Investigate Database Recovery

```
# Import RSQLite library
library(RSQLite)

# Load DBI library for functions (dbConnect, dbListTables, dbWriteTable)
library(DBI)

dbfile = "example.sqlite"

#create dbcon variable
dbconLite <- dbConnect(RSQLite::SQLite(), dbfile)
```

Choose a database recovery problem or scenario (perhaps from work) and then propose a solution using the techniques described in Chapter 11 in the textbook. Briefly describe the technique, when it is appropriate to use and what recovery problem it solves.

A classic example of a bank transfer between two bank accounts from the same database. One account is debited and the other is credited with funds. Assume that an error occurred in between the transfer and database recovery is necessary.

One solution is to use log-based recovery.(Date, C. J., et al. Introduction to Database Systems. Pearson Education, 2006.).The database keeps logs of all transactions and when an error is detected. The database can revert back to a previous state prior to the failed transaction. For example if Person A had $1000 and was to pay Person B $500. If the transaction failed, the database can just reflect the same state it is in now, where Person A will still have $1000 in his/her account and the payment to Person B will not have been processed.

```
/* Enable foreign key constraints */

PRAGMA foreign_keys = ON
```

```
/* Create Author Table */
CREATE TABLE Author (
  aid NUMERIC NOT NULL,
  name TEXT NOT NULL,
  email TEXT NOT NULL,
  affiliation TEXT CHECK(affiliation IN ('PT','FT','Contract','Associate')),
  PRIMARY KEY (aid)
)
```

```
/* Create Modules Table */
CREATE TABLE Module (
  number TEXT NOT NULL,
  title TEXT NOT NULL,
  lengthInMin NUMERIC NOT NULL,
  PRIMARY KEY (number),
  FOREIGN KEY (number) REFERENCES Author(aid)
)
```

```sql
/* Insert into Author table. */

INSERT INTO Author (aid,name,email,affiliation) VALUES
  (1,"Jose","Jose@northeastern.edu","PT"),
  (2,"Dan","Dan@northeastern.edu","Contract"),
  (3,"Martin","Martin@northeastern.edu","FT"),
  (4,"Kathleen","Kathleen@northeastern.edu","Associate");
```

```sql
/* Insert into Modules table. */

INSERT INTO Module (number,title,lengthInMin) VALUES
  ("1","Introduction",20),
  ("2","Test 1", 30),
  ("3","Test 2", 120),
  ("4","Test 3", 360);
```

```sql
/* Using Transactions, update the Author and Module Table concurrently to prevent errors */

BEGIN TRANSACTION;

UPDATE Module
   SET title = "New Intro"
   WHERE number = "1";

UPDATE Author
   SET name = "Vincent", email = "Vincent@northeastern.edu"
   WHERE aid = 1;

COMMIT TRANSACTION;
```

```sql
/* Verify that transaction was completed in Authors table. */
SELECT * FROM Author;
```

Table 1: 4 records

aid	name	email	affiliation
1	Jose	Jose@northeastern.edu	PT
2	Dan	Dan@northeastern.edu	Contract
3	Martin	Martin@northeastern.edu	FT
4	Kathleen	Kathleen@northeastern.edu	Associate

```sql
/* Verify that transaction was completed  in Modules table. */
SELECT * FROM Module;
```

Table 2: 4 records

number	title	lengthInMin
1	Introduction	20
2	Test 1	30
3	Test 2	120
4	Test 3	360

```
#Disconnect From Database
dbDisconnect(dbconLite)
```