CSE462/562: Database Systems (Spring 23)

Lecture 1: Introduction & Course Logistics

1/31/2023

Davis 101, TR 11:00 am – 12:20 pm. In-person attendance required.

Find more on course website & Piazza:
https://cse.buffalo.edu/~zzhao35/teaching/cse562_spring23/
https://piazza.com/buffalo/spring2023/cse462562
Today’s agenda

• Introduction
  • What is a Database?
  • What is a Database Management System?
  • What is this course about and why should I care?

• Logistics
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
    • ...

CSE462/562 (Spring 2023): Lecture 1
What’s a DataBase Management System?

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

<table>
<thead>
<tr>
<th>Application Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL Interpreter</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Query Evaluator</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plan Generator</td>
</tr>
<tr>
<td>Plan Optimizer</td>
</tr>
<tr>
<td>Plan Executor</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Data Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transaction Manager</td>
</tr>
<tr>
<td>Buffer Manager</td>
</tr>
<tr>
<td>Security Manager</td>
</tr>
<tr>
<td>File &amp; Index Manager</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Database</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Files</td>
</tr>
<tr>
<td>Index Files</td>
</tr>
<tr>
<td>System &amp; Metadata Files</td>
</tr>
</tbody>
</table>
Why using a Database Management System?

• Let’s review an example of how to manage a database.
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 February?
  • How much could I have saved if I cook by myself in February?
  • 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

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

<table>
<thead>
<tr>
<th>Date</th>
<th>Amount</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2/1</td>
<td>$20.21</td>
<td>Grocery</td>
</tr>
<tr>
<td>2/2</td>
<td>$10.54</td>
<td>Fast food</td>
</tr>
<tr>
<td>2/3</td>
<td>$39.22</td>
<td>Cell phone bill</td>
</tr>
<tr>
<td></td>
<td></td>
<td>...</td>
</tr>
<tr>
<td>2/27</td>
<td>$33.00</td>
<td>Clothes</td>
</tr>
</tbody>
</table>
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 and store them in a text file

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

<table>
<thead>
<tr>
<th>Date</th>
<th>Amount</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2/1</td>
<td>$20.21</td>
<td>Grocery</td>
</tr>
<tr>
<td>2/2</td>
<td>$10.54</td>
<td>Fast food</td>
</tr>
<tr>
<td>2/3</td>
<td>$39.22</td>
<td>Cell phone bill</td>
</tr>
<tr>
<td></td>
<td></td>
<td>...</td>
</tr>
<tr>
<td>2/27</td>
<td>$33.00</td>
<td>Clothes</td>
</tr>
</tbody>
</table>

```python
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)
```

CSE462/562 (Spring 2023): Lecture 1
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

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

<table>
<thead>
<tr>
<th>Date</th>
<th>Amount</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2/1</td>
<td>$20.21</td>
<td>Grocery</td>
</tr>
<tr>
<td>2/2</td>
<td>$10.54</td>
<td>Fast food</td>
</tr>
<tr>
<td>2/3</td>
<td>$39.22</td>
<td>Cell phone bill</td>
</tr>
<tr>
<td></td>
<td></td>
<td>...</td>
</tr>
<tr>
<td>2/27</td>
<td>$33.00</td>
<td>Clothes</td>
</tr>
</tbody>
</table>
**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/build a personal finance app
  - Step 3: do some analysis
    - How much did I spend on grocery and fast food in February?
    - How much could I have saved if I cook by myself in February?
    - What about January/last quarter/last year/past five years?

<table>
<thead>
<tr>
<th>Date</th>
<th>Amount</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2/1</td>
<td>$20.21</td>
<td>Grocery</td>
</tr>
<tr>
<td>2/2</td>
<td>$10.54</td>
<td>Fast food</td>
</tr>
<tr>
<td>2/3</td>
<td>$39.22</td>
<td>Cell phone bill</td>
</tr>
<tr>
<td></td>
<td></td>
<td>...</td>
</tr>
<tr>
<td>2/27</td>
<td>$33.00</td>
<td>Clothes</td>
</tr>
</tbody>
</table>

```
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
  • …
What does this course cover?

• The design and implementation of DataBase Management System (DBMS)
  • Relational DBMS (RDBMS) as a case study
    • Stores tables that consist of rows and columns
    • Declarative query language (SQL) in the simple yet powerful relational model
  • Focus on principles and techniques generally applicable in Data Management

• Note, this course is not about
  (but we assume you have learned these somewhere else):
  • Database design
  • The relational model and the SQL language (we’ll briefly review them)
  • Programming/data structure/algorithm analysis/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
Logistics

• Davis 101, TR 11:00 am – 12:20 pm.
  • In-person attendance required.
  • We will have random quizzes.

• Instructor: Zhuoyue Zhao
  • Office hours: Monday 9:50 am to 11:50 am, and Tuesday 1:30 pm to 3:20 pm, Davis 338I.

• TA/Grader:
  • Congying Wang -- Office hours
    Wednesday 1:00 pm to 3:00 pm and Thursday 2:00 pm to 4:00 pm, location Davis 300 student lounge (the open space south of Davis 302).
  • Nithin Tellapuri – Q&A on Piazza, Monday and Friday 2:00 - 3:00 pm.

• No office hour in week 1
  • Please post on Piazza for help if there’s any issue with project 1

• Find more on course website:
  https://cse.buffalo.edu/~zzhao35/teaching/cse562_spring23/
Logistics

• We mainly use Piazza for communication:
  • [https://piazza.com/buffalo/spring2023/cse462562](https://piazza.com/buffalo/spring2023/cse462562)
  • Please post any request/question on Piazza instead of sending emails
    • Piazza reminds me of all unresolved questions but outlook doesn’t!

• When you have any private question/request for the instructor or TA:
  • please select “Instructors” in Post To
Logistics

• Important Dates:
  • Add/drop deadline: 2/6/2023
  • Mid-term exam: 3/9/2023, Knox 104, 7:10 pm – 8:40 pm
  • Last day to resign from the course: 4/21/2023
  • Final exam: 5/16/2023, 12:30 pm – 2:00 pm, Knox 104

• Open-book exams (only paper materials allowed)

• Exam conflict policy:
  • No alternative time for mid-term exam (sorry, limited space availability)
  • If you have final exam conflicts as defined by the Office of the Registrar
    • please notify the instructor on Piazza by 2/13/2023
    • (we might not have enough seats if you do not notify us by that date)
    • you may still opt for the original final exam at any time with one-week prior notice
Grading

• Grading
  • Random in-class quizzes: 10% (you may miss up to 3 without losing points)
  • Mid-term exam: 15%
  • Final exam: 20%
  • Projects: 55% + 10% in bonus

• Grading scale for letter grades:
  • No curving.

<table>
<thead>
<tr>
<th>Score Range</th>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>[0, 10)</td>
<td>F</td>
</tr>
<tr>
<td>[10, 20)</td>
<td>D</td>
</tr>
<tr>
<td>[20, 30)</td>
<td>C-</td>
</tr>
<tr>
<td>[30, 40)</td>
<td>C</td>
</tr>
<tr>
<td>[40, 50)</td>
<td>C+</td>
</tr>
<tr>
<td>[50, 60)</td>
<td>B-</td>
</tr>
<tr>
<td>[60, 70)</td>
<td>B</td>
</tr>
<tr>
<td>[70, 80)</td>
<td>B+</td>
</tr>
<tr>
<td>[80, 90)</td>
<td>A-</td>
</tr>
<tr>
<td>[90, +∞)</td>
<td>A</td>
</tr>
</tbody>
</table>
Course project

• Build a mini RDBMS through 5 projects (C++ 11)
  • Project 1 (project sign-up and C++ practice) due on 2/7, 1:00 AM.

• Each project includes:
  • Coding: private Github repo; submit tags to Autolab
  • Write-ups: submit a PDF to UBLearsns with your own answers to a list of questions

• Deadlines and late submission policy:
  • coding: no late submission accepted. 10-min grace period in case of network issues.
    • If you are unable to make submission within the grace period but have committed your code by deadline, please post the commit tag on Piazza for help.
  • write-ups: due 2 days after each project deadline

• Teams allowed with up to 2 students
  • teamwork allowed only within teams and on coding
  • write-ups must be completed independently (without consulting your teammate!)
Course project

• Instructions for projects:
  • Project pages contain very detailed instructions.
  • If something requires clarification, it’s most likely covered there.
  • Still have questions on project or found bugs?
    • Feel free to post it on Piazza (though we may point you back to the instructions).
    • Your team will get 1 extra credit towards your final grade for every validated bug or question that cannot be answered by the project instruction.

• Where to find project pages:
  https://cse.buffalo.edu/~zzhao35/teaching/cse562_spring23/
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
  • unless it’s an honest mistake that does not give anyone any undue advantage
    • (e.g., you accidentally set your Github repo to public but changed it back before anyone accesses it)
More on Academic Integrity Policy

- Think of the course projects as take-home exams:
  - you must complete them by yourself (or with your teammate for coding only)
  - please do not discuss any project specifics outside your team

- Examples of AI violation related to course project:
  - Discussion of code with any student who is not your teammate
  - Viewing/committing/submitting code written by anyone who is not your teammate
    - verbatim or with modification
  - Discussion of project write-ups with any student (including your teammate)
  - Viewing/copying/rephrasing answers found online or from a past or current student

- What is allowed and encouraged (on Piazza/in lecture/offline, publicly or privately)
  - Ask questions about lectures
  - Discuss (the ungraded) written assignments
  - Preparation for mid-term and final exams
  - Looking up C++ references on cpprefernce.com/cplusplus.com
Next time

• Storage