Transactions

In modern databases, transactions also do some other things, such as ensuring that you can’t access data that another person has written halfway. However, the basic idea is the same—transactions are there to ensure that no matter what happens, the data you work with will be in a sensible state. They guarantee that there will not be a situation where money is withdrawn from one account, but not deposited to another.

Yii2 supports a powerful transaction mechanism with savepoints.

A classic example is of transferring money from one bank account to another. To do that, you have to first withdraw the amount from the source account, and then deposit it to the destination account. The operation has to succeed in full. If you stop halfway, the money will be lost, and that is very bad. For instance, we have a recipient account and a sender account. We would like to transfer money from sender to recipient. Let’s assume that we have an account model.

Getting ready...

Our account model will be very simple and it will contain only the id and balance fields.

1. Create a new application using the Composer package manager, as described in the official guide at <http://www.yiiframework.com/doc-2.0/guide-start-installation.html>.

2. Create a migration, which adds an account table, using the following command:

./yii migrate/create create\_account\_table

3. Also, update the just- created migration using the following code:

<?php

use yii\db\Schema; use yii\db\Migration;

class m150620\_062034\_create\_account\_table extends Migration {

const TABLE\_NAME = '{{%account}}';

public function up()

{

$tableOptions = null;

if ($this->db->driverName === 'mysql') {

$tableOptions = 'CHARACTER SET utf8 COLLATE utf8\_general\_ci ENGINE=InnoDB';

}

$this->createTable(self::TABLE\_NAME, [

'id' => Schema::TYPE\_PK,

'balance' => ' NUMERIC(15,2) DEFAULT NULL',

], $tableOptions);

}

public function down() {