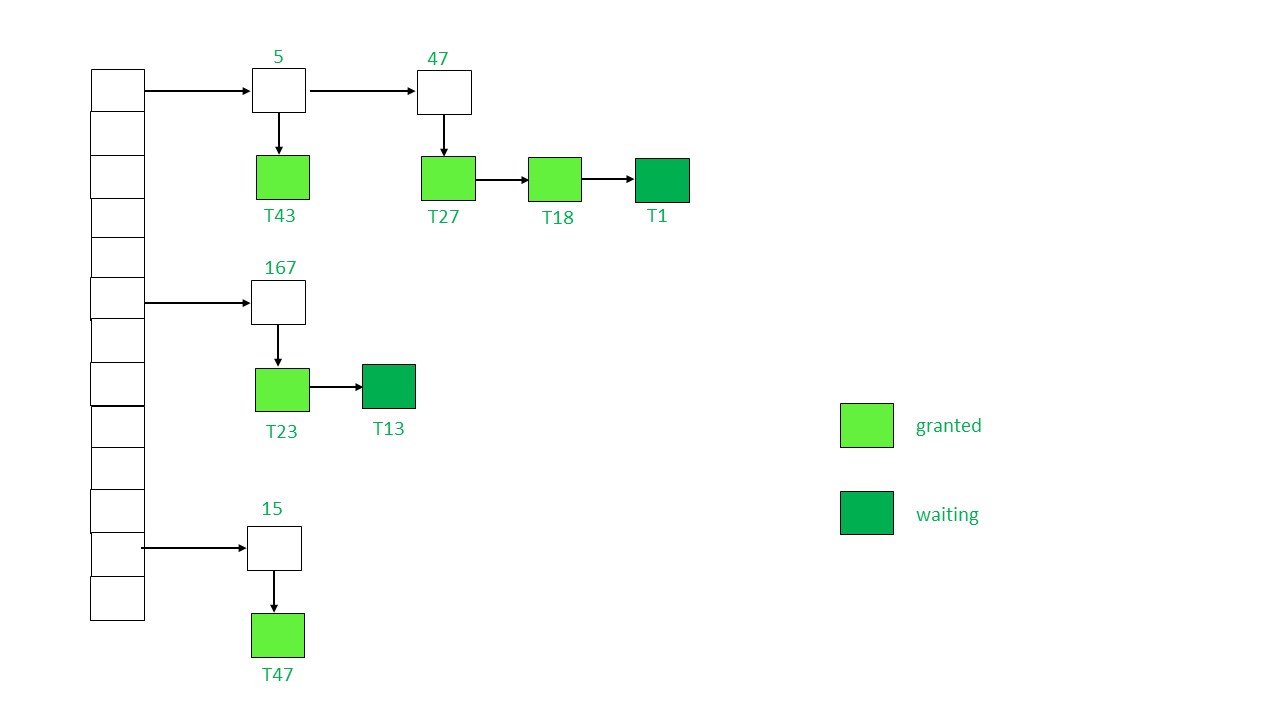
DBMS中锁定的实现

锁定协议在数据库管理系统中用作并发控制的手段。多个事务可能会同时请求锁定数据项。因此，我们需要一种机制来管理事务发出的锁定请求。这种机制称为**锁管理器**。它依赖于消息传递的过程，其中事务和锁管理器交换消息来处理数据项的锁定和解锁。

**锁管理器中使用**  
的数据结构**–**实施锁所需的数据结构称为“ **锁表”**。

1. 它是一个哈希表，其中数据项的名称用作哈希索引。
2. 每个锁定的数据项都有一个与其关联的链表。
3. 链表中的每个节点都代表请求锁定的事务，请求的锁定模式（相互/排他）和请求的当前状态（已授予/正在等待）。
4. 对数据项的每个新锁定请求都将作为新节点添加到链接列表的末尾。
5. 哈希表中的冲突通过单独的链接技术处理。

考虑以下锁表示例：



**说明：**在上图中，锁定表中存在的锁定数据项为5、47、167和15。

已请求锁定的交易已使用向下箭头显示在其下方的链接列表中。

链表中的每个节点都有交易的名称，该交易的名称已请求了数据项，例如T33，T1，T27等。

节点的颜色表示状态，即是否已授予锁定或正在等待锁定。

请注意，数据项5和47发生了冲突。已经通过单独的链接解决了冲突，其中每个数据项都属于一个链表。数据项充当包含锁定请求的链表的标题。

**锁管理器的工作–**

1. 最初，锁定表为空表，因为没有数据项被锁定。
2. 每当锁管理器从事务T i收到特定数据项Q i上的锁请求时， 可能会发生以下情况：
   * 如果尚未锁定Q i，则将创建一个链表，并将锁定授予请求事务T i。
   * 如果数据项已被锁定，则将在其链接列表的末尾添加一个新节点，其中包含有关T i发出的请求的信息。
3. 如果T i请求的锁定模式与当前具有该锁定的交易的锁定模式兼容，则T i也将获得该锁定，并且状态将更改为“已授予”。否则，T i锁的状态将为“正在等待”。
4. 如果事务T i要解锁其当前持有的数据项，它将向锁管理器发送解锁请求。锁定管理员将从此链接列表中删除T i的节点。锁定将授予列表中的下一个事务。
5. 有时可能必须中止事务T i。在这种情况下，将由T i发出的所有等待请求都将从锁定表中存在的链接列表中删除。一旦流产完成，由T i持有的锁也将被释放。