Late write - Late Conflict Insert

- Generate Row ID

FEC: *Part of engine*

    - Write To Transactor (Txactor.Apply)

∿ other trx - read from buff, apply txactor ∿

- Pre detect to optimize time in critical section
- Pre write to log to optimize time in CS?
- Prepare (TM.Prepare → rev)

could be distributed (against current log)

FEC:

    - Detect Conflicts (revisor.detect (revisor.changes) → [error])
FEC: - Start persist (store.log (revisor.changes, affected colIDs)
FEC:

T! [
    - Begin Commit (Buff.Begin)
    - Write change (Buff.Apply)   } *Could optimize*
    - End commit (Buff.End(Rev))   } *to apply if small*
]
- Commit Transaction (TM.Commit)
- Join persist

Column Locker
    - Holds pending {TID, RowID}s

Column Transactor
    - Stores Trxs Changes
    - Commits
    - Redoes

Insert 1 row   - Immediate write + late conflict

- Generate Row ID (Generator. New Row → Row ID)

FEC:
    - Write to Transactor (Txactor.Apply → brev)

    - Write to buffer (Buff. Apply (brev))

∿ other trx ops ∿

- Prepare   (TM.Prepare + rev)

FEC:
    - Detect Conflicts   (Txactor.Changes → revisor → [error])

- Commit   (TM.commit)
FEC: - Storage Commit (Store. commit (affected colIDs))
FEC:
    - Commit (Buff.Commit(rev))

- Return if picky