Lecture 1:
Course Introduction

Importance of Database Systems

- Banking
- Healthcare
- Airlines
- E-Commerce

Why take this course?

- Curiosity
- Scalability
- Efficiency
- Versatility

Why take this course?

- Storage Management
- Query Optimization
- Index Structures
- SIMD Instructions

Course Overview

Course Objectives

- Learn about building a database system from scratch.
- Become proficient in systems programming.
- Understand the impact of hardware trends on software design.

Course Topics

- This course focuses on the internals of a database system.
  - Relational Databases
  - Storage Management
  - Index Structures
  - Query Execution

Next Course

- Course on advanced database implementation
  - Logging and Recovery
  - Concurrency Control
  - Query Optimization
  - Potpourri of advanced topics
- This course is a pre-requisite for that course

Expected Background

- Should have taken an introductory course on computer systems.
- All programming assignments will be in C++.
  - Programming assignment #1 will help get you caught up with C++.
  - If you have not encountered C++ before, need to put in extra effort.
  - Use a large language model like ChatGPT for assistance.
  - Relevant parts of C++ will be briefly covered in this course.

Course Logistics

- Course Website (link on Canvas)
- Discussion Tool: Ed (link on Canvas)
- Grading Tool: Gradescope (link on Canvas)
- In-Class Quiz Tool: Point Solutions (link on Canvas)

Course Rubric

- Exams (50%)
- Programming Assignments (20%)
- Exercise Sheets (15%)
- In-Class Quizzes (15%)

4420 vs 6422

- Advanced lectures on learned index etc.
- Paper reading and questions based on those papers

Course Policies

- Programming assignments & exercise sheets must be own work.
  - Not group assignments.
  - You may not copy source code from other people or the web.
  - Plagiarism will not be tolerated.
  - We will follow the late submission policy listed on Canvas.
- Academic Honesty
  - Refer to Georgia Tech Academic Honor Code.
  - If you are not sure, ask me.

Textbooks for Reference

- Silberschatz, Korth, & Sudarshan:
  - Database System Concepts. McGraw Hill, 2020.
- Hector Garcia-Molina, Jeff Ullman, and Jennifer Widom:
  - Database Systems: The Complete Book. Prentice-Hall, 2008.

Intro Sheet

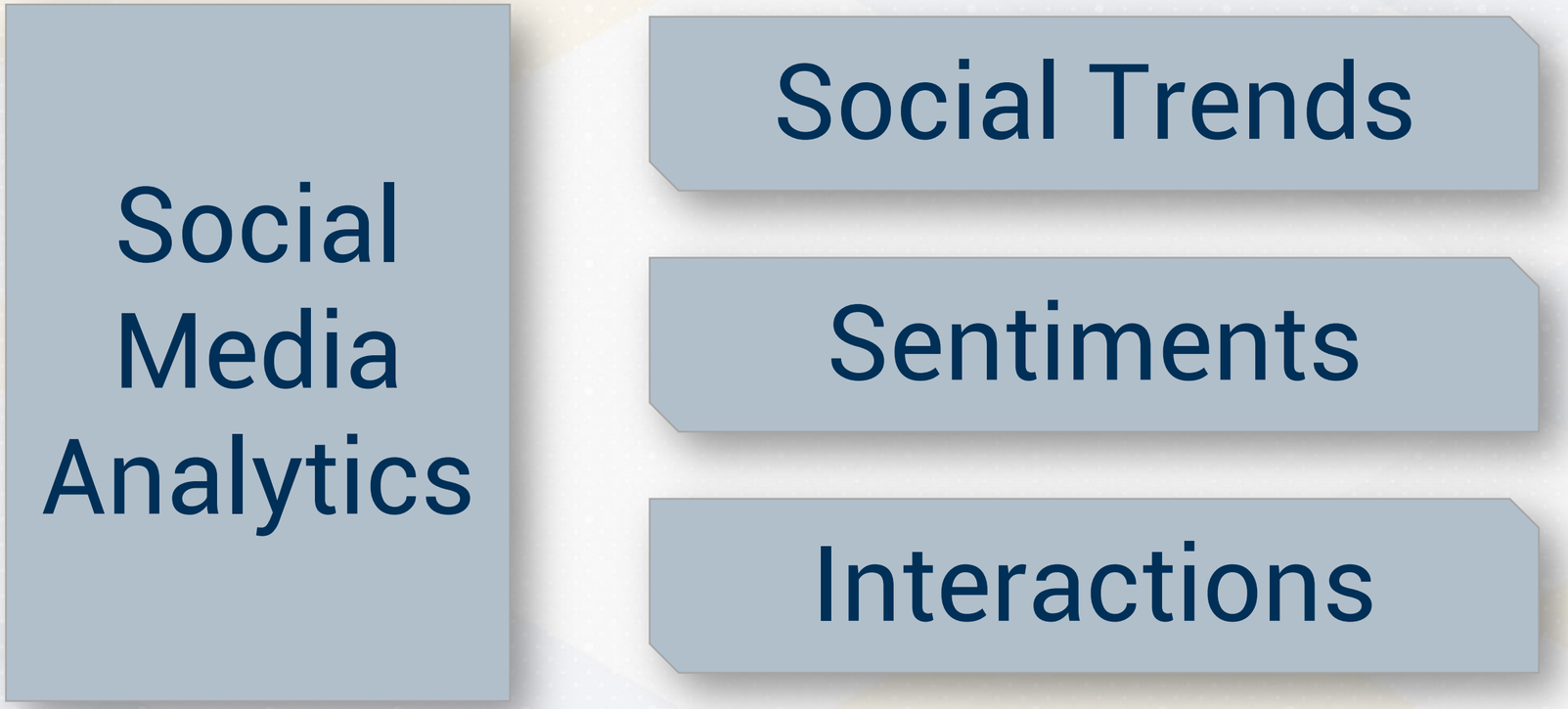
- Upload a one-page PDF with your details on Gradescope.
  - Picture (ideally 2x2 inches of face).
  - Name, interests, and other details mentioned on Gradescope.
- Purpose of this sheet
  - Help me know more about your background for tailoring the course.
  - Recognize you in class.

In-Person Office Hours

- Sign up for a ten-minute slot in the sign-up sheet (link on Canvas)
- Teaching assistants will guide you with assignments & sheets.

Motivating Application

Social Media Analytics Application



Social Media Analytics Application

Users Text File

Users.txt

UserName,	Location
Timothée Chalamet,	Paris
Lana Condor,	Los Angeles
Liu Yifei,	Beijing
Burna Boy,	Lagos
Kriti Sanon,	Mumbai

Posts Text File

Posts.txt

PostID,	UserName,	PostContent
1001,	Timothée Chalamet,	Excited to start filming my new movie!
1002,	Lana Condor,	Had a great time at the beach today! 🌊☀️
1003,	Liu Yifei,	Enjoying the scenery in Beijing! 🏞️
1004,	Burna Boy,	Live performance tonight in Lagos! 🎤🎶
1005,	Kriti Sanon,	Loving the vibrant energy of Mumbai! 🌆

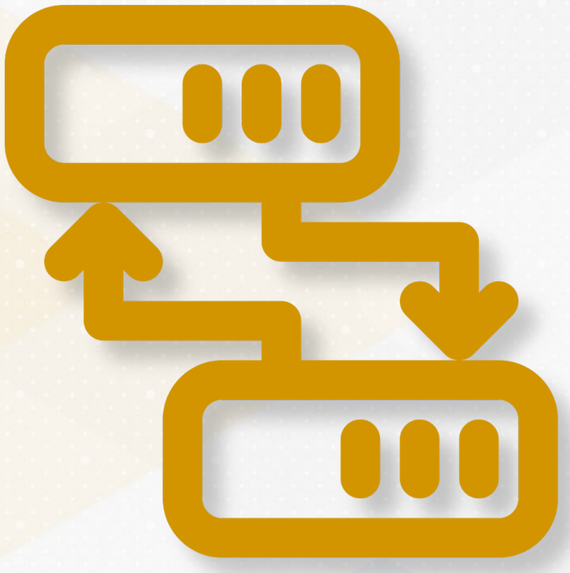
Interactions Text File

Interactions.txt

PostID,	UserName,	Reaction Type,	Comment
1001,	Lana Condor,	Comment,	Love it!
1002,	Liu Yifei,	Like,	-
1003,	Burna Boy	Like,	-
1004,	Kriti Sanon	Comment,	Wish I could be there!

Flat-File Database

Limitation #1: Data Redundancy



PostID,	UserName,	PostContent
1001,	Timothée Chalamet,	Excited to start filming my new movie!
1006,	Timothée Chalamet,	Exploring the streets of Paris! 🇫🇷
1007,	Timothée Lamet,	Just wrapped up a day of filming 🎬
1008,	Timothée Chalamet,	Any book recommendations?

Limitation #2: Slow Operations

UserName	Location
Timothée Chalamet,	Paris
~~Lana Condor,~~	~~Los Angeles~~
Liu Yifei,	Beijing
Burna Boy,	Lagos
Kriti Sanon,	Mumbai

Limitation #3: Slow Queries

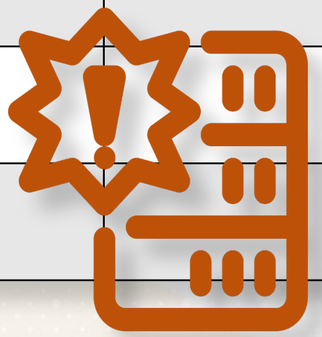
UserName,	Location
Timothée Chalamet,	Paris
Lana Condor,	Los Angeles
Liu Yifei,	Beijing
Burna Boy,	Lagos
Kriti Sanon,	Mumbai

Limitation #4: Concurrent Updates

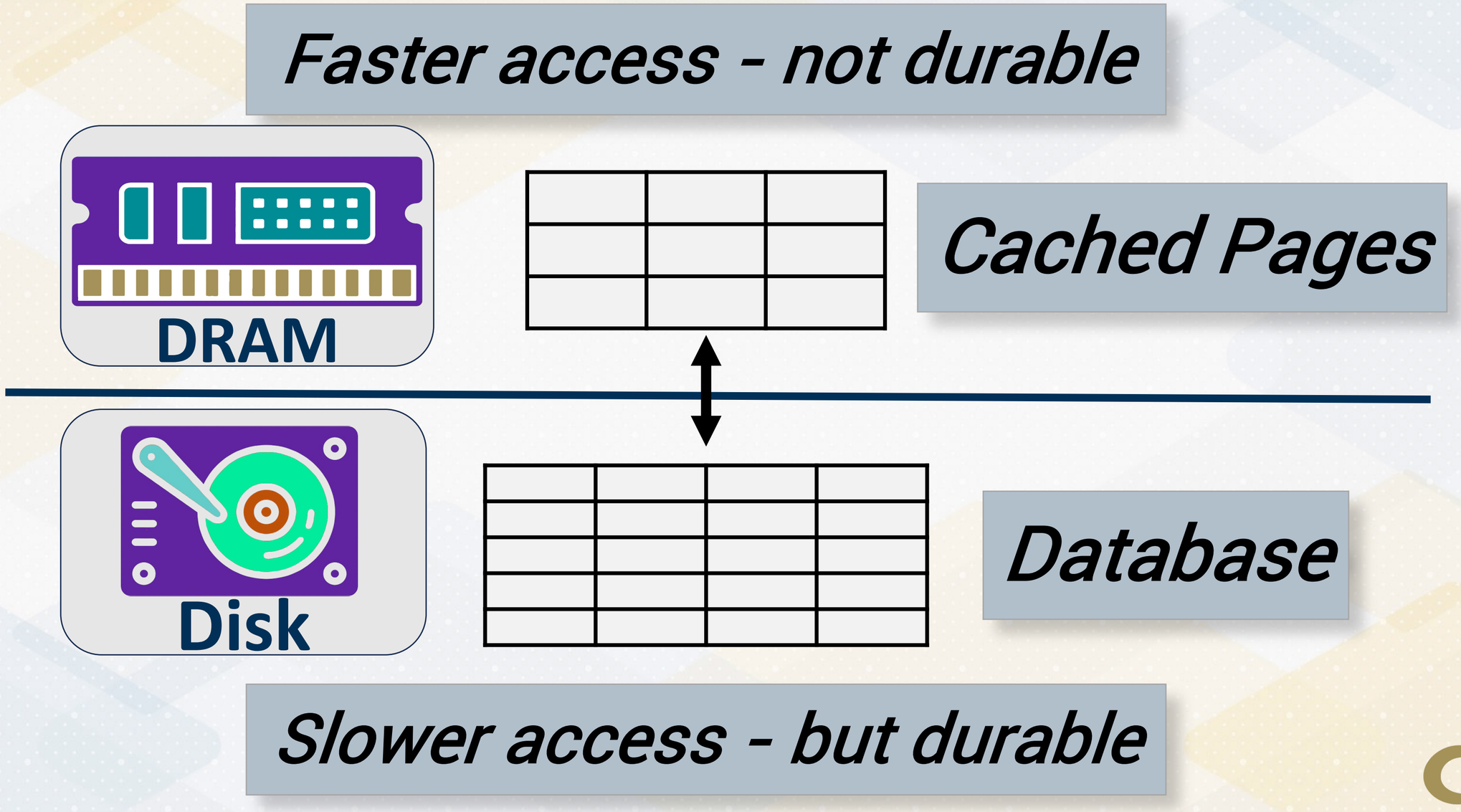
Limitation #5: Handling Disk Failure

Users.txt

UserName,	Location,	Country
Timothée Chalamet,	Paris,	France
Lana Condor,	Los Angeles,	USA
Liu Yifei,	Beijing,	China
Burna Boy,	Lagos	
Kriti Sanon,	Mumbai	



Limitation #6: Memory Management



Limitation #7: Usability

Custom Code

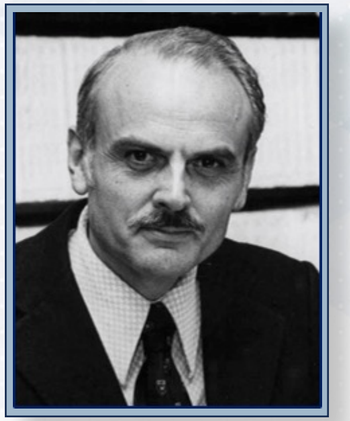
Comments Query Code

```
def get_comments_by_user(file_path, user_name):
    comments = []
    with open(file_path, 'r') as file:
        for line in file:
            post_id, user, reaction_type, comment_text = line.strip().split(', ')
            if user == user_name and reaction_type == "Comment":
                comments.append((post_id, comment_text))
    return comments
```

Relational Database

Relational Database

Ted Codd (1970)



- Scientist at IBM
- Simplify Data Management
- Organize Data as Tables

A Relational Model of Data for Large Shared Data Banks

E. F. CODD
IBM Research Laboratory, San Jose, California

Future users of large data banks must be protected from having to know how the data is organized in the machine (the internal representation). A prompting service which supplies such information is not a satisfactory solution. Activities of users at terminals and most application programs should remain unaffected when the internal representation of data is changed and even when some aspects of the external representation are changed. Changes in data representation will often be needed as a result of changes in query, update, and report traffic and natural growth in the types of stored information.

Existing noninferential, formatted data systems provide users with tree-structured files or slightly more general network models of the data. In Section 1, inadequacies of these models are discussed. A model based on *n*-ary relations, a normal form for data base relations, and the concept of a universal data sublanguage are introduced. In Section 2, certain operations on relations (other than logical inference) are discussed and applied to the problems of redundancy and consistency in the user's model.

KEY WORDS AND PHRASES: data bank, data base, data structure, data organization, hierarchies of data, networks of data, relations, derivability, redundancy, consistency, composition, join, retrieval language, predicate calculus, security, data integrity
CR CATEGORIES: 3.70, 3.73, 3.75, 4.20, 4.22, 4.29

The relational view (or model) of data described in Section 1 appears to be superior in several respects to the graph or network model [3, 4] presently in vogue for noninferential systems. It provides a means of describing data with its natural structure only—that is, without superimposing any additional structure for machine representation purposes. Accordingly, it provides a basis for a high level data language which will yield maximal independence between programs on the one hand and machine representation and organization of data on the other.

A further advantage of the relational view is that it forms a sound basis for treating derivability, redundancy, and consistency of relations—these are discussed in Section 2. The network model, on the other hand, has spawned a number of confusions, not the least of which is mistaking the derivation of connections for the derivation of relations (see remarks in Section 2 on the "connection trap").

Finally, the relational view permits a clearer evaluation of the scope and logical limitations of present formatted data systems, and also the relative merits (from a logical standpoint) of competing representations of data within a single system. Examples of this clearer perspective are cited in various parts of this paper. Implementations of systems to support the relational model are not discussed.

1.2. Data Dependencies in Present Systems

The provision of data description tables in recently developed information systems represents a major advance toward the goal of data independence [5, 6, 7]. Such tables facilitate changing certain characteristics of the data representation stored in a data bank. However, the variety of data representation characteristics which can be changed *without logically impairing some application programs* is still quite limited. Further, the model of data with which users interact is still cluttered with representational prop-

June 1970
Communications
of the ACM

Relational Database

Column / Attribute

Row/ Tuple

UserName	Location
Timothée Chalamet	Paris
Liu Yifei	Beijing
Burna Boy	Lagos
Kriti Sanon	Mumbai

Relational Database

UserID	UserName	Location
1	Timothée Chalamet	Paris
2	Lana Condor	Los Angeles
3	Liu Yifei	Beijing
4	Burna Boy	Lagos
5	Kriti Sanon	Mumbai

Users

PostID	UserID	PostContent
1001	1	Excited to start filming my new movie!
1002	2	Had a great time at the beach today! 🌊☀️
1003	3	Enjoying the scenery in Beijing! 🏞️
1004	4	Live performance tonight in Lagos! 🎤🎶
1005	5	Loving the vibrant energy of Mumbai! 🌆

Posts

PostID	UserID	ReactionType	Content
1001	2	Comment	Love it!
1002	3	Like	-
1003	4	Like	-
1004	5	Comment	Wish I could be there!

Interactions

Relational Database

Mathematic Set Theory

Data Sets Relationship

Efficient Data Set Links

Students	Grades
Alice	B
Bob	A
Charlie	C

R={(Alice,B),(Bob,A),(Charlie,C)}

Relational Database

Physical

Storage Formats

Indexing Data Structures

List of Tables

Logical

Relational Database

Logical Database Design

Simple Query Language for Complex Data Manipulation

Physical Database Design

Optimize Indexing for Storage Hardware

Logical Database Design: Primary Key

UserID	UserName	Location
1	Timothée Chalamet	Paris
2	Lana Condor	Los Angeles
3	Liu Yifei	Beijing
4	Burna Boy	Lagos
5	Kriti Sanon	Mumbai

Logical Database Design: Foreign Key

UserID	UserName	Location
1	Timothée Chalamet	Paris
2	Lana Condor	Los Angeles
3	Liu Yifei	Beijing
4	Burna Boy	Lagos
5	Kriti Sanon	Mumbai

Referential Data Integrity

PostID	UserID	PostContent
1001	1	Excited to start filming my new movie!
1002	2	Had a great time at the beach today! 🌊☀️
1003	3	Enjoying the scenery in Beijing! 🏞️
1004	4	Live performance tonight in Lagos! 🎤🎶
1005	5	Loving the vibrant energy of Mumbai! 🌆

Relational Database Benefits

Benefit #1: No Data Redundancy

UserID	UserName	Location
1	Sir Timothée Chalamet	Paris

PostID	UserID	PostContent
1001	1	Excited to start filming my new movie!
1006	1	Exploring the streets of Paris!
1007	1	Just wrapped up a day of filming
1008	1	Any book recommendations?

Benefit #2: Fast Operations

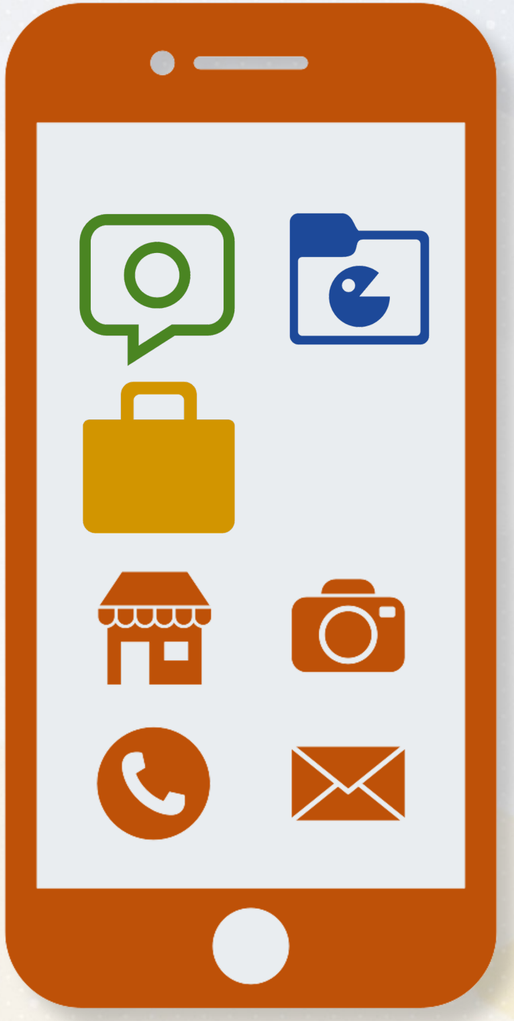
- Efficient Data Deletion
- User (Tuple) Removal
- Fast Deletion

Benefit #3: Fast Queries

Index Database

Apps in labeled folders

Location-based index

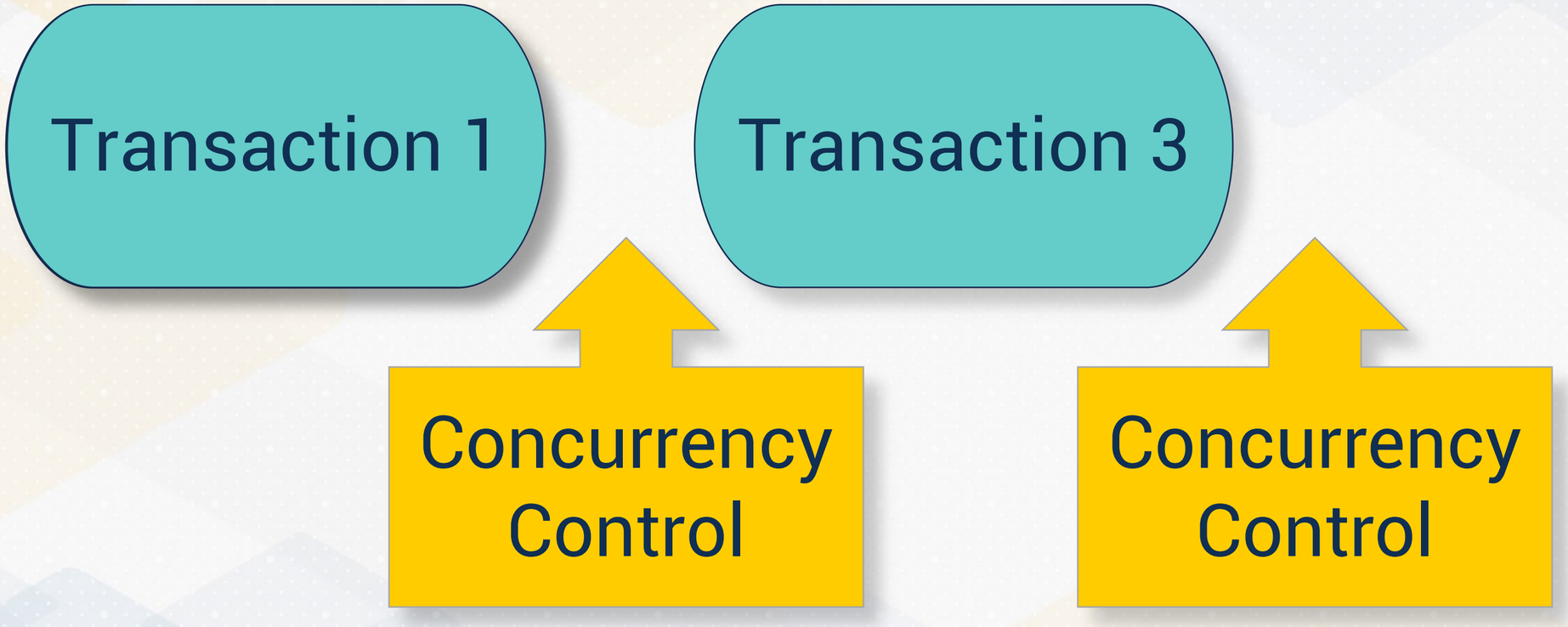


Benefit #3: Fast Queries

```
SELECT *
FROM Users
WHERE LOCATION = 'Mumbai';
```

UserName,	Location
Timothée Chalamet,	Paris
Lana Condor,	Los Angeles
Liu Yifei,	Beijing
Burna Boy,	Lagos
Kriti Sanon,	Mumbai

Benefit #4: Concurrent Updates



USER 1

Timothée Chalamet,
Paris

USER 2

Xavier Laurent,
New York

USER 2

Timothée Chalamet,
Paris

Benefit #5: Handling Failures

Atomic Transactions → "All or Nothing"

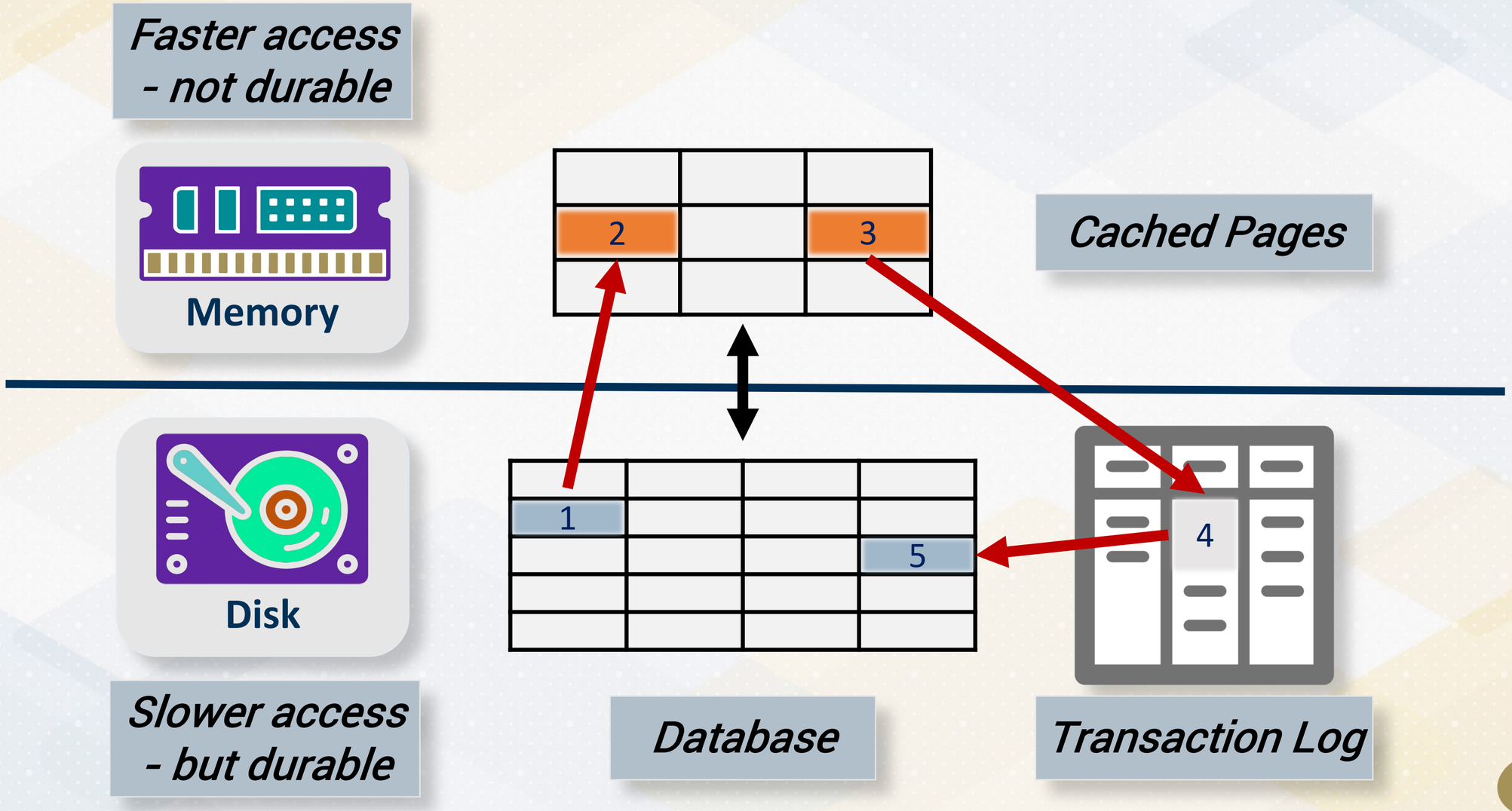
UserName	Location	Country
Timothée Chalamet	Paris	France
Lana Condor	Los Angeles	USA
Liu Lifei	Beijing	China
Burna Boy	Lagos	
Kriti Sanon	Mumbai	

Reversion

Benefit #5: Handling Failures

UserName	Location	Country
Timothée Chalamet	Paris	France
Lana Condor	Los Angeles	USA
Liu Lifei	Beijing	China
Burna Boy	Lagos	
Kriti Sanon	Mumbai	

Benefit #6: Memory Management



Benefit #7: Usability

UserName	Location
Timothée Chalamet	Paris
Lana Condor	Los Angeles
Liu Yifei	Beijing
Burna Boy	Lagos
Kriti Sanon	Mumbai

SQL = Declarative

Python, C++ = Imperative

Conclusion

- Illustrative Social Media Analytics
- Limitations of a Flat-file Database System
- Benefits of a Relational Database System