

Transactions

A. Work in SQL Management Studio

Create a database with three tables:

Customer	
<u>id</u>	name

Account		
<u>id</u>	<u>customer_id</u>	name

Posting		
<u>id</u>	<u>account_id</u>	posting

Insert some default values. In this database, the balance is the sum of all the postings that belong to a given account.

Maybe you want to write an SQL statement that tells you the balance of a given account. Use group by, sum(...) , subselects and inner joins to solve this task.

B. Work from Eclipse

1. From Java, write a transaction that transfers funds between the accounts.

Rules are: if you can withdraw, you must also deposit. If either of these fails, you must roll back the transaction.

1.1 Test with both valid and invalid account_id values in the insert statements! Check if the postings did appear or not!

C. Partial solution

For the SQL, you can refer to the minibank-setup.sql file. Run it and you are set. This way, you can skip ahead to exercise B, but you should go back and solve exercise A on your own at a later point.

D. Add transaction support to an existing 3-layer architecture project (like FFC)

There is an example solution in ffc-db-exercise_10_session_5 – check the **MemberController**, the **TestMemberController**, and the **DBConnection** classes