度

  - 需要处理大量元组的用户事务：以关系为封锁单元
  - 需要处理多个关系的大量元组的用户事务：以数据库为封锁单位
  - 只处理少量元组的用户事务：以元组为封锁单位

# 11.7 封锁的粒度

## 11.7.1 多粒度封锁

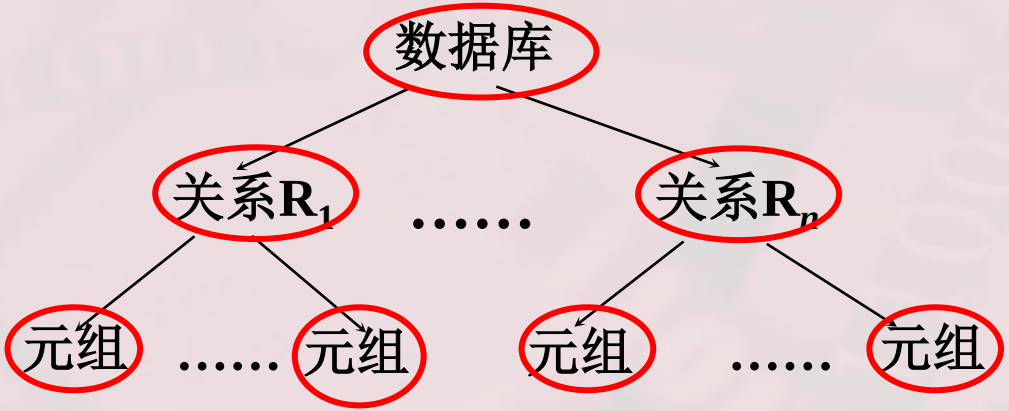
## 11.7.2 意向锁

# 11.7.1 多粒度封锁

- 多粒度树
  - 以树形结构来表示多级封锁粒度
  - 根结点是整个数据库，表示最大的数据粒度
  - 叶结点表示最小的数据粒度

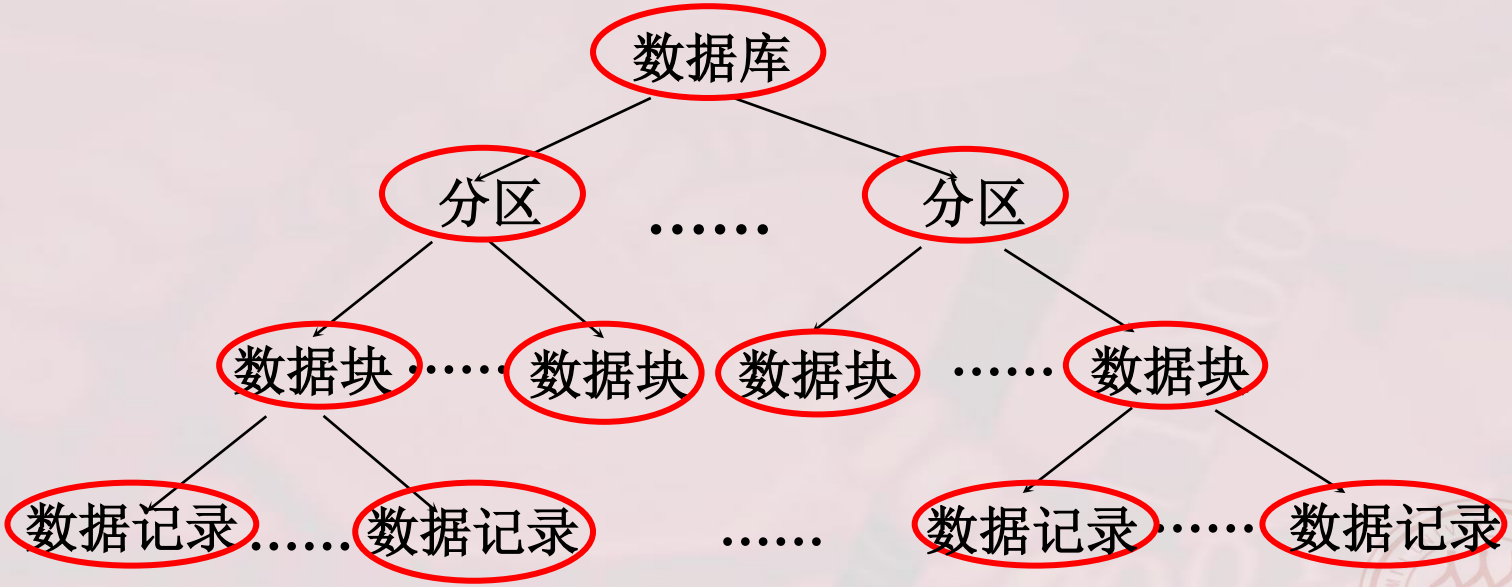
# 多粒度封锁（续）

例1：三级粒度树。根结点为数据库，数据库的子结点为关系，关系的子结点为元组。



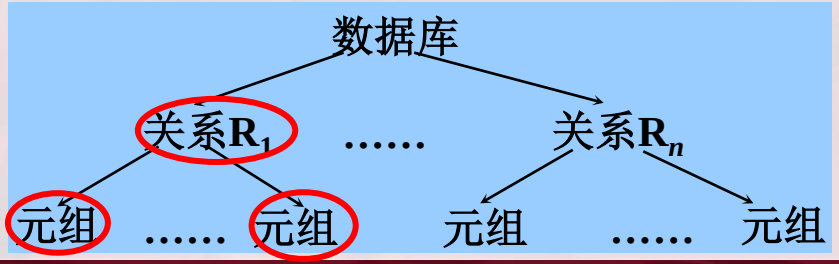
# 多粒度封锁（续）

例2：四级粒度树



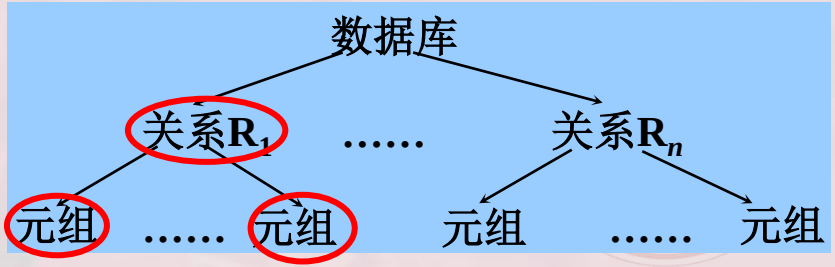
# 多粒度封锁协议

- 允许多粒度树中的每个结点被独立地加锁
- 对一个结点加锁意味着这个结点的所有后裔结点也被加以同样类型的锁
- 在多粒度封锁中一个数据对象可能以两种方式封锁：显式封锁和隐式封锁



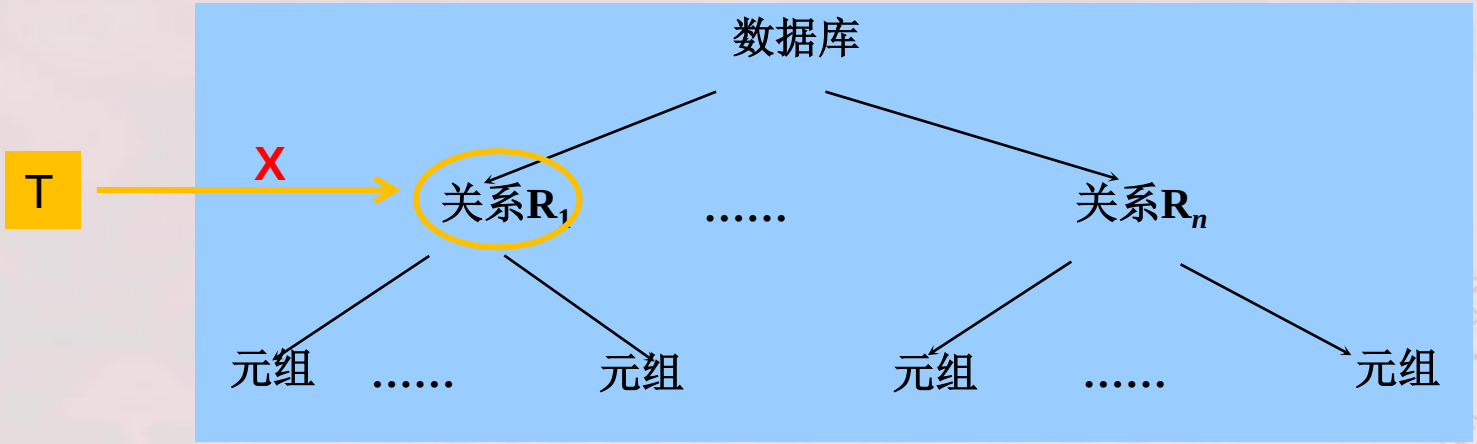
# 显式封锁和隐式封锁

- 显式封锁: 直接加到数据对象上的封锁
- 隐式封锁:是该数据对象没有独立加锁，是由于其上级结点加锁而使该数据对象加上了锁
- 显式封锁和隐式封锁的效果是一样的



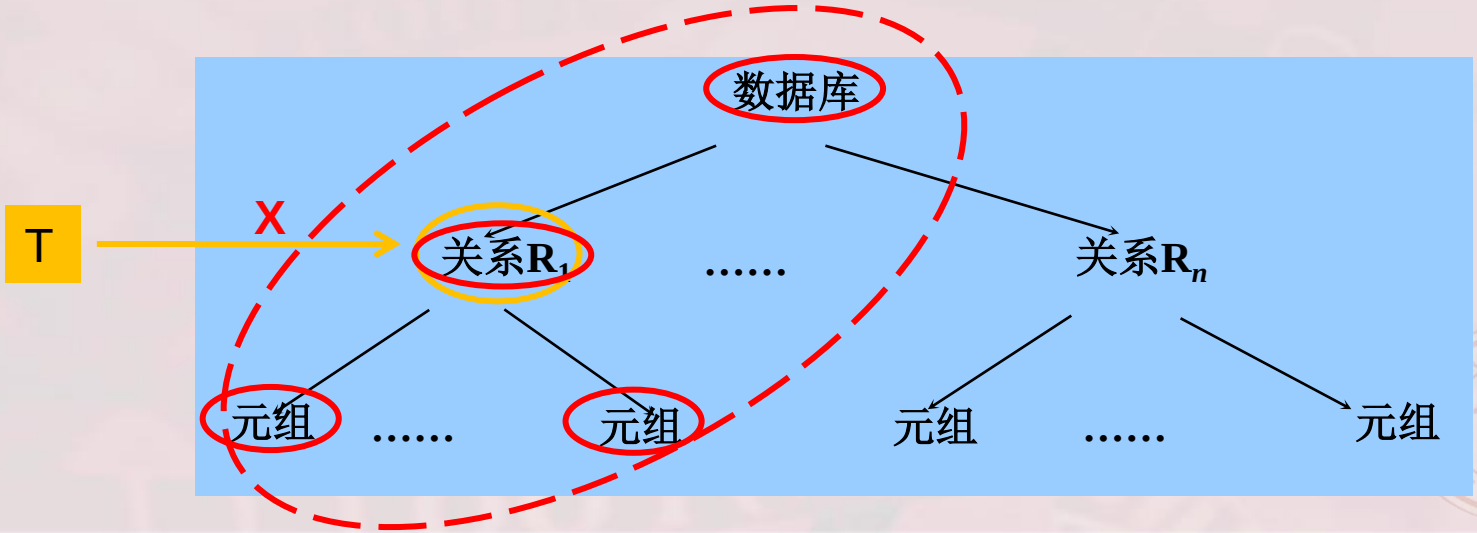
# 显式封锁和隐式封锁（续）

- 系统检查封锁冲突时
  - 要检查显式封锁
  - 还要检查隐式封锁



# 显式封锁和隐式封锁（续）

- 系统检查封锁冲突时
  - 要检查显式封锁
  - 还要检查隐式封锁



# 显式封锁和隐式封锁（续）

- 对某个数据对象加锁，系统要检查
  - 该数据对象
    - 有无显式封锁与之冲突
  - 所有上级结点
    - 检查本事务的显式封锁是否与该数据对象上的隐式封锁冲突(由上级结点已加的封锁造成的）
  - 所有下级结点
    - 看上面的显式封锁是否与本事务的隐式封锁（将加到下级结点的封锁）冲突

# 11.7 封锁的粒度

## 11.7.1 多粒度封锁

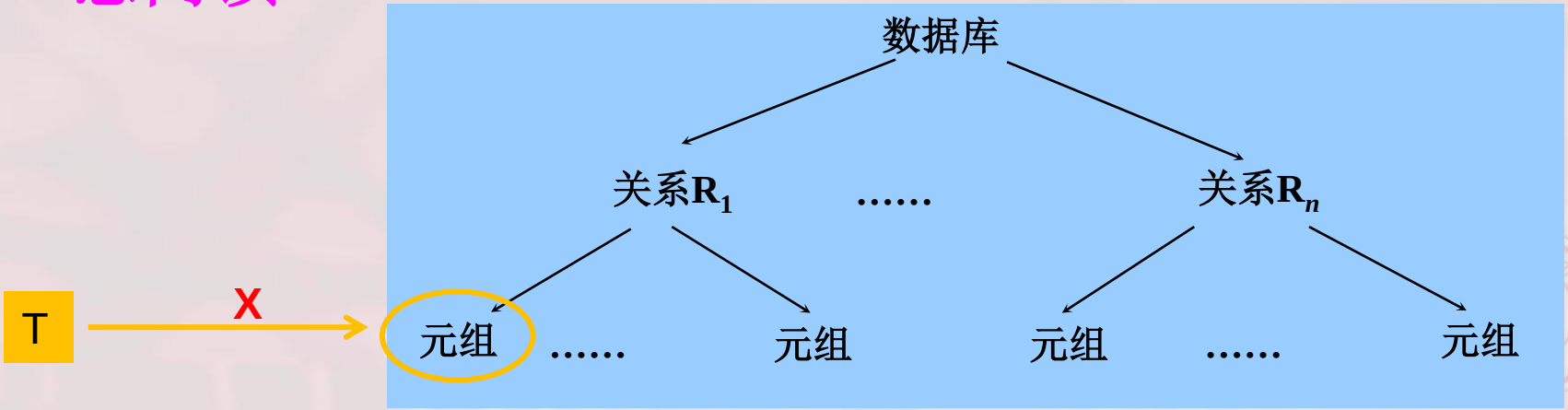
## 11.7.2 意向锁

# 11.7.2 意向锁

- 引进意向锁（intention lock）目的
  - 提高对某个数据对象加锁时系统的检查效率

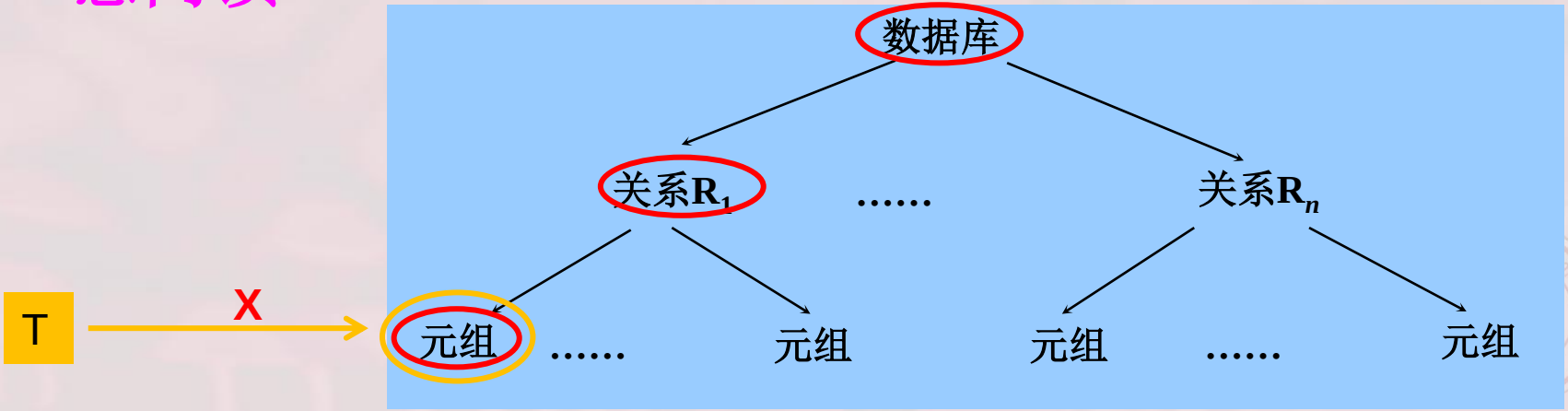
# 意向锁(续)

- 如果对一个结点加意向锁，则说明该结点的下层结点正在被加锁
- 对任一结点加基本锁，必须先对它的上层结点加意向锁



# 意向锁(续)

- 如果对一个结点加意向锁，则说明该结点的下层结点正在被加锁
- 对任一结点加基本锁，必须先对它的上层结点加意向锁



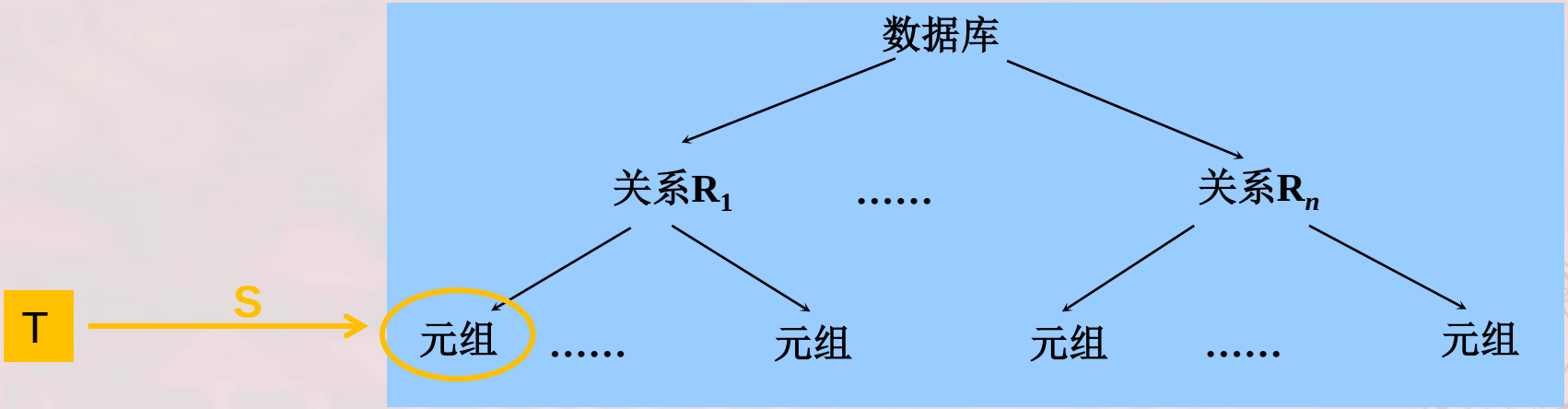
# 常用意向锁

- 意向共享锁(Intent Share Lock，简称IS锁)
- 意向排它锁(Intent Exclusive Lock，简称IX锁)
- 共享意向排它锁(Share Intent Exclusive Lock，简称SIX锁)

# 意向锁（续）

## ❖IS锁

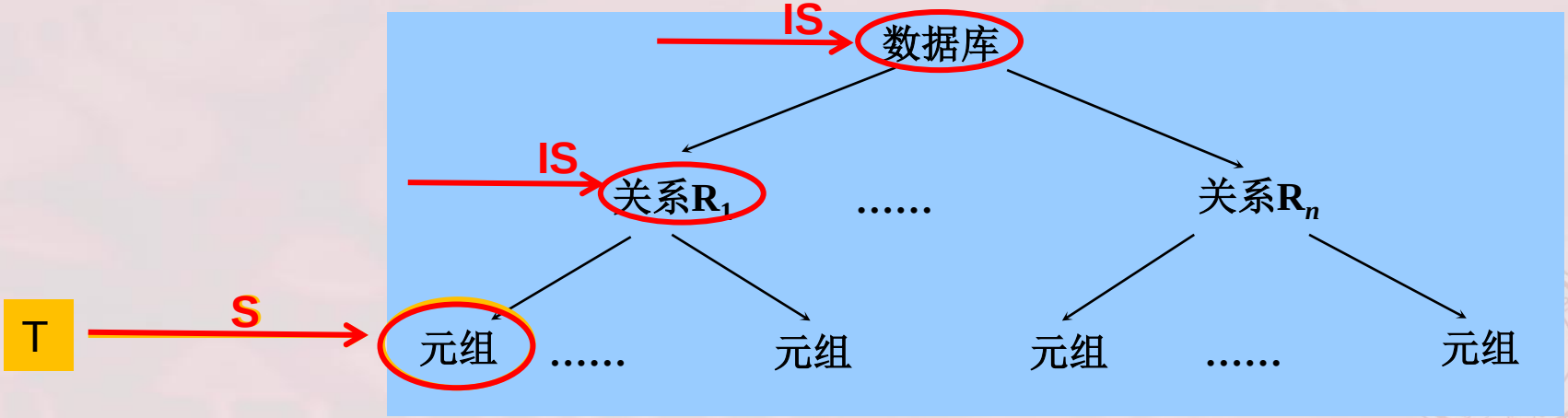
- 如果对一个数据对象加IS锁，表示它的后裔结点拟（意向）加S锁。



# 意向锁（续）

## ❖IS锁

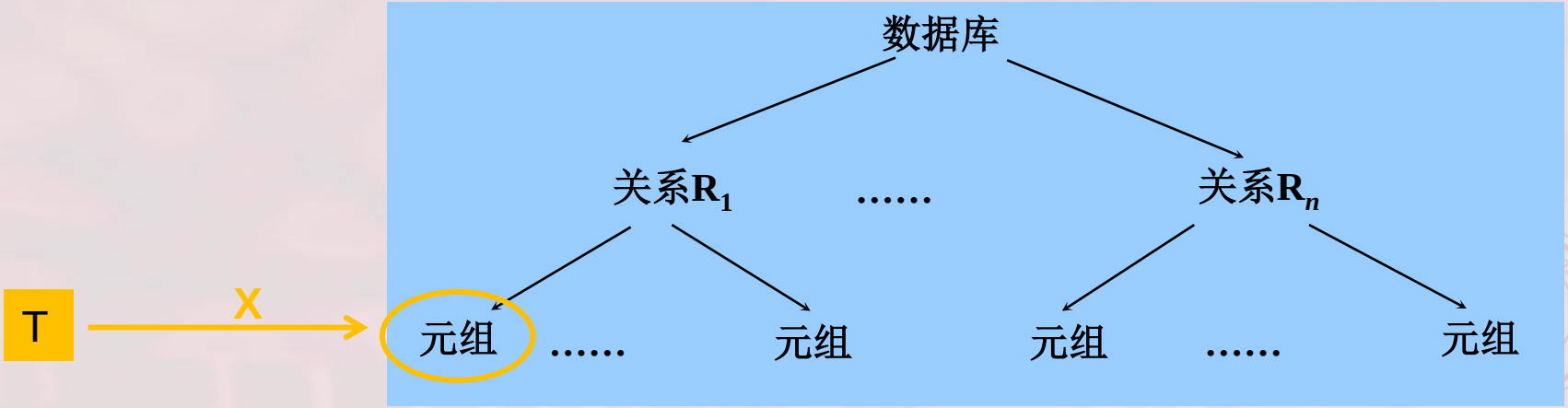
■ 如果对一个数据对象加IS锁，表示它的后裔结点拟（意向）加S锁。



# 意向锁（续）

## ❖IX锁

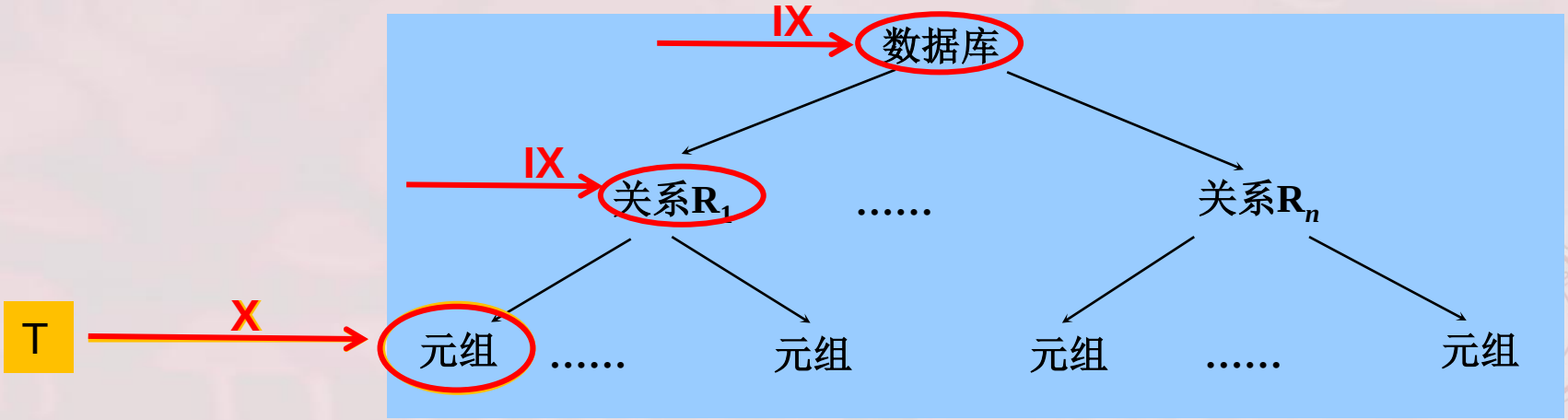
- 如果对一个数据对象加IX锁，表示它的后裔结点拟（意向）加X锁。



# 意向锁（续）

## ❖IX锁

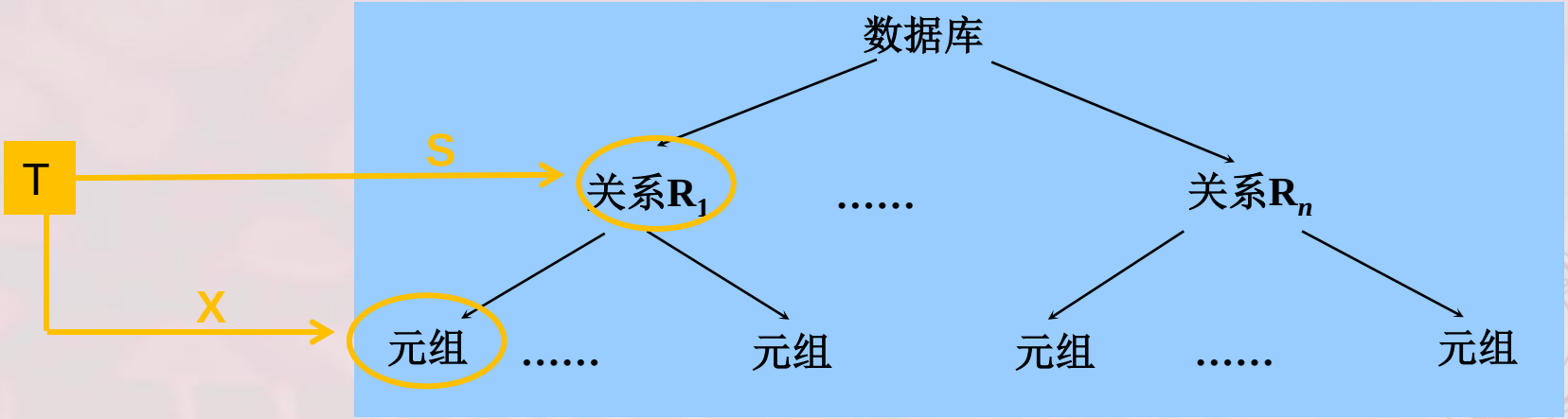
■ 如果对一个数据对象加IX锁，表示它的后裔结点拟（意向）加X锁。



# 意向锁（续）

## ❖SIX锁

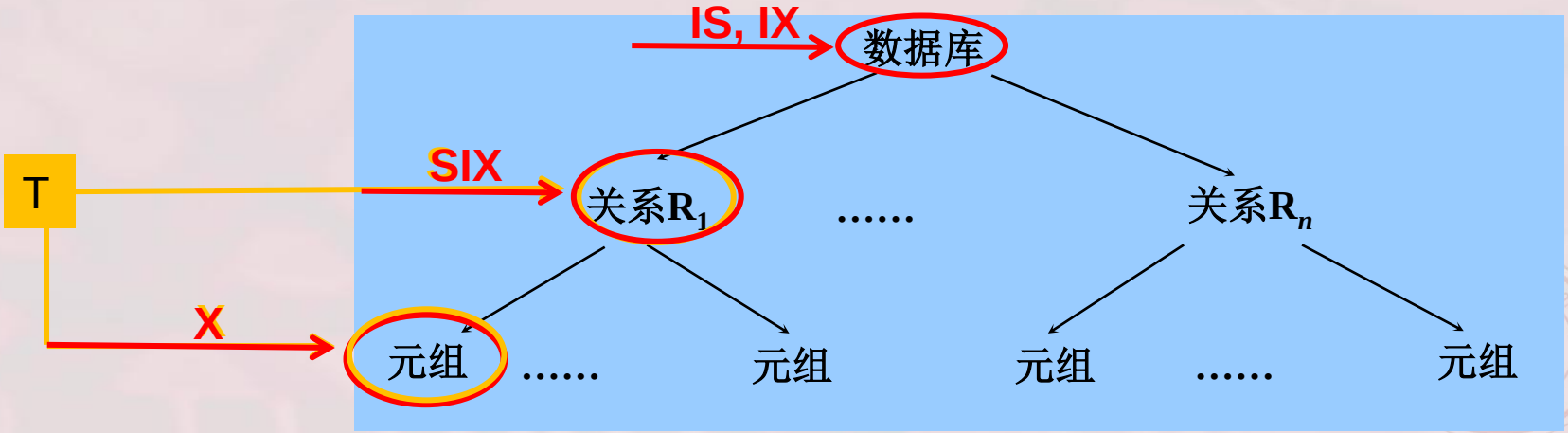
- 如果对一个数据对象加SIX锁，表示对它加S锁，再加IX锁，即SIX = S + IX。



# 意向锁（续）

## ❖SIX锁

- 如果对一个数据对象加SIX锁，表示对它加S锁，再加IX锁，即SIX = S + IX。



# 意向锁（续）

## 数据锁的相容矩阵

| $T_1$ \ $T_2$ | S | X | IS | IX | SIX | – |
|---|---|---|---|---|---|---|
| S | Y | N | Y | N | N | Y |
| X | N | N | N | N | N | Y |
| IS | Y | N | Y | Y | Y | Y |
| IX | N | N | Y | Y | N | Y |
| SIX | N | N | Y | N | N | Y |
| – | Y | Y | Y | Y | Y | Y |

Y=Yes，表示相容的请求　　N=No，表示不相容的请求

# 意向锁（续）

## 数据锁的相容矩阵

| $T_1$ \ $T_2$ | S | X | IS | IX | SIX | – |
|---|---|---|---|---|---|---|
| S | Y | N | Y | N | N | Y |
| X | N | N | N | N | N | Y |
| IS | Y | N | Y | Y | Y | Y |
| IX | N | N | Y | Y | N | Y |
| SIX | N | N | Y | N | N | Y |
| – | Y | Y | Y | Y | Y | Y |

Y=Yes，表示相容的请求　　N=No，表示不相容的请求

(a) 数据锁的相容矩阵

# 意向锁（续）

## 数据锁的相容矩阵

| $T_1$ \ $T_2$ | S | X | IS | IX | SIX | – |
|---|---|---|---|---|---|---|
| S | Y | N | Y | N | N | Y |
| X | N | N | N | N | N | Y |
| IS | Y | N | Y | Y | Y | Y |
| IX | N | N | Y | Y | N | Y |
| SIX | N | N | Y | N | N | Y |
| – | Y | Y | Y | Y | Y | Y |

Y=Yes，表示相容的请求　　N=No，表示不相容的请求

(a) 数据锁的相容矩阵

# 意向锁（续）

## 数据锁的相容矩阵

| $T_1$ \ $T_2$ | S | X | IS | IX | SIX | – |
|---|---|---|---|---|---|---|
| S | Y | N | Y | N | N | Y |
| X | N | N | N | N | N | Y |
| IS | Y | N | Y | Y | Y | Y |
| IX | N | N | Y | Y | N | Y |
| SIX | N | N | Y | N | N | Y |
| – | Y | Y | Y | Y | Y | Y |

Y=Yes，表示相容的请求　　N=No，表示不相容的请求

(a) 数据锁的相容矩阵

# 意向锁（续）

## 数据锁的相容矩阵

| $T_1$ \ $T_2$ | S | X | IS | IX | SIX | – |
|---|---|---|---|---|---|---|
| S | Y | N | Y | N | N | Y |
| X | N | N | N | N | N | Y |
| IS | Y | N | Y | Y | Y | Y |
| IX | N | N | Y | Y | N | Y |
| SIX | N | N | Y | N | N | Y |
| – | Y | Y | Y | Y | Y | Y |

Y=Yes，表示相容的请求　　N=No，表示不相容的请求

(a) 数据锁的相容矩阵

# 意向锁（续）

## 数据锁的相容矩阵

| $T_1$ \ $T_2$ | S | X | IS | IX | SIX | – |
|---|---|---|---|---|---|---|
| S | Y | N | Y | N | N | Y |
| X | N | N | N | N | N | Y |
| IS | Y | N | Y | Y | Y | Y |
| IX | N | N | Y | Y | N | Y |
| SIX | N | N | Y | N | N | Y |
| – | Y | Y | Y | Y | Y | Y |

Y=Yes，表示相容的请求　　N=No，表示不相容的请求

(a) 数据锁的相容矩阵

# 意向锁（续）

## 数据锁的相容矩阵

| $T_1$ \ $T_2$ | S | X | IS | IX | SIX | – |
|---|---|---|---|---|---|---|
| S | Y | N | Y | N | N | Y |
| X | N | N | N | N | N | Y |
| IS | Y | N | Y | Y | Y | Y |
| IX | N | N | Y | Y | N | Y |
| SIX | N | N | Y | N | N | Y |
| – | Y | Y | Y | Y | Y | Y |

Y=Yes，表示相容的请求　　N=No，表示不相容的请求

(a) 数据锁的相容矩阵

# 意向锁（续）

## 数据锁的相容矩阵

| $T_1$ \ $T_2$ | S | X | IS | IX | SIX | – |
|---|---|---|---|---|---|---|
| S | Y | N | Y | N | N | Y |
| X | N | N | N | N | N | Y |
| IS | Y | N | Y | Y | Y | Y |
| IX | N | N | Y | Y | N | Y |
| SIX | N | N | Y | N | N | Y |
| – | Y | Y | Y | Y | Y | Y |

Y=Yes，表示相容的请求　　N=No，表示不相容的请求

(a) 数据锁的相容矩阵

# 意向锁（续）

## 数据锁的相容矩阵

| $T_1$ \ $T_2$ | S | X | IS | IX | SIX | – |
|---|---|---|---|---|---|---|
| S | Y | N | Y | N | N | Y |
| X | N | N | N | N | N | Y |
| IS | Y | N | Y | Y | Y | Y |
| IX | N | N | Y | Y | N | Y |
| SIX | N | N | Y | N | N | Y |
| – | Y | Y | Y | Y | Y | Y |

Y=Yes，表示相容的请求　　N=No，表示不相容的请求

# 意向锁（续）

## 数据锁的相容矩阵

| $T_1$ \ $T_2$ | S | X | IS | IX | SIX | – |
|---|---|---|---|---|---|---|
| S | Y | N | Y | N | N | Y |
| X | N | N | N | N | N | Y |
| IS | Y | N | Y | Y | Y | Y |
| IX | N | N | Y | Y | N | Y |
| SIX | N | N | Y | N | N | Y |
| – | Y | Y | Y | Y | Y | Y |

Y=Yes，表示相容的请求　　N=No，表示不相容的请求

# 意向锁（续）

## 数据锁的相容矩阵

| $T_1$ \ $T_2$ | S | X | IS | IX | SIX | – |
|---|---|---|---|---|---|---|
| S | Y | N | Y | N | N | Y |
| X | N | N | N | N | N | Y |
| IS | Y | N | Y | Y | Y | Y |
| IX | N | N | Y | Y | N | Y |
| SIX | N | N | Y | N | N | Y |
| – | Y | Y | Y | Y | Y | Y |

Y=Yes，表示相容的请求　　N=No，表示不相容的请求

(a) 数据锁的相容矩阵

# 意向锁（续）

## 数据锁的相容矩阵

| $T_1$ \ $T_2$ | S | X | IS | IX | SIX | – |
|---|---|---|---|---|---|---|
| S | Y | N | Y | N | N | Y |
| X | N | N | N | N | N | Y |
| IS | Y | N | Y | Y | Y | Y |
| IX | N | N | Y | Y | N | Y |
| SIX | N | N | Y | N | N | Y |
| – | Y | Y | Y | Y | Y | Y |

Y=Yes，表示相容的请求　　N=No，表示不相容的请求

(a) 数据锁的相容矩阵

# 意向锁（续）

## 数据锁的相容矩阵

| $T_1$ \ $T_2$ | S | X | IS | IX | SIX | – |
|---|---|---|---|---|---|---|
| S | Y | N | Y | N | N | Y |
| X | N | N | N | N | N | Y |
| IS | Y | N | Y | Y | Y | Y |
| IX | N | N | Y | Y | N | Y |
| SIX | N | N | Y | N | N | Y |
| – | Y | Y | Y | Y | Y | Y |

Y=Yes，表示相容的请求　　N=No，表示不相容的请求

(a) 数据锁的相容矩阵

# 意向锁（续）

## 数据锁的相容矩阵

| $T_1$ \ $T_2$ | S | X | IS | IX | SIX | – |
|---|---|---|---|---|---|---|
| S | Y | N | Y | N | N | Y |
| X | N | N | N | N | N | Y |
| IS | Y | N | Y | Y | Y | Y |
| IX | N | N | Y | Y | N | Y |
| SIX | N | N | Y | N | N | Y |
| – | Y | Y | Y | Y | Y | Y |

Y=Yes，表示相容的请求　　N=No，表示不相容的请求

(a) 数据锁的相容矩阵

# 意向锁（续）

## 数据锁的相容矩阵

| $T_1$ \ $T_2$ | S | X | IS | IX | SIX | – |
|---|---|---|---|---|---|---|
| S | Y | N | Y | N | N | Y |
| X | N | N | N | N | N | Y |
| IS | Y | N | Y | Y | Y | Y |
| IX | N | N | Y | Y | N | Y |
| SIX | N | N | Y | N | N | Y |
| – | Y | Y | Y | Y | Y | Y |

Y=Yes，表示相容的请求　　N=No，表示不相容的请求

(a) 数据锁的相容矩阵

# 意向锁（续）

## 数据锁的相容矩阵

| $T_1$ \ $T_2$ | S | X | IS | IX | SIX | – |
|---|---|---|---|---|---|---|
| S | Y | N | Y | N | N | Y |
| X | N | N | N | N | N | Y |
| IS | Y | N | Y | Y | Y | Y |
| IX | N | N | Y | Y | N | Y |
| SIX | N | N | Y | N | N | Y |
| – | Y | Y | Y | Y | Y | Y |

Y=Yes，表示相容的请求　　N=No，表示不相容的请求

(a) 数据锁的相容矩阵

# 意向锁（续）

## 数据锁的相容矩阵

| $T_1$ \ $T_2$ | S | X | IS | IX | SIX | – |
|---|---|---|---|---|---|---|
| S | Y | N | Y | N | N | Y |
| X | N | N | N | N | N | Y |
| IS | Y | N | Y | Y | Y | Y |
| IX | N | N | Y | Y | N | Y |
| SIX | N | N | Y | N | N | Y |
| – | Y | Y | Y | Y | Y | Y |

Y=Yes，表示相容的请求　　N=No，表示不相容的请求

(a) 数据锁的相容矩阵

# 意向锁（续）

## 数据锁的相容矩阵

| $T_1$ \ $T_2$ | S | X | IS | IX | SIX | – |
|---|---|---|---|---|---|---|
| S | Y | N | Y | N | N | Y |
| X | N | N | N | N | N | Y |
| IS | Y | N | Y | Y | Y | Y |
| IX | N | N | Y | Y | N | Y |
| SIX | N | N | Y | N | N | Y |
| – | Y | Y | Y | Y | Y | Y |

Y=Yes，表示相容的请求　　N=No，表示不相容的请求

(a) 数据锁的相容矩阵

# 意向锁（续）

## 数据锁的相容矩阵

| $T_1$ \ $T_2$ | S | X | IS | IX | SIX | – |
|---|---|---|---|---|---|---|
| S | Y | N | Y | N | N | Y |
| X | N | N | N | N | N | Y |
| IS | Y | N | Y | Y | Y | Y |
| IX | N | N | Y | Y | N | Y |
| SIX | N | N | Y | N | N | Y |
| – | Y | Y | Y | Y | Y | Y |

Y=Yes，表示相容的请求　　N=No，表示不相容的请求

(a) 数据锁的相容矩阵

# 意向锁（续）

## 数据锁的相容矩阵

| $T_1$ \ $T_2$ | S | X | IS | IX | SIX | – |
|---|---|---|---|---|---|---|
| S | Y | N | Y | N | N | Y |
| X | N | N | N | N | N | Y |
| IS | Y | N | Y | Y | Y | Y |
| IX | N | N | Y | Y | N | Y |
| SIX | N | N | Y | N | N | Y |
| – | Y | Y | Y | Y | Y | Y |

Y=Yes，表示相容的请求　　N=No，表示不相容的请求

(a) 数据锁的相容矩阵

# 意向锁（续）

## 数据锁的相容矩阵

| $T_1$ \ $T_2$ | S | X | IS | IX | SIX | – |
|---|---|---|---|---|---|---|
| S | Y | N | Y | N | N | Y |
| X | N | N | N | N | N | Y |
| IS | Y | N | Y | Y | Y | Y |
| IX | N | N | Y | Y | N | Y |
| SIX | N | N | Y | N | N | Y |
| – | Y | Y | Y | Y | Y | Y |

Y=Yes，表示相容的请求　　N=No，表示不相容的请求

(a) 数据锁的相容矩阵

# 意向锁（续）

## 数据锁的相容矩阵

| $T_1$ \ $T_2$ | S | X | IS | IX | SIX | – |
|---|---|---|---|---|---|---|
| S | Y | N | Y | N | N | Y |
| X | N | N | N | N | N | Y |
| IS | Y | N | Y | Y | Y | Y |
| IX | N | N | Y | Y | N | Y |
| SIX | N | N | Y | N | N | Y |
| – | Y | Y | Y | Y | Y | Y |

Y=Yes，表示相容的请求　　N=No，表示不相容的请求

(a) 数据锁的相容矩阵

# 意向锁（续）

## 数据锁的相容矩阵

| $T_1$ \ $T_2$ | S | X | IS | IX | SIX | – |
|---|---|---|---|---|---|---|
| S | Y | N | Y | N | N | Y |
| X | N | N | N | N | N | Y |
| IS | Y | N | Y | Y | Y | Y |
| IX | N | N | Y | Y | N | Y |
| SIX | N | N | Y | N | N | Y |
| – | Y | Y | Y | Y | Y | Y |

Y=Yes，表示相容的请求　　N=No，表示不相容的请求

(a) 数据锁的相容矩阵

# 意向锁（续）

## 数据锁的相容矩阵

| $T_1$ \ $T_2$ | S | X | IS | IX | SIX | – |
|---|---|---|---|---|---|---|
| S | Y | N | Y | N | N | Y |
| X | N | N | N | N | N | Y |
| IS | Y | N | Y | Y | Y | Y |
| IX | N | N | Y | Y | N | Y |
| SIX | N | N | Y | N | N | Y |
| – | Y | Y | Y | Y | Y | Y |

Y=Yes，表示相容的请求　　N=No，表示不相容的请求

(a) 数据锁的相容矩阵

# 意向锁（续）

## 数据锁的相容矩阵

| $T_1$ \ $T_2$ | S | X | IS | IX | SIX | – |
|---|---|---|---|---|---|---|
| S | Y | N | Y | N | N | Y |
| X | N | N | N | N | N | Y |
| IS | Y | N | Y | Y | Y | Y |
| IX | N | N | Y | Y | N | Y |
| SIX | N | N | Y | N | N | Y |
| – | Y | Y | Y | Y | Y | Y |

Y=Yes，表示相容的请求　　N=No，表示不相容的请求

(a) 数据锁的相容矩阵

# 意向锁（续）

## 数据锁的相容矩阵

| $T_1$ \ $T_2$ | S | X | IS | IX | SIX | – |
|---|---|---|---|---|---|---|
| S | Y | N | Y | N | N | Y |
| X | N | N | N | N | N | Y |
| IS | Y | N | Y | Y | Y | Y |
| IX | N | N | Y | Y | N | Y |
| SIX | N | N | Y | N | N | Y |
| – | Y | Y | Y | Y | Y | Y |

Y=Yes，表示相容的请求　　N=No，表示不相容的请求

(a) 数据锁的相容矩阵

# 意向锁（续）

## 数据锁的相容矩阵

| $T_1$ \ $T_2$ | S | X | IS | IX | SIX | – |
|---|---|---|---|---|---|---|
| S | Y | N | Y | N | N | Y |
| X | N | N | N | N | N | Y |
| IS | Y | N | Y | Y | Y | Y |
| IX | N | N | Y | Y | N | Y |
| SIX | N | N | Y | N | N | Y |
| – | Y | Y | Y | Y | Y | Y |

Y=Yes，表示相容的请求　　N=No，表示不相容的请求

(a) 数据锁的相容矩阵

# 意向锁（续）

## 数据锁的相容矩阵

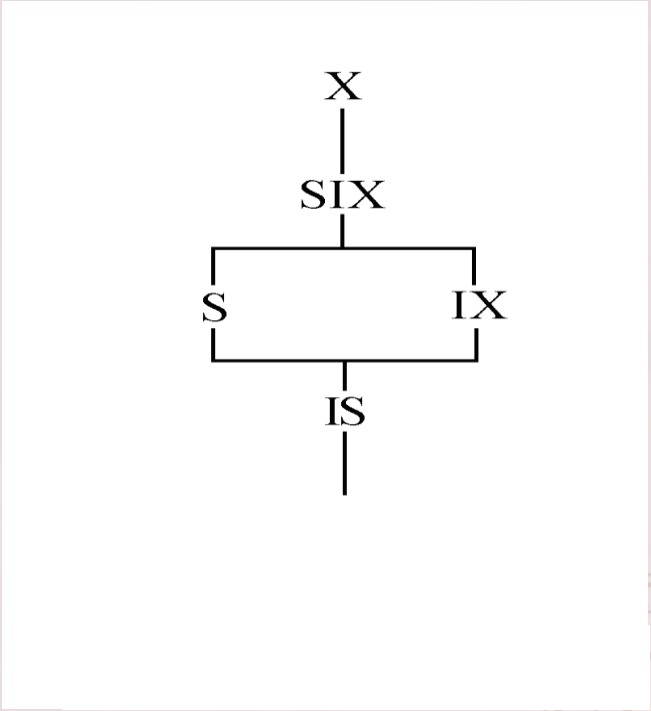
| $T_1$ \ $T_2$ | S | X | IS | IX | SIX | – |
|---|---|---|---|---|---|---|
| S | Y | N | Y | N | N | Y |
| X | N | N | N | N | N | Y |
| IS | Y | N | Y | Y | Y | Y |
| IX | N | N | Y | Y | N | Y |
| SIX | N | N | Y | N | N | Y |
| – | Y | Y | Y | Y | Y | Y |

Y=Yes，表示相容的请求　　N=No，表示不相容的请求

(a) 数据锁的相容矩阵

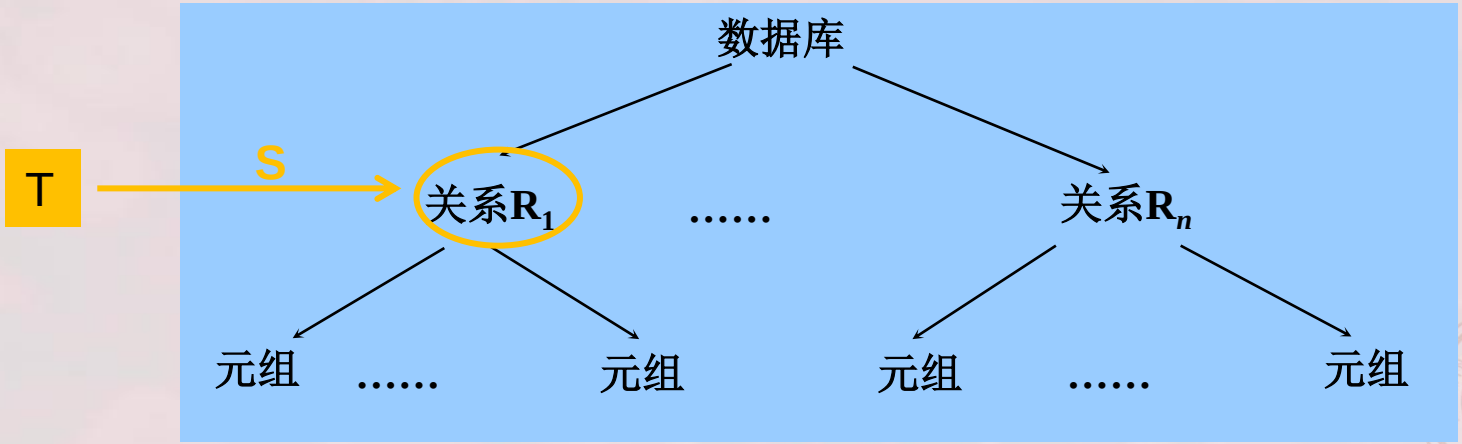
# 意向锁（续）

- 锁的强度
  - 锁的强度是指它对其他锁的排斥程度
  - 一个事务在申请封锁时以强锁代替弱锁是安全的，反之则不然



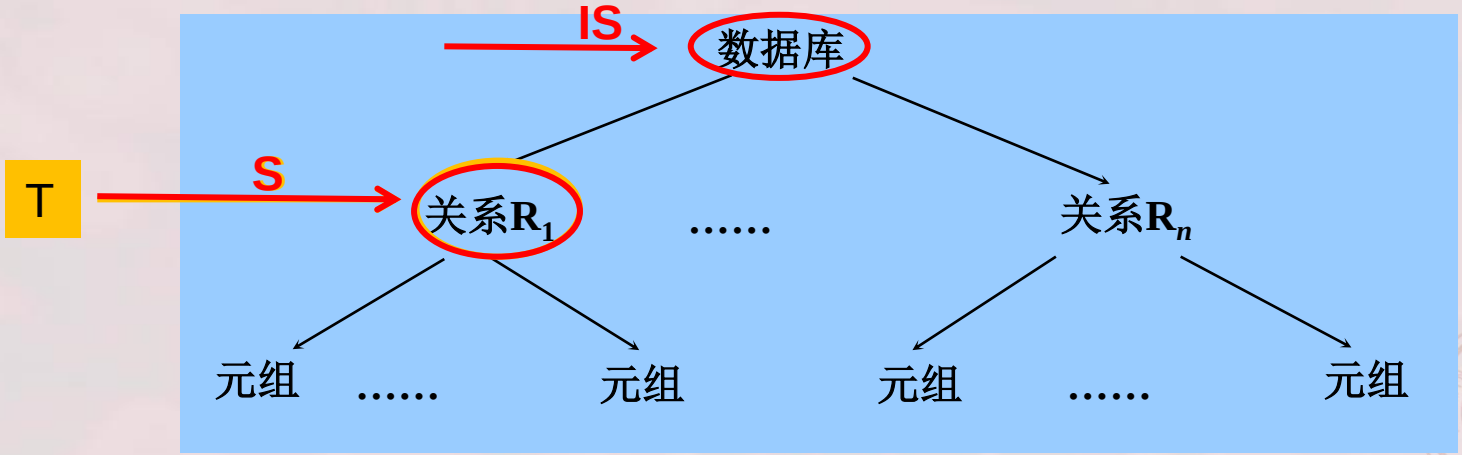
# 意向锁（续）

- 具有意向锁的多粒度封锁方法
  - 申请封锁时应该按自上而下的次序进行
  - 释放封锁时则应该按自下而上的次序进行



# 意向锁（续）

- 具有意向锁的多粒度封锁方法
  - 申请封锁时应该按自上而下的次序进行
  - 释放封锁时则应该按自下而上的次序进行



# 意向锁（续）

- 具有意向锁的多粒度封锁方法
  - 提高了系统的并发度
  - 减少了加锁和解锁的开销
  - 在实际的数据库管理系统产品中得到广泛应用

# 小结

- 封锁粒度
  - 封锁粒度与系统的并发度和并发控制的开销的关系
- 多粒度封锁
  - 什么是多粒度封锁
  - 多粒度封锁方法
- 意向锁
  - 三种意向锁
  - 相容矩阵

# 思考题

- 为什么说具有意向锁的多粒度封锁方法可以提高了系统的并发度，减少了加锁和解锁的开销？

# 第十一章 并发控制

# 11.9 小结

- 并发操作带来的数据不一致性

  1.丢失修改（Lost Update）

  2.不可重复读（Non-repeatable Read）

  3.读“脏”数据（Dirty Read）

# 小结（续）

- 数据库的并发控制通常使用封锁机制
  - 基本封锁（X锁和S锁）
  - 多粒度封锁（意向锁）
- 活锁和死锁
  - 活锁
  - 死锁（预防，检测）
- 解决数据不一致的并发控制协议：三级封锁协议

# 小结（续）

- 并发事务调度的正确性
  - 可串行性
    - 并发操作的正确性则通常由两段锁协议来保证。
    - 两段锁协议是可串行化调度的充分条件，但不是必要条件
  - 冲突可串行性

# 小结（续）

- 本章目标
  - 了解数据库并发控制技术的必要性，掌握并发控制的相关技术。
- 本章重点
  - 牢固掌握并发操作可能产生数据不一致性的情况；基本封锁和多粒度封锁方法，相关的相容控制矩阵；
  - 举一反三：封锁协议与数据一致性的关系；并发调度的可串行性概念。
- 本章难点
  - 两段锁协议与串行性的关系、与死锁的关系。具有意向锁的多粒度封锁方法的封锁过程。