



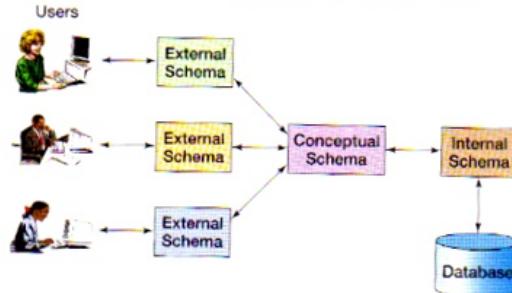
Week1_Course_part1

Course - Introduction

Database System - Nankai



Database System



Yuan Xiaojie
袁晓洁



Song Chunyao
宋春瑶



Wen Yanlong
温延龙



Nie Peng
乜鹏

College of Computer Science, Nankai University

Slides adapted from material by Profs. Jeff Ullman (Stanford) and Art Keller (UCSC)

Database System - Nankai



Why should you study databases?



Computer

High speed processing

Massive storage

Data



The world is increasingly driven by data...

Science: how to handle the data - data poor to data rich

Fundamental ideas to/from all of CS: systems, theory, AI, logic, stats, analysis...

Database System - Nankai



Course Goals & Content

- This class teaches **the basics** of how to use & manage data.



- cover mostly relational databases
 - how to **design** and create such databases
 - how to **use** them (via SQL query language)
 - how to **implement** them (only briefly)

Database System - Nankai



Outline

- Introduction to DBMSs
- Principles of Data Layout and Index
- The Entity Relationship model
- E/R to Relational Transformation
- SQL: Introduction
- SQL: Advanced
- DB Schema Design
- Relation Algebra & Query Processing
- Crash Recovery
- Concurrency Control

} Introduction to
database system



} Use of database
system



} Implement of
database system

学堂云

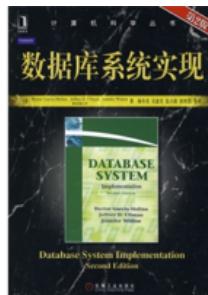
Database System - Nankai



Textbook



- “A First Course in Database System” , Jeffrey D.Ullman, Jennifer Widom
 - 翻译版，《数据库系统基础教程》原书第三版，机械工业出版社（本科教材）



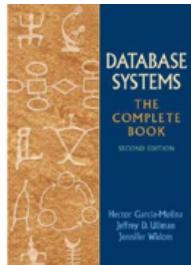
- “Database System Implementation” ,Hector Garcia-Molina, Jeffrey D.Ullman, ...
 - 翻译版，《数据库系统实现》第二版，机械工业出版社（部分本科内容、研究生教材）

Database System - Nankai



Textbook

Database Systems: The Complete Book, Hector Garcia-Molina, Jeffrey D. Ullman, and Jennifer Widom, Prentice Hall, 2008.



原版很贵



已出版

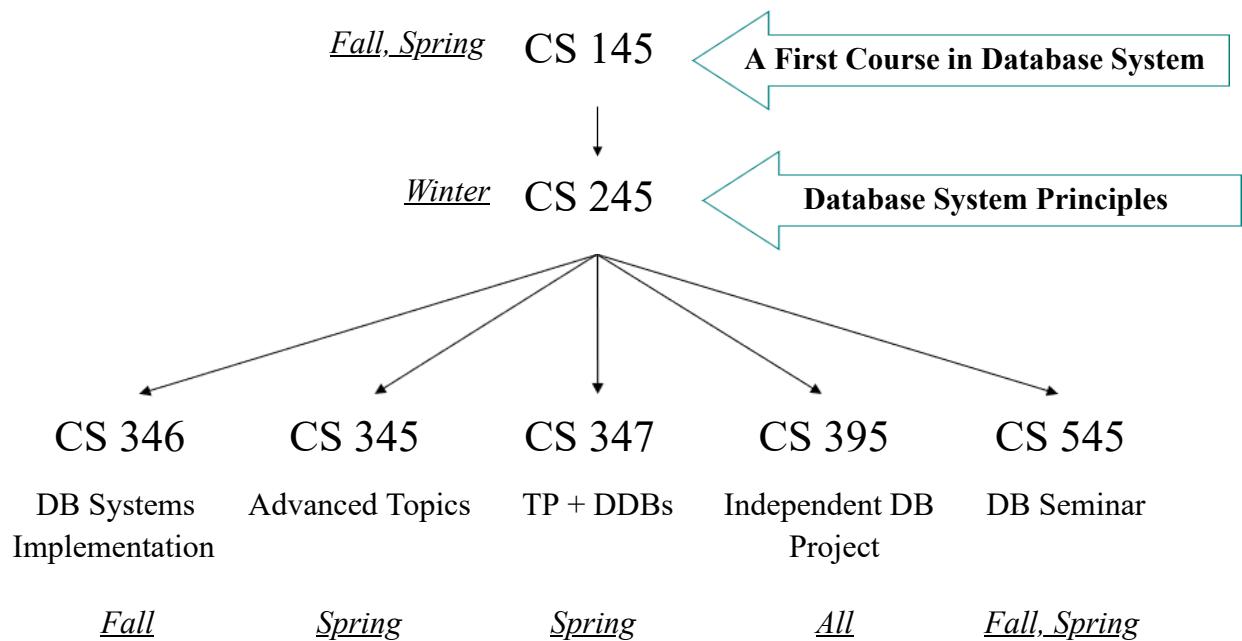
这本书的网址是

<http://infolab.stanford.edu/~ullman/dscb.html>

Database System - Nankai



Stanford Database Courses



Database System - Nankai



斯坦福课程自学网站 Jennifer Widom



- <https://lagunita.stanford.edu/courses/DB/2014/SelfPaced/about>

MINI-COURSES

ABOUT THIS SET OF COURSES

"Databases" was one of Stanford's three inaugural massive open online courses in the fall of 2011; it was offered again in MOOC format in 2013 and 2014. The course is now being offered as a set of smaller self-paced "mini-courses", which can be assembled in a variety of ways to learn about different aspects of databases. All of the mini-courses are based around video lectures and/or video demos. Many of them include in-video quizzes to check understanding, in-depth standalone quizzes, and/or a variety of automatically-checked interactive programming exercises. Each mini-course also includes a discussion

Course Number: Databases
Estimated Effort: 8-12 hours per week
Price: Free

OUR RESEARCH COMMUNITY



Database System - Nankai



Course Format

- For all students
 - Class participation (offline and online) 10%
 - some homeworks 10%
 - Programming projects 20%
 - Midterm Examination 10%, (SQL, Computer)
 - Final Examination {Closed Book} 50%

Database System - Nankai



互动交流

- 请同学们在学堂云上回答如下问题：
 - 在修读此课之前，你是否有数据库方面的相关基础；
 - 在C++课上你是否做过文件存储方面的编程；
 - 目前（还没有学习数据库课程）你对数据库的认识；
 - 本学期需要基于MySQL开发一个信息管理系统，你知道有哪些前端编程语言？
 - SimpleDB的编程环境是Java语言，你可以自学Java吗？（卓越班）
 - 其他关于此课程的其他问题

Database System - Nankai



Week1_Course_part2

Database System-Introduction

Database System - Nankai



Today's Lecture

- Introduction
- – Motivating example
- How do DBMSs work? DDL, DML
- Fundamental concepts
- DBMS users
- Overall system architecture
- Steps in Building a DB Application
- Conclusions



Database System - Nankai



Definitions



‘DBMS’ = ‘Data Base Management System’ :is a piece of software designed to store and Manage databases
the (commercial) system, like: DB2, Oracle, MySQL, MS SQL Server, ...

‘Database system’ :
DBMS + data + application programs

Database System - Nankai



A Motivating Example

- Suppose we are building a system to store the information about:
 - _ students
 - _ courses
 - _ professors
 - _ who takes what, who teaches what

为什么需要DBMS ?

Database System - Nankai



采用文件系统实现存储功能是否可行？

文件系统是否可以完成数据库的存储功能？

— 利用C++编写数据存储程序样例

```
#include <fstream.h>
struct Date{int iMonth,iDay,iYear;};
void main()
{
    Date dt={6,10,92};
    ofstream datafile("ata.dat",ios::binary);
    datafile.write((char *)&dt,sizeof dt);
}
```

Database System - Nankai



aaa - Microsoft Visual C++ - [aaa.cpp *]

File Edit View Insert Project Build Tools Window Help

(Globals) [All global members] main

aaa classes

- Date
- Globals
- main()

```
#include <fstream.h>
struct Date{int iMonth, iDay, iYear;};
void main()
{
    Date dt={6, 10, 92};
    ofstream datafile("ata.dat", ios::binary);
    datafile.write((char *)&dt, sizeof dt);
}
```

aaa.cpp
Linking...

aaa.exe - 0 error(s), 0 warning(s)

Build / Debug / Find in Files 1 / Find in Files 2 / Results / SQL Debugging /

Ready

- ata.dat: 06 00 00 00 0A 00 00 00 5C 00 00 00

Database System - Nankai



建立自己数据库所面临的问题

```
#include <fstream.h>
struct Date{int iMonth,iDay,iYear;};
void main()
{
    Date dt={6,10,92};
    ofstream datafile("ata.dat",ios::binary);
    datafile.write((char *)&dt,sizeof dt);
}
```

- ata.dat: 06 00 00 00 0A 00 00 00 5C 00 00 00

- 在源代码中描述数据的类型(元数据 metadata)，使得任何想使用数据的人必须访问源代码。
- 元数据(metadata)：关于数据描述的数据
- 为防止并发操作出现问题，处理文件封锁和解锁的源代码会相当复杂

采用数据库管理系统(DBMS)是最佳选择

Database System - Nankai



Detailed outline

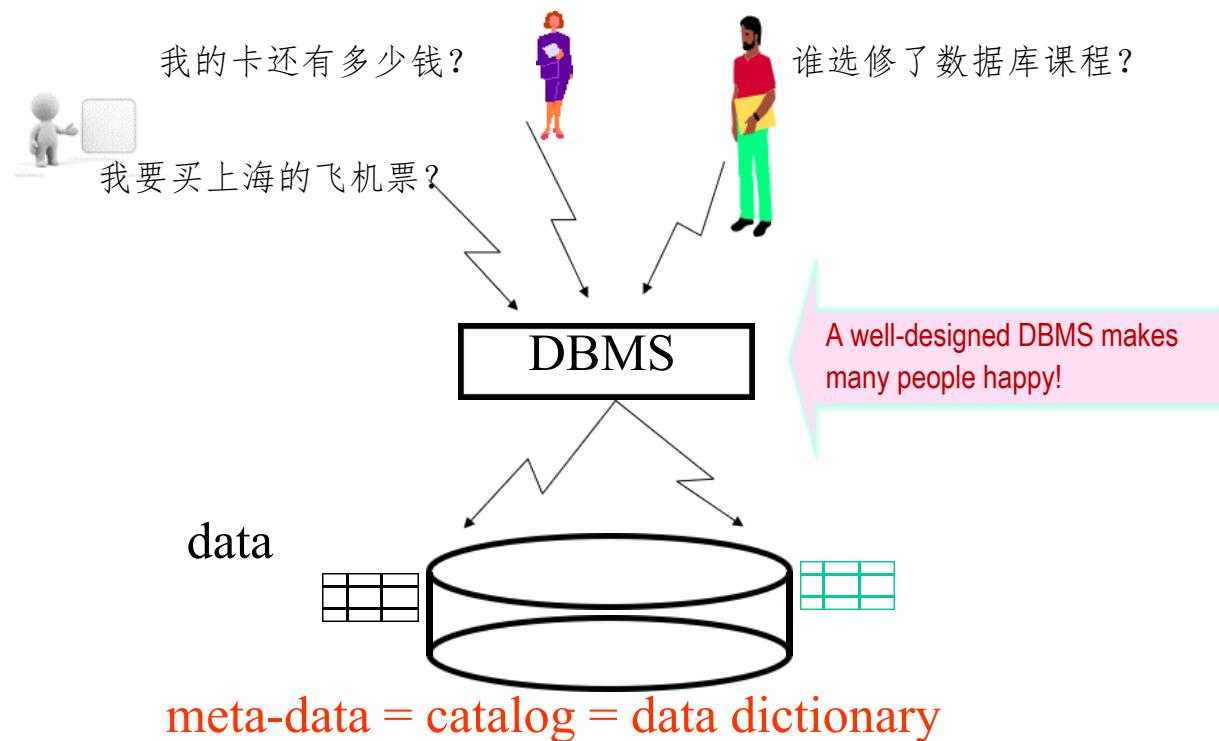
- Introduction
 - Motivating example
 - How do DBMSs work? DDL, DML, views.
 - Fundamental concepts
 - DBMS users
 - Overall system architecture
 - Steps in Building a DB Application
 - Conclusions



Database System - Nankai



How do DBs work?



Database System - Nankai



DBMS应该怎样构造？

- 银行系统

- 数据项有：姓名、地址、帐号、存款、结余等
- 基本操作：查询、更新、转帐等
- 不能出现任何错误：并发操作、支取现金等

- 订票系统

- 数据项有：航班信息、机票信息、预定信息等
- 基本操作：查询时间、票价、座位等，更新信息等
- 并发访问处理，系统崩溃处理，防止信息丢失等

数据项存储

数据项操作

数据并发控制

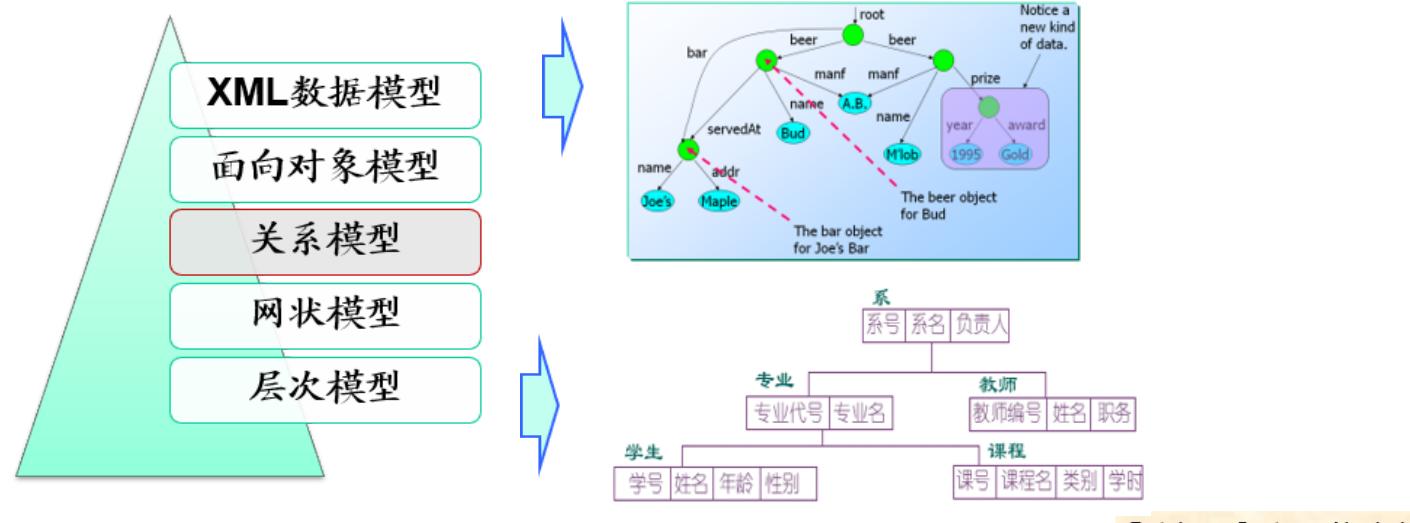
数据库恢复

Database System - Nankai



数据项存储-数据模型 (Data Model)

数据库中的数据是按照一定的逻辑结构存放的，
这种结构是用数据模型来表示的





An Example of a Relation

Table name

↓
student

Attribute names

tuples

course

takes

学号	姓名	出生日期	所属学院
20161001	张三	1994.8.5	计控
20162023	李四	1995.6.7	数学
20164512	王五	1994.11.15	金融
...

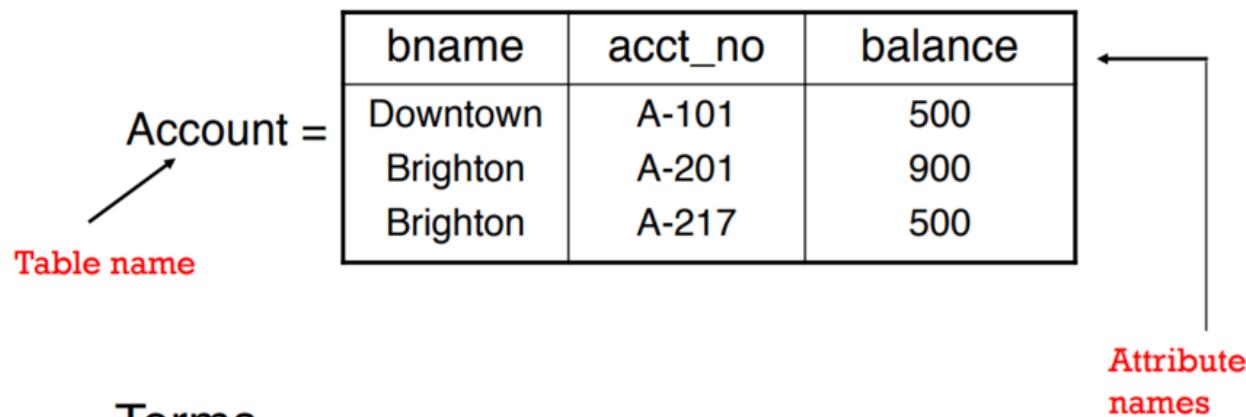
课号	课名	学分
1001	数据库	3
1002	英语	4
1006	操作系统	3
...

课号	学号	成绩
1001	20161001	95
1002	20161001	70
1001	20164512	80
...

Database System - Nankai



RELATIONAL MODEL - TERMS

Account = 

bname	acct_no	balance
Downtown	A-101	500
Brighton	A-201	900
Brighton	A-217	500

Terms

- Tables → Relations
- Columns → Attributes
- Rows → Tuples
- Schema (e.g.: Acct_Schema = (bname, acct_no, balance))

Database System - Nankai



How do RDBMS work? - SQL

SQL (Structured Query Language)

```
create table student  
(学号 char(8) primary key,  
姓名 char(8),  
出生日期 date,  
所属学院 varchar(30))
```

```
insert into student values('20161001', '张三', '1994.08.05', '计控')  
insert into student values('20162023', '李四', '1995.06.07', '数学')  
insert into student values('20164512', '王五', '1994.11.15', '金融')
```

DDL 数据定义语言
(Data Definition Language)

DML 数据操作语言
(Data Manipulation Language)

学号	姓名	出生日期	所属学院
20161001	张三	1994.8.5	计控
20162023	李四	1995.6.7	数学
20164512	王五	1994.11.15	金融
...

Database System - Nankai



SQL-支持任何复杂查询和更新操作

Example:

```
select 学号 from student  
where not exists  
(select * from course  
where not exists  
(select * from takes  
where takes.课号=course.课号  
and takes.学号=student.学号))
```

学号	姓名	出生日期	所属学院
20161001	张三	1994.8.5	计控
20162023	李四	1995.6.7	数学
20164512	王五	1994.11.15	金融
...

课号	学号	成绩
1001	20161001	95
1002	20161001	70
1001	20164512	80
...

课号	课名	学分
1001	数据库	3
1002	英语	4
1006	操作系统	3
...

查找选修了全部课程的学生学号

Database System - Nankai



数据操作语言的特点

- Imperative

for each row a in student

 for each row c in takes

 if a.学号 = c.学号 and c.课号 = 1001
 output a

NESTED
LOOPS

- Declarative

SELECT a.姓名

FROM student AS a, takes AS c

WHERE a.学号 = c.学号 AND c.课号 = 1001

JOIN

Database System - Nankai



本节需要理解的主要知识

DBMS, Database System, Data Model, DDL,
DML, Metadata, Catalog, Data Dictionary

请继续在长江雨课堂上进行交互

Database System - Nankai

多选题 1分



互动交流一

以下哪些软件是DBMS?

- A Windows
- B PostgreSQL
- C MATLAB
- D Python
- E informix
- F Visio
- G Sybase
- H Unix

提交

Database System - Nankai

单选题 1分



互动交流二

ALTER TABLE table_name

ADD column_name datatype

- A 上述SQL命令是 DDL
- B 上述SQL命令是 DML
- C 上述SQL命令是 Metadata

提交

Database System - Nankai

单选题 1分



互动交流二

UPDATE Person SET FirstName = 'Fred'

WHERE LastName = 'Wilson'

- A 上述SQL命令是 DDL
- B 上述SQL命令是 DML
- C 上述SQL命令是 Metadata

提交

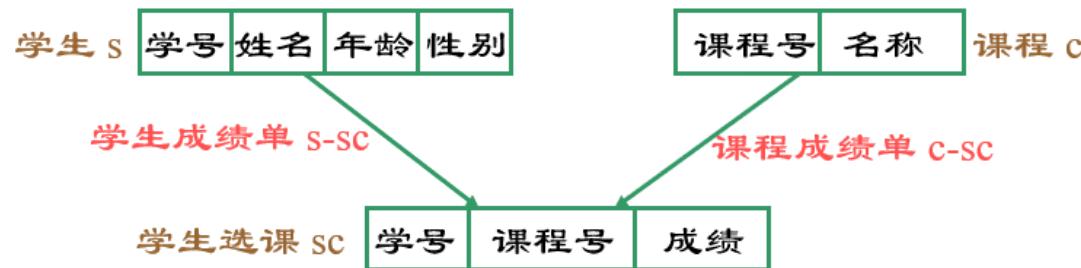
Database System - Nankai

单选题 1分



互动交流四

记录类型按以下连接方式存储的数
据库是什么模型的数据?



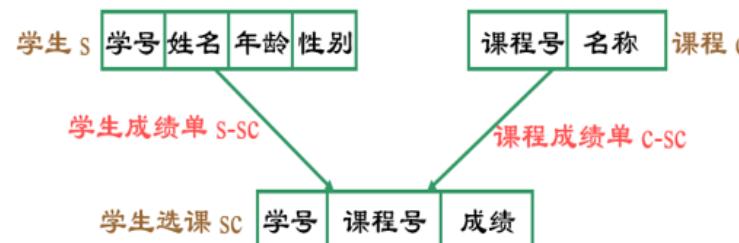
- A 关系模型
- B 层次模型
- C 网状模型
- D 没有模型

提交

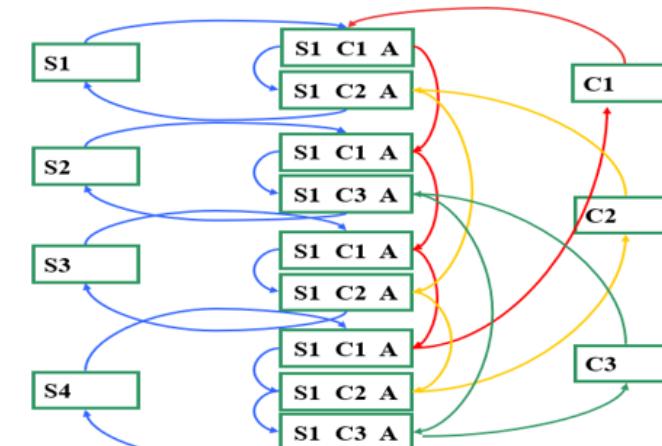
Database System - Nankai



讲解模式 (Schema) 和数据 (Data) 的关系



采用网状模型 (Network Model) 组织的学生选课数据库，上述例子是一个具体数据模式 (Data Schema)，它反映了学生、课程和学生选课三个记录类型之间的关系



这是该数据模式下具体的数据示例

Schema - Types

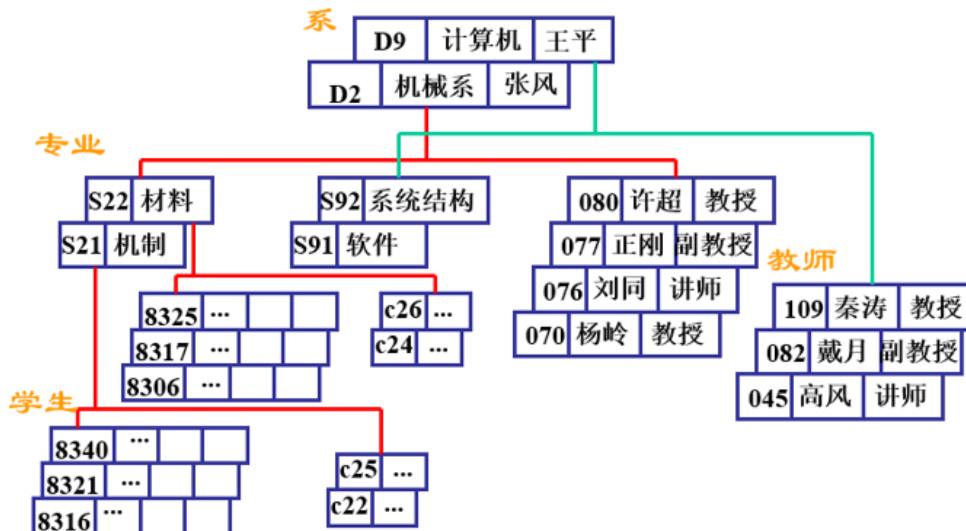
Instances - Variable

Database System - Nankai



互动交流五

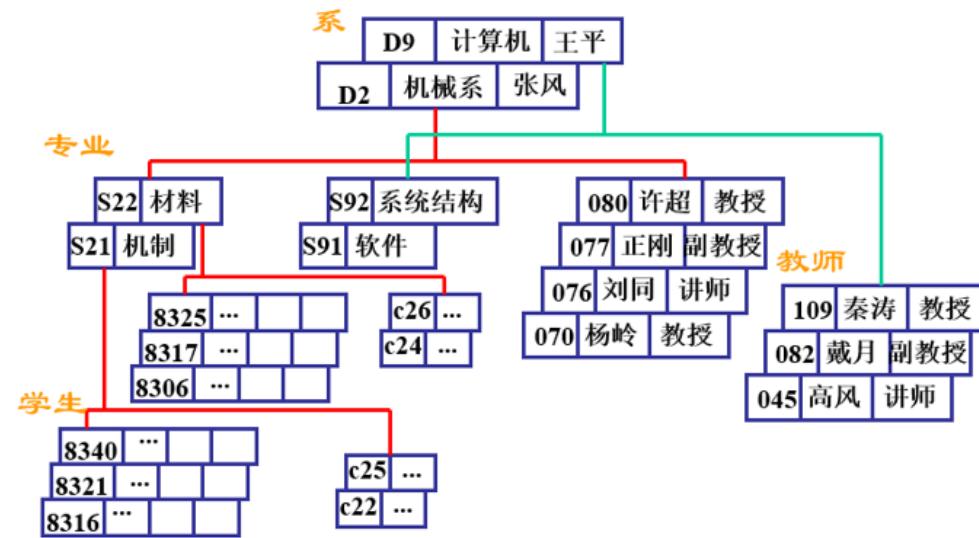
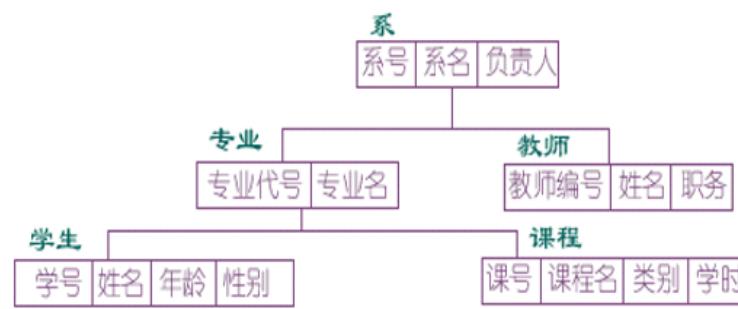
数据按以下方式进行存储的数据库是什么数据模型的数据库？



- A 网状模型
- B 层次模型
- C 关系模型
- D 没有模型

提交

Database System - Nankai



Schema - Types Instances - Variable

Database System - Nankai

主观题 2分



互动交流五

Consider a two-dimensional integer array of size $n*m$ that is to be used in your favorite programming language. Using the array as an example, illustrate the difference between a schema and instances.

提交

Database System - Nankai

主观题 2分



互动交流六

请举出你日常使用过的5个数据库应用系统的例子，你在这些数据库上经常进行哪种类型的操作？（在什么场景下进行何种操作，在数据库里相当于查询、插入、删除和更改哪种操作）

提交

Database System - Nankai



Week1_Course_part3

Database System-Introduction

Database System - Nankai



Detailed outline

- Introduction
 - Motivating example
 - How do DBMSs work? DDL, DML, views.
- — Fundamental concepts
- DBMS users
- Overall system architecture
- Steps in Building a DB Application
- Conclusions



Database System - Nankai

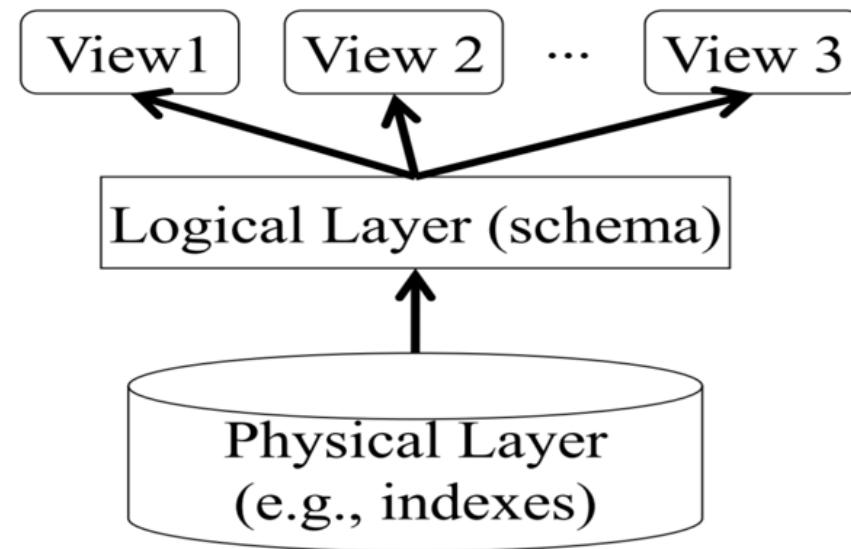


Data Independence

Data Independence

Logical Data
Independence

Physical Data
Independence



Changes at one layer do not affect another layer!

Database System - Nankai



Levels of Abstraction

- **Physical level** (物理模式、内模式、存储模式)
 - how are these tables stored, how many bytes/attribute etc, indices, hash, ...
- **Logical level** (模式、逻辑模式)
 - describes data stored in database, and the relationships among the data.
 - STUDENT(学号, 姓名, 出生日期, 所属学院)
 - TAKES (学号, 课号, 成绩)
- **View level** (外模式、子模式、用户模式)
 - application programs hide details of data types. Views can also hide information (e.g., salary) for security purposes.

Database System - Nankai



view level example

- view level, eg:

```
create view cstudent(ssn,sname)
as select 学号,姓名
from student;
```

视图的优点：

- 限制表中特定列和行的使用
- 隐藏复杂查询的细节
- 提供安全保证

The screenshot shows a database interface with a SQL query window and a results window. The SQL query is:

```
select *
from cstudent
```

The results window displays the following data:

	ssn	sname
1	20161001	张三
2	20162023	李四
3	20164512	王五

Database System - Nankai



Data independence

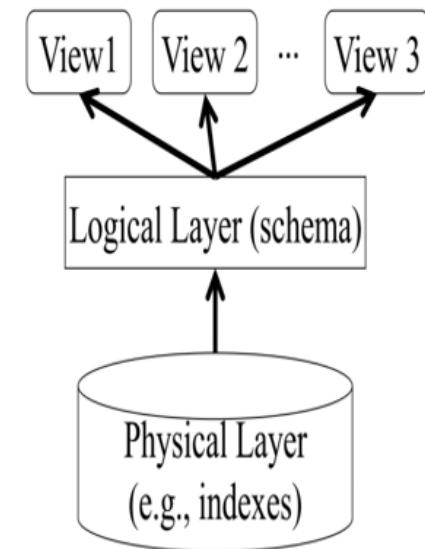
Concept: Applications do not need to worry about *how the data is structured and stored*

- Logical data independence (逻辑独立性)
 - protection from changes in the *logical structure of the data*
 - can add (drop) column; add/drop table
- Physical data independence (物理独立性)
 - protection from *physical layout changes*
 - can add index; change record order

Data Independence

Logical Data
Independence

Physical Data
Independence



Changes at one layer do not affect another layer!

Database System - Nankai



Detailed outline

- Introduction
 - Motivating example
 - How do DBMSs work? DDL, DML, views.
 - Fundamental concepts
 - DBMS users
 - Overall system architecture
 - Steps in Building a DB Application
 - Conclusions

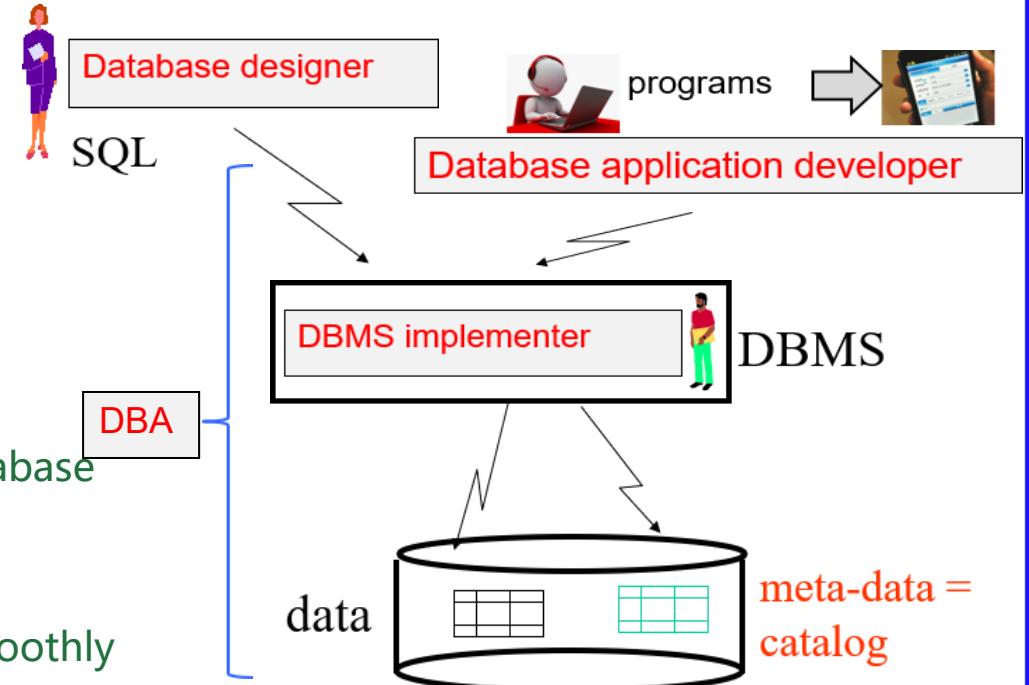


Database System - Nankai



Database Key People

- DBMS implementer
 - _ Builds system
- Database designer
 - _ Establishes schema
- Database application developer
 - _ Programs that operate an database
- Database administrator(DBA)
 - _ Loads data, Keeps running smoothly



Database System - Nankai



DB Administrator (DBA)

- schema definition ('logical' level)
- physical schema (storage structure, access methods)
- schema modifications
- granting authorizations
- integrity constraint specification



Database System - Nankai



Detailed outline

- Introduction
 - Motivating example
 - How do DBMSs work? DDL, DML, views.
 - Fundamental concepts
 - DBMS users
 - Overall system architecture
 - Steps in Building a DB Application
 - Conclusions

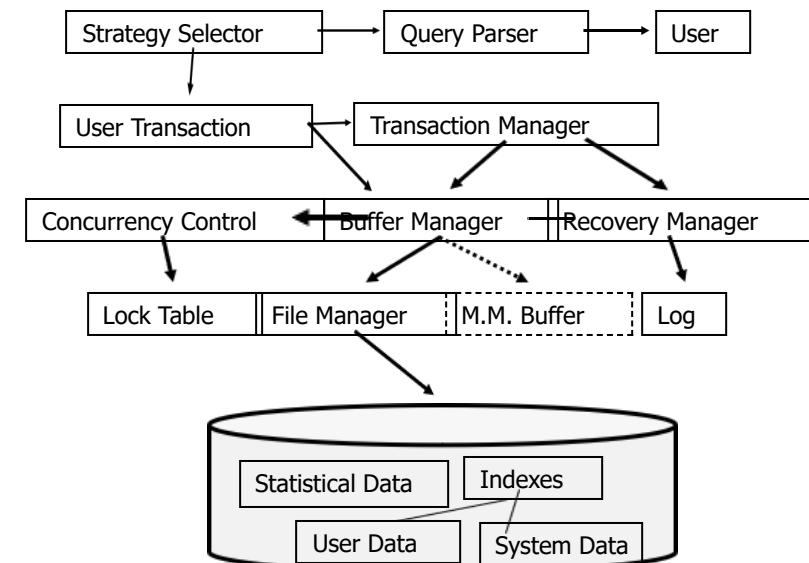


Database System - Nankai



Overall system architecture

- File & System Structure
 - Records in blocks, dictionary, buffer management,...
- Indexing & Hashing
 - B-Trees, hashing,...
- Query Processing
 - Query costs, join strategies,...
- Crash Recovery
 - Failures, stable storage,...
- Concurrency Control
 - Correctness, locks,...
- Transaction Processing
 - Logs, deadlocks,...



Database System - Nankai



Detailed outline

- Introduction
 - Motivating example
 - How do DBMSs work? DDL, DML, views.
 - Fundamental concepts
 - DBMS users
 - Overall system architecture
 - Steps in Building a DB Application
 - Conclusions

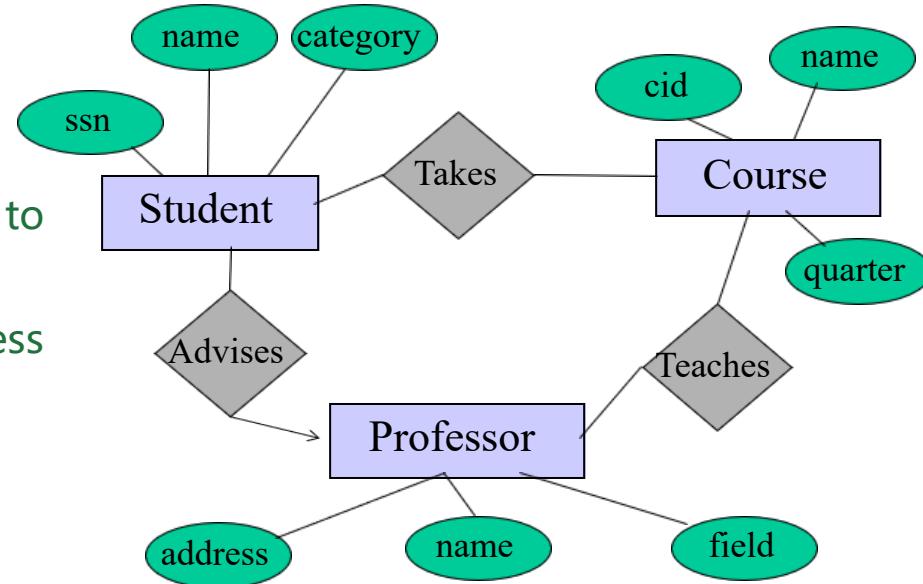


Database System - Nankai



Steps in Building a DB Application

- Step 0: pick an application domain
 - we will talk about this later
- Step 1: conceptual design
 - discuss with your team mates what to model in the application domain
 - need a modeling language to express what you want
 - ER model is the most popular such language
 - output: an ER diagram of the app. domain

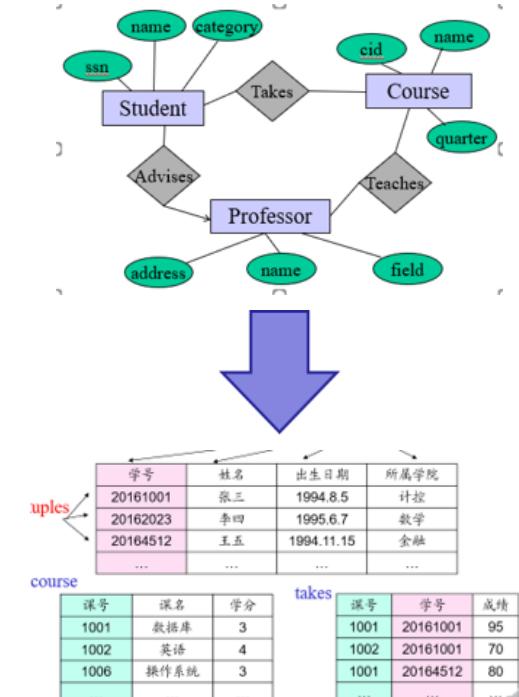


Database System - Nankai



Steps in Building a DB Application

- Step 2: pick a type of DBMS
 - relational DBMS is most popular and is our focus
- Step 3: translate ER design to relational schema
 - use a set of rules to translate from ER to rel. schema
 - use a set of schema refinement rules to transform the above rel. schema into a **good** rel. schema
- Step 4: Schema Design and Implementation
 - you have a good relational schema on paper



Database System - Nankai



Steps in Building a DB Application

- Step 5: Subsequent steps include
 - implement your relational DBMS using a "database programming language" called **SQL**
 - ordinary users cannot interact with the database directly
 - and the database also cannot do everything you want
 - hence **write your application program** in C++, Java, ASP, etc to handle the interaction and take care of things that the database cannot do



Database System - Nankai



Main vendors/products

Commercial

- Oracle
- IBM/DB2
- MS SQL-Server
- Sybase
- (MS Access,
- ...)

Open source

- Postgres (UCB)
mySQL/mariaDB
sqlite (sqlite.org)

(www.acm.org/sigmod)

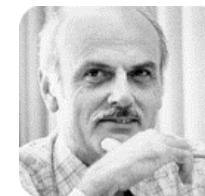


Database System - Nankai



数据库领域图灵奖简介

- 1973年,Charles W.Bachman,网状数据库之父
 - 主持设计与开发了最早的网状数据库系统IDS
 - 推动和促成了数据库标准的制定
- 1981年,Edgar Frank Codd,关系数据库之父
 - A relational Model of Data for Large Shared Data Banks
- 1998年,James Gray, 数据库技术和事务处理专家
 - 完整性、安全性、并发性、故障恢复



Database System - Nankai



数据库领域图灵奖简介

This year: Turing Award for Innovations in RDBMSs

“The Nobel of Computing”

- 2014 A.M. Turing Award Winner: Michael Stonebraker
- Helped to invent many RDBMS (Relational DBMS) concepts:
 - Query modification
 - The Object-Relational model
 - More recently: work on column-store, streaming data
- Made / helped to start many popular RDBMS implementations:
 - Postgres, Vertica, Streambase, VoltDB, ...



The relational data model is one of the most important concepts in computing!

8

Database System - Nankai



Conclusions

The world is increasingly driven by data...

This class teaches the basics of how to use
& manage data.

Microsoft

ORACLE



Google

Bai^必度^度

阿里巴巴^{Alibaba.com}



DBMS, Data Model, Database system, DDL, DML,
Data Schema, Instances, Metadata, Catalog.....

Database System - Nankai

多选题 1分



互动交流一

数据库系统的三级模式结构由（ ）模式和内模式组成？

- A 存储模式
- B 外模式
- C 逻辑模式
- D 用户模式
- E 物理模式
- F 子模式

提交

Database System - Nankai

多选题 1分



互动交流二

以下哪个描述是数据库模式Schema的含义?

- A 装配数据的框架
- B 物理存储策略
- C 局部逻辑结构

提交

Database System - Nankai

多选题 1分



互动交流三

在三级模式之间引入两级映像，其主要功能之一是

- A 保持数据与程序的一致性
- B 使系统具有较高的通道能力
- C 使数据与程序具有较高的独立性

提交

Database System - Nankai

单选题 1分



互动交流四

数据库的最小存取单位是

- A 数据项
- B 记录
- C 数据块

提交

Database System - Nankai

单选题 1分



互动交流五

当数据的总体逻辑结构改变时，通过映像的相应改变，可保持局部逻辑结构不变，这就是数据与程序的（）

- A 一致性
- B 逻辑独立性
- C 完整性
- D 物理独立性

提交

Database System - Nankai

单选题 1分



互动交流六

对于数据库系统，负责定义数据库内容，决定存储结构和存储策略及安全授权等工作的是（ ）

- A 应用程序员
- B 数据库管理员
- C 数据库管理系统的软件设计员

提交

Database System - Nankai

主观题 2分



互动交流七

Consider a two-dimensional integer array of size $n*m$ that is to be used in your favorite programming language. Using the array as an example, illustrate the difference between the three levels of data abstraction.

提交

Database System - Nankai