CSE462/562: Database Systems (Spring 22)

Lecture 1: Introduction & Course Logistics

2/1/2022

University at Buffalo
Department of Computer Science and Engineering
School of Engineering and Applied Sciences

Logistics

- Knox 110, Tuesday and Thursday 11:00 am – 12:20 pm.
- Instructor: Zhuoyue Zhao, zzhao35 [at] buffalo [dot] edu
  - Office hour: Monday 1-3 pm, Davis 113A; Wednesday 2-4 pm, David 338A.
- Teaching assistants:
  - Meng Ding, mengding [at] buffalo [dot] edu
    - Office hour: Tuesday, 2-4 pm, zoom or Davis 113A on request.
  - Yunnan Yu, yunnanyu [at] buffalo [dot] edu, office hour
    - Office hour: Thursday, 2-4 pm, zoom or Davis 113A on request.
- No office hour in week 1.
- See course website and Piazza announcements for update on office hours.

- Find more on course website: https://cse.buffalo.edu/~zzhao35/teaching/cse562_spring22/

Logistics

- We mainly use Piazza for communication:
  - https://piazza.com/class/kxf79wjbzz52zw
  - Please prefer using Piazza over sending email
    - Emails may end up in junk box or getting buried!
- When you post something privately to the teaching staff:
  - post to all TAs and the instructor
- Try to post under the correct folder
  - See announcements for details.

Logistics

- Important Dates:
  - Add/drop deadline: 2/7/2022
  - Mid-term exam: 3/17/2022 (tentative)
  - Last day to resign from the course: 4/22/2022
  - Final exam: 5/17/2022

- Exams are open-book. Paper-copy of slides, lecture notes, homework & solution only. No electronic devices except for a calculator!
- Post privately under exam_conflicts to all TAs and the instructor if you have known conflicts.
  - No make-up exams unless you notify us early (no later than 2/28).
  - If you have any medical emergency, please notify us as soon as possible on Piazza.

Logistics

- Grading
  - Mid-term exam: 20%
  - Final exam: 20%
  - Projects: 60%

- Projects
  - Building a mini-DB system Taco-DB in C++
    - A teaching-oriented framework currently used here at UB and PSU
  - 1 warm-up project + 4 main projects
  - More on project at the end of this lecture
  - No late submission accepted.

Prerequisites for the course project

- It's best if you know C/C++ and have some experience with large projects
  - You're likely to complete the projects with reasonable effort.
- If you at least know some static-typed object-oriented language
  - Java, Scala, …
  - Chances are you'll need to spend (maybe significant) extra efforts.
- If none of the above apply,
  - you'll have a hard time to catch up.

- We will introduce the project and C++11 on Thursday, 2/3.

Academic Integrity Policy

- Academic integrity is critical to the learning process. It is your responsibility to understand and follow all the departmental and university academic integrity policies.
- Zero tolerance towards academic integrity violations, which includes but are not limited to
  - Sharing/copying code in projects or
  - Plagiarizing write-ups
  - Cheating in exam
  - Making project code publicly available or available to any current or future students
  - Submitting code repository that does not belong to you
- Any AI violation will result in an F grade and will be reported to the Office of Academic Integrity
  - Not for an honest mistake if it does not give you any undue advantage

What dose this course cover?

- The design and implementation of DataBase Management System (DBMS)
  - Relational DBMS (RDBMS) as a case study
  - Focus on principles and techniques generally applicable in Data Management

- Note, this course is not about:
  - Database design
  - The relational model and the SQL language
  - Database (web/mobile) application development
  - Programming/data structure/algorithm analysis/discrete math…

Why should I care about DBMS internals?

- > 60 billion dollar worth industry
  - Many more are directly or indirectly using DBMS products
- Many vendors and products:
  - Relational: MySQL, Oracle DB, Microsoft SQL Server, IBM Db2, PostgreSQL, SQLite…
  - Graph DB and Graph data processing: Neo4j, Virtuoso, GraphLab, Spark GraphX, …
  - Stream Processing: Apache Flink, Spark Streaming, Apache Storm, …
  - Semi-structured DB: MongoDB, CouchBase, DocumentDB, …
  - Distributed database: Google Spanner, Microsoft CosmosDB, …
  - …
- Used by many other research and application areas:
  - Artificial Intelligence/data mining/search engine/social media/fintech/…

Why should I care about DBMS internals?

- Huge demand in industry for those who can
  - query/manipulate data in database efficiently
  - fine-tune the imperfect DBMS/big data processing systems
  - work seamlessly with the data infrastructure team

- An actively researched area that
  - has strong real-life impacts and connection to the industry
  - has many related open engineering and research positions

- The goal of this course:
  - understanding the common problems and solutions in data management
  - gaining hands-on experience with building a complex software system
  - to be helpful in your future industrial/academic career

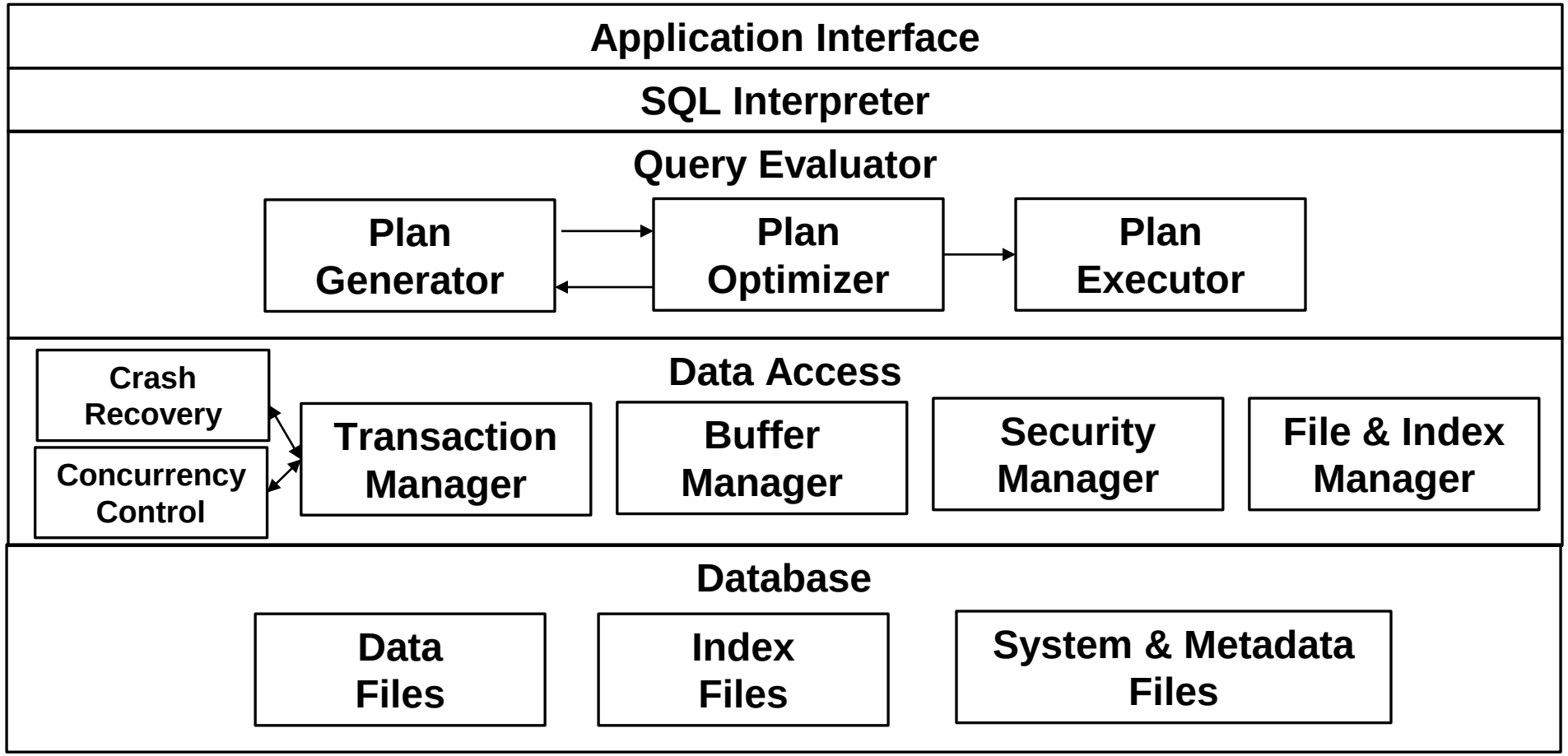
What is a Database?

- Database is
  - a collection of interrelated data
  - often organized in a certain structure for convenient and efficient access

- Databases are found almost everywhere, sometimes unnoticed
  - Business: sales, accounting, human resource, IT support, …
  - Financial industry: banking, credit card, investment platform
  - University: student records, course registration, LMS (e.g., UB Learns), ...
  - Some less obvious examples of databases
    - Software package and configuration DB (e.g., windows registry)
    - Your photo library (e.g., Google Photos)
    - Your personal finance records
    - …

What's a DataBase Management System?

- DataBase Management System (DBMS) is a software system for convenient and efficient data access over databases.

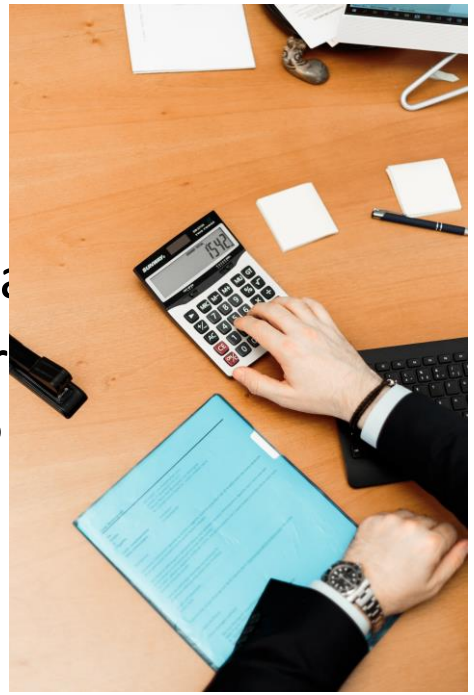


Why using a DataBase Management System?

How to manage a database?

- Suppose I'd like to track my daily spending
- What I can do:
  - Step 1: collect all the receipts

  - Step 2: do some analysis
    - How much did my spend on grocery and fast food in Febura
    - How much could I have saved if I cook by myself in Feburar
    - What about January/last quarter/last year/past five years?

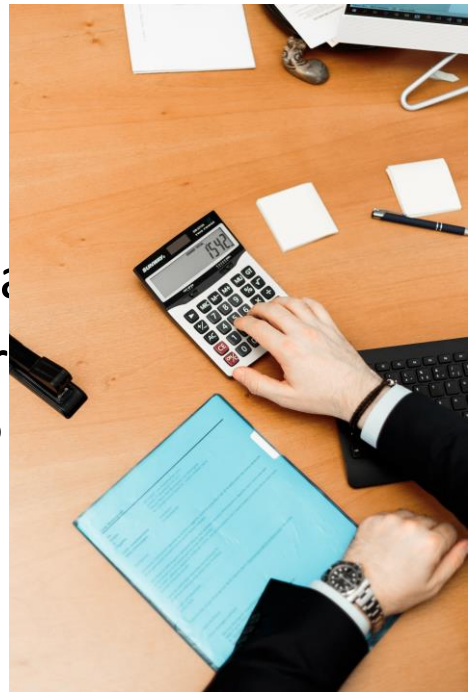


How to manage a database?

- Suppose I'd like to track my daily spending
- What I can do:
  - Step 1: collect all the receipts
  - Step 2: write them down on a notebook

Date	Amount	Description
2/1	$20.21	Grocery
2/2	$10.54	Fast food
2/3	$39.22	Cell phone bill
…		
2/27	$33.00	Clothes

  - Step 3: do some analysis
    - How much did my spend on grocery and fast food in Febura
    - How much could I have saved if I cook by myself in Feburar
    - What about January/last quarter/last year/past five years?



How to manage a database?

- Suppose I'd like to track my daily spending
- What I can do:
  - Step 1: collect all the receipts
  - Step 2: ~~write them down on a notebook~~
    store them in a text file

```
Date   Amount   Description
2/1    $20.21    Grocery
2/2    $10.54   Fast food
2/3    $39.22    Cell phone bill
…
2/27   $33.00    Clothes
```

  - Step 3: do some analysis
    - How much did my spend on grocer
    - How much could I have saved if I co
    - What about January/last quarter/la

```
f = open('myspend_feb_22.txt', 'r')
grocery = 0
fast_food = 0
for line in f:
    date, amount, desc = line.split(' ')
    if desc == 'Fast food':
        fast_food += eval(amount)
    elif desc == 'Grocery':
        grocery += eval(amount)
……
```

How to manage a database?

- Suppose I'd like to track my daily spending
- What I can do:
  - Step 1: collect all the receipts
  - Step 2: ~~write them down on a notebook~~
    ~~store them in a text file~~
    use a spreadsheet

```
Date   Amount   Description
2/1    $20.21    Grocery
2/2    $10.54   Fast food
2/3    $39.22    Cell phone bill
…
2/27   $33.00    Clothes
```

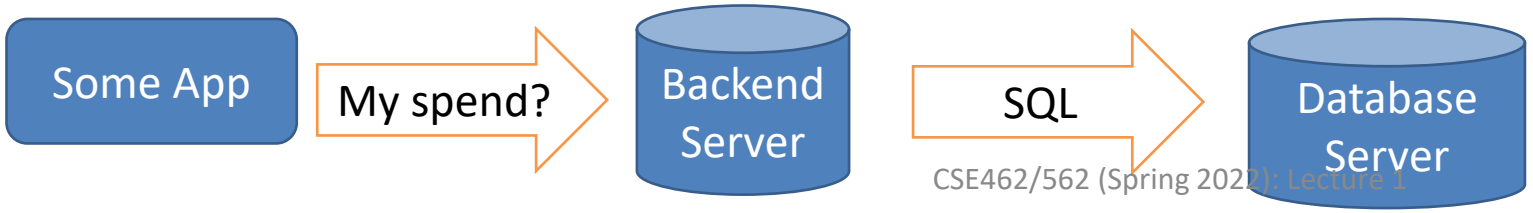
  - Step 3: do some analysis
    - How much did my spend on grocery and fast f
    - How much could I have saved if I cook by myse
    - What about January/last quarter/last year/pas

	A	B	C	D	E
1	Date	Amount	Description		
2	1-Feb	20.21	Grocery		
3	2-Feb	10.54	Fast food		
4	3-Feb	39.22	Cell phone		
5					
6					
7		Grocery	=SUMIFS(B2:B4,C2:C4,"Grocery")		

How to manage a database?

- Suppose I'd like to track my daily spending
- What I can do:
  - Step 1: collect all the receipts
  - Step 2: ~~write them down on a notebook~~
    ~~store them in a text file~~
    ~~use a spreadsheet~~
    use some personal finance app
  - Step 3: do some analysis
    - How much did my spend on grocery and fast food in Feburary?
    - How much could I have saved if I cook by myself in Feburary?
    - What about January/last quarter/last year/past five years?

```
Date   Amount   Description
2/1    $20.21    Grocery
2/2    $10.54   Fast food
2/3    $39.22    Cell phone bill
…
2/27   $33.00    Clothes
```




```
SELECT category, SUM(amount)
FROM spend
WHERE userid = 123456
GROUP BY category;
```

Why using a DataBase Management System?

- DataBase Management System (DBMS) is a software system for convenient and efficient data access over databases,

  which provides:
  - Data abstraction
    - Flexible data manipulation and query interfaces
    - Scalable data storage
    - Efficient query and transaction processing
  - Integrity checks
  - Concurrency control and atomicity
  - Fault tolerance
  - Security and privacy
  - …

Course overview

- Brief review of relational model and SQL
- Database storage
- Indexing
- Query processing and optimization
- Transaction processing and concurrency control
- Crash Recovery
- Advanced topics (tentative):
  - Approximate query processing
  - Parallel and distributed database
  - Hybrid transaction and analytical processing

More on Course Project

- Project info available on course website
- Language and tools: C++ (11), cmake, git
- Teams of up to 2 students
  - Code is submitted and graded as a team
- Each student: independently complete a write-up for each project 2 – 5
  - No team work!
  - Should include:
    - Description of your team's design in your own words
    - Description of division of coding responsibility (if you're in team)
    - Answers to additional questions
  - Submit to UBLearns.
- For those who don't have access to the required hardware,
  - you may use timberlake or metallica CSE student servers.

More on Course Project

- Code submission: autolab
  - If you do not have access or submission log says your UBIT name is unrecognized:
    - post privately under project_signup_questions folder before attempting another submission
  - Debug locally on your own machine or CSE student servers.
  - Don’t solely rely on Autolab log output for debugging. It’s shared.
    - Max 10 submissions per team per hour.
- No late submission accepted.
  - There’s a 10-min grace period after the posted deadline in case of network delays.
  - No individual extensions.

More on Course Project

- How we grade your submission
  - The team receives the same credit for coding
    - Online testing with Autolab
    - Offline testing on your last submission after deadline
      - No offline test for project 1 - lab 0: Project Sign-up
    - For each test case,
      - We take the higher score between the online and the offline tests.
  - Each student receives his/her own credit for write-up
    - No write-up -> no credits for the entire project
    - Write-up due 2 days after project deadline

Project 1: Project Sign-up and File I/O interface

- This is meant to be a warm-up project. *No write-up required for project 1.*
- Divided into two separate labs to guide you through the submission system:
  - Lab 0: setting up your code repository and make a successful submission
    - Due 2/8/2022, 11:59 pm EST.
    - Only one successful submission needed for each team
    - Caution: double check your teammate's UBIT name before you make the first submission. Once the system accepts it (even if your submission failed), you can't change it by yourself and without justification.
  - Lab 1: complete the file I/O interface using Linux I/O syscalls.
    - Due 2/15/2022, 11:59 pm EST.
    - You must have an accepted submission for lab 0 before you can submit lab 1.

Next time

- Project introduction and C++11 primer.