Relational Model and Relational Algebra
A Short Review
Class Notes - CS582 - 01

Tran Cao Son

January 12, 2002

0.1 Basics

The following definitions are from the book [1]

Relational Model. Relations are tables representing information. Columns are headed by attributes.; each attribute has an associated domain (also: data type). Rows are called tuples, and a tuple has one component for each attribute of the relation.

Example: a relation about movies[1]

Title	Year	Length	Type
Harry Porter	2001	180	Color
Gone with the wind	1938	125	Black and White
Snow White	1950	90	Color

The Movie Relation

Relation schema: The name of the relation and the set of attributes for the relation.

Example:

$$Movie(Title, Year, Length, Type)$$

is the relation schema of a relation named $Movie$ (table above) and the set of attributes

$$\{Title, Year, Length, Type\}.$$

Tuples: Each row of the relation is a *tuple*. In the above

$$(HarryPorter, 2001, 180, Color)$$

is a tuple.

Relation Instances: the set of tuples belonging to the relation.

[1]Data are invented to illustrate the concepts. They might or might not be accurate.

Functional Dependencies. A functional dependency is of the form

$$A_1 \dots A_n \rightarrow B.$$

It states that tuples agree on the values of the attributes $A_1, \dots, A_n$ must also agree on the value of B.

Keys. A set of attributes $\{A_1, \dots, A_n\}$ is a *key* of a relation R if

- it functionally determines all other attributes, and
- none of its proper subsets functionally determines all other attributes (i.e., it is *minimal*)

A set of attributes that contains a key is a *superkey*.

Rules to derive new functional dependencies

- Reflexivity: If $\{B_1, \dots, B_m\} \subseteq \{A_1, \dots, A_n\}$ then $A_1 \dots A_n \rightarrow B_1 \dots B_m$.
- Augmentation: If $A_1 \dots A_n \rightarrow B_1 \dots B_m$ then $A_1 \dots A_n C_1 \dots C_k \rightarrow B_1 \dots B_m C_1 \dots C_k$.
- Transitivity: If $A_1 \dots A_n \rightarrow B_1 \dots B_m$ and $B_1 \dots B_m \rightarrow C_1 \dots C_k$ then $A_1 \dots A_n \rightarrow C_1 \dots C_k$.

Closure of Attributes.

$$\{A_1, \dots, A_n\}^+ = \{A \mid A_1 \dots A_n \rightarrow A\}.$$

Non-trivial Functional Dependency. $A_1 \dots A_n \rightarrow B_1 \dots B_m$ in nontrivial if at least on of the B's is not among the A's and completely nontrivial if none of the B's is one of the A's.

BCNF and Third Normal Forms *BCNF*: R is in BCNF if for every nontrivial functional dependency $A_1 \dots A_n \rightarrow B_1 \dots B_m$ of R, $\{A_1, \dots, A_n\}$ is a superkey for R.

Decomposing into BCNF: See algorithm in book.

Third Normal Form (or 3NF): R is in 3NF if for every nontrivial functional dependency $A_1 \dots A_n \rightarrow B_1 \dots B_m$ of R, $\{A_1, \dots, A_n\}$ is a superkey for R or B is part of some key.

Decomposing into 3NF: See algorithm in book.

Multivalued Dependencies and Fourth Normal Form A multivalued dependency is of the form

$$A_1 \dots A_n \twoheadrightarrow B_1 \dots B_m.$$

It states that for each pair of tuples t and u of R that agree on the values of the attributes $A_1, \dots, A_n$ we can find in R some tuple r that agrees

1. with both t and u on the A's,
2. with t on the A's, and
3. with u on the attributes that are not among the A's or B's.

Rules to derive new multivalued dependencies

- Trivial: If $A_1 \dots A_n \twoheadrightarrow B_1 \dots B_m$
  then $A_1 \dots A_n \twoheadrightarrow C_1 \dots C_k$ where $\{C_1, \dots, C_k\} \subseteq \{A_1, \dots, A_n\} \cup \{B_1, \dots, B_m\}$.
- Transitivity: If $A_1 \dots A_n \twoheadrightarrow B_1 \dots B_m$
  and $B_1 \dots B_m \twoheadrightarrow C_1 \dots C_k$
  then $A_1 \dots A_n \twoheadrightarrow C_1 \dots C_k$.
- Completementation: If $A_1 \dots A_n \twoheadrightarrow B_1 \dots B_m$
  then $A_1 \dots A_n \twoheadrightarrow C_1 \dots C_k$ where $\{C_1, \dots, C_k\} = AR \setminus (\{A_1, \dots, A_n\} \cup \{B_1, \dots, B_m\})$
  and AR is the set of attributes of R.
- FD implication: If $A_1 \dots A_n \to B_1 \dots B_m$
  then $A_1 \dots A_n \twoheadrightarrow B_1 \dots B_m$.

Non-trivial multivalued Dependency. $A_1 \dots A_n \twoheadrightarrow B_1 \dots B_m$ in nontrivial if none of the B's is among the A's and $\{A_1, \dots, A_n\} \cup \{B_1, \dots, B_m\})$ is not the set of all attributes of R.

Fourth Normal Form R is in 4NF if for every nontrivial multivalued dependency $A_1 \dots A_n \twoheadrightarrow B_1 \dots B_m$ for R, $\{A_1 \dots A_n\}$ is a superkey.

Relational Algebra. This algebra is an important form of query language for the relational model.

The operators of the relational algebra: divided into the following classes:

1. *Set operators*: union, intersection, and set difference
2. *Operators that remove part of the relation*: projection or selection
3. *Operatosr that combine part of the relation*: cartesian product or joint
4. *Renaming operators*: rename the schema of the relation

0.2 Set operators

Union ($\cup$): $R \cup S$ represents a new relation whose tuples are either tuples belonging to R, to S, or both.

Set difference ($\setminus$): $R \setminus S$ represents a new relation whose tuples are tuples belonging to R but not to S.

Intersection ($\cap$) : $R \cap S$ represents a new relation whose tuples are tuples belonging to R and S.

Note: Set operators only work on relations with the same schema.

Example 1 *Let R be the following relation:*

A	*B*	*C*
1	*2*	*3*
2	*1*	*1*
3	*2*	*2*
4	*1*	*3*

and S be the following relation:

A	*B*	*C*
2	*3*	*4*
2	*1*	*1*
1	*3*	*4*
4	*3*	*4*

Then:
$R \cup S$ is

A	*B*	*C*
1	*2*	*3*
2	*1*	*1*
3	*2*	*2*
4	*1*	*3*
2	*3*	*4*
1	*3*	*4*
4	*3*	*4*

$R \cap S$ is

A	*B*	*C*
2	*1*	*1*

and
$R \setminus S$ *is*

A	*B*	*C*
1	*2*	*3*
3	*2*	*2*
4	*1*	*3*

NOTE: *If the schema of S changes, say $S(A1, B1, C1)$ then all of the three operations ($\cup$, $\cap$, $\setminus$) will yield the empty relation.*

0.3 Operators that remove part of the relation

Projection (π) This operation removes some columns from a relation. We write $\pi_{A_1,\ldots,A_k}(R)$ and the result is a new relation that has only the columns $A_1, \ldots, A_k$ of R and whose schema is the set of attributes $\{A_1, \ldots, A_k\}$ of the relation R.

For the relation R in Example 1,

$$\pi_{A,B}(R)$$

is the relation:

A	B
1	2
2	1
3	2
4	1

Selection (σ) The operation selects some tuples from the current relation and defines a new relation. $\sigma_C(R)$ represents a new relation that has tuples belonging to R and each of them satisfying the conditional expression (or condition) C.

C is a conditional expression constructed from attributes of R or constants. For example,

$$A < B + C$$

would be a valid conditional expression with respect to the relation R in Example 1.

For the relation R in Example 1, $\sigma_{A<B+C}(R)$ is the following relation:

A	B	C
3	2	2

0.4 Operators that combine relations

Cartesian Product ($\times$) Given two relations R and S. The Cartesian product $R \times S$ is the set of pairs that can be formed by chossing the first element of the pair to be any element of R and the second an element of S.

Example 2 *Let R be the following relation:*

A	*B*	*C*
1	*2*	*3*
2	*1*	*1*
3	*2*	*2*
4	*1*	*3*

and S be the following relation:

A	*D*
2	*3*
2	*1*

Then:
$R \times S$ *is*

A.R	*B*	*C*	*A.S*	*D*
1	*2*	*3*	*2*	*3*
2	*1*	*1*	*2*	*3*
3	*2*	*2*	*2*	*3*
4	*1*	*3*	*2*	*3*
1	*2*	*3*	*2*	*1*
2	*1*	*1*	*2*	*1*
3	*2*	*2*	*2*	*1*
4	*1*	*3*	*2*	*1*

Natural Joins ($\bowtie$) Combine two relations but using only tuples that are matching in some way. Natural join considers only those pairs that have identical values on whatever attributes the two relations share. $R \bowtie S$ is a new relation whose schema contains the attributes in R and those attributes in S that are not in R. It has the tuples formed by pairing a tuple in R with a tuple in S provided that both have the same values on attributes shared by R and S.

Example 3 *Let R be the following relation:*

A	*B*	*C*
1	*2*	*3*
2	*1*	*1*
3	*2*	*2*
4	*1*	*3*

and S be the following relation:

B	*D*
2	*3*
3	*1*

Then:
$R \bowtie S$ *is*

A	*B*	*C*	*D*	
1	*2*	*3*	*3*	
3	*2*	*2*	*3*	

The first tuple of $R \bowtie S$ *is combined from the 1*st *tuple of* R *and the 1*st *tuple of* S*. The second tuple of* $R \bowtie S$ *is combined from the 3*st *tuple of* R *and the 1*st *tuple of* S*. For the second tuple of* S*, the value of* B *is 3 and none of the tuples in* R *has* $B = 3$*. So, no new tuple for* $R \bowtie S$ *can be formed using the second tuple of* S*.*

Theta-join This is a combination of Cartesian product and selection. $R \bowtie_C S$ is defined by

$$R \bowtie_C S = \sigma_C(R \times S).$$

0.5 Renaming Operator

Renaming (ρ): $\rho_{S(A_1,\ldots,A_k)}(R)$ creates a new relation, named S, and attributes $\{A_1, \ldots, A_k\}$ with all the tuples of R.

Example 4 *Let* R *be the following relation:*

A	*B*	*C*
1	*2*	*3*
2	*1*	*1*
3	*2*	*2*
4	*1*	*3*

Then, $\rho_{T(M,N,P)}(R)$ *is a new relation* T *with the following instance*

M	*N*	*P*
1	*2*	*3*
2	*1*	*1*
3	*2*	*2*
4	*1*	*3*

NOTE: $\rho_S(R)$ only changes the name of the relation.

0.6 Combining Operations to Form Queries

The operators of relational algebra allows us to form expressions of arbitrary complexity by applying operators to either a given relation or to the relations that are the result of applying operators to relations.

Example like the theta-join operator, or something like $\pi_{A,B}(\sigma_{C>D}(R \times S))$ etc.

References

[1] J.D. Ullman and J. Widom. *A First Course in Database Systems*. Prentice Hall, 1997.