InfoLab

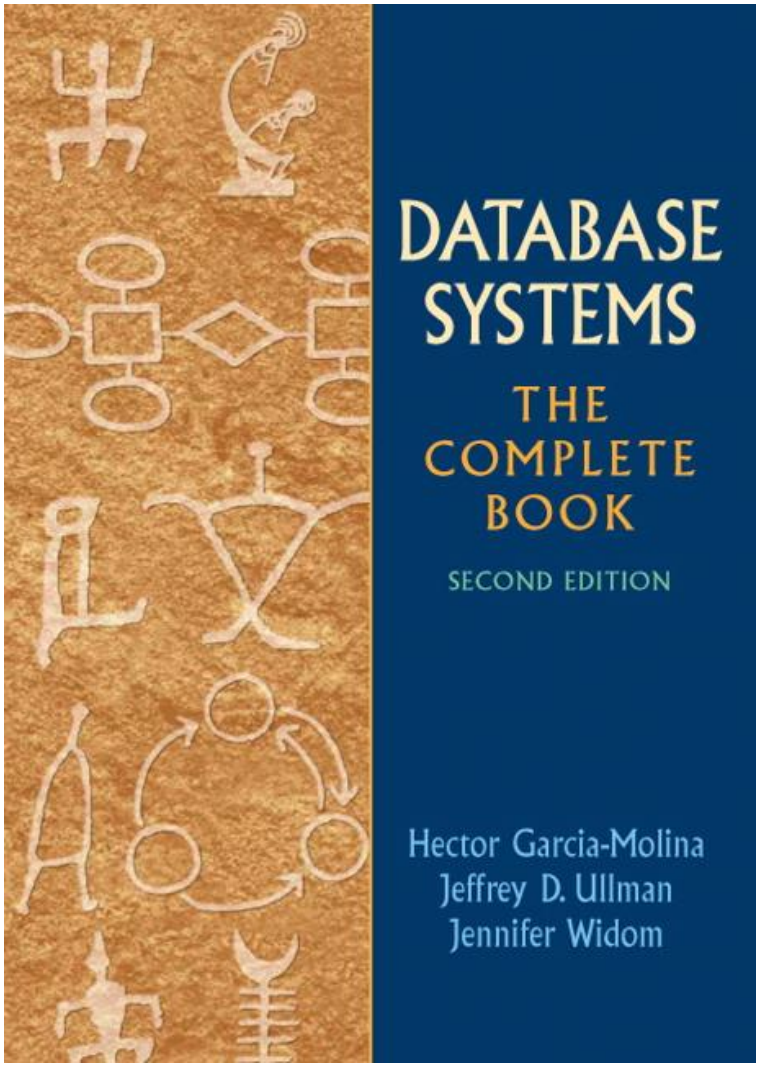
IC611: Database Systems Design

Course Information

Min-Soo Kim

DGIST 대구경북과학기술원
Daegu Gyeongbuk Institute of Science & Technology

Text book



- **Authors**
  - Hector Garcia-Molina
  - Jeffrey D. Ullman
  - Jennifer Widom
- **Edition : 2nd**
- **Publisher : Prentice Hall (2008)**

Related courses

500 Level

IC512
Object-Oriented Programming

600 Level

IC606
Computer Architecture

IC607
Operating Systems

IC619
System Programming

IC621
Distributed and Parallel Computing

IC605
Algorithms

IC611
Database Systems Design

IC612
Data Warehousing and Data Mining

700 Level

IC709
Theory of Computation

IC705
Algorithms in Bioinformatics

IC611 Course outline – Part I

- Lecture 1: Introduction
- Lecture 2: The Relational Model
- Lecture 3: Relational Algebra
- Lecture 4: Relational Calculus
- Lecture 5: SQL
- Lecture 6: Relational Design Theory
- Lecture 7: Storage Management
- Lecture 8: Index Structures
- Lecture 9: Query Execution
- Lecture 10: Query Optimization

IC611 Course outline – Part II

- **Lecture 11: Parallel & Distributed Data Processing**
- **Lecture 12: Graph Database Systems**
- **Lecture 13: Key-Value Database Systems**
- **Lecture 14: Array Database Systems**

Course information

- **Class Homepage:** http://infolab.dgist.ac.kr/~mskim/IC611/
- **Time: Wed. 9:00~12:00am**
- **Place: #113 (#224), E-3**

- **Evaluation**
  - **Attendance (10%)**
  - **Homework (20%)**
  - **Midterm Exam (50%)**
  - **Final term (or Paper presentation) (20%)** (at least 2 papers)

Purpose of paper presentation

- **To capture the trend of the database research area**
- **To understand the real problems and solutions dealt with in the industry**
- **To improve your presentation skills**
  - Make people understand the core idea clearly
  - Make the slides clear, concise, and easy to understand
  - Prepare the script of your presentation
  - Don’t add the contents(even a symbol) you couldn’t understand clearly
  - Don’t add too much contents

Tips for paper presentation

- **Script for presentation MUST be prepared**
  - otherwise, there would be some penalty
- **Answering questions readily is VERY important**
  - otherwise, there would be some penalty
- **Presenting all contents of the paper is NOT important**
  - only focus on delivering the core ideas of the paper
- **Tentative schedule : May-June**
  - Coordinator will arrange the presentation schedule

How to select your papers

- **Choose three papers from the pre-selected list of papers (from major conferences in computer science)**
  - VLDB / SIGMOD
  - KDD
  - ODSI / SOSP
  - …
- **The pre-selected list will be announced in May**

InfoLab

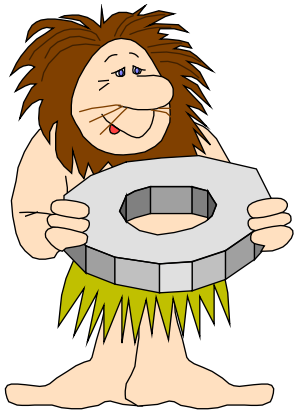
IC611: Database Systems Design

Introduction

from Database Management Systems, 3rd ed.

DGIST 대구경북과학기술원 Daegu Gyeongbuk Institute of Science & Technology

What Is a DBMS?



- A very large, integrated collection of data
- Models real-world enterprise
  - Entities (e.g., students, courses)
  - Relationships (e.g., Kim is taking IC611)
- A Database Management System (DBMS) is a software package designed to store and manage databases

Files vs. DBMS

- Application must stage large datasets between main memory and secondary storage (e.g., buffering, page-oriented access, etc.)
- Special code for different queries
- Must protect data from inconsistency due to multiple concurrent users
- Crash recovery
- Security and access control

Why Use a DBMS?

- Data independence and efficient access
- Reduced application development time
- Data integrity and security
- Uniform data administration
- Concurrent access, recovery from crashes

Why Study Databases??

- **Shift from computation to information**
  - at the “low end”: scramble to webspace (a mess!)
  - at the “high end”: scientific applications

- **Datasets increasing in diversity and volume**
  - Digital libraries, interactive video, Human Genome project, NASA’s EOS project
  - ... need for DBMS exploding

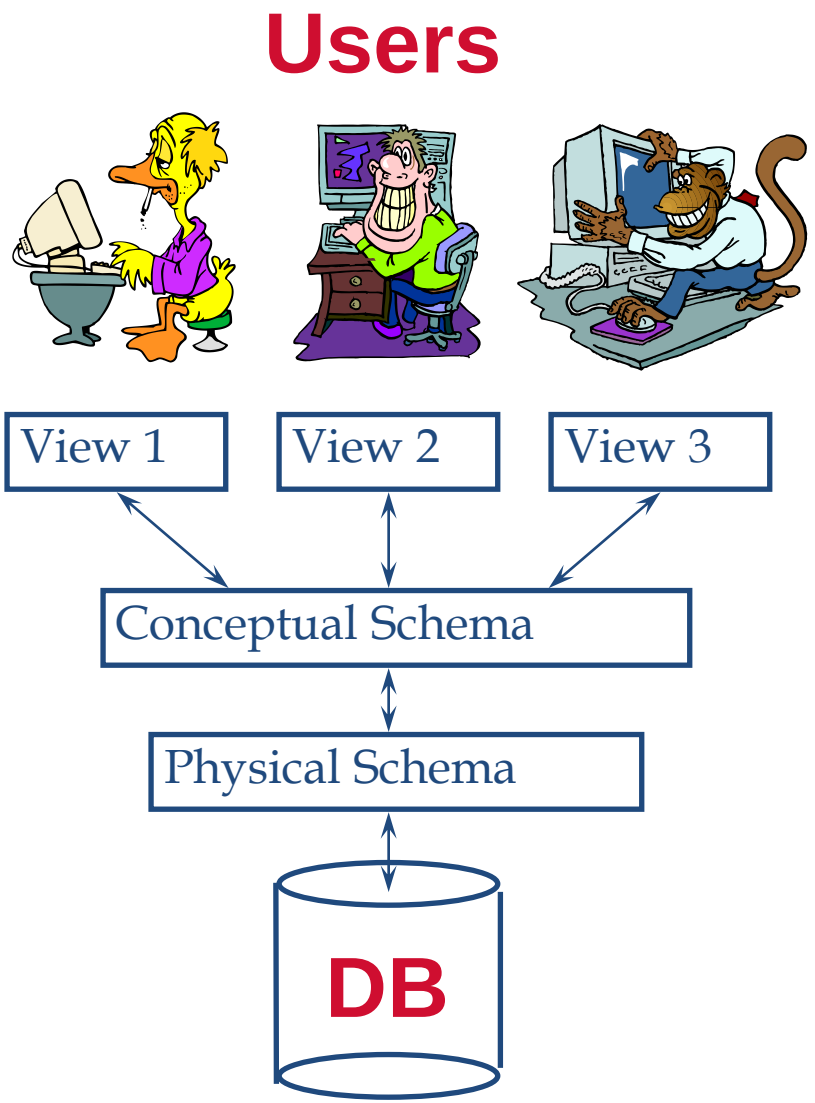
- **DBMS encompasses most of CS**
  - OS, languages, theory, AI, multimedia, logic

Data Models

- **A data model is a collection of concepts for describing data**
- **A schema is a description of a particular collection of data, using the a given data model**
- **The relational model of data is the most widely used model today**
  - Main concept: relation, basically a table with rows and columns
  - Every relation has a schema, which describes the columns, or fields

Levels of Abstraction

- **Many views, single conceptual (logical) schema and physical schema**
  - Views describe how users see the data
  - Conceptual schema defines logical structure
  - Physical schema describes the files and indexes used
  - (sometimes called the ANSI/SPARC architecture)



- **Schemas are defined using DDL**
- **data is modified/queried using DML**

Example: University Database

- **Conceptual schema:**
  - *Students(sid: string, name: string, login: string, age: integer, gpa:real)*
  - *Courses(cid: string, cname:string, credits:integer)*
  - *Enrolled(sid:string, cid:string, grade:string)*

- **Physical schema:**
  - Relations stored as unordered files
  - Index on the first column of Student

- **External Schema (View):**
  - *Course_info(cid:string,enrollment:integer)*

Data Independence

- **Applications insulated from how data is structured and stored**
- **Logical data independence: Protection from changes in logical structure of data**
- **Physical data independence: Protection from changes in physical structure of data**

- **Q: Why is this particularly important for DBMS?**

Because rate of change of DB applications is incredibly slow.
More generally:

dapp/dt << dplatform/dt

Concurrency Control

- **Concurrent execution of user programs is essential for good DBMS performance**
  - Because disk accesses are frequent, and relatively slow, it is important to keep the CPU busy by working on several user programs concurrently

- **Interleaving actions of different user programs can lead to inconsistency**

- **DBMS ensures such problems don't arise: users can pretend they are using a single-user system**

Transaction: An Execution of a DB Program

- **Key concept is a transaction: an atomic sequence of database actions (reads/writes)**

- **Each transaction, executed completely, must leave the DB in a consistent state if DB is consistent when the transaction begins**
  - Users can specify some simple integrity constraints on the data, and the DBMS will enforce these constraints
  - Beyond this, the DBMS does not really understand the semantics of the data. (e.g., it does not understand how the interest on a bank account is computed)
  - Thus, ensuring that a transaction (run alone) preserves consistency is ultimately the user's responsibility!

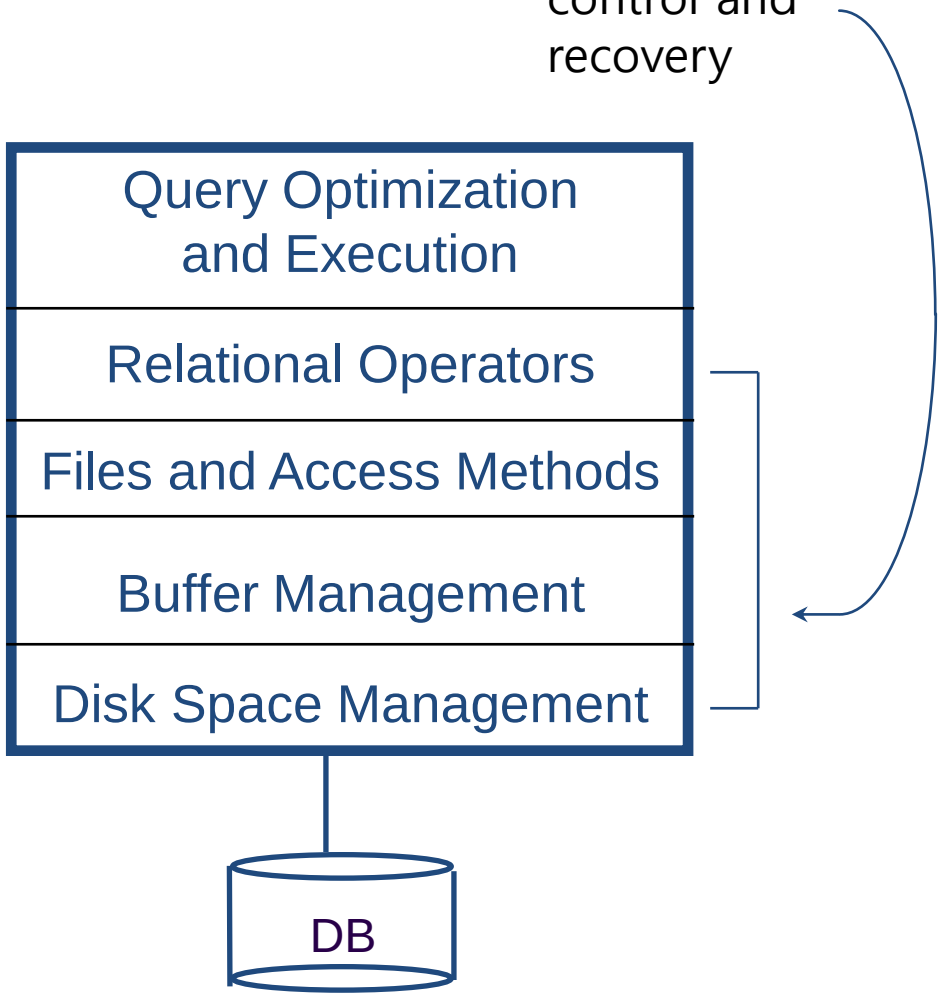
Databases make these folks happy …

- **End users and DBMS vendors**
- **DB application programmers**
  - E.g. smart webmasters
- **Database administrator (DBA)**
  - Designs logical /physical schemas
  - Handles security and authorization
  - Data availability, crash recovery
  - Database tuning as needs evolve

- *Must understand how a DBMS works!*

Structure of a DBMS

- **A typical DBMS has a layered architecture**
- **The figure does not show the concurrency control and recovery components**
- **This is one of several possible architectures; each system has its own variations**



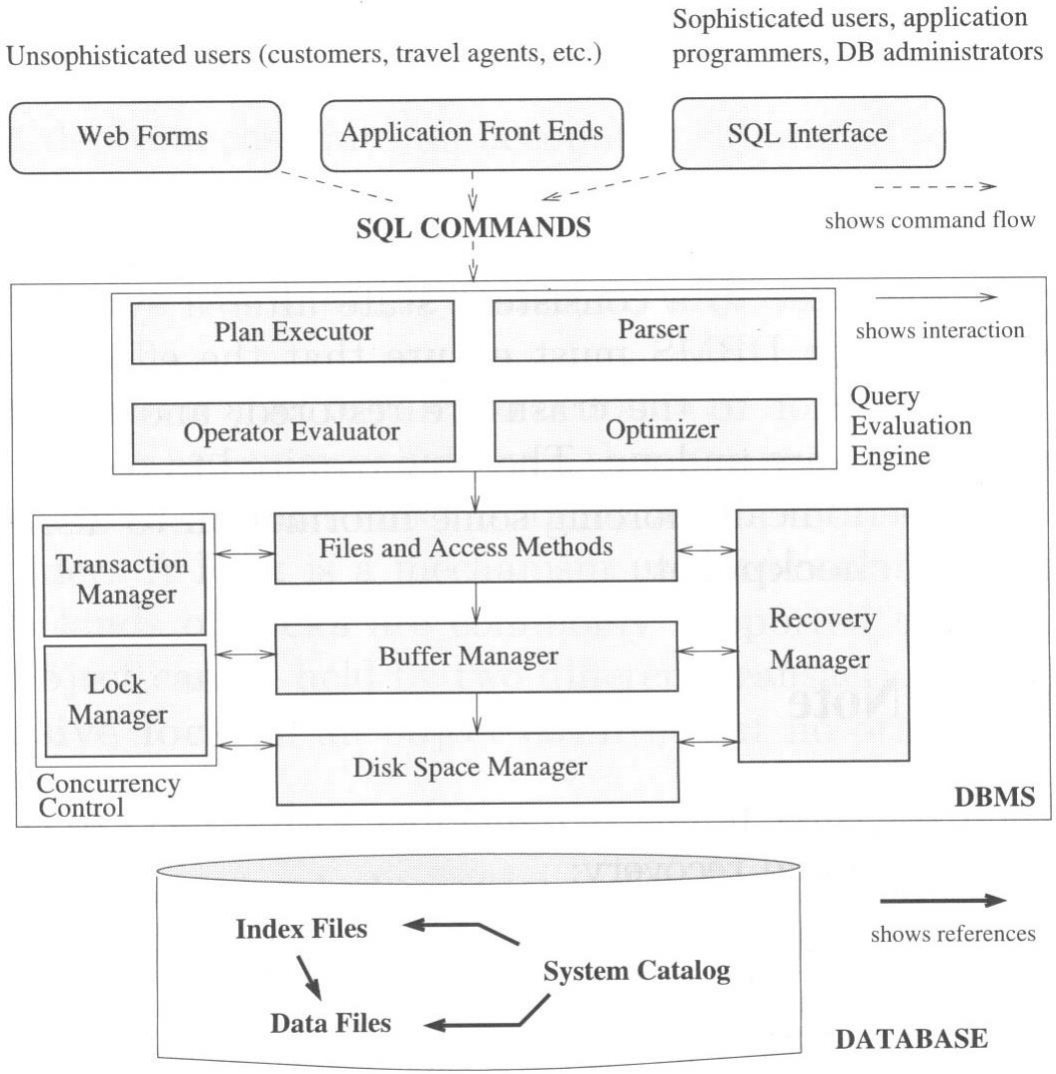
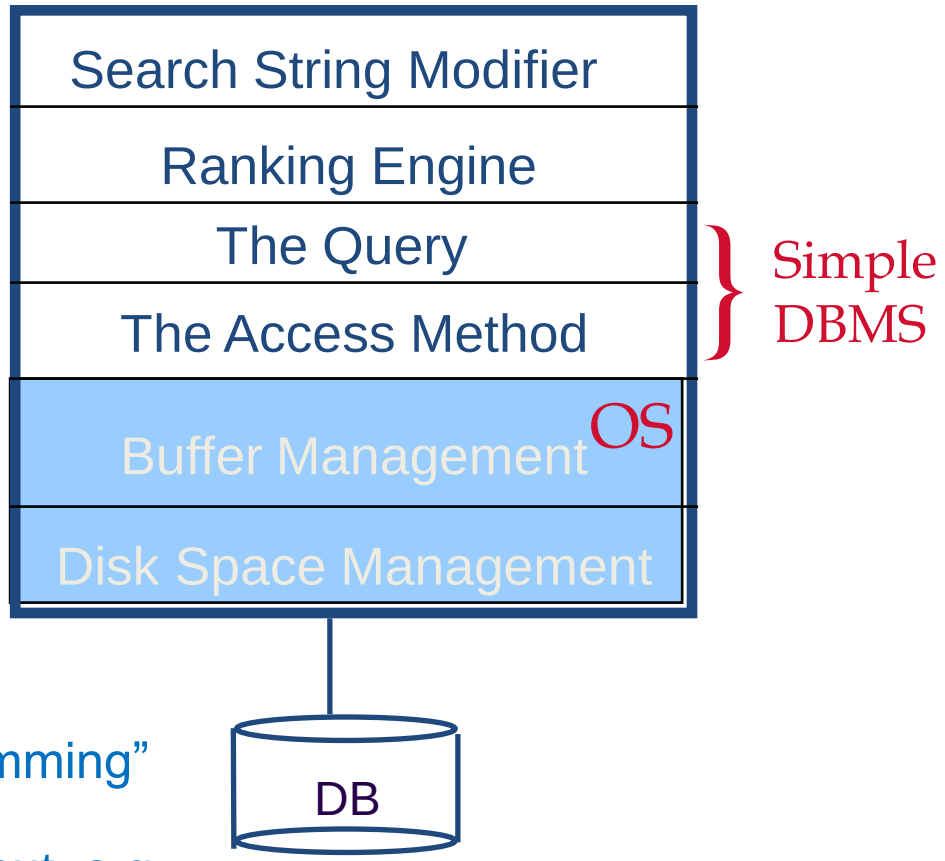


Figure 1.3 Architecture of a DBMS

FYI: A text search engine

- **Less "system" than DBMS**
  - Uses OS files for storage
  - Just one access method
  - One hardwired query
    - regardless of search string
- **Typically no concurrency or recovery management**
  - Read-mostly
  - Batch-loaded, periodically
  - No updates to recover
  - OS a reasonable choice
- **Smarts: text tricks**
  - Search string modifier (e.g. "stemming" and synonyms)
  - Ranking Engine (sorting the output, e.g. by word or document popularity)
  - no semantics: WYGIWIGY



Advantages of a DBMS

- **Data independence**
- **Efficient data access**
- **Data integrity & security**
- **Data administration**
- **Concurrent access, crash recovery**
- **Reduced application development time**
- **So why not use them always?**
  - Expensive/complicated to set up & maintain
  - This cost & complexity must be offset by need
  - General-purpose, not suited for special-purpose tasks (e.g. text search!)

Databases make these forks happy

- **DBMS vendors, programmers**
  - Oracle, IBM, MS, Sybase, NCR, …
- **End users in many fields**
  - Business, education, science, …
- **DB application programmers**
  - Build enterprise applications on top of DBMSs
  - Build web services that run off DBMSs
- **Database administrators (DBAs)**
  - Design logical/physical schemas
  - Handle security and authorization
  - Data availability, crash recovery
  - Database tuning as needs evolve

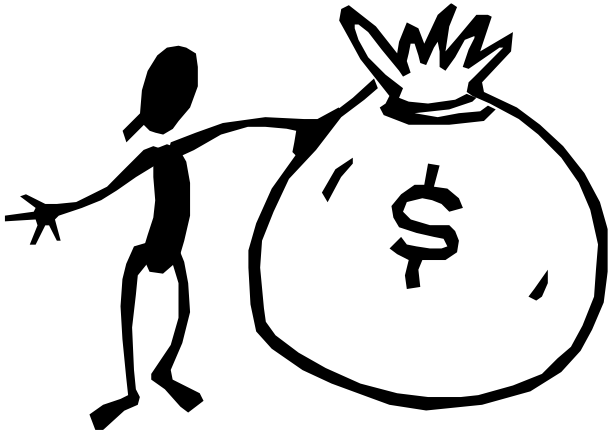
…must understand how a DBMS works

Summary (1)

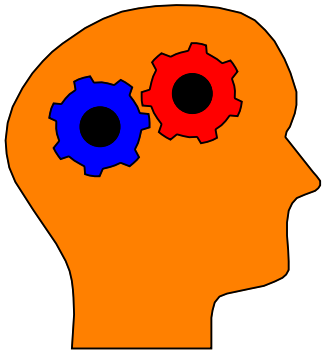
- **DBMS used to maintain, query large datasets**
  - can manipulate data and exploit semantics
- **Other benefits include:**
  - recovery from system crashes
  - concurrent access
  - quick application development
  - data integrity and security
- **Levels of abstraction provide data independence**
  - Key when *dapp/dt << dplatform/dt*
- **In this course we will explore:**
  - How to be a sophisticated user of DBMS technology
  - What goes on inside the DBMS

Summary (cont'd)

- DBAs, DB developers the bedrock of the information economy



- DBMS R&D represents a broad, fundamental branch of the science of computation



What are important in database tech?