NO
DATE

第二章 关系数据库系统

·关系模型

一、数据结构——关系.

集合代数，数据的逻辑结构是二维表

关系数据库由表构成，每个表有唯一的名字，称为关系

-元组：表中的行，关系是元组集合，次序不重要

-属性：表中的列，n元关系有n个属性.

-域 Domain 每个允许取值的关系的属性都存在一个集合，称为该属性域

所有可能属性值的集合.

-原子域 - 域中元素被看作是不可再分的单元

-复合域 - 原子域的组合.

-空值 Null：值未知或不存在

关系的数学定义：设$A_1, A_2, \cdots A_n$是值域为$D_1, D_2 \cdots D_n$的n个属性.

具有属性$A_1, \cdots A_n$的关系R是一个元组集合，其中，每个元组是一个映射集合

$\{A_1 \to D_1, \cdots A_n \to D_n\}$.

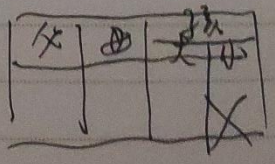
关系的性质：-列是同质的 Homogeneous.

-不同的列可来自同一个域，其中每一列称为一个属性，不同的属性要给予不同属性名

-行列的顺序无所谓.

-任意两个元组不能完全相同；由集合性质决定，但 Oracle 等允许存在

-分量必须取原子值，若即必不可空



关系模式.

具有形式 R(U, D, DOM, I, F)，其中：

R是关系名，U是R的属性集合$\{A_1, \cdots A_n\}$，D是U中属性的域集合$\{D_1 \cdots D_n\}$

DOM是U到D的映射，I是完整性约束集合，F是属性间的函数依赖关系

NO
DATE

·关系模式简单地记作R(U)，U={A_1，…A_n}的关系模式R记作 $R(A_1, A_2, \cdots A_n)$.
例 Student(姓名，学号，年级，专业，系)
·关系：实例 -关系模式在给定时刻的一个快照

关系、关系模式与关系实例：
·关系是一个数据集合 | ·关系模式描述关系的数据结构和语义约束，非集合
·关系模式是相对稳定的 | 关系实例是随时间而变化的
·关系实例是某一时刻现实世界状态的真实反映

关系数据库模式 - 一组关系模式的集合 DB={R_1，…R_n}. R_i是第i个关系模式
关系数据库实例

2. 完整性约束
·超码 - 是一个或多个属性的集合，这些属性的组合可以在一个关系中唯一地标识一个元组
如果K是一个超码，那么K的任意超集也是超码，→ 超码中可能包含元关系模式的属性
候选码 - K为关系R(U)的候选码，如果K是R(U)是个超码，K是任意真子集都不能成为R(U)的超码，最小的超码
一个关系模式可能具有多个候选码
·主码 - 被数据库设计者选中的，主要用来在一个关系中区分不同元组的候选码
应选值不变/少变的属性，一个关系模式只有一个主码
主属性 -主码中属性
外码 - 一个关系模式R，可能在它的属性中包括另一个关系模式上的主码，这个属性集合在R上称作参照上的外码
不一定要与相应的主码同名，一般情况下，外码与相应主码采用相

(4). Select S#, SNAME From S Order by
Select S#, SNAME From

NO
DATE

关系数据模型的完整性约束

- 参照完整性约束 — 定义主/外码之间规则，要求在参照关系中任意元组在外特定属性上的取值必等于被参照关系中某个元组在特定属性上的取值
- 实体完整性约束
- 如果A是关系模式R(U)的主属性，则A不能取NULL

三、关系运算 < 关系代数 ; 关系演算 < 元组演算 ; 域演算

3.1 关系代数 — 包括一个运算集合，这些以一个或两个关系为输入，产生一个新的关系作为结果。

六种基本操作符

- 选择 Select. $\sigma_p(R)=\{t \mid t\in R \text{ and } p(t)\}$

  p称为选择谓词 — 是由∧, ∨, ¬ 连接的若干元子表达式构成的公式

  └ 元子表达式的形式为: <属性> θ <属性>

  <属性> θ <常数>

  $\theta\in\{=,\neq,>,\geq,<,\leq\}$ <常数> θ <属性>

$\sigma_{A=5\wedge B>7}(R)$

- 投影 Project $\Pi_{A_1,A_2,\ldots,A_k}(R)$

  $A_1\ldots A_k$是属性名，R为关系名，返回所选择的k个列。

  删去重复元组

- 并 Union — $R\cup S=\{t \mid t\in R \text{ or } t\in S\}$

重复元组 注意:必须保证做并运算的关系是相容的，即 R和S的属性个数都相同，对于所有i，R的第i个属性的域与S的第i个属性域相同

差 Difference $R-S=\{t \mid t\in R \text{ and } t\notin S\}$

差运算的关系也要求是相容的

NO
DATE

· 笛卡尔积 Cartesian product. $R\times S=\{tq \mid t\in R \text{ and } q\in S\}$.
R中有n_1个元组，S中有n_2个元组，则$R\times S$中有$n_1\times n_2$个元组。
注意：关系R和S的属性是不相交的，若相交，则需进行重命名，可将关系名字附加到属性上。如 R.A
不去除重复属性

· 改名 Rename $\rho_{S(A_1,A_2,\ldots A_n)}(R)$.
表示将关系R重新命名为S，且S的各个属性按从左到右属性命名$A_1\ldots A_n$.
S与R具有完全相同元组。如果只修改关系，不修改属性名. $\rho_S(R)$

附加的关系代数运算 — 简化常用的查询
· 交 Intersection $R\cap S=\{t \mid t\in R \text{ and } t\in S\}$.
$R\cap S=R-(R-S)$
交运算要求关系相容

· 自然连接 Natural join $R\bowtie S$ 去除重复属性
— 要求进行笛卡尔积的两个关系在所有相同属性上的值一致
满足：属性集合 $=U_R\cup U_S$，其中U_R和U_S分别是关系R和S的属性集合
$R\bowtie S=\pi_{U_R\cup U_S}(\sigma_{R.A_1=S.A_1\wedge\cdots\wedge R.A_n=S.A_n}(R\times S))$.

· θ连接 θjoin — 设R是n元关系$R(A_1,\ldots,A_n)$，S是m元关系$S(B_1,B_2,\cdots B_m)$.
$R\underset{C}{\bowtie}S=\sigma_C(R\times S)$
· 其中C是由$\wedge,\vee,\neg$连接若干原子公式构成的.
元子公式为 $R.A_i\,\theta\,S.B_j$. A_i和B_j值域具有相同的数据类型
$\theta\in\{=,\neq,>,<,\leq,\geq\}$.

Eg:

A	B	C
1	2	3
4	5	6
7	8	9

关系R

D	E
3	1
6	2

关系S

A	B	C	D	E
1	2	3	3	1
1	2	3	6	2
4	5	6	6	2

$R\underset{B<D}{\bowtie}S$

(3). Select S# SNAME From S Order by

NO
DATE

·赋值 assignment ←，将←右侧的表达式结果赋给左侧的关系变量

Eg. temp1 ← R×S;
temp2 ← $\sigma_{R.A_1=S.A_1 \wedge R.A_2=S.A_2 \wedge \cdots \wedge R.A_k=S.A_k}$(temp1);
result = $\Pi_{R \cup S}$(temp2).

对关系代数查询而言，赋值只是将关系赋给一个临时关系变量，因此对永久关系的赋值形成了数据修改.

-赋值运算不能增强关系运算的表达能力

·外连接 -自然连接运算的扩展，用以处理缺失值

-左外连接 ⟕ ·取出左侧关系中所有与右侧关系的任一元组都不匹配的元组，用空值填充所有来自右侧关系的属性.
·再把所产生的元组加到自然连接中.

-右外连接 ⟖ ·同上换成"右"

-全外连接 ⟗ ·左外连接与右外连接的并

$$R ⟕ S = (R \bowtie S) \cup [(R - \Pi_R(R \bowtie S)) \times \{(null, \cdots, null)\}]$$

·除操作 division · 设R/S两关系.

· $X = U(R) = (A_1, \cdots A_m, B_1, \cdots, B_n)$ · $Y = U(S) = (B_1, \cdots, B_n)$

$Z = X - Y = (A_1, \cdots A_m)$

$R \div S$ 的属性为 $(A_1, \cdots, A_m)$.

$R \div S = \{t \mid t \in \Pi_Z(R) \wedge \forall u \in S(tu \in R)\}$.

→不是存在!! 是任意S中的元组u tu∈R!!

↘ tu是指元组t与元组u相连接构成新元组

$R \div S = \Pi_Z(R) - \Pi_Z((\Pi_Z(R) \times S) - R)$.

R:

A	B
a	1
a	2
a	3
b	1
b	2
c	1
d	1
d	4

S:

B
1
2

R÷S

A
a
b

NO
DATE

3.2 元组演算：非过程化的查询，只需描述所需信息，不给出获得该信息过程

表达式 {t | P(t)} —是所有使得公式P为真的元组t的集合

—t表示元组变量，t[A]表示t在A属性上的取值

P由原子公式组成 形式
- S∈R，S是元组变量，R是关系
- S[x] θ u[y]，S,u是元组变量，x,y是定义所基于的关系模式属性，θ是比较运算符
- S[x] θ c，c为x所属域的常量

3.3 域演算：与元组演算联系紧密，使用从属性域中取值的域变量，而不是整个元组的值

形式化定义—{<x₁,x₂,…,xₙ> | P(x₁,x₂,…,xₙ)} 其中 x₁…xₙ是域变量

—P是由原子公式构成的公式，查询结果是所有包含<x₁,…,xₙ>的元组，并且<x₁,…,xₙ>使得公式P(x₁,…,xₙ)为真

3.4 安全性—如果一个关系运算系统不产生无限关系和无穷验证，则这个运算是安全

关系代数—安全

关系演算—元组—不安全 {t | t∉R(t)}是无限集
　　　　　　　　　　—若t定义域无穷，判断{t | ∀t(F(t))}为真要无穷次
　　　　　—域—不安全 {<t₁,t₂> | <t₁,t₂>∈R ∨ t₂>9}无限集

引入元组关系公式域的概念 → 若有

等价
基本关系代数（不含扩展）= 限制在安全的元组 = 限制在安全…的域 表达式

(5). Select S#, SNAME From S Order by