

:years: 2015-2022

Loan account locking

Keeping a consistent state of loan accounts become quite important when we start talking about doing a business day closing each day for loans.

There are 2 concepts for loan account locking:

1. Soft-locking loan accounts
2. Hard-locking loan accounts

Soft-locking simply means that when the Loan COB has been kicked off but workers not yet processing the chunk of loan accounts (i.e. the partition is waiting in the queue to be picked up) and during this time a real-time write request (e.g. a repayment/disbursement) comes in through the API, we simply do an "inlined" version of the Loan COB for that loan account. From a practical standpoint this will mean that before doing the actual repayment/disbursement on the loan account on the API, we execute the Loan COB for that loan account, kind of like prioritizing it.

Hard-locking means that when a worker picks up the loan account in the chunk, real-time write requests on those loan accounts will be simply rejected with an [HTTP 409](#).

The locking is strictly tied to the Loan COB job's execution but there could be other processes in the future which might want to introduce new type of locks for loans.

The loan account locking is solved by maintaining a database table which stores the locked accounts, it's called [m_loan_account_locks](#).

When a loan account is present in the table above, it simply means there's a lock applied to it and whether it's a soft or hard lock can be determined by the [lock_owner](#) column.

And when a loan account is locked, loan related write API calls will be either rejected or will trigger an inline Loan COB execution. There could be a corner case here when the Loan COB fails to process some loan accounts (due to a bug, inconsistency, etc) and the loan accounts stay locked. This is an intended behavior to mark loans which are not supposed to be used until they are "fixed".

Since the fixing might involve making changes to the loan account via API (for example doing a repayment to fix the loan account's inconsistent state), we need to allow those API calls. Hence, the lock table includes a [bypass_enabled](#) column which disables the lock checks on the loan write APIs.