Relational Algebra

- a collection of operations for building tables that constitute answers to queries
- an abstract language

Fundamental Relational Algebra Operations

- *set-theoretic*: regard relations as sets of tuples
- *native relational*: focus on the structure of tuples

Set-theoretic operations:

Name	Symbol
Union	$\cup$
Intersection	$\cap$
Difference	$-$
(Cartesian) Product	$\times$

Native relational operations:

Name	Symbol
Projection	π or `R[ ]`
Selection	σ or `R where C`
Join	$\bowtie$
Division	$\div$

Set-theoretic Operations:

It only makes sense to apply set-theoretic operations to relations representing "compatible" sets.

Definition 1 Relations R and S are *compatible* if they have the same schema, i.e., $Head(R) = Head(S)$.

Definition 2 If R and S are compatible relations with $Head(R) = Head(S) = A_1, \ldots, A_n$, then

- the *union*, $R \cup S$, of R and S is a relation with the same schema as R, S that consists of all tuples that are in R or in S or in both.
- the *intersection*, $R \cap S$, of R and S is a relation with the same schema as R, S that consists of all tuples that are in both R and S.
- the *difference*, $R - S$, of R and S is a relation with the same schema as R, S that consists of all tuples that are in R but not in S.

Set-theoretic operations can be applied recursively: operands may be the result of other operations.

Example

R

A	B	C
a_1	b_1	c_1
a_1	b_2	c_3
a_2	b_1	c_2

S

A	B	C
a_1	b_1	c_1
a_1	b_1	c_2
a_1	b_2	c_3
a_3	b_2	c_3

$R \cup S$

A	B	C
a_1	b_1	c_1
a_1	b_1	c_2
a_1	b_2	c_3
a_2	b_1	c_2
a_3	b_2	c_3

$R \cap S$

A	B	C
a_1	b_1	c_1
a_1	b_2	c_3

$R - S$

A	B	C
a_2	b_1	c_2

$S - R$

A	B	C
a_1	b_1	c_2
a_3	b_2	c_3

- Keeping intermediate results of algebraic expressions:

  Definition 3 Let R be a relation with schema $A_1, \ldots, A_n$ and let $B_1, \ldots, B_n$ be attributes such that $Dom(B_i) = Dom(A_i)$, $i = 1, \ldots, n$. A new relation S with schema $B_1, \ldots, B_n$ is defined by the assignment statement $S(B_1, \ldots, B_n)$:=R$(A_1, \ldots, A_n)$. The content of S is the same as the content of R:
  $$u \in S \ iff \ \exists t \in R \ u[B_i] = t[A_i], i = 1, \ldots, n$$

- When attributes of S are identical to those of R, S is called an *alias* of R. The assignment is written as $S := R$.

Example: $T := (R \cup S) - (R \cap S)$
$(T_1 := R \cup S, T_2 := R \cap S, T := T_1 - T_2)$

R

A	B	C
a_1	b_1	c_1
a_1	b_2	c_3
a_2	b_1	c_2

S

A	B	C
a_1	b_1	c_1
a_1	b_1	c_2
a_1	b_2	c_3
a_3	b_2	c_3

T_1

A	B	C
a_1	b_1	c_1
a_1	b_1	c_2
a_1	b_2	c_3
a_2	b_1	c_2
a_3	b_2	c_3

T_2

A	B	C
a_1	b_1	c_1
a_1	b_2	c_3

T

A	B	C
a_1	b_1	c_2
a_2	b_1	c_2
a_3	b_2	c_3

Definition 4 The *Cartesian product* of relations $R(A_1, \ldots, A_n)$ and $S(B_1, \ldots, B_m)$ is a relation T with schema $R.A_1, \ldots, R.A_n, S.B_1, \ldots, S.B_m$. T contains all possible associations of tuples in R and S: if $r \in R$ and $s \in S$, then the *concatenation*, $r \| s$, of r and s is a tuple in T. For every pair of tuples r, s in R, S respectively, there exists a tuple t in $R \times S$ such that $t(R.A_i) = r(A_i), i = 1, \ldots, n$ and $t(S.B_j) = s(B_j), j = 1, \ldots, m$.

Example

R

A	B	C
a_1	b_1	c_1
a_1	b_2	c_3
a_2	b_1	c_2

S

B	C	D
b_1	c_1	d_1
b_1	c_1	d_3
b_2	c_2	d_2
b_1	c_2	d_4

$R \times S$

R.A	R.B	R.C	S.B	S.C	S.D
a_1	b_1	c_1	b_1	c_1	d_1
a_1	b_1	c_1	b_1	c_1	d_3
a_1	b_1	c_1	b_2	c_2	d_2
a_1	b_1	c_1	b_1	c_2	d_4
a_1	b_2	c_3	b_1	c_1	d_1
a_1	b_2	c_3	b_1	c_1	d_3
a_1	b_2	c_3	b_2	c_2	d_2
a_1	b_2	c_3	b_1	c_2	d_4
a_2	b_1	c_2	b_1	c_1	d_1
a_2	b_1	c_2	b_1	c_1	d_3
a_2	b_1	c_2	b_2	c_2	d_2
a_2	b_1	c_2	b_1	c_2	d_4

Native Relational operations

- assume relations R and S have schemas $A_1, \ldots, A_n$ and $B_1, \ldots, B_m$ respectively

- **Definition 5** The *projection*, $R[A_{i_1}, \ldots, A_{i_k}]$ ($\pi_{A_{i_1},\ldots,A_{i_k}}(R)$), of relation R on attributes $A_{i_1}, \ldots, A_{i_k}$, where $\{A_{i_1}, \ldots, A_{i_k}\} \subseteq \{A_1, \ldots, A_n\}$, is a relation with schema $A_{i_1}, \ldots, A_{i_k}$. For every tuple $r \in R$ there exists a single tuple $t \in \pi_{A_{i_1},\ldots,A_{i_k}}(R)$ such that $r[A_{i_j}] = t[A_{i_j}]$ for every $A_{i_j} \in \{A_{i_1}, \ldots, A_{i_k}\}$.

  Note: distinct rows of relations may become identical when projected onto a subset of the attributes. Duplicate tuples in the relation resulting from a projection operation are eliminated.

  Example `CN:= CUSTOMERS[cname]`, `CND:= CUSTOMERS[cname,city]`

  `CUSTOMERS`

`cid`	`cname`	`city`	`discnt`
c001	TipTop	Duluth	10.00
c002	Basics	Dallas	12.00
c003	Allied	Dallas	8.00
c004	ACME	Duluth	8.00
c006	ACME	Kyoto	0.00

  `CN`

`cname`
TipTop
Basics
Allied
ACME

  `CND`

`cname`	`city`
TipTop	Duluth
Basics	Dallas
Allied	Dallas
ACME	Duluth
ACME	Kyoto

- **Definition 6** The *selection*, R *where* C ($\sigma_C(R)$), is a relation with the same schema as R that contains those tuples of R that obey the *selection condition* C. C can be of the form:

  1. $A_i\ \theta\ A_j$ or $A_i\ \theta\ c$, where A_i, A_j have the same domain, c is a constant and $\theta \in \{<, >, \leq, \geq, =, \neq\}$.
  2. If C, C' are conditions, then so are $C \wedge C', C \vee C', \neg C$. If $U := \sigma_{C_1}(S)$ and $V := \sigma_{C_2}(S)$, then $\sigma_{C_1 \wedge C_2}(S) = U \cap V$, $\sigma_{C_1 \vee C_2}(S) = U \cup V$ and $\sigma_{\neg C_1}(S) = S - U$.

  Example:

  1. Find all customers based in Kyoto.

     `CUSTOMERS where city="Kyoto"`

     or $\sigma_{city="Kyoto"}(CUSTOMERS)$

cid	cname	city	discnt
c006	ACME	Kyoto	0.00

  2. Find the products stored in Dallas that cost more than \$0.50.

     `PRODUCTS where city="Dallas" and price > 0.50`

     or $\sigma_{city="Dallas" \wedge price>0.50}(PRODUCTS)$

pid	pname	city	quantity	price
p05	pencil	Dallas	221400	1.00
p06	folder	Dallas	123100	2.00

3. Find all agents who have a commission percentage of 6% or more.

   `L := AGENTS where percent >= 6`

   or $L := \sigma_{percent \geq 6}(AGENTS)$

   L

aid	aname	city	percent
a01	Smith	New York	6
a02	Jones	Newark	6
a03	Brown	Tokyo	7
a04	Gray	New York	6

4. Find all pairs of agents that are such that, both have a commission percentage of at least 6% and both are stationed in the same city.

   `L := AGENTS where percent >= 6`

   `M := AGENTS where percent >= 6`

   `PAIRS:= (L x M) where L.city = M.city`

   or `PAIRS:=` $\sigma_{L.city=M.city}(L \times M)$

   PAIRS:

L.aid	L.aname	L.city	L.%	M.aid	M.aname	M.city	M.%
a01	Smith	New York	6	a01	Smith	New York	6
a01	Smith	New York	6	a04	Gray	New York	6
a04	Gray	New York	6	a01	Smith	New York	6
a04	Gray	New York	6	a04	Gray	New York	6

The answer to the query contains redundant information. To retrieve distinct pairs of distinct **aid** values we need a more restrictive expression:

`PAIRS2:= (L x M) where L.city = M.city and L.aid < M.aid`

or `PAIRS2:=` $\sigma_{L.city=M.city \wedge L.aid<M.aid}(L \times M)$

PAIRS2:

L.aid	L.aname	L.city	L.%	M.aid	M.aname	M.city	M.%
a01	Smith	New York	6	a04	Gray	New York	6

- Find all pairs of agents that are such that, both have a commission percentage of at least 6% and both are stationed in New York.

`PAIRS3:= (L x M) where L.city = "New York" and M.city = "New York" and L.aid < M.aid`

The same answer would be obtained. However, the two queries are not in general equivalent: the latter is content dependent.

They give different answers if the relation `L` is changed to

aid	aname	city	percent
a01	Smith	New York	6
a02	Jones	Newark	6
a03	Brown	Tokyo	7
a04	Gray	New York	6
a07	Green	Newark	7

Precedence of Operations

- also called *binding strength*
- determines which operations are performed first in a relational algebra expression with no parentheses
- parentheses override precedence of operations: sub-expressions included in parentheses are evaluated first

Precedence	Operation	Symbol
Highest	projection	π
$\vert$	selection	σ
$\vert$	product	$\times$
$\vert$	join, division	$\bowtie, \div$
$\downarrow$	difference	$-$
Lowest	union, intersection	$\cup, \cap$

Example: Find the cities in which there exist customers with a discount of 10% or less or agents with a commission percentage of 6% or less.

$$\pi_{city}(\sigma_{discnt \leq 10}(CUSTOMERS)) \cup \pi_{city}(\sigma_{percent \leq 6}(AGENTS))$$

If parentheses are omitted:

$$\pi_{city}\sigma_{discnt \leq 10}(CUSTOMERS) \ \cup \ \pi_{city}\sigma_{percent \leq 6}(AGENTS)$$

...not the intended answer!

- **Join:** combines tuples of relations that have equal values in identical attributes (*natural join* or *equijoin*).

Definition 7 Let R, S be relations with schemas $A_1, \ldots, A_n, B_1, \ldots, B_k$ and $B_1, \ldots, B_k, C_1, \ldots, C_m, n, k, m \geq 0$, respectively. The *join*, $R \bowtie S$, of R and S is a relation with schema $A_1, \ldots, A_n, B_1, \ldots, B_k, C_1, \ldots, C_m$. Tuple t is in $R \bowtie S$ iff there exist tuples u and v in R and S respectively, such that $u[B_j] = v[B_j], j = 1, \ldots, k$; moreover, $t[A_i] = u[A_i], i = 1, \ldots, n, t[B_i] = u[B_i] = v[B_i], i = 1, \ldots, k$ and $t[C_i] = v[C_i], i = 1, \ldots, m$.

Note: the order in which the attributes B_i appear is not important.

Example: $T := R \bowtie S$

R

A	B_1	B_2
a_1	b_1	b'_1
a_1	b_2	b'_1
a_2	b_1	b'_2

S

B_1	B_2	C
b_1	b'_1	c_1
b_1	b'_1	c_2
b_1	b'_2	c_3
b_2	b'_2	c_4

T

A	B_1	B_2	C
a_1	b_1	b'_1	c_1
a_1	b_1	b'_1	c_2
a_2	b_1	b'_2	c_3

Special cases:

1. $\{B_1, \ldots, B_k\} = \emptyset$, i.e., R and S have no attributes in common. Then, $R \bowtie S = R \times S$.
2. $head(R) = head(S)$, i.e., the relations are compatible. Then, $R \bowtie S = R \cap S$.

Example: Find the names of the customers who have ordered product p01.

CUSTOMERS

cid	cname	city	discnt

PRODUCTS

pid	pname	city	qtty	price

ORDERS

ordno	month	cid	aid	pid	qty	dollars

1. CUSTORDS:=CUSTOMERS $\bowtie$ ORDERS

cname	city	discnt	ordno	month	cid	aid	pid	qty	dollars
TipTop	Duluth	10.0	1011	Jan	c001	a01	p01	1000	450.00
TipTop	Duluth	10.0	1012	Jan	c001	a01	p01	1000	450.00
TipTop	Duluth	10.0	1019	Feb	c001	a02	p02	400	180.00
TipTop	Duluth	10.0	1018	Feb	c001	a03	p04	600	540.00
TipTop	Duluth	10.0	1023	Mar	c001	a04	p05	500	450.00
TipTop	Duluth	10.0	1022	Mar	c001	a05	p06	400	720.00
TipTop	Duluth	10.0	1025	Apr	c001	a05	p07	800	720.00
Basics	Dallas	12.0	1013	Jan	c002	a03	p03	1000	880.00
Basics	Dallas	12.0	1026	May	c002	a05	p03	800	704.00
Allied	Dallas	8.0	1015	Jan	c003	a03	p05	1200	1104.00
Allied	Dallas	8.0	1014	Jan	c003	a03	p05	1200	1104.00
ACME	Duluth	8.0	1021	Feb	c004	a06	p01	1000	460.00
ACME	Kyoto	0.0	1016	Jan	c006	a01	p01	1000	500.00
ACME	Kyoto	0.0	1020	Feb	c006	a03	p07	600	600.00
ACME	Kyoto	0.0	1024	Mar	c006	a06	p01	800	800.00

2. CP01:= $\sigma_{pid=p01}$(CUSTORDS)

cname	city	discnt	ordno	month	cid	aid	pid	qty	dollars
TipTop	Duluth	10.0	1011	Jan	c001	a01	p01	1000	450.00
TipTop	Duluth	10.0	1012	Jan	c001	a01	p01	1000	450.00
ACME	Duluth	8.0	1021	Feb	c004	a06	p01	1000	460.00
ACME	Kyoto	0.0	1016	Jan	c006	a01	p01	1000	500.00
ACME	Kyoto	0.0	1024	Mar	c006	a06	p01	800	800.00

3. CNP01:= π_{cname}(CP01)

cname
TipTop
ACME

Example: Find the names of customers who order at least one product costing \$0.50.

CHEAPS:= $\pi_{pid}(\sigma_{price=0.50}(\texttt{PRODUCTS}))$

CN:= $\pi_{cname}((\texttt{ORDERS} \bowtie \texttt{CHEAPS}) \bowtie \texttt{CUSTOMERS})$

The expression
$\pi_{cname}((\texttt{ORDERS} \bowtie \sigma_{price=0.50}(\texttt{PRODUCTS})) \bowtie \texttt{CUSTOMERS})$ is not a correct answer.

Properties:

1. $\bowtie$ and $\times$ are associative:
   - $(R \times S) \times T = R \times (S \times T)$
   - $(R \bowtie S) \bowtie T = R \bowtie (S \bowtie T)$
2. $\bowtie$ and $\times$ are commutative:
   - $R \times S = S \times R$
   - $R \bowtie S = S \bowtie R$