CS 445
Introduction to Database Systems

TTh 2:45-4:20pm

Chadd Williams

Overview

- Practical introduction to databases
  - theory + hands on projects
- Topics
  - Relational Model
    - Relational Algebra/Calculus/ Theory
  - Database Design
    - ER Diagrams
  - Structured Query Language (SQL)
  - Web accessible databases / Architecture / Model-View-Controller
  - Non-structured Data (NoSQL)
- There will be a number of lab days for hands on work

Syllabus

- *Database Management Systems* (3rd), Ramakrishnan & Gehrke
- Grades:

Midterm 1	15%
Midterm 2	15%
Final	20%
Homework/Labs/Quizzes	10%
Database Projects	40%

First DB Assignment	Design Docs	8 pts
	MySQL DB	12 pts
Big DB Assignment	Design Docs	25 pts
	MySQL DB	30 pts
	Web Interface	15 pts
	Presentations	10 pts

- Quizzes: frequent, unannounced, open-note quizzes will be given
- Late Policy: No late assignments accepted
- Grade Complaints: one paragraph summary of why the grade is wrong, **within one week of receiving the graded material**

- http://zeus.cs.pacificu.edu/chadd/cs445f19

Database Projects

- All database projects are to be done using MariaDB
  - https://mariadb.org/
  - http://www.apachefriends.org/ (XAMPP) LAMP
  - PHP 7 for web projects
- Tools
  - dbeaver
  - CodeLite
- First DB Assignment
  - Learn to use MySQL & SQL
- Big Database Project
  - You (and a friend) design, document, and implement a database
  - Build a web-based front end
  - 5 minute presentation of your design / 7-10 minute presentation of your final design and implementation

- Linux skills are a must
  - ssh
  - cp
  - mv
  - cd
  - scp
  - geany
  - ls
  - cat
  - less
  - diff
  - tar

  If you don't have your zeus password, see me soon!

Refresh:

http://zeus.cs.pacificu.edu/chadd/cs300f18/Assignments/BasicLinuxExercise.pdf

Introduction to Databases

- Read Chapter 1

- What's a database?
  - DBMS?

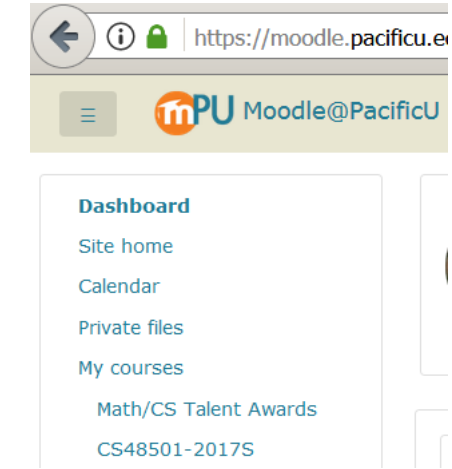
- Why do we use one?

- Who uses one?

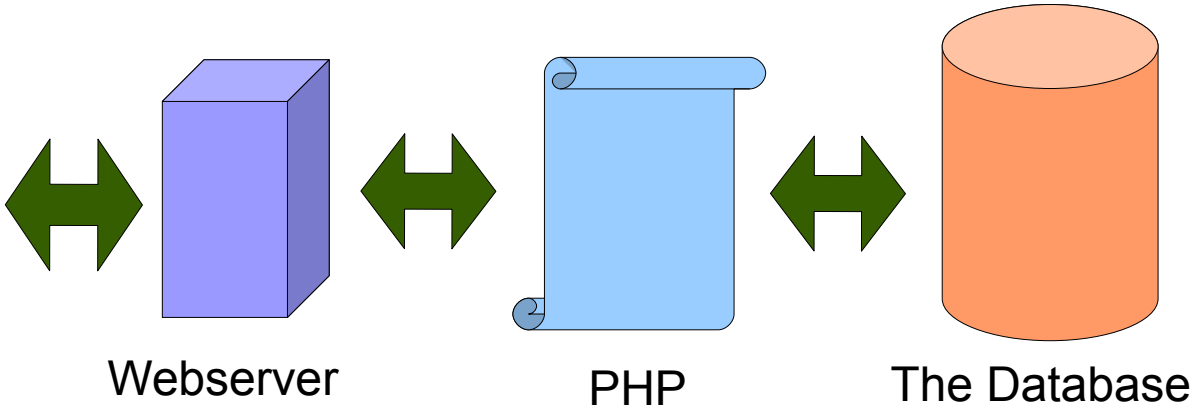
- How do we model the data?

Review Questions at the end of each chapter are great exam and quiz study guides.

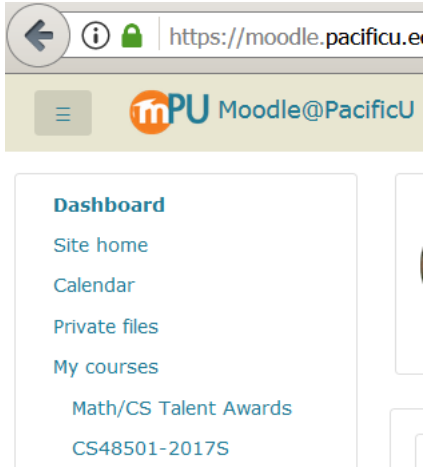
Basic Database Usage Scenario



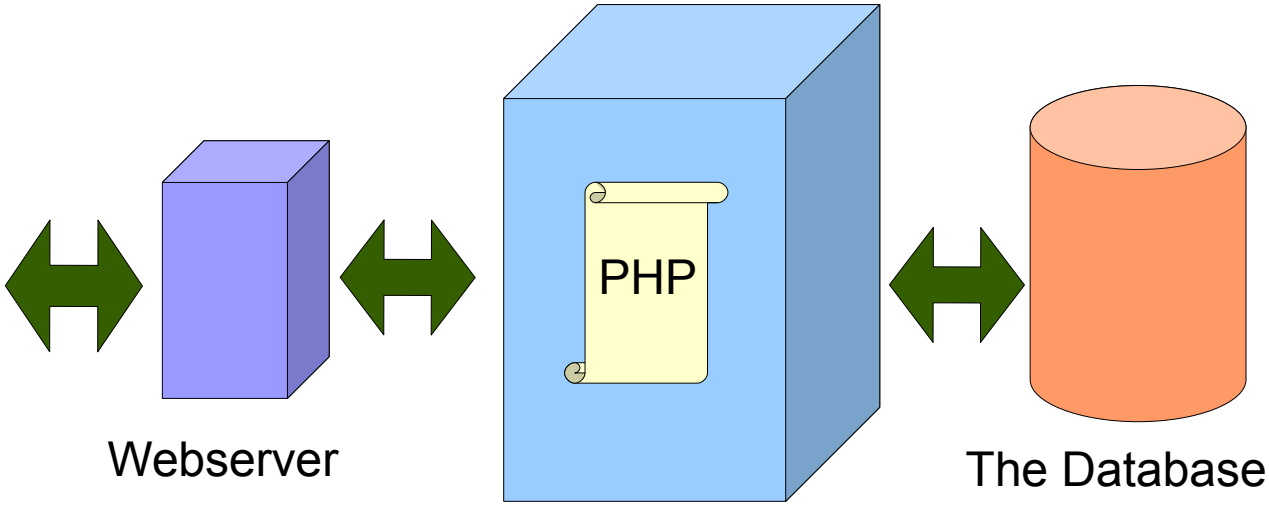
User



More Complicated Database Usage Scenario



User



Database Trends

- SQL / RDBMS
  - relational database systems
  - entities are related to other entities
  - relationships my optionally have attributes
  - strict structure of data (tables, (rows, columns, datatypes))
  - SQL: Structured Query Language
  - Very powerful and expressive
    - ad-hoc queries
  - ACID: strict guarantees of correctness
    - Atomicity, Consistency, Isolation, Durability
  - Online Analytical Processing (OLAP)
    - business intelligence
    - complex queries / ad hoc queries
    - structured data
  - originally: poor performance on web scale data

Database Trends

- NoSQL
  - built for web data
  - Online Transaction Processing (OLTP)
    - poorly defined
    - availability, speed, concurrency, recoverability
    - high throughput/insert or update intensive
    - web services
    - often no ad-hoc queries
  - Schema-less (collection of fields)
  - built for big data / lots of text data (document store, graphs)
  - Availability/Scalability/Low Latency
  - Distributed (Consistency/Availability/Partition Tolerance (2))
  - may sacrifice consistency

Database Trends

- NewSQL
  - Relational data model
  - Make SQL scalable like NoSQL
  - Maintain ACID
  - non-locking concurrency (reads and writes don’t conflict)
  - high per node performance
  - scale-out, shared nothing architecture
  - Online Transaction Processing (OLTP)
  - VoltDB
  - MySQL Cluster database engine

Why not just use a text file/file system/XML?

- Data Independence
- Efficient Data Access
- **Data Integrity and Security**
- Data Administration
- Concurrent Access/Crash Recovery
- Reduced Application Development Time

(page 9)

Storing data in the DB

- Data Models
- Semantic Data Model (high level)
  - Entity-Relationship (ER) Model
    - Entity:
    - Relationship:
- Relational Data Model (low level)
- Schema
- Constraints/Integrity

Relational Databases

- Well defined structure of data
  - schema
- Flexible queries

NoSQL databases
schema

queries

What's inside a Relational database?

- Tables
- Indexes/Keys
- Data

How do we access the data?

- Query Language
  - Structured Query Language (SQL)
  - What types of queries can we run?

What about multiple users?

- Transactions

- Concurrency

Dirty Details

- Figure 1.3 page 20

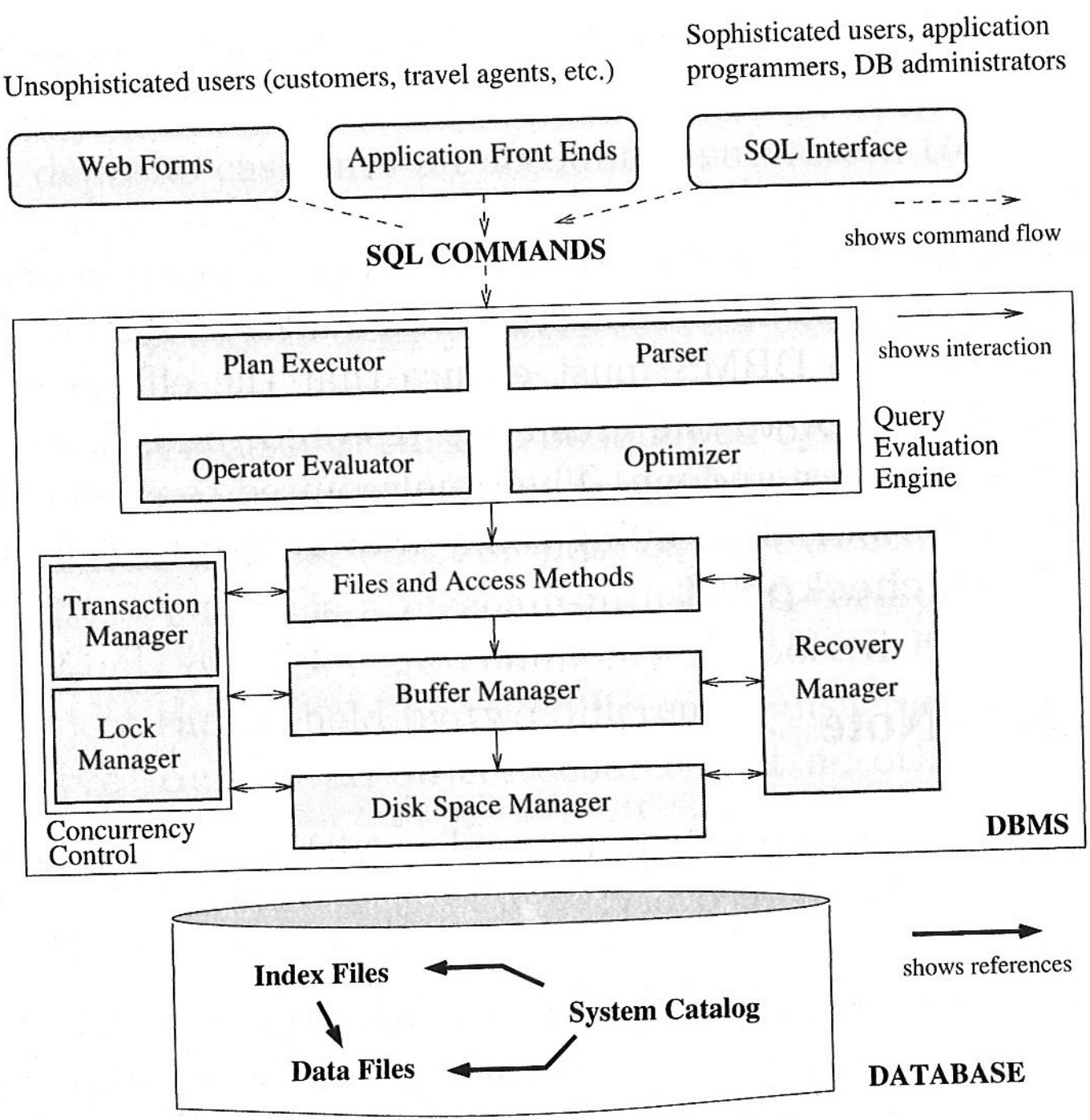


Figure 1.3 Architecture of a DBMS

Car Dealership

- Work in pairs
- You have 15 minutes then we will discuss