These lecture notes include some material from Professors Bertossi, Kolaitis, Guagliardo and Libkin

Relational Algebra 1

Lecture Handout

Dr Eugenia Ternovska

Simon Fraser University

In this part

We will briefly review some basic notions of Discrete Mathematics,

and then

introduce the main operations of Relational Algebra (RA)

Relational algebra

As we discussed earlier, the main feature of relational algebra is that strikes a good balance between expressive power and efficiency.

Codd's key contribution was to identify a small set of basic operations on relations and to demonstrate that useful and interesting queries can be expressed by combining these operations.

- Thus, relational algebra is a rich enough language, even though, as we will see later on, it suffers from certain limitations in terms of expressive power.

Already the very first RDBMS prototype implementations (System R and Ingres) demonstrated that the relational algebra operations can be implemented efficiently.

Basic Notions from Discrete Mathematics

A **Domain** is a finite set of objects, e.g., the set of people in our class, the set of all possible grades

A **k-tuple** is an ordered sequence of k objects (need not be distinct and can be from different domains)
$(2, 0, 1)$ is a 3-tuple; (a, b, a, a, c) is a 5-tuple, and so on.

If $D_1, D_2, \ldots, D_k$ are finite sets (domains), then
the **cartesian product** $D_1 \times D_2 \cdots \times D_k$ of these sets is the set of all k-tuples $(d_1, d_2, \ldots, d_k)$ such that $d_i \in D_i$, for $1 \leq i \leq k$.

Basic Notions from Discrete Mathematics

Example

If $D_1 = \{0, 1\}$ and $D_2 = \{a, b, c, d\}$, then
$D_1 \times D_2 = \{(0, a), (0, b), (0, c), (0, d), (1, a), \ldots\}$
(usually written as a table)

Warning: Computing Cartesian products is an expensive operation!

Fact: Let $|D|$ denote the cardinality ($=$ # of elements) of a set D.
Then $|D_1 \times D_2 \times \cdots \times D_k| = |D_1| \times |D_2| \times \cdots \times |D_k|$.

In the example above, $|D_1| \times |D_2| = 8$.

Basic Notions from Discrete Mathematics

A k-ary **relation** R is a subset of a cartesian product of k sets, i.e.,
$R \subseteq D_1 \times D_2 \times \cdots \times D_k$.

Example

Unary $R = \{0, 2, 4, \ldots, 100\}$ $(R \subseteq D)$
Binary $T = \{(a, b) : a$ and b have the same birthday$\}$
Ternary $S = \{(m, n, s) : s = m + n\}$
...

In Relational Data Model, relations are recorded as tables that have names for their columns, called **attributes**

Example of a table

Customer

CustID	Name	City	Age
cust1	Renton	Edinburgh	24
cust2	Watson	London	32
cust3	Holmes	London	35

What are the attributes in this table?

What are the tuples in this table?

The Basic Operations of Relational Algebra

Group I: Three standard set-theoretic binary operations:

- Union
- Difference
- Cartesian Product.

Group II. Two unary operations on relations:

- Projection
- Selection.

Group III. One special operation:

- Renaming

Relational Algebra consists of all expressions obtained by combining these basic operations in syntactically correct ways.

Relational algebra

Procedural query language

A relational algebra expression

- takes as input one or more relations
- applies a **sequence of operations**
- returns a relation as output

Operations:

Projection	(π)	Union	$(\cup)$
Selection	(σ)	Intersection	$(\cap)$
Product	$(\times)$	Difference	$(-)$
Renaming	(ρ)		

The application of each operation results in a new relation that can be used as input to other operations

Projection

- **Vertical operation**: choose some of the **columns**
- Syntax: $\pi_{\text{sequence of attributes}}(\text{relation})$
- $\pi_{A_1,\ldots,A_n}(R)$ takes only the values of attributes $A_1, \ldots, A_n$ for each tuple in R

Customer

CustID	Name	City	Address
cust1	Renton	Edinburgh	2 Wellington Pl
cust2	Watson	London	221B Baker St
cust3	Holmes	London	221B Baker St

$\pi_{\text{Name,City}}(\text{Customer})$

Name	City
Renton	Edinburgh
Watson	London
Holmes	London

Selection

- **Horizontal operation**: choose **rows** satisfying some condition
- Syntax: σ_{Θ}(relation), where Θ is a condition
- A *family* of unary operations, one for each condition Θ
- $\sigma_{\theta}(R)$ takes only the tuples in R for which θ is satisfied

$$\begin{aligned} \text{term} &:= \text{attribute} \mid \text{constant} \\ \theta &:= \text{term } \mathbf{op} \text{ term with } \mathbf{op} \in \{=, \neq, >, <, \geqslant, \leqslant\} \\ &\quad \mid \theta \wedge \theta \mid \theta \vee \theta \mid \neg\theta \end{aligned}$$

Example of selection

Customer

CustID	Name	City	Age
cust1	Renton	Edinburgh	24
cust2	Watson	London	32
cust3	Holmes	London	35

$\sigma_{\text{City} \neq \text{'Edinburgh'} \wedge \text{Age} < 33}$(Customer)

CustID	Name	City	Age
cust2	Watson	London	32

More on the Selection Operator

Note: The use of the comparison operators $<$, $>$, $\leq$, $\geq$ assumes that the underlying domain of values is **totally ordered**.

If the domain is not totally ordered, then only $=$ and $\neq$ are allowed.

If we do not have attribute names (hence, we can only reference columns via their column number), then we need to have a special symbol, say \$, in front of a column number.

Thus, $\$4 > 100$ is a meaningful basic clause,
$\$1 =$"Apto" is a meaningful basic clause, and so on.

Combine Operations: Cartesian product

$R \times S$ **concatenates** each tuple of R with all the tuples of S

Example

R	A	B
	1	2
	3	4

$\times$

S	C	D
	1	a
	2	b
	3	c

$=$

$R \times S$	A	B	C	D
	1	2	1	a
	1	2	2	b
	1	2	3	c
	3	4	1	a
	3	4	2	b
	3	4	3	c

Note: all attributes must be different
If attribute A in common, R.A and S.A used to disambiguate

Expensive operation:

- $\text{card}(R \times S) = \text{card}(R) \times \text{card}(S)$
- $\text{arity}(R \times S) = \text{arity}(R) + \text{arity}(S)$

Resulting relation schema is the union of R and S schemas

Join

Combining Cartesian product and selection

Customer: ID, Name, City, Address
Account: Number, Branch, CustID, Balance

We can join customers with the accounts they own as follows

$$\sigma_{\text{ID}=\text{CustID}}(\text{Customer} \times \text{Account})$$

Renaming

Gives a new name to some of the attributes of a relation

Syntax: $\rho_{\text{replacements}}(\text{relation})$,
where a replacement has the form $C \to D$, $A \to A'$, etc.

$$\rho_{A\to A', C\to D}\left(\begin{array}{ccc} \textbf{A} & \textbf{B} & \textbf{C} \\ \hline \text{a} & \text{b} & \text{c} \\ 1 & 2 & 3 \end{array}\right) = \begin{array}{ccc} \textbf{A}' & \textbf{B} & \textbf{D} \\ \hline \text{a} & \text{b} & \text{c} \\ 1 & 2 & 3 \end{array}$$

Example

Customer: **CustID**, Name, City, Address
Account: Number, Branch, **CustID**, Balance

$$\sigma_{\text{CustID}=\text{CustID}'}(\text{Customer} \times \rho_{\text{CustID}\to\text{CustID}'}(\text{Account}))$$

Set operations

Union

R	A	B
	a1	b1
	a2	b2

$\cup$

S	A	B
	a1	b1
	a3	b3

$=$

R ∪ S	A	B
	a1	b1
	a2	b2
	a3	b3

Intersection

R	A	B
	a1	b1
	a2	b2

$\cap$

S	A	B
	a1	b1
	a3	b3

$=$

R ∩ S	A	B
	a1	b1

Difference

R	A	B
	a1	b1
	a2	b2

$-$

S	A	B
	a1	b1
	a3	b3

$=$

R − S	A	B
	a2	b2

The relations must have the same set of attributes

Union and renaming

R	Father	Child
	George	Elizabeth
	Philip	Charles
	Charles	William

S	Mother	Child
	Elizabeth	Charles
	Elizabeth	Andrew

We want to find the relation **parent-child**

$\rho_{\text{Father}\rightarrow\text{Parent}}(R) \cup \rho_{\text{Mother}\rightarrow\text{Parent}}(S) \quad =$

Parent	Child
George	Elizabeth
Philip	Charles
Charles	William
Elizabeth	Charles
Elizabeth	Andrew

Full relational algebra

Primitive operations: π , σ , $\times$, ρ , $\cup$, $-$

Removing any of these results in a **loss of expressive power**

Schemas For Results

• Union, intersection, and difference: The schemas of the two operands must be the same, so use that schema for the result

• Selection: Schema of the result is the same as the schema of the operand

• Projection: List of projected attributes tells us the new schema

• Renaming: The operator tells the schema

• Product: Schema is the attributes of both relations
(use R.A, P.A, etc., to distinguish two attributes named A)

Acknowledgements

[1] Database Systems: The Complete Book, 2nd EditionHector Garcia-Molina, Jeffrey D. Ullman, Jennifer WidomPrentice Hall, 2009

[2] Database System Concepts, Seventh EditionAvi Silberschatz, Henry F. Korth, S. SudarshanMcGraw-Hill, March 2019www.db-book.com

Additional references and resources used in preparation of this course are listed on the course webpage or mentioned in slides.