Read *[Concurrency Control and Recovery](http://db.lcs.mit.edu/6.893/F04/ccandr.pdf)*, by Michael J. Franklin. **Skip section 3.2.**

This paper easiest to digest in chunks. You don't have to—in fact, probably shouldn't—read it all at once (or even read it in order).

* Section 1 introduces some basic terms and goals for database systems.
* Section 2 gives a good review of the basics of locking and logging that will be discussed in lectures this week.
* Section 3.1 (remember, skip 3.2) discusses some solutions to the problem of concurrency control (how to keep a database consistent even with interleaved operations from multiple users).
* Section 4 discusses some of the trade-offs of the transaction model discussed in the paper.

You should come to understand concepts such as serializability, (no-)force and (no-)steal, write-ahead logging, two-phase locking, degrees of isolation, etc.

As you read, think about the following:

* What failure models are we dealing with in this paper?
* Under what circumstances would you want transaction executions to respect the ACID properties? Are there systems that don't need to have all four properties?

**Question for Recitation**

**Before you come to this recitation**, write up (on paper) a *brief* answer to the following (really—we don't need more than a couple sentences for each question). If your TA has requested that you email your answer to them, you may do that instead, but it should still be handed in before your recitation begins.

Your answers to these questions should be **in your own words**, not direct quotations from the paper.

* What is an example from the paper that illustrates the trade-off between implementing ACID transaction properties and maintaining good performance?
* How does that policy or technique trade off performance?
* Why would you use this policy or technique? (In what context, under what circumstances, etc.)