第17讲 从【表】的管理看数据库
--结构化与非结构化数据管理

战德臣

哈尔滨工业大学计算学部教学委员会主任
国家教学名师

为什么需要数据库--工作方式的转变

信息社会的工作方式

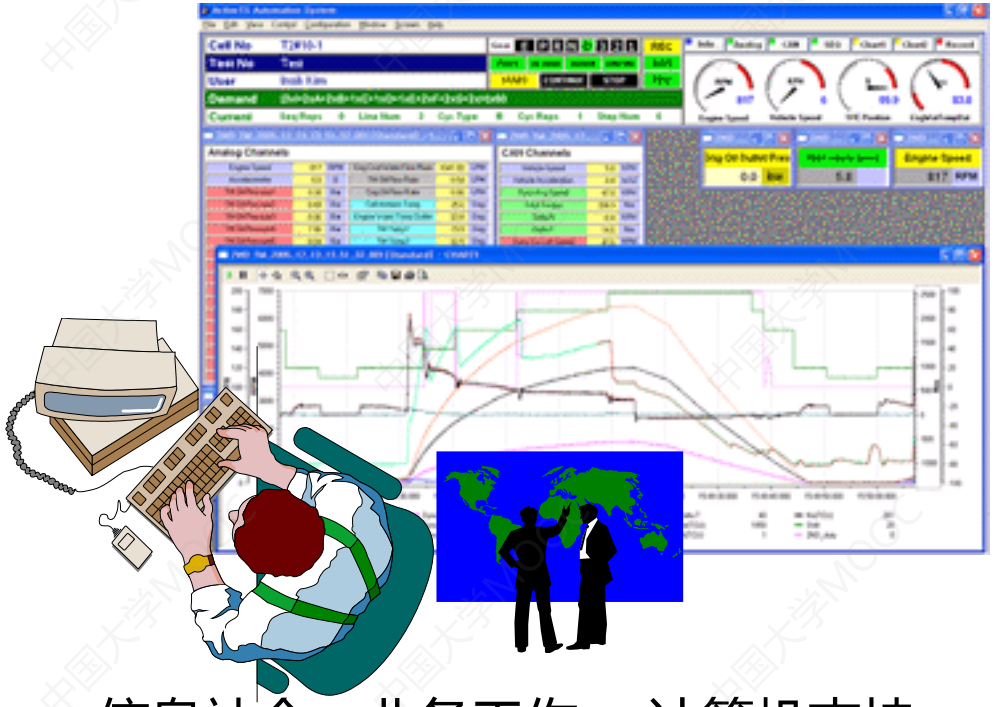
- 网络/Internet
- 数据库

什么是数据库?

Everything Over DB
Everything With Internet

不要相信经验，一切以数据说话

传统社会：业务工作

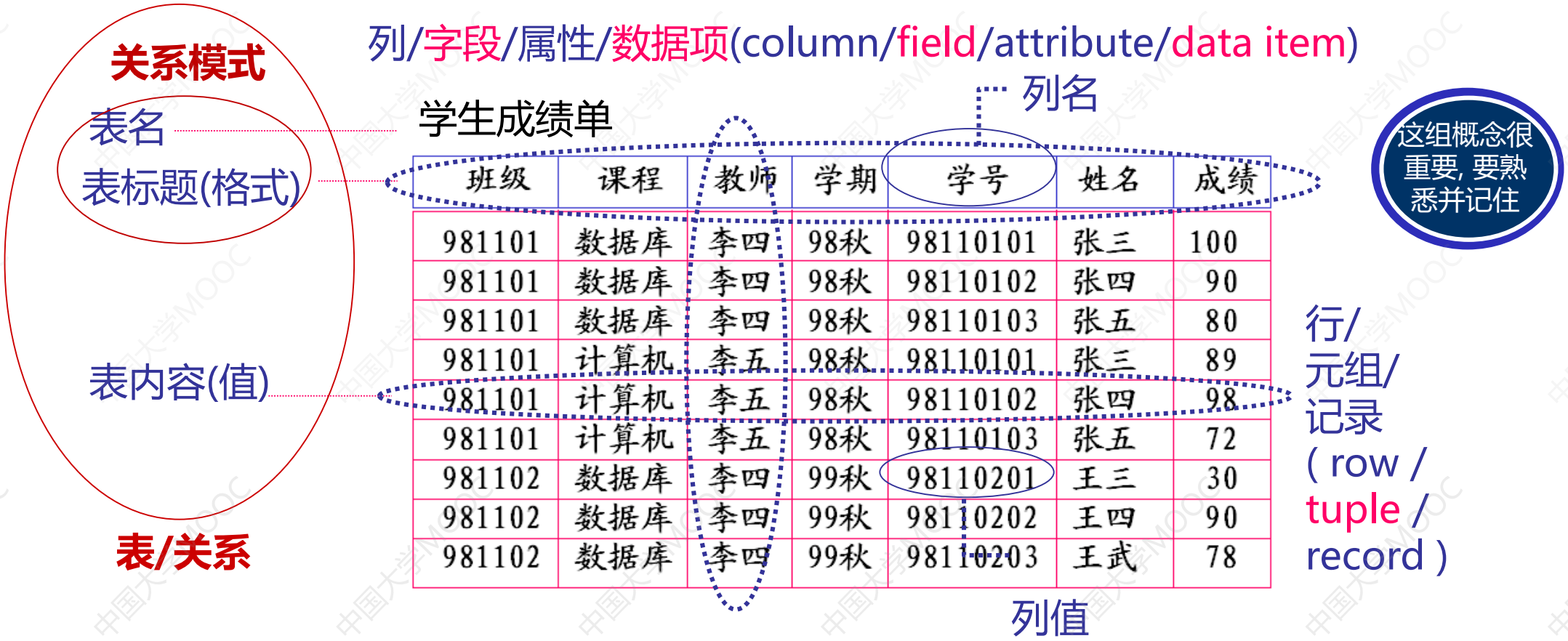


信息社会：业务工作 + 计算机支持

什么是数据库

基本形式的数据：“表”和“关系”

数据库起源于规范化“表(Table)”的处理
Table: 以按行按列形式组织及展现的数据。



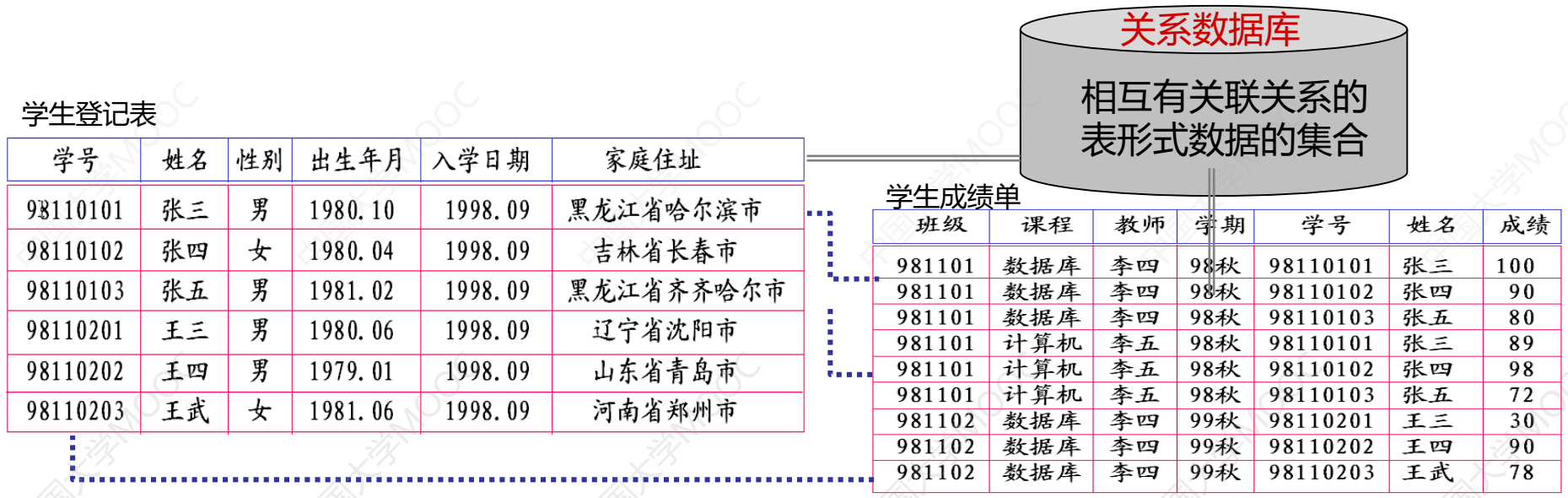
班级	课程	教师	学期	学号	姓名	成绩
981101	数据库	李四	98秋	98110101	张三	100
981101	数据库	李四	98秋	98110102	张四	90
981101	数据库	李四	98秋	98110103	张五	80
981101	计算机	李五	98秋	98110101	张三	89
981101	计算机	李五	98秋	98110102	张四	98
981101	计算机	李五	98秋	98110103	张五	72
981102	数据库	李四	99秋	98110201	王三	30
981102	数据库	李四	99秋	98110202	王四	90
981102	数据库	李四	99秋	98110203	王武	78

什么是数据库

结构化数据库：关系数据库

数据库：相互有关联关系的数据的集合

- 一个表聚集了具有相同结构类型的若干个对象，表与表间也存在着相互关联
- 一行数据反映了某一对象的相关内容，一列数据具有相同的数据类型
- 这种“表”形式的数据，被认为是结构化数据
- 用数学定义的无重复元组的【表】又被称为【关系】，其数据库又被称为关系数据库



学生登记表

学号	姓名	性别	出生年月	入学日期	家庭住址
98110101	张三	男	1980.10	1998.09	黑龙江省哈尔滨市
98110102	张四	女	1980.04	1998.09	吉林省长春市
98110103	张五	男	1981.02	1998.09	黑龙江省齐齐哈尔市
98110201	王三	男	1980.06	1998.09	辽宁省沈阳市
98110202	王四	男	1979.01	1998.09	山东省青岛市
98110203	王武	女	1981.06	1998.09	河南省郑州市

学生成绩单

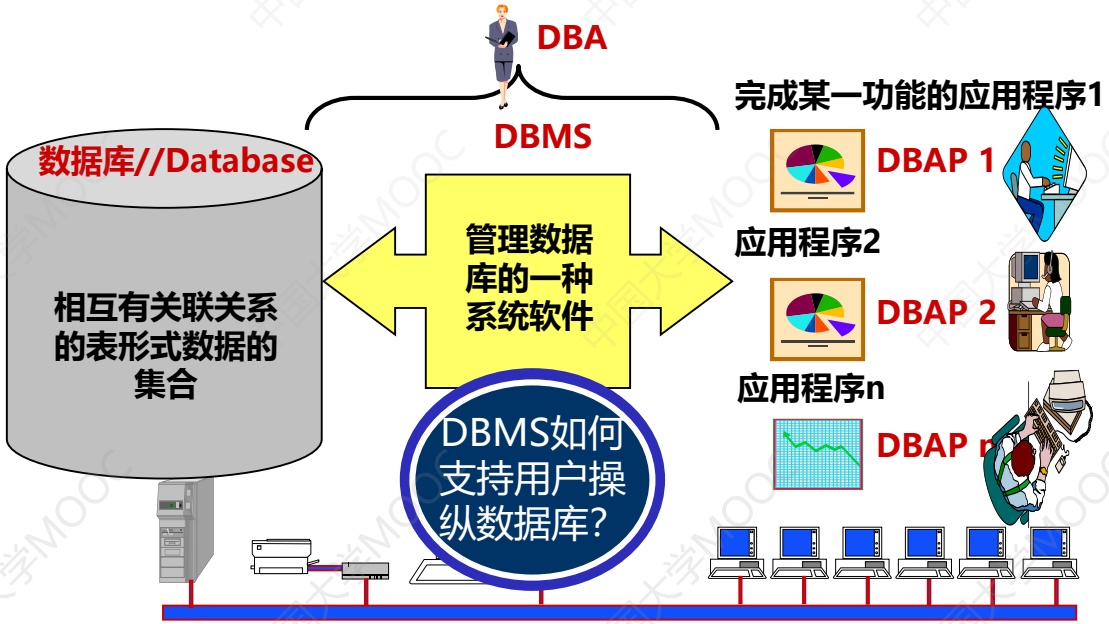
班级	课程	教师	学期	学号	姓名	成绩
981101	数据库	李四	98秋	98110101	张三	100
981101	数据库	李四	98秋	98110102	张四	90
981101	数据库	李四	98秋	98110103	张五	80
981101	计算机	李五	98秋	98110101	张三	89
981101	计算机	李五	98秋	98110102	张四	98
981101	计算机	李五	98秋	98110103	张五	72
981102	数据库	李四	99秋	98110201	王三	30
981102	数据库	李四	99秋	98110202	王四	90
981102	数据库	李四	99秋	98110203	王武	78

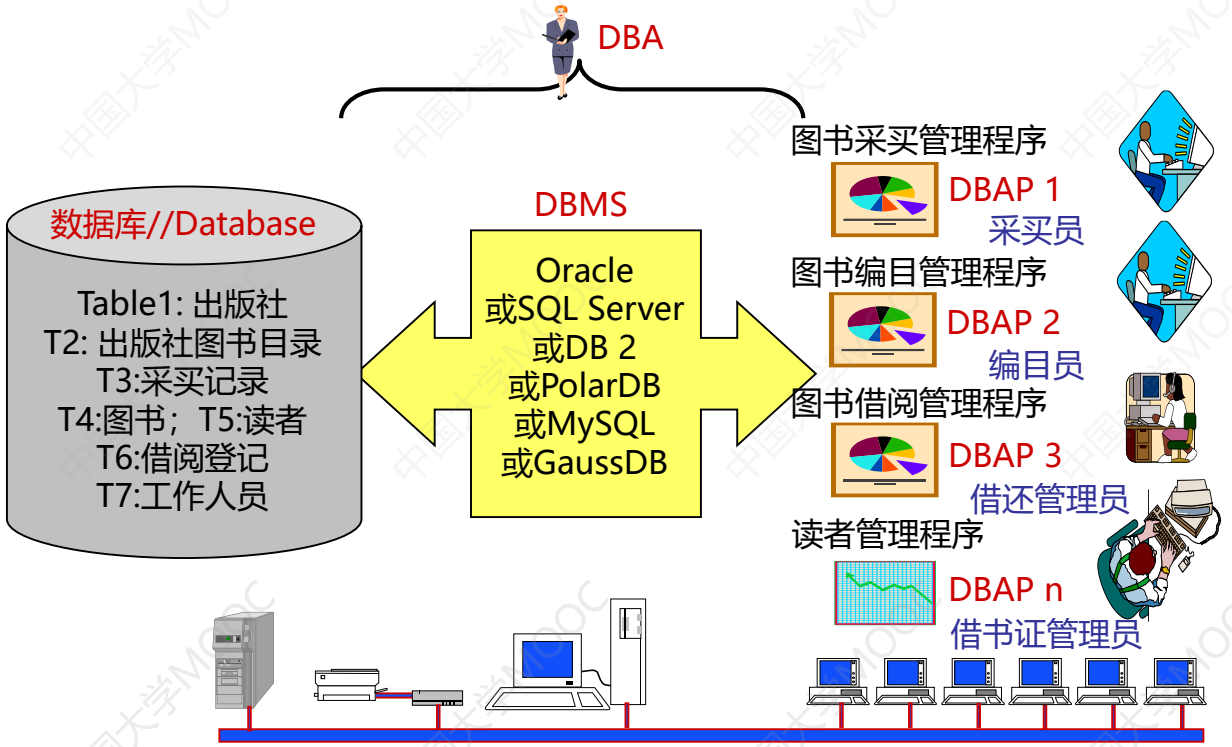
什么是数据库系统

数据库系统的构成

数据库系统(工作环境)

- ■数据库(DB): Database
- ■数据库管理系统(DBMS): Database Management System
- ■数据库应用(DBAP): DataBase Application
- ■数据库管理员(DBA): DataBase Administrator
- ■计算机基本系统





怎样操纵数据库

表之间有哪些运算

学生登记表

学号	姓名	性别	出生年月	入学日期	家庭住址
98110101	张三	男	1980.10	1998.09	黑龙江省哈尔滨市
98110102	张四	女	1980.04	1998.09	吉林省长春市
98110103	张五	男	1981.02	1998.09	黑龙江省齐齐哈尔市
98110201	王三	男	1980.06	1998.09	
98110202	王四	男	1979.01	1998.09	
98110203	王武	女	1981.06	1998.09	

学生成绩单

班级	课程	教师	学期	学号	姓名	成绩
981101	数据库	李四	98秋	98110101	张三	100
981101	数据库	李四	98秋	98110102	张四	90
981101	数据库	李四	98秋	98110103	张五	80
981101	计算机	李五	98秋	98110101	张三	89
981101	计算机	李五	98秋	98110102	张四	98
981101	计算机	李五	98秋	98110103	张五	72
981102	数据库	李四	99秋	98110201	王三	30
981102	数据库	李四	99秋	98110202	王四	90
981102	数据库	李四	99秋	98110203	王武	78

有哪些运算?

并： $R \cup S$
差： $R - S$
积： $R \times S$
选择： $\sigma(R)$
投影： $\pi(R)$

怎样操纵数据库

第1种运算：【并】运算

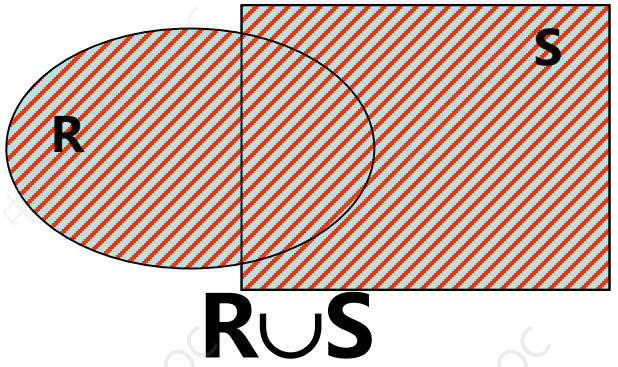
并(Union)

- **定义**：设关系R和关系S是并相容的(即：属性数目相同，其对应的域也相同)，则关系R与关系S的并运算结果也是一个关系，记作：R∪S, 它由或者出现在关系R中，或者出现在S中的元组构成
- **数学描述**：　$R \cup S = \{ t \mid t \in R \vee t \in S \}$，其中t是元组
- **特性**：R ∪S 与 S ∪R 运算的结果是同一个关系

R		
A1	A2	A3
a	b	c
a	d	g
f	b	e

S		
B1	B2	B3
a	b	c
a	b	e
a	d	g
h	d	g

R∪S		
C1	C2	C3
a	b	c
a	d	g
f	b	e
a	b	e
h	d	g

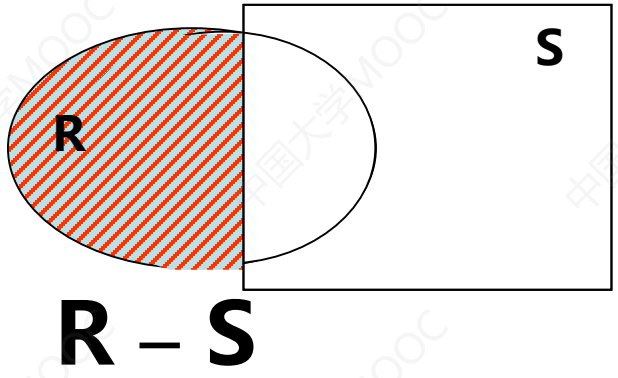


怎样操纵数据库

第2种运算：【差】运算

差(Difference)

- **定义**：假设关系R 和关系S是并相容的，则关系R 与关系S 的差运算结果也是一个关系，记作：R – S, 它由出现在关系R中但不出现在关系S中的元组构成
- **数学描述**： $R - S = \{ t \mid t \in R \wedge t \notin S \}$，其中t是元组
- **注意**：R – S 与 S – R 是不同的



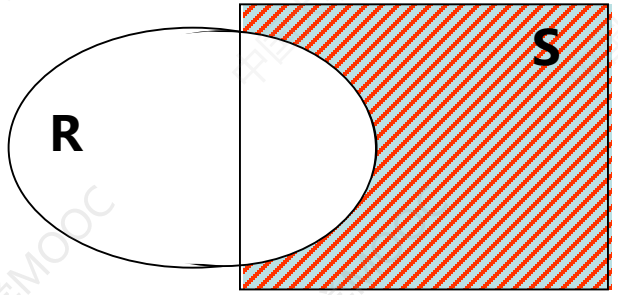
R – S

R		
A1	A2	A3
a	b	c
a	d	g
f	b	e

S		
B1	B2	B3
a	b	c
a	b	e
a	d	g
h	d	g

R – S		
D1	D2	D3
f	b	e

S – R		
E1	E2	E3
a	b	e
h	d	g



S – R

怎样操纵数据库

第3种运算：【积】运算

广义笛卡尔积 (Cartesian Product)

□ **定义**：关系R ($<a_1, a_2, \ldots, a_n>$) 与关系S($<b_1, b_2, \ldots, b_m>$) 的广义笛卡尔积 (简称广义积)

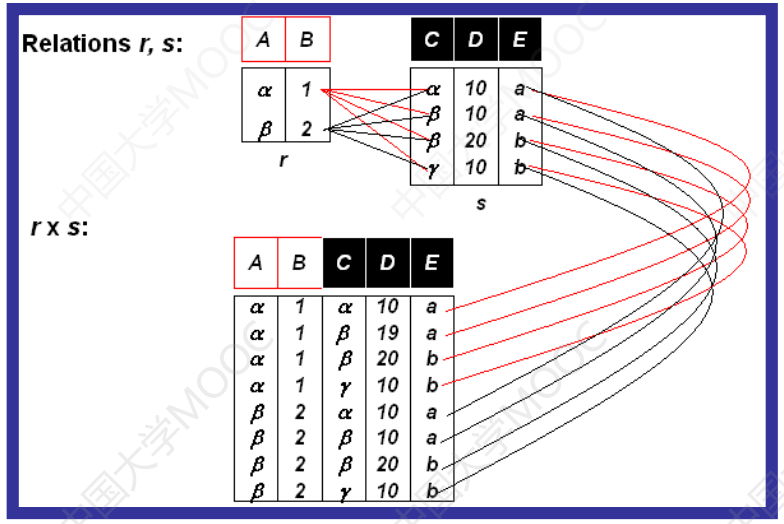
运算结果也是一个关系，记作： R x S, 它由关系R中的元组与关系S的元组进行所有可能的拼接(或串接)构成。 笛卡尔积可将两个表串接起来作为一个表进行操作

□ **数学描述**： $R \times S = \{ <a_1, a_2, \ldots, a_n, b_1, b_2, \ldots, b_m> \mid <a_1, a_2, \ldots, a_n> \in R \wedge <b_1, b_2, \ldots, b_m> \in S \}$

R		
A1	A2	A3
a	b	c
a	d	g
f	b	e

S		
B1	B2	B3
a	b	c
a	b	e
a	d	g
h	d	g

R × S					
A1	A2	A3	B1	B2	B3
a	b	c	a	b	c
a	b	c	a	b	e
a	b	c	a	d	g
a	b	c	h	d	g
a	d	g	a	b	c
a	d	g	a	b	e
a	d	g	a	d	g
a	d	g	h	d	g
f	b	e	a	b	c
f	b	e	a	b	e
f	b	e	a	d	g
f	b	e	h	d	g



怎样操纵数据库

第4种运算：【选择】运算

选择(Selection)

□ **定义**：给定一个关系R，同时给定一个选择的条件condition(简记con)，选择运算结果也是一个关系，记作$\sigma_{con}(R)$，它从关系R中选择出满足给定条件condition的元组构成

□ **数学描述**：$\sigma_{con}(R)=\{t \mid t \in R \wedge con(t) = \text{‘真’}\}$

R

A1	A2	A3
a	a	10
a	d	-4
f	b	5

$\sigma_{A3>0}(R)$

A1	A2	A3
a	a	10
f	b	5

$\sigma_{A2="a" \vee A2="b"}(R)$

A1	A2	A3
a	a	10
f	b	5

$\sigma_{A3>0 \wedge A1=A2}(R)$

A1	A2	A3
a	a	10

R

A1	A2	A3

怎样操纵数据库

第5种运算：【投影】运算

投影(Projection)

口 **定义**：给定一个关系R，投影运算结果也是一个关系，记作 $\Pi_A(R)$，它从关系R中选出属性包含在A中的列构成

口 **数学描述**： $\Pi_{A_{i1}, A_{i2}, \ldots, A_{ik}}(R) = \{ <t[A_{i1}], t[A_{i2}], \ldots, t[A_{ik}]> \mid t \in R \}$

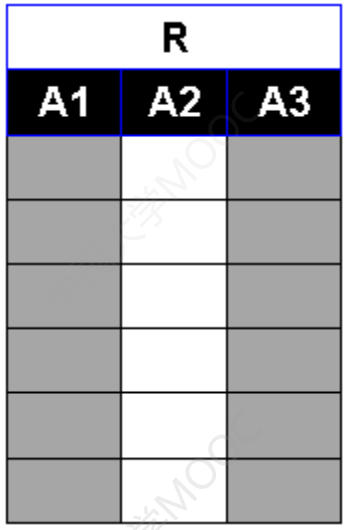
R		
A1	A2	A3
a	b	c
a	d	g
f	b	e

$\Pi_{A3}(R)$

A3
c
g
e

$\Pi_{A3, A1}(R)$

A3	A1
c	a
g	a
e	f



怎样操纵数据库

综合运用关系运算表达复杂的查询

查询表达式组合各种运算

Student

S#	Sname	Ssex	Sage	D#	Sclass
98030101	张三	男	20	03	980301
98030102	张四	女	21	03	980301
98030103	张五	男	19	03	980301
98040201	王三	男	18	04	980402
98040202	王四	男	21	04	980402
98050104	孙六	女	19	05	980501

SC

S#	C#	Score
98030101	001	92.0
98030101	002	85.0
98030101	003	88.0
98040202	002	90.5
98040202	003	80.0
98040202	001	55.0
98050104	003	56.0
98030102	001	54.0
98030102	002	85.0
98030102	003	48.0

- 查询学习课程号为002的学生学号和成绩

$$\pi_{S\#,\ Score}(\ \sigma_{C\#=\text{“002”}}(\ SC\ ))$$

- 查询学习课程号为001的学生学号、姓名

$$\pi_{S\#,Sname}(\sigma_{C\#=\text{“001”}\ \text{and}\ Student.S\#=SC.S\#}(\ Student \times\ SC\ ))$$

- 查询学习课程名称为数据结构的学生学号、姓名和这门课程的成绩

Course

C#	Cname	Chours	Credit	T#
001	数据库	40	6	001
003	数据结构	40	6	003
004	编译原理	40	6	001
005	C 语言	30	4.5	003
002	高等数学	80	12	004

$$Student \times SC \times Course$$

$$\sigma_{Cname=\text{“数据结构”}\ \text{and}\ Student.S\#=SC.S\#\ \text{and}\ SC.C\#=Course.C\#}(Student \times SC \times Course)$$

$$\pi_{S\#,Sname,\ Score}(\sigma_{Cname=\text{“数据结构”}\ \text{and}\ Student.S\#=SC.S\#\ \text{and}\ SC.C\#=Course.C\#}(Student \times SC \times Course))$$

若要掌握，则需要不断练习...

结构化数据库操纵语言

由关系模型到结构化数据库语言SQL

关系运算式

$$\Pi_{列名,\ldots,列名}(\sigma_{检索条件}(表名1 \times 表名2 \times \ldots))$$

$$\pi_{S\#,Sname,\ Score}(\sigma_{Cname=“数据结构” \wedge Student.S\#=SC.S\# \wedge Course.C\#=SC.C\#}(Student \times SC \times Course))$$

数据库语言SQL（结构化的查询语言）

Select **列名 [[, 列名] …]**

From **表名1 [[, 表名2], …]**

[*Where* 检索条件] ;

语义：将From后面的所有表串接起来，检索出满足“检索条件”的元组，并按给定的列名及顺序进行投影显示。

SQL语言有丰富的表达能力-如嵌套查询、模糊查询、统计、分组统计、分组过滤等，同学可继续自学之…

```
Select S#, Sname, Score
From Student, SC, Course
Where Cname= ‘数据结构’ and Student.S#=SC.S# and Course.C#=SC.C#;
```

结构化数据库管理系统的实现思路

关系运算➔数据库语言➔关系数据库管理系统

数据库管理系统

SQL语言

程序

```
Select   Sname   From  Student, SC
Where   Student.S# = SC.S#  and  SC.C# =   '001'
Order By  Score  DESC;
```

$$\Pi_{Sname}(\sigma_{student.s\#=sc.s\#}(Student \times SC))$$

复杂动作 = 基本动作的各种方式的组合

关系模型基本运算的各种组合

基本动作

基本动作	对基本动作的抽象
【并】操作	∪
【差】操作	−
【积】操作	×
【选择】操作	σ
【投影】操作	Π

指令

关系模型基本运算

解释这种组合，并按次序调用基本动作予以执行

程序执行机构

由结构化数据库管理到非结构化数据管理

关系数据库➔键值对数据库➔文档数据库

【NoSQL】“不仅是SQL，而不是NO-to-SQL”，不仅能管理结构化数据，而且能管理半结构化甚至非结构化数据的数据库。为处理大数据，多数都采用分布式存储技术

关系数据库（按行存储数据，按列按类型区分）

学生

学号	姓名	年龄	家庭住址
“00001”	“张一”	18	“吉林省长春市南关区”
“00002”	“张二”	19	“黑龙江省哈尔滨市南岗区”
“00003”	“张三”	20	“辽宁省沈阳市铁西区”
“00004”	“张四”	19	“四川省成都市武侯区”
“00005”	“张五”	18	“贵州省贵阳市花溪区”

第一种NoSQL数据库（按“属性名：属性值”对存储数据，均为字符串数据）

学生

对象标识（自动产生）	属性名	属性值
“00000001”	“学号”	“00001”
“00000002”	“学号”	“00002”
“00000003”	“学号”	“00003”
“00000004”	“学号”	“00004”
“00000005”	“学号”	“00005”
“00000001”	“姓名”	“张一”
“00000002”	“姓名”	“张二”
“00000003”	“姓名”	“张三”
“00000005”	“姓名”	“张五”
“00000001”	“年龄“	“18“
“00000002”	“年龄“	“19“
“00000005”	“年龄“	“18“
“00000001”	“家庭住址“	“吉林省长春市南关区”
“00000004”	“家庭住址“	“四川省成都市武侯区”
“00000005”	“家庭住址“	“贵州省贵阳市花溪区”
“00000005”	“家庭住址“	“黑龙江省哈尔滨市道里区”

第二种NoSQL数据库（按文档存储数据，一行是一个文档）

学生

对象标识（自动产生）	{ “属性名“ : ”属性值“, ”属性名“ : ”属性值“, …}
“00000001”	{ “学号”:“00001”,”姓名”:“张一”,“年龄” : “18”,“地址” : “吉林省长春市南关区”}
“00000002”	{ “学号”:“00002“,”姓名”:“张二”,“年龄” : “19”,“地址” : “黑龙江省哈尔滨市南岗区”}
“00000003”	{ “学号”:“00003”,”姓名”:“张三”,“年龄” : “20”,“地址” : “辽宁省沈阳市铁西区”}
“00000004”	{ “学号”:“00004”,”姓名”:“张四”,“年龄” : “19”,“地址” : “四川省成都市武侯区”}
“00000005”	{ “学号”:“00005”,”姓名”:“张五”,“年龄” : “18”,“地址” : “贵州省贵阳市花溪区”}

<标记>文本</标记>

“标记”：“文本”

第二种NoSQL数据库（按文档存储数据，一行是一个文档，文档中还可能嵌入文档）

学生

对象标识（自动产生）	{ “属性名“ : ”属性值“, ”属性名“ : ”属性值“, …}
“00000001”	{ “学号”:“00001”,”姓名”:“张一”,“年龄” : “18”,“地址” : {“省”:”吉林省”,“市”:”长春市”,”区” : “南关区”}}
“00000002”	{ “学号”:“00002“,”姓名”:“张二”,“年龄” : “19”,“地址” : {“省”:”黑龙江省”,“市”:”哈尔滨市”,”区” : “南岗区”}}
“00000003”	{ “学号”:“00003”,”姓名”:“张三”,“年龄” : “20”,“地址” : {“省”:”辽宁省”,“市”:”沈阳市”,”区” : “铁西区”} }
“00000004”	{ “学号”:“00004”,”姓名”:“张四”,“年龄” : “19”,“地址” : {“省”:”四川省”,“市”:”成都市”,”区” : “武侯区”}}
“00000005”	{ “学号”:“00005”,”姓名”:“张五”,“年龄” : “18”,“地址” : {“省”:”贵州省”,“市”:”贵阳市”,”区” : “花溪区”}}

与关系数据库相比，最大的优点：

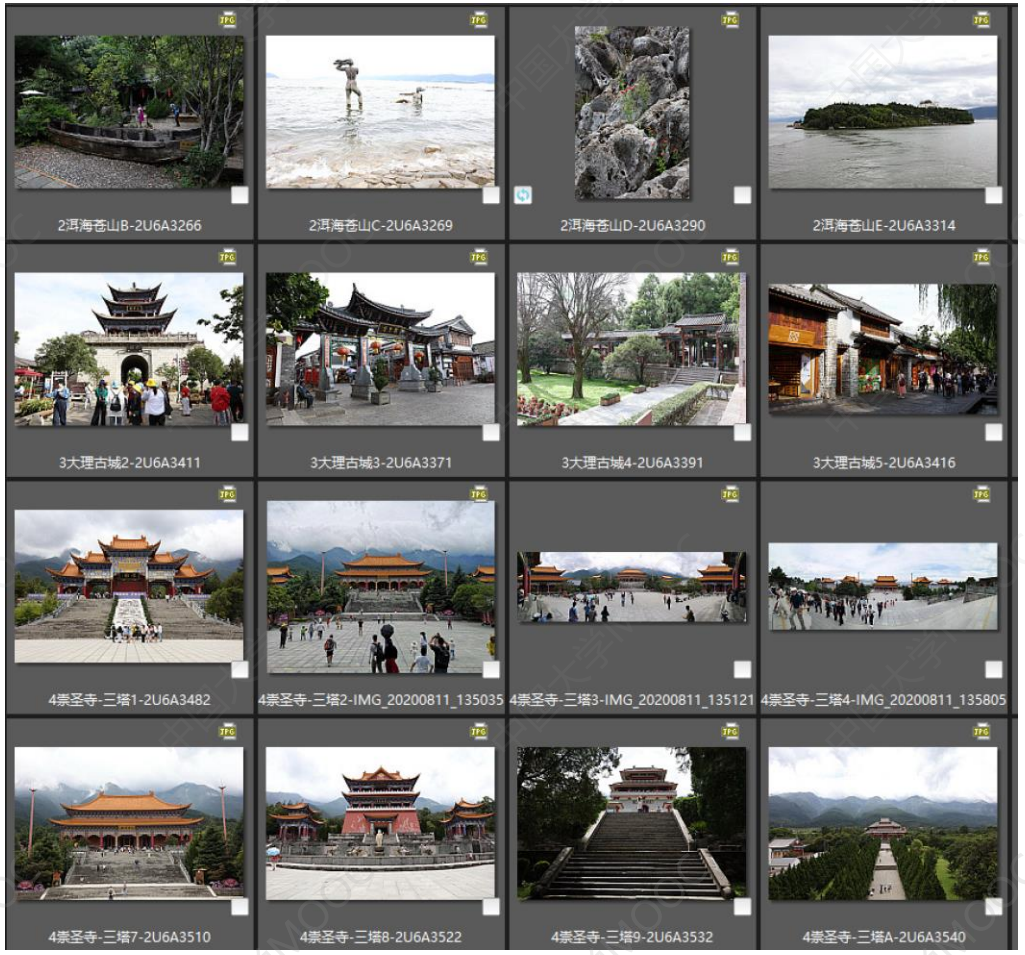
（1）可扩展性—可随时增加新属性列和减少属性列，而无须改变以前存储的数据。

（2）无需事先定义模式，可直接操纵数据

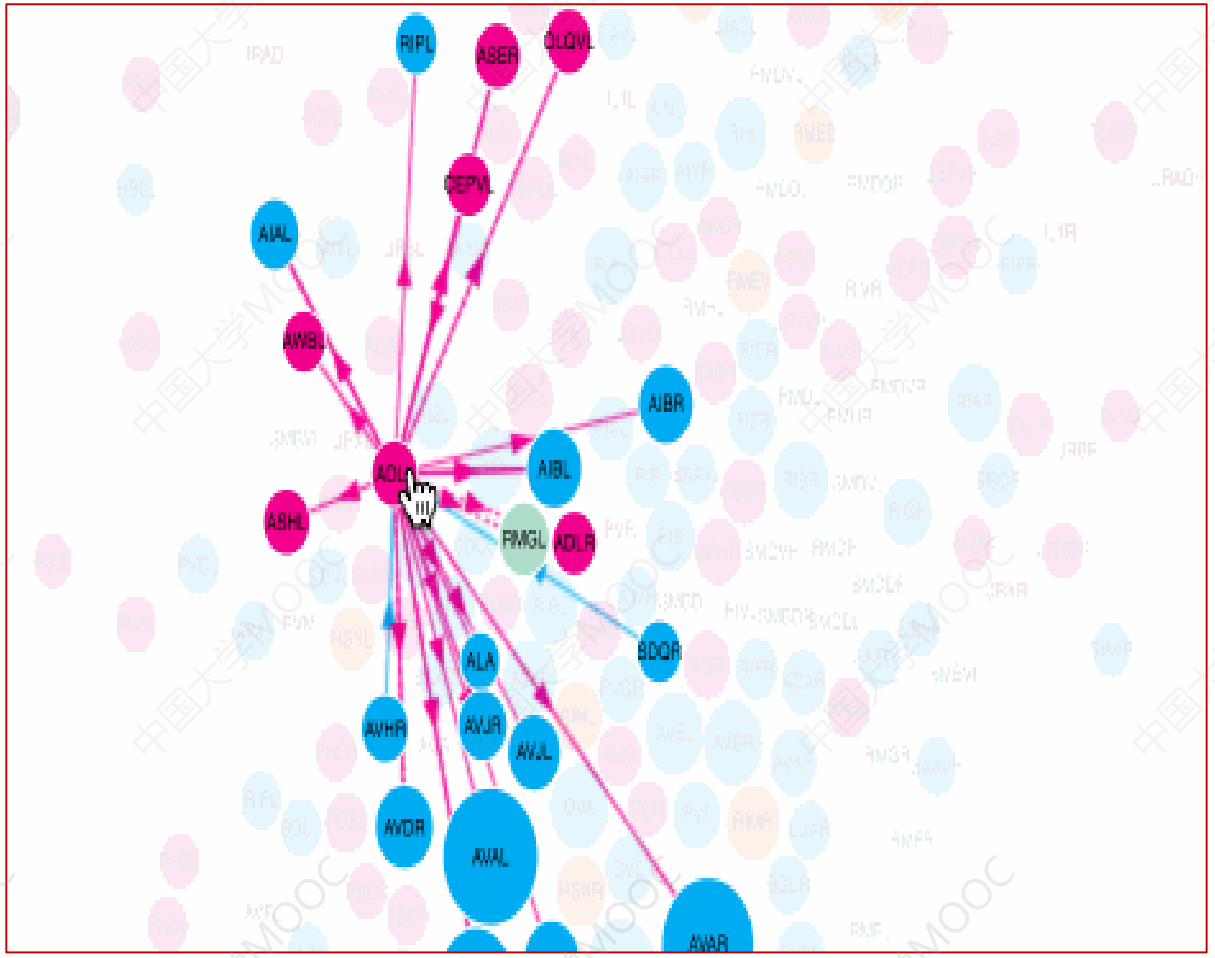
（3）并行/分布处理—可适应大规模并行/分布计算。

由结构化数据库管理到非结构化数据管理

其他非结构化数据库：图像数据库与图数据库



图像/视频/多媒体数据库



图数据库——一种由结点和边表达数据的数据库

从【表】的管理看数据库

小结

SC

S#	C#	Score
98030101	001	92.0
98030101	002	85.0

Student

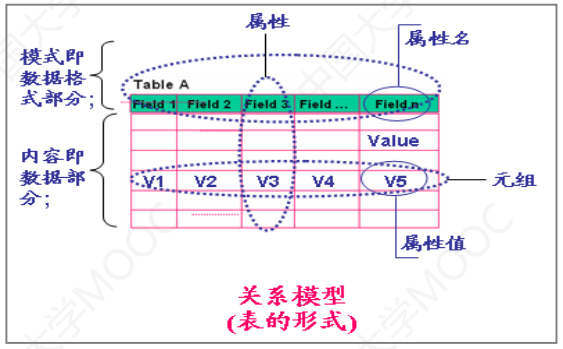
S#	Sname	Ssex	Sage	D#	Sclass
98030101	张三	男	20	03	980301

Course

C#	Cname	Chours	Credit	T#
001	数据库	40	6	001
003	数据结构	40	6	003
004	编译原理	40	6	001
005	C 语言	30	4.5	003
002	高等数学	80	12	004

抽象：区分并命名表的每一个形式要素

抽象



关系模型
(表的形式)

理论指导下的抽象：抽象更为严密

设计

先抽象再设计：从管理一个具体的表，到可管理所有的表

理论

域(Domain)：一组值的集合

笛卡尔积(Cartesian Product)

$D_1 \times D_2 \times \ldots \times D_n = \{ (d_1, d_2, \ldots, d_n) \mid d_i \in D_i, i=1,\ldots,n \}$

关系(Relation)

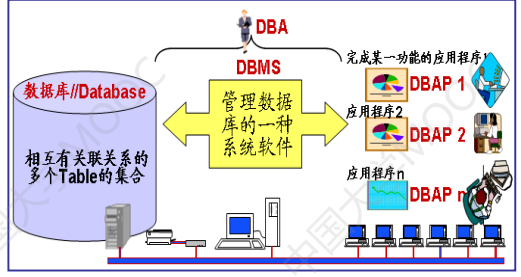
一组域D1，D2，…，Dn的笛卡尔积的子集

∪、－、×、σ、π

$\pi_{S\#,Sname}(\sigma_{C\#="001"}(Student \bowtie SC))$

```
CREATE  TABLE 表名(列名1  类型  [NOT NULL]
        [, 列名2  类型  [NOT NULL]].....);
SELECT  [DISTINCT]  列名1[, 列名2…]
FROM  表名1[, 表名2…]
[WHERE  条件1]
[GROUP BY  列名i1  [, 列名i2 …] [HAVING  条件2]]
[ORDER BY  表达式1  [ASC / DESC]…]
```

理论支持设计：设计正确性、完备性判定方法



理论：数学化逻辑严密化各种概念；

设计：语言/实现/系统

E.F.Codd，基于对“表(Table)”的理解：

- 提出了“关系”及关系模型， 提出了关系数据库理论
- 开创了数据库的时代，当前普遍应用的数据库管理系统的奠基者
- 获得了计算机领域最高奖“图灵奖”