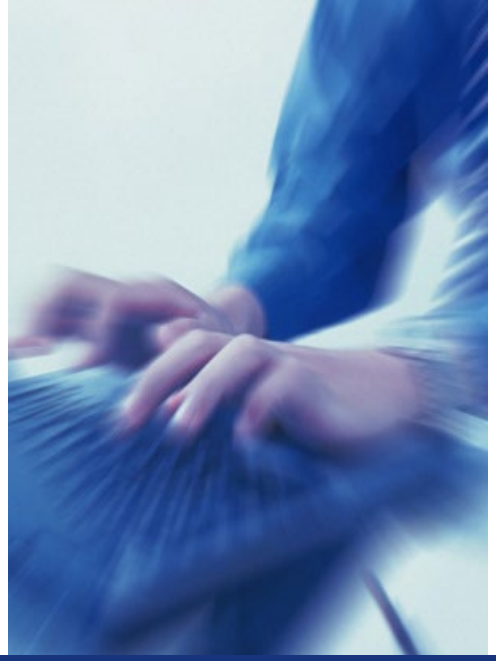
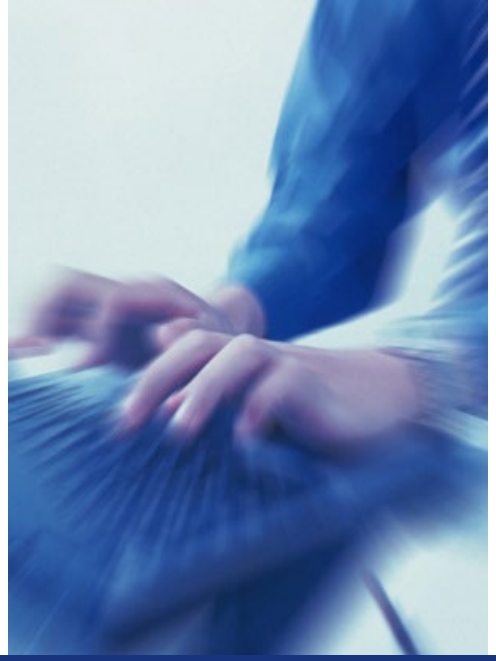


Logo

Discrete Mathematics

Dr. Han Huang

South China University of Technology

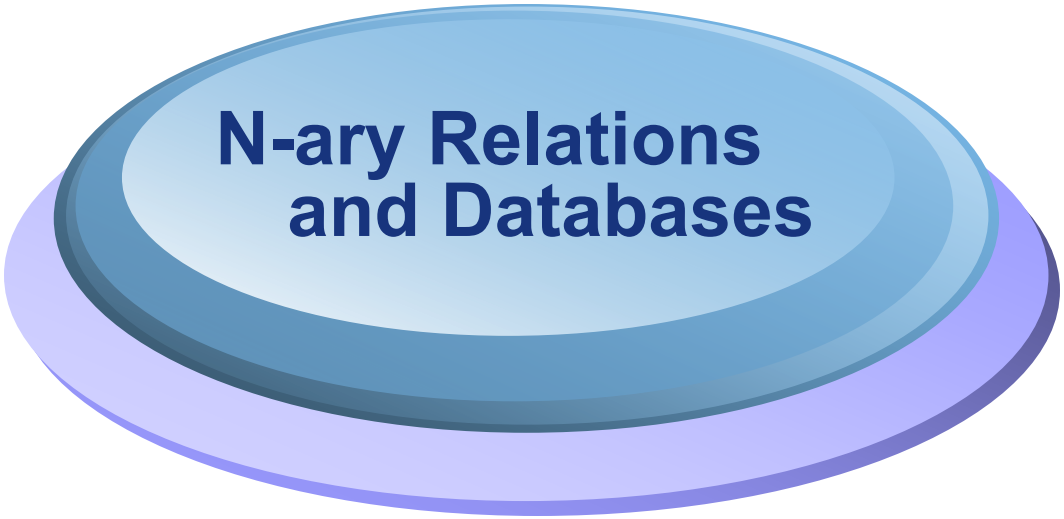


Logo

n-ary Relations and Their Applications

Section 3.2

Contents


N-ary Relations
and Databases

n-ary Relations

- An *n*-ary relation *R* on sets $A_1,\ldots,A_n$, is a subset

$$R \subseteq A_1 \times \ldots \times A_n.$$

- This is a straightforward generalisation of a binary relation. For example:
- 3-ary relations:
  - a is between b and c;
  - a gave b to c

n-ary Relations

- An *n*-ary relation *R* on sets $A_1,\ldots,A_n$, is a subset

$$R \subseteq A_1 \times \ldots \times A_n.$$

- The sets A_i are called the *domains* of *R*.
- The *degree* of *R* is *n*.
- *R* is *functional in the domain* A_i if it contains at most one *n*-tuple $(\ldots, a_i, \ldots)$ for any value a_i within domain A_i.

n-ary Relations

- ***R* is *functional in the domain* A_i if it contains at most one *n*-tuple (…, a_i,…) for any value a_i within domain A_i.**
- **Generalisation: being functional in a combination of two or more domains.**

Relational Databases

- A *relational database* is essentially just a set of relations.
- A domain A_i is a *primary key* for the database if the relation R is functional in A_i.
- A *composite key* for the database is a set of domains $\{A_i, A_j, \ldots\}$ such that R contains at most 1 n-tuple $(\ldots,a_i,\ldots,a_j,\ldots)$ for each composite value

  $(a_i, a_j,\ldots)\in A_i\times A_j\times\ldots$

Example of Relational Data Model

- (Student Name, ID Number, Major, GPA)
- (小白, 98601,计算机科学, 3.76)
- (小黄, 98602, 物理, 3.34)
- (小黑, 98603,计算机科学, 3.59)
- (小红, 98604, 数学, 3.99)
- (小绿, 98605, 体育, 2.40)
- Primary key is ID Number.


Operations on N-ary Relations

(Selection Operators)

- Let A be any n-ary domain $A=A_1 \times \ldots \times A_n$, and let $C:A \rightarrow \{T,F\}$ be any *condition* (predicate) on elements (n-tuples) of A.
- Then, the *selection operator* s_C is the operator that maps any n-ary relation R on A to the n-ary relation consisting of all n-tuples from R that satisfy C. That is,

$$s_C(R) = \{a \in R \mid C(a) = T\}$$

(Selection Operator Example)

- Suppose we have a domain
  A = StudentName × Standing × SocSecNos
- Suppose we define a condition Upperlevel on A: $UpperLevel(name, standing, ssn) :\equiv [(standing = junior) \vee (standing = senior)]$
- Then, $s_{UpperLevel}$ is the selection operator that takes any relation R on A (database of students) and produces a relation consisting of *just* the upper-level students (juniors and seniors).

(Projection Operators)

- Let $A = A_1 \times \ldots \times A_n$ be any *n*-ary domain, and let $\{i_k\}=(i_1,\ldots,i_m)$ be a sequence of indices all falling in the range 1 to *n*,
  - That is, $1 \le i_k \le n$ for all $1 \le k \le m$.
- Then the *projection operator* on *n*-tuples is defined by:

$$P_{\{i_k\}} : A \to A_{i_1} \times \ldots \times A_{i_m}$$

$$P_{\{i_k\}}(a_1,\ldots,a_n) = (a_{i_1},\ldots,a_{i_m})$$

(Projection Example)

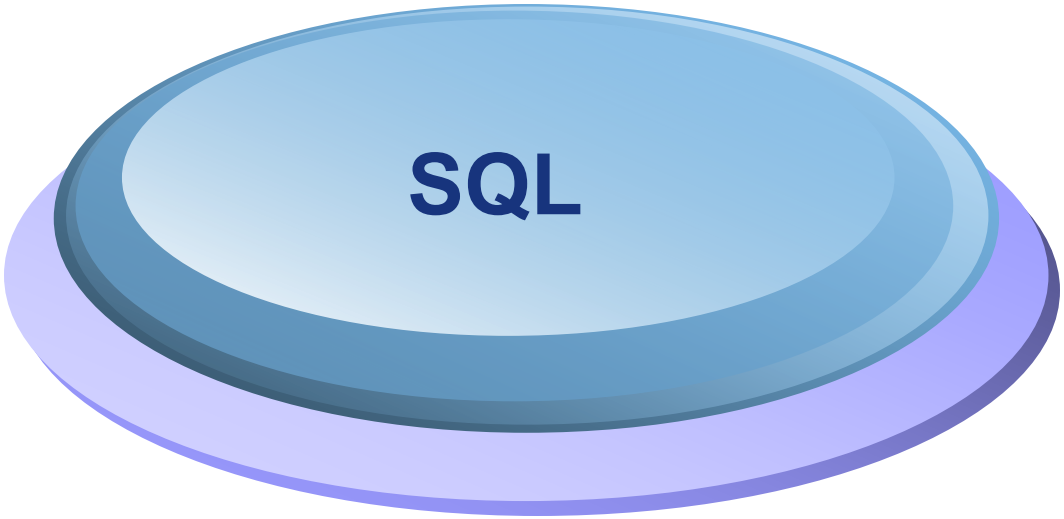
- Suppose we have a ternary (3-ary) domain ***Cars=Model×Year×Color.*** (note ***n*=3**).
- Consider the index sequence $\{i_k\}$= 1,3. (***m*=2**)
- Then the projection $P_{\{i_k\}}$maps each tuple (a_1,a_2,a_3) = (*model*,*year*,*color*) to its image:
- $$(a_{i_1}, a_{i_2}) = (a_1, a_3) = (model, color)$$
- This operator can be usefully applied to a whole relation ***R⊆Cars*** (a database of cars) to obtain a list of the model/color combinations available.

(Join Operator)

- Puts two relations together to form a combined relation which is their composition:
- Iff the tuple (*A*,*B*) appears in R_1, and the tuple (*B*,*C*) appears in R_2, then the tuple (*A*,*B*,*C*) appears in the join $J(R_1,R_2)$.
  - *A*, *B*, and *C* can also be sequences of elements.

(Join Example)

- Suppose R_1 is a teaching assignment table, relating *Lecturers* to *Courses*.
- Suppose R_2 is a room assignment table relating *Courses* to *Rooms*,*Times*.
- Then $J(R_1,R_2)$ is like your class schedule, listing (*lecturer*,*course*,*room*,*time*).
- (For precise definition, see Rosen, p.486)


SQL

Example of SQL

- SELECT Departure_time FROM Flights WHERE destination = ‘广州’

- SELECT professor, time FROM teaching_assignments, class_schedule WHERE department = ‘离散数学’

Applications

- 设有如下表所示的三个关系S、C和SC。用SQL语句表示:

S

学号	姓名	年龄	性别	籍贯
98601	王晓燕	20	女	北京
98602	李　波	23	男	上海
98603	陈志坚	21	男	长沙
98604	张　兵	20	男	上海
98605	张　兵	22	女	武汉

C

课程号	课程名	教师姓名	办公室
C601	高等数学	周振兴	416
C602	数据结构	刘建平	415
C603	操作系统	刘建平	415
C604	编译原理	王志伟	415

SC

学号	课程号	成绩
98601	C601	90
98601	C602	90
98602	C601	90
98603	C601	75
98603	C602	70
98604	C604	85
98605	C603	80

(1)检索籍贯为福建的学生的姓名、学号和选修的课程号。

(2)检索选修了全部课程的学生姓名和年龄。

Applications

- 设有如下表所示的三个关系S、C和SC。用SQL语句表示:

S

学号	姓名	年龄	性别	籍贯
98601	王晓燕	20	女	北京
98602	李　波	23	男	上海
98603	陈志坚	21	男	长沙
98604	张　兵	20	男	上海
98605	张　兵	22	女	武汉

C

课程号	课程名	教师姓名	办公室
C601	高等数学	周振兴	416
C602	数据结构	刘建平	415
C603	操作系统	刘建平	415
C604	编译原理	王志伟	415

SC

学号	课程号	成绩
98601	C601	90
98601	C602	90
98602	C601	90
98603	C601	75
98603	C602	70
98604	C604	85
98605	C603	80

(1)检索籍贯为福建的学生的姓名、学号和选修的课程号。

select 姓名,S.学号,课程号 from SC,S where 籍贯='福建' and S.学号=SC.学号

(2)检索选修了全部课程的学生姓名和年龄。

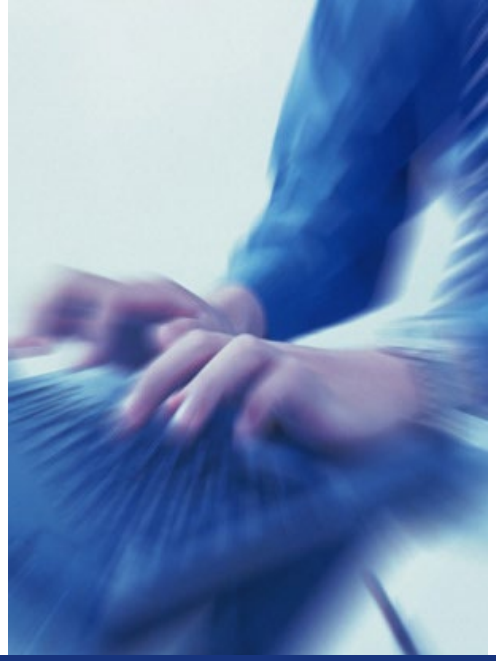
select 姓名,年龄 from S,C,SC where C.课程号=SC.课程号 and S.学号=SC.学号


Exercises

Exercises

1. There are $2^{(m^n)}$ n-tuple relations on the set S×S…×S when |S|=m.

Logo



End of Section 3.2