A Relational Model of Data for Large Shared Data Bank

Chang Su

Before Relational Model

Tree Model

- Information Management System (IMS)

Network Model

- Conference/Committee on Data Systems Languages (CODASYL)

Library Example

Database system: Library

Data: Books

Application: programmable robots which help you fetch books

Target: A book named “Effective Modern C++” under Catalog
“Computer Science”

Before Relational Model

1. Go to the 2nd Floor
2. Move to the Northwest corner of current floor
3. Start scanning until find a bookshelf named “Computer Science”
4. Scan current bookshelf until finding a book named “Effective Modern C++”

Fatal Error

Access methods are highly dependent on physical storage representation

Applications upon database system fail when physical storage changes

Proposal

A declarative data-access mechanism

which remain unaffected when changes happen to physical storage representation

What is a relation

The term “Relation” here is originated from its mathematic sense.

Given sets S1, S2 …. Sn(not necessarily distinct), A relation R is a subset of the Cartesian product S1 x S2 x S3 x S4…SN

We call S1 as the first domain of R, S2 as the second domain of R

A relation with n domains is called n-ary relation

What is a relation

Supplier: {s1, s2, s3}

Part: {p1,p2,p3,p4}

Project: {j1,j2,j3,j5}

A binary relation “supply” may look like :

Supplier	Part
S1	P2
S2	P3
S1	P4

What is a relation

For representation convenience, relations are presented in the form of tables, which entails following properties

1. All rows are distinct
2. Row order does not matter
3. Column order matters
4. The identification of columns is partially labeled the domain name

“Partially”

Component(part, part, quantity)

Part	Part	quantity
P1	P2	2
P2	P3	3
P3	P4	4

To free users from remembering column order of high degree relation,

Unique role names for each domain are suggested

Normal Form

When a relation contains only atomic domain, it is in a normal form.

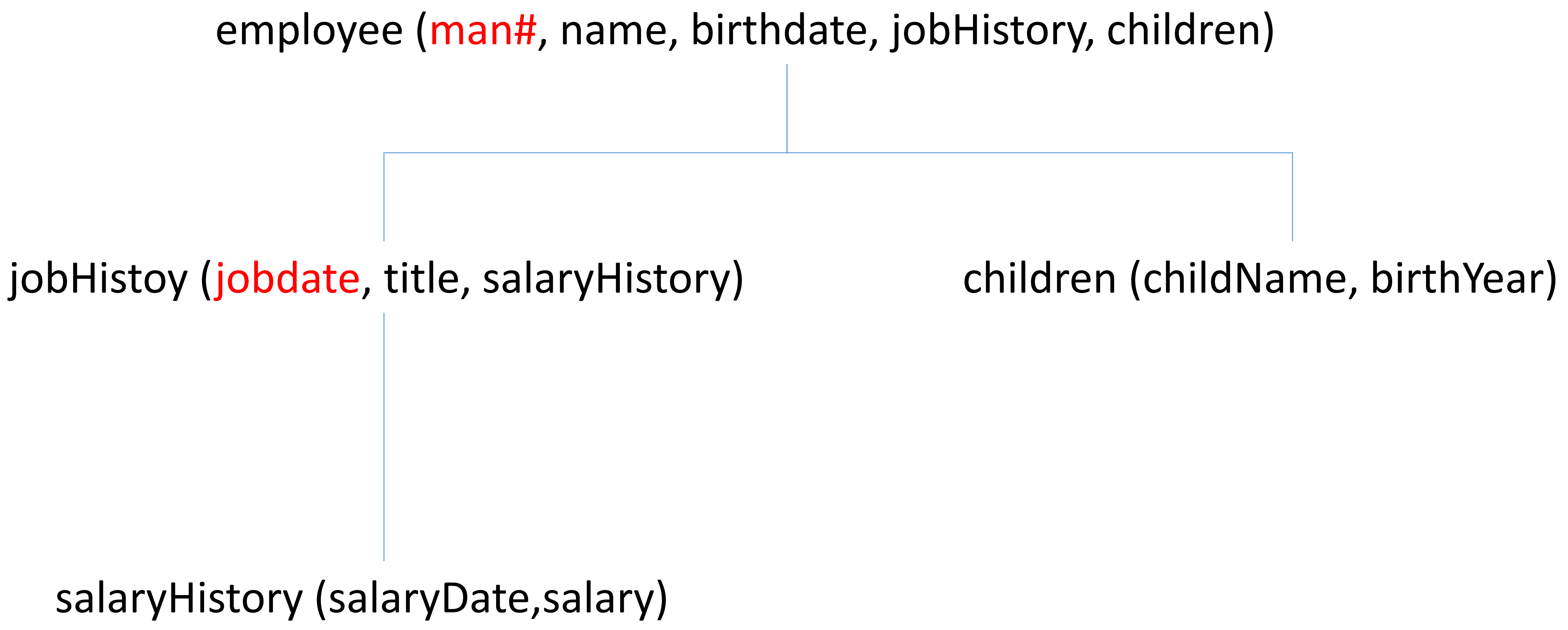
employee (man#, name, birthdate, jobHistory, children)

jobHistoy (jobdate, title, salaryHistory)

salaryHistory (salaryDate,salary)

children (childName, birthYear)

Structure


employee (man#, name, birthdate, jobHistory, children)
jobHistoy (jobdate, title, salaryHistory)
children (childName, birthYear)
salaryHistory (salaryDate,salary)

Normalization

employee (man#, name, birthdate)

jobHistoy (man#, jobdate, title, salaryHistory)

children (man#, childName, birthYear)

salaryHistory (salaryDate,salary)

Normalization

employee (man#, name, birthdate)

jobHistoy (man#, jobdate, title)

children (man#, childName, birthYear)

salaryHistory (man#, jobdate, salaryDate,salary)

Conditions for Normalization

The graph of interrelationships of the non-atomic domains is a collection of trees

No primary key has a component domain which is non-atomic

Normalization

Supply (part, project, date)

pay (supply, price)

Normalized:

pay (part,project, date, price)

Supply(part, project, date)

or

Pay(part, project, date, price)

Benefits from Normal Form

Easy to store with out complex data structure for composite domain

Get rid of pointers and indices in data communication

A easy expression to locate a domain

R(v).r.d

R: relation name
v: version number of relation
r: role name resolving the same repeated domains in a relation
d: Domain Name

Linguistic Aspect

The adoption of relational data model, permits the development of a universal data language based on first-order logic.

The universality of this language lies in its descriptive ability in qualifying data for retrieval.

Supplier	Part
S1	P2
S2	P3
S1	P4

Predicates: supply(x,y) , is_S1(x)

Formula F: $\text{supply}(x,y) \wedge \text{is\_S1}(x)$

Summary

Purpose

- provide a declarative method for specifying data and queries

Users directly state what information they want from it

And, let the database management system software take care of retrieval procedures for answering queries

Operations on Relations

Aimed at enhancing the fine-grained and relation-across descriptive ability on data

Fine-grained

permutation, projection, selection

Relation-across

join, composition, restriction, tie

Fine-grained Operation

Permutation

- The operation turn permute its column order

  No meaning in the user view, since users identify domain by role name

  Considered in the storage representation

Projection

- Select certain columns from a relation, then remove duplication in rows

Fine-grained Operation

Selection

- projection, permutation or their combination

π_{21}(supply)

Part	Supplier
P2	S1
P3	S2
P4	S1

supply

Supplier	Part
S1	P2
S2	P3
S1	P4

π_{1}(supply)

Supplier
S1
S2

Relation-across Operation

Join

- Image two binary relations, R and S, Join of R and S is a relation U such that π_{12}(U) = R and π_{23}(U) = S
- A special join called natural join

R*S = { (s,p,j) | R(s,p) and S(p,j) }

R(s,p) means tuple (s,p) is a row in R

- Easy to prove by contradiction that any join is a subset of the natural join

Some Concern in Database Design

Redundancy

Consistency

Redundancy

Derivable:

- A relation R is derivable from S, if R can be obtained by a sequence of operations, including projection, natural join, restriction and tie.

Strong Redundancy

Informal definition

- Data is stored duplicately but duplicate data brings no extra information

Form definition

A set of relations is strongly redundant if it contains at least one relation that possesses a projection which is derivable from other projections of relations in the set.

Strong Redundancy

Three relations

- SD(s,d) , SJ(s,j) and DJ(d,j)
- d: department, j: projects & s : supplier
- Condition C held all time:

  supplier s supplies department d (relation SD)

  if and only if

  supplier s supplies some project j (relation SJ) to which d is assigned (relation DJ)

$$\pi_{21}(\text{SD}) = \pi_{12}\{\ \text{SJ} * \pi_{21}(\text{DJ})\}$$

Weak Redundancy

Informal Definition

- Data is stored duplicately but duplicate data brings information

Formal Defintion

- A collection of relations is weakly redundant if it contains a relation that has a projection which is not derivable from other relations but is at all times a projection of some join of other projections of relations in the collection.

Weak Redundancy

Remove Condition C

The supply to department is independent of supply to projects

s	d
1	a
2	a
2	b

s	j
a	x
a	y
b	x
b	y

j	d
x	1
x	2
y	2

Consistency

A collection of relations are in consistent state, if they satisfy constraints applied on them.

Condition C is such a constraint

Difficulty

- Unable to tell inconsistency made by normal commit or mistake
- Time consuming in verify constraints

Consistency

Set a timer when inconsistency detected, if not get corrected when timer counts to 0, report to database admin

- This approach slows down operation performance

Maintain data manipulation log and verify consistency once per day.

- Manipulations after the manipulation introducing inconsistent may not be recoverable

Summary

Relational data model

- Aimed at allowing application access data through a declarative way to achieve data independency

Redundancy

- Strong redundancy, redundant but bring no more information
- Weak redundancy, redundant but bring more information

Consistency

- Difficulties
- Approaches and their drawbacks

Unsolved

Relational data model only provides a mean to describe your target data.

We still need an approach to data description into retrieval procedures

- Can these procedures be designed and implemented efficiently?

Q & A