```
CREATE TABLE "Account" (
	"Id"	INTEGER NOT NULL,
	"New_amount"	INTEGER NOT NULL,
	"Old_amount"	INTEGER NOT NULL,
	"Account_holder"	TEXT NOT NULL,
	CONSTRAINT "PK_Trip" PRIMARY KEY("Id"),
	FOREIGN KEY("Account_holder") REFERENCES "User"("User_id")
);
CREATE TABLE "Transaction_log" (
	"Id"	INTEGER NOT NULL,
	"Date"	DATE,
	"Transaction_Amount"	TEXT NOT NULL,
	"Account_affected"	TEXT NOT NULL,
	"Overdraft"	TEXT,
	CONSTRAINT "PK_Trip" PRIMARY KEY("Id" AUTOINCREMENT),
	FOREIGN KEY("Account_affected") REFERENCES "Account"("Id")
);
CREATE TABLE "user" (
	"User_id"	INTEGER NOT NULL,
	"Drivers_License"	TEXT NOT NULL,
	"First_name"	TEXT,
	"Last_name"	TEXT,
	"DOB"	TEXT,
	"Street"	TEXT,
	"City"	TEXT,
	"State"	TEXT,
	"Zip"	INTEGER NOT NULL,
	PRIMARY KEY("User_id" AUTOINCREMENT)
);
CREATE TRIGGER Transaction_logger AFTER UPDATE ON Account
WHEN NEW.New_amount > NEW.Old_amount
BEGIN
	INSERT INTO Transaction_log(Date, Transaction_Amount, Account_affected, Overdraft)
	VALUES (datetime('now','localtime'), (NEW.New_amount - NEW.Old_amount), NEW.Id,
'False') ;
END

CREATE TRIGGER Transaction_logger2 AFTER UPDATE ON Account
WHEN NEW.New_amount < NEW.Old_amount AND NEW.New_amount >= 0
BEGIN
	INSERT INTO Transaction_log(Date, Transaction_Amount, Account_affected, Overdraft)
	VALUES (datetime('now','localtime'), -(NEW.Old_amount - NEW.New_amount), NEW.Id,
'False') ;
END
```

```
CREATE TRIGGER Transaction_logger3 AFTER UPDATE ON Account
WHEN NEW.New_amount < NEW.Old_amount AND NEW.New_amount < 0
BEGIN
        INSERT INTO Transaction_log(Date, Transaction_Amount, Account_affected, Overdraft)
        VALUES (datetime('now','localtime'), -(NEW.Old_amount - NEW.New_amount), NEW.Id,
'True') ;
END
```

All things with an @ are variables that can be found in the C# code

```
INSERT INTO user (Drivers_License, First_Name, Last_Name, DOB, Street, City, State, Zip)
VALUES (@dl, @fn, @ln, @dob, @add, @ci, @st, @zip )
```

```
SELECT user.First_name, user.Last_name, Account.New_amount FROM Account INNER JOIN
user ON Account.Account_holder = user.user_id
```

```
INSERT INTO Account (New_amount, Old_amount, Account_holder) VALUES (@na, @oa,
@ah )
```

```
UPDATE Account SET Old_amount = New_amount WHERE Id = @id
```

```
SELECT * FROM Transaction_log
```

```
UPDATE Account SET Old_amount = New_amount WHERE Id = @id
```

```
UPDATE Account SET New_amount = New_amount - @amount WHERE Id = @id
```

```
UPDATE Account SET Old_amount = New_amount WHERE Id = @id (used twice to transfer
between accounts)
```

```
UPDATE Account SET New_amount = New_amount - @amount WHERE Id = @id
```

```
UPDATE Account SET New_amount = New_amount + @amount WHERE Id = @id
```

```
SELECT user.First_name, user.Last_name, Account.New_amount FROM Account INNER JOIN
user ON Account.Account_holder = user.user_id
```

```
SELECT * FROM Transaction_log WHERE date BETWEEN datetime('now', '-30 days') AND
datetime('now', 'localtime')
```

```
SELECT * FROM user
```