University of Southampton

Relational Algebra

COMP3211 Advanced Databases

Dr Heather Packer – hp3@ecs.soton.ac.uk

What is a Relational Algebra?

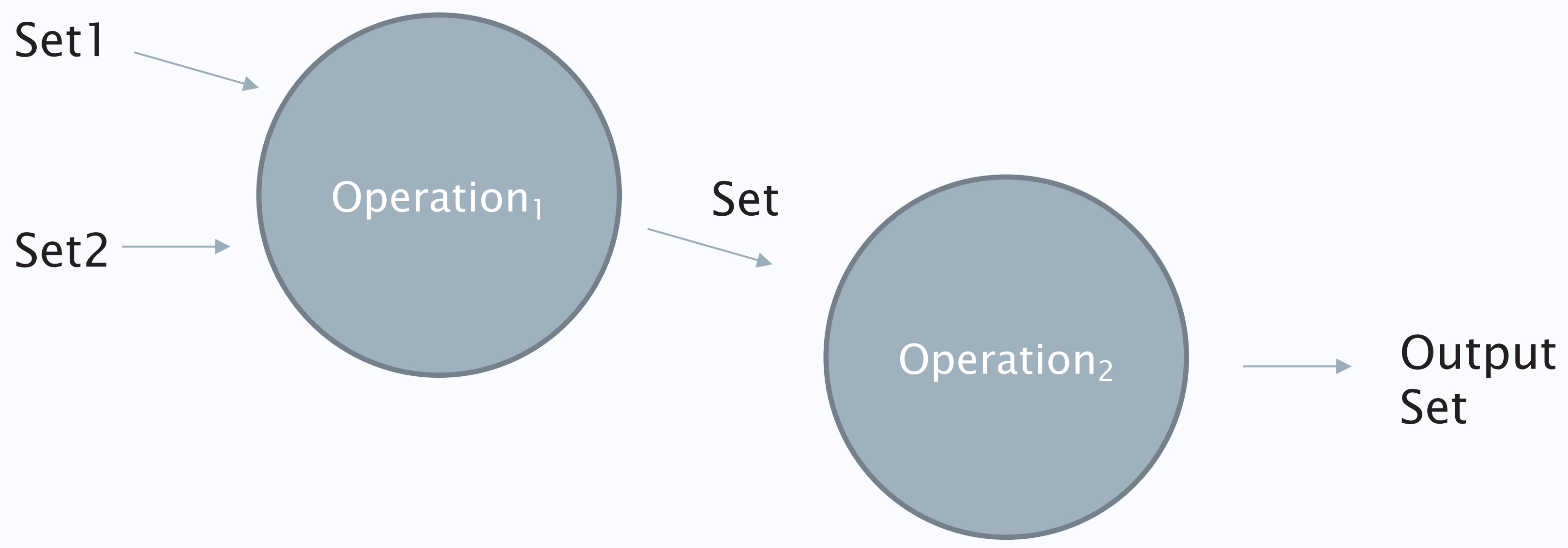
- An Algebra is a mathematical system consisting of:
  - Operands – variables or values from which new values can be constructed
  - Operators – symbols denoting procedures that construct new values from the given values
- A Relational Algebra
  - Operands – relations or variables that represent relations
  - Operators – common things that can are performed on relations

Relational Algebra

- Set Operations
  - $\cup$ Set union
  - $\cap$ Set intersection
  - $-$ Set difference
  - X Cartesian product
  - $\div$ Set Division
- Relational Database Specific Operations
  - σ Selection
  - π Projection
  - $\bowtie$ Join
- Set Functions
  - sum
  - avg
  - count
  - any
  - max
  - min

Relational Algebra

- The input of a relational algebra operator is one or more relations
- The result of an operation is always a relation
- Necessary to use relational algebra in a cascaded manner



What is a relation?

Relations as Subset of Cartesian Products of Sets

Set S_1 (Student IDs)

68943
94433
78215
...

x

Set S_2 (Student Name)

Mary
John
William
...

=

S_1 x S_2 =

68943, Mary
68943, John
68943, William
94433, Mary
94433, John
94433, William
78215, Mary
78215, John
78215, William
...

Students $\subseteq S_1 \times S_2$

	ID	Name
tuple	68943	William
	94433	John
	78215	Mary

Relations as Subset of Cartesian Products of Sets

Set S_1 (Student IDs)

68943
94433
78215
…

Domain

x

Set S_2 (Student Name)

Mary
John
William
…

Domain

=

S_1 x S_2 =

68943, Mary
68943, John
68943, William
94433, Mary
94433, John
94433, William
78215, Mary
78215, John
78215, William
…

Students $\subseteq S_1$ x S_2

ID	Name
68943	William
94433	John
78215	Mary

Tuples and attributes

- $R \subseteq D_1 \times D_2 \times \ldots \times D_k$ is a set of k-tuples
- R can be represented in a table with k columns
- Values from domain D_i are the only values allowed in the i^{th} column
- The columns in relational data models have names called attributes

Students

ID	Name	DeptID
68943	William	ECS
94433	John	ECS

In the relation Students, attributes are: ID, Name, DeptID

Properties of Relations

A relation $R(A_1,\ldots,A_k)$ has the following properties:

1. Each row represents a k-tuple of R
2. The values of an attribute are all from the same domain
3. Each attribute of each tuple in a relation contains a single atomic value
4. The ordering of rows is immaterial (relations are just sets)
5. All rows are distinct (relations are just sets)
6. Named perspective: the semantics of each column is conveyed its name
7. Named perspective: the ordering of the attributes is not significant
8. Unnamed perspective: the ordering of attributes is significant (we access columns by their positions)

Relational Operators

Relational Algebra

- Set Operations
  - ∪ Set union
  - \- Set difference
  - X Cartesian product
  - ∩ Set intersection
- Relational Specific Operations
  - ρ Renaming
  - σ Selection
  - π Projection
  - ⋈ Join

Set Operations

Union

- The union of two relations is the set of tuples where each tuple appears in either set

$$R \cup S$$

B	C
1	2
3	4

$\cup$

B	C
4	4
5	6

=

B	C
1	2
3	4
4	4
5	6

- $R \cup S = \{(a_1,\ldots,a_k): (a_1,\ldots,a_k) \text{ is in } R \text{ or } (a_1,\ldots,a_k) \text{ is in } S\}$
- Note: no duplicate tuples

Union

- The Union operation must work on relations with the same attributes

R

B	C
1	2
3	4

∪

S

D	E
4	4
5	6

= ?

- If relations have different attributes, we can rename the attributes of S by using the renaming operator

Union

- Some versions of the algebra allow the attributes of R and S to have different names

R

B	C
1	2
3	4

∪

S

D	E
4	4
5	6

=

R ∪ S

F	G
1	2
3	4
4	4
5	6

- The columns in the result are given new names

Difference

- The difference between R and S is all the tuples that appear in R but not in S

$$R - S$$

B	C
1	2
3	4

\-

B	C
3	4
5	6

=

B	C
1	2

- R - S = $\{(a_1,\ldots,a_k)$: $(a_1,\ldots,a_k)$ is in R and $(a_1,\ldots,a_k)$ is **not** in S$\}$

Cartesian Product

R X S

R

A	B
X	2
Y	3
Z	3

X

S

B	C
sda	1
bfd	0

=

R X S

A	R.B	S.B	C
X	2	sda	1
Y	3	sda	1
Z	3	sda	1
X	2	bfd	0
Y	3	bfd	0
Z	3	bfd	0

- Notice the clashing attribute names are renamed

Cartesian Product

- Cartesian Product of relations R X S, given R and S having a m-ary and n-ary relation respectively, is calculated by:
  - Every tuple of R is paired with every tuple of S
  - The tuples are concatenated together to give a (m+n)-ary tuple
- Note that the resulting size of the the relation is the product of the size of each relation.
  - The Cartesian Product can get large
- Concretely:

$$R \times S = \{(a_1,\ldots,a_m,b_1,\ldots,b_n) : (a_1,\ldots,a_m) \text{ is in R and } (b_1,\ldots,b_n) \text{ is in S}\}$$

$$|R \times S| = |R| \times |S|$$

Set Operation Rules

- Union
  - Commutativity **does** hold because order within a set is unimportant : R ∪ S = S ∪ R
  - Associativity **does** hold: R ∪ (S ∪ T) = (R ∪ S) ∪T
- Difference:
  - Commutativity **does not** hold for Difference: R - S ≠ S – R (except if R=S)
  - Associativity **does not** hold for Difference: R – (S –T) ≠ (R – S) – T
- Cartesian Product:
  - Note: Ordering of attributes is important here
  - Commutativity **does not** hold: R × S ≠ S × R
  - Associativity **does** hold: R × (S × T) = (R × S) × T
  - Distributivity across Union **does** hold: R × (S ∪ T) = (R × S) ∪ (R × T)

Derived Operations

- Some operations can be derived by other relational algebra operations
- Intersection:

$$R \cap S = R - (R - S)$$

R

B	C
1	2
3	4

-

S

B	C
3	4
5	6

=

R - S

B	C
1	2

R

B	C
1	2
3	4

-

R - S

B	C
1	2

=

R ∩ S

B	C
3	4

Relational Specific Operations

Renaming Operator ρ

- The renaming operator takes one relation and changes attribute names

$$\rho_{B\,/\,A}\,(S)$$

- Returns a relation with schema identical to S but the attribute name A has been replaced by B
- Rename more than one attributes using ‘,’ in the subscript
- E.g. let S with schema S(A, B, C, D, E)

$$\rho_{X/B,Y/D}\,(S)$$

- Creates a relation with schema S(A, X, C, Y, E)

Renaming Operator ρ Example

- If the relational algebra variant needs same attribute names for union
- Then use the renaming operator

R

B	C
1	2
3	4

$\cup$

S

D	E
4	4
5	6

$=$

Renaming Operator ρ Example

- If the relational algebra variant needs same attribute names for union
- Then use the renaming operator

$$\rho_{B/D,\ C/E}\ (S)$$

R

B	C
1	2
3	4

$\cup$

$\rho_{B/D,\ C/E}\ (S)$

B	C
4	4
5	6

=

$R \cup \rho_{B/D,\ C/E}\ (S)$

B	C
1	2
3	4
4	4
5	6

Projection Operator π

- The Projection Operator removes changes what attributes appear in the relation

$$\pi_{L}\,(R)$$

- It removes all columns whose attributes do not appear in the list *L*
- Columns may be re-arranged according to the order in the list
- Any duplicate rows are also eliminated

Projection Operator π Example

Sells

Shop	Food	Price	Units
Union	Apples	0.50	2
Union	Bananas	0.80	4
Co-op	Apples	0.50	5
Co-op	Peaches	0.75	3
Costa	Bananas	0.90	1
Costa	Peaches	1.10	1

$\pi_{\text{Food,Price}}(\text{Sells})$

Food	Price
Apples	0.50
Bananas	0.80
Peaches	0.75
Bananas	0.90
Peaches	1.10

Note: only one entry for apples

Extended Projection

- Some algebras extend projection to allow arbitrary expressions involving attributes

$$\pi_{B+C \rightarrow A}(R)$$

- Arithmetic on attributes eg B + C → A
- Allows duplicate occurrences of same attribute eg:

$$\pi_{B,B}\left(\begin{array}{|c|c|}\hline B & C \\ \hline 1 & 2 \\ 3 & 4 \\ \hline\end{array}\right) = \begin{array}{|c|c|}\hline B1 & B2 \\ \hline 1 & 1 \\ 3 & 3 \\ \hline\end{array}$$

Extended Projection Example

Sells

Shop	Food	Price	Units
Union	Apples	0.50	2
Union	Bananas	0.80	4
Co-op	Apples	0.50	5
Co-op	Peaches	0.75	3
Costa	Bananas	0.90	1
Costa	Peaches	1.10	1

$\pi_{\text{Food, Price/Units} \rightarrow \text{Cost}}$ (Sells)

Food	Cost
Apples	0.25
Bananas	0.20
Apples	0.10
Peaches	0.25
Bananas	0.90
Peaches	1.10

Selection Operator

- The Selection Operator returns a subset of the relation where the tuples satisfy a predicate

$$\sigma_{\Theta}(R)$$

- R is a relation and Θ is a condition or predicate
- The condition Θ is an expression built from:
  - Comparison operators $=, <, >, \neq, \leq, \geq$ applied to operands that are constants or attribute names (or positions)
  - The Boolean logic operators $\wedge, \vee, \neg$ applied to basic clauses.

Selection Operator σ Predicates

- Example predicates:
  - Price > 0.70
  - Shop = “Co-op”
  - (Shop = “Co-op”) $\wedge$ (Units > 1)
- We reference columns via attribute names or via their “component number’’ (position) by using $
  - Shop = “Co-op”
  - $1 = “Co-op”
  - $4 > 1

Selection Operator σ Example

Sells

Shop	Food	Price	Units
Union	Apples	0.50	2
Union	Bananas	0.80	4
Co-op	Apples	0.50	5
Co-op	Peaches	0.75	3
Costa	Bananas	0.90	1
Costa	Peaches	1.10	1

$\sigma_{Food\ =\ \text{“Bananas”}}(\sigma_{Units>1}(Sells))$

Shop	Food	Price	Units
Union	Bananas	0.80	4

This is equivalent to

$\sigma_{Food=\text{“Bananas”}\ \land\ Units>1}\ (Sells)$

Algebraic Laws for the Selection Operation

- Algebraic laws can be useful in query optimization
  - $\sigma_{\Theta 1}(\sigma_{\Theta 2}(R)) = \sigma_{\Theta 1 \wedge \Theta 2}(R)$
  - Commutative: $\sigma_{\Theta 1}(\sigma_{\Theta 2}(R)) = \sigma_{\Theta 2}(\sigma_{\Theta 1}(R))$
  - $\sigma_{\Theta}(R \times S) = \sigma_{\Theta}(R) \times S$
    - if Θ mentions only attributes of R
    - Cartesian product is expensive so reducing the size of R is beneficial

Sets vs. Multisets

University of Southampton

Sets vs. Multisets (Bags)

- A Multiset or Bag is like a Set but elements may appear more than once
- Example:
  - R = {1,3,3,6,6,6,7}, S ={1,1,6,6,7} are both multisets
- Union of two multisets:
  - {1,3,3,6,6,6,7} ∪ {1,1,6,6,7} = {1,1,1,3,3,6,6,6,6,6,7,7}
- Difference between two multisets:
  - {1,3,3,6,6,6,7} - {1,1,6,6,7} = {3, 3, 6}
- Cartesian product:
  - {1, 3, 3} × {1, 1, 6} = {<1,1>, <1,1>, <1,6>, <3,1>, <3,1>, <3,6>, <3,1>, <3,1>, <3,6>}

Operations on Multisets (Bags)

- Given $\mu(x, B)$, defined as the number of occurrences of x in multiset B
- Union $R \cup S$
  - $\mu(t, R \cup S) = \mu(t, R) + \mu(t, S)$ for all t in R and S
- Difference $R - S$
  - $\mu(t, R - S) = \max\{\mu(t, R) - \mu(t, S), 0\}$ for all t in R and S
- Intersection $R \cap S$
  - $\mu(t, R \cap S) = \min\{\mu(t, R), \mu(t, S)\}$ for all t in R and S
- Cartesian Product:
  - $\mu(tt', R \times S) = \mu(t, R) * \mu(t', S)$ for all t in R and t' in S

Operations on Multisets (Bags)

- Projection:
  - If R is a multiset, then $\pi_X(R)$ is also a multiset
  - If R is a set, then $\pi_X(R)$ **may be** a multiset
- Selection:
  - If R is a multiset, then the selection $\sigma_\Theta(R)$ may be a set
  - If R is a set, then the selection $\sigma_\Theta(R)$ is a set

Multisets in SQL

- SQL is multiset based
- Efficiency:
  - Duplicate elimination may take quadratic time
    - For example, after a projection
- Necessity:
  - Eliminating duplicates might result in information loss/errors (e.g., in computing averages)
- How to eliminate duplicates in SQL? Use the DISTINCT keyword

SELECT DISTINCT <attribute list>

FROM <relation list>

Next Lecture:
Relational Algebra 2