由关系模型到
结构化数据库语言SQL

战德臣

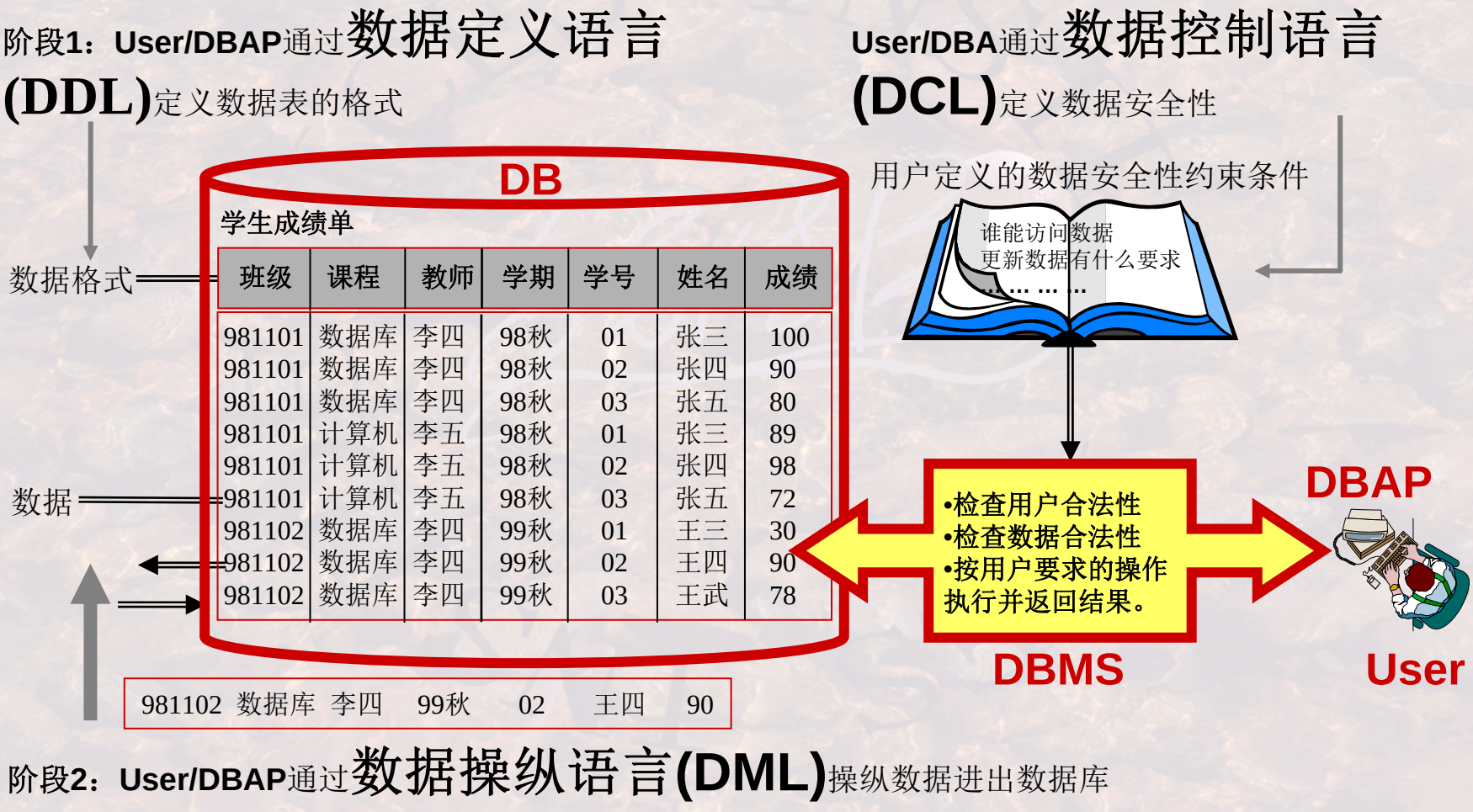
哈尔滨工业大学　教授.博士生导师
教育部大学计算机课程教学指导委员会委员

Research Center on Intelligent Computing for Enterprises & Services, Harbin Institute of Technology

由关系模型到结构化数据库语言SQL

(1)回顾：数据库系统的概念

战德臣 教授



战德臣 教授

关系运算式

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

□数据库语言SQL

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。

战德臣 教授

SQL: Structural Query Language

□SQL语言是数据库系统的标准语言，它可以定义数据库、操纵数据库和进行数据库控制。

□SQL语言主要由以下9个单词引导的操作语句来构成，但每一种语句都能表达复杂的操作请求。

- DDL语句引导词：Create(建立), Alter(修改), Drop(撤消)
  - ✓定义Database, Table, View, Index
- DML语句引导词：Insert(插入), Update(更新), Delete(删除), Select(查询)
  - ✓各种方式的更新与检索操作
  - ✓各种条件的查询操作，如连接查找，模糊查找，分组查找，嵌套查找等
  - ✓各种聚集操作，求平均、求和、…等，分组聚集，分组过滤等
- DCL语句引导词：Grant, Revoke
  - ✓安全性控制---授权和撤消授权

由关系模型到结构化数据库语言SQL
(4)用SQL语言创建数据库并定义表-简介

战德臣 教授

创建课程学习数据库：**SCT**

```
Create Database SCT;
```

Student

S#	Sname	Ssex	Sage	D#	Sclass

Course

C#	Cname	Chours	Credit	T#

➢定义学生表：**Student**

```
Create Table Student ( S# char(8) not null , Sname char(10),
        Ssex char(2), Sage integer, D# char(2), Sclass char(6) );
```

➢定义课程表：**Course**

```
Create Table Course ( C# char(3) , Cname char(12),
        Chours integer, Credit float(1), T# char(3) );
```

➢同学可自己定义其他的表： **Dept, Teacher, SC**

```
create table 表名( 列名 数据类型 [not null] [, 列名 数据类型 , … ]) ;
```

由关系模型到结构化数据库语言SQL
(5)用SQL语言在所定义表中增加记录-简介

战德臣 教授

- 追加学生表中的元组

```
Insert Into  Student
Values ( '98030101' , '张三', '男', 20,  '03',  '980301');

Insert Into  Student ( S#, Sname, Ssex, Sage, D# , Sclass)
Values ( '98030102' , '张四', '女', 20,  '03',  '980301');
```

- 追加课程表中的元组

```
Insert Into  Course                          /*所有列名省略，须与定义或存储的列名顺序一致
Values ( '001' , '数据库', 40, 6,  '001');

                                             /*如列名未省略，须与语句中列名的顺序一致
Insert Into  Course(Cname, C#, Credit, Chours, T#);
Values ('数据库', '001', 6, 20, '001');
```

Student

S#	Sname	Ssex	Sage	D#	Sclass
98030101	张三	男	20	03	980301
98030102	张四	女	20	03	980301
98030103	张五	男	19	03	980301
98040201	王三	男	20	04	980402
98040202	王四	男	21	04	980402
98040203	王五	女	19	04	980402

insert into 表名[(列名 [, 列名]...] ***values*** (值 [, 值] , ...);

战德臣 教授

已经建立好的数据库---供后面学习和训练使用

Student

S#	Sname	Ssex	Sage	D#	Sclass
98030101	张三	男	20	03	980301
98030102	张四	女	20	03	980301
98030103	张五	男	19	03	980301
98040201	王三	男	20	04	980402
98040202	王四	男	21	04	980402
98040203	王五	女	19	04	980402

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

SC

S#	C#	Score
98030101	001	92
98030101	002	85
98030101	003	88
98040202	002	90
98040202	003	80
98040202	001	55
98040203	003	56
98030102	001	54
98030102	002	85
98030102	003	48

Dept

D#	Dname	Dean
01	机电	李三
02	能源	李四
03	计算机	李五
04	自动控制	李六

Teacher

T#	Tname	D#	Salary
001	赵三	01	1200.00
002	赵四	03	1400.00
003	赵五	03	1000.00
004	赵六	04	1100.00

SQL-SELECT之简单使用

战德臣

哈尔滨工业大学　教授.博士生导师

教育部大学计算机课程教学指导委员会委员

Research Center on Intelligent Computing for Enterprises & Services, Harbin Institute of Technology

SQL-SELECT之简单使用
(1)基本检索操作的表达方法

战德臣 教授

Select 的简单语法形式为：

Select 列名 **[[,** 列名**] ...]**

From 表名

[***Where*** 检索条件 **];**

●语义：从<表名>所给出的表中，查询出满足<检索条件>的元组，并按给定的<列名>及顺序进行投影显示

$$\Pi_{\text{列名},\ldots,\text{列名}}(\sigma_{\text{检索条件}}(\text{表名}))$$

SQL-SELECT之简单使用
(2)基本检索条件书写

战德臣 教授

- 例如：检索学生表中所有学生的信息

```
Select    S#, Sname, Ssex, Sage, Sclass, D#
  From  Student ;

Select    *    From  Student ;          //如投影所有列，则可以用*来简写
```

- 再如：检索学生表中所有学生的姓名及年龄

```
Select    Sname, Sage                  //投影出某些列
  From  Student ;
```

- 再如：检索学生表中所有年龄小于19岁的学生的年龄及姓名

```
Select   Sage, Sname                   //投影的列可以重新排定顺序
  From  Student
Where   Sage <= 19;
```

SELECT … FROM … WHERE …

Student

S#	Sname	Ssex	Sage	D#	Sclass
98030101	张三	男	20	03	980301
98030102	张四	女	20	03	980301
98030103	张五	男	19	03	980301
98040201	王三	男	20	04	980402
98040202	王四	男	21	04	980402
98040203	王五	女	19	04	980402

SQL-SELECT之简单使用
(2)基本检索条件书写

战德臣 教授

- 例如：求或者学过001号课程, 或者学过002号课程的学生的学号

```
Select   S#   From  SC
Where   C# = '001'   OR   C#='002';
```

- 再例如：求既学过001号课程, 又学过002号课程的学生的学号? 如下书写SQL语句会得到正确结果吗？它能得到什么结果？怎样正确书写？

```
Select   S#   From  SC
Where   C# = '001'   AND   C#='002';
```

//正确的SQL语句在讲义后面的示例中讲解

对于每一行应用Where子句的条件

SC

S#	C#	Score
98030101	001	92
98030101	002	85
98030101	003	88
98040202	002	90
98040202	003	80
98040202	001	55
98040203	003	56
98030102	001	54
98030102	002	85
98030102	003	48

SELECT … FROM … WHERE …

SQL-SELECT之简单使用
(3)如何去掉检索结果的重复元组或记录呢?

战德臣 教授

检索结果去重复：DISTINCT

➢例如：在选课表中，检索成绩大于80分的所有学号

```
Select   S#
  From  SC
Where   Score > 80 ;  //有重复元组出现，比如一个同学两门以上课程大于80
```

```
Select   DISTINCT  S#
  From  SC
Where   Score > 80;   //重复元组被DISTINCT过滤掉，只保留一份
```

SC

S#	C#	Score
98030101	001	92
98030101	002	85
98030101	003	88
98040202	002	90
98040202	003	80
98040202	001	55
98040203	003	56
98030102	001	54
98030102	002	85
98030102	003	48

表(Table)和关系(Relation)在大部分情况下概念通用，但有细微差别: 前者可允许有重复元组, 而后者不允许

SQL-SELECT之简单使用
(4)如何对检索结果进行排序呢?

战德臣 教授

检索结果的排序

Select语句中结果排序是通过增加**order by**子句实现的

order by 列名 [***asc*** | ***desc***]

➢意义为结果按指定列名进行排序，若后跟**asc**或省略，则为升序；若后跟**desc,** 则为降序。

➢例如：按学号由小到大的顺序显示出所有学生的学号及姓名

```
Select   S#, Sname   From  Student
Order  By   S#  ASC ;
```

➢ 再如：检索**002**号课大于**80**分的所有同学学号并按成绩由高到低顺序显示

```
Select   S#   From  SC   Where   C# = '002'  and  Score > 80
Order  By   Score  DESC ;
```

SELECT … FROM … WHERE … ORDER BY …

战德臣 教授

模糊查询

- 比如检索“姓张的学生”，检索“张某某”，这类查询问题，Select语句是通过在检索条件中引入运算符like来表示的
- 含有like运算符的表达式

  列名 [*not*] *like* “字符串”
- 找出匹配给定字符串的字符串。其中给定字符串中可以出现%, _等匹配符.
- 匹配规则：
  - “%” 匹配零个或多个字符
  - “_” 匹配任意单个字符
  - “ \ ” 转义字符，用于去掉一些特殊字符的特定含义，使其被作为普通字符看待, 如用 “\%”去匹配字符%，用_ 去匹配字符_

SQL-SELECT之简单使用
(5)如何表达模糊性的查询内容呢?

- 例如：检索所有姓张的学生学号及姓名

```
Select   S#, Sname   From  Student
Where   Sname  Like   '张%' ;
```

- 再如：检索名字为张某某的所有同学姓名

```
Select   Sname   From  Student
Where   Sname Like '张_ _';
```

- 再如：检索名字不姓张的所有同学姓名

```
Select   Sname   From  Student
Where   Sname Not  Like '张%';
```

SQL-SELECT之简单使用
(6)小结?

战德臣 教授

- 例如：检索学生表中所有学生的信息

SELECT … FROM … WHERE …

SELECT DISTINCT … FROM … WHERE …

- 再如

SELECT … FROM … WHERE … ORDER BY …

- 再如

SELECT … FROM … WHERE … LIKE …

SELECT … FROM … WHERE …

Student

S#	Sname	Ssex	Sage	D#	Sclass
98030101	张三	男	20	03	980301
98030102	张四	女	20	03	980301
98030103	张五	男	19	03	980301
98040201	王三	男	20	04	980402
98040202	王四	男	21	04	980402
98040203	王五	女	19	04	980402

SQL-SELECT之多表联合操作

战德臣

哈尔滨工业大学　教授.博士生导师

教育部大学计算机课程教学指导委员会委员

Research Center on Intelligent Computing for Enterprises & Services, Harbin Institute of Technology

SQL-SELECT之多表联合操作
(1)多表联合检索的表达方法

战德臣 教授

- **Select** 的多表联合检索语句如下：

  Select 列名 **[[, 列名] …]**

  From 表名1, 表名2, …

  Where 检索条件；

- 相当于$\Pi_{列名,\ldots,列名}(\sigma_{检索条件}(表名1 \times 表名2 \times \ldots))$
- 检索条件中要包含连接条件，通过不同的连接条件可以实现各种连接操作。

SQL-SELECT之多表联合操作
(2)多表联合检索之连接条件书写

战德臣 教授

➢例如：按“001”号课成绩由高到低的顺序显示出所有学生的姓名(二表连接)

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

➢ 当多表连接时，如果两个表的属性名相同，则需采用表名.属性名方式来限定该属性是属于哪一个表

➢ 再如：按‘数据库’课程成绩由高到低顺序显示所有同学姓名(三表连接)

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

Student(S#,Sname,Ssex,Sage,D#,Sclass)
Course(C#,Cname,Chours,Credit,T#)
SC(S#,C#,Score)
Dept(D#,Dname,Dean)
Teacher(T#,Tname,D#,Salary)

SQL-SELECT之多表联合操作
(3)多表联合检索之表与列的别名

战德臣 教授

- 连接运算涉及到重名的问题，如两个表中的属性重名，连接的两个表重名(同一表的连接)等，因此需要使用别名以便区分
- **select**中采用别名的方式：

  Select 列名 **as** 列别名 **[[,** 列名 **as** 列别名**] …]**

  From 表名**1 as** 表别名**1,** 表名**2 as** 表别名**2, …**

  Where 检索条件；
- 上述定义中的**as** 可以省略
- 当定义了别名后，在检索条件中可以使用别名来限定属性

SQL-SELECT之多表联合操作
(3)多表联合检索之表与列的别名

战德臣 教授

➢例如：求有薪水差额的任意两位教师

```
Select   T1.Tname as Teacher1, T2.Tname  as Teacher2
From  Teacher  T1,  Teacher  T2
Where   T1.Salary > T2.Salary ;
```

➢求年龄有差异的任意两位同学的姓名

```
Select   S1.Sname as Stud1, S2.Sname  as Stud2
From  Student  S1,  Student  S2
Where   S1.Sage > S2.Sage ;
```

➢请同学书写一下：求‘001’号课程有成绩差的任意两位同学

➢有时表名很长时，为书写条件简便，也定义表别名，以简化书写。

Student(S#,Sname,Ssex,Sage,D#,Sclass)
Course(C#,Cname,Chours,Credit,T#)
SC(S#,C#,Score)
Dept(D#,Dname,Dean)
Teacher(T#,Tname,D#,Salary)

SQL-SELECT之多表联合操作
(4)多表联合检索之表与自身的连接

战德臣 教授

- 再如：求既学过“001”号课又学过 “002”号课的所有学生的学号(二表连接)

```
Select   SC1.S#   From  SC   SC1, SC  SC2
Where   SC1.S# = SC2.S#  and  SC1.C#='C01'
          and  SC2.C#='C02 ;
```

- 再如：求“C01”号课成绩比“C02”号课成绩高的所有学生的学号(二表连接)

```
Select   SC1.S#   From  SC   SC1, SC   SC2
Where   SC1.S# = SC2.S#  and  SC1.C#='C01'
          and  SC2.C#='C02'  and  SC1.Score > SC2.Score;
```

SC: SC1			SC: SC2		
SC1.S#	SC1.C#	SC1.Score	SC2.S#	SC2.C#	SC2.Score
S01	C01	80	S01	C02	90
S01	C01	80	S01	C03	85
S01	C02	90	S01	C01	80
S01	C02	90	S01	C03	85
S01	C03	85	S01	C01	80
S01	C03	85	S01	C02	90

SQL-SELECT之多表联合操作
(5)多表联合检索之语义之理解

战德臣 教授

- 正确理解汉语表达的查询语义，正确表达为SQL语句
- 例如：列出没学过李明老师讲授课程的所有同学的姓名？如下书写SQL语句会得到正确结果吗？它能得到什么结果？怎样正确书写？

```
Select   Sname   From  Student  S, SC, Course  C, Teacher  T
Where   T.Tname <> '李明'   and  C.C# = SC.C#
        and  SC.S# = S.S#  and  T.T# = C.T#;
```

//正确的SQL语句在讲义后面的示例中讲解

SQL-SELECT之多表联合操作
(6)多表联合检索之嵌套子查询

战德臣 教授

IN子查询。其基本语法为：

表达式 [*not*] *in* (子查询)

- 语义：判断某一表达式的值是否在子查询的结果中。

- 再例如：列出选修了**001**号课程的学生的学号和姓名

```
Select   S#, Sname   From  Student
Where   S#   in  ( Select   S#   From  SC  Where   C# = '001' ) ;
```

- 再例如：求既学过**001**号课程, 又学过**002**号课程的学生的学号

```
Select   S#   From  SC
Where   C# = '001'   and
        S#   in  ( Select   S#   From  SC  Where   C# = '002' ) ;
```

SQL-SELECT之多表联合操作
(6)多表联合检索之嵌套子查询

战德臣 教授

再例如：列出没学过李明老师讲授课程的所有同学的姓名?

```
Select   Sname   From  Student
Where   S#   not  in ( Select    S#    From  SC, Course  C, Teacher  T
                    Where   T.Tname = '李明'  and  SC.C# = C.C#
                    and  T.T# = C.T#  );
```

非相关子查询

外层查询

```
Select    Sname
From  Student
Where   S#   not  in (Select    S#
                    From  SC, Course  C, Teacher  T
                    Where   T.Tname = '李明'  and  SC.C# = C.C#
                              and  T.T# = C.T# ) ;
```

内层查询

- 内层查询独立进行，没有涉及任何外层查询相关信息的子查询被称为非相关子查询。

相关子查询

- 有时，内层查询需要依靠外层查询的某些参量作为限定条件才能进行，这样的子查询称为相关子查询。
- 外层向内层传递的参量需要使用外层的表名或表别名来限定
- 例如：求学过001号课程的同学的姓名

```
Select   Sname
From  Student  Stud
Where   S#   in ( Select    S#
                  From  SC
                  Where   S# = Stud.S#   and  C# = '001'  );
```

- 注意：相关子查询只能由外层向内层传递参数，而不能反之；这也称为变量的作用域原则。

SQL-SELECT之多表联合操作
(6)小结?

战德臣 教授

➢再如：求既学过“001”号课又学过 “002”号课的所有学生的学号(二表连接)

```
Select   SC1.S#  From  SC  SC1, SC  SC2
Where   SC1.S# = SC2.S#  and  SC1.C#='C01'
        and  SC2.C#='C02' ;
```

➢ 再如

```
Se
Wh
```

SELECT … FROM 表名1,表名2,… WHERE …

SELECT … FROM … WHERE … IN (SELECT … FROM … WHERE …)

SC: SC1　　SC: SC2

SC1.S#	SC1.C#	SC1.Score	SC2.S#	SC2.C#	SC2.Score
S01	C01	80	S01	C02	90
S01	C01	80	S01	C03	85
S01	C02	90	S01	C01	80
S01	C02	90	S01	C03	85
S01	C03	85	S01	C01	80
S01	C03	85	S01	C02	90

SQL-SELECT之分组聚集操作

战德臣

哈尔滨工业大学　教授.博士生导师

教育部大学计算机课程教学指导委员会委员

Research Center on Intelligent Computing for Enterprises & Services, Harbin Institute of Technology

结果计算与聚集函数

➢**select**子句可以是一些计算表达式或聚集函数，表明在选择和投影的同时直接进行一些运算，如下所示：

Select 列名 | **expr** | **agfunc(列名)** [[, 列名 | **expr** | **agfunc(列名)**] …]

From 表名**1** [, 表名**2** …]

[***Where*** 检索条件] ;

➢ 计算表达式可以是常量、列名或由常量、列名、特殊函数及算术运算符构成的算术运算式。

➢求有差额(差额>0)的任意两位教师的薪水差额

```
Select   T1.Tname as TR1, T2.Tname  as TR2, T1.Salary – T2.Salary
From  Teacher  T1,  Teacher  T2
Where   T1.Salary > T2.Salary;
```

战德臣 教授

结果计算与聚集函数

- **SQL提供了五个作用在简单列值集合上的内置聚集函数agfunc, 分别是：COUNT、SUM、AVG、MAX、MIN**
- **SQL聚集函数的参数类型、结果类型与作用如下：**

Name	Argument type	Result type	Description
Count	any (can be *)	numeric	count of occurrences
sum	numeric	numeric	sum of arguments
avg	numeric	numeric	average of arguments
max	char or numeric	same as arg	maximum value
min	char or numeric	same as arg	minimum value

战德臣 教授

结果计算与聚集函数

- 求教师的工资总额

```
Select  Sum(Salary)  From  Teacher;
```

- 求计算机系教师的工资总额

```
Select  Sum(Salary)  From  Teacher  T, Dept
Where  Dept.Dname = '计算机' and Dept.D# = T.D#;
```

- 求数据库课程的平均成绩

```
Select  AVG(Score)  From  Course  C, SC
Where  C.Cname = '数据库' and C.C# = SC.C#;
```

战德臣 教授

分组计算与聚集

- 为解决同时求解若干个集合的聚集运算问题，引出了分组的概念。
- SQL可以将检索到的元组按照某一条件进行分类，具有相同条件值的元组划到一个组或一个集合中，这一过程就是分组过程。
- 分组可以在基本Select语句基础上引入分组子句来完成：

```
Select   列名 | expr | agfunc(列名)  [[, 列名 | expr | agfunc(列名) ] … ]
  From   表名1 [, 表名2 … ]
[ Where   检索条件 ]
[ Group  by   分组条件 ] ;
```

- 分组条件可以是

  列名1, 列名2, …

SELECT … FROM … WHERE … GROUP BY … HAVING … ORDER BY …

SQL-SELECT之分组聚集操作
(2)SELECT之分组计算与聚集?

战德臣 教授

分组计算与聚集

- 例如： 求每一个学生的平均成绩

```
Select  S#, AVG(Score)  From  SC
Group  by  S#;
```

- 再如：求每一门课程的平均成绩

```
Select  C#, AVG(Score)  From  SC
Group  by  C#;
```

S#	C#	Score
98030101	001	92
98030101	002	85
98030101	003	88
98040202	002	90
98040202	003	80
98040202	001	55
98040203	003	56
98030102	001	54
98030102	002	85
98030102	003	48

Group by S#

S#	C#	Score
98030101	001	92
98040202	001	55
98030102	001	54
98030101	002	85
98030102	002	85
98040202	002	90
98040202	003	80
98030101	003	88
98040203	003	56
98030102	003	48

Group by C#

SQL-SELECT之分组聚集操作
(3)SELECT之分组过滤

战德臣 教授

分组过滤----过滤掉分组，而不是元组

➢求不及格课程超过两门的同学的学号，下述写法正确吗？

```
Select   S#  From  SC
Where  Score < 60  and  Count(*)>2
Group by  S#;
```

➢若要对分组(集合)进行条件过滤，可使用Having子句

```
Select   列名 | expr | agfunc(列名)  [[, 列名 | expr | agfunc(列名) ] … ]
 From   表名1 [, 表名2 … ]
[ Where   检索条件 ]
[ Group  by   分组条件 [ Having  分组过滤条件] ] ;
```

战德臣 教授

分组过滤----过滤掉分组，而不是元组

HAVING子句与WHERE子句表达条件的区别

每一分组检查满足与否的条件要用Having子句表达。

注意:不是每一行都检查，所以使用Having子句一定要有Group by子句

S#	C#	Score
98030101	001	92
98030101	002	85
98030101	003	88
98040202	002	90
98040202	003	80
98040202	001	55
98040203	003	56
98030102	001	54
98030102	002	85
98030102	003	48

每一行都要检查满足与否的条件要用WHERE子句表达

SQL-SELECT之分组聚集操作
(3)SELECT之分组过滤

战德臣 教授

分组过滤----过滤掉分组，而不是元组

- 例如 求不及格课程超过两门的同学的学号

```
Select   S#  From  SC
Where  Score < 60
Group by   S#   Having  Count(*)>2;
```

- 再如 求有10人以上不及格的课程号

```
Select   C#  From  SC
Where  Score < 60
Group by   C#   Having  Count(*)>10;
```

SQL-SELECT之分组聚集操作
(3)SELECT之分组过滤

战德臣 教授

分组过滤----过滤掉分组，而不是元组

- 例如：求有两门以上不及格课程的同学的学号及其平均成绩

```
Select   S#, Avg(Score)  From  SC
Where  Score < 60
Group by   S#   Having  Count(*)>2;
```

- 上述写法正确吗？
- 正确的如下书写，为什么呢?

```
Select   S#, AVG(Score)  From  SC
Where  S#  in
          ( Select   S#  From  SC
            Where  Score < 60
            Group by   S#   Having  Count(*)>2 )
Group by S# ;
```

战德臣 教授

SQL-SELECT的完整语法

```
Subquery ::==
 SELECT  [ ALL | DISTINCT ] { *  | expr [[AS]  c_alias] {, … } }
   FROM  tableref {, … }
  [WHERE search_condition]
  [GROUP BY column {, … }]
  [HAVING search_condition]
 | subquery [UNION [ALL] | INTERSECT [ALL] | EXCEPT [ALL]]
      [CORRESPONDING [BY] (colname {, … })] subquery;

Tableref  ::== tablename    [corr_name]

Select statement ::==
   Subquery  [ORDER  BY  result_column  [ASC | DESC] { , … }]
```

由数据库到数据挖掘I
-数据挖掘示例之背景与概念

战德臣

哈尔滨工业大学　教授.博士生导师

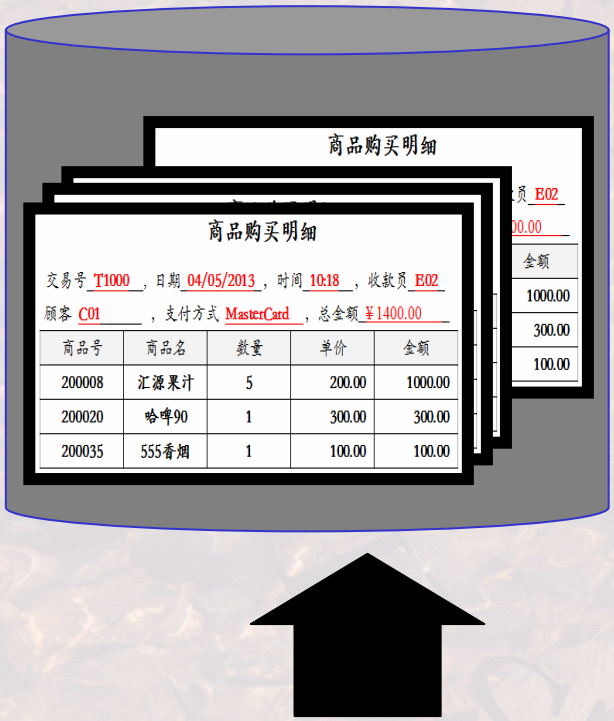
教育部大学计算机课程教学指导委员会委员

Research Center on Intelligent Computing for Enterprises & Services, Harbin Institute of Technology

由数据库到数据挖掘I-数据挖掘示例之背景与概念
(1)数据也是生产力？

战德臣 教授

数据对超市经营有无帮助呢?



客户购买习惯
商品组合方式及策略
… …

营销策略
价格策略
货源组织

由数据库到数据挖掘I-数据挖掘示例之背景与概念
(2)数据运用的前提—数据的聚集与管理?

战德臣 教授

超市数据库


销售单号:XS130113020013
顾客名称: 普通顾客

商品名称 折后价 数量 金额

欧登宝淡奶油
5900002 12.00 2 24.00
雪花烘培纸杯 高温杯/马芬杯/耐高温纸杯 10个
5900226 1.50 1 1.50
sweet烘培纸杯 高温杯/马芬杯/耐高温纸杯 1.5元/10个
5900061 1.50 1 1.50
裱花嘴 曲奇 嘴
5900410 3.20 1 3.20
水玉波点 /雪花/马芬杯/蛋糕杯 小号 3元/10个
5900073 3.00 1 3.00
中号格子烘培纸杯 高温杯/马芬杯/耐高温纸杯 1.5元/10个
5900093 1.50 1 1.50
邮费
5900177 1.00 5 5.00

消费7项,折后合计:39.7元
原价合计:39.7元,为您节省:0.0元
信用卡付款:¥39.70

收银台:可可
销售时间:2013-01-13 15:31:55

商品购买明细

交易号 T1000 , 日期 04/05/2013 , 时间 10:18 , 收款员 E02

顾客 C01 , 支付方式 MasterCard , 总金额 ¥1400.00

商品号	商品名	数量	单价	金额
200008	汇源果汁	5	200.00	1000.00
200020	哈啤90	1	300.00	300.00
200035	555香烟	1	100.00	100.00

商品购买单

交易号	日期	时间	收款员号	顾客号	支付方式	总金额
T1000	**04/05/2013**	**10:18**	**E02**	**C01**	**MasterCard**	**1400.00**
T1001	04/05/2013	11:10	E01	C03	Visa	1200.00
……	……					
T1101	04					

商品购买单明细

交易号	商品号	商品名	数量	单价	金额
T1000	**200008**	**汇源果汁**	**5**	**200.00**	**1000.00**
T1000	**200020**	**哈啤90**	**1**	**300.00**	**300.00**
T1000	**200035**	**555香烟**	**1**	**100.00**	**100.00**
T11001	200020	哈啤90	2	300.00	600.00
T11001	200009	巧克力	2	300.00	600.00
……	……	……	……	……	……
T1101	200008	汇源果汁	1	200.00	200.00
T1101	200020	哈啤90	1	300.00	300.00

由数据库到数据挖掘I-数据挖掘示例之背景与概念
(3)什么是数据挖掘?

战德臣 教授

数据挖掘，又称为数据库中知识发现，它是一个从大量数据中抽取挖掘出未知的、有价值的模式或规律等知识的复杂过程。简单地讲就是从大量数据中挖掘或抽取出知识。

- 概要归纳
- 关联分析
- 分类与预测
- 聚类分析
- 异类分析
- 演化分析

(4)怎样挖掘数据--一个例子？

ICES 战德臣 教授

数据挖掘之关联规则挖掘

商品购买明细

交易号 T1000，日期 04/05/2013，时间 10:18，收款员 E02

顾客 C01，支付方式 MasterCard，总金额 ￥1400.00

商品号	商品名	数量	单价	金额
08	汇源果汁	5	200.00	1000.00
20	哈啤90	1	300.00	300.00
200035	555香烟	1	100.00	100.00

商品的关联规则

“由尿布的购买，能够推断出啤酒的购买”

“尿布”⇒“啤酒” [支持度=2%，置信度=60%]

支持度2%意味着所分析事务的2%同时购买尿布和啤酒

置信度60%意味着购买尿布的顾客60%也购买啤酒。

是否相信这条规则呢?—让数据说话

ICES 战德臣 教授

关联规则挖掘相关的基本概念

1. 项、项集与事务

设 $P = \{p_1, p_2, ..., p_m\}$是所有**项(Item)**的集合。$D$是数据库中所有事务的集合，其中每个**事务T(Transaction)**是项的集合，是P的子集，即$T \subset P$；每一个事务有一个关键字属性，称作**交易号**或**事务号**以区分数据库中的每一个事务。设A 是一个**项集(ItemSet)**，事务T 包含A 当且仅当$A \subseteq T$。

2. 关联规则

关联规则是形如$A \Rightarrow B$ 的蕴涵式，即命题A(如“项集A的购买”)蕴涵着命题B(“如项集B的购买”)，或者说由命题A能够推出命题B，其中$A \subseteq P$，$B \subseteq P$，并且$A \cap B = \varnothing$。

商品购买明细

交易号 T1000 ，日期 04/05/2013 ，时间 10:18 ，收款员 E02

顾客 C01 ，支付方式 MasterCard ，总金额 ￥1400.00

商品号	商品名	数量	单价	金额
200008	汇源果汁	5	200.00	1000.00
200020	哈啤90	1	300.00	300.00
200035	555香烟	1	100.00	100.00

关联规则挖掘相关的基本概念

3. 支持度与置信度

***Support* ($A \Rightarrow B$)** = $P(A \cup B)$ = 包含A和B的事务数 ÷ D中事务总数。

***confidence* ($A \Rightarrow B$)** = $P(B|A)$ = 包含A和B的事务数 ÷ 包含A的事务数。

支持度反映一条规则的实用性，置信度反映规则的“值得信赖性”的程度

4. 强规则

同时满足最小支持度阈值(*min_s*)和最小置信度阈值(*min_c*)的规则称作强规则。

5. k-项集与k-频繁项集

项的集合称为项集，包含*k* 个项的项集称为***k*-项集**。

项集的出现频率是包含项集的事务数，简称为项集的频率、支持计数或计数。如果项集的出现频率大于或等于*min_s*与*D* 中事务总数的乘积, 则项集满足最小支持度*min_s*。如果项集满足最小支持度，则称它为频繁项集。频繁*k* -项集的集合通常记作*Lk*。

{面包，果酱} --- 2-项集

{面包，果酱，奶油} ---3-项集

ICES
战德臣 教授

关联规则挖掘的基本思想

找出所有频繁项集。依定义，这些项集出现的频率至少和预定义的最小出现频率一样。

如何挖掘频繁项集?

Apriori 算法

由频繁项集产生强关联规则。依定义，这些规则必须满足最小支持度和最小置信度。

由数据库到数据挖掘I-数据挖掘示例之背景与概念
(7)小结

战德臣 教授

数据挖掘之关联规则挖掘

{面包，果酱} --- 2-项集

{面包，果酱，奶油} ---3-项集

商品的关联规则

商品购买明细

交易号 T1000 ，日期 04/05/2013 ，时间 10:18 ，收款员 E02

顾客 C01 ，支付方式 MasterCard ，总金额 ￥1400.00

商品号	商品名	数量	单价	金额
08	汇源果汁	5	200.00	1000.00
20	哈啤90	1		300.00
				100.00

项 | K-项集 | 事务/交易 | K-频繁项集 | 关联规则 | 支持度 | 置信度

“尿布”⇒“啤酒” [支持度=2%，置信度=60%]

支持度2%意味着所分析事务的2%同时购买尿布和啤酒

置信度60%意味着购买尿布的顾客60%也购买啤酒。

是否相信这条规则呢?—让数据说话

由数据库到数据挖掘II
-数据挖掘示例之计算过程-1

战德臣

哈尔滨工业大学　教授.博士生导师

教育部大学计算机课程教学指导委员会委员

Research Center on Intelligent Computing for Enterprises & Services, Harbin Institute of Technology

(0)回顾：频繁项集的挖掘算法-Apriori算法的相关概念

ICES 战德臣 教授

关联规则挖掘的基本思想

项 | K-项集 | 事务/交易 | K-频繁项集 | 关联规则 | 支持度 | 置信度

{*面包，果酱*} --- 2-项集

{*面包，果酱，奶油*} ---3-项集

找出所有频繁项集。依定义，这些项集出现的频率至少和预定义的最小出现频率一样。

如何挖掘频繁项集?

Apriori 算法

由频繁项集产生强关联规则。依定义，这些规则必须满足最小支持度和最小置信度。

由数据库到数据挖掘II-数据挖掘示例之计算过程-1

(1)频繁项集发现的计算过程？

战德臣 教授

频繁项集挖掘算法计算示例

1.对问题域数据进行抽象

商品购买明细

交易号 T1000，日期 04/05/2013，时间 10:18，收款员 E02

顾客 C01，支付方式 MasterCard，总金额 ￥1400.00

商品号	商品名	数量	单价	金额
200008	汇源果汁	5	200.00	1000.00
200020	哈啤90	1	300.00	300.00
200035	555香烟	1	100.00	100.00

商品购买明细数据库

交易号	一次交易中购买的商品列表		交易号	一次交易中购买的商品列表
T0000	P1, P2, P3, P5		T0050	P1, P3, P5
T1000	P1, P2, P6, P8		T1500	P2, P4, P8
T2000	P2, P3, P7, P8		T2500	P1, P3, P5
T3000	P1, P2, P6		T3500	P2, P3, P7
T4000	P1, P2, P3, P5, P6, P7		T4500	P1, P2, P6, P8
T5000	P1, P3, P5, P6		T5500	P1, P2, P5, P6
T6000	P2, P3, P6		T6500	P1, P2, P5, P6
T7000	P1, P4, P6		T7500	P1, P2, P4, P6
T8000	P2, P3, P4, P5		T8500	P1, P2, P4, P5, P6
T9000	P3, P4, P5		T9500	P1, P2, P4, P5, P6
总交易次数：20				

战德臣 教授

频繁项集挖掘算法计算示例

2.形成候选1-项集，并求出频繁1-项集

候选1项集.

项集	支持度计数
{ P1 }	14
{ P2 }	15
{ P3 }	10
{ P4 }	7
{ P5 }	11
{ P6 }	12
{ P7 }	3
{ P8 }	3

频繁1项集．支持度计数≥最小支持度计数5 (min_sup=5/20=25%)

项集	支持度计数
{ P1 }	14
{ P2 }	15
{ P3 }	10
{ P4 }	7
{ P5 }	11
{ P6 }	12

战德臣 教授

频繁项集挖掘算法计算示例

3.形成候选2-项集，并求出频繁2-项集

候选2项集.

$C_2=L_1$ Join L_1

频繁1项集

项集	支持度计数
{ P1 }	14
{ P2 }	15
{ P3 }	10
{ P4 }	7
{ P5 }	11
{ P6 }	12

项集	支持度计数
{ P1, P2 }	10
{ P1, P3 }	5
{ P1, P4 }	4
{ P1, P5 }	9
{ P1, P6 }	11
{ P2, P3 }	6
{ P2, P4}	5
{ P2, P5}	7
{ P2, P6}	10
{ P3, P4 }	2
{ P3, P5 }	7
{ P3, P6 }	3
{ P4, P5 }	4
{ P4, P6 }	3
{ P5, P6 }	6

频繁2项集. 支持度计数≥最小支持度计数5

项集	支持度计数
{ P1, P2 }	10
{ P1, P3 }	5
{ P1, P5 }	9
{ P1, P6 }	11
{ P2, P3 }	6
{ P2, P4}	5
{ P2, P5}	7
{ P2, P6}	10
{ P3, P5 }	7
{ P5, P6 }	6

战德臣 教授

频繁项集挖掘算法计算示例

4.形成候选3-项集，并剪枝，进一步求出频繁3-项集

候选3项集.
$C_3=L_2$ Join L_2

频繁2项集

项集	支持度计数
{P1, P2}	10
{P1, P3}	5
{P1, P5}	9
{P1, P6}	11
{P2, P3}	6
{P2, P4}	5
{P2, P5}	7
{P2, P6}	10
{P3, P5}	7
{P5, P6}	6

项集	
{P1, P2, P3}	
{P1, P2, P5}	
{P1, P2, P6}	
{P1, P3, P5}	
{P1, P3, P6}	被剪掉,因{P3,P6}
{P1, P5, P6}	
{P2, P3, P4}	被剪掉,因{P3,P4}
{P2, P3, P5}	
{P2, P3, P6}	被剪掉,因{P3,P6}
{P2, P4, P5}	被剪掉,因{P4,P5}
{P2, P4, P6}	被剪掉,因{P4,P6}
{P2, P5, P6}	
{P3, P5, P6}	被剪掉,因{P3,P6}

候选3项集的支持度计数

项集	支持度计数
{P1, P2, P3}	2
{P1, P2, P5}	6
{P1, P2, P6}	8
{P1, P3, P5}	4
{P1, P5, P6}	6
{P2, P3, P5}	3
{P2, P5, P6}	5

频繁3项集

项集	支持度计数
{P1, P2, P5}	6
{P1, P2, P6}	8
{P1, P5, P6}	6
{P2, P5, P6}	5

战德臣 教授

频繁项集挖掘算法计算示例

5.迭代地求出最终结果-频繁项集

频繁3项集

项集	支持度计数
{ P1, P2, P5 }	6
{ P1, P2, P6 }	8
{ P1, P5, P6 }	6
{ P2, P5, P6 }	5

候选4项集---频繁4项集支持度计数>=5>

项集	支持度计数
{ P1, P2, P5, P6 }	5

项集	支持度计数
{ P1 }	14
{ P2 }	15
{ P3 }	10
{ P4 }	7
{ P5 }	11
{ P6 }	12
{ P1, P2 }	10
{ P1, P3 }	5
{ P1, P5 }	9
{ P1, P6 }	11
{ P2, P3 }	6
{ P2, P4}	5
{ P2, P5}	7
{ P2, P6}	10
{ P3, P5 }	7
{ P5, P6 }	5
{ P1, P2, P5 }	6
{ P1, P2, P6 }	8
{ P1, P5, P6 }	6
{ P2, P5, P6 }	5
{ P1, P2, P5, P6 }	5

频繁项集全集 = 频繁1项集 ∪ 频繁2项集 ∪ 频繁3项集 ∪ 频繁4项集

由数据库到数据挖掘II-数据挖掘示例之计算过程-1
(2)频繁项集的发现算法

战德臣 教授

发现频繁项集的Apriori 算法

输入：事务数据库D；最小支持度计数阈值min_s。

输出：D 中的频繁项集L。

算法：

1) L_1 = find_frequent_1_itemsets(D); //算法始自频繁一项集的产生结果L_1

2) **for** (k = 2; $L_{k-1} \neq \varnothing$; k++) { //自频繁2项集的获取开始迭代(k=2),直至频繁k-1项集为空时停止迭代.

3) C_k = aproiri_gen(L_{k-1},min_s); //由频繁k-1项集L_{k-1}产生候选k项集C_k 由下面的一个子过程实现;

4) **for each** transaction t∈ D { //对候选项集进行支持度计数，即对D中每一个事务进行处理;

5) C_t = subset(C_k,t); //从t的项集中找出是C_k中元素的候选子集C_t

6) **for each** candidate c∈ C_t { //是C_t中元素的项集，则其支持度计数加1

7) c.count++; //支持度计数

8) }

9) }

10) L_k={c∈ C_k | c.count ≥ min_s} //形成频繁k-项集

11) }

12) **return** L = $\bigcup_k L_k$;

procedure apriori_gen(L_{k-1}: frequent (k-1)-itemset) //由频繁k-1项集L_{k-1}，产生候选k项集的子过程

1) **for each** itemset $l_1 \in L_{k-1}$ {

2) **for each** itemset $l_2 \in L_{k-1}$ { //对L_{k-1}中每一个项集与其中的另一个项集进行组合

3) **if** ($l_1[1]=l_2[1] \wedge ... \wedge (l_1[k-2]=l_2[k-2]) \wedge (l_1[k-1]<l_2[k-2])$ **then** {

4) c = l_1 (Join) l_2; //如果可连接，则连接起来，生产候选k-项集C_k

5) **if** has_infrequent_subset(c,L_{k-1}) **then**

6) **delete** c; // 剪枝，如果c包含有不是频繁k-1项集的子集，则删除c

7) **else add** c to C_k;

8) }

9) }}

10) **return** Ck;

procedure has_infrequent_subset(c:candidate k-itemset; L_{k-1}:frequent (k-1)-itemset)

// 判断一个候选k-项集是否包含有不是频繁k-1项集的项集

1) **for each** (k-1)-subset s **of** c

2) **if** s ∉ L_{k-1} **then**

3) return TRUE;

4) **return** FALSE;

由数据库到数据挖掘II
-数据挖掘示例之计算过程-2

战德臣

哈尔滨工业大学　教授.博士生导师

教育部大学计算机课程教学指导委员会委员

Research Center on Intelligent Computing for Enterprises & Services, Harbin Institute of Technology

战德臣 教授

关联规则挖掘的基本思想

项 | K-项集 | 事务/交易 | K-频繁项集 | 关联规则 | 支持度 | 置信度

{*面包，果酱*} --- 2-项集
{*面包，果酱，奶油*} ---3-项集

找出所有频繁项集。依定义，这些项集出现的频率至少和预定义的最小出现频率一样。

由频繁项集产生强关联规则。依定义，这些规则必须满足最小支持度和最小置信度。

项集	支持度计数
{ P1 }	14
{ P2 }	15
{ P3 }	10
{ P4 }	7
{ P5 }	11
{ P6 }	12
{ P1, P2 }	10
{ P1, P3 }	5
{ P1, P5 }	9
{ P1, P6 }	11
{ P2, P3 }	6
{ P2, P4}	5
{ P2, P5}	7
{ P2, P6}	10
{ P3, P5 }	7
{ P5, P6 }	5
{ P1, P2, P5 }	6
{ P1, P2, P6 }	8
{ P1, P5, P6 }	6
{ P2, P5, P6 }	5
{ P1, P2, P5, P6 }	5

由数据库到数据挖掘II-数据挖掘示例之计算过程-2
(1)什么是关联规则

战德臣 教授

商品的关联规则

商品购买明细

交易号 T1000，日期 04/05/2013，时间 10:18，收款员 E02

顾客 C01，支付方式 MasterCard，总金额 ¥1400.00

商品号	商品名	数量	单价	金额
08	汇源果汁	5	200.00	1000.00
20	哈啤90	1	300.00	300.00
200035	555香烟	1	100.00	100.00

“由尿布的购买，能够推断出啤酒的购买”

“尿布”⇒“啤酒” [支持度=2%，置信度=60%]

支持度2%意味着所分析事务的2%同时购买尿布和啤酒

置信度60%意味着购买尿布的顾客60%也购买啤酒。

由数据库到数据挖掘II-数据挖掘示例之计算过程-2
(2)关联规则的产生

战德臣 教授

关联规则的生成计算示例

关联规则

{P1,P2,P5,P6}可以产生的潜在规则A⇒B,
其中A∪B={P1,P2,P5,P6}, A∩B=∅.

项集 A	项集 A 支持度计数(支持度)	项集 B	项集 A∪B 的支持度计数(支持度)	置信度＝项集(A∪B)的支持度÷项集 A 的支持度
{ P1, P2, P5 }	6 (30%)	{ P6 }	5 (25%)	5/6=83.33%
{ P1, P2, P6 }	8 (40%)	{ P5 }	5 (25%)	5/8=62.50%
{ P2, P5, P6 }	5 (25%)	{ P1 }	5 (25%)	5/5=100.00%
{ P1, P5, P6 }	6 (30%)	{ P2 }	5 (25%)	5/6=83.33%
{ P1, P2 }	10 (50%)	{ P5, P6}	5 (25%)	5/10=50.00%
{ P1, P5 }	9 (45%)	{ P2, P6}	5 (25%)	5/9=55.55%
{ P1, P6 }	11 (55%)	{ P2, P5}	5 (25%)	5/11=45.45%
{ P2, P5 }	7 (35%)	{ P1, P6}	5 (25%)	5/7=71.42
{ P2, P6 }	10 (50%)	{ P1, P5}	5 (25%)	5/10=50.00%
{ P5, P6 }	6 (30%)	{ P1, P2}	5 (25%)	5/6=83.33%
{ P1 }	14 (70%)	{ P2, P5, P6}	5 (25%)	5/14=35.71%
{ P2 }	15 (75%)	{ P1, P5, P6}	5 (25%)	5/15=33.33%
{ P5 }	11 (55%)	{ P1, P2, P6}	5 (25%)	5/11=45.45%
{ P6 }	12 (60%)	{ P1, P2, P5}	5 (25%)	5/12=41.66

项集	支持度计数
{ P1 }	14
{ P2 }	15
{ P3 }	10
{ P4 }	7
{ P5 }	11
{ P6 }	12
{ P1, P2 }	10
{ P1, P3 }	5
{ P1, P5 }	9
{ P1, P6 }	11
{ P2, P3 }	6
{ P2, P4}	5
{ P2, P5}	7
{ P2, P6}	10
{ P3, P5 }	7
{ P5, P6 }	5
{ P1, P2, P5 }	6
{ P1, P2, P6 }	8
{ P1, P5, P6 }	6
{ P2, P5, P6 }	5
{ P1, P2, P5, P6 }	5

频繁项集

战德臣 教授

关联规则的生成计算示例

输出的规则表，$A \cap B = \varnothing$，“购买A能够推出购买B”．置信度>＝70%的规则.

项集 A	项集 A 支持度计数(支持度)	项集 B	项集 A∪B 的支持度计数(支持度)	置信度＝项集(A∪B)的支持度÷项集 A 的支持度
{ P1, P2, P5 }	6 (30%)	{ P6 }	5 (25%)	5/6=83.33%
{ P2, P5, P6 }	5 (25%)	{ P1 }	5 (25%)	5/5=100.00%
{ P1, P5, P6 }	6 (30%)	{ P2 }	5 (25%)	5/6=83.33%
{ P2, P5 }	7 (35%)	{ P1, P6}	5 (25%)	5/7=71.42
{ P5, P6 }	6 (30%)	{ P1, P2}	5 (25%)	5/6=83.33%

由数据库到数据挖掘II-数据挖掘示例之计算过程-2

(2)关联规则的产生

关联规则的生成计算示例

组合形成规则表，频繁3项集能推出哪些频繁项集？置信度标记红色为置信度>＝70%的规则．支持度标记蓝色的为满足置信度前提下的支持度>=50%的规则

项集 A	项集 A 支持度计数(支持度)	项集 B	项集 A∪B 的支持度计数(支持度)	置信度＝项集(A∪B)的支持度÷项集 A 的支持度
{P1, P2, P5}	6 (30%)	{P6}	5 (25%)	5/6=83.33%
{P1, P2, P6}	8 (40%)	{P5}	5 (25%)	5/8=62.50%
{P2, P5, P6}	5 (25%)	{P1}	5 (25%)	5/5=100.00%
{P1, P5, P6}	6 (30%)	{P2}	5 (25%)	5/6=83.33%
{P1, P2}	10 (50%)	{P5, P6}	5 (25%)	5/10=50.00%
{P1, P2}	10 (50%)	{P5}	6 (30%)	6/10=60.00%
{P1, P2}	10 (50%)	{P6}	8 (40%)	8/10=80.00%
{P1, P5}	9 (45%)	{P2, P6}	5 (25%)	5/9=55.55%
{P1, P5}	9 (45%)	{P6}	6 (30%)	6/9=66.66%
{P1, P5}	9 (45%)	{P2}	6 (30%)	6/9=66.66%
{P1, P6}	11 (55%)	{P2, P5}	5 (25%)	5/11=45.45%
{P1, P6}	11 (55%)	{P2}	8 (40%)	8/11=72.72%
{P1, P6}	11 (55%)	{P5}	6 (30%)	6/11=54.54%
{P2, P5}	7 (35%)	{P1, P6}	5 (25%)	5/7=71.42
{P2, P5}	7 (35%)	{P1}	6 (30%)	6/7=85.71%
{P2, P5}	7 (35%)	{P6}	5 (25%)	5/7=71.42%
{P2, P6}	10 (50%)	{P1, P5}	5 (25%)	5/10=50.00%
{P2, P6}	10 (50%)	{P1}	8 (40%)	8/10=80.00%
{P2, P6}	10 (50%)	{P5}	5 (25%)	5/10=50.00%
{P5, P6}	6 (30%)	{P1, P2}	5 (25%)	5/6=83.33%
{P5, P6}	6 (30%)	{P1}	6 (30%)	6/6=100.00%
{P5, P6}	6 (30%)	{P2}	5 (25%)	5/6=83.33%
{P1}	14 (70%)	{P2, P5, P6}	5 (25%)	5/14=35.71%
{P1}	14 (70%)	{P2, P5}	6 (30%)	6/14=42.85%
{P1}	14 (70%)	{P2, P6}	8 (40%)	8/14=57.14%
{P1}	14 (70%)	{P5, P6}	6 (30%)	6/14=42.85%
{P1}	14 (70%)	{P2}	10 (50%)	10/14=71.42%
{P1}	14 (70%)	{P5}	9 (45%)	9/14=64.28%
{P1}	14 (70%)	{P6}	11 (55%)	11/14=78.57%
A 为{P2},{P5},{P6}能推出邻由 B，可类同 A 为{P1}时那样处理，此处略				

战德臣 教授

关联规则的生成计算示例

最终输出的规则表

项集 A	项集 A 支持度计数(支持度)	项集 B	项集 A∪B 的支持度计数(支持度)	置信度＝项集(A∪B)的支持度÷项集 A 的支持度
{ P1, P2 }	10 (50%)	{ P6 }	8 (40%)	8/10=80.00%
{ P1, P6 }	11 (55%)	{ P2 }	8 (40%)	8/11=72.72%
{ P2, P6 }	10 (50%)	{ P1 }	8 (40%)	8/10=80.00%
{ P1 }	14 (70%)	{ P2 }	10 (50%)	10/14=71.42%
{ P1 }	14 (70%)	{ P6 }	11 (55%)	11/14=78.57%
A 为{P2},{P5}, {P6}能推出哪些 B，可类同 A 为{P1}时那样处理，此处略.				

"P1,P2"⇒"P6"[支持度=50%，置信度=80%]
"P1,P6"⇒"P2"[支持度=55%，置信度=72.72%]
"P2,P6"⇒"P1"[支持度=50%，置信度=80%]
"P1"⇒"P2"[支持度=70%，置信度=71.42%]
"P1"⇒"P6"[支持度=70%，置信度=78.57%]

由数据库到数据挖掘II-数据挖掘示例之计算过程-2
(3)还能挖掘什么？

战德臣 教授

关联规则挖掘

单维度单层次规则

***buys*(*X*,"*面包*")⇒*buys*(*X*," *果酱*")** X代表顾客

多维度多层次规则

***age*(*X* ,"30...39") ∧ *income*(*X* ,"42*K*...48*K*")⇒*buys*(*X* ,"*high _ resolution _ TV*")**

由数据库到数据挖掘II-数据挖掘示例之计算过程-2
(3)还能挖掘什么？

战德臣 教授

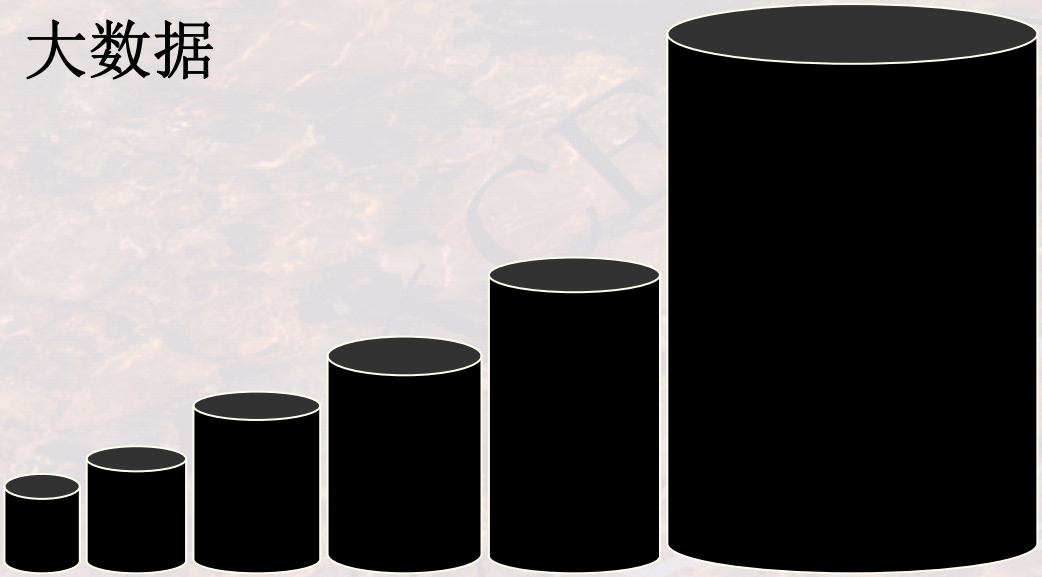
Web数据挖掘

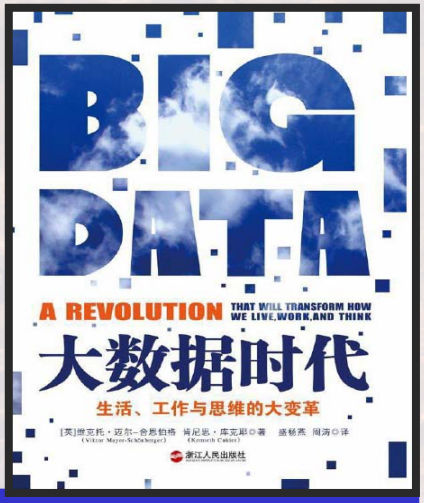
	“微博”挖掘	“超市数据”挖掘
数据基本组织形式	文本---非结构化数据	“表”---结构化数据
被挖掘数据D的集合	众多人众多次：发表的微博	众多人众多次：购买的商品
事务数据T的涵义	一次发表的“微博”可以看作是“若干词汇”的集合	一次购买的商品可以看作是“若干商品”的集合
项的集合	“词汇”的集合	“商品”的集合
频繁项集	频繁使用的“词汇”集合	频繁购买的“商品”集合
规则“A ⇒ B”	使用了“词汇A”也使用了“词汇”B	购买了“商品A”也购买了“商品B”
规则挖掘的意义	通过分析，可发现“可以组合在一起的关键词汇”，进而进行主题词设置、读者兴趣引导，以提高某主题的关注度、粉丝的聚集度等	通过分析，可发现“可被组合在一起的商品”进而进行位置、政策等的调整，以提高客户的购买兴趣等

由数据库到数据挖掘II-数据挖掘示例之计算过程-2
(5)小结

战德臣 教授

大数据





只求关系，不求因果

不要相信经验，一切以数据说话

大数据环境下什么不能发生呢?

bit & Byte

1KB(Kilobyte) = 2^{10}字节

1MB(Megabyte) = 2^{10}KB

1GB(Gigabyte) = 2^{10}MB

1TB(Trillionbyte) = 2^{10}GB = 2^{20}MB

1PB(Petabyte) = 2^{10}TB = 2^{30}MB

1EB(Exabyte) = 2^{10}PB = 2^{40}MB

1ZB(Zettabyte) = 2^{10}EB = 2^{50}MB

1YB(Yottabyte) = 2^{10}ZB = 2^{60}MB

1BB(Brontobyte) = 2^{10}YB = 2^{70}MB

数据抽象与设计I
---抽象之理解-区分-命名-表达

战德臣

哈尔滨工业大学　教授.博士生导师

教育部大学计算机课程教学指导委员会委员

Research Center on Intelligent Computing for Enterprises & Services, Harbin Institute of Technology

战德臣 教授

抽象-理论-设计概述

◆抽象、理论和设计是计算学科的三种形态或三种过程

◆设计--是指构造计算系统来改造世界的手段，是工程的主要内容，只有设计才能造福于人类(设计的价值)

◆理论--是发现世界规律的手段，理论，如果不能指导设计，则是反映不出其价值的；设计，如果没有理论指导，则设计的严密性、可靠性、正确性是没有保证的(理论的价值)

◆抽象--是感性认识世界的手段。理论和设计的前提都需要抽象，没有抽象二者都是没有办法达成目标的(抽象的价值)

数据抽象与设计I---抽象之理解-区分-命名-表达

(1)科学方法论中关于“抽象”的描述

战德臣 教授

- 抽象：指在思维中对同类事物去除其现象的次要方面，抽取其共同的主要方面，从而做到从个别中把握一般，从现象中把握本质的认知过程和思维方法。
- 抽象：一方面是建立对客观事物进行抽象描述的方法(方法论)，另一方面是要采用现有的抽象方法建立具体问题的概念模型，从而实现对客观世界的感性认识(方法论的应用)。
- 抽象：是现实事物的概念化。

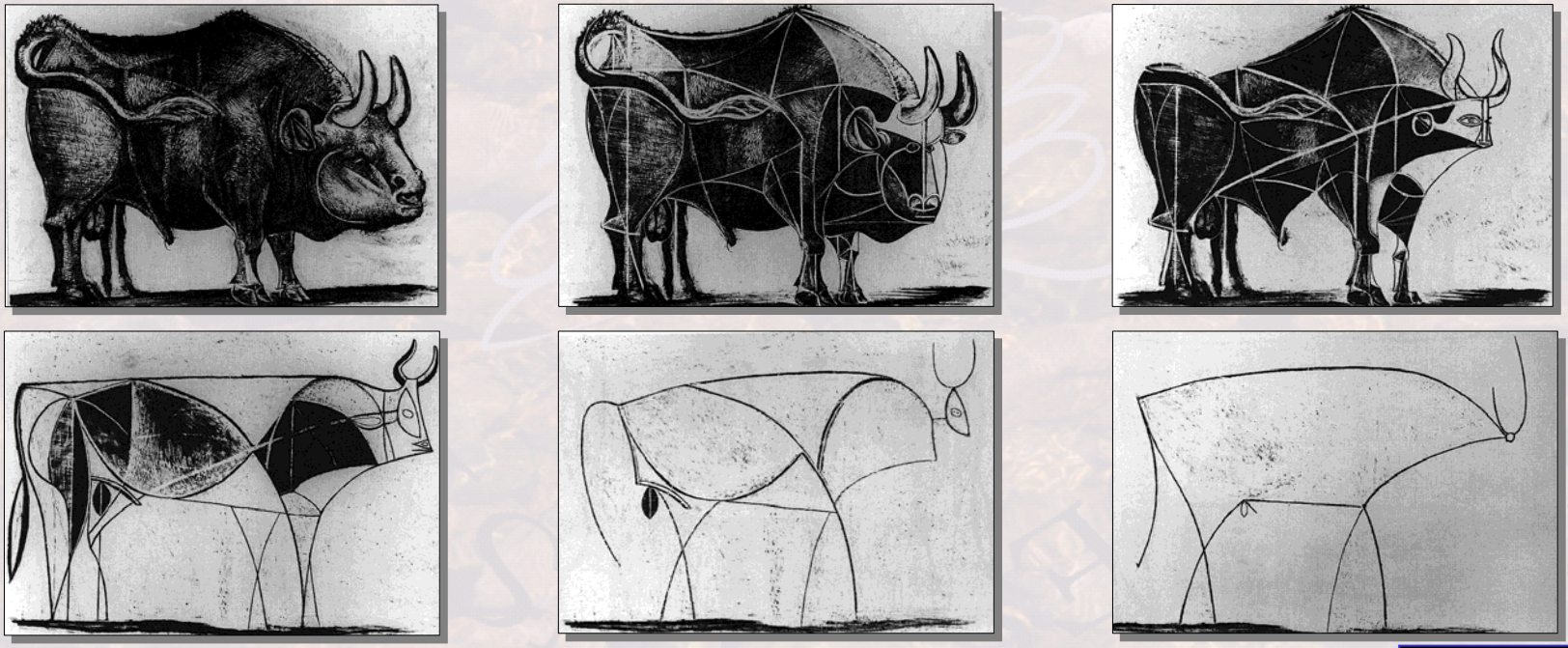
怎样进行抽象？有没有抽象的技巧？

数据抽象与设计I---抽象之理解-区分-命名-表达
(2)怎样进行“抽象”?

战德臣 教授

抽象：是指由具体事物中发现其本质性特征和方法的过程。

具体化



Quelle: Piccasso

抽象化

抽象就是“逐渐剥离语义、逐渐去掉细节”

数据抽象与设计I---抽象之理解-区分-命名-表达
(2)怎样进行“抽象”?

战德臣 教授

抽象：就是寻找相同的形式，处理可变的内容

“理解 → 区分 → 命名 → 表达”

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

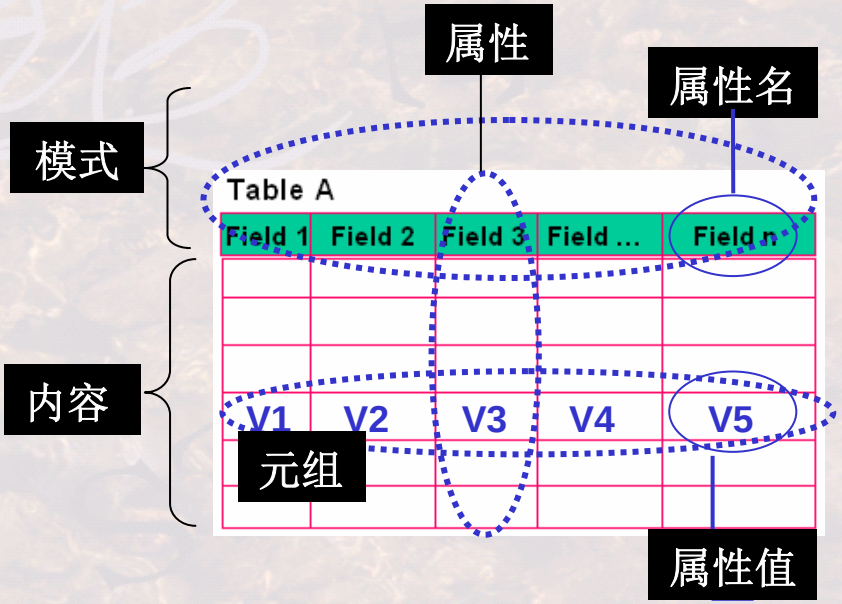
学生成绩单

班级	课程	教师	学			
981101	数据库	李四	98			
981101	数据库	李四	98			
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

.....

若干具有相似性的具体的表

抽象：区分并命名表的每一个形式要素



用抽象出的概念表示的表的形式

数据抽象与设计I---抽象之理解-区分-命名-表达
(2)怎样进行“抽象”?

战德臣 教授

抽象：就是共性中寻找差异,差异中寻找共性

“理解 → 区分 → 命名 → 表达”

商品(大类)= “食品”

		时间			
		1 季	2 季	3 季	4 季
地区	南岗区	35000	45000	53000	52000
	道里区	45000	33000	28000	46000

与前一个示例一样吗? 怎样进行抽象呢?

商品(大类)= “食品”

			时间										
			1 季			2 季			3 季			4 季	
			1 月	2 月	3 月	4 月	5 月	6 月	7 月	8 月	9 月	10 月	11 月	12 月
地区	南岗区	南岗区一分店	8000	8000	4000	12000	9000	9000	8000	8000	9000	8000	10000	10000
		南岗区四分店	4000	4000	7000	10000	7000	8000	9000	10000	9000	8000	8000	8000
	道里区	道里区二分店	8000	4000	8000	6000	4000	3000	4000	6000	4000	6000	5000	7000
		道里区三分店	8000	8000	9000	8000	6500	5500	5000	5000	4000	10000	9000	9000

战德臣 教授

抽象：就是共性中寻找差异,差异中寻找共性

“理解 → 区分 → 命名 → 表达”

二维交叉表

分地区(区县)分时间(季度)“食品”类商品销售对比分析表

商品(大类)= “食品”

地区	时间	1季	2季	3季	4季
南岗区		35000			
道里区		45000			

维度-2

维度--1

交叉格

<(1季，南岗区)，35000>
<(时间维度，地区维度)，销售额>

分地区(分店)分时间(月)销售对比分析表

商品(大类)= “食品”

地区		时间	1季			2季			3季			4季		
			1月	2月	3月	4月	5月	6月	7					
南岗区	南岗区一分店		8000	8000	4000	12000	9000	9000	8					
南岗区	南岗区四分店		4000	4000	7000	10000	7000	8000	9000	10000	9000	8000	8000	8000
道里	道里区二分店									00	4000	6000	5000	7000
										00	4000	10000	9000	9000

层次/粒度-1

多维度多粒度交叉表

层次/粒度-2

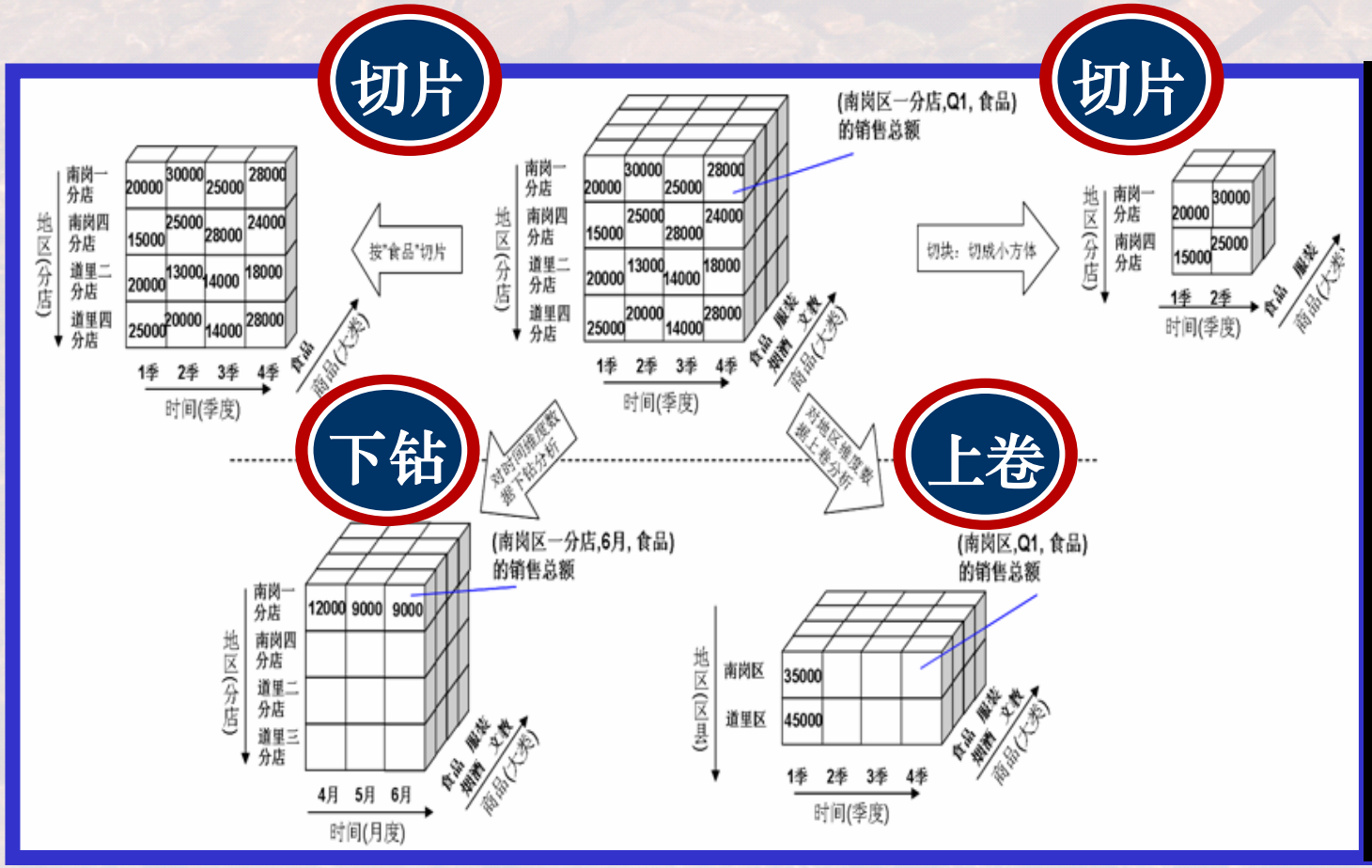
<(1季2月，南岗区四分店)，4000>
<(时间维度，地区维度)，销售额>

数据抽象与设计I---抽象之理解-区分-命名-表达
(2)怎样进行“抽象”?

ICES 战德臣 教授

抽象：就是共性中寻找差异,差异中寻找共性

“理解 ➔ 区分 ➔ 命名 ➔ 表达”



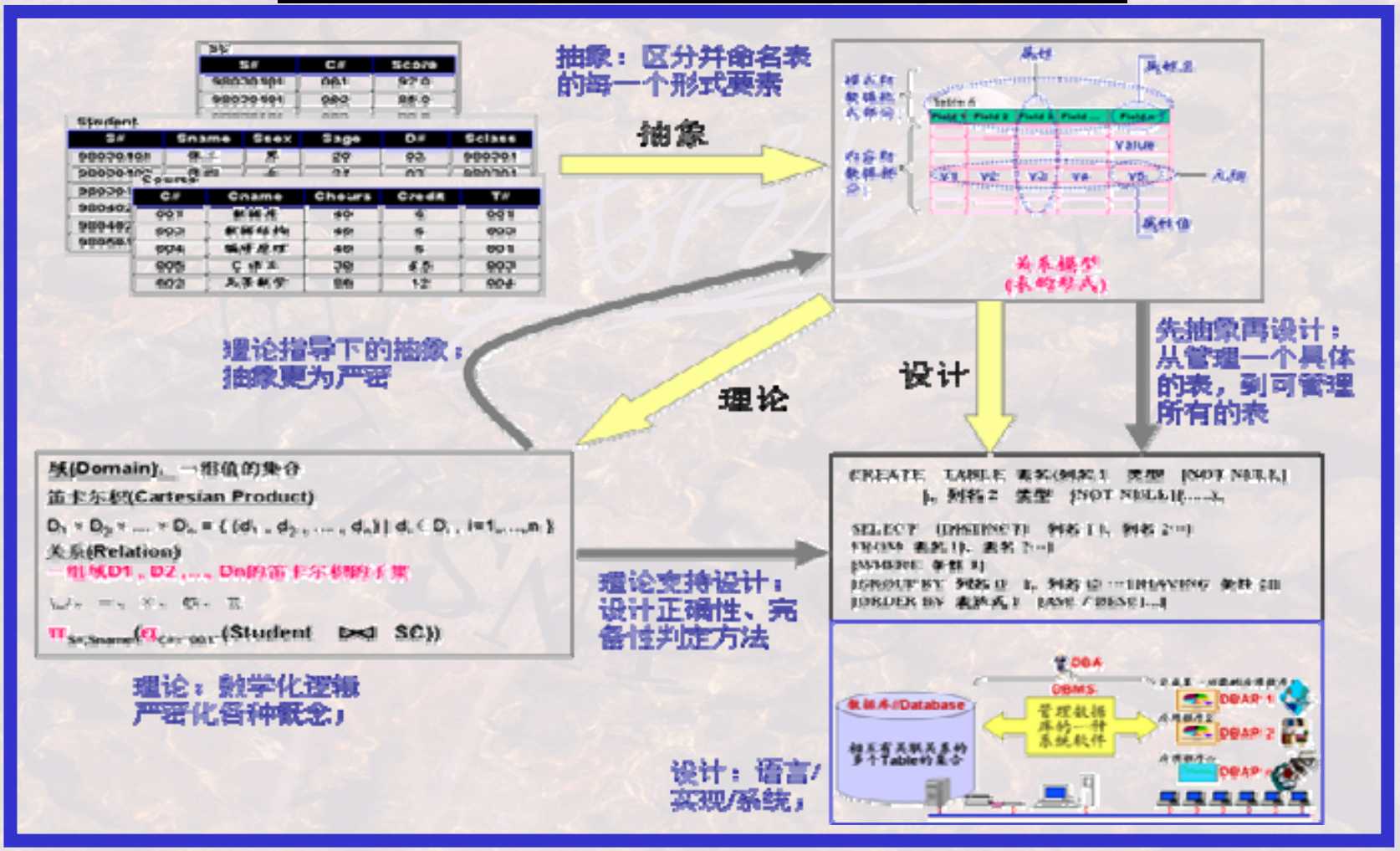
数据的抽象
- 维度
- 粒度/概念层次
- 度量-交叉格
- “事实”

操作的抽象
- 下钻, 上卷
- 切片/切块
- 转轴

战德臣 教授

抽象结果的表达---用设计手段表达? 用数学手段表达?

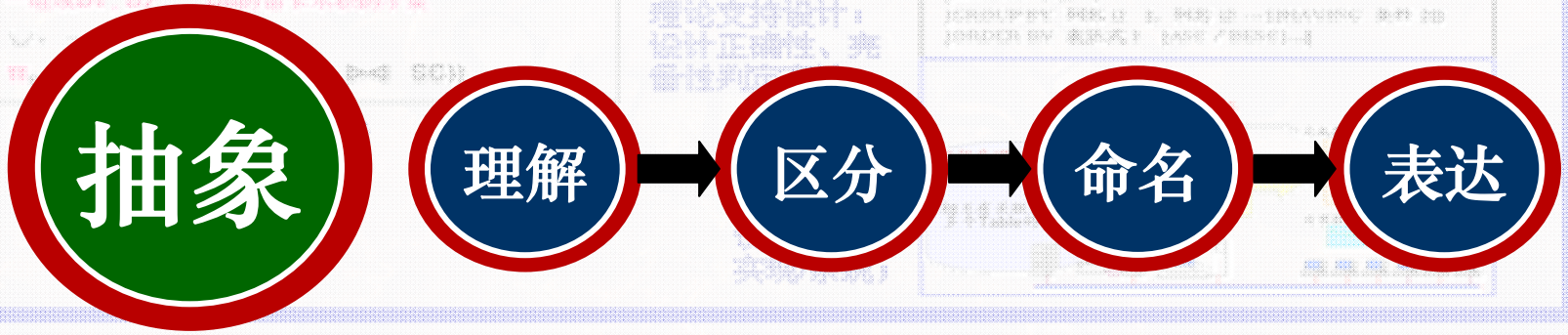
“理解 ➔ 区分 ➔ 命名 ➔ 表达”



战德臣 教授

最关键的

- 抽象的基本特征：本质性、区分与概念
- 记住：
  - 共性中寻找差异，差异中寻找共性----思维方法
  - 寻找相同的形式，处理可变的内容----计算学科“抽象”过程的本质
  - 理解➔区分➔命名➔表达----计算学科“抽象”的过程



数据抽象与设计I
---抽象之不同层次

战德臣

哈尔滨工业大学　教授.博士生导师

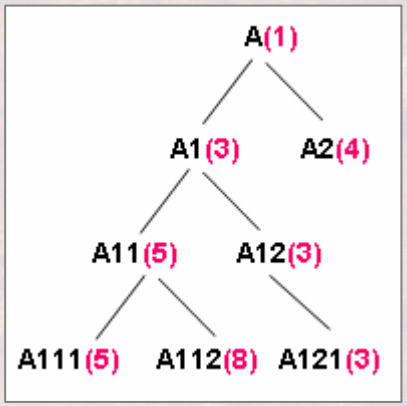
教育部大学计算机课程教学指导委员会委员

Research Center on Intelligent Computing for Enterprises & Services, Harbin Institute of Technology

数据抽象与设计I---抽象之不同层次
(1)“抽象”层次?

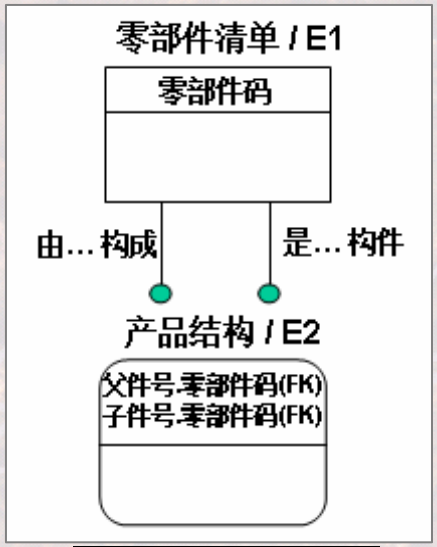
战德臣 教授

分层级抽象



待表达的问题：产品构成关系

概念的抽象



概念级抽象

关系数据表的抽象

零部件清单

零部件码	零部件名称	其他属性……
A	产品A	
A1	部套A1	
A2	部套A2	
A11	部件A11	
A12	部件A12	
A111	零件A111	
A112	零件A112	
A121	零件A121	

产品结构

父件号	子件号	数量
A	A1	3
A	A2	4
A1	A11	5
A1	A12	3
A11	A111	5
A11	A112	8
A12	A121	3

实现级抽象

不同层面的抽象：现实世界➔概念/信息世界➔计算机世界
逻辑世界(语义结构)➔物理世界(存储结构)

数据抽象与设计I---抽象之不同层次
(2)方法论及其应用?

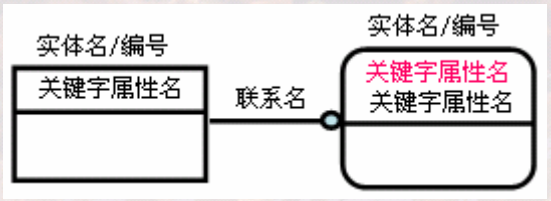
战德臣 教授

分层级抽象

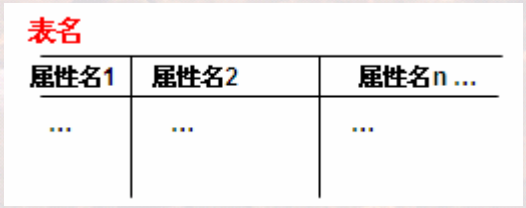
抽象方法/或称方法论
(抽象过程;抽象结果的表达方法)

按抽象方法进行抽象

概念的一般性表达方法

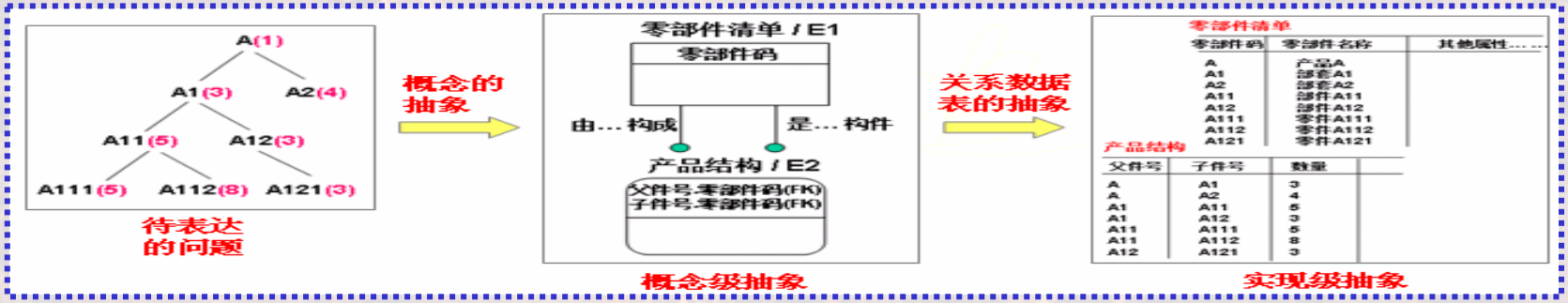


关系数据表的一般性表达方法



用一般性方法指导抽象

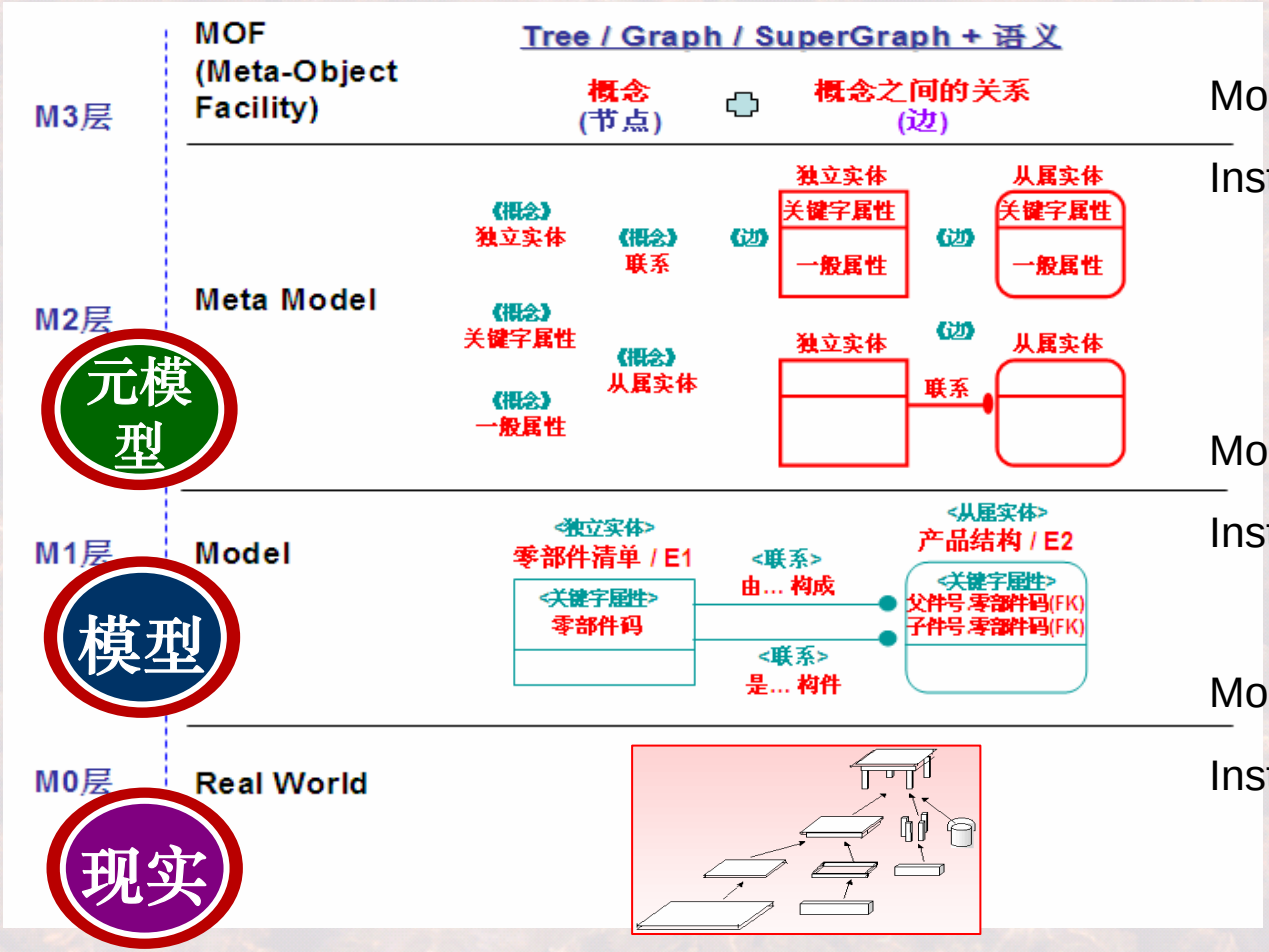
用一般性方法指导抽象



战德臣 教授

建模层次

建模的不同层次：模型与元模型，模型(型)与实例(值)



战德臣 教授

最关键的

数据抽象与设计II
---设计之形式-构造-自动化

战德臣

哈尔滨工业大学　教授.博士生导师

教育部大学计算机课程教学指导委员会委员

Research Center on Intelligent Computing for Enterprises & Services, Harbin Institute of Technology

战德臣 教授

抽象-理论-设计概述

◆抽象、理论和设计是计算学科的三种形态或三种过程

◆设计--是指构造计算系统来改造世界的手段，是工程的主要内容，只有设计才能造福于人类(设计的价值)

◆理论--是发现世界规律的手段，理论，如果不能指导设计，则是反映不出其价值的；设计，如果没有理论指导，则设计的严密性、可靠性、正确性是没有保证的(理论的价值)

◆抽象--是感性认识世界的手段。理论和设计的前提都需要抽象，没有抽象二者都是没有办法达成目标的(抽象的价值)

战德臣 教授

科学方法论中关于“设计”的描述

◆设计：源于工程并用于系统或设备的开发以实现给定的任务

◆设计：作为变革控制和利用自然界的手段，必须以对自然规律的认识为前提，可以是科学形态的认识，也可以是经验形态的认识。

◆设计：要达到变革控制和利用自然界的目的，必须创造出相应的人工系统和人工条件，还必须认识自然规律在这些人工系统中和人工条件下的具体表现形式。

◆设计形态具有较强的实践性、社会性和综合性。因此要记住：设计要对社会和人类有贡献，要有责任感!

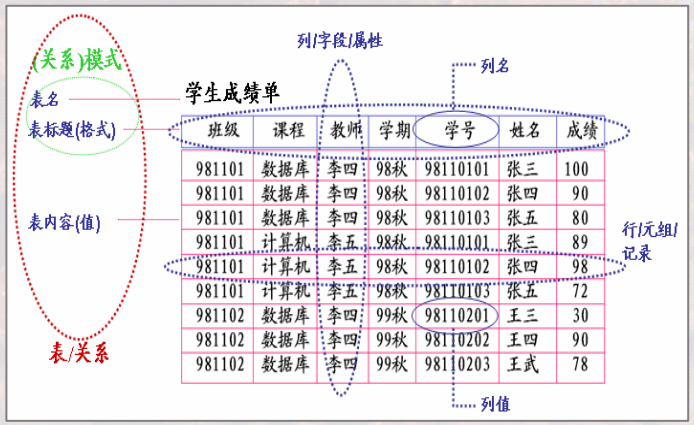
怎样进行设计?

战德臣 教授

设计：是构建计算系统的过程，是技术、原理在计算系统中实现的过程

◆计算机语言/协议的设计和实现是重要的“设计”形态的内容

“形式➔构造➔自动化”



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

抽象形态

设计：语法化各种概念，便于操作及自动化

计算机语言

```
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]...]
```

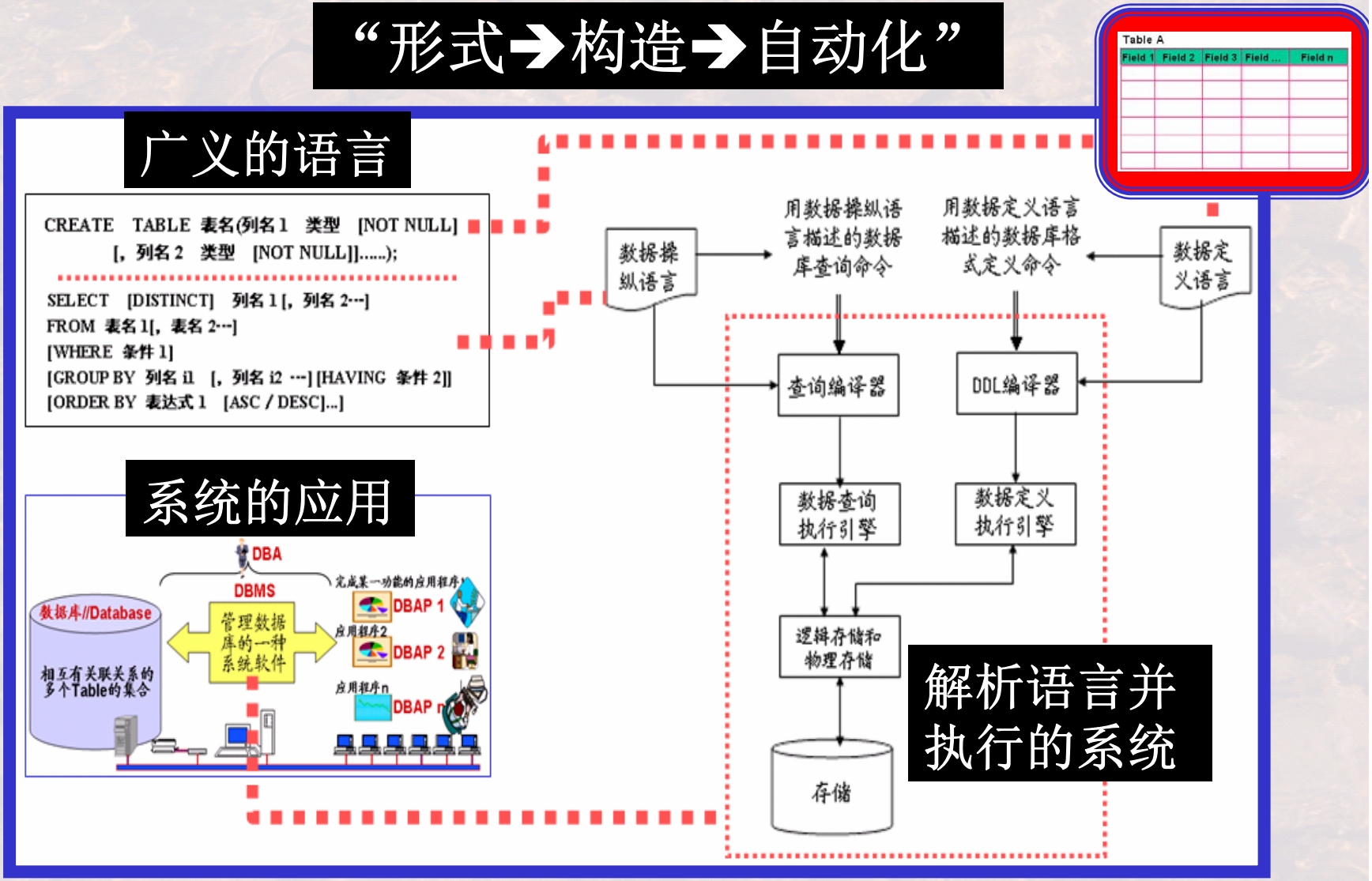
设计形态

数据抽象与设计II---设计之形式-构造-自动化?
(2)怎样进行“设计”?

战德臣 教授

基于广义的计算机语言的设计和实现软件/硬件/网络系统

“形式➔构造➔自动化”

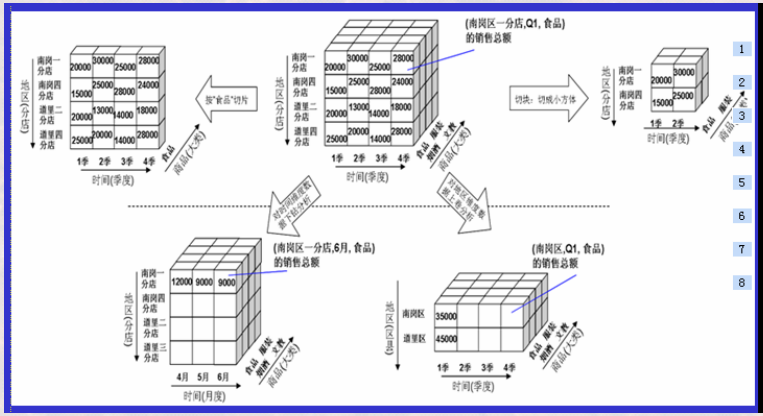


数据抽象与设计II---设计之形式-构造-自动化?
(2)怎样进行“设计”?

战德臣 教授

◆广义的计算机语言/协议的设计和实现是重要的“设计”形态的内容

“形式➔构造➔自动化”



数据的抽象
- 维度
- 粒度/概念层次
- 度量-交叉格
- “事实”

操作的抽象
- 下钻, 上卷
- 切片/切块
- 转轴

Cube语言-语法结构

```
define cube <cube_name> [<dimension_list>] : <measure_list>
define dimension <dimension_name> as (<attribute_or_subdimension_list>)
```

```
define cube 销售数据方体 [地区, 时间, 商品]:  销售额 = sum(金额)
define dimension 时间 as (时间标识, 日, 月, 季, 年)
define dimension 商品 as (商品号, 商品名, 细类, 大类)
define dimension 地区 as (地区标识, 分店, 区县)
```

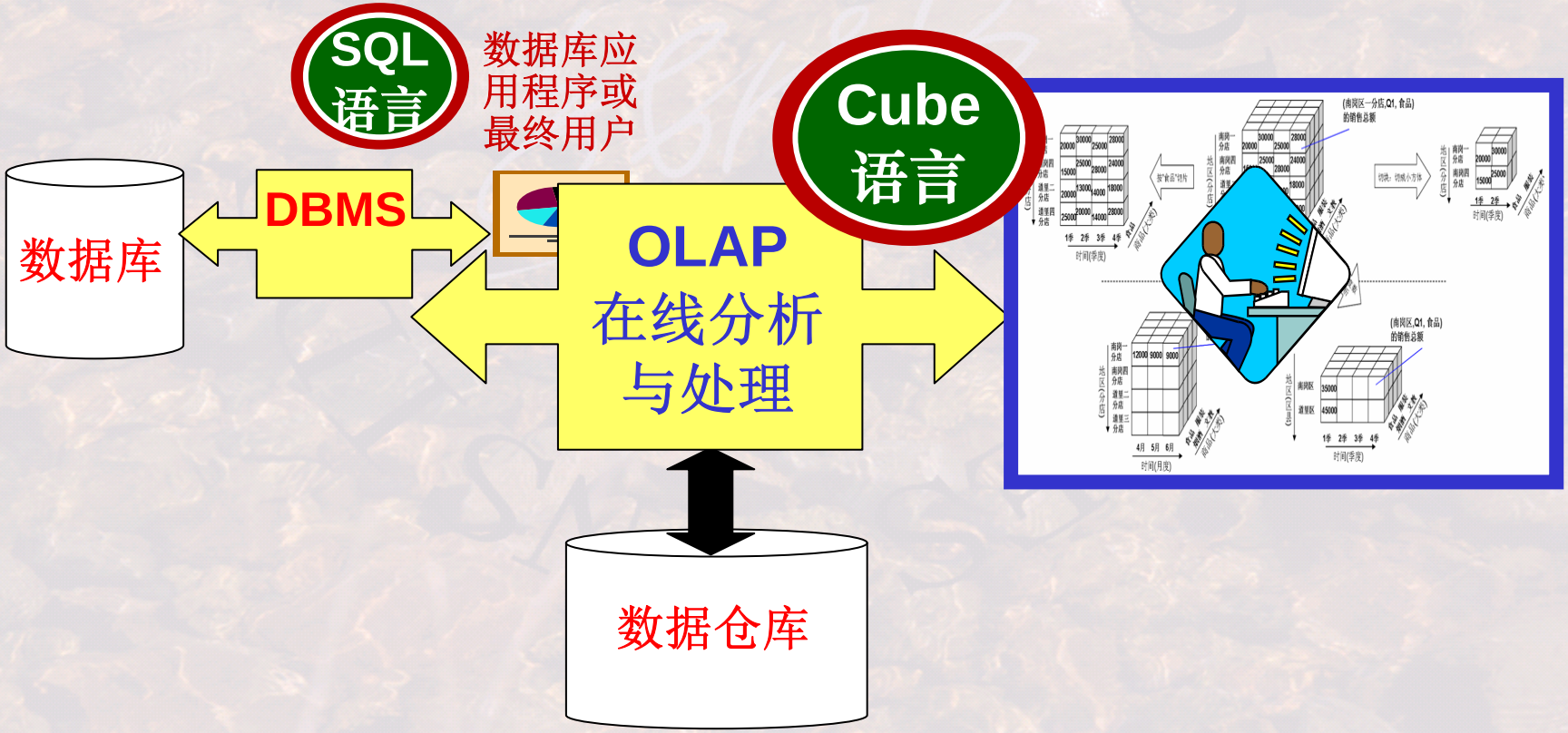
示例

数据抽象与设计II---设计之形式-构造-自动化?
(2)怎样进行“设计”?

战德臣 教授

基于广义的计算机语言的设计和实现软件/硬件/网络系统

“形式➔构造➔自动化”

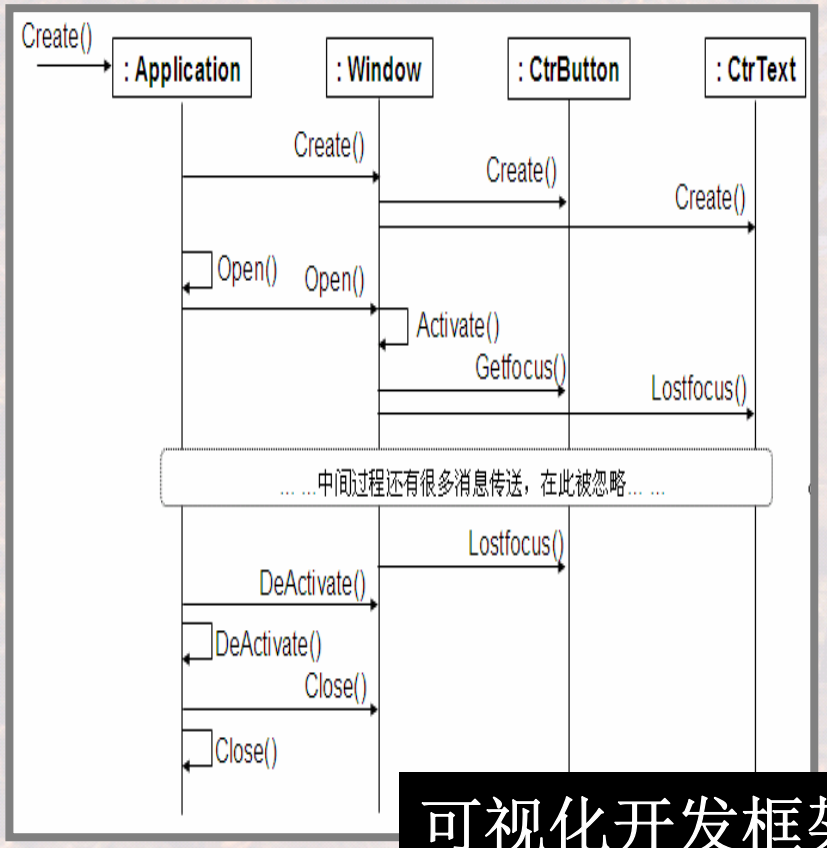


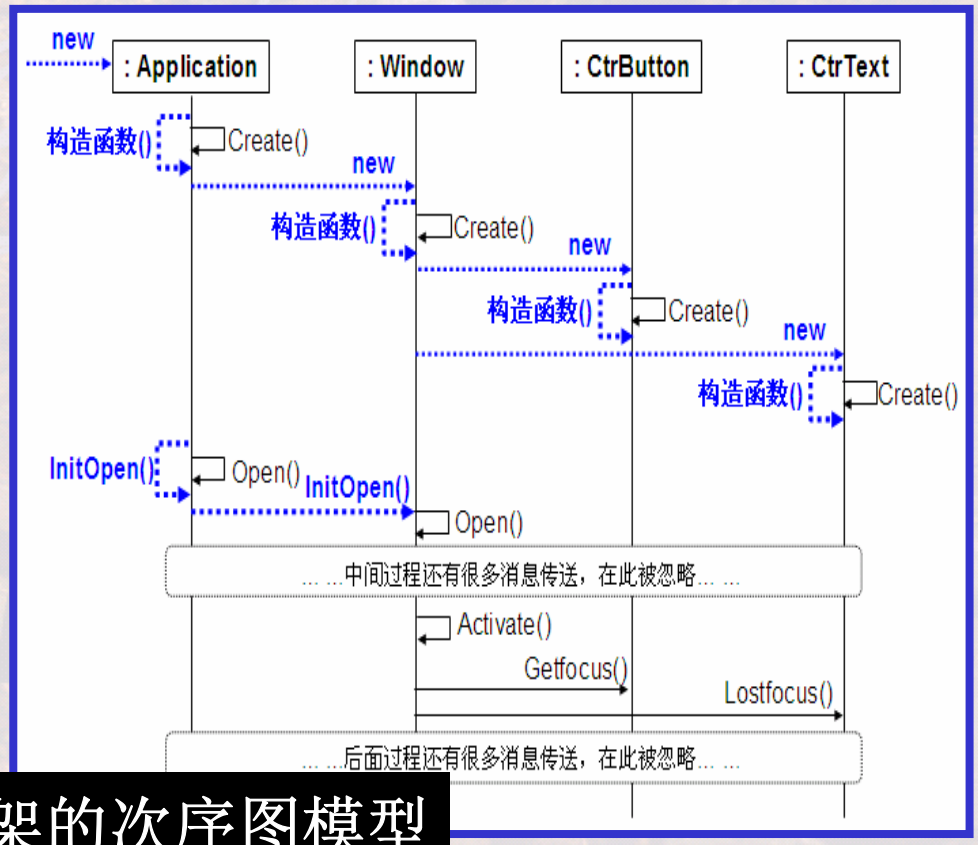
战德臣 教授

设计形态的其他示意性示例

◆软件、硬件产品的设计与实现是重要的“设计”形态的内容。

◆设计形态的结果可以是：各种模型。





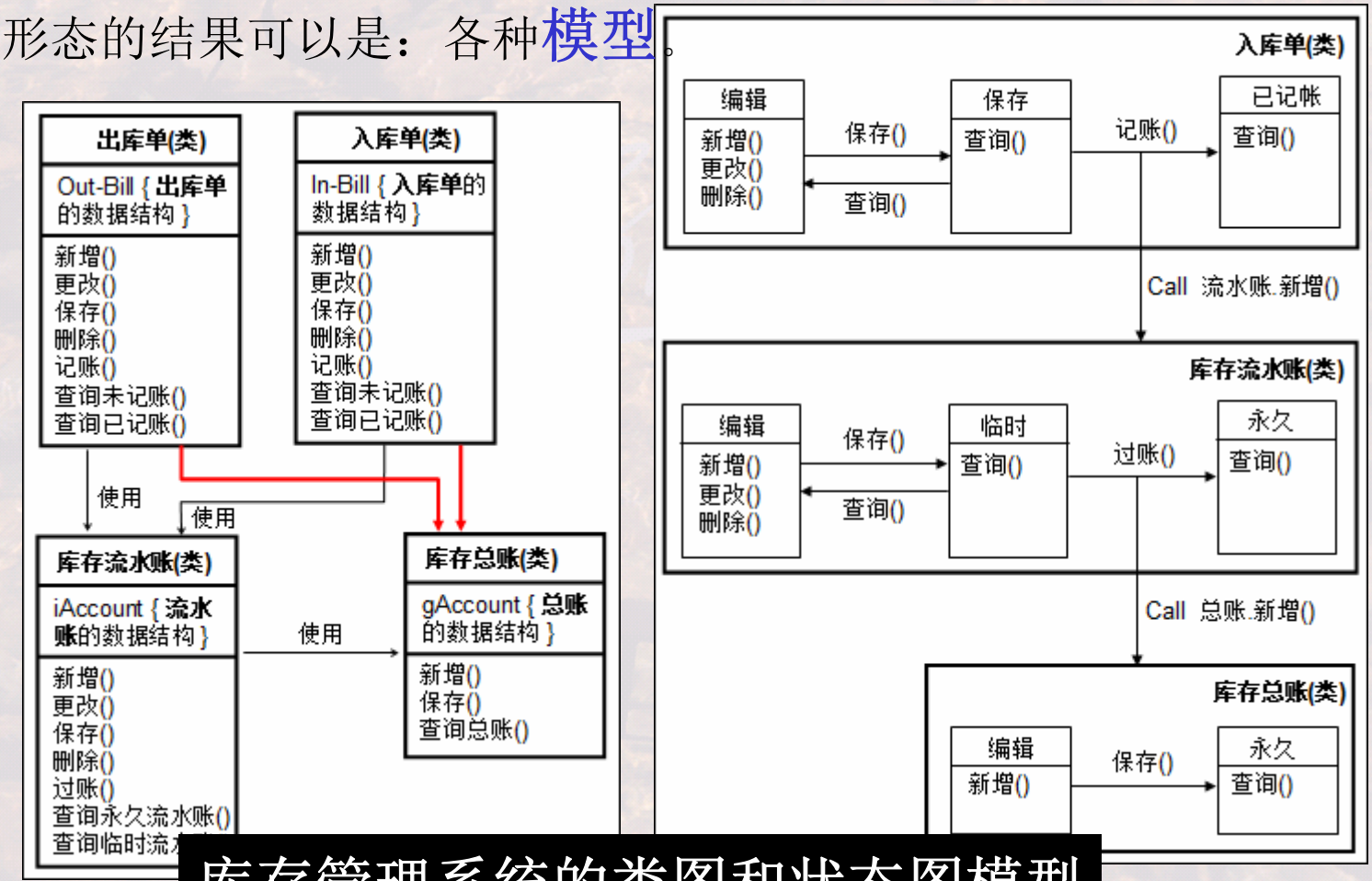
可视化开发框架的次序图模型

战德臣 教授

设计形态的其他示意性示例

◆软件、硬件产品的设计与实现是重要的“设计”形态的内容。

◆设计形态的结果可以是：各种模型。



库存管理系统的类图和状态图模型

战德臣 教授

设计形态的其他示意性示例

- 设计形态的结果也可以是程序代码和软件产品
- 设计结果的正确性是怎样的呢?
- 怎样度量设计形态的成果呢?
- 怎样判断设计成果的有效性呢?

软件产品(界面)示意

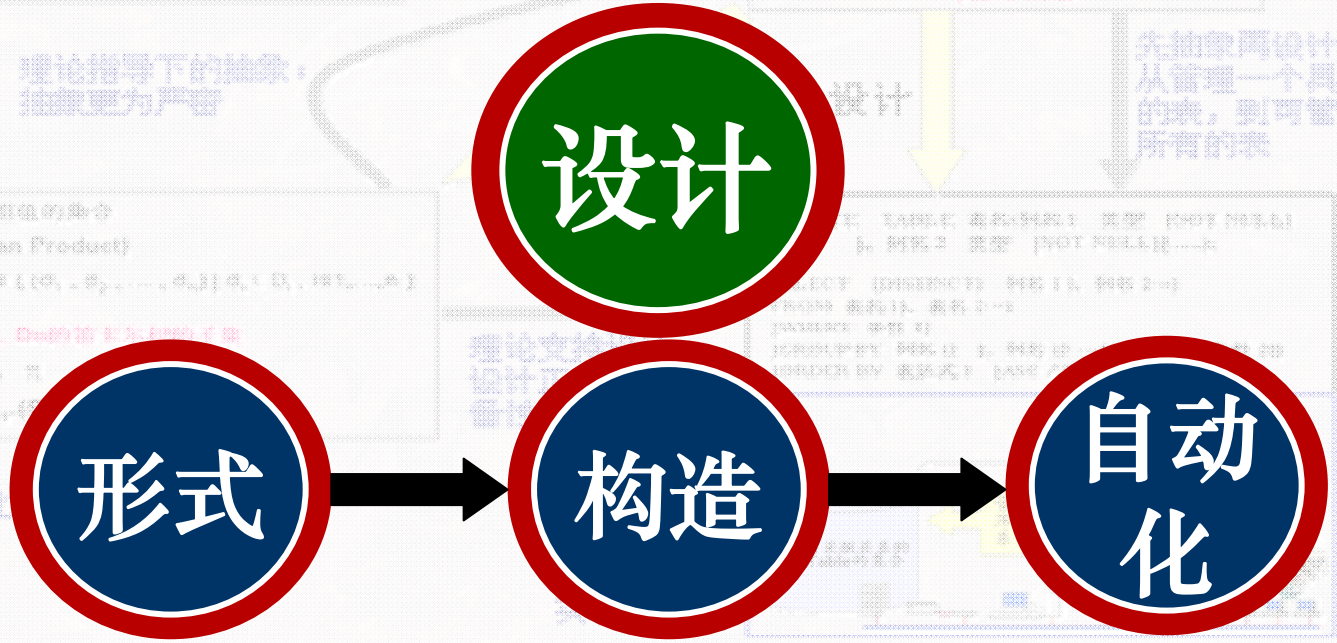
程序代码示意

```
public class Inventory
{
    private String Item_ID;
    private Integer Inv;
        public Inventory(String item, Integer iniInv)
    {
     this.Item_ID=item;
     this.Inv=iniInv;
    }
    public void setInv(Integer iniInv)
    {
     this.Inv=iniInv;
    }
    public Integer getInv()
    {
     return this.Inv;
    }
    public void Input(Integer inputQty)
    {
     this.Inv=this.Inv+inputQty;
    }
    public boolean Output(Integer outputQty)
    {
      If(this.Inv<outputQty)
       return false; //数量不够，出库失败
      else
       this.Inv=this.Inv-outputQty; //有足够的数量，出库
    }
}
```

战德臣 教授

最关键的

- 构造系统(表示-存储-执行)，实现："寻找相同的形式，处理可变的内容"
- 形式➔构造➔自动化----计算学科"设计"的过程



数据抽象与设计III
---理论之定义-定理-证明

战德臣

哈尔滨工业大学　教授.博士生导师

教育部大学计算机课程教学指导委员会委员

Research Center on Intelligent Computing for Enterprises & Services, Harbin Institute of Technology

战德臣 教授

抽象-理论-设计概述

◆抽象、理论和设计是计算学科的三种形态或三种过程

◆设计--是指构造计算系统来改造世界的手段，是工程的主要内容，只有设计才能造福于人类(设计的价值)

◆理论--是发现世界规律的手段，理论，如果不能指导设计，则是反映不出其价值的；设计，如果没有理论指导，则设计的严密性、可靠性、正确性是没有保证的(理论的价值)

◆抽象--是感性认识世界的手段。理论和设计的前提都需要抽象，没有抽象二者都是没有办法达成目标的(抽象的价值)

ICES 战德臣 教授

科学方法论中关于“理论”的描述

◆理论：是经过实践检验的系统化了的科学知识体系，它是由科学概念、科学原理以及对这些概念原理的理论论证所组成的体系。科学认识由感性阶段上升为理性阶段就形成了科学理论。

◆理论：源于数学，是从抽象到抽象的升华，它们已经完全脱离现实事物，不受现实事物的限制，具有精确的优美的特征，因而更能把握事物的本质。

◆理论研究的基本方法是：表述研究对象的特征(定义和公理)➔假设对象之间的基本性质和对象之间可能存在的关系(定理)➔确定这些关系是否为真(证明)➔形成最终的结论。

怎样研究理论?

战德臣 教授

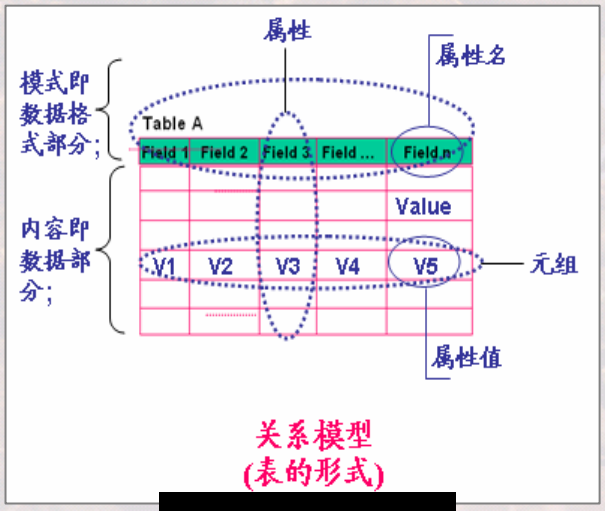
理论：是对规律进行严密化描述及论证的过程

- 通常由定义、公理、性质、定理和证明等内容构成。
- 定义是对概念的严密化描述。
- 公理是可由概念及其固有性质证明其正确性的结论性的描述。
- 定理是可由定义、公理和其他定理证明其正确性的结论性的描述。
- 证明是公理、定理正确性的论证过程。

战德臣 教授

理论：前提是抽象，基础是形式化和数学化。

能够形式化数学化的概念和规律是严密的概念和规律。



抽象形态

理论：数学化逻辑严密化各种概念；

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

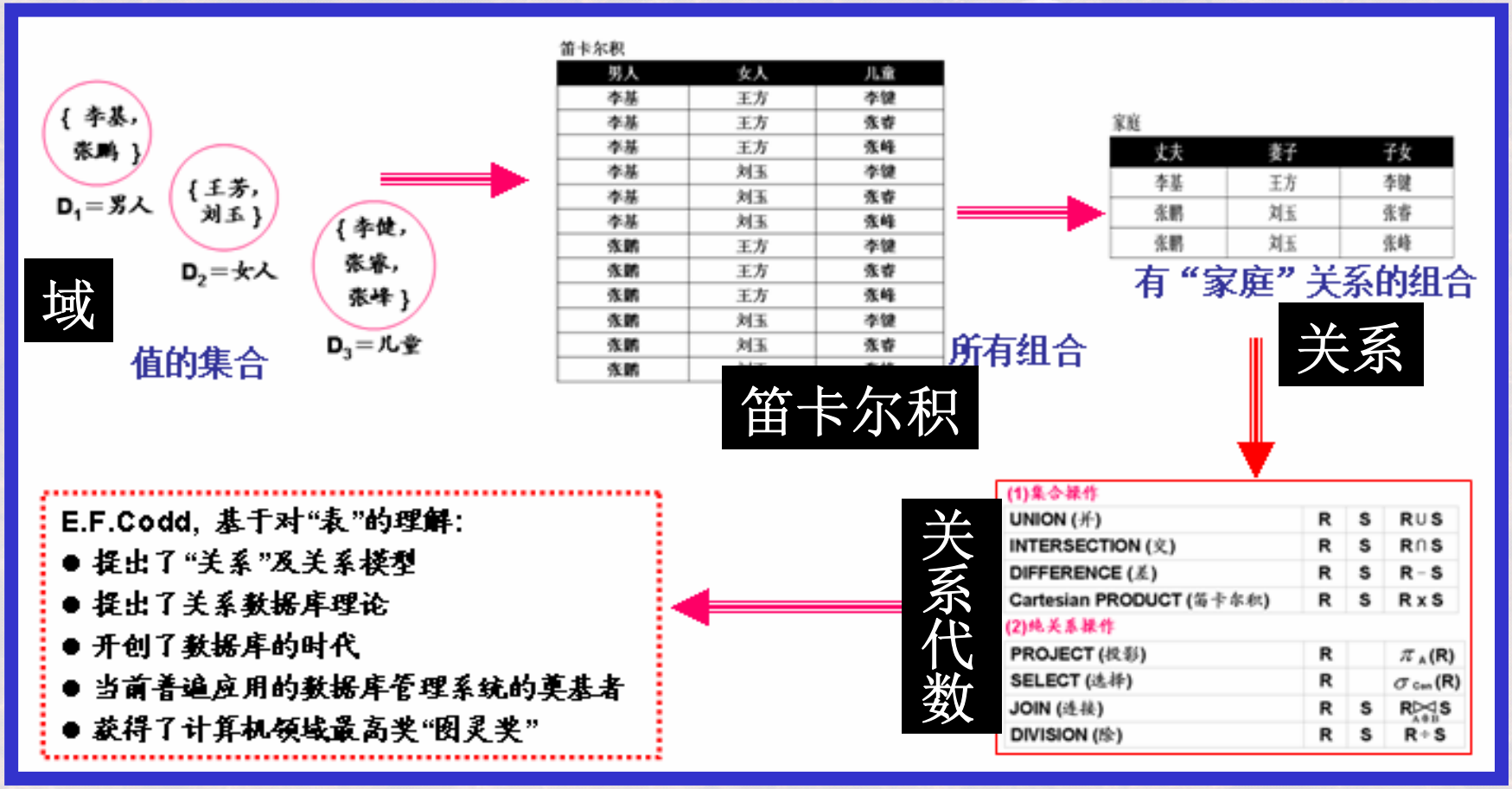
$\pi_{S\#,Sname}(\sigma_{C\#="001"}(Student \bowtie SC))$

理论形态

战德臣 教授

理论：前提是抽象，基础是形式化和数学化。

能够形式化数学化的概念和规律是严密的概念和规律。



数据抽象与设计III---理论之定义-公理-定理-证明
(3)理论是怎样指导实践的(示例)??

战德臣 教授

学生表

学号	姓名	年龄	住址
981101	李四	22	3010
981103	李三	21	3011
981105	李六	22	3011

课程表

课程号	课程名	教师	学时
C1	计算机	教师 1	52
C2	物理	教师 2	36
C3	高数	教师 5	40

笛卡尔积(×)的实现算法

```
For   i=1 to R的记录数
      读取R的i-th记录;
      For j=1 to S的记录数
            读取S的j-th记录;
            将R的第i-th记录和S
             的第j-th记录拼接成一条记录;
            存入结果关系;
      Next j
Next i
```

按照数学确定的算法

```
For   i=1 to R的存储块数
    读取R的i-th个磁盘块进入内存;
    For j=1 to S的存储块数
        读取S的j-th个磁盘块进入内存;
        For   p=1 to   磁盘块存储R的记录数
            内存中读取R的第p-th记录;
            For q=1 to   磁盘块存储S的记录数
                内存中读取S的第q-th记录;
                内存中将R的第i-th记录和S的第j-th记录拼接成一条记录;
                存入结果关系所对应的内存块中;如果内存块满,则将其写入磁盘;
            Next q
        Next p
    Next j
Next i
```

考虑计算环境确定的算法

基本动作

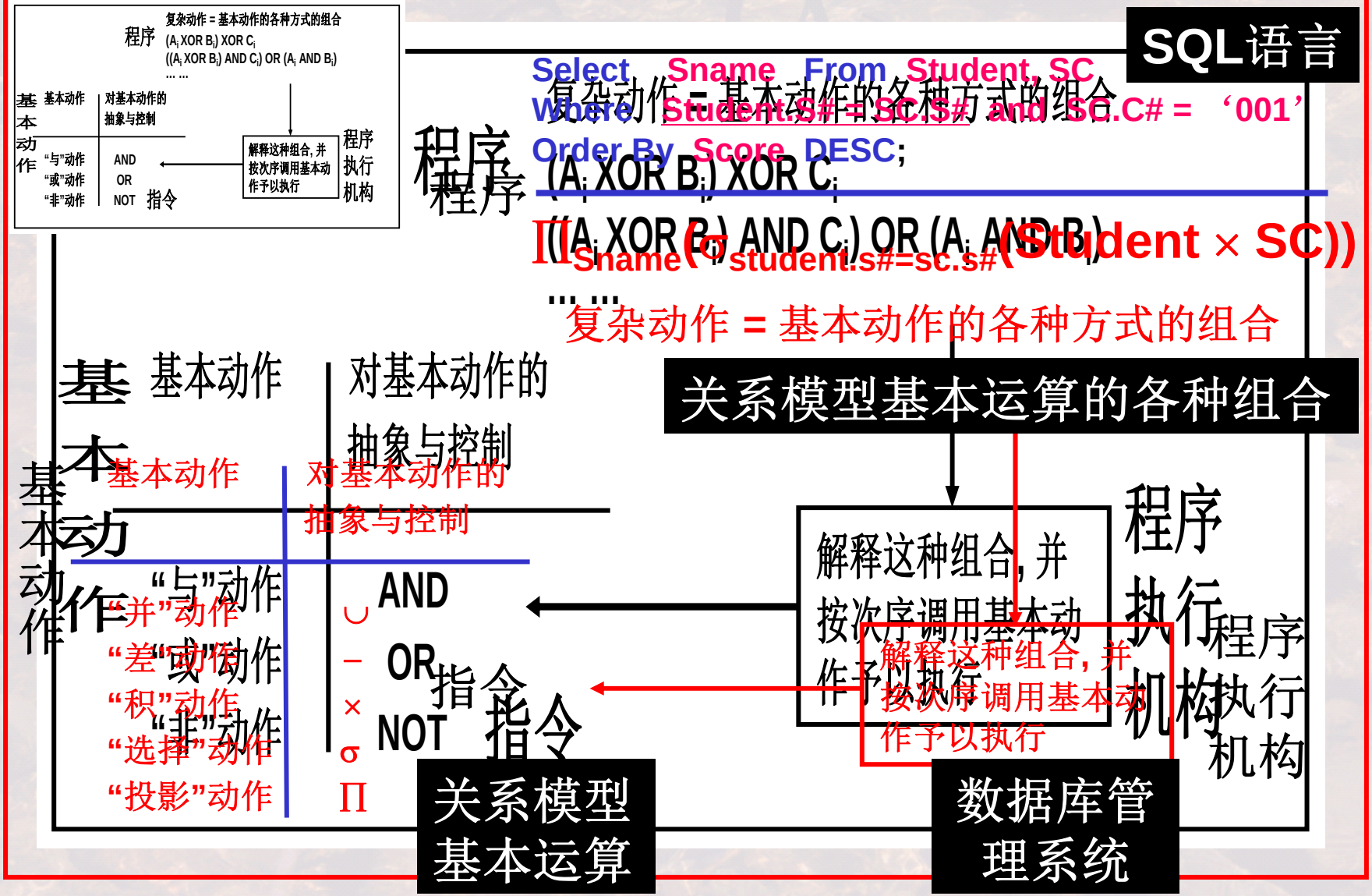
基本动作	对基本动作的抽象与控制
“并”动作	∪
“差”动作	−
“积”动作	×
“选择”动作	σ
“投影”动作	Π

指令

关系模型基本运算

数据抽象与设计III---理论之定义-公理-定理-证明
(3)理论是怎样指导实践的(示例)??

战德臣 教授



数据抽象与设计III---理论之定义-公理-定理-证明
(3)理论是怎样指导实践的(示例)??

战德臣 教授

SQL语言

Select Sname From Student, SC
Where Student.S# = SC.S# and SC.C# = '001'

$$\pi_{姓名}(\sigma_{(选课.成绩>80\ and\ 课程.课名='程序设计')}(\sigma_{(学生.学号=选课.学号\ and\ 课程.课号=选课.课号}(学生 \times 选课 \times 课程)));$$

$$\pi_{姓名}(\sigma_{(学生.学号=选课.学号\ and\ 课程.课号=选课.课号\ and\ 选课.成绩>80)}(学生 \times 选课 \times \sigma_{(课名='程序设计')}(课程)));$$

$$\pi_{姓名}(\sigma_{(学生.学号=选课.学号)}(学生 \times \sigma_{(课程.课号=选课.课号\ and\ 选课.成绩>80)}(选课 \times \sigma_{(课名='程序设计')}(课程))));$$

这三个语句产生的结果是相同的, 但执行效率却大不相同!

问：能否选择出最快的?怎样选择?怎样证明其相互的等价性?

?

基本动作

“积”动作 ×

“选择”动作 σ

“投影”动作 Π

关系模型基本运算

数据库管理系统

程序执行机构

战德臣 教授

理论形态的价值

◆当提出一新算法时，新算法的正确性有效性需要证明，也就需要理论；

◆针对问题，当提出一种解决方案时，则方案的正确性有效性需要证明，也就需要理论；

◆当提出一套计算机语言时，该语言的完备性、安全性、正确性，则需要证明，也就需要理论；

◆当提出一套求解问题的规则时，该套规则的完备性、安全性、正确性，则需要证明，也就需要理论；

◆……

◆这些理论就需要定义、公理、定理及其证明。

数据抽象与设计III---理论之定义-公理-定理-证明
(5)抽象-理论与设计?

战德臣 教授

基本目标: 理解抽象、理论与设计