The Relational Model

陆伟

Database Systems

January 10, 2022

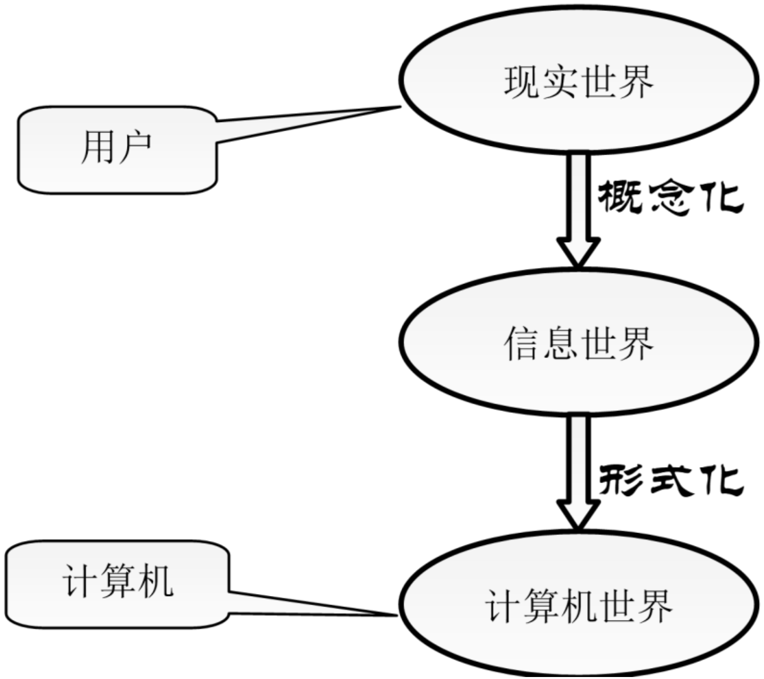
目录

CONTENTS

Data Model

- Model
  - The abstraction of the real world that simulate the real world.
  - A map, the sand table of a building, …
- Data model
  - The abstraction about data characteristics of the real world.
  - An integrated collection of concepts for describing and manipulating data, relationships between data, and constraints on the data in an organization.

Data Model

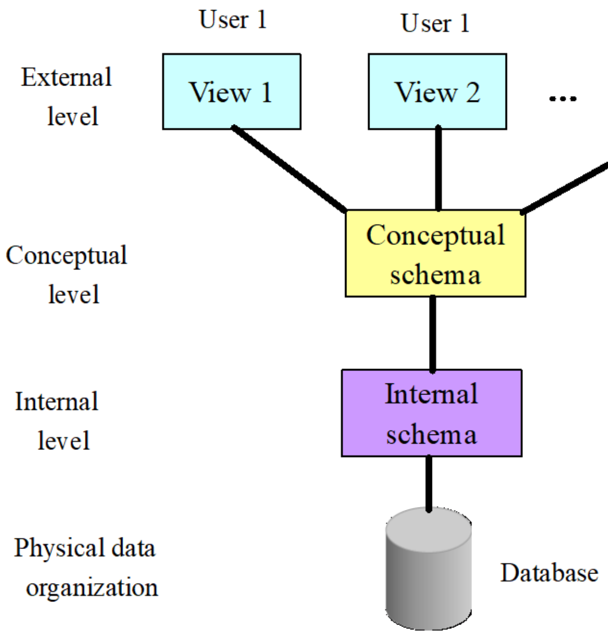


Data Model

- Requirements for abstracting data model
  - Represent the “real world” really
  - Make the data understandable for variety of users
  - Easy to implement on computer
- The three components of data model
  - Data structure
  - Data manipulation
  - Constraints of data integrity

Data Model

- The Three-Level ANSI-SPARC Architecture



Data Model

- The objective of the three-level architecture is to separate each user's view of the database from the way the database is physically represented.
- The reasons for separation
  - Each user should be able to access the same data in different view. Each user should be able to change the way he or she views the data, and this change should not affect other users.
  - A user's interaction with the database should be independent of storage considerations.

Data Model

- The DBA should be able to change the database storage structures without affecting the user's views.
- The internal structure of the database should be unaffected by changes to the physical aspects of storage.
- The DBA should be able to change the conceptual structure of the database without affecting all users.

Data Model

- Schemas
  - The overall description of the database is called the database schema.
  - External schemas (subschemas)
    - Correspond to different views of the data
  - Conceptual schema
    - Describes all the entities, attributes, and relationships together with integrity constraints.
  - Internal schema
    - Describes internal model. Contains the definitions of stored records, the method of representation, the data fields,…

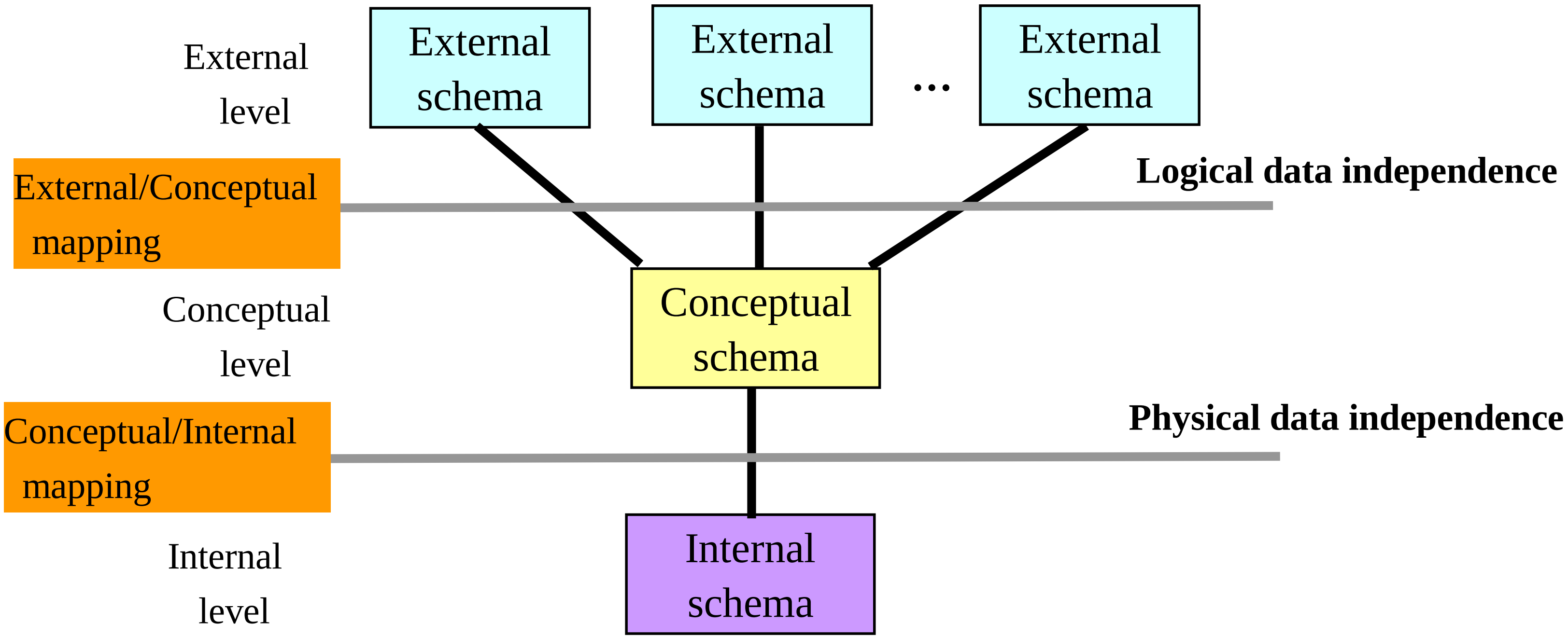
Data Model

- Mapping
  - The three types of schema of database must be consistent.
  - The DBMS is responsible for mapping between these three types of schema.
  - The conceptual schema is related to the internal schema through a conceptual/internal mapping.
  - Each external schema is related to the conceptual schema by the external/conceptual mapping.

Data Model

- Data independence
  - A major objective for the three-level architecture is to provide data independence, which means that upper levels are unaffected by changes to lower levels.
  - Logical data independence
    - Refers to the immunity of the external schemas to changes in the conceptual schema.
  - Physical data independence
    - Refers to the immunity of the conceptual schema to changes in the internal schema

Data Model


External level
External schema
External schema
...
External schema
External/Conceptual mapping
Logical data independence
Conceptual level
Conceptual schema
Conceptual/Internal mapping
Physical data independence
Internal level
Internal schema

Data Model

- The three-level ANSI-SPARC architecture and two-stage mapping may be inefficient, but provides greater data independence.
- For more efficient mapping, the ANSI-SPARC model allows the direct mapping of external schema on to the internal schema, thus by passing the conceptual schema. But …

Data Model

- The procedure of modeling
  - Real world—information world—computer world
- Conceptual schema is the ‘heart’ of the database.
- It supports all the external views and is supported by the internal schema.
- The conceptual schema should be a complete and accurate representation of the data requirements of the enterprise.

Relational Model

- Brief History of the Relational Model
  - E.F.Codd – ‘A relational model of data for large shared data banks’ (1970)
  - RDBMS System R – IBM San Jose Research Laboratory in California
  - INGERS (Interactive Graphics Retrieval System) – University of California at Berkeley
  - Peterlee Relational Test Vehicle – IBM UK Scientific Centre (Todd,1976)

Relational Model

- In the late 1970s and early 1980s, commercial system based on the relational model started to appear.
- Some extensions to the relational model
  - Capture more closely the meaning of data (Codd, 1979)
  - Support object-oriented concepts (Stonebraker and Rowe, 1986)
  - Support deductive capabilities (Gardarin and Valduriez, 1989)

Relational Model

- Relational Data Structure
  - The relational model is based on the mathematical concept of a relation, which is physically represented as a table.
  - The relational data structure are principally based on set theory and predicate logic from mathematics.

Relational Model

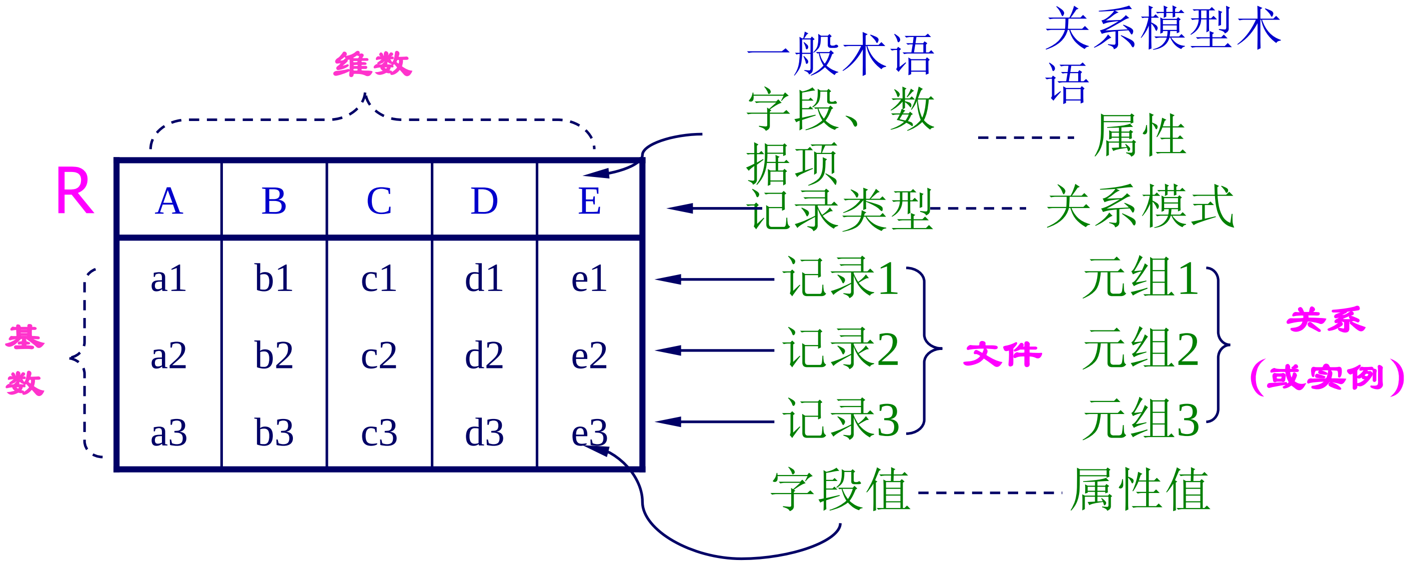
- Key concepts and terminologies
  - Relation
    - A relation is a table with columns and rows.
    - Mathematical definition -- Any subset of n-tuples from the Cartesian product of n sets is a relation on the n sets.
  - Cartesian product
    - Let $D_1, D_2, \ldots, D_n$ be n sets. Their Cartesian product is defined as:

$$D_1 \times D_2 \times \ldots \times D_n = \{(d_1, d_2, \ldots, d_n) \mid d_1 \in D_1,\ d_2 \in D_2,\ \ldots,\ d_n \in D_n\}$$

    - Usually written as:

$$\mathop{\mathrm{X}}_{i=1}^{n} D_i$$

Relational Model



Relational Model

- Properties of Relations
  - A relation must be a finite set. Otherwise, it could not be stored in a computer
  - The order of attributes has no significance
  - The values of an attribute are homogeneous that is the values of an attribute are all from the same domain
  - The order of tuples has no significance, theoretically. (In practice, the order may affect the efficiency of accessing tuples)
  - Each cell of the relation contains exactly one atomic (single) values
  - Each tuple is distinct; there are no duplicate tuples

Relational Model

- Relational keys
  - Superkey
    - An attribute, or set of attributes, that uniquely identifies a tuple within a relation
  - Candidate key
    - A superkey such that no proper subset is a superkey within a relation
  - Primary key
    - The candidate key that is selected to identify tuples uniquely within a relation
    - The selection of primary key – semantic, pragmatic

Relational Model

- Alternate key (辅关键字)
  - A candidate key that is not selected to be the primary key
- Foreign key
  - An attribute, or set of attributes, within one relation that matches the candidate key of some (possibly the same) relation
- Examples about keys

Relational Model

- Representing Relational Database Schemas
  - Give the name of the relation followed by the attribute names in parentheses.
  - Underline the primary key
  - Examples

Student (sNo, sName, sSex, sAge, sDept)
Course (cNo, cName, cPNo, cCredit)
SC (sNo, cNo, score)

Relational Model

- Relational Integrity
  - Null
    - Represents a value for an attribute that is currently unknown or is not applicable for this tuple.
    - Null can cause implementation problems because the relational model is based on predicate calculus, which is a two-valued or Boolean logic.
    - The incorporation of nulls in the relational model is a contentious issue.

Relational Model

- Entity integrity
  - In a base relation, no attribute of a primary key can be null.
- Referential integrity
  - If a foreign key exists in a relation, either the foreign value must match a candidate key value of some tuple in its home relation or the foreign key value must be wholly null.
- Enterprise constraints
  - Additional rules specified by the users or database administrators of a database
- Examples about integrity

Relational Model

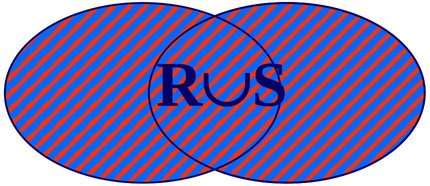
- Relational manipulation
  - The data manipulation on relational model is in fact the manipulation on relation or set.
  - The relational algebra and relational calculus are two formal, non-user-friendly languages but they have been used as the basis for other higher-level Data Manipulation Languages (DMLs) for relational database.
  - The relational algebra and relational calculus are equivalent to one another.

Relational Algebra

- The relational algebra is a theoretical language with operations that work on one or more relations to define another relation without changing the original relation(s).
- Relations are closed under the algebra.
  - Both the operands and the results are relations
- Informally, we may describe the relational algebra as a (high-level) procedural language.
  - It can be used to tell the DBMS how to build a new relation from one or more relations in the database.

Relational Algebra

- Set Operations
  - Union (并)
    - The union to two relations R and S defines a relation that contains all the tuples of R, or S, or both R and S, duplicate tuples being eliminated.
    - R and S must be union-compatible. – the arities (degree) of the two relations must be equivalent.
    - $R \cup S = \{ t \mid t \in R \vee t \in S \}$



Relational Algebra

R

A	B	C
3	6	7
2	5	7
7	2	3
4	4	3

S

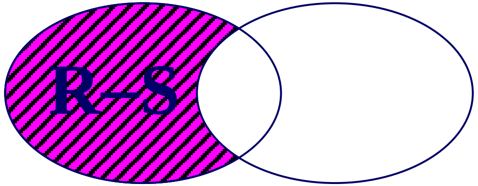
A	B	C
3	4	5
7	2	3

R U S

A	B	C
3	6	7
2	5	7
7	2	3
4	4	3
3	4	5

Relational Algebra

- Set difference (差)
  - The set difference operation defines a relation consisting of the tuples that are in relation R, but not in S.
  - R and S must be union-compatible.
  - $R - S = \{t \mid t \in R \wedge t \notin S\}$



Relational Algebra

R

A	B	C
3	6	7
2	5	7
7	2	3
4	4	3

S

A	B	C
3	4	5
7	2	3

S — R

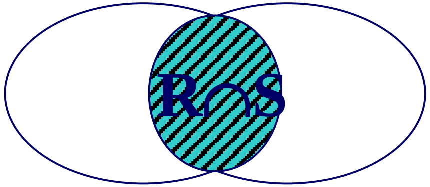
A	B	C
3	6	7

R — S

A	B	C
3	6	7
2	5	7
4	4	3

Relational Algebra

- Intersection (交)
  - The intersection operation defines a relation consisting of the set of all tuples that are in both R and S.
  - R and S must be union-compatible.
  - $R \cap S = \{t \mid t \in R \wedge t \in S\}$
  - We can express the intersection operation in terms of the set difference operation: $R \cap S = R - (R - S)$



Relational Algebra

R

A	B	C
3	6	7
2	5	7
7	2	3
4	4	3

S

A	B	C
3	4	5
7	2	3

$R \cap S$

A	B	C
7	2	3

Relational Algebra

- Cartesian product
  - The Cartesian product operation defines a relation that is the concatenation of every tuple of relation R with every tuple of relation S.

$$R \times S = \{t \mid t = \langle t_r, t_s \rangle \wedge t_r \in R \wedge t_s \in S\}$$

  - Let the arities of R and S be m and n. Let cardinalities of R and S be k1 and k2
  - The arity (degree) of the new relation will be m+n
  - The cardinality of the new relation will be k1 $\times$ k2

Relational Algebra

R

A	B
α	1
β	2

S

C	D	E
α	10	a
β	10	a
β	20	b
γ	10	b

R x S

A	B	C	D	E
α	1	α	10	a
α	1	β	19	a
α	1	β	20	b
α	1	γ	10	b
β	2	α	10	a
β	2	β	10	a
β	2	β	20	b
β	2	γ	10	b

Relational Algebra

- Unary Operations
  - Selection (Restriction) 选择
    - The selection operation works on a single relation R and defines a relation that contains only those tuples of R that satisfy the specified condition (predicate).
    - $\sigma_F(R) = \{t \mid t \in R \wedge F(t) = \text{'true'}\}$
    - The selection operate the relation from the view of rows.

Relational Algebra

R

A	B	C
3	6	7
2	5	7
7	2	3
4	4	3

$\sigma_{A<5}(R)$

A	B	C
3	6	7
2	5	7
4	4	3

$\sigma_{A<5 \wedge C=7}(R)$

A	B	C
3	6	7
2	5	7

Relational Algebra

- Projection (投影)
  - The projection operation works on a single relation R and defines a relation that contains a vertical subset of R, extracting the values of specified attributes and eliminating duplicates.
  - $\Pi_A(R) = \{ t[A] \mid t \in R \}$
  - The projection operate the relation from the view of columns.

Relational Algebra

R

A	B	C
a	b	c
d	e	f
c	b	c

$\Pi_{B\,,\,C}(R)$

B	C
b	c
e	f

Relational Algebra

- Join Operations
  - Theta join (θ-join)
    - The theta join operation defines a relation that contains tuples satisfying the predicate *F* from the Cartesian product of R and S. The predicate *F* is of the form $R.a_i \theta S.b_i$ where θ may be one of the comparison operators ($<$, $\leq$, $>$, $\geq$, $=$, $\neq$)
    - $R \bowtie_{A\theta B} S = \{ t_r t_s \mid t_r \in R \wedge t_s \in S \wedge t_r[A] \,\theta\, t_s[B] \}$
    - We can rewrite the Theta join in terms of basic selection and Cartesian product operation

$$R \bowtie_{A \theta B} S = \sigma_{r[A] \theta S[B]} (R \times S)$$

Relational Algebra

R

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

S

D	E
3	1
6	2

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

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

Relational Algebra

- Equijoin (相等连接)
  - When θ is '=' in theta join
  - A particular type of Theta join
  - $R \underset{A=B}{\bowtie} S = \{ \widehat{t_r t_s} \mid t_r \in R \land t_s \in S \land t_r[A] = t_s[B] \}$

Relational Algebra

- Natural join
  - The natural join is an equijoin of two relations R and S over all common attributes x. One occurrence of each common attribute is eliminated from the result.
  - A particular type of equijoin

$$R \bowtie S = \{ \widehat{t_r t_s} \mid t_r \in R \wedge t_s \in S \wedge t_r[X] = t_s[X] \}$$

  - When there are no common attributes in R and S, the equijoin is equal to Cartesian product.

$$R \bowtie S = R \times S$$

Relational Algebra

R

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

S

C	D
3	1
6	2

$R \bowtie S$

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

Relational Algebra

R

A	B	C	D
α	1	α	a
β	2	γ	a
γ	4	β	b
α	1	γ	a
δ	2	β	b

S

B	D	E
1	a	α
3	a	β
1	a	γ
2	b	δ
3	b	$\in$

R $\bowtie$ S

A	B	C	D	E
α	1	α	a	α
α	1	α	a	γ
α	1	γ	a	α
α	1	γ	a	γ
δ	2	β	b	δ

Relational Algebra

- A question in natural join

T

tNo	tName	salary
P01	赵明	800
P02	钱广	700
P03	孙立	600
P04	李三	500

C

cNo	cName	tNo
C01	DS	P01
C02	DB	P02
C03	OS	P04

How to produce a relation consisting of all teachers' information with the courses they teach including the teachers who have no course.

Relational Algebra

T

tNo	tName	salary
P01	赵明	800
P02	钱广	700
P03	孙立	600
P04	李三	500

$\bowtie$

C

cNo	cName	tNo
C01	DS	P01
C02	DB	P02
C03	OS	P03

tNo	tName	salary	cNo	cName
P01	赵明	800	C01	DS
P02	钱广	700	C02	DB
P03	孙立	600	C03	OS

Relational Algebra

- Outer join
  - Often in joining two relations, a tuple in one relation does not have a matching tuple in the other relation. We may want a tuple from one of the relation to appear in the result even when there is no matching value in the other relation.
  - The (left) Outer join is a join in which tuples from R that do not have matching values in the common attributes of S are also included in the result relation.
  - Missing values in the second relation are set to null.
  - Left outer join, right outer join, full outer join
  - R ⟕ S, R ⟖ S, R ⟗ S

Relational Algebra

T

tNo	tName	salary
P01	赵明	800
P02	钱广	700
P03	孙立	600
P04	李三	500

⟕

C

cNo	cName	tNo
C01	DS	P01
C02	DB	P02
C03	OS	P03

tNo	tName	salary	cNo	cName
P01	赵明	800	C01	DS
P02	钱广	700	C02	DB
P03	孙立	600	C03	OS
P04	李三	500	null	null

Relational Algebra

- Semijoin
  - The semijoin operation defines a relation that contains the tuples of R that participate in the join of R with S.

$$R \triangleright_{F} S = \Pi_{A}(R \bowtie_{F} S)$$

A is the set of all attributes for R

Relational Algebra

- An example for semijoin

R

A	B	C	D
α	1	α	a
β	2	γ	a
γ	4	β	b
α	1	γ	a
δ	2	β	b

S

B	D	E
1	a	α
3	a	β
1	a	γ
2	b	δ
3	b	∈

R ▷ S

A	B	C	D
α	1	α	a
α	1	α	a
α	1	γ	a
α	1	γ	a
δ	2	β	b

Relational Algebra

- Division Operation
  - Images set (象集)
    - Relation R(X,Z), X and Z are sets of attributes of R. For each t[X] = x, we can define the images set of x Z_x in R :
    - $Z_x=\{t[Z] \mid t \in R, t[X]=x\}$
    - 它表示R中属性组X上值为x的诸元组在Z上分量的集合。

X	Z
姓名	课程
张军	物理
王红	数学
张军	数学

x=张军

Z_x
课程
数学
物理

张军同学所选修的全部课程

Relational Algebra

- Division
  - 给定关系R(X,Y)和S(Y,Z)，其中X, Y, Z为属性组。 R中的Y与S中的Y可以有不同的属性名，但必须出自相同的域集。R与S的除运算得到一个新的关系P(X)，P是R中满足下列条件的元组在X属性列上的投影：**元组在X上分量值x的象集Y_x包含S在Y上投影的集合**。记作:

$$R \div S = \{ t_r[X] \mid t_r \in R \wedge \Pi_Y(S) \subseteq Y_x \}$$

  - We can rewrite the division expression using difference and Cartesian operation as follow:

$$R \div S = \Pi_X(R) - \Pi_X(\Pi_X(R) \times \Pi_Y(S) - R)$$

Relational Algebra

R

A	B	C	D
a	b	c	d
a	b	e	f
a	b	d	e
b	c	e	f
e	d	c	d
e	d	e	f

S

C	D
c	d
e	f

R÷S=?

Relational Algebra

$CD_{(a,b)} = \{(c,d),(e,f),(d,e)\}$
$CD_{(b,c)} = \{(e,f)\}$
$CD_{(e,d)} = \{(c,d),(e,f)\}$
$\{(c,d),(e,f)\} \subseteq CD_{(a,b)}$
$\{(c,d),(e,f)\} \subseteq CD_{(e,d)}$

$\Rightarrow R \div S = \{(a,b),(e,d)\}$

Relational Algebra

R

A	B	C	D
a	b	c	d
a	b	e	f
a	b	d	e
b	c	e	f
e	d	c	d
e	d	e	f

S

C	D
c	d
e	f

$\Pi_{AB}(R)$

A	B
a	b
b	c
e	d

$\Pi_{AB}(R) \times \Pi_{CD}(S)$

A	B	C	D
a	b	c	d
a	b	e	f
b	c	c	d
b	c	e	f
e	d	c	d
e	d	e	f

$\Pi_{AB}(R) \times \Pi_{CD}(S) - R$

A	B	C	D
b	c	c	d

$R \div S =$

A	B
a	b
b	c
e	d

-

A	B
b	c

=

A	B
a	b
e	d

Relational Algebra

- Try : Given relations R and S as following, compute the result of R÷S

R

A	B	C
a1	b1	c2
a1	b2	c3
a1	b2	c1
a2	b3	c7
a2	b2	c3
a3	b4	c6
a4	b6	c6

S

B	C	D
b1	c2	d1
b2	c3	d2
b2	c1	d3
b1	c2	d4

Relational Algebra

在关系R中，A可以取四个值{a1, a2, a3, a4}。其中：

a1的象集为{(b1,c2), (b2,c3), (b2,c1)}

a2的象集为{(b3,c7), (b2,c3)}

a3的象集为{(b4,c6)}

a4的象集为{(b6,c6)}

S在(B,C)上的投影为{(b1,c2),(b2,c3),(b2,c1)}

显然只有a1的象集$(B,C)_{a1}$包含S在(B,C)属性组上的投影，所以

R÷S={a1}

Relational Algebra

- The indication of division

姓名	课程
张军	物理
王红	数学
张军	数学
王红	物理

÷

课程
数学
物理

＝

姓名
张军
王红

选修了全部课程的学生

姓名	课程	成绩
张军	物理	93
王红	数学	86
张军	数学	93
王红	物理	92

÷

课程
数学
物理

＝

姓名	成绩
张军	93

选修了全部课程并且成绩都相同的学生

Examples of Relational Algebra

- Note: The following exercises are all based on two relation schemas

Student (<u>sNo</u>, sName, sSex, sAge, sDept)
Course (<u>cNo</u>, cName, cPNo, cCredit)
SC (<u>sNo, cNo</u>, score)

Examples of Relational Algebra

- List all students in information system department

  $\sigma_{sDept='IS'}(Student)$

- List all male students whose age no less than 20

  $\sigma_{sAge \geq 20 \wedge sSex='M'}(Student)$

- List all students with name and department attributes

  $\Pi_{sName,sDept}(Student)$

- List the course numbers which elected by student whose number is '070001'

  $\Pi_{cNo}(\sigma_{sNo='070001'}(SC))$

Examples of Relational Algebra

- List the students who have elected course ‘001’ or ‘002’

$$\Pi_{sNo}(\sigma_{cNo='001' \lor cNo='002'}(SC)) \bowtie Student$$

$$(\Pi_{sNo}(\sigma_{cNo='001'}(SC)) \cup \Pi_{sNo}(\sigma_{cNo='002'}(SC))) \bowtie Student$$

Examples of Relational Algebra

- List the students who have elected course ‘001’ and ‘002’

$$\Pi_{sNo}(\sigma_{cNo='001' \wedge cNo='002'}(SC)) \bowtie Student$$ ✗

$$(\Pi_{sNo}(SC \div \sigma_{cNo='001' \vee cNo='002'}(C))) \bowtie Student$$ ✗

$$(\Pi_{sNo,cNo}(SC) \div \sigma_{cNo='001' \vee cNo='002'}(C)) \bowtie Student$$ √

$$(\Pi_{sNo}(\sigma_{cNo='001'}(SC)) \cap \Pi_{sNo}(\sigma_{cNo='002'}(SC))) \bowtie Student$$ √

Examples of Relational Algebra

- List the students who have elected course '001' but have not elected course '002'

$(\Pi_{sNo}(\sigma_{cNo='001'}(SC) - \sigma_{cNo='002'}(SC))) \bowtie Student$ ✗

$(\Pi_{sNo}(\sigma_{cNo='001'}(SC)) - \Pi_{sNo}(\sigma_{cNo='002'}(SC))) \bowtie Student$ √

Examples of Relational Algebra

- List the students who have not elected course '001'

$(\Pi_{sNo}(\text{Student}) - \Pi_{sNo}(\sigma_{cNo='001'}(SC))) \bowtie \text{Student}$ √

$\Pi_{sNo}(\sigma_{cNo\neq'001'}(SC)) \bowtie \text{Student}$ ✗

sNo	sName	sAge
s1	…	…
s2	…	…
s3	…	…

sNo	cNo	score
s1	001	90
s2	002	95
s1	002	96

Examples of Relational Algebra

- List the student who have only elected course '001'

$$(\Pi_{sNo}(SC) - \Pi_{sNo}(\sigma_{cNo \neq '001'}(SC))) \bowtie Student$$

$$||$$

$$(\Pi_{sNo}(\sigma_{cNo = '001'}(SC)) - \Pi_{sNo}(SC - \sigma_{cNo='001'}(SC))) \bowtie Student$$

Examples of Relational Algebra

sNo	cNo	score
s01	001	96
s02	001	90
s03	002	88
s01	003	92

—

sNo	cNo	score
s01	001	96
s02	001	90

=

sNo	**cNo**	**score**
s03	**002**	**88**
s01	**003**	**92**

sNo
s01
s02

—

sNo
s01
s03

=

sNo
s02

Examples of Relational Algebra

- Produce a status report on property viewings.
  - Note: In this case, we want to produce a relation consisting of the properties that have been viewed with comments and those that have not been viewed.

$$\Pi_{\text{propertyNo,street,city}}(\text{PropertyForRent}) ⟕ \text{Viewing}$$

Decomposing complex operations

- The relational algebra operations can be of arbitrary complexity. We can decompose such operations into a series of smaller relational algebra operations.
- The operations we often use to decompose complex algebra expression: assignment operation ($\leftarrow$) and rename operation (ρ).

Decomposing complex operations

- Assignment operation (←)
  - Used to give a name to the results of intermediate expressions

$$R \div S = \Pi_X(R) - \Pi_X(\Pi_X(R) \times \Pi_Y(S) - R)$$

We can use assignment operation to rewrite it as follow：

$temp1 \leftarrow \Pi_X(R)$

$temp2 \leftarrow \Pi_X(temp1 \times \Pi_Y(S) - R)$

$result \leftarrow temp1 - temp2$

Decomposing complex operations

- Rename operation (ρ)
  - Used to give a name to the result of a relational algebra operation and allows an optional name for each of the attributes of the new relation to be specified.

$\rho_S(E)$ or $\rho_{S(a1,a2,\ldots,an)}(E)$

The rename operation provides a new name S for the expression E, and optionally names the attributes as $a_1, a_2, \ldots, a_n$.

Relational Calculus

- A certain order is always explicitly specified in a relational algebra expression and a strategy for evaluating the query is implied.
- A relational calculus query specifies *what* is to be retrieved rather than *how* to retrieve it.
- Informally, we may describe the relational calculus as a (high-level) non-procedural language.
  - It can be used to formulate the definition of a relation in terms of one or more database relations.

Relational Calculus

- Relational calculus is based on predicate calculus in symbolic logic.
- Two forms of relational calculus:
  - Tuple relational calculus (Codd, 1972)
  - Domain relational calculus (Lacroix and Pirotte, 1977)

Relational Calculus

- Tuple relational calculus
  - In the tuple relational calculus we are interested in finding tuples for which a predicate is true.
  - It is based on the use of *tuple variables*. A tuple variable is a variable that ‘range over’ a named relation.
  - Stuff(S) – specify the range of a tuple variable S as the *Stuff* relation.
  - General form of the domain relational calculus:

$$\{S \mid F(S)\}$$

Relational Calculus

- We can use the existential and universal quantifiers (存在量词∃和全称量词∀)in formulate of relational calculus. (┐)

$\{S.fName, S.lName \mid Staff(S) \wedge S.position='Manager' \wedge S.salary>25000\}$

List the staff who manage properties for rent in Glasgow

$\{S \mid Staff(S) \wedge (\exists P)(PropertyForRent(P) \wedge (P.staffNo=S.staffNo) \wedge P.city='Glasgow')\}$

Relational Calculus

- Domain Relational Calculus
  - In the tuple relational calculus, we use variables that range over tuples in a relation. In the domain relational calculus, we also use variables but in this case the variables take their values from domains of attributes rather than tuples of relations.
  - General form of the domain relational calculus:

$$\{d_1,d_2,\ldots,d_n \mid F(d_1,d_2,\ldots,d_m)\} \qquad m \geq n$$

$\{fN,lN \mid (\exists sN,posn,sex,DOB,sal,bN)$
$(Staff(sN,fN,lN,posn,sex,DOB,sal,bN) \wedge posn=\text{'Manager'} \wedge sal>25000)\}$

QBE (Query By Example)

Relational DBMS

- PostgreSQL
  - https://www.postgresql.org/
- OpenGauss
  - https://www.opengauss.org/zh/

Summary

- In this chapter you should have learned:
  - The concept of data model and it’s three components
  - The Three-Stage architecture and Two-Level mapping of database
  - Concepts in relational model
  - The three components and their complementation of relational model
  - The five fundamental operations in relational algebra, Selection, Projection, Cartesian product, Union, and set difference. In addition, Join, Intersection, and Division operations which can be expressed in terms of the five basic operations.

祝各位学习愉快！