Before we begin

- A note on expectations
- This is a 400 level class, and as such you are expected to be quite independent
- I want you to ask questions, but these should be material-related and specific
  - Example: I found there are two ways of representing this in ER model, which one is better?
- About tooling
  - Be resourceful and figure out tooling problems, i.e., installation, troubleshooting
    - Canvas discussion forum is a great place to discuss this
  - I won't answer software troubleshooting questions, because most likely, I don't know the problem specific to your computer and setup
    - Example: I do not know why your database engine is throwing "Error 0x3489F0"

INTRODUCTION

Databases are everywhere

- Who has experience working with databases?
- What type of databases?
- In what role?
- What type of application

Databases are everywhere

- What is a database?
- Where are databases used?
- Examples
  - Traditional database applications – business, banking, libraries, school, etc.
  - Geographic information Systems (GIS)
  - Data Warehouses and Online analytical processing (OLAP)

What is a database?

- **Database** – collection of related **data**
- **Data** – known facts that can be recorded and have implicit meaning
- Size of database?
  - Can vary greatly

Common types of databases (big picture)

- Relational
  - Tables, keys, relating data across tables
- Object oriented
  - Translating between programmed objects and tables (object-relational mappings)
- NoSQL
  - XML data, document-oriented, XML queries
- We'll focus on Relational, and discuss others if time allows

Example and discussion

- If you were asked to develop a software system for a university which has the following data records saved in excel:
  - STUDENT
  - COURSE
  - SECTION
  - GRADE_REPORT
  - PREREQUISITE
- Can this collection of data be called a database?

- Discussion (10-minutes):
  - At a high-level: What kind of functionalities/properties your software system should provide/have?
    - assume using file-based system & without budget/schedule constraints

Database Management System (DBMS)

- Collection of programs that enables users to create and maintain a database
  - **Facilitates** defining, constructing, manipulating, and sharing databases
  - **Provides** support protecting and maintaining database

Example

STUDENT

Name	Student_number	Class	Major
Smith	17	1	CS
Brown	8	2	CS

COURSE

Course_name	Course_number	Credit_hours	Department
Intro to Computer Science	CS1310	4	CS
Data Structures	CS3320	4	CS
Discrete Mathematics	MATH2410	3	MATH
Database	CS3380	3	CS

SECTION

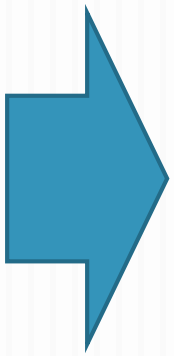
Section_identifier	Course_number	Semester	Year	Instructor
85	MATH2410	Fall	07	King
92	CS1310	Fall	07	Anderson
102	CS3320	Spring	08	Knuth
112	MATH2410	Fall	08	Chang
119	CS1310	Fall	08	Anderson
135	CS3380	Fall	08	Stone

GRADE_REPORT

Student_number	Section_identifier	Grade
17	112	B
17	119	C
8	85	A
8	92	A
8	102	B
8	135	A

PREREQUISITE

Course_number	Prerequisite_number
CS3380	CS3320
CS3380	MATH2410
CS3320	CS1310



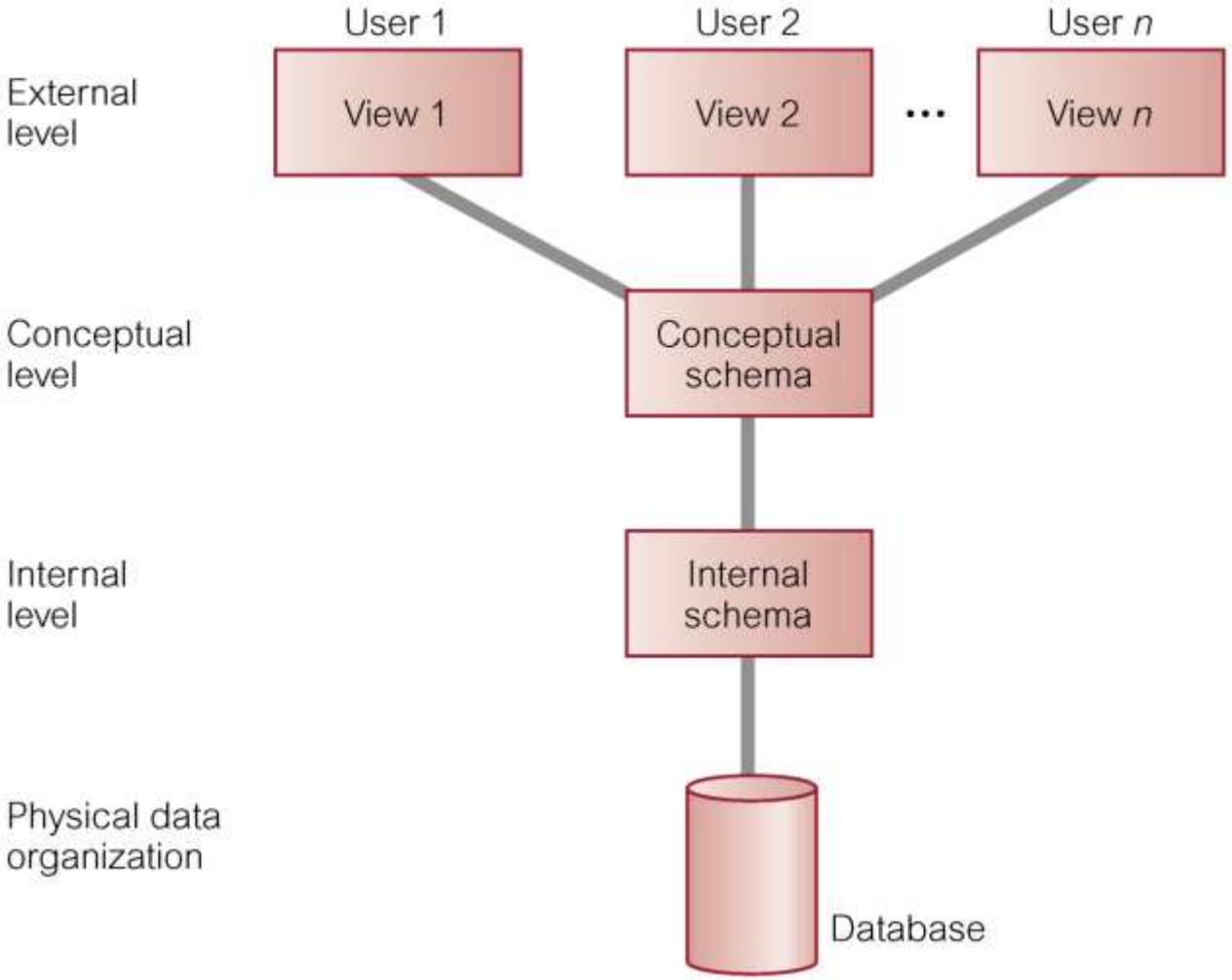
Defining
Constructing
Manipulating
Sharing
Protecting
Maintaining

A database that stores student and course information.

Concepts

- **Defining**: specifying data types, structures, and constraints (saved as Meta data)
- **Constructing:** storing data physically
- **Manipulating:** query, update, generate report, etc.
- **Sharing:** concurrent access
- **Protecting:** System protection; Security protection
- **Maintaining:** evolving as requirements change

ANSI-SPARC Three-level Architecture



Part of database that is relevant to a particular user

Meta-data, schema, System catalogue

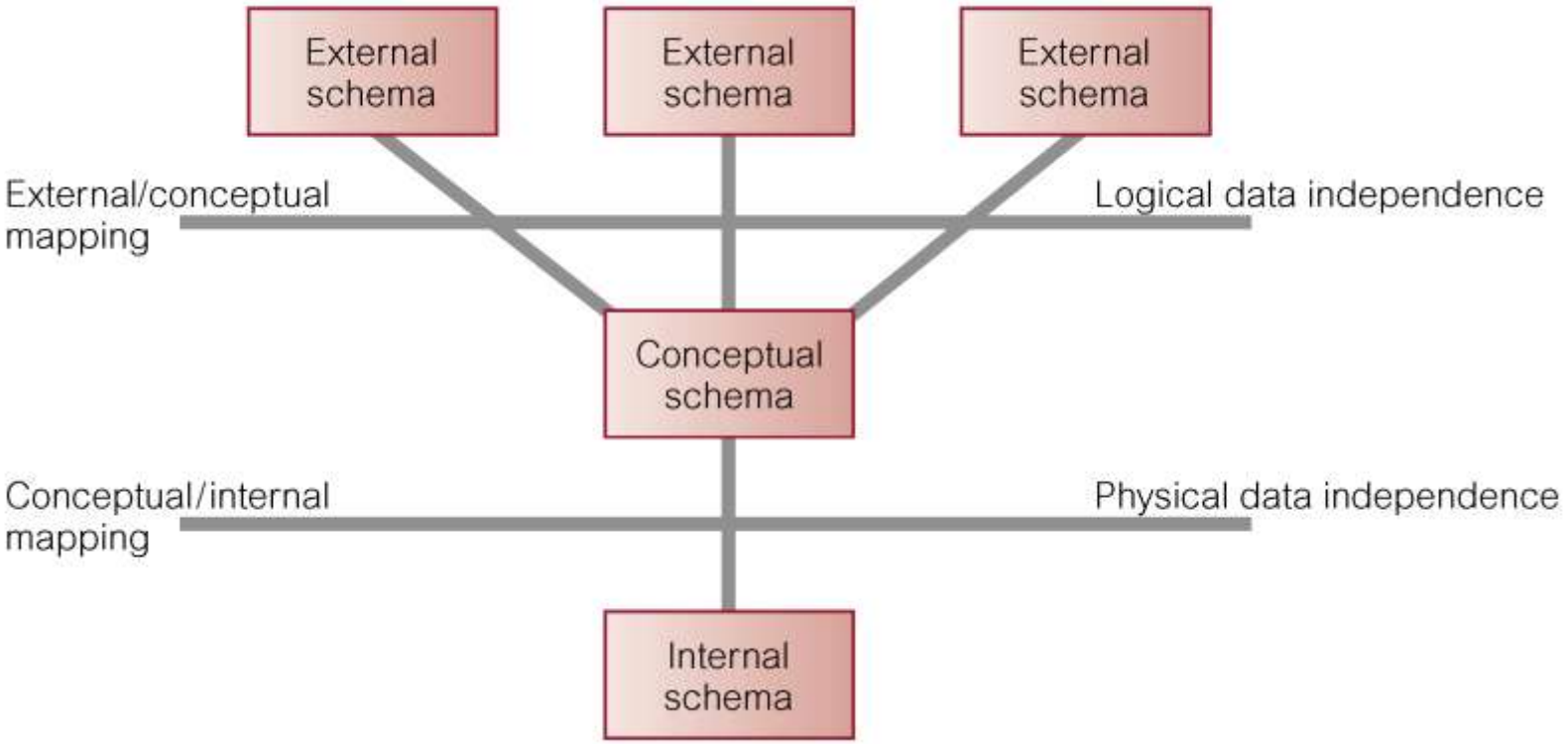
Details of data storage and access path

Characteristics of the Database Approach

- **Database approach**: single repository of data is maintained

- Main characteristics of database approach
  - Data **independence**
  - Support of multiple **views** of the data
  - **Sharing** of data and multiuser transaction processing
    - Concurrency control

Data Independence



Some Advantages of Using the DBMS Approach

- Enforcing Integrity Constraints
  - Data type, value length, range, etc.
  - Referential integrity constraint
    - E.g, every section record must be related to a course record
  - Key or uniqueness constraint
    - E.g., every course record must have a unique value for Course_number
- Controlling redundancy
  - Example
  - Problems
  - Data normalization

Advantages (cont.)

- Providing storage structures and search techniques for efficient query processing
  - Indexes
  - Buffering and caching
  - Query processing and optimization
- Representing complex relationships among data

What are problems with traditional file processing?

- each person controls their own file
  - Difficult to share data
  - Harder to maintain
  - Duplicate data
  - More prone to errors
  - Wasted space
  - Etc.

When Not to Use a DBMS

- More desirable to use regular files for:
  - Simple, well-defined database applications not expected to change at all
  - Stringent, real-time requirements that may not be met because of DBMS overhead
  - Embedded systems with limited storage capacity
  - No multiple-user access to data is required

TO-DOs

- Read textbook
  - Chapter 1, 2,
  - Chapter 3 ER model
- Review Introduction notes under our OneNote

Material→Resources→Introduction

- Next class
  - ER model