

Lab Homework 9: Database Transactions

In this lab we want to help you figure out when you should use database transactions and when you don't need to.

1. Learning Objectives

- Learn through exploration when you should use transactions and when it's not necessary.
- Code transactions using PDO's `beginTransaction()` and `commit()` functions.

2. Deadline

Lab Homework	Deadline	Slip Days	Credit	Solution
All Parts	Sun 4/26, 11:59pm ET	Max: 2 days	20 points (completion)	Provided

3. Instructions

1. **Clone** your lab repository.

Clone the following URL:

```
git@github.coecis.cornell.edu:info2300-2020sp/YOUR_GITHUB_USERNAME-lab09.git
```

Replace **YOUR_GITHUB_USERNAME** in the URL with **your Cornell GitHub username**.

2. **Work together.**

Feel free to work with your peers to complete this lab. Use your section specific chat rooms. Organize a Zoom hangout to work on the assignment together. Take this as an opportunity for some *virtual* human contact!

Note: You are encouraged to work together so long as you do your own work and you don't give away answers.

3. **Ask questions** or **say hi** during your registered section live Zoom Q & A.

Your section leaders will hold a Zoom Q & A during your registered section time. Feel free to pop in and **say hi** or **ask a question!** Again, use this as another opportunity to keep up with your fellow Cornell community members!

4. **Submit.**

When you're finished, follow the instructions in **submit.md** to submit your assignment.

Part I: Database Transactions

A transaction is a set of actions that you want treated as a whole. Meaning, all actions of a transaction are executed or no action is executed at all. Grouping actions into transactions make it so we prevent incomplete processes which would compromise the integrity of the data in our database.

Classic examples of transactions often deal with banks or sales. A simple example is the process of transferring \$100 from our 1300 bank account to our 2300 bank account. This is a two-step process. In a successful transfer, we...

1. We first have to withdraw \$100 from the 1300 account
2. Then we have to deposit the \$100 to the 2300 account.

However, if we were to check our balance in both accounts after only step 1 is executed, we would be given the wrong information. We would see we are missing \$100 from our 1300 bank account so it looks like we lost money. Additionally, if we finish step 1 and fail at step 2, the withdrawn money is now forever lost! Our balance is now incorrect.

With database transactions, we can ensure that this situation never happens and the data is never changed halfway. Additionally, it prevents people from disrupting and even accessing data while it has only been executed partway.

Transactions ensure that changes only happen if they are valid. For example, think of the situation when someone wants to borrow a book. We can break down the process into two simple steps; a successful borrow would have...

2. The librarian log that you have the book on your account
3. The librarian hand you the book

Let's say the librarian has signed the book out to you, but when she goes to get it for you, she discovers it's missing. If there was no transaction, you would have the book listed as borrowed when you never got it. Imagine how much in late fees you'd accumulate! If this were a transaction, the steps were not all completed and thus the changes made would not be committed.

1. Practice

Objective: Now that you understand what a transaction is, we will give you a variety of scenarios as well as queries. Open **transactions.md** and complete the questions for which scenarios should be transactions.

Feel free to discuss this with your peers to figure out when you need a transaction and when you don't. Just remember to do your own work!

2. Execution

Transactions are very simple to execute. The hardest part is often learning when they are necessary and which steps to group together, which we just practiced above.

To start a transaction with PHP's PDO extension, we use the `$db->beginTransaction();` function. To end a transaction we use the `$db->commit();` function. If you're unsure how this works, review your class notes and reference the PHP documentation.

Objective: Discuss with your peers whether you need a transaction and why. Next, edit the **bank.php** file to ensure the bank account balance has no chance of being recorded incorrectly if necessary.

Submit

Follow the instructions at the beginning of this document to submit your assignment.

1. Contributors

The following individuals made contributions to this assignment:

- Kyle Harms
- Sharon Jeong