**Select加锁查询语句中，WEHRE条件是非聚簇索引的unique索引时，无论是等值查询还是范围查询，结论和WHERE条件是聚簇索引的时候是一样的，不同的是，不仅会在相应的unique索引树上给相应的索引值上锁，而且会到相应的聚簇索引树上给相应的聚簇索引值上锁，并相当于锁住数据页上的相应的位置！**

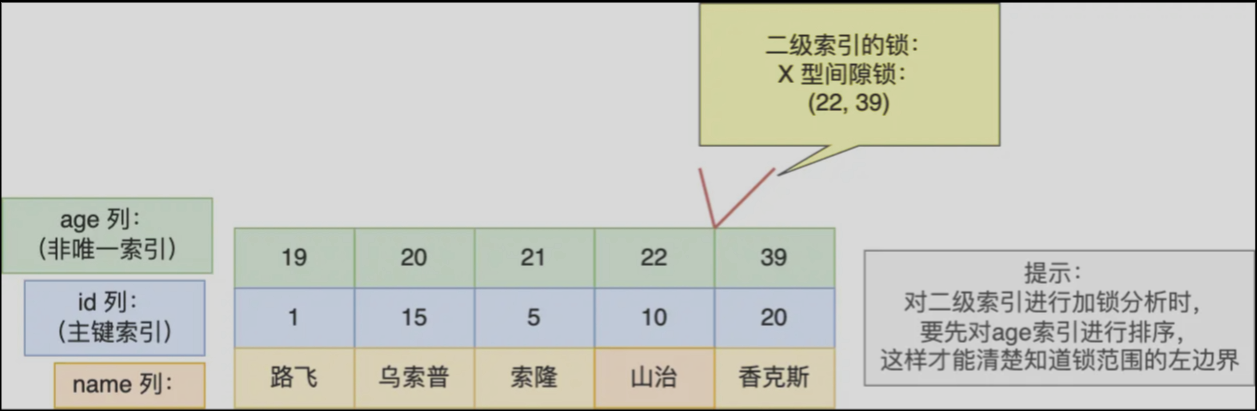
**Select加锁查询语句中，WEHRE条件是非聚簇索引非unique索引，也即一般索引等值查询**

start transaction;

select \* from user where age=25 for update;

该事务a提交前：

事务 A 中该sql在age索引树上按序扫描，在 age = 39 这个一般索引值上加锁，锁住的范围是 (22, 39)，因为要避免其他事务插入age=25的一行，而无法避免伤及无辜 地(22, 39)都锁住。{由锁住的范围可知是一个Gap Lock间隙锁，由原本加的Next-Key Lock 临键锁退化而来}



如果其他事务要插入 age 值为 23、24...、37、38 这一些新的行，则会被阻塞。

需要知道的是，非聚簇索引的age索引树中，是age列的值按顺序存放的，在age值相等的情况下（age索引并非unique索引），对应的聚簇索引id的值按序存放。

因此，其他事务要插入age=22,id<10的行不会被阻塞，

要插入age=22,id>10的行是会被阻塞的（若update别的行的age，age索引树随之更新后，此时若被改后的age按序应该放到这里被锁的范围来，那么和插入一样也被阻塞，已验证）；

其他事务要插入age=39,id<20的行是会被阻塞的（若update别的行的age，age索引树随之更新后，此时若被改后的age按序应该放到这里被锁的范围来，那么和插入一样也被阻塞，已验证），

要插入age=39,id>20的行不会被阻塞。

start transaction;

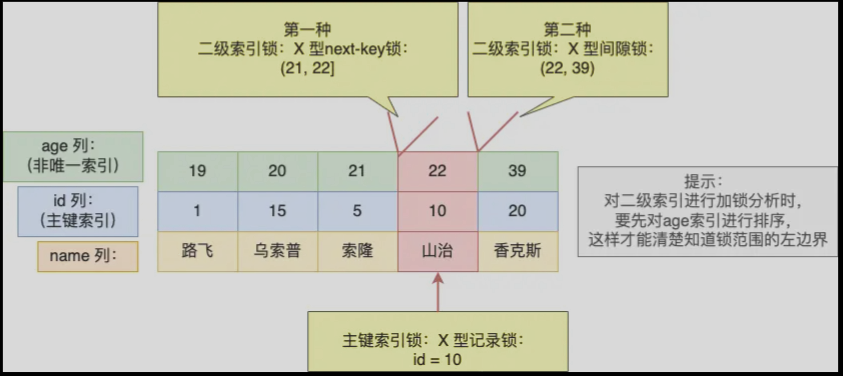
select \* from user where age=22 for update;

该事务a提交前：

事务 A 中该sql中该sql按序扫描age索引树，在 age = 22 这个一般索引值上加锁，锁住的范围是 (21,22]，因为age不是unique，是可以重复的，要避免别的事务插入age=22的行，而无法避免伤及无辜地 (21,22]都锁住。{由锁住的范围可知是一个临键锁，就是原本加的Next-Key Lock 临键锁，没退化}

由于 age =22 是存在的，因此也要对age索引树中age=22这个一般索引值对应的聚簇索引树上的id=10这个索引值加上锁，===这就相当于锁住了数据页上id=10这一行。

事务A 继续按序扫描age索引树（因为age不是unique，是可以重复的，要避免别的事务插入age=22的行，所以要继续扫描，并无可避免地可能伤及无辜），在 age = 39 这个一般索引值上加锁，锁住的范围是 (22,39)。{由锁住的范围可知是一个Gap Lock间隙锁，由原本加的Next-Key Lock 临键锁退化而来}



聚簇索引：

如果其他事务要更新或删除id=10的行，则会被阻塞。

一般索引：

如果其他事务要更新或删除age=22的所有行，则都会被阻塞。

要插入age为23、24、..... 、37、38的行，也会被阻塞（若update别的行的age，age索引树随之更新后，此时若被改后的age按序应该放到这里被锁的范围来，那么和插入一样也被阻塞，已验证）。

如果其他事务要插入age=21,id<5的行，则不会被阻塞；若要插入age=21,id>5的行，会被阻塞（伤及无辜）（若update别的行的age，age索引树随之更新后，此时若被改后的age按序应该放到这里被锁的范围来，那么和插入一样也被阻塞，已验证）；

如果其他事务要插入age=39,id<20的行，会被阻塞（伤及无辜）（若update别的行的age，age索引树随之更新后，此时若被改后的age按序应该放到这里被锁的范围来，那么和插入一样也被阻塞，已验证）；

若要插入age=39,id>20的行，则不会被阻塞。

**Select加锁查询语句中，WEHRE条件是非聚簇索引非unique索引，也即一般索引范围查询**

start transaction;

select \* from user where age>=22 for update;

该事务a提交前：

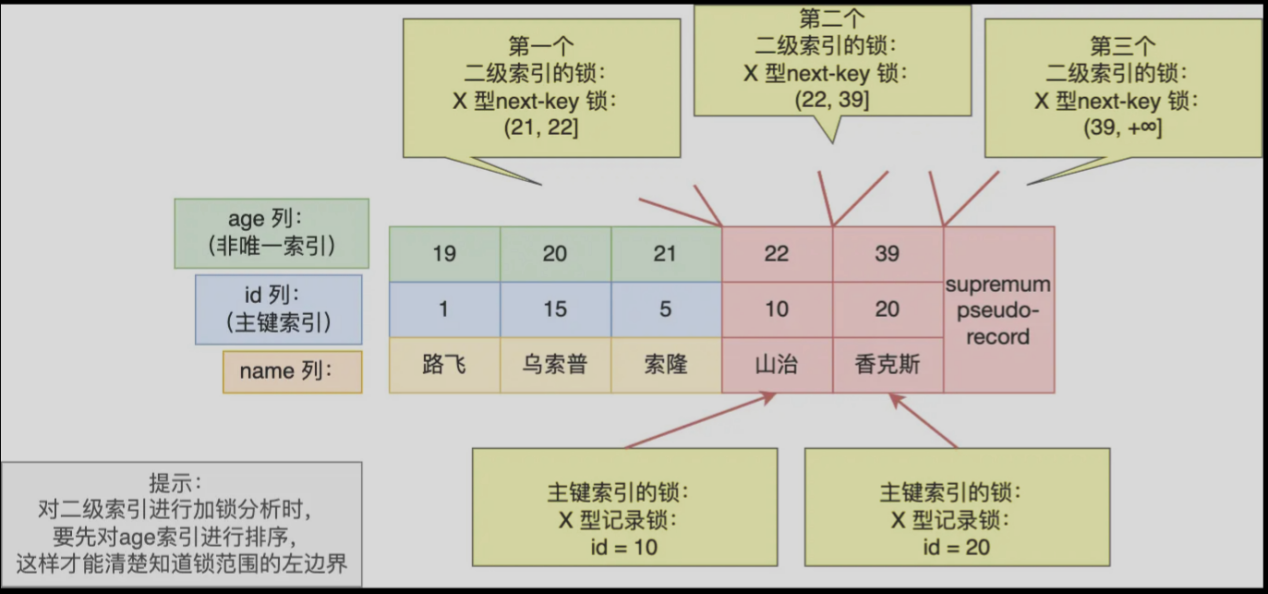
事务A按序扫描age索引树（因为age索引不是unique索引，age值可以重复，要避免别的事务插入age=22的行，而无法避免伤及无辜），在 age = 22 这个一般索引上加锁，锁住的范围是 (21,22]。{由锁住的范围可知是一个临键锁，就是原本加的Next-Key Lock 临键锁，没退化}

由于 age =22 是存在的，因此也要对age索引树中age=22这个一般索引值对应的聚簇索引树上的id=10这个索引值加上锁。===这就相当于数据页上id=10这一行被锁。

事务A 继续按序扫描age索引树，在 age = 39 这个一般索引值上加锁，锁住的范围是 (22,39]。{由锁住的范围可知是一个临键锁，就是原本加的Next-Key Lock 临键锁，没退化}

由于 age =39 是存在的，因此也要对age索引树中age=39这个一般索引值对应的聚簇索引上的id=20这个索引值加上锁。===这就相当于数据页上id=20这一行被锁。

事务 A 中该sql中该sql继续按序扫描age索引树，在 age =supremum pseudo-record 这个索引值上加锁，锁住的范围是(39, +∞]。{由锁住的范围可知是一个临键锁，就是原本加的Next-Key Lock 临键锁，没退化}



聚簇索引：

如果其他事务要更新或删除id=10、id=20的行，则会被阻塞。

一般索引：

如果其他事务要更新或删除age=22、age=39的所有行，则都会被阻塞。

如果其他事务要新增age=23、24、...、37、38的行，则会被阻塞（若update别的行的age，age索引树随之更新后，此时若被改后的age按序应该放到这里被锁的范围来，那么和插入一样也被阻塞，已验证）。

如果其他事务要新增age=39、... 的行，则会被阻塞（若update别的行的age，age索引树随之更新后，此时若被改后的age按序应该放到这里被锁的范围来，那么和插入一样也被阻塞，已验证）。

如果其他事务要插入age=21,id<5的行，则不会被阻塞；若要插入age=21,id>5的行，会被阻塞（伤及无辜）（若update别的行的age，age索引树随之更新后，此时若被改后的age按序应该放到这里被锁的范围来，那么和插入一样也被阻塞，已验证）；

（完）