Introduction to Databases
《数据库引论》

Lecture 1: Introduction to Database Systems
第1讲：数据库系统简介

周水庚 / Shuigeng Zhou

邮件: sgzhou@fudan.edu.cn 网址：admis.fudan.edu.cn/sgzhou

复旦大学计算机科学技术学院

Content of the Course

- **Part 0: Overview**
  - Lect. 0/1 (Feb. 20) - Ch1: Introduction
- **Part 1 Relational Databases**
  - Lect. 2 (Feb. 27) - Ch2: Relational model (data model, relational algebra)
  - Lect. 3 (Mar. 6) - Ch3&4: SQL (Introduction and intermediate)
  - Lect. 4 (Mar. 13) - Ch5: Advanced SQL
- **Part 2 Database Design**
  - Lect. 5 (Mar. 20) - Ch6: Database design based on E-R model
  - Lect. 6 (Mar. 27) - Ch7: Relational database design (Part I)
  - Lect. 7 (Apr. 3) - Ch7: Relational database design (Part II)
- **Midterm exam: Apr. 10**
- **Part 3 Data Storage & Indexing**
  - Lect. 7 (Apr. 17) - Ch12/13: Storage systems & structures
  - Lect. 8 (Apr. 24) - Ch14: Indexing
- **Part 4 Query Processing & Optimization**
  - May 1, holiday, no classes
  - Lect. 9 (May 8) - Ch15: Query processing
  - Lect. 10 (May 15) - Ch16: Query optimization
- **Part 5 Transaction Management**
  - Lect. 11 (May 22) - Ch17: Transactions
  - Lect. 12 (May 29) - Ch18: Concurrency control
  - Lect. 13 (Jun. 5) - Ch19: Recovery system
  - Lect. 14 (Jun. 5) – Course review

Final exam: 13:00-15:00, Jun. 18

Outline

- Data, information, knowledge and beyond
- Database system
- Data view
- DB languages
- DB design
- DB engine
- Database architecture
- DB user and administrator
- History of DB
- Future directions

Outline

- ☞ Data, information, knowledge and beyond
- Database system
- Data view
- DB languages
- DB design
- DB engine
- Database architecture
- DB user and administrator
- History of DB
- Future directions

Data

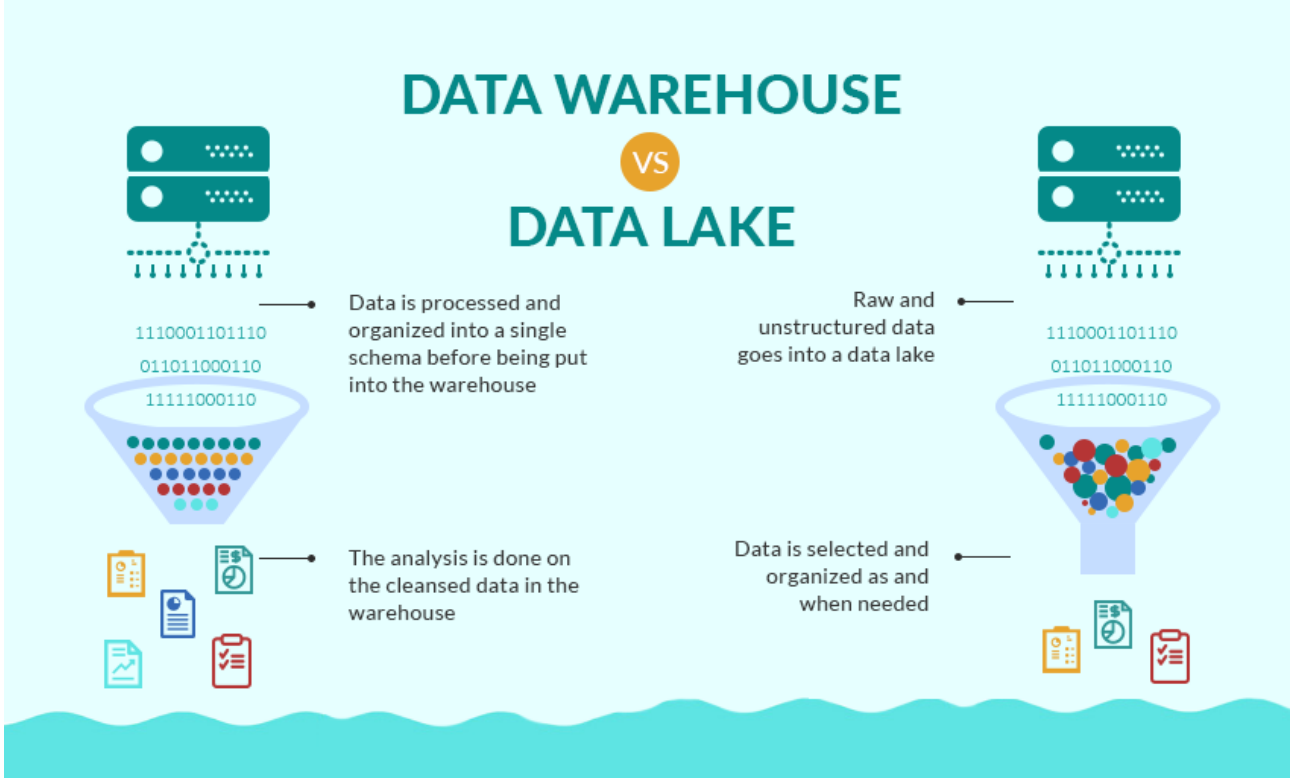
- Data is a set of values of qualitative or quantitative variables.
  - The results of describing or quantifying anything
  - 描述/量化事物的符号/数字记录

- Data is collected, stored and analyzed. It can be visualized using graphs, images and other analysis tools.
- Data has been described as the new oil of Digital Economy (数字经济).
- Data, information, knowledge, intelligence, and wisdom are closely related concepts, but each has its own role in relation to the others, and each term has its own meaning.

Data related concepts and technologies

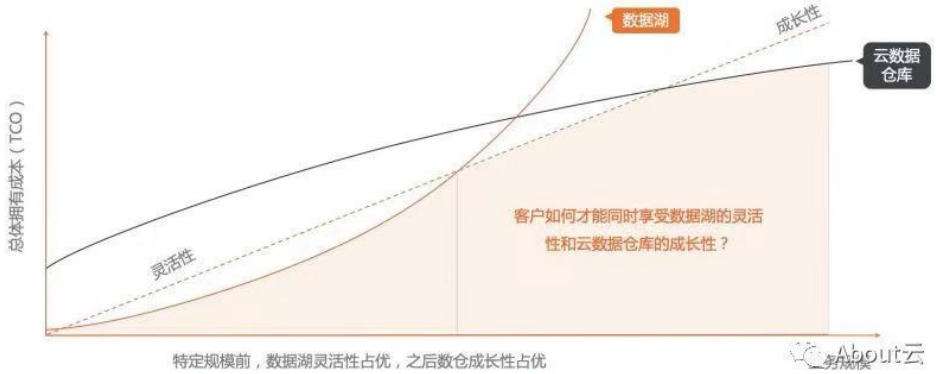
- Data related concepts and technologies
  - Data acquisition, Data modeling, Data structure
  - Data integrity, Data maintenance, Data management, Data governance
  - Data processing, Data analysis, Data mining, Data visualization
  - Data publication, Data protection, Data privacy
  - Big Data, Data science
- Data resources
  - Experimental / biological / environmental / transportation … data
- Data set, Datasheet, Database, Data mart（数据集市）, Data warehouse（数据仓库）, Data lake (数据湖), Data lakehouse (数据湖仓), Data middle office (数据中台)

Data warehouse vs. Data lake



Data warehouse vs. Data lake

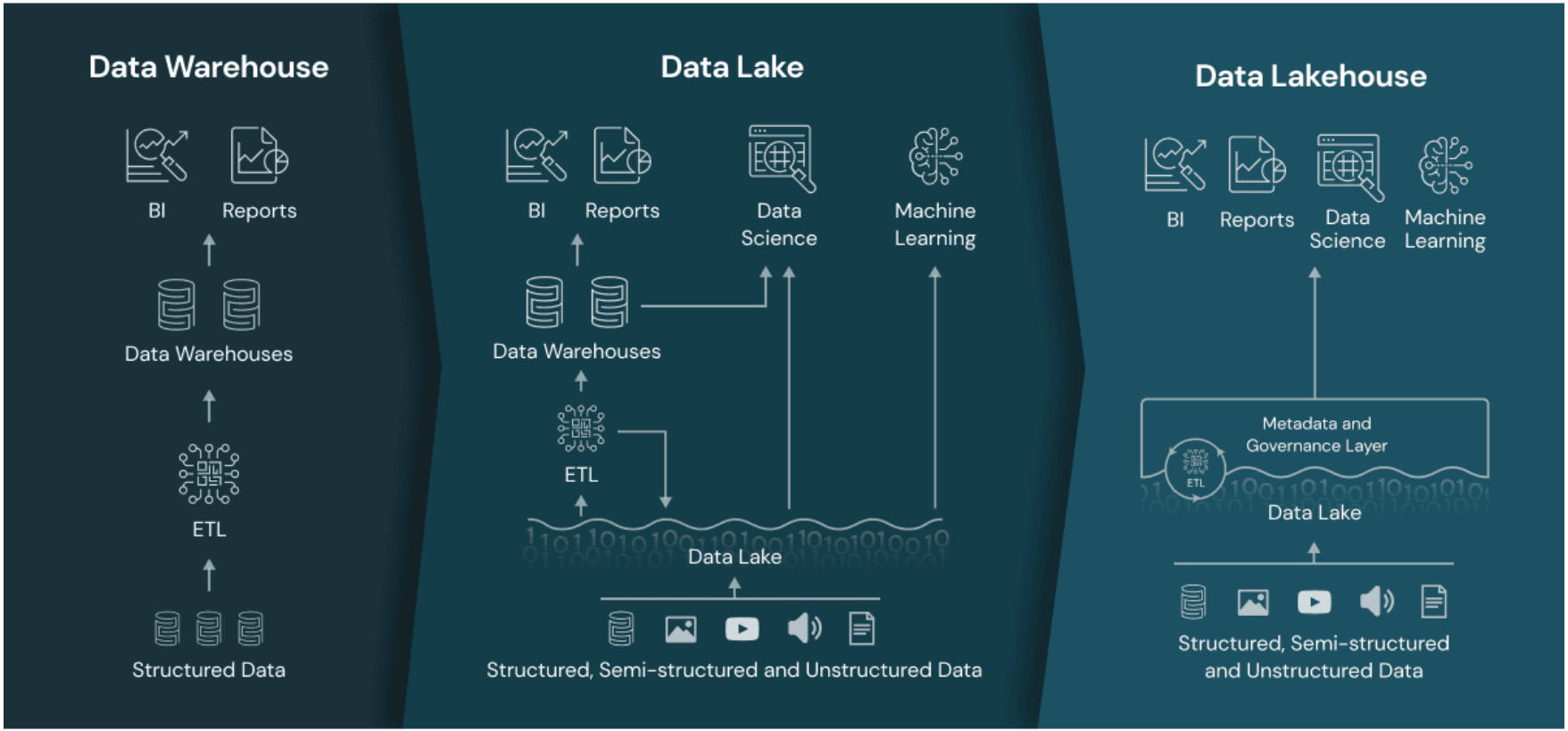
	DATA LAKE	DATA WAREHOUSE
Structure	Raw	Processed
Purpose	Not Yet Determined	Pre-determined
Use Case	Data Scientists	Business Users
Accessibility	Highly Accessible and Quick to Update	Complicated and Costly to Change



Data warehouse vs. Data lake = 餐馆 vs. 菜市场

Data warehouse, Data lake, Lakehouse

Data warehouse vs. Data lake vs. Data lakehouse = 餐馆 vs. 菜市场 vs. 超市



数据中台（Data Middle Office）

- 数据中台是一种集成了**数据管理**、**数据治理**、**数据服务**等多个方面的平台
- 旨在为企业提供稳定、高效、安全的数据支持和服务，从而帮助企业更好地进行数字化转型。数据中台的核心思想是将所有的数据资源和服务整合到一个统一的平台上，实现数据的集中管理和服务
- 从技术角度来看，数据中台是一个基于**云计算**的技术架构，采用分布式、**微服务**、**容器**化等技术手段，实现数据的采集、存储、计算、管理、服务等多个环节

ETLCloud数据中台架构

Big Data

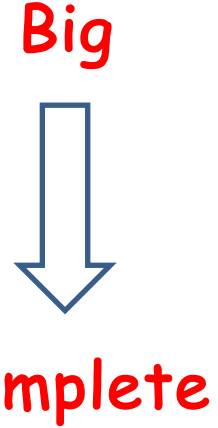
- It's said that the term "**Big data**" in its current use was coined by *Roger Magoulas* @ O'Reilly Media
- There is not a consensus as to how to define big data

"Big data exceeds the reach of commonly used hardware environments and software tools to capture, manage, and process it with in a tolerable elapsed time for its user population."
- *Teradata Magazine article, 2011*

"Big data refers to data sets whose size is beyond the ability of typical database software tools to capture, store, manage and analyze."
- *The McKinsey Global Institute, 2011*

The *Vs* of Big Data

- **3Vs model**
  - high-volume, high-velocity, and/or high-variety
    - Gartner (2012)
- **4Vs models**
  - Volume, velocity, variety and virtual
    - Courtney Lambert (2012)
  - Volume, velocity, variety and veracity
    - IBM (2012)
  - Volume, velocity, variety and value
    - DataStax (2012)
- **5Vs model**
  - Volume, velocity, variety, veracity and value



Big Data: A Joke

某比萨店的电话铃响了，客服人员拿起电话。
客服：XXX比萨店。您好，请问有什么需要我为您服务？
顾客：你好，我想要一份.....
客服：先生，烦请先把您的会员卡号告诉我。
顾客：16846146***。
客服：陈先生，您好！**您是住在泉州路一号12楼120x室**，请问您想要点什么？
顾客：我想要一个海鲜比萨.....
客服：陈先生，海鲜比萨不适合您。
顾客：为什么？
客服：**根据您的医疗记录，你的血压和胆固醇都偏高**。
顾客：那你们有什么可以推荐的？
客服：您可以试试我们的低脂健康比萨。
顾客：你怎么知道我会喜欢吃这种的？
客服：**您上星期一在中央图书馆借了一本《低脂健康食谱》**。
顾客：好。那我要一个家庭特大号比萨，要付多少钱？
客服：**99元，这个足够您一家六口吃了。但您母亲应该少吃，她上个月刚刚做了心脏搭桥手术，还处在恢复期**。
顾客：那可以刷卡吗？
客服：陈先生，对不起。请您付现款，因为**您的信用卡已经刷爆了，您现在还欠银行4807元，而且还不包括房贷利息**
顾客：那我先去附近的提款机提款。
客服：陈先生，**根据您的记录，您已经超过今日提款限额**。
顾客：算了，你们直接把比萨送我家吧，家里有现金。你们多久会送到？
客服：大约30分钟。如果您不想等，可以自己骑车来。
顾客：为什么？
客服：**根据我们全球定位系统的车辆行驶自动跟踪系统记录。您登记有一辆车号为XX-748的摩托车，而目前您正在解放路东段华联商场右侧骑着这辆摩托车**。
顾客当即晕倒.....

家庭住址
医疗记录
医疗记录
借阅记录
借贷记录
定位跟踪

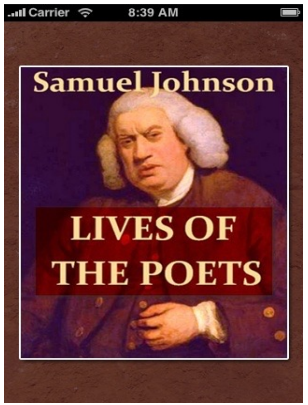
Information & Information Retrieval

- **Information**
  - Data with meaning or semantics
  - Provide the answer to a question of some kind or resolves uncertainty. It is thus related to data and knowledge, as data represents values attributed to parameters, and knowledge signifies understanding of real things or abstract concepts.

- **Information retrieval (IR)**
  - The activity of obtaining information from a collection of information resources. Searches can be based on full-text or other content-based indexing.

Knowledge ⇨ knowledge base + inference engine = Expert System

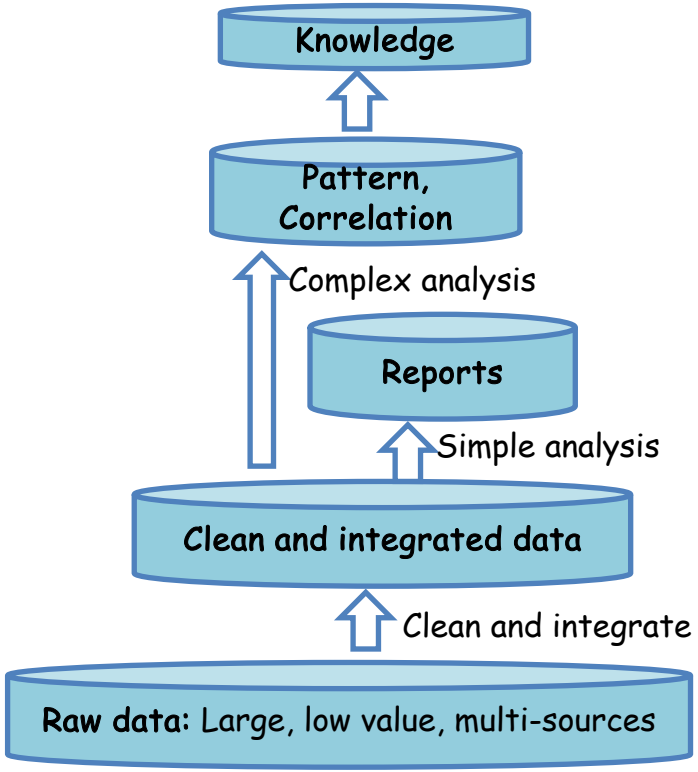
- **Knowledge**
  - Information with value
  - Familiarity, awareness, or understanding of someone or something, such as facts, information, descriptions, or skills. Knowledge is acquired through experience or education by perceiving, discovering, or learning.
  - In philosophy, the study of knowledge is called epistemology (认识论). Theories of knowledge, Communicating knowledge, Situated knowledge, Partial knowledge, Scientific knowledge, Religious meaning of knowledge.

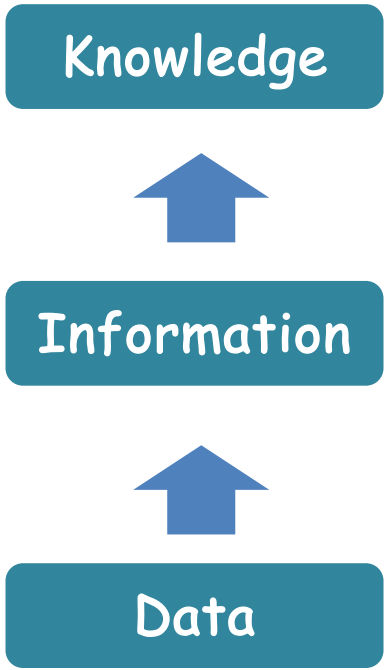


"Knowledge is of two kinds: we know a subject ourselves, or we know where we can find information upon it."

Samuel Johnson (1709-1784)

Data, Information and Knowledge





Intelligence VS. Artificial Intelligence

- The ability to acquire and apply knowledge and skills

- **Intelligence** has been defined in many ways to include the capacity for logic, understanding, self-awareness, learning, emotional knowledge, reasoning, planning, creativity, and problem solving. It can be more generally described as the ability to perceive or infer information, and to retain it as knowledge to be applied towards adaptive behaviors within an environment or context.

- **Artificial intelligence (or AI)** is both the intelligence of machines and the branch of computer science which aims to create it, through "**the study and design of intelligent agents**" or "**rational agents**", where an intelligent agent is a system that perceives its environment and takes actions which maximize its chances of success.

Intelligence vs. Artificial Intelligence

- General intelligence or strong AI
  - Intelligence like human's intelligence
  - Adaptive to environment, creativity
  - has not yet been achieved, but some great advances have made (e.g. ChatGPT)

- Researchers have been trying to make machines exhibit reasoning, planning, learning, communication, perception, and the ability to move and to manipulate objects, closely simulating human brain.

- DeepMind: AlphaGo, Alphafold2, ……
- OpenAI: ChatGPT/GPT3/3.5/4, SORA, o1……
- DeepSeek

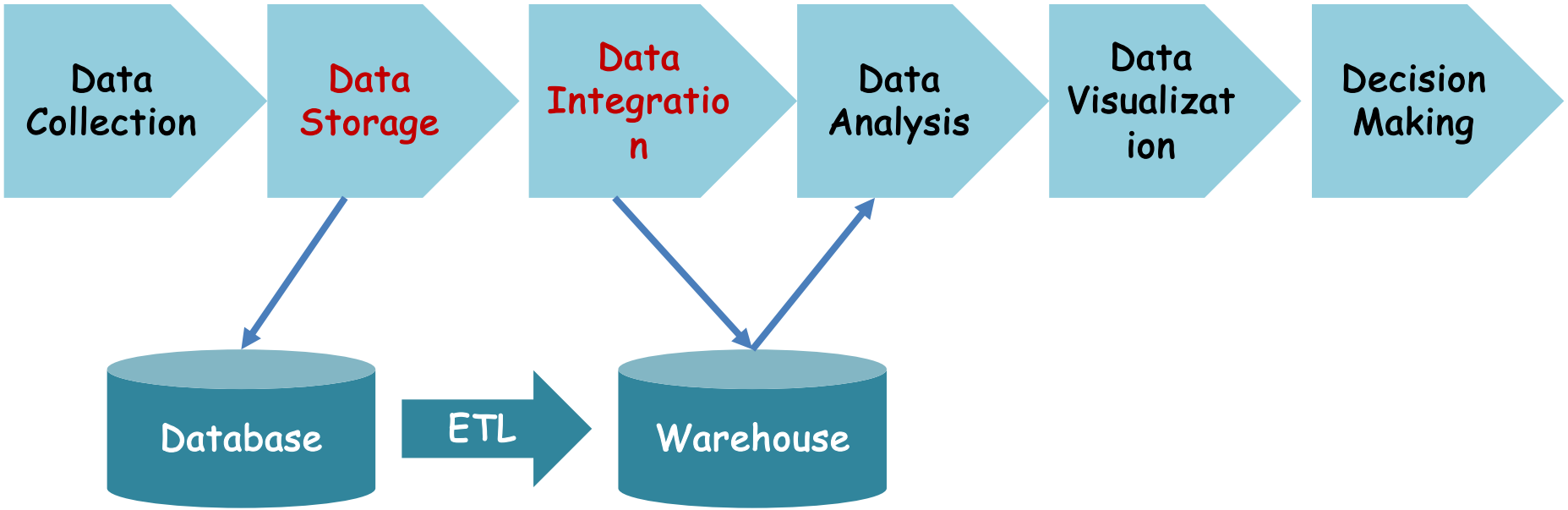
Intelligence vs. Wisdom

- Intelligence is the ability to learn and acquire and apply knowledge
  - deals more with practical facts
- Wisdom is intelligence garnered through experience. We learn through our experiences and we use this knowledge to make decisions
  - based on right and wrong
- Wisdom is high-level intelligence (?)
- Wisdom = IQ+EQ (?)

Data Mining/Warehousing, Query Processing and Information Retrieval

- **Data Mining**: finding hidden, nontrivial, previously unknown and useful knowledge from massive data
  - Rules/patterns
- **Data Warehousing**
  - The process of building data warehouse
- **Query Processing**
  - Extracting desirable data from databases
- **Information Retrieval**
  - Searching the matching information from unstructured textual data

Data Processing Flow



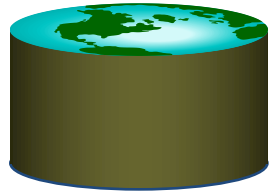
ETL: Extract, Transform and Load

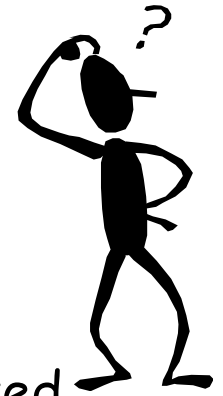
Outline

- Data, information, knowledge and beyond
- ☞ Database system
- Data view
- DB languages
- DB design
- DB engine
- Database architecture
- DB user and administrator
- History of DB
- Future directions

What Is a Database

- **An integrated collection of data**
  - The data is structured and interrelated
  - The data is integrated
- **Model real-world enterprises or organizations**
  - **Entities** (e.g., students, courses)
  - **Relationships** (e.g. Li is taking Database and OS)
- **Databases touch various aspects of our lives**
  - People interact with databases from indirect way (printed report) to direct way (teller machines, online shopping)





Database Applications

- **Various applications**
  - Banking: all transactions
  - Ticket systems: 12306
  - E-commerce platforms: Toubao, Jingdong, ……
  - Airlines: reservations, schedules
  - Universities: registration, grades, students
  - Sales: customers, products, purchases
  - Online retailers: order tracking, customized recommendations
  - Manufacturing: production, inventory, orders, supply chain
  - Human resources: employee records, salaries, tax deductions
  - ……

Example: University Database

- **Application scenarios**
  - Add new students, instructors and courses
  - Register students for courses, and generate class rosters（名单）
  - Assign grades to students, calculate grade point averages (GPA) and generate transcripts

- **In the early days, applications were built directly on top of file systems**

Drawbacks of using File Systems to Store Data

- Data redundancy (冗余) and inconsistency (不一致)
  - Multiple file formats (different programmers/languages/structures), duplication of information in different files
- Difficulty in accessing data
  - Need to write a new program to carry out each new task
- Data isolation (隔离)
  - Multiple files and formats
- Integrity (完整性) problems
  - Integrity constraints (e.g., account balance > 0) become "buried" in program code rather than being stated explicitly
  - Hard to add new constraints or change existing ones

Drawbacks of using File Systems to Store Data

- **Atomicity (原子性) of updates**
  - Failures may leave database in an inconsistent state with partial updates carried out
    - E.g., transfer of funds from one account to another should either complete or not happen at all
- **Concurrent access (并发访问) by multiple users**
  - Concurrent access needed for better performance
  - Uncontrolled concurrent accesses can lead to inconsistencies
    - E.g., two people reading a balance (say 100) and updating it by withdrawing money (say 50 each) at the same time
- **Security (安全) problems**
  - Hard to provide user access to some, but not all, data

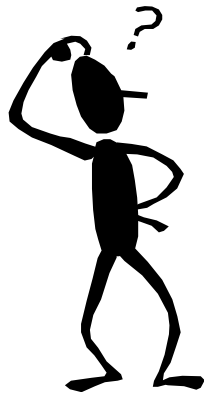
Database Management System (DBMS)

- **DBMS**
  - A software package designed to store and manage databases
- **Functions of DBMS**
  - Manages a large amount of data
  - Supports efficient and concurrent access to a large amount of data
  - Supports secure, atomic access to a large amount of data: two people editing the same file – last to write "wins"
- **Advantages of DBMS**
  - Data independence and efficient access
  - Reduce application development time
  - Data integrity and security
  - Uniform data administration
  - Concurrent access
  - Recovery from crashes

Is a File System a DBMS?

- **Experiment 1:**
  - You and your partner are editing the same file.
  - You both save it at the same time.
  - Whose changes survive?

  A) Yours B) Partner's C) Both D) Neither E) ???

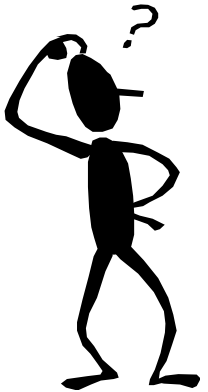


- **Experiment 2:**
  - You're updating a file.
  - The power goes out.
  - Which of your changes cannot survive?

  A) All B) None C) All since last save D) ???

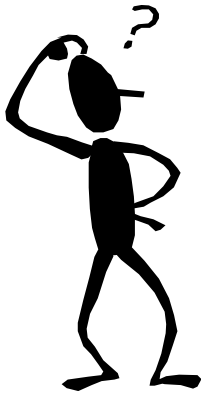
Is a Traditional IR System a DBMS?

- **Traditional IR (Information Retrieval, 信息检索) system**
  - Mainly for data retrieval, not data update
  - Store/manage free structured or semi-structured document
  - Support semantic-oriented matching (语义匹配)
  - Consider no data consistency, and concurrency is easy to be implemented



- **DBMS**
  - Support data access and update
  - Store and manage structured data (RDBMS)
  - Require declarative（陈述性）query language support
  - Support data consistency, atomic transaction, concurrency control and failure recovery

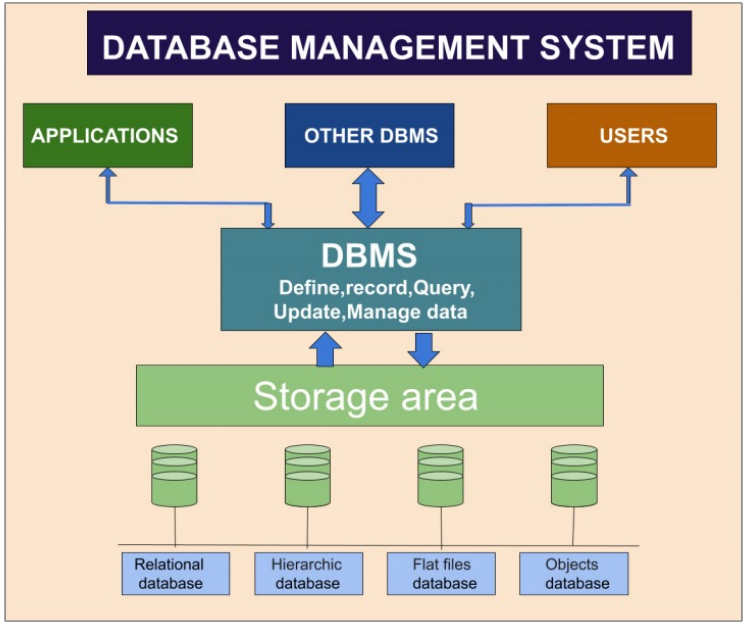
Is Web Search Engine a DBMS?



- Keyword-based search for pages
- Data is mostly unstructured and untyped
- Search only
  - can't modify the data
  - can't get summaries（概括）, complex combinations of data
- Few guarantees provided for freshness of data, consistency across data items, fault tolerance, etc.
- Web sites (e.g., e-commerce 电子商务) typically have a DBMS in the backend to provide these functions

Database Systems

- A Database System (DBS) contains the following components
  - Hardware platform (PC, Workstation, Cluster, Mainframe, etc.)
  - DBMS
  - A (number of) database(s)
  - Human interface that is both convenient and efficient to use



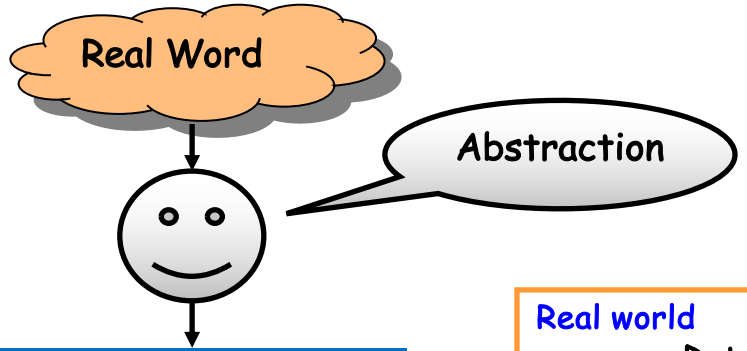
What's the Intellectual Content?

- Representing information
  - data modeling
- Languages and systems for querying data
  - complex queries with real semantics over massive data sets
- Concurrency control for data manipulation
  - controlling concurrent access
  - ensuring transactional semantics
- Reliable data storage
  - maintain data semantics even if you pull the plug

Outline

- Data, Information, Knowledge and beyond
- Database system
- ☞ **Data view**
- DB languages
- DB design
- DB engine
- Database architecture
- DB user and administrator
- History of DB
- Future directions

Data Abstraction



Information world

Conceptual Model

Machine world

Data model supported by DBMS

Real world ⇨ Conceptual model
Database Designers

Conceptual model ⇨ Logical model
Database Designers

Logical model ⇨ Physical model
DBMS

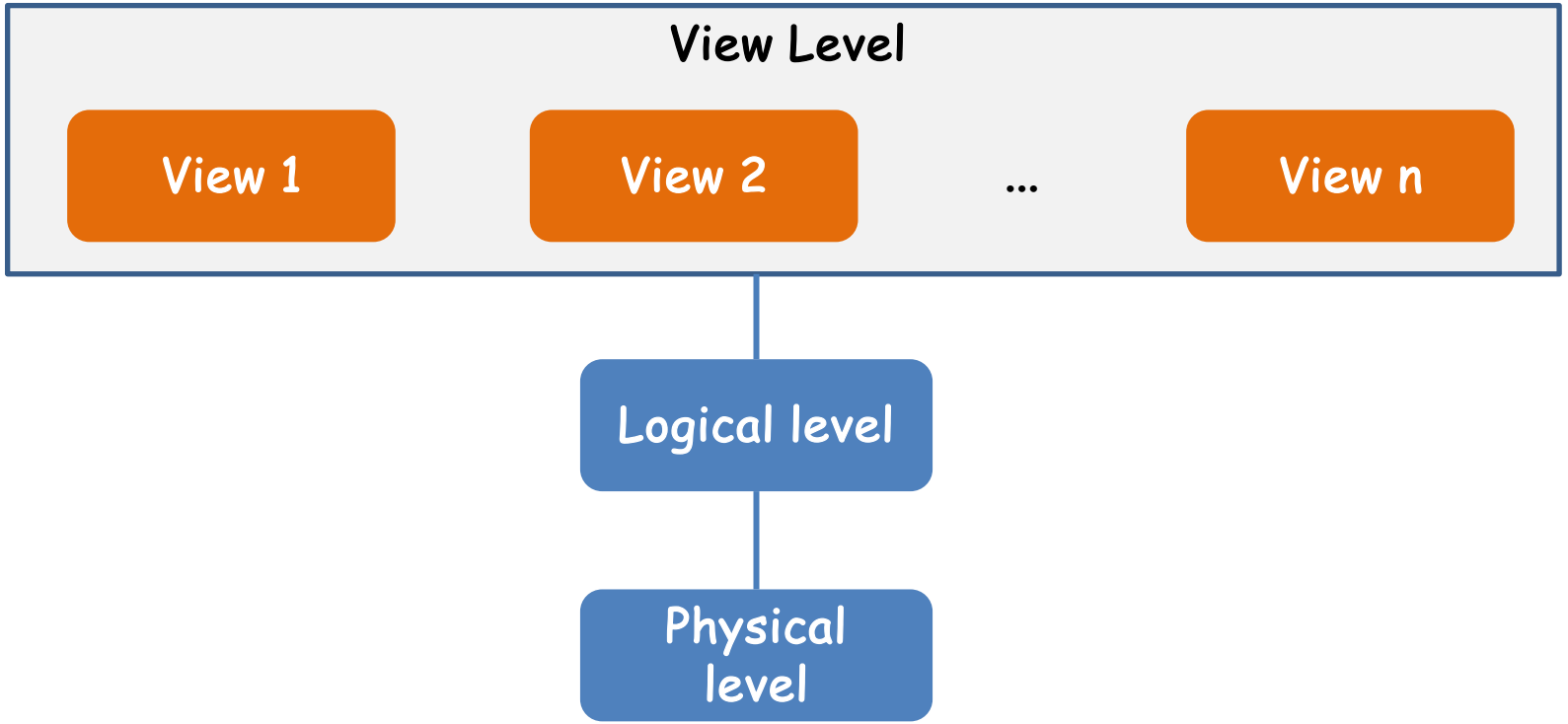
Levels of Abstraction

- **Physical level**
  - Describe **how** a record is stored on storage devices.
- **Logical level**
  - Describe **what** data stored in database, and the relationships among the data.
  - Although implementation of the simple structures at the logical level may involve complex physical-level structures, the user of the logical level does not need to be aware of this complexity. This is referred to as **physical data independence**.

```
type customer = record
        customer_id : string;
        customer_name : string;
        customer_street : string;
        customer_city : integer;
end;
```

- **View level**
  - Many users of the database system do not need all this information; instead, they need to access only a part of the database. The view level of abstraction simplifies their interaction with the system.
  - Views can also hide information (such as an employee's salary) for security purposes.

View of Data



Instances and Schemas

- Similar to variables and types in programming languages
- Schema (模式): the logical structure of the database
  - E.g., the database consists of information about a set of customers and accounts, and the relationship between them
  - Analogous to type information of a variable in a program
    - Physical schema (物理模式): database design at the physical level
    - Logical schema (逻辑模式): database design at the logical level
    - External schema (外模式): user defined schema at the app level (view level)
- Instance (实例): the actual content of the database at a particular point of time, analogous to the value of a variable

Instances and Schemas

- Schema – the logical structure of the database
  - Physical schema: internal schema, storage schema
  - Logical schema: schema
  - External schema: user's schema, subschema
- Physical data independence – the ability to modify the physical schema without changing the logical schema
  - In general, the interfaces between the various levels and components should be well defined so that changes in some parts do not seriously influence others
- Logical data independence – the ability to modify the logical schema without changing the external schema

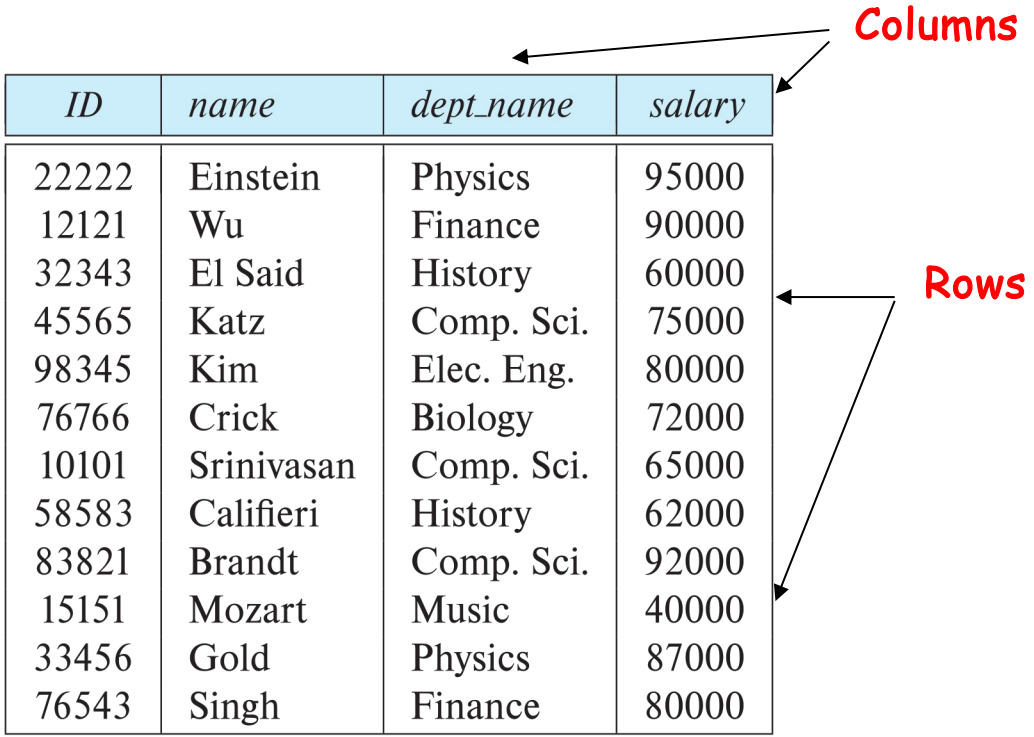
Data Models

A collection of **tools** for describing data structures, data relationships, data semantics, and data constraints

- **Relational model**
  - Tables as relations to represent data and their relationships
  - Record-based model
- **Entity-Relationship data model (mainly for database design)**
  - Entity-relationship (E-R) data model
- **Object-based data models (Object-oriented/Object-relational)**
  - Extending the E-R model with notions of encapsulation, methods(functions), and object identity
- **Semi-structured data model (JSON, XML)**
- **Other older models:**
  - Network model (e.g. Honeywell IDS)
  - Hierarchical model (e.g. IBM IMS)

Relational Model

- Example of tabular data in the relational model



ID	*name*	*dept_name*	*salary*
22222	Einstein	Physics	95000
12121	Wu	Finance	90000
32343	El Said	History	60000
45565	Katz	Comp. Sci.	75000
98345	Kim	Elec. Eng.	80000
76766	Crick	Biology	72000
10101	Srinivasan	Comp. Sci.	65000
58583	Califieri	History	62000
83821	Brandt	Comp. Sci.	92000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
76543	Singh	Finance	80000

(a) The *instructor* table

A Sample of Relational Database

ID	*name*	*dept_name*	*salary*
22222	Einstein	Physics	95000
12121	Wu	Finance	90000
32343	El Said	History	60000
45565	Katz	Comp. Sci.	75000
98345	Kim	Elec. Eng.	80000
76766	Crick	Biology	72000
10101	Srinivasan	Comp. Sci.	65000
58583	Califieri	History	62000
83821	Brandt	Comp. Sci.	92000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
76543	Singh	Finance	80000

(a) The *instructor* table

dept_name	*building*	*budget*
Comp. Sci.	Taylor	100000
Biology	Watson	90000
Elec. Eng.	Taylor	85000
Music	Packard	80000
Finance	Painter	120000
History	Painter	50000
Physics	Watson	70000

(b) The *department* table

Object-Relational Data Models

- Extend the relational data model by including
  - Inheritance (继承)/ encapsulation (封装) /methods(functions) /object identity
- Allow attributes of tuples to have complex types, including non-atomic values such as nested relations
- Preserve relational foundations, in particular the declarative access to data, while extending modeling power
- Provide upward compatibility (向上兼容) with existing relational languages

XML: eXtensible Markup Language

- Defined by WWW Consortium (W3C, 万维网联盟)
- Originally intended as a document markup language not a database language
- The ability to specify new tags, and to create nested tag structures made XML a great way to exchange data, not just documents
- XML has become the basis for various data interchange formats
- A wide variety of tools is available for parsing, browsing and querying XML documents/data

XML: An Example

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<CDLIST>
 <CD artist="Yung Jo Yee, Joey">
  <TITLE>Ten Most Wanted</TITLE>
  <RELEASE>Feb 26, 2006</RELEASE>
  <PRICE>$20</PRICE>
 </CD>
 <CD artist="Twins">
  <TITLE>The Missing Piece</TITLE>
  <RELEASE>Feb 20, 2006</RELEASE>
  <PRICE>$10</PRICE>
 </CD>
 <CD artist="Twins">
  <TITLE>2006 Concert Live</TITLE>
  <RELEASE>Feb 20, 2006</RELEASE>
  <PRICE>$10</PRICE>
 </CD>
 <CD artist="Justin">
  <TITLE>No Protection</TITLE>
  <RELEASE>March 24, 2006</RELEASE>
  <PRICE>$15</PRICE>
 </CD>
</CDLIST>
```

Info about one CD

Outline

- Database system
- Data view
- ☞ **DB languages**
- DB design
- DB engine
- Database architecture
- DB user and administrator
- History of DB
- Future directions

Database Language

- A database system provides a data-definition language(DDL) to specify the database schema and a data-manipulation language (DML) to express database queries and updates.

- **Data-definition language (DDL, 数据定义语言)**
  - Specify the database schema
  - SQL provides a rich DDL that allows one to define tables, integrity constraints, assertions, etc.
- **Data-manipulation language (DML, 数据操纵语言)**
  - Express database queries and updates

Data Definition Language (DDL)

- Defining the database schema
  - Example:
```
create table department (
  dept_name    char(20),
  building     char(15)
  budget       numeric(12,2));
```
- DDL compiler generates a set of tables stored in a **data dictionary**
- **Data dictionary** contains **metadata** (i.e., data about data)
  - Database schema
  - Data storage and definition language
    - Specifies the storage structure and access methods
  - **Consistency constraints**
    - Domain constraints
    - Referential integrity (references constraint in SQL)
    - Assertions
    - Authorization

Data Manipulation Language (DML)

- DML is for accessing and manipulating the data organized by the appropriate data model
  - DML also known as query language
  - Retrieval/ Insertion/ Deletion/ Modification
- Two classes of languages
  - **Procedural** – user specifies what data is required and how to get those data
  - **Nonprocedural** (Declarative, 陈述性) – user specifies what data is required without specifying how to get those data
- **SQL** is the most widely used **non-procedural** query language

SQL (Structured Query Language)

- **SQL:** widely used **non-procedural** language
  - E.g., find all instructors in the department of History

```
select   instructor.name
from     instructor
where    instructor.dept_name = 'History'
```

  - E.g., find the IDs of all instructors from those departments with a budget larger than 95000

```
select   instructor.ID
from     instructor, department
where    instructor.dept_name = department.dept_name and
         department.budget > 95000
```

- Application programs generally access databases through one of
  - Language extensions to allow embedded SQL
  - Application program interface (e.g., ODBC/JDBC) which allow SQL queries to be sent to a database

Outline

- Data, information, knowledge and beyond
- Database system
- Data view
- DB languages
- ☞ **DB design**
- DB engine
- Database architecture
- DB user and administrator
- History of DB
- Future directions

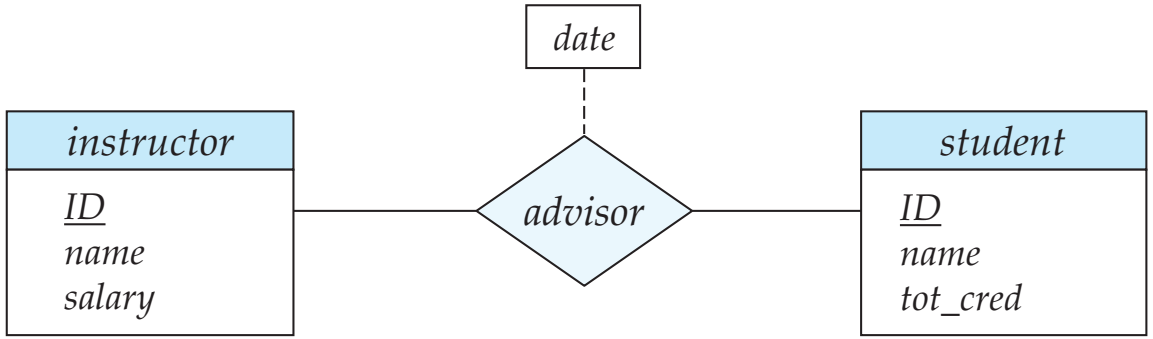
Database Design

The process of designing the structure of the database:

- **Conceptual design**
  - Decide what should be covered and their relations
  - Deciding on the database conceptual schema via ER model
  - **Business decision** – What attributes should we record in the database?
- **Logical design** –Decide the database schema, find a "**good**" collection of relation schemas.
  - **Computer science decision** – What relation schemas should we have and how should the attributes be distributed among the various relation schemas?
    - Employ a set of algorithms (collectively known as **normalization**) that takes as input the set of all attributes and generates a set of tables
- **Physical design** – Decide the physical layout of the database

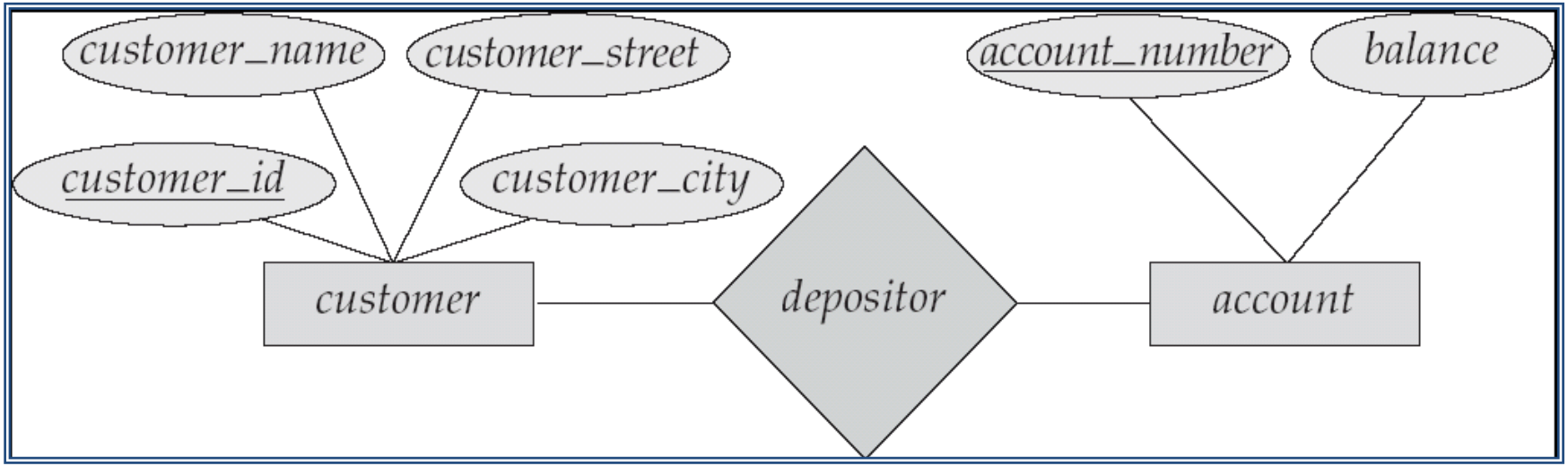
Entity-Relationship Model

- Model an enterprise as a collection of **entities** and **relationships**
  - **Entity**: a "thing" or "object" in the enterprise that is distinguishable from other objects
    - Described by a set of attributes
  - **Relationship** (diamond): an association among several entities



The Entity-Relationship Model

- Represented diagrammatically by an *entity-relationship diagram*:



Relational Model

- Example of tabular data in the relational model

Attributes

Customer-id	customer-name	customer-street	customer-city	account-number
192-83-7465	Johnson	Alma	Palo Alto	A-101
019-28-3746	Smith	North	Rye	A-215
192-83-7465	Johnson	Alma	Palo Alto	A-201
321-12-3123	Jones	Main	Harrison	A-217
019-28-3746	Smith	North	Rye	A-201

A Sample Relational Database

customer-id	*customer-name*	*customer-street*	*customer-city*
192-83-7465	Johnson	12 Alma St.	Palo Alto
019-28-3746	Smith	4 North St.	Rye
677-89-9011	Hayes	3 Main St.	Harrison
182-73-6091	Turner	123 Putnam Ave.	Stamford
321-12-3123	Jones	100 Main St.	Harrison
336-66-9999	Lindsay	175 Park Ave.	Pittsfield
019-28-3746	Smith	72 North St.	Rye

(a) The *customer* table

account-number	*balance*
A-101	500
A-215	700
A-102	400
A-305	350
A-201	900
A-217	750
A-222	700

(b) The *account* table

customer-id	*account-number*
192-83-7465	A-101
192-83-7465	A-201
019-28-3746	A-215
677-89-9011	A-102
182-73-6091	A-305
321-12-3123	A-217
336-66-9999	A-222
019-28-3746	A-201

(c) The *depositor* table

Data Manipulation Language

ID	*name*	*dept_name*	*salary*
22222	Einstein	Physics	95000
12121	Wu	Finance	90000
32343	El Said	History	60000
45565	Katz	Comp. Sci.	75000
98345	Kim	Elec. Eng.	80000
76766	Crick	Biology	72000
10101	Srinivasan	Comp. Sci.	65000
58583	Califieri	History	62000
83821	Brandt	Comp. Sci.	92000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
76543	Singh	Finance	80000

(a) The *instructor* table

dept_name	*building*	*budget*
Comp. Sci.	Taylor	100000
Biology	Watson	90000
Elec. Eng.	Taylor	85000
Music	Packard	80000
Finance	Painter	120000
History	Painter	50000
Physics	Watson	70000

(b) The *department* table

```
select instructor.name
from instructor
where instructor.dept_name='History';
```

```
select instructor.ID, department.dept_name
from instructor, department
where instructor.dept_name=department.dept_name and
department.budget > 95000;
```

Outline

- Data, information, knowledge and beyond
- Database system
- Data view
- DB languages
- DB design
- ☞ DB engine
- Database architecture
- DB user and administrator
- History of DB
- Future directions

Database engine

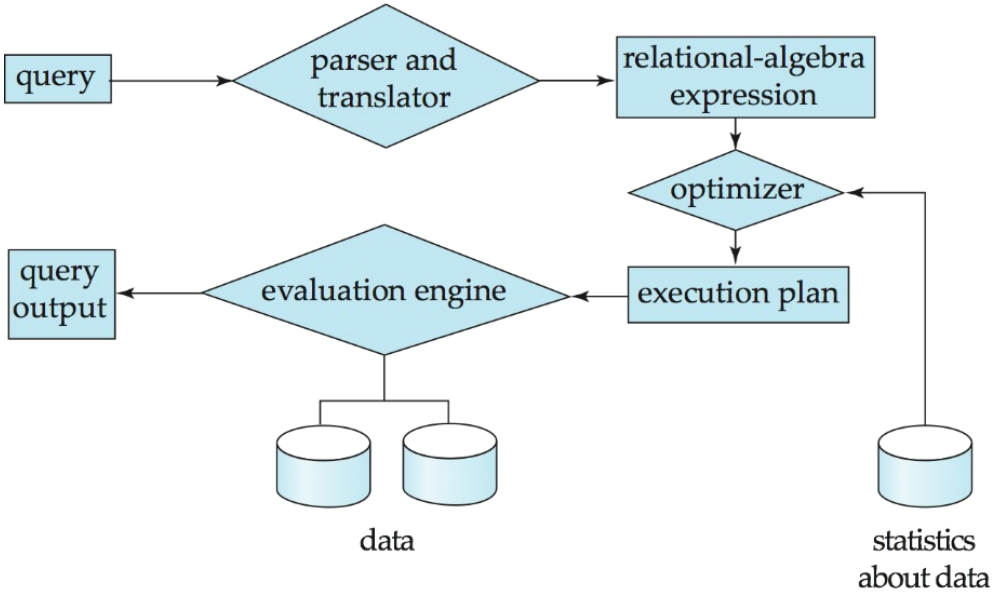
- **Storage manager**
  - minimize the need to move data between disk and main memory
- **Query processor**
  - translate updates and queries written in a nonprocedural language, at the logical level, into an efficient sequence of operations at the physical level such that helps the database system to simplify and facilitate access to data
- **Transaction manager**
  - ensures that the database remains in a consistent (correct) state despite system failures, and that concurrent transaction executions proceed without conflicting

Storage Management

- **Storage manager** is a program module that provides the interface between the low-level data stored in the database and the application programs and queries submitted to the system.
- The storage manager is responsible for the following tasks:
  - Interaction with the file manager
    - The raw data are stored on the disk using the file system provided by the operating system. The storage manager translates the various DML statements into low-level file-system commands.
  - Efficient storing, retrieving and updating of data
- **Issues:**
  - Storage access
  - File organization
  - Indexing and hashing

Query Processor Components

- **DDL interpreter**
  - Parsing and translation: interprets DDL statements and records the definitions in the data dictionary.
- **DML compiler**
  - Translates DML statements in a query language into an evaluation plan consisting of low-level instructions that the query evaluation engine understands.
  - Query optimization
- **Query evaluation engine**
  - Executes low-level instructions generated by the DML compiler



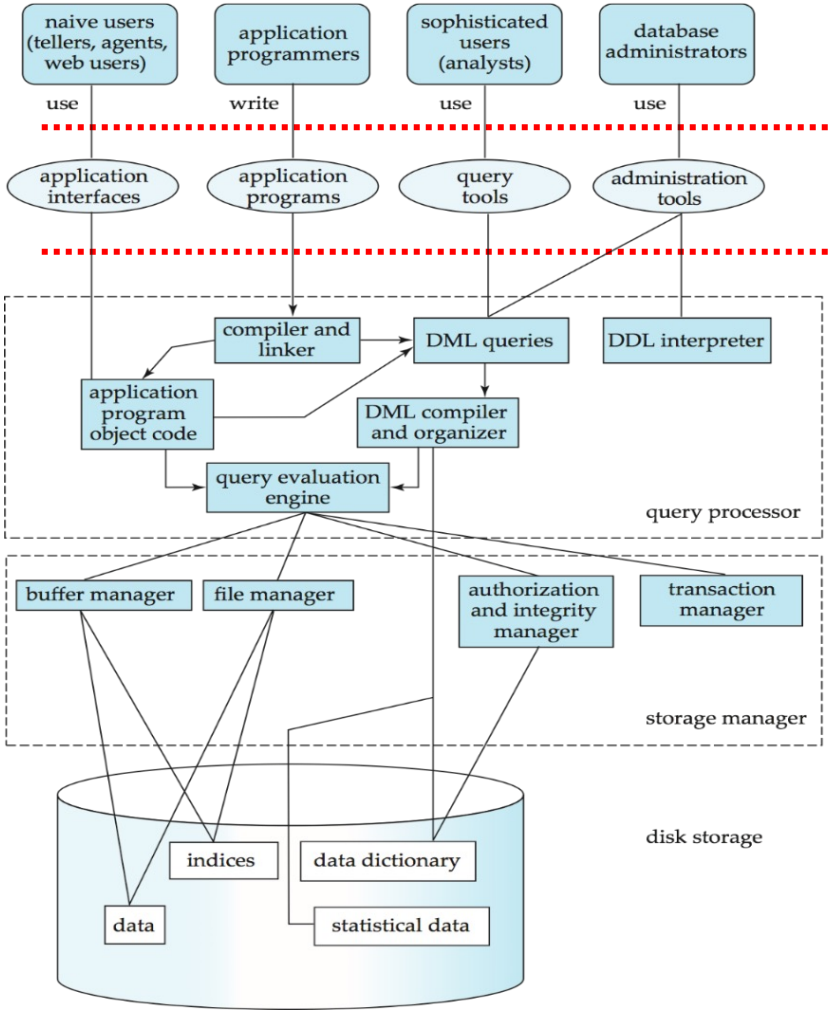
Transaction Management

- **Transaction**
  - a collection of operations that perform a single logical function in a database application
  - **Atomicity/consistency/isolation/durability (ACID)**
- **Transaction manager**
  - Ensure that the database remains in a consistent (correct) state despite system failures (e.g., power failures and operating system crashes) and transaction failures.
- **Concurrency control manager**
  - Control the interaction among the concurrent transactions to ensure the consistency of the database.
- **Recovery manager**
  - Failure recovery

Outline

- Data, information, knowledge and beyond
- Database system
- Data view
- DB languages
- DB design
- DB engine
- ☞ **Database architecture**
- DB user and administrator
- History of DB
- Future directions

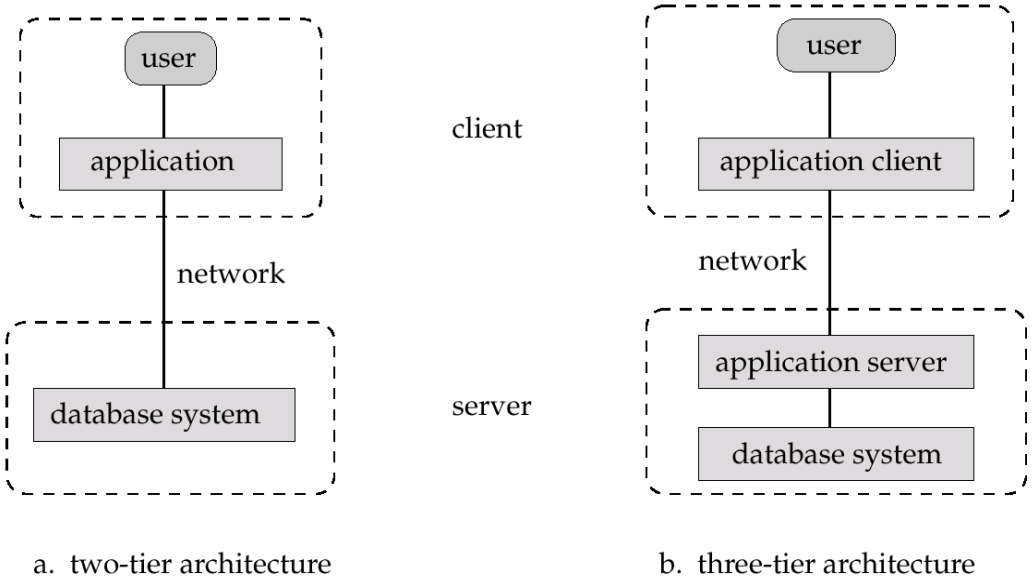
Database System Structure



Database Architecture

- The architecture of a database systems is greatly influenced by the underlying computer system on which the database is running:
  - Centralized
  - Client-server / Browser-server
  - Parallel (multi-processor)
  - Distributed

Application Architectures



a. two-tier architecture

b. three-tier architecture

- **Two-tier architecture**: e.g., client programs using ODBC/JDBC to communicate with a database
- **Three-tier architecture**: e.g., web-based applications, and applications built using "middleware"

Outline

- Database system
- Data view
- DB languages
- DB design
- DB engine
- Database architecture
- ☞ **DB user and administrator**
- History of DB
- Future directions

Database Users

- Users are differentiated by the way they interact with the system
  - **Naive users**（无经验用户）
    - Use application programs that have been developed previously
    - E.g., people accessing database over the web, bank tellers, various apps
  - **Application programmers** (应用程序员)
    - Interact with system through DML calls
  - **Sophisticated users**（熟练用户）
    - Form requests in a database query language
  - **Specialized users** （专业用户）
    - Write specialized database applications that do not fit into the traditional data processing framework
    - computer-aided design systems, knowledge base and expert systems, systems that store data with complex data types (e.g., graphics data and audio data), and environment-modeling systems

People Working with DBMS

- **End users**
  - Query/update databases through application user interfaces (e.g., Amazon.com, 同心云, 本研一体化系统, etc.)
- **Database designers**
  - Design database "schema" to model the real world
- **Database application developers**
  - Build applications that interact with databases
- **Database administrators** (a.k.a. **DBA**)
  - Load, back up, and restore data
  - Fine-tune databases for better performance
- **DBMS implementors**
  - Develop the DBMS or specialized data management software
  - Implement new techniques for query processing and optimization inside DBMS

Database Administrator (DBA)

- Coordinate all the activities of the database system, and have a good understanding of the enterprise's information resources and needs, as well as DBMS
- DBA's duties
  - Schema definition
  - Storage structure and access method definition
  - Schema and physical organization modification
  - Granting of authorization for data access
  - Routine maintenance
  - Specify integrity constraints
  - Act as liaison with users
  - Monitor performance, and respond to changes in requirements

Outline

- Data, information, knowledge and beyond
- Database system
- Data view
- DB languages
- DB design
- DB engine
- Database architecture
- DB user and administrator
- ☞ History of DB
- Future directions

History of Database Systems

- **1950s and early 1960s:**
  - Data processing using magnetic tapes for storage
    - Tapes provide only sequential access
  - Punched cards for input
- **Late 1960s and 1970s:**
  - Hard disks allow direct access to data
  - Network and hierarchical data models in widespread use
  - **Edgar F Codd defines the relational data model**
    - Won the ACM Turing Award for this work
    - IBM Research begins System R prototype
    - UC Berkeley begins Ingres prototype
  - High-performance (for the era) transaction processing
- **1980s:**
  - Relational prototypes evolve into commercial systems
    - **SQL becomes industrial standard**
  - Parallel and distributed database systems
  - Object-oriented database systems
- **1990s:**
  - Large decision support and data-mining applications
  - Large multi-terabyte data warehouses
  - Emergence of Web commerce
- **2000s**
  - **Big data** storage systems
    - Google BigTable, MapReduce, Yahoo PNuts, Amazon
    - "NoSQL" systems.
  - Big data analysis: beyond SQL
- **2010s**
  - Massive distributed database systems
  - Multi-core main-memory databases
  - Automated database administration
  - Streaming, mobile, trusted, uncertain data management
  - Cloud-native databases
- **2020s**
  - DB+AI (LLMs)

Milestones in DBMS History (1)

- **Factor out data management** functionalities from applications and standardize these functionalities
  - CODASYL (Conference on Data Systems Language) standard (circa 1960's)
  - Developed the first database management system IDS (Integrated Data Store)
  - Charles Bachman got a Turing award in 1973

Charles Bachman
(Dec. 11, 1924 – July 13, 2017)

Master of Upen

He received the ACM Turing Award in 1973 for *"his outstanding contributions to database technology"*

Milestones in DBMS History (2)

- **The relational revolution (1970's)**
  - A simple data model: data is stored in relations (tables)
  - A declarative query language: SQL
  - Provided physical data independence
    - The single most important reason behind the success of DBMS today
  - A Turing Award for E. F. Codd in 1981

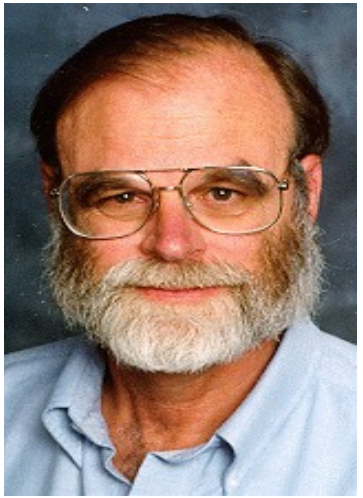
E. F. Codd
(August 23rd, 1923 - April 18th, 2003)

Bachelor of Oxford, PhD of Michigan

E. F. Codd, A Relational Model of Data for Large Shared Data Banks, Communications of ACM 13, No. 6, June 1970.

Milestones in DBMS History (3)

- **Transaction processing in 1980s**
- Jim Gray got a Turing Award for this work in 1998
- Jim Gray was lost at sea on 28 Jan, 2007
- In 2008, Microsoft announced the opening of Microsoft Jim Gray Systems Lab in Madison, Wisconsin



Jim Gray (1944-2007)
the first CS PhD of Berkeley

He received the Turing Award in 1998 "for seminal contributions to database and transaction processing research and technical leadership in system implementation."

Back in 2009, I had a guess……

Who Will Be the Next Turing Award Winner in Database Field? (1)

- Some information about the three Turing Award winners in database field
  - Graduate from famous universities
    - Charles Bachman (Master of Upen)
    - E. F. Codd (Bachelor of Oxford, PhD of Univ. of Michigan)
    - Jim Gray (the first CS PhD of Berkeley)
  - All worked for IT industries for a long time
  - Active in academy and contributed a lot to database development

Slide from my Database ppt (Lect. 1) in 2009

Back in 2009, I had a guess……

Who Will Be the Next Turing Award Winner in Database Field? (2)

- My guess: Michael Stonebraker (1943-)
  - Bachelor (Princeton); Master/PhD (U. of Michigan)
  - Prof. (Berkeley); Adjunct Prof. (MIT)
  - His career covers, and helped create, the majority of the existing relational database market today. He is also the founder of Ingres, Illustra, StreamBase Systems, Vertica, VoltDB, SciDB and was previously the CTO of Informix
  - He has received several awards, including the IEEE John von Neumann Medal and the first SIGMOD Edgar F. Codd Innovations Award. In 1994 he was inducted as a Fellow of ACM

Dept. of C.S. Fudan Univ. 46

Slide from my Database ppt (Lect. 1) in 2009

Milestones in DBMS History (4)

- **Michael Stonebraker (1943-)**
  - His career covers, and helped create, the majority of the existing relational database market today. He is also the founder of Ingres (PostgreSQL的前身), Illustra, StreamBase Systems, Vertica, VoltDB, SciDB and was previously the CTO of Informix
  - He has received several awards, including the IEEE John von Neumann Medal and the first SIGMOD Edgar F. Codd Innovations Award. He was inducted as a Fellow of ACM in 1994
  - 2014 ACM Turing Award Winner for fundamental contributions to the concepts and practices underlying modern database systems

Michael Stonebraker (1943-)

Bachelor (Princeton);
Master/PhD (U. of Michigan)
Prof. (Berkeley);
Adjunct Prof. (MIT)

A Slide in My Lecture 1 of Database in 2016

And, the next?

- The deep learning big guys
  - Geoffrey E. Hinton
  - Yoshua Bengio
  - Yann LeCun



They Got the 2018 Award!!!

Association for Computing Machinery

Advancing Computing as a Science & Profession

Digital Library | CACM | Queue | TechNews | Career Center

Join | Volunteer | myACM | Search

ABOUT ACM | MEMBERSHIP | PUBLICATIONS | SIGS | CONFERENCES | CHAPTERS | AWARDS | EDUCATION | LEARNING CENTER | PUBLIC POLICY

Home > Newsletters > ACM Bulletins > 2018 Turing Award Recipients

ACM Announces 2018 Turing Award Recipients

March 27, 2019

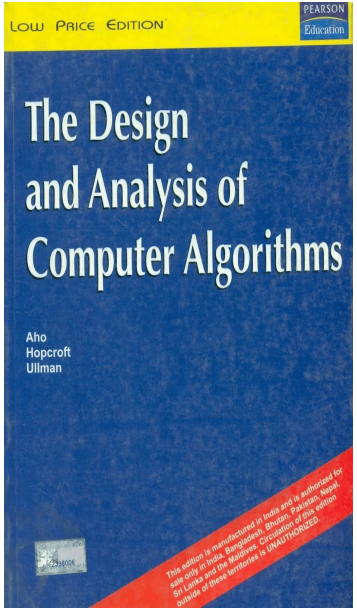
ACM has named Yoshua Bengio, Geoffrey Hinton, and Yann LeCun recipients of the 2018 ACM A.M. Turing Award for conceptual and engineering breakthroughs that have made deep neural networks a critical component of computing.

Jeffrey Ullman

AWARDS & RECOGNITION

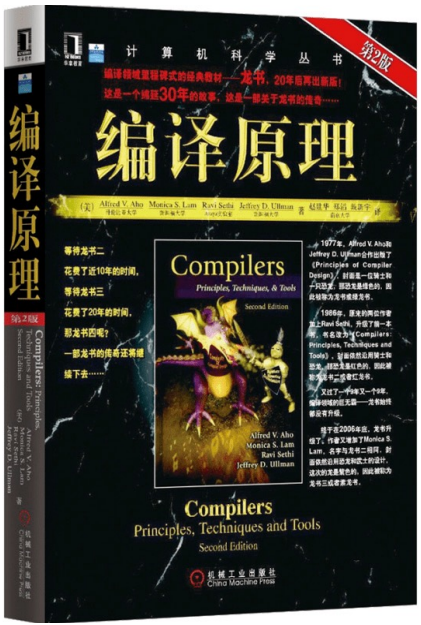
ACM Announces 2020 Turing Award Recipients

ACM has named Alfred Aho, Lawrence Gussman Professor Emeritus at Columbia University, and Jeffrey Ullman, Stanford W. Ascherman Professor Emeritus at Stanford University and CEO of Gradiance Corporation, recipients of the 2020 ACM A.M. Turing Award for fundamental algorithms and theory underlying programming language implementation, and for synthesizing these results and those of others in their highly influential books, which educated generations of computer scientists.



《计算机算法设计与分析》(1974)

《 The Design and Analysis of Computer Algorithms》



《编译程序设计原理》(1977)

《Principles of Compiler Design 》(1977)

这本书的最新版本《编译原理》（与Ravi Sethi和Monica Lam合著）于2007年出版，至今仍是有关编译器设计的标准教科书

A Slide in My Lecture 1 of Database in 2016

Then, who will be the next lucky guy?

A Slide in My Lecture 1 of Database in 2016

Then, who will be the next database lucky guy?

- It is too early to say which database guy will get the award
- But, if I'm a member of Turing Award Committee, I would like to nominate Dr. Rakesh Agrawal for his contribution to the techniques and applications of data minining

Prof. Rakesh Agrawal

- He is the recipient of the ACM-SIGKDD Inaugural Innovation Award, ACM-SIGMOD Edgar F. Codd Innovations Award, ACM-SIGMOD Test of Time Award (twice), VLDB 10-Yr Most Influential Paper Award, ICDE Most Influential Paper Award, and the Computerworld First Horizon Award
- Scientific American named him to its first list of 50 top scientists and technologists
- He is a Member of the National Academy of Engineering, a Fellow of ACM, and a Fellow of IEEE
- He has written the 1st as well as 2nd highest cited of all papers in the fields of databases and data mining (18th and 26th most cited across all computer science)

Alan Turing Award Committee

ACM A.M. Turing Award Committee

Chair	Pat Hanrahan, Stanford University , hanrahanp@acm.org *[DL Author Page]*
Member	Surajit Chaudhuri, Microsoft, surajitc@acm.org *[DL Author Page]*
	Monika Henzinger, University of Vienna, henzingerm@acm.org *[DL Author Page]*
	Norman Jouppi, Google, jouppi@acm.org *[DL Author Page]*
	Michael Kearns, University of Pennsylvania, michaelkearns@acm.org *[DL Author Page]*
	Greg Morrisett, Cornell Tech, morrisettj@acm.org *[DL Author Page]*
	Stuart Russell, UC Berkeley, stuartrussell@acm.org *[DL Author Page]*
	Margo Seltzer, University of British Columbia , seltzerm@acm.org *[DL Author Page]*
	Manuela Veloso, Carnegie Mellon University/ J.P. Morgan *[DL Author Page]*
	Avi Wigderson, Institute for Advanced Study, Princeton University, wigdersona@acm.org *[DL Author Page]*

Major DBMS Today

- **Major commercial DBMS products**
  - Oracle
  - IBM DB2 (from System R, System R*, Starburst)
  - Microsoft SQL Server
  - Sybase
  - Informix (acquired by IBM), …
- **Open source DBMS**
  - PostgreSQL (from UC Berkeley's Ingres, Postgres)
  - MySQL
- **Others**
  - NoSQL (Not only SQL, for big data): HBase, MongoDB, Neo4J, Redis, Cassandra
  - NewSQL: OceanBase, openGauss, etc.

DB-Engine Ranking

422 systems in ranking, September 2023

Rank Sep 2023	Rank Aug 2023	Rank Sep 2022	DBMS	Database Model	Score Sep 2023	Score Aug 2023	Score Sep 2022
1.	1.	1.	Oracle	Relational, Multi-model	1240.88	-1.22	+2.62
2.	2.	2.	MySQL	Relational, Multi-model	1111.49	-18.97	-100.98
3.	3.	3.	Microsoft SQL Server	Relational, Multi-model	902.22	-18.60	-24.08
4.	4.	4.	PostgreSQL	Relational, Multi-model	620.75	+0.37	+0.29
5.	5.	5.	MongoDB	Document, Multi-model	439.42	+4.93	-50.21
6.	6.	6.	Redis	Key-value, Multi-model	163.68	+0.72	-17.79
7.	7.	7.	Elasticsearch	Search engine, Multi-model	138.98	-0.94	-12.46
8.	8.	8.	IBM Db2	Relational, Multi-model	136.72	-2.52	-14.67
9.	↑ 10.	↑ 10.	SQLite	Relational	129.20	-0.72	-9.62
10.	↓ 9.	↓ 9.	Microsoft Access	Relational	128.56	-1.78	-11.47
11.	11.	↑ 13.	Snowflake	Relational	120.89	+0.27	+17.39
12.	12.	↓ 11.	Cassandra	Wide column, Multi-model	110.06	+2.67	-9.06
13.	13.	↓ 12.	MariaDB	Relational, Multi-model	100.45	+1.80	-9.70
14.	14.	14.	Splunk	Search engine	91.40	+2.42	-2.65
15.	↑ 16.	↑ 16.	Microsoft Azure SQL Database	Relational, Multi-model	82.73	+3.22	-1.69
16.	↓ 15.	↓ 15.	Amazon DynamoDB	Multi-model	80.91	-2.64	-6.51
17.	↑ 18.	↑ 20.	Databricks	Multi-model	75.18	+3.84	+19.56
18.	↓ 17.	↓ 17.	Hive	Relational	71.83	-1.52	-6.60
19.	19.	↓ 18.	Teradata	Relational, Multi-model	60.33	-0.98	-6.25
20.	20.	↑ 24.	Google BigQuery	Relational	56.46	+2.56	+6.34

Oracle：Multi-model

Relational DBMS,
Document store,
Graph DBMS,
RDF store,
Spatial DBMS

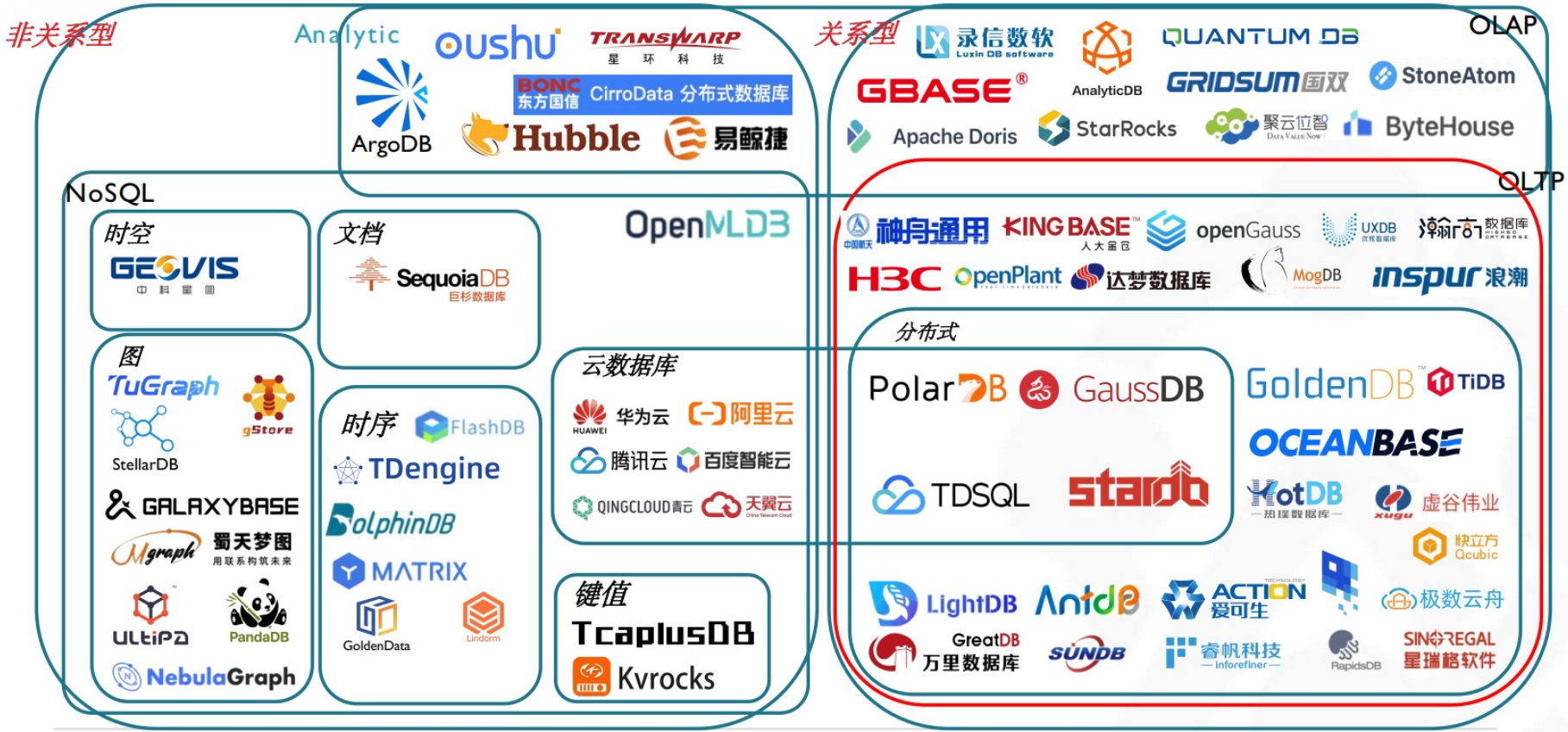
Redis：Multi-model

Key-value store,
Document store,
Graph DBMS,
Spatial DBMS,
Search engine,
Time Series DBMS,
Vector DBMS

https://db-engines.com/en/ranking

产业图谱 - 中国数据库产品图谱家族

墨天轮



蚂蚁金服OceanBase简介

- OceanBase由蚂蚁金服完全自主研发的**金融级分布式关系数据库**，创于2010年，具有数据强一致、高可用、高性能、在线扩展、高度兼容SQL标准和主流关系数据库、低成本特点。
- OceanBase已经在建设银行、南京银行、西安银行、人保健康险、常熟农商行、苏州银行、广东农信、网商银行等多家商业银行和保险机构上线。
- 2020年5月，OceanBase以**7.07亿（707,351,007）tpmC**（即每分钟内系统处理的新订单个数，transactions per minute, TPC-C）的在线事务处理性能，打破了OceanBase自己在2019年10月创造的**6088万（60,880,800）tpmC**的TPC-C世界纪录(Oracle 为30,249,688)。
- 2020年6月8日，蚂蚁集团宣布，将自研数据库产品OceanBase独立进行公司化运作，成立由蚂蚁100%控股的数据库公司**北京奥星贝斯科技**，并由蚂蚁集团CEO胡晓明亲自担任董事长。
- OceanBase是一个**支持海量数据的高性能分布式数据库系统**，实现了数千亿条记录、数百TB数据上跨行跨表事务，由淘宝核心系统研发、运维、DBA、广告、应用研发等部门共同完成。在设计和实现OceanBase的时候暂时摒弃不紧急的DBMS的功能，例如临时表、视图view等，研发团队把有限的资源集中到关键点上，**当前 OceanBase主要解决数据更新一致性、高性能的跨表读事务、范围查询、join、数据全量及增量dump、批量数据导入等**

华为openGauss简介

- GaussDB 数据库是2019年 5 月 15 日华为在北京面向全球发布的，被称为**全球首个人工智能原生（AI-Native）数据库**。
- 根据华为的介绍，GaussDB 具有两大革命性突破：
  - **首次将 AI 技术融入分布式数据库的全生命周期**，实现自运维、自管理、自调优、故障自诊断和自愈。在交易、分析和混合负载场景下，基于最优化理论，首创**基于深度强化学习的自调优算法**，调优性能比业界提升 85%
  - 通过**异构计算创新框架**充分发挥多种算力优势，在权威标准测试集 TPC-DS 上，性能比业界提升 48%
- 2019年9月19日在华为全联接大会上，华为宣布将**开源**其数据库产品，开源后命名为**openGauss**，2020 年 6 月30日openGauss数据库源代码正式开放。
- openGauss是一款**开源关系型数据库管理系统**，采用木兰宽松许可证v2发行。openGauss**内核源自PostgreSQL**，深度融合华为在数据库领域多年的经验，结合企业级场景需求，持续构建竞争力特性。同时openGauss也是一个开源、免费的数据库平台，鼓励社区贡献、合作。

武汉达梦上市——中国数据库第一股

- 2024年6月12日，**武汉达梦数据库股份有限公司**在上交所科创板挂牌上市
  - 股票名称：达梦数据
  - 股票代码：688692

- **达梦数据**成立于2000年，是一家专注于数据库及相关配套产品研发及销售的基础软件企业。公司创始人、董事长：**冯裕才**

- 冯裕才获中国计算机学会2024年“***CCF最高科学技术奖***”（终身成就奖）

Outline

- Data, information, knowledge and beyond
- Database system
- Data view
- DB languages
- DB design
- DB engine
- Database architecture
- DB user and administrator
- History of DB
- ☞ **Future directions**

A Great Tradition

- Over the last thirty years, small groups of database researchers, practitioners and opinionated professionals have periodically gathered to assess the state of the field and propose directions for future research
  - 1988, 1990, 1995, 1998, 2003, 2008, 2013, 2018, 2023

- A final report will come out after each meeting, which aims to serve various functions: to foster debate within the database research community, to explain research directions to external organizations, and to help focus community efforts on timely challenges

The Database Meetings

- 1988 The Laguna Beach Participants
- 1990 The Palo Alto Report (NSF)
- 1995 Database research: achievements and opportunities into 21st century (NSF)
- 1998 The Asilomar Report on Database Research
- 2003 The Lowell Database Research Self-Assessment
- 2008 The Claremont Report on Database Research
- 2013 The Beckman Report on Database Research
- 2018 The Seattle Report on Database Research
- 2023 The Boston Report on Database Research

2008 Claremont Meeting

- Eric A. Brewer, Michael Stonebraker, Joseph M. Hellerstein, Michael J. Franklin (Berkeley)
- Rakesh Agrawal (Yahoo!)
- Philip A. Bernstein, Surajit Chaudhuri (Microsoft)
- Michael J. Carey (UC Irvine); AnHai Doan (UWM)
- Hector Garcia-Molina (Stanford)
- Johannes Gehrke (Cornell), Le Gruenwald (OU)
- Laura M. Haas (IBM)
- Raghu Ramakrishnan (Google); Alon Y. Halevy (Washington U./Google!)
- Samuel Madden (MIT); Hank F. Korth (Lehigh U.); Alexander S. Szalay (HJU)
- Roger Magoulas, Tim O'Reilly (O'Reilly Media)
- Donald Kossmann (ETH); Anastasia Ailamaki (EPFL)
- Gerhard Weikum (MPI for CS); Yannis E. Ioannidis (UOA); Daniela Florescu (INRIA)
- Beng Chin Ooi (NUS), Sunita Sarawagi (IIT Bombay)

2008 Claremont Meeting: Challenges

- **Big Data**
  - the number of communities working with large volumes of data has grown considerably, to include not only traditional enterprise applications and Web search, but also "e-science" efforts (in astronomy, biology, earth science, etc.), digital entertainment, natural language processing, social network analysis, and more
- **Data analysis as a profit center (Data center / Cloud Computing)**
- **Ubiquity of structured and unstructured data**
- **Expanded developer demands**
  - Programmer adoption of relational DBMSs and query languages has grown significantly in recent years
- **Architectural shifts in computing**
  - At the macro scale, the rise of "cloud" computing services suggests fundamental changes in software architecture
  - At a micro scale, computer architectures have shifted the focus of Moore's Law from increasing clock speed per chip to increasing the number of processor cores and threads per chip
  - In storage technologies, major changes are underway in the memory hierarchy, due to the availability of more and larger on-chip caches, large inexpensive RAM, and flash memory
  - Power consumption has become an increasingly important aspect of the price/performance metric of large systems

2013 Beckman Meeting

2013 Beckman Meeting: Challenges

- Scalable big/fast data infrastructures
  - How dealing data volume ?
- Coping with diversity in the data management landscape
  - How dealing data variety ?
- End-to-end processing and understanding of data
  - How dealing data velocity?
- Cloud services
  - How deploying Big Data applications in the cloud ?
- The roles of people in the data life cycle
  - How managing the involvement of people in these applications ?

2018 Seattle Meeting: Participants

- Daniel Abadi, Anastasia Ailamaki, David Andersen
- Peter Bailis, Magdalena Balazinska, Phil Bernstein
- Peter Boncz, Surajit Chaudhuri, Alvin Cheung
- Anhai Doan, Luna Dong, Mike Franklin, Juliana Freire
- Alon Halevy, Joe Hellerstein, Stratos Idreos
- Donald Kossman, Tim Kraska, Sailesh Krishnamurthy,
- Volker Markl, Sergey Melnik, Tova Milo, C. Mohan,
- Thomas Neumann, Beng Chin Ooi, Fatma Ozcan
- Jignesh Patel, Andy Pavlo, Raluca Popa
- Raghu Ramakrishnan, Christopher Re
- Mike Stonebraker, and Dan Suciu

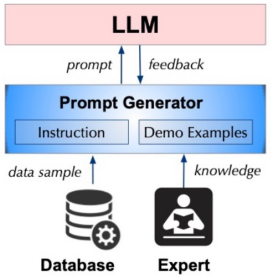
2018 Seattle: Challenges

- Data science
- Data governance
- Cloud services
- Database engines
  - Data lakes and modern data warehousing applications:
  - Leveraging machine learning
- Community

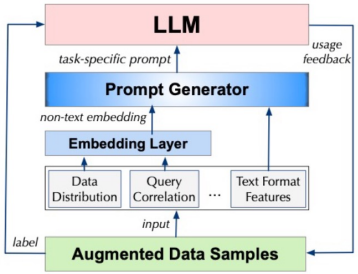
2023 Boston Meeting

2023 Boston Meeting: Challenges

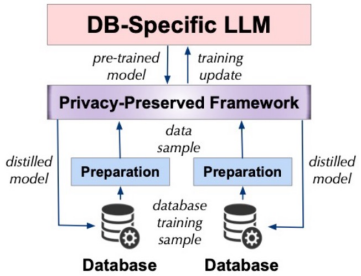
- Cloud-native
  - Data Federation
  - Data Marketing, Data Sharing
  - Data ecosystem
  - Data compiler
- AI/LLM
  - Control, Compute, Storage
  - Retrieval Augmented Generation
  - LLM Agent
  - NL2SQL
  - LLM as Chips/OS
- Future of Database Engines
- Applications, industry, and DB community



(a) Input Prompt Generation.



(b) LLM Fine-Tuning.



(c) DB-Specific LLM Pre-Training.

Topics of DB Meetings

	1998	1990	1995	1998	2003	2008	2013	2018	2023
More data types: Image, spatial, time, genetics	√	√	√	√		√			
Information retrieval	√			√	√				
Data mining		√	√	√	√				
Extendible DBMSs, object-oriented DBMSs	√		√	√					
Exploit hardware advances	√			√	√			√	
Parallelism, scale-up, scale-out	√	√			√				
High availability, replication	√		√						
Workflow models, long transactions, workflow engines	√	√	√	√					
Heterogeneous DBMSs, semantic consistency, data fusion	√	√	√	√	√				
Uncertain and probabilistic data, data quality		√	√	√	√				
Privacy and Trustworthy	√				√				
One-Hundred-Year Storage	√				√				
Real-time DBs, streams, sensor networks, sensor	√				√				
Schema-less DBs, unify unstructured and structured data	√				√	√			
User interfaces for DBs	√		√		√	√			
Web, Multimedia, Mobile App			√	√	√	√			

Topics of DB Meetings

	1998	1990	1995	1998	2003	2008	2013	2018	2023
Mobile Applications and Virtual Worlds					√	√			
Federate Database Systems, Declarative Programming						√			
Human-in-the-loop data management							√		
Data science pipeline, Data system landscape							√		
Big data						√	√		
DB engines: AQP								√	
DB engines: ML								√	√
DB engines: benchmark								√	√
DB engines: heterogeneous computation								√	√
DB engines: data lakes								√	
Cloud DB: disaggregation, multi-tenancy							√	√	√
Cloud DB: hybrid cloud							√	√	
Cloud DB: edge and cloud							√	√	
Data Science: data integration and wrangling								√	√
Data Science: data provenance, governance, sharing, ethical								√	√
Community: ecosystem, impact, education							√	√	√

Summary

- Database, database system, DBMS and its benefits
- Drawbacks of file system to store data
- Levels of abstraction: physical/logical/view levels
- Physical/logical independence
- Entity-Relational(ER) model/Relational model
- DDL/DML (SQL)
- Database design: logical/physical design
- Database engine: storage/query/transaction manager
- Components of storage management
- Components of query processor
- Transaction manager: Recovery/concurrency-control manager
- Architecture of database system
- Users of database
- Duties of database administrator (DBA)
- Differences among database, data mining and information retrieval

Review Terms

- Database management system (DBMS)
- Data system applications
- File processing systems
- Data inconsistency
- Consistency constraints
- Data abstraction
- Instance
- Schema
  - Physical schema
  - Logical schema
- Physical data independence
- Data models
  - Entity-relationship model
  - Relational data model
  - Object-based data model
  - Semi-structured data model
- Database languages
  - Data definition language (DDL)
  - Data manipulation language (DML)
  - Query language
- Metadata
- Application program
- Normalization
- Data dictionary
- Storage manager
- Query processor
- Transactions
  - Atomicity
  - Failure recovery
  - Concurrency control
- Two- and three-tier database architectures
- Data mining
- Database administrator (DBA)

Some useful terms

- **Byte**
- **Kilobyte** – 2^{10} bytes (i.e. 1024 bytes)
- **Megabyte** – 2^{20} bytes (i.e. 1024 kilobytes)
- **Gigabyte** – 2^{30} bytes (i.e. 1024 megabytes)
- **Terabyte** – 2^{40} bytes (i.e. 1024 gigabytes)
- **Petabyte** – 2^{50} bytes (i.e. 1024 terabytes)
- **Exabyte** – 2^{60} bytes (i.e. 1024 petabytes)
- **Zettabyte** – 2^{70} bytes (i.e. 1024 exabytes)
- **Yottabyte** – 2^{80} bytes (i.e. 1024 zettabytes)

提醒：以下内容介绍我实验室，和课程无关，同学如果没兴趣，可以不看。

先进数据与机器智能系统实验室

Advanced Data and Machine Intelligence Systems (ADMIS) Lab

（周水庚实验室）

周水庚

上海市智能信息处理重点实验室

复旦大学计算机学院

Email: sgzhou@fudan.edu.cn

URL: http://admis.fudan.edu.cn/sgzhou

主页

http://admis.fudan.edu.cn/sgzhou

Shuigeng Zhou(周水庚)

Shanghai Key Lab of Intelligent Information Processing
School of Computer Science
Fudan University

- Home
- Funding
- Publications
- Books/Proceedings
- Software/Databases
- Supervising
- Teaching
- Patents
- Prizes
- Services
- Others

Shanghai Key Lab of Intelligent Information Processing
School of Computer Science
No. 2 Interdisciplinary Building, Fudan University
2005 Songhu Road, Shanghai 200438, China
Phone/fax: +86-21-31242359
Email: sgzhou AT fudan DOT edu DOT cn
My Google Scholar Profile
My DBLP Records
My Lab Website

Shuigeng Zhou is a full professor (since 2003) at the **School of Computer Science**, **Fudan University**. He leads the research group of **Advanced Data and Machine Intelligence Systems (ADMIS)**. He was the director (Sept. 2018 – May 2024) and vice-director (2004 to Aug. 2018) of **Shanghai Key Lab of Intelligent Information Processing** (SKLIIP), Fudan University. He held the post of Vice Chairman (in charge of research affairs) of the Department of Computer Science and Engineering (CSE), Fudan University, from January 2005 to May 2008. In May 2008, the CSE Department was merged into the School of Computer Science, Fudan University.

Shuigeng Zhou received his Bachelor Degree of Engineering from **Huazhong University of Science and Technology (HUST)** in July 1988, his Master Degree of Engineering from **University of Electronic Science and Technology of China (UESTC)** in March 1991, and his PhD of Computer Science from **Fudan University** in July 2000. He served in **Shanghai Academy of Spaceflight Technology** from April 1991 to August 1997, as an engineer and a senior engineer (since August 1995) respectively. Before joining Fudan University, he was a post-doctoral researcher in **Wuhan University** from August 2000 to August 2002.

Google Scholar Profile

Shuigeng Zhou

FOLLOW

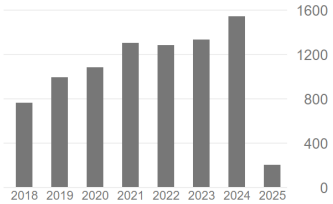
GET MY OWN PROFILE

Fudan University
Verified email at fudan.edu.cn - Homepage
Database Bioinformatics Machine Learning

TITLE	CITED BY	YEAR
Focusing attention: Towards accurate text recognition in natural images Z Cheng, F Bai, Y Xu, G Zheng, S Pu, S Zhou Proceedings of the IEEE international conference on computer vision, 5076-5084	611	2017
Aon: Towards arbitrarily-oriented text recognition Z Cheng, Y Xu, F Bai, Y Niu, S Pu, S Zhou Proceedings of the IEEE conference on computer vision and pattern …	363	2018
Improving compound–protein interaction prediction by building up highly credible negative samples H Liu, J Sun, J Guan, J Zheng, S Zhou Bioinformatics 31 (12), i221-i229	308	2015
Cassava genome from a wild ancestor to cultivated varieties W Wang, B Feng, J Xiao, Z Xia, X Zhou, P Li, W Zhang, Y Wang, BL Møller, ... Nature communications 5 (1), 5110	286	2014
Single-cell trajectories reconstruction, exploration and mapping of omics data with STREAM H Chen, L Albergante, JY Hsu, CA Lareau, G Lo Bosco, J Guan, S Zhou, ...	258	2019

Cited by VIEW ALL

	All	Since 2020
Citations	13892	6760
h-index	61	37
i10-index	271	167

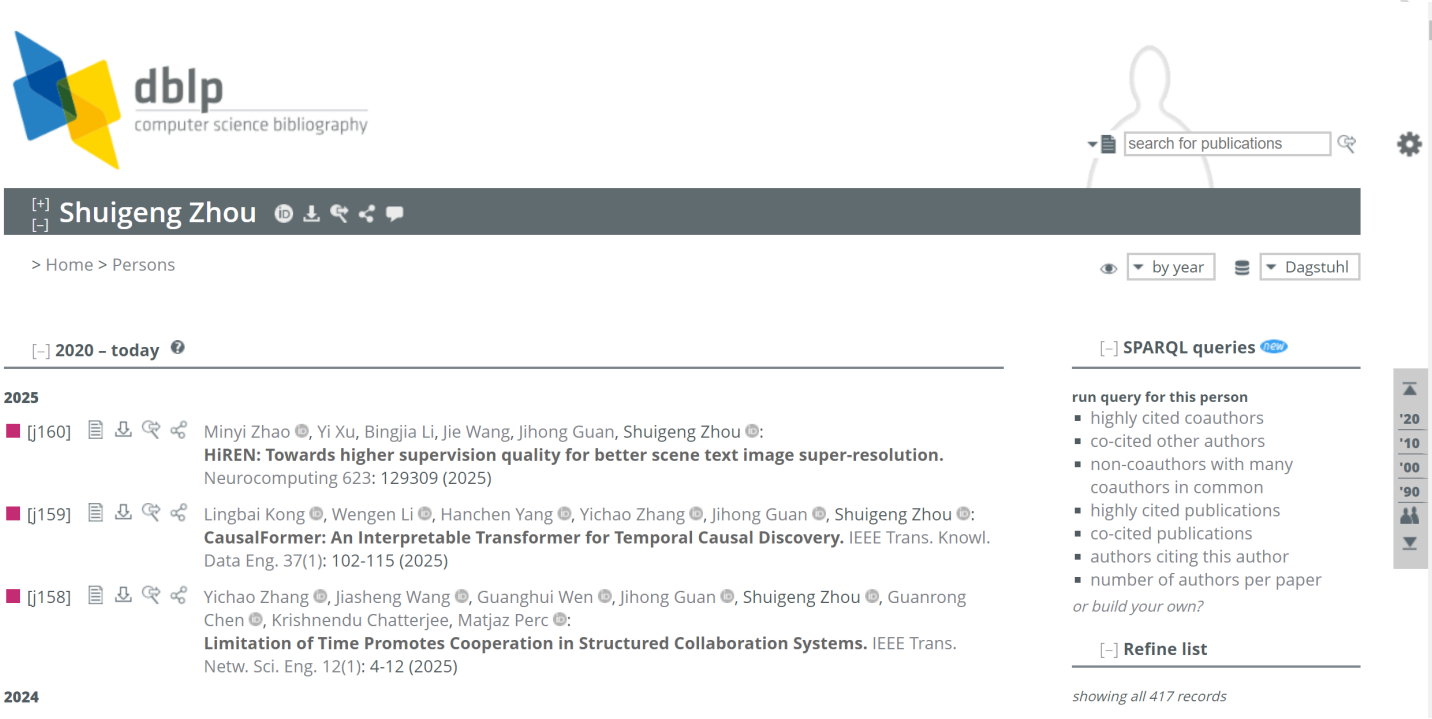


Public access VIEW ALL

30 articles 135 articles

最新数据：论文引用13892篇次， h-index= 61

DBLP数据库记录



最新数据：总共收录论文417篇

先进数据与机器智能系统实验室

Advanced Data and Machine Intelligence Systems (ADMIS) Lab

- 人员组成（41）
  - 1位教授
  - 2位博士后、13博士生和25硕士生
- 主要研究领域
  - 人工智能、大数据管理与分析、网络与博弈、与生物和金融等的交叉研究
- 学术成果
  - 论文：国际学术论文400多篇，其中：国际期刊论文200多篇，国际顶级会议/期刊论文100多篇；
  - 专利：申请/授权发明专利62项目，授权23项
  - 获奖：获得省部级（教育部）科研奖励二等奖7项

研究资助

- 国家与省部及地方政府资助
  - 国家自然科学基金重点/面上项目9项
  - 国家重点研发计划（课题、子课题）
  - 工信部
  - 上海市科委/经信委/教委

- 企业资助与合作
  - 国企
    - 中国航天科工、航天科技、中船、中石化、中电、海康威视、上海宝信等
  - 民企
    - 华为、阿里、腾讯、字节跳动、百度、优图、哔哩哔哩等

研究领域：人工智能

- 深度学习及其应用
  - 零（小）样本学习/目标检测、图像文本识别、视频分析、异常检测、推荐系统、声纹识别等
  - NeurIPS’21/22 , ICCV’17/19/23, CVPR’18/19/21, AAAI’21/20/19 , IJCAI’22, ACM-MM’19/21/22/23, SIGIR’23
- 人工智能安全与隐私保护
  - 隐私保护的人脸识别和虹膜识别、联邦学习
  - ICLR’25, AAAI’25, CVPR’24, ICCV’23, ACM MM’22
- 机器学习
  - 因果推断（IEEE TPAMI 2024，TCYB 2020/2022, ACM TKDD, TIST 2019, KDD’25, NeurIPS’24, AAAI’17/18/19/21/22/23/24）
  - 稀疏表示（AAAI’14/15/16, ECAI’16）
- 数据挖掘
  - 图数据分类（KDD’16）
  - 流数据挖掘（ICDE’08/09）
  - 金融数据挖掘（IJCAI’13/15, IEEE TKDE 2016, ACM TIST 2018）
  - 异常检测（ACM TKDD, IEEE TAI, KDD’06, AAAI’19 ）

研究领域：大数据管理与分析

- 大规模图数据查询处理
  - SIGMOD’21/20/19/18/13/11, VLDB’17, ICDE’07/12/15/16, VLDBJ 2014/2020
- 隐私保护
  - ICDE’14, SIGMOD’13, EDBT’12, IEEE TKDE 2009
- 路网查询处理
  - SIGMOD’13/15, VLDB’12, IEEE TKDE 2017, IEEE TSC 2018
- 分布式查询与索引
  - IEEE TKDE 2010/2009, IEEE TPDS 2011/2008
- 基于外存储的算法
  - Algorithmica 2018, ICDT’15, VLDBJ 2014, SPAA’14, SODA’13

研究领域： 生物信息学与计算生物学

- 基因组
  - Nat. Comm. 2014/2019, Bioinformatics 2015, ACM/IEEE TCBB 2015 etc.
- 宏基因组（Metagenomics）
  - ACM/IEEE TCBB 2014 etc.
- 表观遗传（Epigenetics）
  - NAR 2016, BIB 2014 etc.
- 蛋白质组学（蛋白质结构与功能预测）
  - NAR 2014, BIB 2014, RECOMB'13, Bioinformatics 2009/2023 etc.
- 转录组学（miRNA聚类、分类及其靶基因预测）
  - ACM/IEEE TCBB 2015, Bioinformatics 2009 etc.
- 单细胞测序数据聚类分析
  - ACM/IEEE TCBB 2022, BIB 2022, JBCB 2020 etc.
- 基于人工智能的药物设计与发现
  - ACM/IEEE TCBB 2016, ISMB'15, Bioinformatics 2015/2021/2022 etc.

研究领域：金融信息处理

- 投资组合优化（Portfolio）
  - IJCAI 2013/2015, IEEE TKDE 2016, ACM TIST 2018
- 市场预测
  - DASFAA 2015
- 最优交易
  - DASFAA 2020
- 金融欺诈检测
  - Neurocomputing, 2012

近年学生获得多个AI大赛奖

- 2024年CVPR人脸合成挑战赛冠军
- 2023年海军“金海豚”算法挑战赛两项冠军
- 2021 CVPR压缩视频质量增强挑战赛 冠军
- 2018年全球AI挑战赛零样本学习（ZSL）冠军
- 2018年未来高校AI挑战赛声纹认证组冠军
- 2017年ICDAR CoCo-Text第一名
- 2019年“默克杯”逆合成反应预测大赛三等奖
- 近年来，实验室研究生在人工智能相关的重要国际学术会议（包括ICLR, NeurIPS, AAAI, IJCAI, ICCV, CVPR, SIGKDD, SIGIR, ACM-MM等）和期刊（包括IEEE TPAMI/TKDE/TCYB, ACM TKDD等）发表/录用论文40多篇

欢迎同学们加盟并申报以下计划

- 暮政学者
- 曦源学者
- 望道学者
- ……

End of Lecture 1