UNIVERSITY OF ILLINOIS
AT URBANA-CHAMPAIGN

CS411 - Welcome!

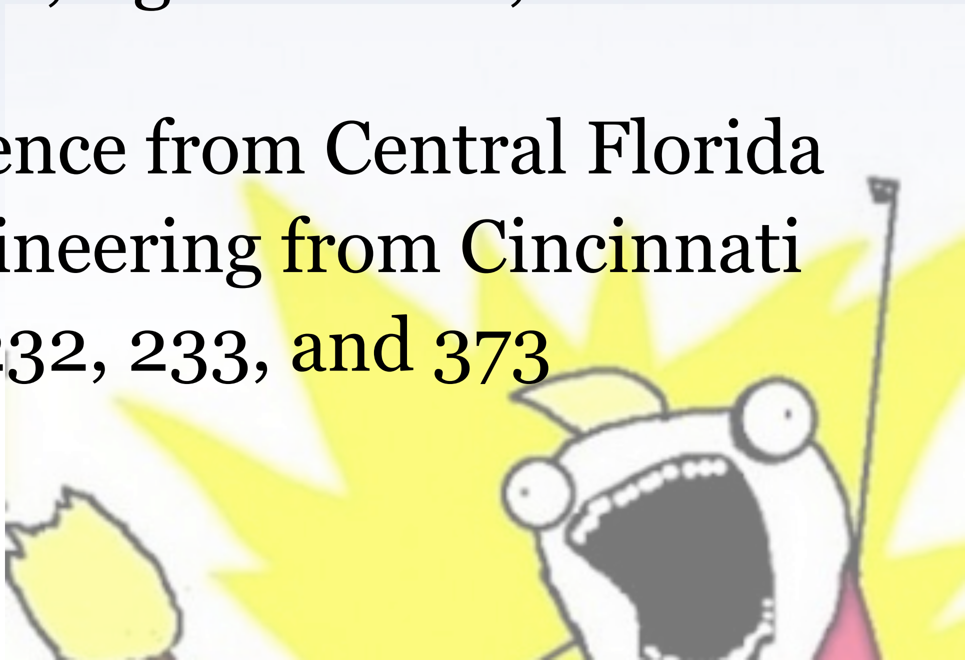
illinois.edu

About me

- Ryan “Not Dr.” Cunningham
  - A PhD student in bioinformatics
    - Dabbled in NLP, information retrieval, security, machine learning
  - Worked for DoD, DoE, Agribusiness, and Telecoms
  - MS in Computer Science from Central Florida
  - BS in Computer Engineering from Cincinnati
  - Taught in 125, 225, 232, 233, and 373

About me

- Ryan “Not Dr.” Cunningham
  - A PhD student in bioinformatics
    - Dabbled in NLP, information retrieval, security, machine learning
  - Worked for DoD, DoE, Agribusiness, and Telecoms
  - MS in Computer Science from Central Florida
  - BS in Computer Engineering from Cincinnati
  - Taught in 125, 225, 232, 233, and 373



About staff

- Nikita Spirin
  - Course projects and online students
- Khuram Shahzad
  - Course project track 1 and Piazza
- Magnesh Bendre
  - MP design
- Rui Wang
  - Written assignments

Course Website

- https://wiki.engr.illinois.edu/display/cs411sp13/Overview
- Syllabus, assignments, etc.
- All official course policies posted here

Grading Policy

Homework	25%
Projects	30%
Midterm	20%
Final Exam	25%

Course Projects

- Track 1: Database Web Application
  - Teams of 3-4 (form by Feb 6th)
  - Semester long project with several stages
  - Opportunity to be creative and ambitious
  - Start brainstorming!

Course Projects

- Track 2: Literature Survey or Research Extension
  - Required for those registered for 4 credits
  - Optional extra credit for others
  - Groups of 1-2
  - Either do a high quality literature review or expand your semester project into a serious research project

Homework

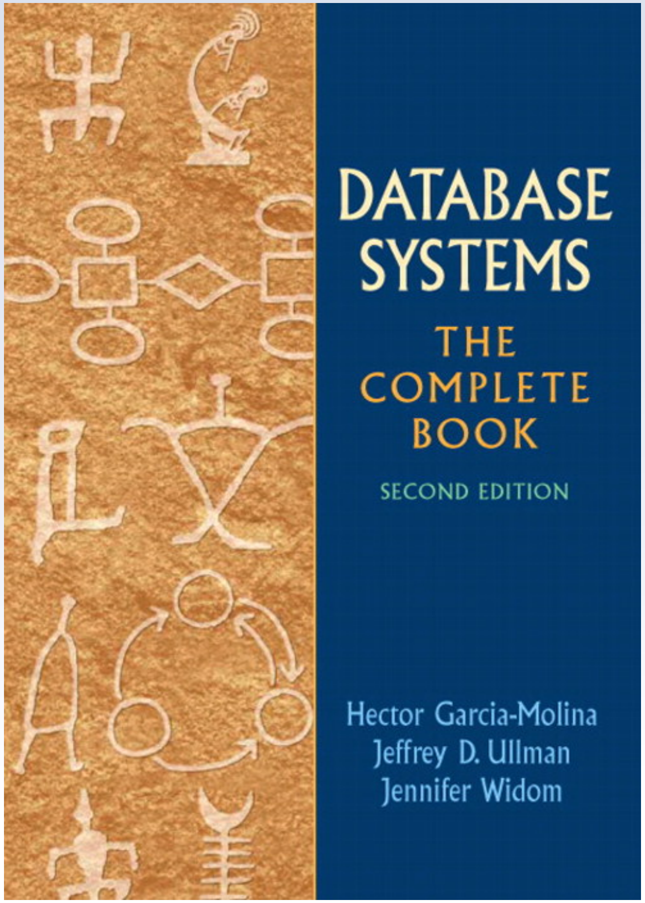
- 4-5 written assignments
  - Meant to reinforce concepts and prepare you for the midterm and final
- 3-4 programming assignments
  - Meant to help you understand the complexities of implementing a DBMS

Piazza

- https://piazza.com/class#spring2013/cs411
- A web forum where you can post questions
- Sign up ASAP so you don't miss out!
  - If you prefer your UIUC email, just sign up
  - If you prefer to use another account, please send me an email

Texbook

- *Database Systems: The Complete Book, Second Edition* by Hector Garcia-Molina, Jeffrey D. Ullman, and Jennifer Widom



Why study databases?

- Most computer science assumes we can manipulate data in RAM
- What to do data is much larger than RAM?
- This is very common:
  - credit card transactions, mobile phones, search engines
  - Google operates on *petabytes* of data

Why study databases?

- Without them, our current way of life would be impossible.
  - – No Google, iPhone, Facebook, or Amazon!
- Database systems are crucial for our infrastructure and economy

Why study databases?

- Concepts are extremely useful in other domains

What makes databases different?

- Can't restructure the data for each computation (only one *schema*)
- Efficiently use the entire system (CPU, RAM, Disk, *and* Network)
- Data should be *persistent* and continuously updated
- Multiple concurrent users

What is a database?

- A *database management system* (DBMS)
  1. Allows users to specify *schema* (logical structure) of their data with a *data-definition language* (DDL)
  2. Allows users to *query* the data (perform computation on the data) with a *data-manipulation language* (DML)

What is a database?

- A *database management system* (DBMS)
  3. Supports *persistent storage* of large amounts of data in a way that supports 1 and 2 above
  4. Enables *durability* in the face of failures
  5. Controls accesses by multiple users, ensuring
     - *isolation* (user's access is independent of others)
     - *atomicity* (an action is never performed partially)

History

- Problems first encountered in the 1960's
  - Banking systems
  - Airline reservations (surprisingly important)
  - Corporate records
- Essentially, people were building *ad hoc* systems on top of file systems
- Each query required a *custom program!*

History

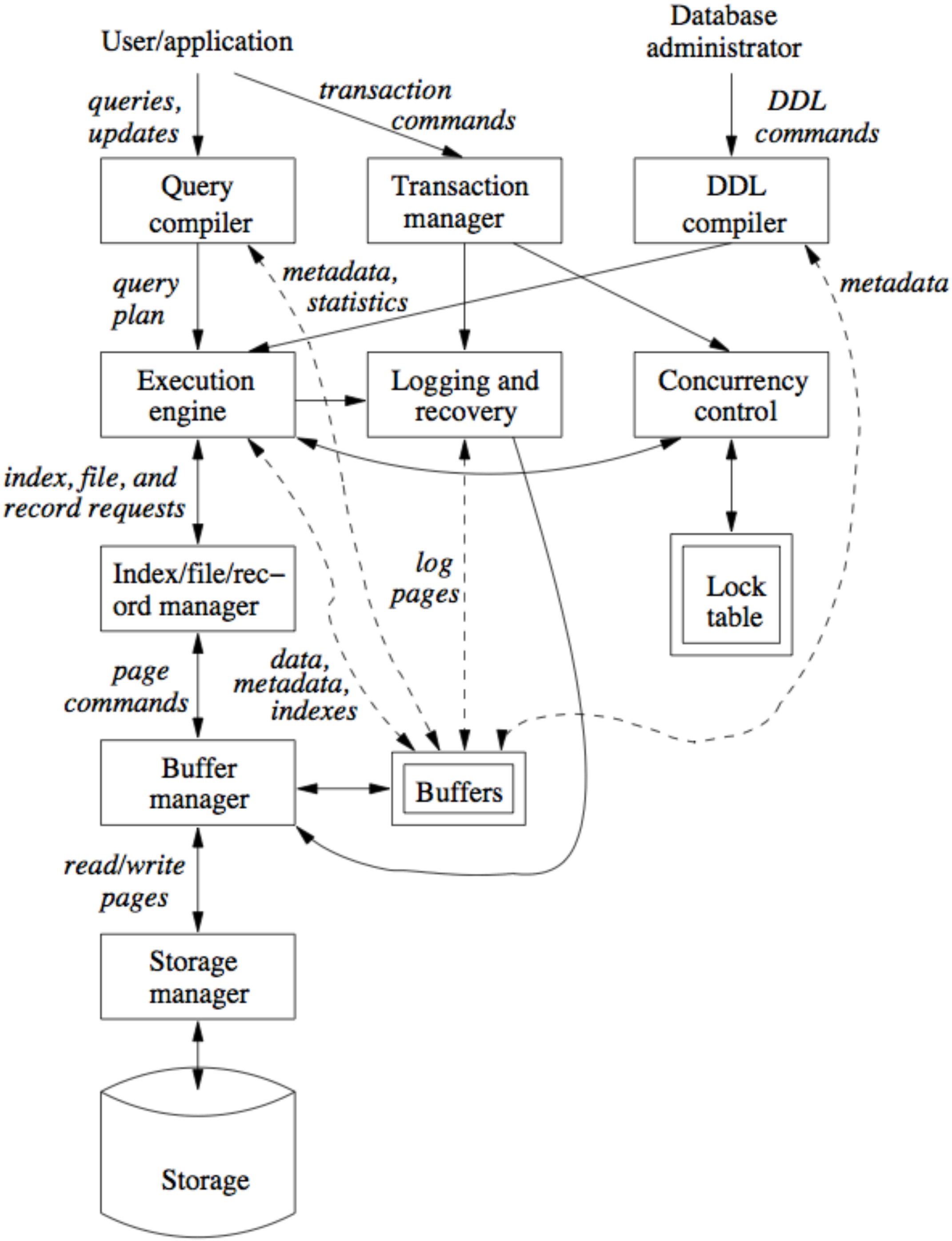
- In 1970’s Ted Codd wrote “A relational model for large shared data banks”
  - Proposed a *relational model* of data
  - Data storage abstracted from user
  - Supported high level query language
- Through the 1980’s and 1990’s, this model became standard and widely adopted

History

- From 2000's to present:
  - – Codd's model is still the core paradigm of the DBMS infrastructure
  - – But much more data that is less organized
    - Images, video, social networking
  - – Peer-to-peer and parallel systems developed
  - – Extended and supplemented relational model in light of these developments

An overview

- How does a DBMS work?
  - Here’s an overview

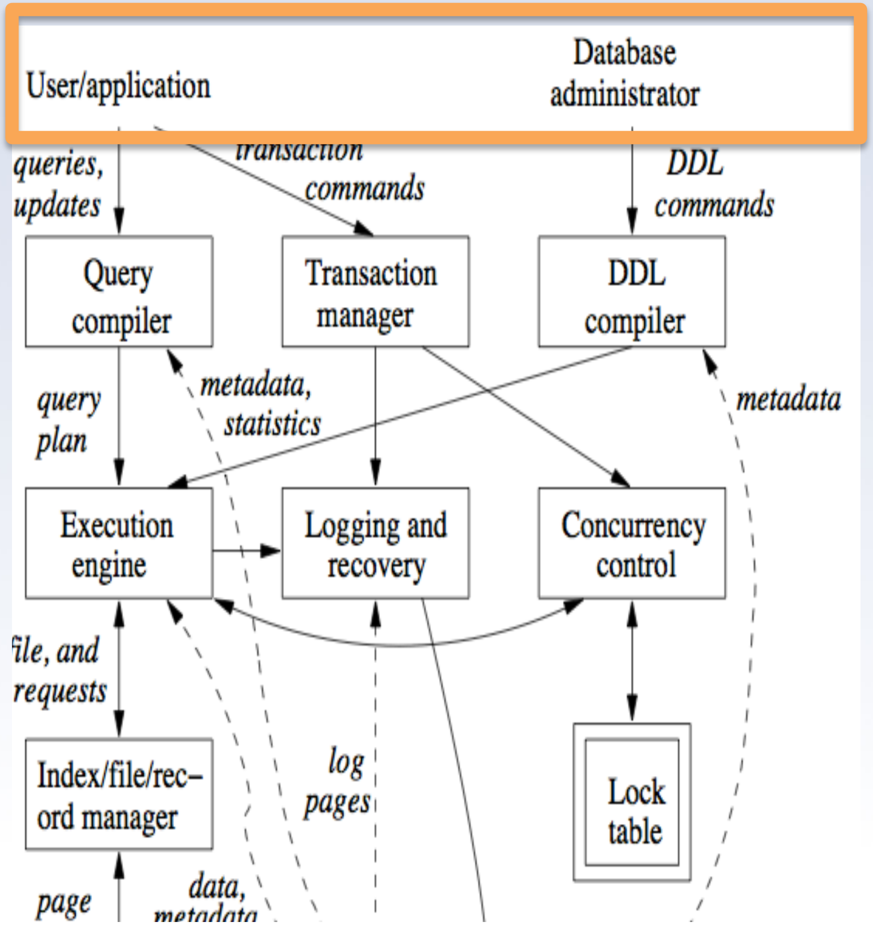

User/application
Database administrator
queries, updates
transaction commands
DDL commands
Query compiler
Transaction manager
DDL compiler
query plan
metadata, statistics
metadata
Execution engine
Logging and recovery
Concurrency control
index, file, and record requests
Index/file/rec–ord manager
log pages
Lock table
page commands
data, metadata, indexes
Buffer manager
Buffers
read/write pages
Storage manager
Storage

Did you get all that?

- We'll spend all semester learning about these systems
- But let's break this down to get a little preview

Interacting with the DBMS

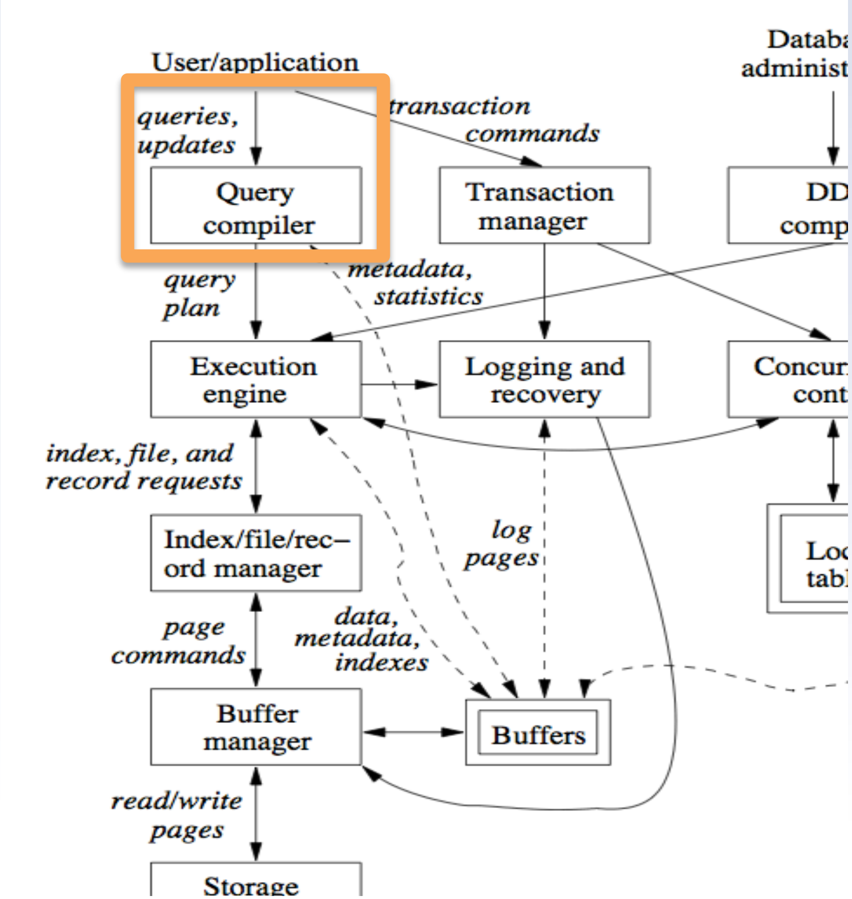
- Two ways to interact with the DBMS
  1. As a "user" interacting with the data
  2. As an "administrator" modifying the structure of the data



illinois.edu

Focusing on the user

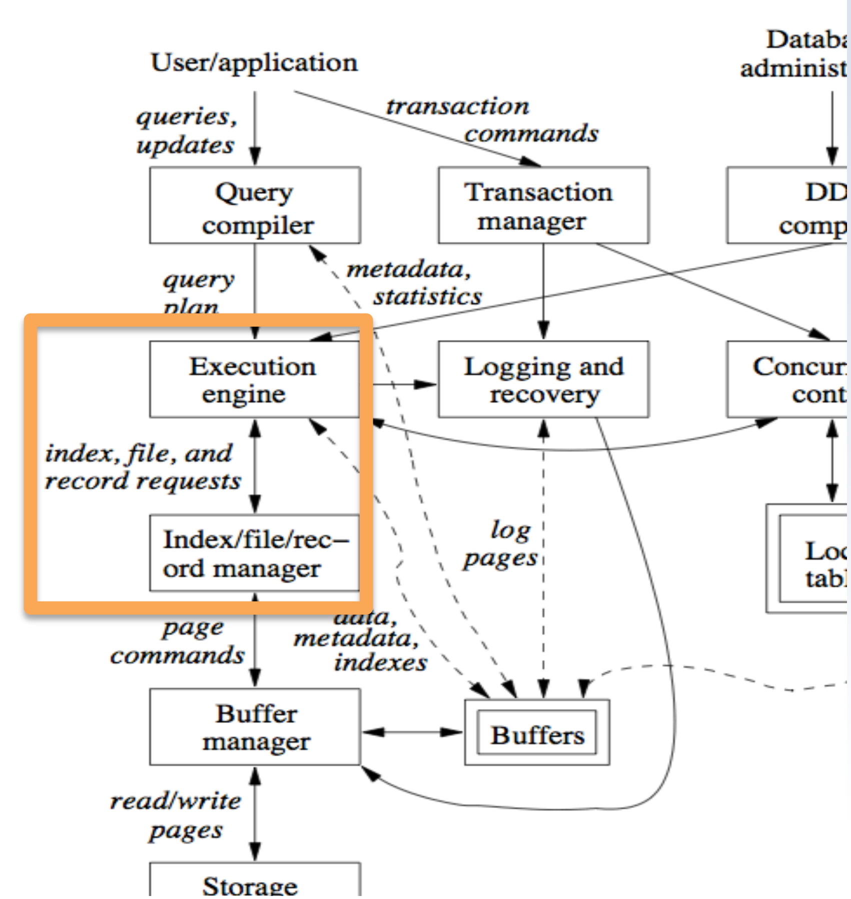
- Users submit queries to the query compiler in a data manipulation language (DML)
- Parsed by the query compiler into a query plan



illinois.edu

Focusing on the user

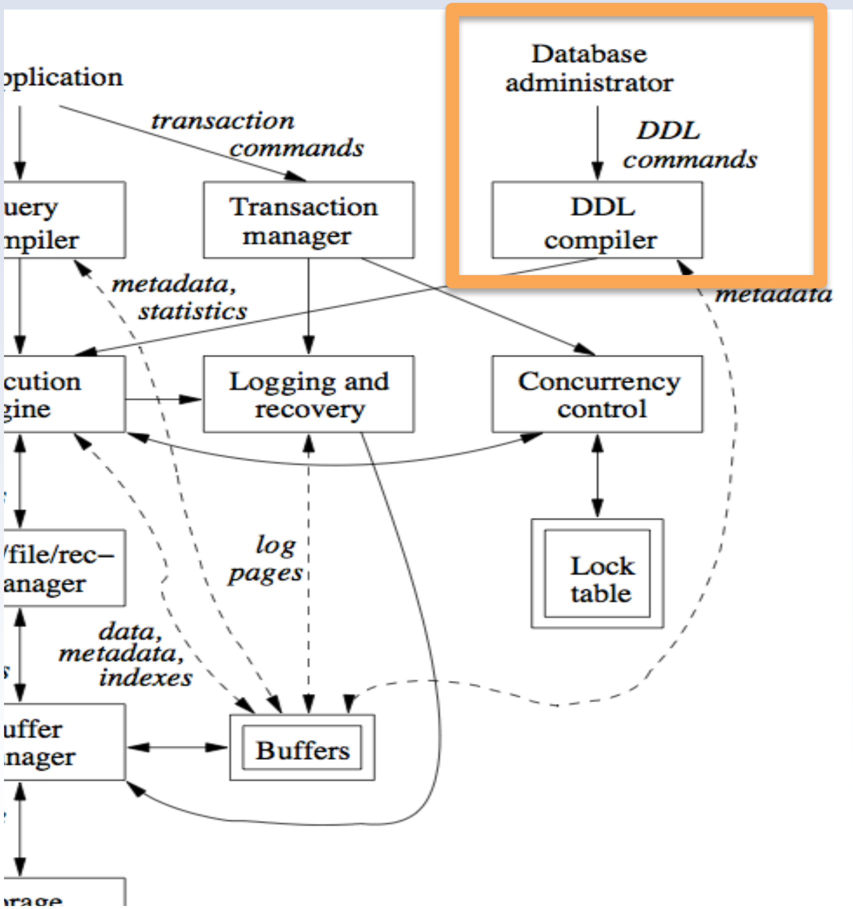
- Query plan is executed by the execution engine
- Sends specific low level requests to the index/record manager to get the data



illinois.edu

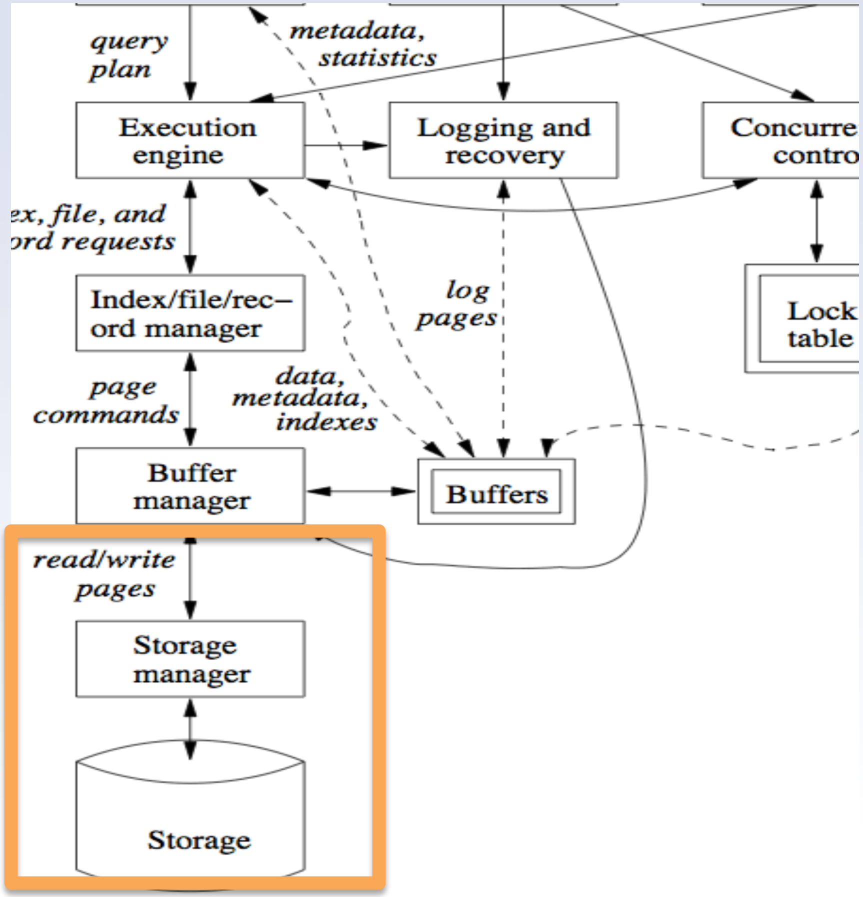
Focusing on the admin

- The database administrator (DBA) sends data definition language DDL commands to the DDL compiler
- Also sent to the execution engine



Storage and Buffer Management

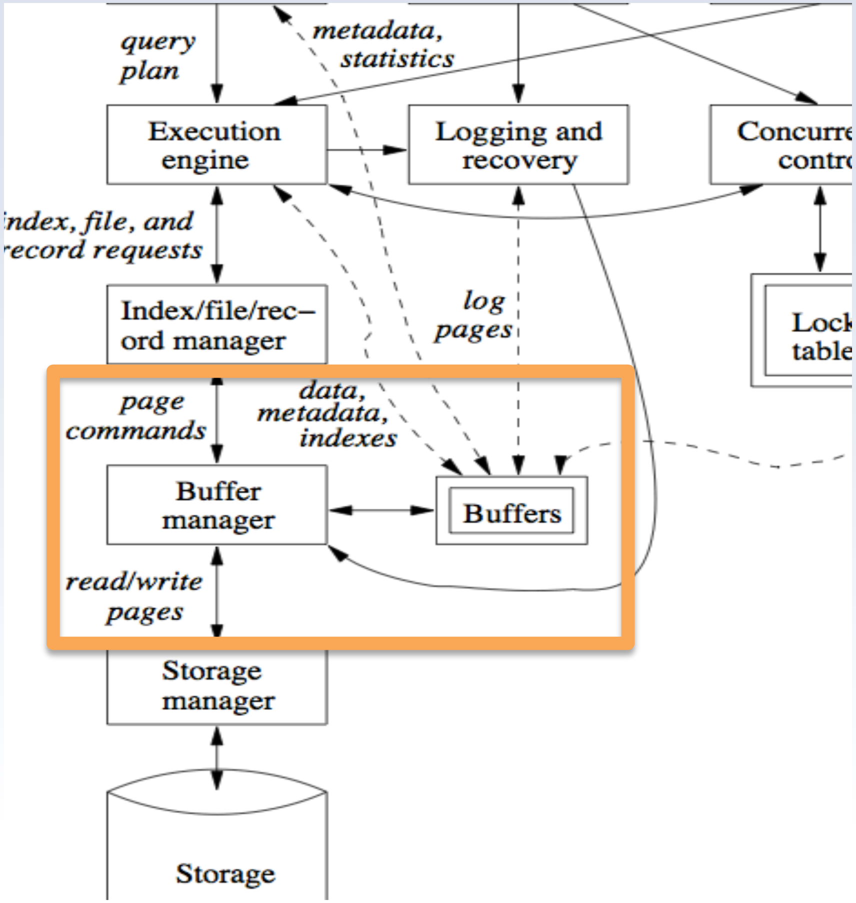
- Storage manager keeps track of where the data is
  - Stored in large chunks so we can access it in bulk
  - Transferred in and out of RAM in *pages*



illinois.edu

Storage and Buffer Management

- Buffer manager partitions RAM into *buffers*
  - essentially keeps data in page sized chunks that we can perform computation on

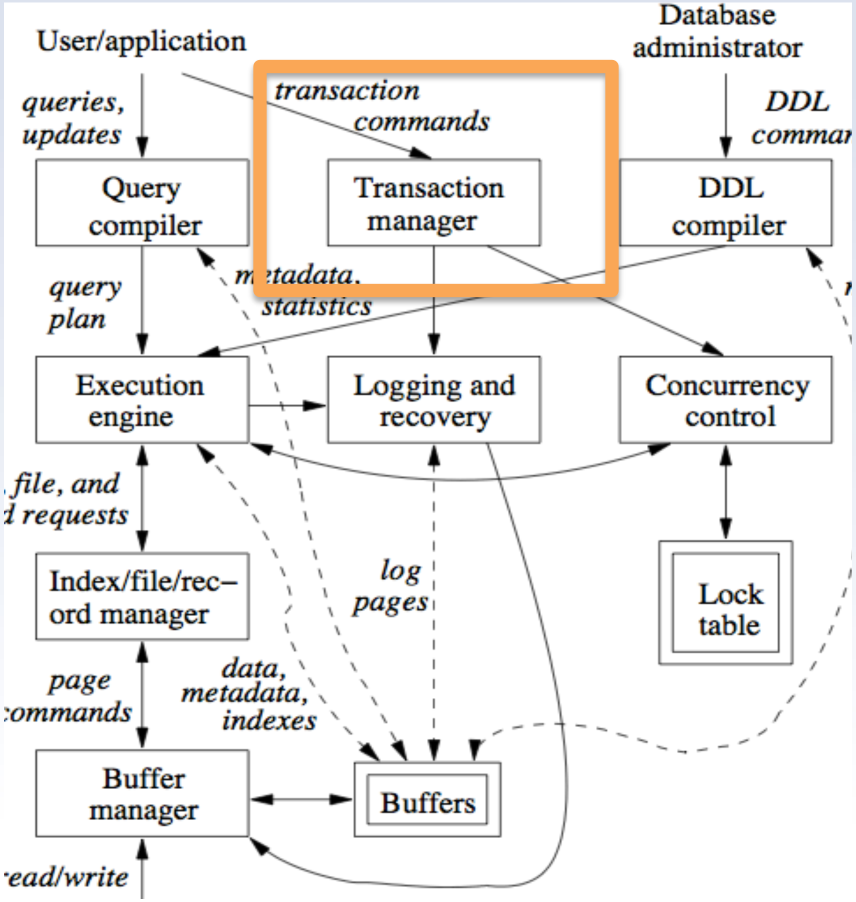


Transaction Management

- ACID Test
  - – Atomicity - “all or nothing”
  - – Isolation - “don’t interfere”
  - – Consistency - “maintain constraints”
  - – Durability - “don’t lose anything”

Transaction Management

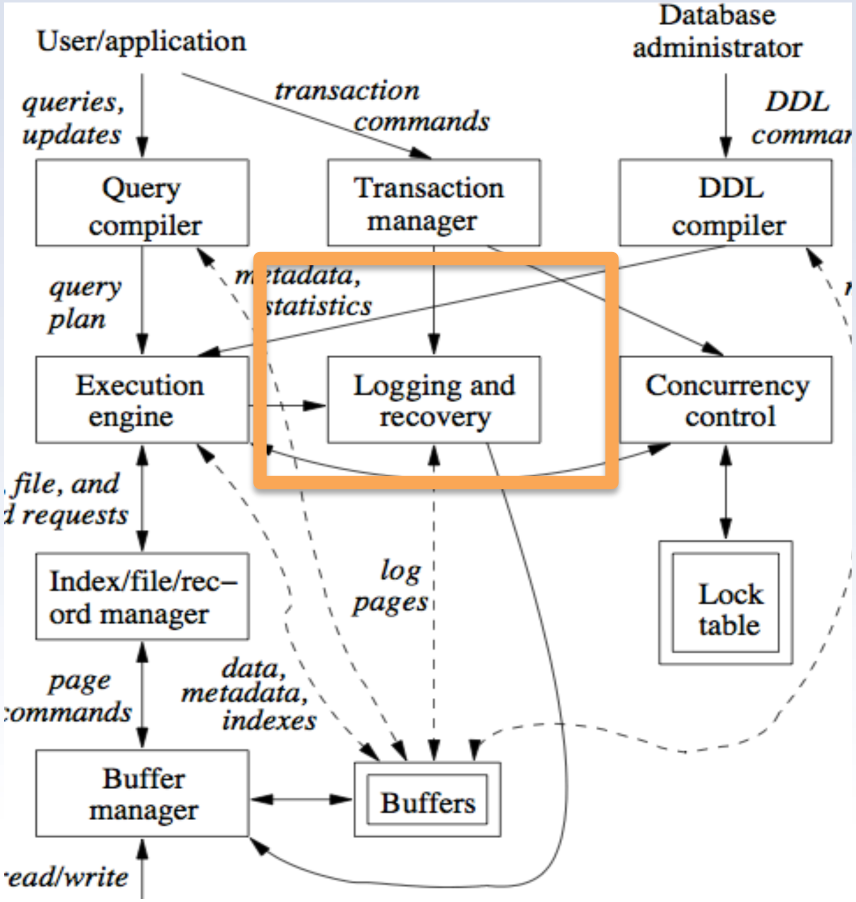
- Transaction manager receives units of work called *transaction commands*
- It makes sure ACID test is satisfied for all transactions



illinois.edu

Transaction Management

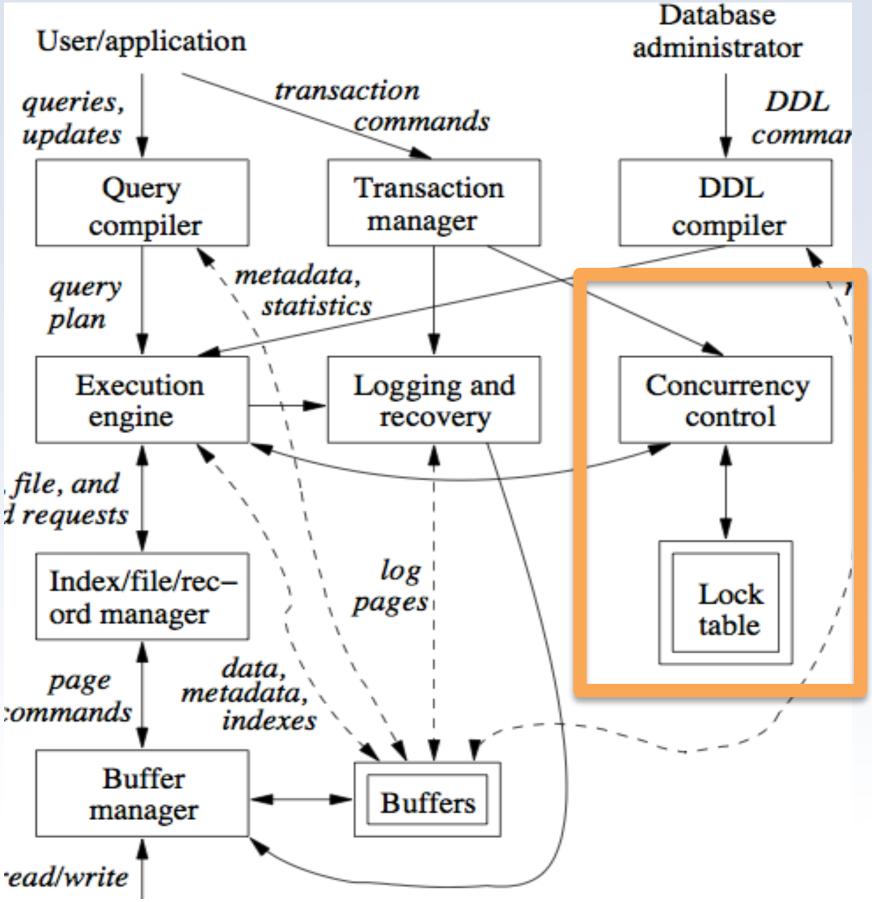
- Logs execution of transactions so transactions that fail can be recovered



illinois.edu

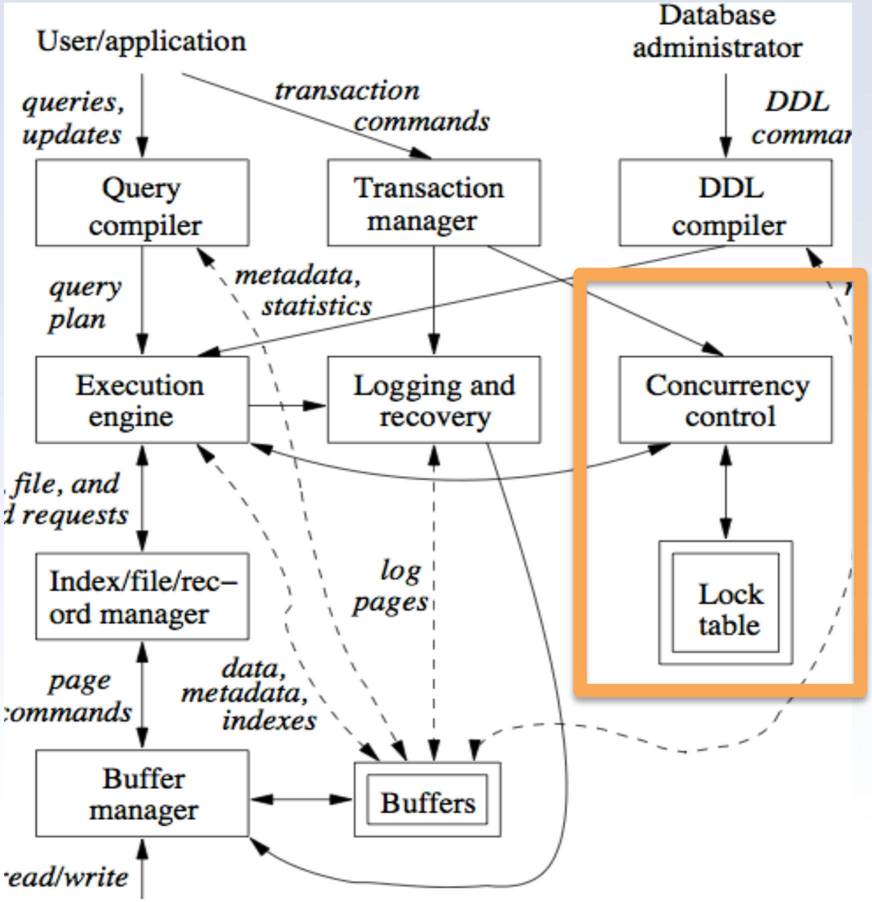
Transaction Management

- Tracks concurrently executing transaction commands
- Locks parts of the database as needed to ensure transactions don't interfere with each other



Transaction Management

- If multiple conflicting requests are waiting for the same data, must perform *deadlock resolution*



Course Overview

1. Relational Model: Query/DML
   - Theoretical and practical perspective
2. Relational Model: Design/DDL
   - Theoretical and practical perspective
   - Advanced Manipulation concepts
3. DBMS Implementation

Course Overview

4. Advanced Topics
   - Parallel/Distributed Databases
   - Information Integration
   - Data Mining/Information Retrieval

Next Lecture

- We’ll start learning about Codd’s relational model