But log-structured tables make locking hard...

Because <u>any new row added can change the total</u>,

we'd have to lock the entire BalanceTransfer table to prevent concurrent processes from adding new transfers that change the total.

How else can we prevent concurrent writes from changing a user's balance?

Store a total separately from the log, require they be updated together

```
class UserBalance(models.model):
    user = models.OneToOneField(User)
    total = models.DecimalField(max_digits=20, decimal_places=2)
```

A single-row lock must now be obtained on the **total** before adding new BalanceTransfer rows for that user.