PURDUE UNIVERSITY | Department of Computer Science

CS 44800: Introduction To Relational Database Systems

Prof. Chris Clifton

26 August 2021

Course Overview

Indiana
Center for
Database
Systems

PURDUE UNIVERSITY
Department of Computer Science

What are we studying?

- Methods to build databases
  - Data modeling
  - Query languages

  We'll try to cover this quickly – many may know it
- Methods to build DBMSs
  - Storage (safe, persistent)
  - Query (how to make them fast)
  - Transactions (how to make a lot happen at once)

Course Outline
(very rough)

1. Course Introduction
   - Intro / history lesson
   - Relational Model, Data Modeling
2. Relational Database Queries
   - Relational Algebra and SQL
4. Storage mechanisms: Rotating and Otherwise
5. Indexing and Hashing
6. Query Processing
7. Query Optimization

8. Handling Failure
9. Concurrency Control
10. Transaction Management
11. Using a Relational Database
    - Views
    - Constraints
    - Triggers
12. Big Data and Other Advanced Topics

What goes in to a DBMS?
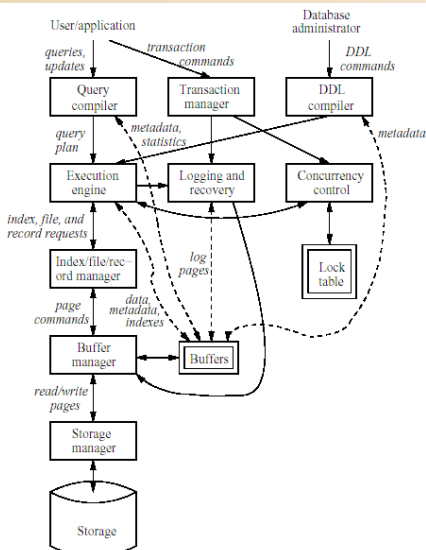
- Query Compilation
  - Turn a declarative query to procedural execution
  - What is the fastest way to get the result?
- Transaction Management
  - Try to run lots at once
  - Ensure queries don't interfere with each other
- Storage Management
  - Disks are slow – how do we get to the data fast?
  - Minimize trips to the disk

Some Goals of a DBMS

Data Integration	——————→	Enhances the accessibility of data, reduces redundancies and inconsistencies
Data Independency	——————→	Simplifies the development of new applications, and the maintainance of existing applications
Centralized Data Control	——————→	Assures data quality, confidentiality, and integrity

Data Models

- A data model allows one to represent real-world entities of interest to a given set of applications
- It is thus useful to identity the basic concepts of such representation; relevant concepts include:
  - *Entity*: an "object" of the application domain
  - *Attribute*: a property of a given entity which meaningful, for the description of the application domain

    Each entity is thus characterized by one or more attributes; an attribute takes one or more values, referred to as *attribute values*, from a set of possible values; such set if referred to as *attribute domain*