CS1555/2055

Database Management Systems

Instructor: Panos K. Chrysanthis
Recitations/TA: Xioazhong Zhang

Fall 2018 (2191, 19-1)

db.cs.pitt.edu/courses/cs1555/current.term

Why CS1555/2055?

- **Top reasons to take CS1555/2055**
  - My friends are taking it too
  - Works with my schedule
  - I wanna know how database systems work
  - I wanna do research in database systems
  - I want a job in/with database systems / data science

Data Management - Industry

Robert Half International, a firm providing specialized staffing services, noted the following salary ranges for computer-related occupations in their 2007 Salary Guide:

Database manager	$84,750 - $116,000
Network architect	78,000 - 112,250
Database developer	73,500 - 103,000
Senior web developer	71,000 - 102,000
Database administrator	70,250 - 102,000
Network manager	68,750 - 93,000
Web developer	54,750 - 81,500
LAN/WAN administrator	51,000 - 71,500
Web administrator	49,750 - 74,750
Web designer	47,000 - 71,500
Telecommunications specialist	47,500 - 69,500

[US Department of Labor]

Some Recent Salary Numbers

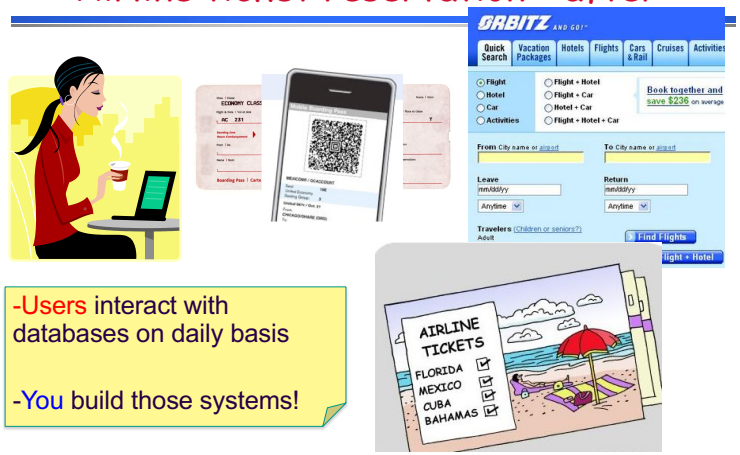
	Database Administrators	Database administrators (DBAs) use specialized software to store and organize data, such as financial information and customer shipping records. They make sure that data are available to users and secure from unauthorized access.	Bachelor's degree	$87,020
	Information Security Analysts	Information security analysts plan and carry out security measures to protect an organization's computer networks and systems. Their responsibilities are continually expanding as the number of cyberattacks increases.	Bachelor's degree	$95,510
	Network and Computer Systems Administrators	Computer networks are critical parts of almost every organization. Network and computer systems administrators are responsible for the day-to-day operation of these networks.	Bachelor's degree	$81,100
	Software Developers	Software developers are the creative minds behind computer programs. Some develop the applications that allow people to do specific tasks on a computer or another device. Others develop the underlying systems that run the devices or that control networks.	Bachelor's degree	$103,560
	Web Developers	Web developers design and create websites. They are responsible for the look of the site. They are also responsible for the site's technical aspects, such as its performance and capacity, which are measures of a website's speed and how much traffic the site can handle. In addition, web developers may create content for the site.	Associate's degree	$67,990

[US Department of Labor: Friday, April 13, 2018]

Databases everywhere

Airline ticket reservation - before

Airline ticket reservation - after



-Users interact with databases on daily basis

-You build those systems!

When Larry met Sergy

Data Management - Research

Google

Panos K. Chrysanthis

List of publications from the DBLP Bibliography Server - FAQ

Facets and more with CompleteSearch

Ask others: ACM DL/Guide - - CSB - MetaPress - Google - Bing - Yahoo

author:panos_k_chrysanthis:

2011

155 Sean M. Snyder, Shimin Chen, Panos K. Chrysanthis, Alexandros Labrinidis: QMD: exploiting flash for energy efficient disk arrays. DaMoN 2011: 41-49

154 Marian Kamal Iskander, Dave W. Wilkinson, Adam J. Lee, Panos K. Chrysanthis: Enforcing Policy and Data Consistency of Cloud Transactions. ICDCS Workshops 2011: 253-262

153 Thao N. Pham, Lory Al Moakar, Panos K. Chrysanthis, Alexandros Labrinidis: DILoS: A dynamic integrated load manager and scheduler for continuous queries. ICDE Workshops 2011: 10-15

152 Alexander G. Connor, Panos K. Chrysanthis, Alexandros Labrinidis: Key-key-value stores for efficiently processing graph data in the cloud. ICDE Workshops 2011: 88-93

151 Panayiotis Neophytou, Panos K. Chrysanthis, Alexandros Labrinidis: CONFLuEnCE: CONtinuous workFLow ExeCution Engine. SIGMOD Conference 2011: 1311-1314

150 Panayiotis Andreou, Demetrios Zeinalipour-Yazti, Panos K. Chrysanthis, George Samaras: Power efficiency through tuple

Refine by AUTHOR
Panos K. Chrysanthis (155)
Alexandros Labrinidis (31)
George Samaras (25)
Mohamed A. Sharaf (18)
[top 4] [top 50] [all 116]

Refine by VENUE
ICDE (10)
SAC (9)
DMSN (7)
MobiDE (6)
[top 4] [top 50] [all 78]

Refine by YEAR
2004 (17)
2009 (14)
2005 (13)
2003 (13)
[top 4] [all 24]

"Big" Data is an important problem

- You know Big Data is an *even more* important problem if... It has a Dilbert cartoon!

Focus of the CS1555/2055 course

- CS1555/2055: Application-oriented study of databases
  - Introduce fundamental concepts of data management
  - Design and use of a database system
  - Provide practical experience in applying these concepts using commercial DBMS
- CS1655: Advanced topics of databases
  - XML, XML Query Optimizations, applications & Web
- CS1656: Introduction to Data Science
  - Different data manipulation & data analysis techniques beyond traditional data management
- CS2550: System-oriented study of databases
  - Design of database management system

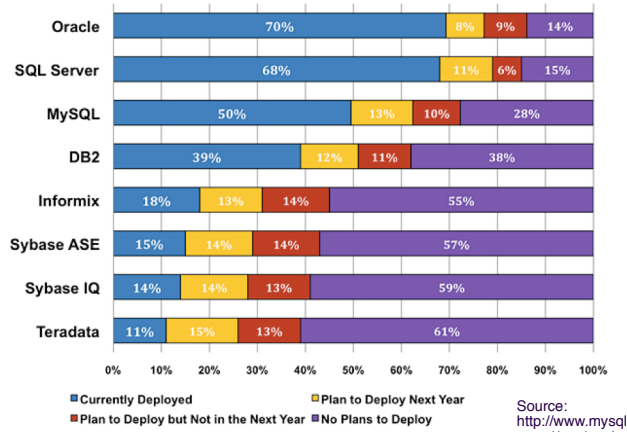
Quick Facts - Lectures

- When: Tue & Thu 11:00 am – 12:15 pm
- Where: G28 Benedum Hall
- What:
  - *Lecture Notes*
  - *Fundamentals of Database Systems*, R. Elmasri and S. B. Navathe, 7th edition, 2015
  - *Oracle10g Programming: A Primer*, R. Sunderraman, Addison-Wesley, 2007
  - *Learning SQL*, Alan Beaulieu, O'Reilly 2009 (Available online from campus computers through Safari Bookshelf)

Recitations

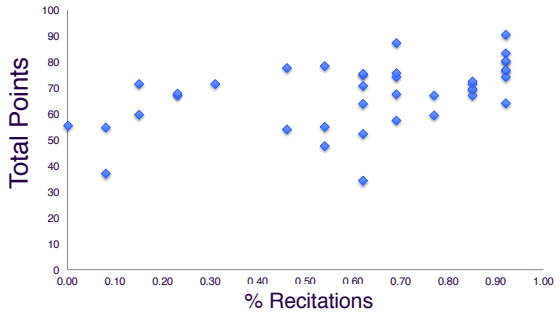
- When & Where (1st Recitation Aug 31st)
  - Friday, 11:00 - 11:50 am
  - Friday, 2:00 - 2:50 pm
  - @ 6110 Sennott Square Building
- What: ***Complements*** *&* ***goes beyond the lectures***
  - solve problems & homework,
  - study more Examples,
  - practice with Oracle,
  - it is required… attendance and participation in lecture and recitation may be used to decide borderline grades.

Why Oracle?



Source: http://www.mysql.com/why-mysql/marketshare

Recitation Attendance Vs. Scores



Assessments - Grading

Assessment	Percentage	Dates
Homeworks	20%	Multiple
Term Project	15%	3-phases
Midterm Exam	30%	Oct. 18, 2018
Final Exam	30%	Dec. 11, 2018
Participation (Class & Recitations)	5%	

Administrative

- web page: check often!
  http://db.cs.pitt.edu/courses/cs1555/current.term
  http://db.cs.pitt.edu/courses/cs2055/current.term
- class email: `cs1555-staff@cs.pitt.edu`
  - For *confidential* matters only
  - use keyword `cs1555` in all emails to instructor/TA (as part of the subject line)
  - it works *only* within pitt.edu
- Piazza:
  - for all clarifications to lectures, recitations and assignments
- assignments:
  - To be submitted electronically
  - **No** Piazza or email clarifications 4 hours prior a deadline

Piazza Guidelines

- Remember that **everything you post is**
- Please do not post any assignment code on Piazza, even as a private message. Visit office hours for any code related questions.
- Keep posts on Piazza course-related.
- Please read all questions and responses that are on Piazza before asking a question. Utilize the excellent Piazza search facilities.
- Use a meaningful subject heading.
- Make sure your questions/posts are in the appropriate folder (e.g., hw0, hw1, hw2, ...).
- Tag your post with all the applicable tags.
- Please don't post things to the group that give no useful information.
- Please keep complaints about the course out of the newsgroup. If you have a concern about anything to do with the course, please talk to the instructor.
- **Please be respectful of your peers and others in your posts.**

Overview of Database Management Systems

What is a database?

- A very large, integrated collection of **related** data
  - data is raw facts on some aspect of the world
- Models a real-world enterprise (e.g., university)
  - **Entities** (e.g., students, courses)
  - **Relationships** (e.g., Bob took CS 1550)

Students

SID	*Name*	*Age*	*GPA*
546007	Susan	18	3.8
546100	Bob	19	3.65
546500	Bill	20	3.7

Courses

CID	*CName*
CS 1555	DB
CS 1530	SW
CS 1550	OS

Enrollment

SID	*CID*	*Grade*
546007	CS 1550	A
546007	CS 1530	B+
546100	CS 1550	B

Integrated Data

- All data are stored and manipulated in a *uniform way* on a secondary storage

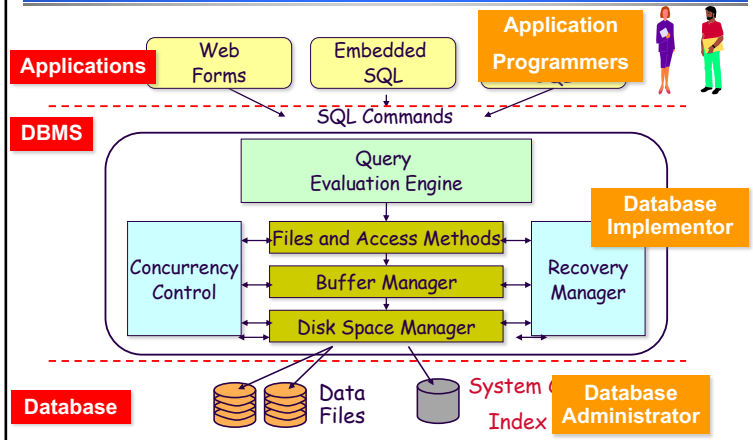
  - Databases store large amounts of data that cannot fit in main memory.
  - Data are stored for long and indefinite period
  - Data are shared across multiple applications

What is a Database Management System?

- ***Database Management System (DBMS):***
  - A general purpose software package designed to store and manage databases *conveniently & efficiently*
- **DBMSs:**
  - Oracle, IBM DB2, SQLServer, MySQL, PostgreSQL …
- **Usage:**

  Database system = *DB* + *DBMS* + *Application Logic*
  - Resource Planning Applications:
    - PeopleSoft, SAP, …
  - Web-based Applications:
    - amazon, ebay, orbitz, trip-advisor, yelp, …

Database Management System (DBMS)



Approaches to management of data

- File system approach
  - Traditional (flat) files +
    C (Java, ...) programs to access them
  - E.g., use one (or more) UNIX/DOS files, with student records and their courses
  - Decide on a layout for the student records, etc.
- Database approach

Database Vs. File Systems Approaches

- Abstraction
  - Data
  - Execution
- Reliability
- Efficiency/Performance

Data Abstraction

- Data are structured in a way meaningful to applications
- **Data Model:**
  - A collection of high-level data description constructs that hide low-level storage details
- **The Relational / Object-Relational Model:**
  - Is the most widely used data model today
  - Main construct is a ***relation***: table of records
  - Every relation has a ***schema***:
    - Relation name
    - Names of fields
    - Types of fields

Example

Attribute or Field

Schema

Record or Tuple

SID	*Name*	*Age*	*GPA*
546007	Peter	18	3.8
546100	Bob	19	3.65
546500	Bill	20	3.7

- Schema:
  - Students (*sid:* `string`, *name*: `string`, *age*: `integer`, *gpa*: `real`)
- State: Actual data at a given point in time

Database Languages

- **Data Definition Language (*DDL*):**
  - Define schemas
  - Define **Integrity Constraints**
    - Example: unique *SID*s
  - More…
- **Data Manipulation Language (*DML*):**
  - To ask questions = ***Query***
    - Example: Which students have GPA > 3.75?
  - To create and modify data
- ***SQL:*** Most widely used database language

Good DBMS ≠ Good Design!

Attribute or Field

Schema

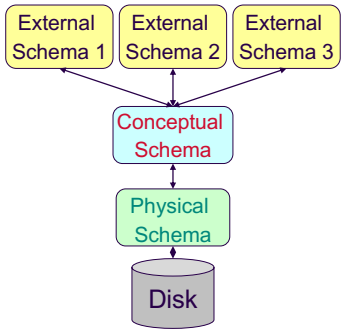
Record or Tuple

SID	*Name*	*Age*	*GPA*
546007	Peter	18	3.8
546100	Bob	19	3.65
546500	Bill	20	3.7

- Schema:
  - Students (*sid*: `string`, *name*: `string`, *age*: `integer`, *gpa*: `real`)
  - Alternative Schema:
    - Students (*sid*: `integer`, *fname*: `string`, *lname*: `string`, *dob*: `date`, *gpa*: `real`)

Levels of Data Abstraction in a DBMS

- The data in a DBMS is described at three levels of abstraction:
  1. *Conceptual Schema*
     DDL: data definition language
  2. *Physical Schema*
     SDL: storage definition language
  3. *External Schema* (Views)
     VDL: view definition language



- Many external schemas
- plus one conceptual
- plus one physical

External Schema - Views

- Allows data access to be customized and authorized at the user-level
  - Defined in terms of data model
  - Consists of a collection of **views**
  - Example:
    - CourseEnroll (*cid*: `string`, *enrollment*: `int`)

CID	*Enroll*
CS 1555	37
CS 2550	17
CS 3551	15

- Guided by end-user requirements
- Views are computed as needed
- Multiple Views of data allows each user/application to get different perspective of the database

3-level Architecture

- view level
  - CSMajors
  - MathMajors
- logical level: entire database schema
  - Courses (CourseNo,CourseName,Credits,Dept)
  - Students (StudentID,Lname,Name,Class,Major)
  - GradeReport (StudentID,CourseNo,Grade,Term)
- physical level:
  - how these tables are stored
  - how many bytes and attribute, etc.

Execution Abstraction

- *A **transaction** is a **logical unit of work** in DBMSs*
  - *It is the execution of a **program segment** that performs some function or task by accessing shared data (e.g., a db)*
  - logical grouping of query and update requests needed to perform a task
- Examples:
  - deposit, withdraw, transfer money (banking transaction)
  - reserve a seat on a flight (airline reservation)
  - print monthly payment checks (business transaction)
  - update inventory (inventory transaction)

ACID Properties

- **A**tomicity (alias failure atomicity)
  Either all the operations associated with a transaction happen or none of them happens
- **C**onsistency Preservation
  A transaction is a correct program segment. It satisfies the integrity constraints on the database at the transaction's boundaries
- **I**solation (alias concurrency atomicity / serializability)
  Transactions are independent, the result of the execution of concurrent transactions is the same as if transactions were executed serially, one after the other
- **D**urability (alias persistence / permanence)
  The effects of completed transactions become permanent surviving any subsequent failures

Database Vs. File Systems Approaches

- Abstraction
  - Data
  - Execution
- **Reliability**
- Efficiency/Performance

Reliability

- Enforcing *integrity constraints*
  - E.g., data type, relationship between values
  - Data in DB must satisfy the integrity constraints
  - Transactions are committed if they do not violate any integrity constraint
  - Integrity constraints are stored in the catalog
- Ensuring *data integrity* despite failures
  - Data are not lost when the system or a transaction fails for whatever reason

Reliability…

Ensuring *data integrity* …

- Security
  - Encryption & Private Information Retrieval
  - Authentication
  - Data Domains, Compartmentalization
- Access control
  - *Who* (user/role), *what* (data), *how* (operations)
  - Views and access permissions in the catalog

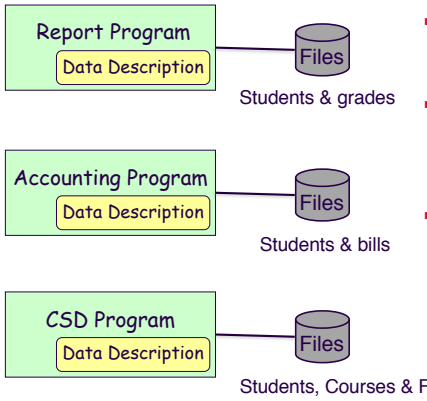
Example: Database Access ☺

"On Facebook, 273 people know I'm a dog.
The rest can only see my limited profile."

Database Vs. File Systems Approaches

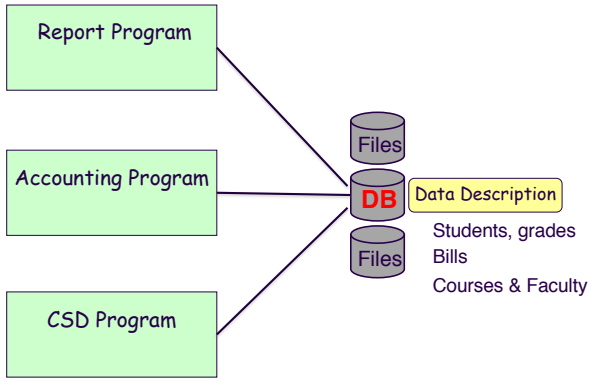
- Abstraction
  - Data
  - Execution
- Reliability
- **Efficiency/Performance**

Performance Problems with Files



- **Reduntancy**
  - waste of space
- **Waste of effort**
  - Same change in multiple places
- **Need for coordination**
  - inconsistent data
  - independent updates
  - E.g., Jan-19-1974 vs. Jan-29-1974

No Performance Problems with DBs



Efficiency and Performance

- Space efficiency:
  - minimizes data redundancy by storing data only once
- Time efficiency (response time):
  - eliminates the need for multiple updates to keep the replicas consistent and up-to-date
  - Enhances query performance by means of ***optimizations*** and ***access methods***
  - Allows many users (transactions) to access and share the database ***concurrently***

Performance Requirements

- Abstraction
  - Data abstraction
  - Execution abstraction
- Reliability
  - High availability: recovery time is *short*
  - Trusted/Quality data
- Efficiency/Performance
  - High throughput (Committed transactions per unit time)
  - Short or bounded response time
  - Energy Efficiency

When an SQL-DBMS is Inappropriate?

- Disadvantages:
  - Price to buy (DBMS & Hardware)
  - additional expertise (SQL/DBA)
- Hence, it is ***over-kill*** when
  - the database has simple structure and/or its size is small
  - the application is simple, special purpose and is not expected to change
  - Concurrent, multiple-user access is not required
  - Can tolerate failures
  - Monitor applications, high volume of updates

When a DBMS is needed?

- To integrate efficiently and correctly large volumes of related data
  - Central control, tuning for better performance, security
- Building of large applications
  - New applications using a DBMS is estimated 1/6 to ¼ of the time of application using a file system
  - Economy of scale
- Expandability/Flexibility
  - Supports evolution without affecting existing applications
- Supports sharing and discovering of information
  - Data mining, deductive databases, active databases
- DBMS ensure true QoS

Topics for the Term

1	Relational Model	Theory
2	DDL	
3	Normalization	
4	ER Modeling	
5	Relational Algebra	
6	DML & VDL	Systems
7	Access Control	
8	Transactions	
9	Integrity Constraints	
10	Database Programming	SQL/Programming
11	File Storage & Indexing	
12	Query Processing	
13	Concurrency Control	
14	Recovery	