CS315: Principles of Database Systems
Introduction

Arnab Bhattacharya
arnabb@cse.iitk.ac.in

Computer Science and Engineering,
Indian Institute of Technology, Kanpur
http://web.cse.iitk.ac.in/˜cs315/

2nd semester, 2013-14

Tue, Fri 1530-1700 at CS101

Rules

- Email `arnabb@cse.iitk.ac.in` to meet or for any thing else
- Put “CS315” in the subject for automatic mail filters
- Participate
  - Attend classes
  - Clear doubts
  - Answer questions
- Do homeworks *individually*
- *No* extension of deadlines for degradation of health of
  - Yourself
  - Your computer
  - Your family members
  - Your special friends

  unless notified well in advance
- If you are sick, follow IITK procedure
  - Produce a sick certificate, etc.

Grading policy

- Exams: 50%
  - End-semester: 30%
  - Mid-semester: 20%
- Assignments + Quiz: 20%
- Project: 25%
- Class participation: 5%

Project details

- At most groups of 3
- Groups will be formed randomly to maintain uniform average CPI
- Form your *own* idea
  - Get it approved by me
- I will try to put up some of my ideas as well
- Deadlines
  1. Area of project: 29th January
  2. Demonstration: 29th March
  3. Final report: 5th April

Course material

- Slides
- Classwork
- Books
  1. “Database System Concepts” by Silberschatz, Korth & Sudarshan. *McGraw-Hill*, Fifth Edition, 2006.
  2. “Fundamentals of Database Systems” by Elmasri & Navathe. *Pearson Education*, Sixth Edition, 2013.
  3. “Database Management Systems” by Ramakrishnan & Gerhke. *McGraw-Hill*, Third Edition, 2003.

Course contents

1. Motivation
2. Relational model
3. Relational algebra
4. SQL
5. Relational calculus
6. ER diagrams
7. Normalization theory
8. Physical design
9. Indexing
10. Query processing
11. Query optimization
12. Transactions
13. Concurrency control
14. Recovery systems

Concept of a database

- A database is a collection of *interrelated* data
- A database management system (DBMS) provides an environment that is *efficient* and *convenient* to use
- Programs and interface to
  - Store data
  - Visualize data
  - Access (query) data
  - Manipulate data

Advantages over file systems

Advantages over file systems

- Reduces data redundancy and data inconsistency
  - Stores data once
  - Can be combined at runtime

Advantages over file systems

- Reduces data redundancy and data inconsistency
  - Stores data once
  - Can be combined at runtime
- Data isolation
  - Data is stored in an internal format
  - One interface to access data

Advantages over file systems

- Reduces data redundancy and data inconsistency
  - Stores data once
  - Can be combined at runtime
- Data isolation
  - Data is stored in an internal format
  - One interface to access data
- Data integrity
  - Example: CPI is positive and not more than 10
  - Logic is encoded in databases, not buried in program code

Advantages over file systems

- Reduces data redundancy and data inconsistency
  - Stores data once
  - Can be combined at runtime
- Data isolation
  - Data is stored in an internal format
  - One interface to access data
- Data integrity
  - Example: CPI is positive and not more than 10
  - Logic is encoded in databases, not buried in program code
- Atomicity of operations
  - Transactions
  - Example: money transfer from one account to the other

Advantages over file systems

- Reduces data redundancy and data inconsistency
  - Stores data once
  - Can be combined at runtime
- Data isolation
  - Data is stored in an internal format
  - One interface to access data
- Data integrity
  - Example: CPI is positive and not more than 10
  - Logic is encoded in databases, not buried in program code
- Atomicity of operations
  - Transactions
  - Example: money transfer from one account to the other
- Concurrency for multiple users
  - Automatically handles access problems and inconsistency issues

Advantages over file systems

- Reduces data redundancy and data inconsistency
  - Stores data once
  - Can be combined at runtime
- Data isolation
  - Data is stored in an internal format
  - One interface to access data
- Data integrity
  - Example: CPI is positive and not more than 10
  - Logic is encoded in databases, not buried in program code
- Atomicity of operations
  - Transactions
  - Example: money transfer from one account to the other
- Concurrency for multiple users
  - Automatically handles access problems and inconsistency issues
- Security
  - Provides access to only some part of the data