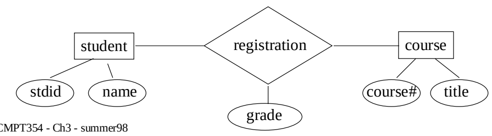
CMPT 354
Database Systems and Structures

Osmar R. Zaïane

Summer 1998

Summary of Chapter 2

- Basic Concepts of Entity-Relationship Model
  - entity set
  - relationalship set
- Design Issues and Mapping Constraints
- Notations - Entity-Relationship Diagram
- Extended Entity-Relationship features
- Reduction of E-R Schema to tables



Chapter 3 Objectives

Relational Model

Understand the structure of Relational Databases.
Learn to use formal languages, like Relational Algebra and Relational Calculus.

Contents

- Structure of Relational Databases
- Relational Algebra
- Tuple Relational Calculus
- Domain Relational Calculus
- Extended Relational-Algebra-Operations
- Modification of the Database
- Views

Relational Database Model

Account

bname	account#	cname	balance
Downtown	101	Johnson	500
Lougheed	215	Smith	700
SFU	102	Hayen	400
SFU	301	Adams	1300

Customer

cname	street	ccity
Johnson	Pender	Vancouver
Smith	North	Burnaby
Hayes	Austin	Burnaby
Adams	N3 Road	Richmond
Jones	Oak	Vancouver

- Basic concepts
  - A relational database is a collection of ***tables***
  - Each table has a unique name
  - Column => attribute
  - For each attribute there is a permitted set of values called ***Domain***

    D1={Downtown, Lougheed, SFU,...} D2={101, 215, 102, 301,...}
    D3={Johnson, Smith, Hayen, Adams,...} D4={0,1,2,3,4,5,6,...}
  - Row => relationship among a set of values
  - Table = ***Relation***

Account

bname	account#	cname	balance
Downtown	101	Johnson	500
Lougheed	215	Smith	700
SFU	102	Hayen	400
SFU	301	Adams	1300

D1={Downtown, Lougheed, SFU,...} D2={101, 215, 102, 301,...}
D3={Johnson, Smith, Hayen, Adams,...} D4={0,1,2,3,4,5,6,...}

The Relation *Account* is a subset of the cartesian product D1 x D2 x D3 x D4

Given sets $A_1, A_2, ..., A_n$ a relation r is a subset of $A_1 x A_2 x ...x A_n$
Thus a relation is a set of n-tuples $(a_1, a_2, ..., a_n)$ where $a_i \in A_i$

Account ={(Downtown,101,Johson,500), (Lougheed,215,Smith,700), (SFU,1002,Hayen,400),(SFU,301,Adams,1300)}

Tables = Relations

- Attribute: the name of columns
- tuple: a row in a table
- Domain: the set of possible values for a given attribute

Let t = (SFU,301,Adams,1300)
Then t[bname]=SFU= t[1]
t[account#]=301= t[2]
t[cname]=Adams= t[3]
t[balance]=1300= t[4]
t is a tuple variable

Account

bname	account#	cname	balance
Downtown	101	Johnson	500
Lougheed	215	Smith	700
SFU	102	Hayen	400
SFU	301	Adams	1300

- atomic: All attributes of any relation must be atomic (indivisible units)
- null: null values for either nonexistant or unknown value

Relations are sets => no duplicate rows

- The current values of a relation (relation instance) are specified by a table.
- The logical design of a relational database is represented by a relational scheme
  - A relation scheme is a set of attributes
  - A relational database scheme is a set of relation schemes:
    i.e., a set of sets of attributes

Account=(bname, account#, cname, balance)
Customer=(cname,street,ccity)

Account

bname	account#	cname	balance
Downtown	101	Johnson	500
Lougheed	215	Smith	700
SFU	102	Hayen	400
SFU	301	Adams	1300

Customer

cname	street	ccity
Johnson	Pender	Vancouver
Smith	North	Burnaby
Hayes	Austin	Burnaby
Adams	N3 Road	Richmond
Jones	Oak	Vancouver

- Keys
  - Superkey: a set of attributes that uniquely identifies tuples
  - Candidate key: a minimum set of attributes that form a key
  - Primary key: a designated candidate key

- Strong entity set: primary key of relation = primary key of entity set
- Weak entity set:primary key of relation = union of primary key of strong entity set and descriptor of weak entity set
- Relationship set: primary key of relation = union of primary keys of the related entity sets
  - many to many: as above
  - many to one: primary key of the “many” entity set
  - one to one: either entity sets

Query Languages

For the user to request info from the database
High level languages
facilities to insert, delete and modify data

- Formal languages
  - relational algebra (procedural)
  - relational calculus (non-procedural)
    - tuple relational calculus
    - domain relational calculus
- SQL and other languages

Relational Algebra

Selection $\sigma_{predicate}(relation)$

Account

bname	account#	cname	balance
Downtown	101	Johnson	500
Lougheed	215	Smith	700
SFU	102	Hayen	400
SFU	301	Adams	1300

find all tuples for SFU branch

$\sigma_{bname="SFU"}(account)$ →

bname	account#	cname	balance
SFU	102	Hayen	400
SFU	301	Adams	1300

find all tuples with balance > 600

$\sigma_{balance>600}(account)$ →

bname	account#	cname	balance
Lougheed	215	Smith	700
SFU	301	Adams	1300

Relational Algebra

Projection $\Pi_{attributes-to-retain}(relation)$

Account

bname	account#	cname	balance
Downtown	101	Johnson	500
Lougheed	215	Smith	700
SFU	102	Hayen	400
SFU	301	Adams	1300

find all customers with an account

$\Pi_{cname}(account)$ →

cname
Johnson
Smith
Hayen
Adams

find all customers and their balance

$\Pi_{cname,balance}(account)$ →

cname	balance
Johnson	500
Smith	700
Hayen	400
Adams	1300

duplicates are eliminated

Relational Algebra

<u>Composing relational operation</u>

find names customer with a balance higher than 600

select on balance > 600
project cname

$$\Pi_{\text{cname}}(\sigma_{\text{balance} > 600}(account))$$

cname
Smith
Adams

Relational Algebra

Queries are expressed by applying specialized set operators to relations.

- Five fundamental operators
  - Union: $R \bigcup S = \{t | t \in R \text{ or } t \in S\}$
  - Difference: $R - S = \{t | t \in R \text{ and } t \notin S\}$
  - Select: $\sigma_\theta(R) = \{t | t \in R \text{ and } \theta(t) \text{ is true}\}$

    θ is a formula involving arithmetic comparison expressions
  - Project: $\Pi_{A_1,\ldots,A_n}(R) = \{ t\,[A_1,\ldots,A_n] \mid t \in R \}$
  - Cartesian Product:

    $R \times S = \{ t \mid t[A_1,\ldots A_n] \in R \text{ and } t[A_{n+1},\ldots A_{n=m}] \in S \}$

- Additional Operators
  - Intersect: $R \bigcap S = \{ t \mid t \in R \text{ and } t \in S \}$

    $= R - (R - S)$
  - Natural join:

    $R \rhd\lhd S = \{ t[R \cup S] \mid t[R] \in R \wedge t[S] \in S \}$

    $= \Pi_{R \cup S}(\sigma_\theta(R \times S))$
  - Division:

    $R \div S = \{ t[R-s] \mid \forall u \in S \exists v \in R(v[S] = u \wedge v[R-s] = t\}$

    $= \Pi_{R-s}(R) - \Pi_{R-S}((\Pi_{R-S}(R) \times S) - R)$

- Five fundamental operators are independent of each other.
- The relational algebra is complete in that any reasonable query can be expressed by an algebra expression.

Relational Calculus

- Tuple Calculus
  - A tuple relational calculus expression is of the form

    { t | P(t) }

  where t is a tuple variable, P is a formula built up from atoms using the following rules:
    - an atom is a formula of the form:

      $s \in R,\ s[x]\ \theta\ t[y],\ s[x]\ \theta\ c.$
    - If f is a formula, then so are $\neg$ f and (f).
    - If f_1 and f_2 are formulas, then so are $f_1 \vee f_2$ and $f_1 \wedge f_2$
    - If $f(s)$ is a formula cotaining a free vairable s, then

      $\exists s\ (f(s))$ and $\forall s(f(s))$

      are also formulas.
  - Any query is specified by a tuple relational calculus expression.

- how to determine the relation scheme of a calculus expression {t | P(t)}
  - If $t \in R$ is an atom in P(t) then the scheme of {t|P(t)} is the same as that of R.
  - Otherwise, the scheme is determined by all atoms of the form t[x] = ... appearing in P(t).
- Safety issue
  - the size of {t | P(t)} must be finite.
- The relational calculus and the relational algebra are equivalent in terms of expressive power.