CSE 202: Fundamentals of Database Systems
Winter 2023

Welcome!!!

Thanks for registering in this class

Instructor: **Mukesh Mohania (mukesh@iiitd.ac.in)**

Vikram Goyal (vikram@iiitd.ac.in)

Class hours: Monday and Wednesday: 9:30-11:00 AM

Office hours: Monday and Tuesday 11:00 AM-12:30 PM, or by appointment

Teaching Fellow (TF): **Ankita Mittal (ankita@iiitd.ac.in)**

Office hours: Monday, Wednesday, Friday: 1 PM to 2.30 PM

Text and Lecture Material

- A (YouTube)video and/or slides will be provided in advance for self-study.
  - https://www.youtube.com/playlist?list=PL_uaeekrhGzJmfQhBXj5H3pUPhBSOG_fe
  - https://www.youtube.com/playlist?list=PLSE8ODhjZXjaKScG3l0nuOiDTTqpfnWFf

- Each class will run as a tutorial, discussing the video/slides and how the learnt concepts applied in DB application development.

- You may refer the book ‘Database System Concepts' by Silberschatz, Korth, and Sudarshan, McGraw Hill, 7th Ed.
  - **Book slides available online** https://www.db-book.com/db6/slide-dir/

Evaluation Scheme

- Mid-term exam: 20% (as per IIITD exam calendar) [Common for both sections]

- Final exam**:** 40% (as per IIITD exam calendar) [Common for both sections]

- Group (2 students) discussion exam (date will be announced at least one week in advance): 10% [Groups will be formed by the Teaching Fellow]
- Project: 30% [5*6]
- Continuous evaluation
- The following tasks (date should be read as *on or before* midnight of that day) will be evaluated by your tutor in/outside the tutorial class.
  - Project Scope, technical and functional requirements document [Submission: Jan 25, Evaluation: Jan 30]
  - Design of Conceptual Model of your project and converting it to Relational model [Submission: Feb 03, Evaluation: Feb 10]
  - Database schema and indexes creation (with integrity constraints) and data insertion (populate simulated data satisfying the constraints) [Submission: Feb 10, Evaluation: Feb 17]
  - Write and execute at least TEN SQL queries pertaining to your application involving various relational algebraic operations supporting the application features involving database access and manipulation. [Submission: Feb 17, Evaluation: Feb 24]
  - Write and execute at least TWO embedded SQL in your favorite programming language, FOUR OLAP queries and define TWO triggers which checks the database conditions and takes appropriate actions as desired by your application. [Submission: March 24, Evaluation: March 31]
  - Write and execute db transactions (including conflicting ones) and check the effect on your db. [Submission: April 14, Evaluation: April 21]
- **Late submission policy: ONE mark will be deducted per (part of) day late submission. For late evaluation by the tutor, please flag it to the TF immediately.**

Covid Vaccine Management System project example – business requirement (Mid-sem W22)

The GoI plans to build a DB for managing the information about procurement, distribution of vaccines and booking of appointments in hospitals. The dept procures vaccines from various vendors and then distributes them to local health centers for vaccinations to citizens. A citizen books an appointment in a hospital for vaccination.

A citizen may choose the vaccination type (Covishield/Covaxin) and whether it is their first, second or booster shot [based on their vaccination status] while booking an appointment in a hospital. The citizen also chooses the date, time slot and a hospital (Hid, Hname, Hlocation, License number) they want to get vaccinated at. While booking an appointment, a citizen is required to give all personal information including Aadhaar number, name, date of birth, city, and a phone number. It may happen that some citizens may not appear on their scheduled appointment date/time. In this case, they need to rebook and get fresh appointments. Your database design should register the information whether a citizen appeared at the scheduled appointment date/time, and the current vaccination status of a citizen. The system calculates the age based on DOB and categorizes citizens as teenagers, adults and senior citizens for analysis. Only double vaccinated people in the third category can opt for a booster shot while the teenagers are not eligible for vaccination. Citizens may book vaccination appointments for multiple people.

Since there are a sufficient number of vaccines in hospitals, CMO would like to get the status of vaccine inventory and how many folks are covered in their health center each day. If the inventory falls below a certain threshold, the CMO office will send a fresh order for the next supply. To make sure a citizen gets an appointment, the health center checks whether appointments are available before booking the appointment and updates the DB once the citizen gets an appointment.

Vendors must register on the portal to sell vaccines to the GoI by entering details about the vendor, their organisation, details of type and number of vaccines available for sale, and vaccine price quotation. They must also upload a document to prove that they are authorized to sell vaccines. The GoI advertises for vendors via digital and print media.

Vaccine requests by hospitals and applications from vendors are approved by GoI officials. The costs of vaccines sold by GoI to health centers and by health centers to the citizens are fixed. However, the GoI may have a mutual negotiation with vendors for different bulk rates. All information about the sale of vaccines to hospitals and procurement of vaccines from vendors is stored in the system.

Sangeet Talents Management System – business requirements (Mid-sem W21)

- A highly popular 'McM Sangeet company' plans to organize a competition for searching the right talents to train them for creating music albums. The company advertises their recruitment requirements on different channels (both print and digital media) for inviting talents by submitting their personal information, prior experience, and a 2-5 minutes media file, which is of either an audio file (for songs) or a video (for songs and/or playing music instruments), to a given URL in the advertisement. The company writes in the advertisement that the shortlisted candidate will be informed by Phone and/or Email.
- A candidate can submit more than one entry and has an option to provide more than one phone number in the submission, which are unique to him/her.
- After the closing date of entry submission, a panel evaluates all entries and recommends a set of candidates to invite them for the next round. The company maintains the information about the panellist (personal information, industry experience, association with the McM company). A candidate can see the outcome of their application online.
- These shortlisted candidates are then invited to perform a 'Live' show in Mumbai and finally top '*n*' candidates are selected in each album category (i.e. audio and video). These *2n* candidates are called as the member of McM-2020.
- Different music groups (pop, classic, leisure, evergreen, …) are formed to create the music albums (audio/video). Every member belongs to one or more music group which is moderated by a director who himself/herself is a member of the group. Each member has a different role to play in each album.
- Once the album is created and approved by McM Director, its trailer is released online for limited time to the outsiders to give their comments (like and dislike), and all this recorded in the database (album name, date of release, number of visits, number of likes/dislikes, …).
- The McM decides the price of the album after analysing the data collected from its trailer release, and then the album is released to distributors who eventually will sell it online. Note, each distributor may be charged a different price per unit depending on the negotiation between the McM and distributor.
- When a distributor sells an album, the download request comes to the McM site. Thus, the McM company maintains the record for each download (Incoming URL – identifying the distributor, Album#, Date, Download Status – success/failure) so that they can track the number of downloads for raising the invoice.

More about your Project ---

- Project team: Make your own **team of two students** from your tutorial class. (Tutorial class will be announced in this week)

- Inform to the Teaching Fellow about your team member names and Tutorial#.

- Your team is required to develop an e2e db application, primary focus on the design of a back-end db of one of the below applications that require extensive use of data entities selection and relationship between them, modeling of these data entities, relationships and constraints, populating the fictitious data in data tables, database access and data manipulation.

- Each team member contribution should be clearly defined and communicated to your TA.

- The outcome will be judged by the basic concept followed through and implemented in the project. The TA, during the evaluation stage, will ask the questions to each team member how and why/why-not 'XXX' has been considered for design and/or implemented?

- Your team need to choose one of the following project.
  - Design of an online retail store (like Big Bazaar)
  - Design of a cab booking system (like Ola)
  - Design of an pharmacy store (like Apollo pharmacy)
  - Design of a dairy product Company (like Amul)

Grade Distribution

- Relative grading
- Rough distribution
  - **A(+), A and A(-):** 10-20%
  - B and B(-): 20-40%
  - C and C(-): 20-40%
  - D and **F**: 5-10%

Today's task

- Download mysql (www.mysql.com) and start playing to get yourself familiarized.
- Help: https://www.youtube.com/watch?v=6dC0xjdIPZ0

CSE 202
DBMS
Introduction
Lecture 1

Acknowledgement: Following Slides taken from Mohammad Hammoud's (from CMU, Qatar Campus) presentation with minor edits

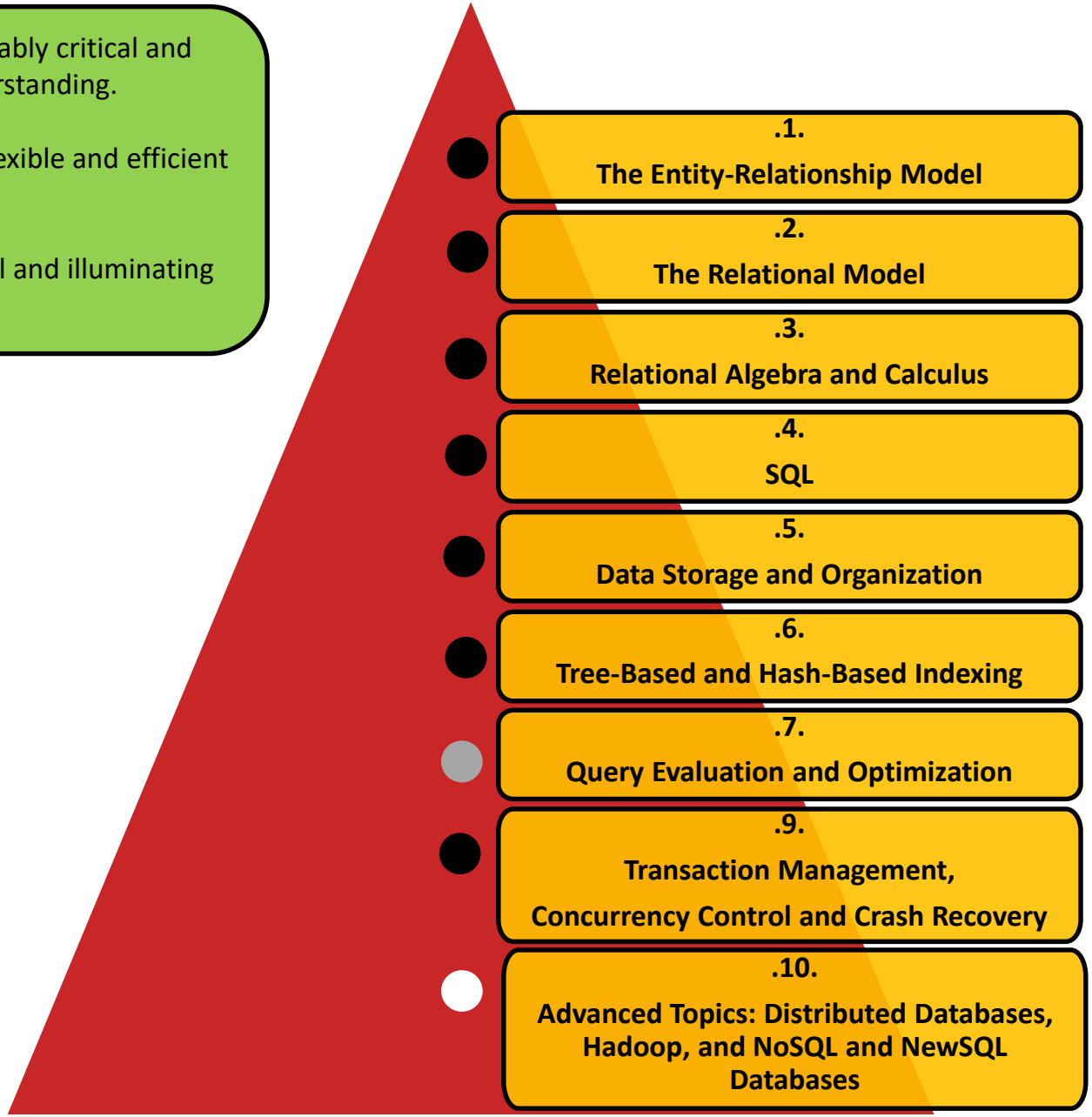
Why Studying Databases?

- Data is *everywhere* and is *critical* to our lives
- Data need to be recorded, maintained, accessed, shared, mined and manipulated *correctly*, *securely*, *efficiently* and *effectively*
- Database management systems (DBMSs) are indispensable software for achieving such goals
- The principles and practices of DBMSs are now an integral part of computer science curricula
  - They encompass OS, languages, theory, AI, multimedia, and logic, among others

As such, the study of database systems can prove to be richly rewarding in more ways than one!

List of Topics

- **Considered:** a reasonably critical and comprehensive understanding.
- **Thoughtful:** fluent, flexible and efficient understanding.
- **Masterful:** a powerful and illuminating understanding.



A Motivating Scenario

- Dept of Education, NCT Govt, has a “large” collection of data (say 500GB) on employees, students, universities, research centers, etc.,
- This data is accessed *concurrently* by several people
- Queries on data must be answered *quickly*
- Changes made to the data by different users must be applied *consistently*
- Access to certain parts of data (e.g., salaries) must be *restricted*
- This data should survive system crashes/failures

Performance (Concurrency Control)

Performance (Response Time)

Correctness (Consistency)

Correctness (Security)

Correctness (Durability and Atomicity)

Managing Data using File Systems

- **What about managing data using local file systems?**
  - Files of fixed-length and variable-length records as well as formats
  - Main memory vs. disk
  - Computer systems with 32-bit addressing vs. 64-bit addressing schemes
  - Special programs (e.g., C++ and Python programs) for answering user questions
  - Special measures to maintain atomicity
  - Special measures to maintain consistency of data
  - Special measures to maintain data isolation
  - Special measures to offer software and hardware fault-tolerance
  - Special measures to enforce security policies in which different users are granted different permissions to access diverse subsets of data

This becomes tedious and inconvenient, especially at large-scale, with evolving/new user queries and higher probability of failures!

Database Management Systems

- A special software is accordingly needed to make the preceding tasks easier
- This software is known as Database Management System (DBMS)
- DBMSs provide automatic:
  - Data independence
  - Efficient data access
  - Data integrity and security
  - Data administration
  - Concurrent access and crash recovery
  - Reduced application development and tuning time

Some Definitions

- **A database is a collection of data which describes one or many real-world enterprises**
  - E.g., a university database might contain information about entities like students and courses, and relationships like a student enrollment in a course
  -
- **A DBMS is a software package designed to store and manage databases**
  - E.g., DB2, Oracle, MS SQL Server, MySQL and Postgres
- **A database system = (Big) Data + DBMS + Application Programs**

Data Models

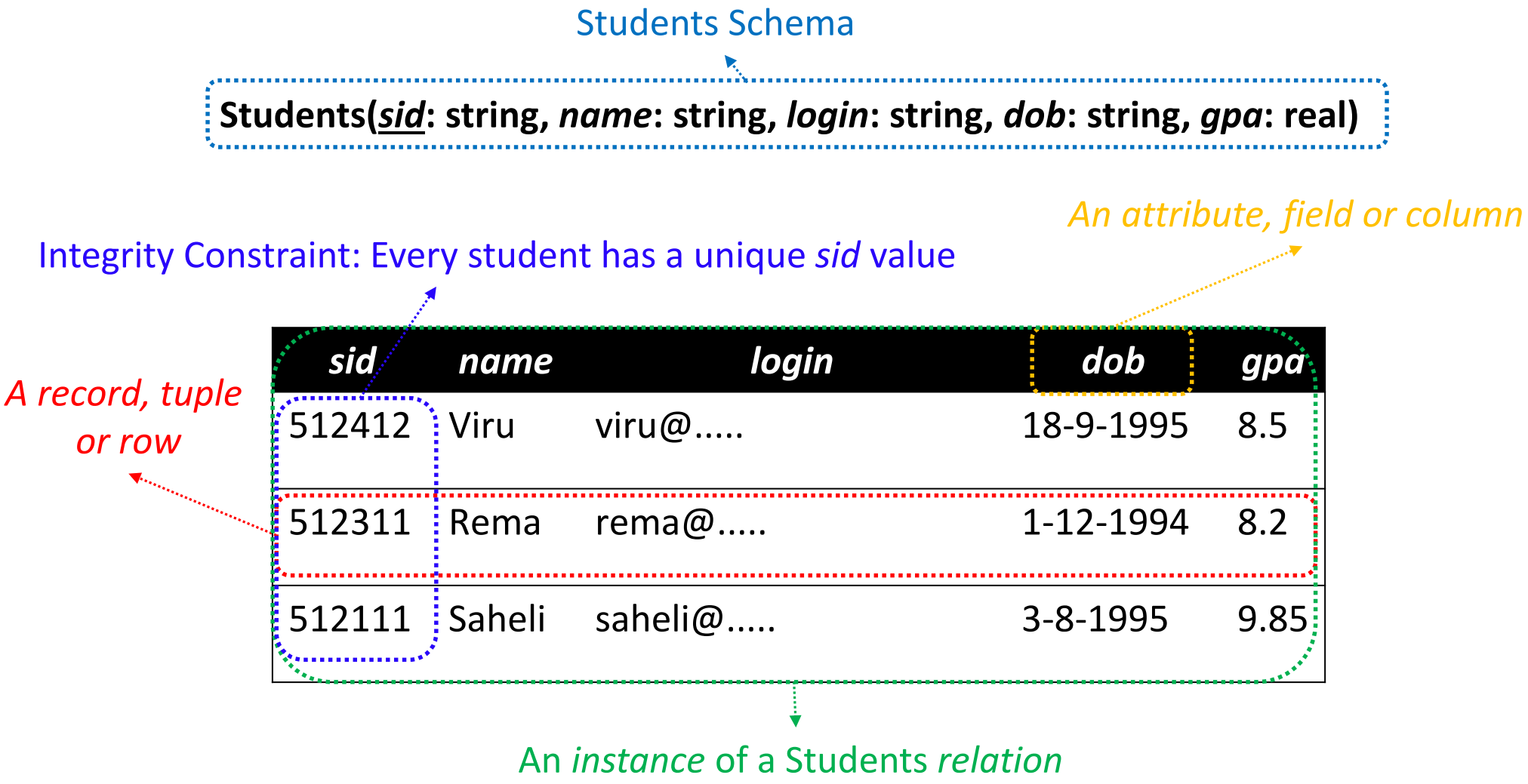
- The user of a DBMS is ultimately concerned with some real-world enterprises (e.g., a University)
- The data to be stored and managed by a DBMS *describes* various aspects of the enterprises
  - E.g., The data in a university database describes students, faculty and courses entities and the relationships among them
- A data model is a collection of high-level data description constructs that hide many low-level storage details
- A widely used data model called the entity-relationship (ER) model allows users to pictorially denote entities and the relationships among them

The Relational Model

- The relational model of data is one of the most widely used models today
- The central data description construct in the relational model is the relation
- A relation is basically a table (or a set) with rows (or records or tuples) and columns (or fields or attributes)
- Every relation has a schema, which describes the columns of a relation
- Conditions that records in a relation must satisfy can be specified
  - These are referred to as integrity constraints

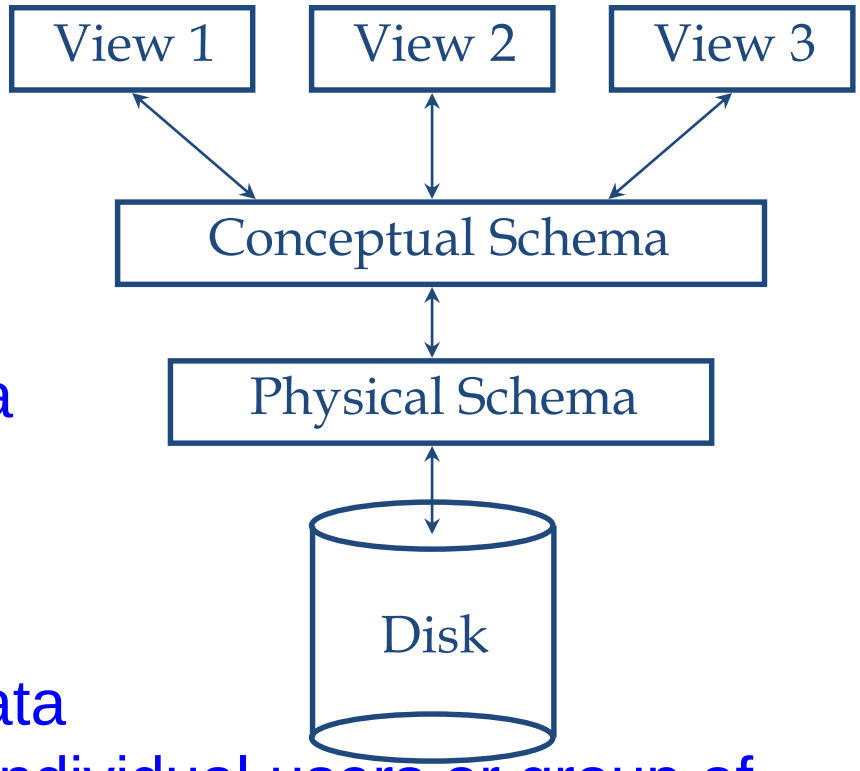
The Relational Model: An Example

- **Let us consider the student entity in a university database**



Levels of Abstraction

- The data in a DBMS is described at three levels of abstraction, the conceptual (or logical), physical and external schemas
- The conceptual schema describes data in terms of a specific data model (e.g., the relational model of data)
- The physical schema specifies how data described in the conceptual schema are stored on secondary storage devices
- The external schema (or views) allow data access to be customized at the level of individual users or group of users (views can be 1 or many)



Views

- **A view is conceptually a relation**
- **Records in a view are computed as needed and usually not stored in a DBMS**
- **Example: University Database**

Conceptual Schema	Physical Schema	External Schema (View)
• Students(sid: string, name: string, login: string, dob: string, gpa:real) • Courses(cid: string, cname:string, credits:integer) • Enrolled(sid:string, cid:string, grade:string)	• Relations stored as heap files • Index on first column of Students	Students can be allowed to find out course enrollments: • Course_info(cid: string, enrollment: integer)

Can be computed from the relations in the conceptual schema (so as to avoid data redundancy and inconsistency).

Data Independence

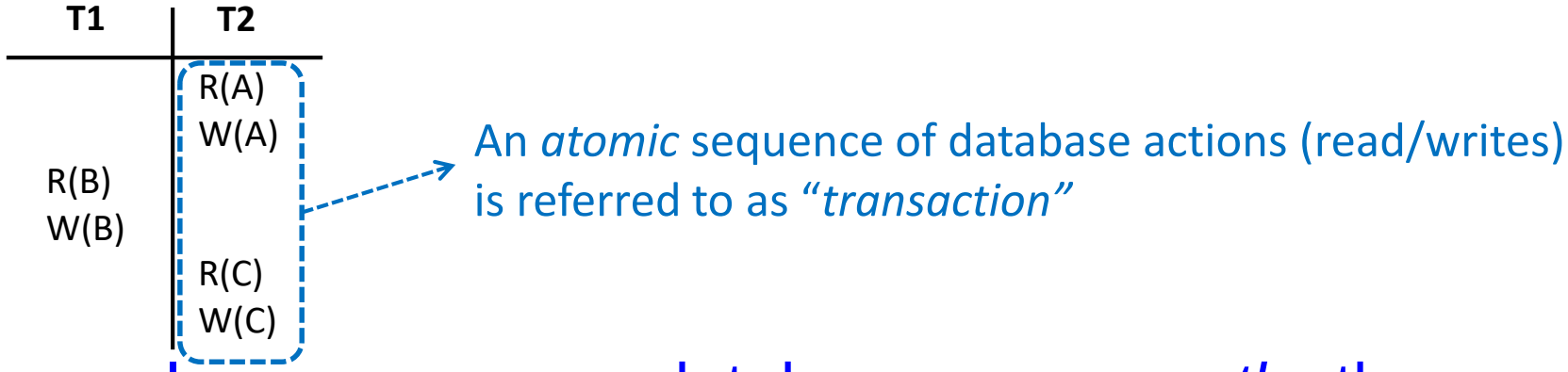
- One of the most important benefits of using a DBMS is data independence
- With data independence, application programs are insulated from how data are structured and stored
- Data independence entails two properties:
  - Logical data independence: users are shielded from changes in the conceptual schema (e.g., add/drop a column in a table)
  - Physical data independence: users are shielded from changes in the physical schema (e.g., add index or change record order)

Queries in a DBMS

- The ease with which information can be queried from a database determines its value to users
- A DBMS provides a specialized language, called the query language, in which queries can be posed
- The relational model supports powerful query languages
  - Relational calculus: a formal language based on mathematical logic
  - Relational algebra: a formal language based on a collection of operators (e.g., selection and projection) for manipulating relations
  - Structured Query Language (SQL):
    - Builds upon relational calculus and algebra
    - Allows creating, manipulating and querying relational databases
    - Can be embedded within a host language (e.g., Java)

Concurrent Execution and Transactions

- An important task of a DBMS is to *schedule* concurrent accesses to data so as to improve performance



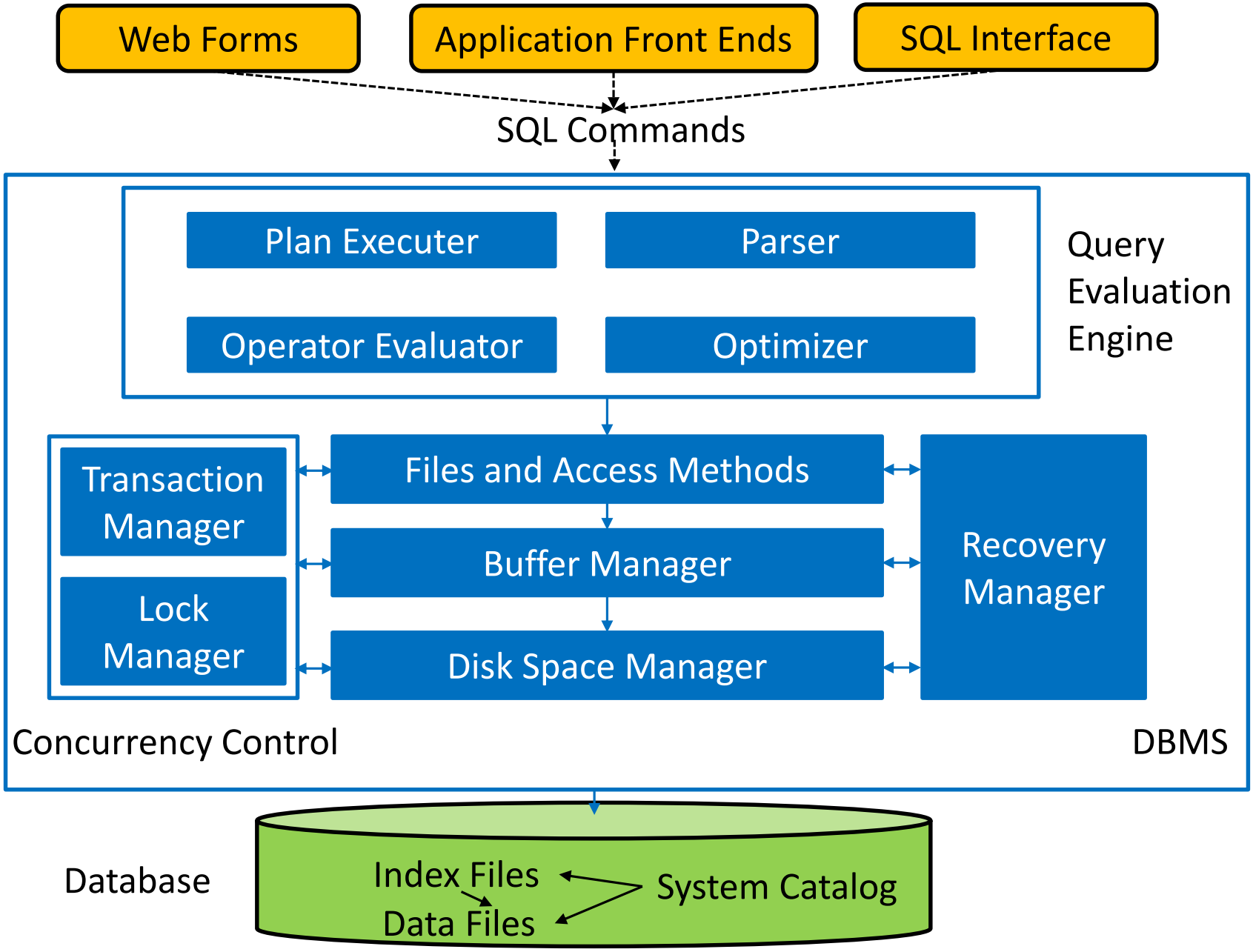
- When several users access a database *concurrently*, the DBMS must order their requests carefully to avoid conflicts
  - E.g., A check might be cleared while account balance is being computed!

- DBMS ensures that conflicts do not arise via using a locking protocol
  - Shared vs. Exclusive locks

Ensuring Atomicity

- Transactions can be interrupted before running to completion for a variety of reasons (e.g., due to a system crash)
- DBMS ensures atomicity (all-or-nothing property) even if a crash occurs in the middle of a transaction
- This is achieved via maintaining a log (i.e., history) of all writes to the database
  - *Before* a change is made to the database, the corresponding log entry is forced to a safe location (this protocol is called Write-Ahead Log or WAL)
  - After a crash, the effects of partially executed transactions are *undone* using the log

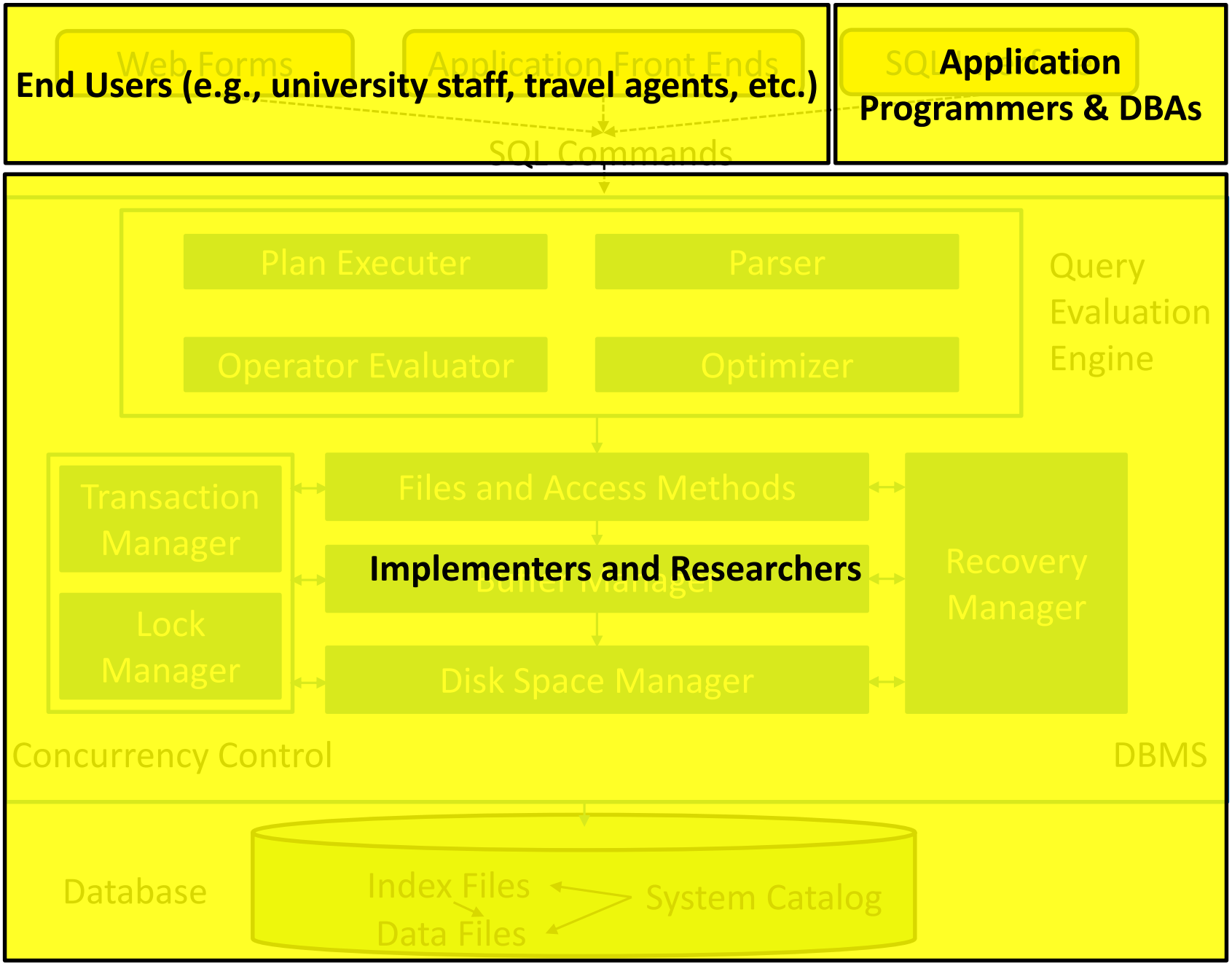
The Architecture of a Relational DBMS



People Who Work With Databases

- **There are five classes of people associated with databases:**
  1. End users
     - Store and use data in DBMSs
     - Usually not computer professionals
  2. Application programmers
     - Develop applications that facilitate the usage of DBMSs for end-users
     - Computer professionals who know how to leverage host languages, query languages and DBMSs altogether
  3. Database Administrators (DBAs)
     - Design the conceptual and physical schemas
     - Ensure security and authorization
     - Ensure data availability and recovery from failures
     - Perform database tuning
  4. Implementers
     - Build DBMS software for vendors like IBM and Oracle
     - Computer professionals who know how to build DBMS internals
  5. Researchers
     - Innovate new ideas which address evolving and new challenges/problems

The Architecture of a Relational DBMS



Summary

- We live in a world of data
- The explosion of data is occurring along the 3Vs dimensions
- DBMSs are needed for ensuring logical and physical data independence and ACID properties, among others
- The data in a DBMS is described at three levels of abstraction
- A DBMS typically has a layered architecture
- Studying DBMSs is one of the broadest and most exciting areas in computer science!
- This course provides an in-depth treatment of DBMSs with an emphasis on how to *design*, *create*, *refine*, *use* and *build* DBMSs and real-world enterprise databases
- Various classes of people who work with databases hold responsible jobs and are well-paid!