Describing Data: Data Models

- A *data model* is a collection of concepts for describing data.
- A *schema* is a description of a particular collection of data, using a given data model.
- The *relational data model* is the most widely used model today.
  - Main concept: *relation*, basically a table with rows and columns.
  - Every relation has a schema, which describes the columns, or fields.

1

Data Independence

- Applications insulated from how data is structured and stored.
- Logical data independence: Protection from changes in *logical* structure of data.
- Physical data independence: Protection from changes in *physical* structure of data.
- Q: Why is this particularly important for DBMS?

Because rate of change of DB applications is slow. More generally: *dapp/dt << dplatform/dt*

2

Concurrency Control

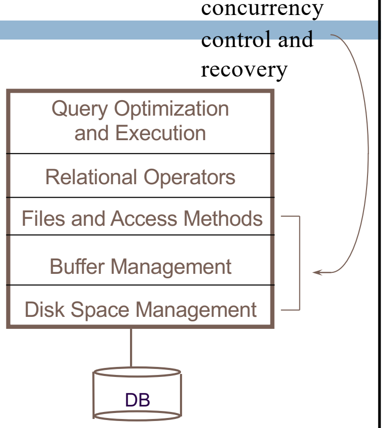
- Concurrent execution of user programs: key to good DBMS performance.
  - Disk accesses frequent
  - Keep the CPU working on several programs concurrently.
- Interleaving actions of different programs: trouble!
  - e.g., account-transfer & print statement at same time
- DBMS ensures such problems don't arise.
  - Users/programmers can pretend they are using a single-user system. (called "Isolation")
  - Thank goodness! Don't have to program "very, very carefully".

3

Structure of a DBMS

These layers must consider concurrency control and recovery

- A typical DBMS has a layered architecture.
- The figure does not show the concurrency control and recovery components.
- Each system has its own variations.



Query Optimization and Execution

Relational Operators

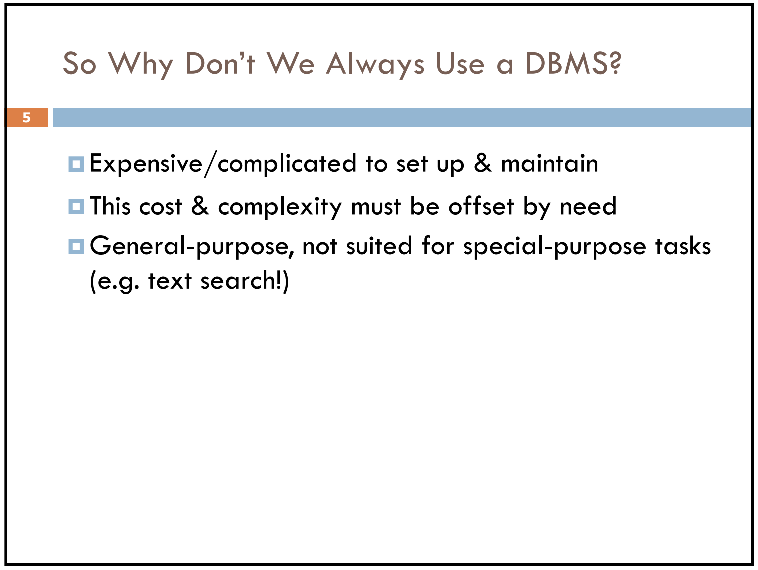
Files and Access Methods

Buffer Management

Disk Space Management

DB

4

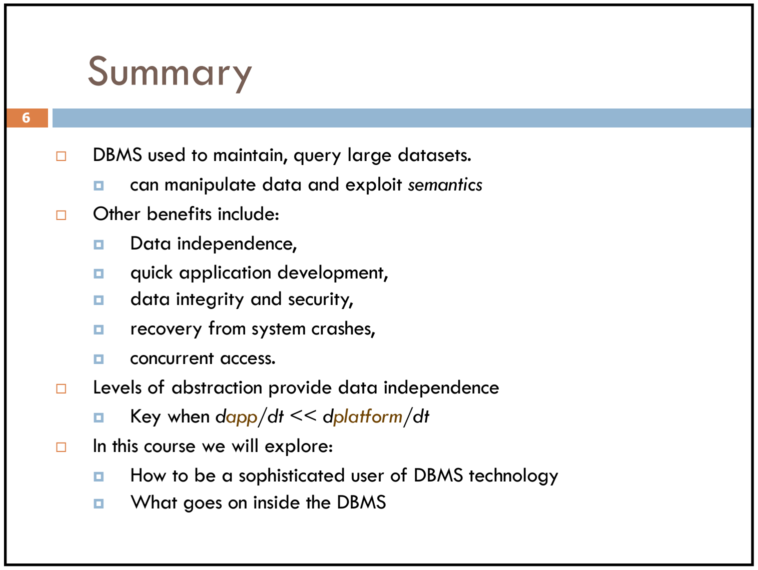


So Why Don’t We Always Use a DBMS?

5

- Expensive/complicated to set up & maintain
- This cost & complexity must be offset by need
- General-purpose, not suited for special-purpose tasks (e.g. text search!)

5



Summary

6

- DBMS used to maintain, query large datasets.
  - can manipulate data and exploit *semantics*
- Other benefits include:
  - Data independence,
  - quick application development,
  - data integrity and security,
  - recovery from system crashes,
  - concurrent access.
- Levels of abstraction provide data independence
  - Key when $dapp/dt << dplatform/dt$
- In this course we will explore:
  - How to be a sophisticated user of DBMS technology
  - What goes on inside the DBMS

6