**Transactions**

**Les verrous dans SQLite**

Il y a 5 états de verrous, pour chaque état il y a un verrou correspondant (sauf **UNLOCKED** qui n’a pas besoin de verrou) :

**UNLOCKED** : Aucune session n’effectue de traitement sur les données de la base.  
**SHARED** : dans cet état, plusieurs sessions peuvent lire les données simultanément, mais pas les écrire.  
**RESERVED** : Une session à la fois maintiens le droit d’écrire sur la base de données, cet état peut coexister avec d’autres états **SHARED**. Les modifications faites sont mises en cache et ne sont pas réellement écrites sur le disque.  
Exclusive : Lorsque la session souhaite valider les modifications (ou transactions).  
**PENDING** : Signifie que le processus qui détient le verrou veut écrire dans la base de données dès que possible et attend que tous les verrous **SHARED** en cours soient effacés pour pouvoir obtenir un verrou **EXCLUSIVE**.

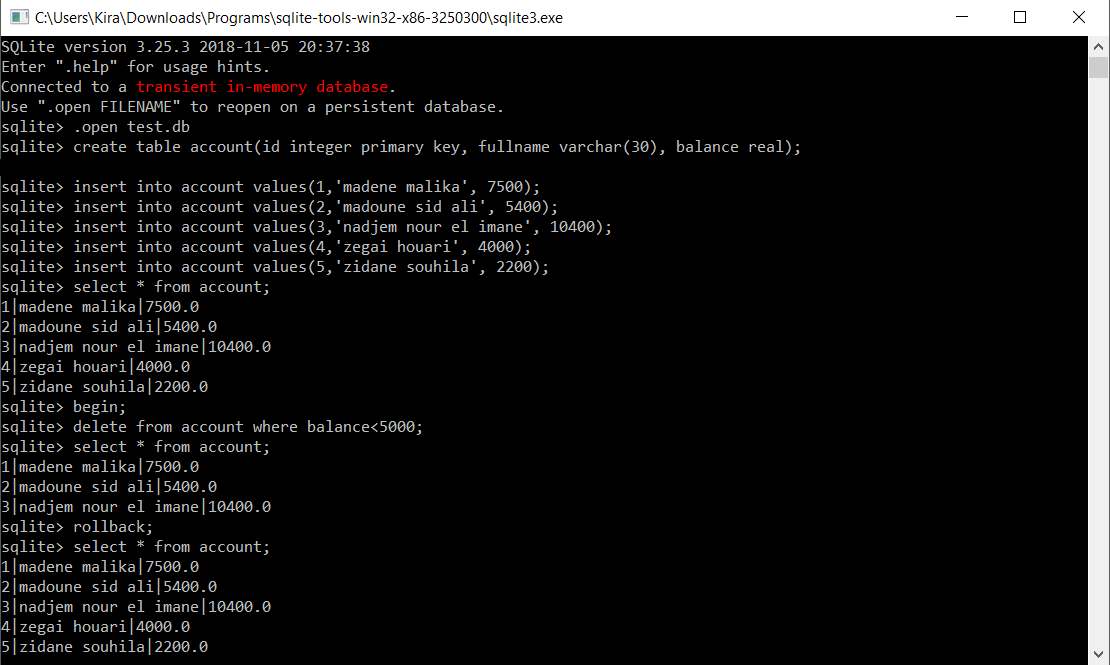
**Les types de transactions**

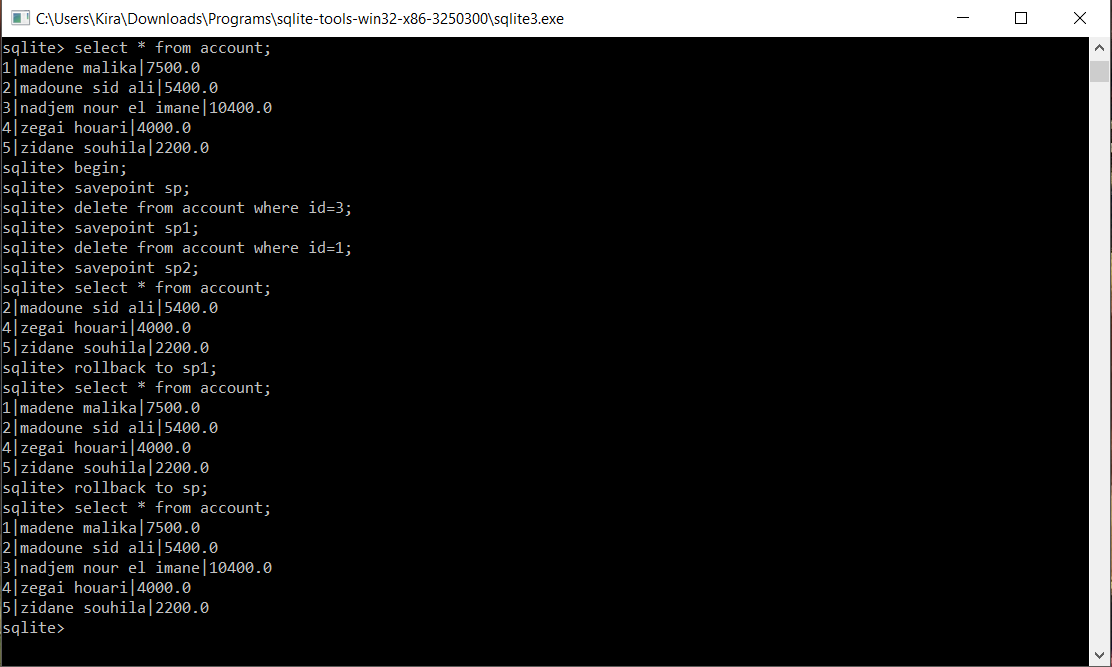
Par défaut une transaction est **DEFERRED** (n’acquiert aucun verrou jusqu'à ce que cela soit nécessaire).  
Une transaction **IMMEDIATE** tente d'obtenir un verrou **RESERVED** dès que la commande **BEGIN** est exécutée.  
Une transaction **EXCLUSIVE** obtient un verrou **EXCLUSIVE** sur la base de données. Elle garantit qu'aucune autre session n'est active dans la base de données et que vous pouvez lire ou écrire en toute impunité.

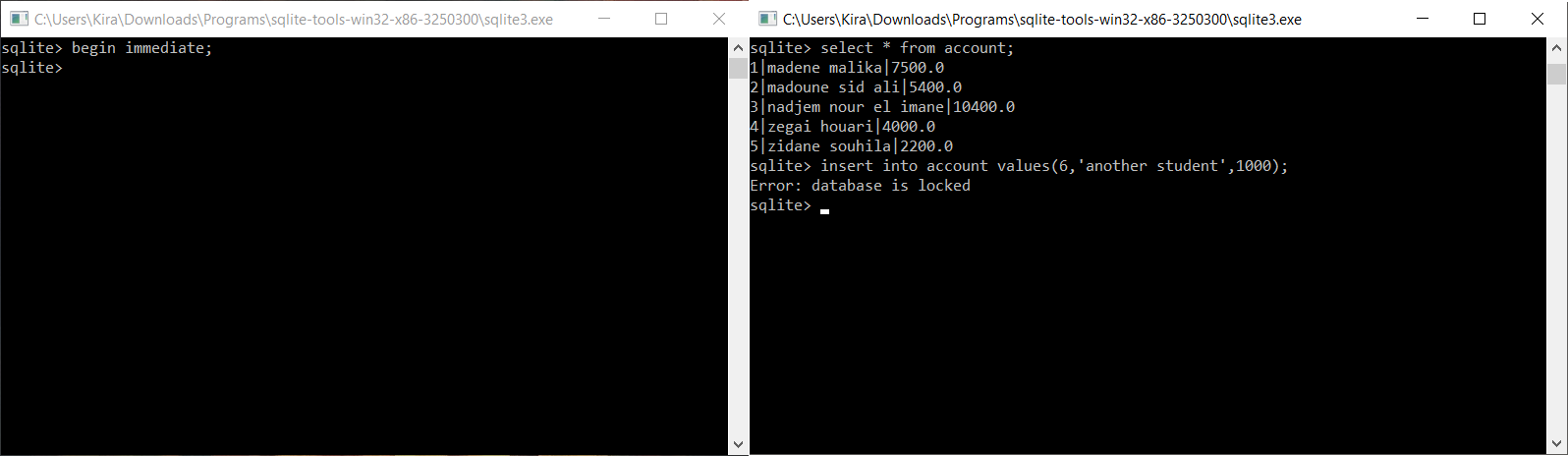
Syntaxe: **BEGIN [ DEFERRED | IMMEDIATE | EXCLUSIVE ];**

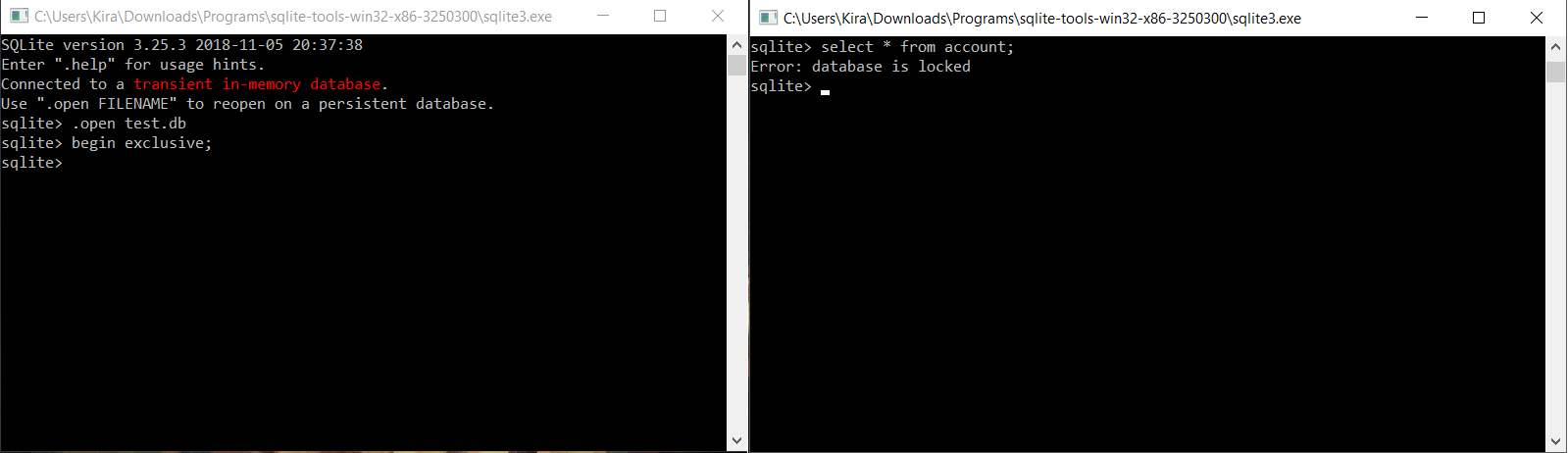
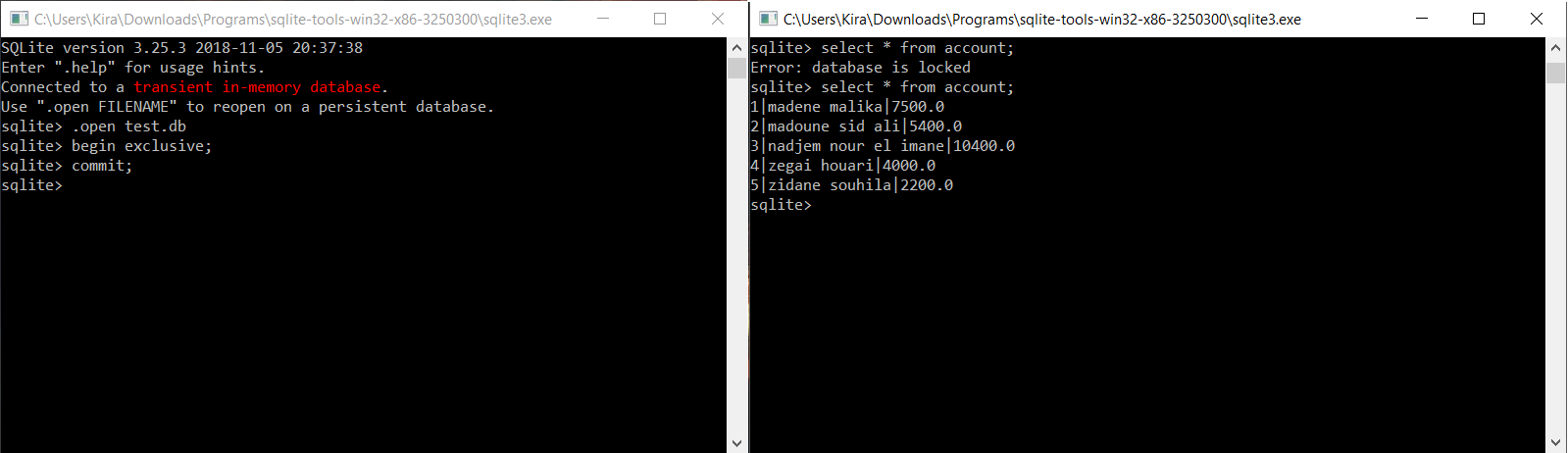
La libération se fait avec un **COMMIT** ou un **ROLLBACK** comme nous allons voir dans la partie des exemples.

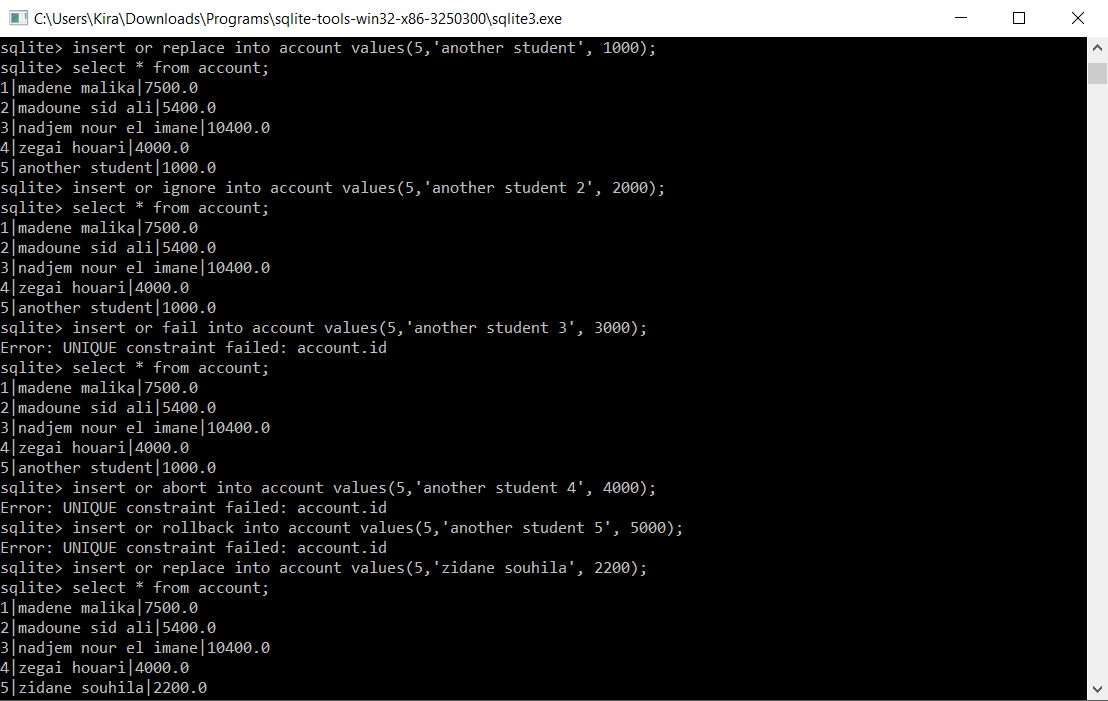
**Les examples**

BEGIN et ROLLBACK  


SAVEPOINT et ROLLBACK TO  


BEGIN IMMEDIATE  


BEGIN EXCLUSIVE  
  
COMMIT  


REPLACE, IGNORE, FAIL, ABORT  


Journalisation avec TRIGGERS  
