Algebraic and Logical Query Languages

Thomas Schwarz, SJ

Bags, Lists, Sets

- Bags are multi-sets
  - An element can appear more than once
- They are not sets
  - In a set, each element can appear at most once
- They are not lists
  - In a list, elements are indexed

Bags, Lists, Sets

- Why bags:
  - Union, seletion and projection can create the same tuple many times
  - Removing duplicates is difficult:
    - Either use a hash table or use sorting
      - Both of which are expensive in different ways

Bags, Lists, Sets

- Why bags:
  - For some temporary tables, bags are appropriate
    - Aggregation query like find the average salaries of all female employees hired in 2010, 2011, 2012
    - Form a temporary table with salary as only attribute
    - You <u>need</u> to keep values separate

Union, Intersection, Differences of Bags

- Union:
    - Just concatenate the two bags
        - If an element appears twice in one bag and thrice in the other, it will appear five times in the union

Union, Intersection, Differences of Bags

- Intersection
  - $R \cap S$:
    - Bags match each tuple with another tuple
      - If a tuple appears n times in R and m times in S, then it appears $\min(m, n)$ times in $R \cap S$.

Union, Intersection, Differences of Bags

- Difference:
    - Again, bags use one-to-one matching
        - Tuple appears n times in R
        - Tuple appears m times in S
        - Tuple appears $\max(0, n - m)$ times in $R - S$.
            - Each occurrence in S cancels out a single appearance in R

Union, Intersection, Differences of Bags

- In short: bags are different from sets

Projection of Bags

- Projection of bags:
  - Each tuple in the mother relation gives rise to one tuple in the projection

Selection of Bags

- Again: selection condition is applied to each tuple
- There is no duplicate elimination

Products of Bags

- Recall: Product assumes that attribute sets are different
- Each tuple of R is paired with each tuple of S

Joins of Bags

- Join tuple by tuple

Joins of Bags

- Example:

R

A	B
1	2
1	2

S

B	C
2	3
4	5
4	5

$R \times S$

$R \bowtie S$

Joins of Bags

- Example:

R

A	B
1	2
1	2

S

B	C
2	3
4	5
4	5

$R \times S$

A	R.B	S.B	C
1	2	2	3
1	2	4	5
1	2	4	5
1	2	2	3
1	2	4	5
1	2	4	5

$R \bowtie S$

A	B	C
1	2	3
1	2	3

Joins of Bags

R

A	B
1	2
1	2

S

B	C
2	3
4	5
4	5

$R \bowtie_{R.B<S.B} S$

A	R.B	S.B	C
1	2	2	3
1	2	4	5
1	2	4	5
1	2	2	3
1	2	4	5
1	2	4	5

Joins of Bags

R

A	B
1	2
1	2

S

B	C
2	3
4	5
4	5

$R \bowtie_{R.B<S.B} S$

A	R.B	S.B	C
1	2	4	5
1	2	4	5
1	2	4	5
1	2	4	5

Relational Algebra Operators

- Deduplication operator δ
- Aggregation operators such as sum, averages are used by grouping operators
- Grouping: Partitions tuples into groups
  - Usually, aggregation is then applied to each group
- Extended projections
  - Allow to create new attributes using arithmetic operations
- Sorting operator
- Outer join operator

Aggregations

- SUM
- AVG
- MIN, MAX
- COUNT
    - not necessarily distinct values in a column

Aggregation

- Example:
  - Find the aggregations of this table

A	B
1	2
3	4
1	2
1	2

Aggregation

- Example:
  - Find the aggregations of this table

A	B
1	2
3	4
1	2
1	2

```
SUM(A)   = 6     SUM(B)   = 10
AVG(A)   = 1.5   AVG(B)   = 2
MIN(A)   = 1     MIN(B)   = 2
MAX(A)   = 3     MAX(B)   = 4
COUNT(A) = 4     COUNT(B) = 4
```

Grouping

- Find the length of all movies produced by a certain studio
- Project onto studio, length
- Group by studioName

studioName	movieLength
Disney	89
Disney	103
Disney	132
Disney	76
MGM	89
MGM	103
MGM	89

Grouping

- Find the length of all movies produced by a certain studio
- Project onto studio, length
- Group by studioName
- Aggregate on movieLength

studioName	movieLength
Disney	1493
MGM	3981

Grouping Operator

- $\gamma_{\text{op}(A)}(R)$
  - A — the ***<u>grouping attribute</u>***
  - op — the aggregation operator (e.g. AVG)
  - R — the relation

Grouping operator

- $\gamma_{\text{op}(A)}(R)$
  - Partition the tuples of R into groups according to values of A
  - For each group produce one tuple with
    - the grouping attributes' values for that group
    - the aggregation over all tuples of that group
- Generalize to several attributes

Grouping Operator

- Find all stars that appeared in at least three movies and the earliest year in which they appeared
- $\gamma_{\text{starName},\text{MIN}(\text{year})\rightarrow\text{minYear},\text{COUNT}(\text{title})\rightarrow\text{ctTitle}}(\text{StarsIn})$
- Result has starName, minYear, and ctTitle attributes
- Then select based on the last attribute: ctTitle ≥ 3
- Finally project onto starName and minYear

Extended Projection Operator

- Classic projection $\pi_L(R)$
  - L — set of attributes of R
- Extended projection $\pi_L(R)$
  - L
    - — single attributes (as before)
    - — expressions $x \to y$ renaming attribute x to y
    - — expressions $E \to z$ where E is an expression in terms of attributes and operators

Extended Projection Operator

- Example

A	B	C
0	1	2
0	1	2
3	4	5

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

Extended Projection Operator

- Example

A	B	C
0	1	2
0	1	2
3	4	5

$$\pi_{A,B+C \to X}(R)$$

A	X
0	3
0	3
3	9

Sorting Operator

- $\tau_L(R)$
  - L is a list of attributes
  - Result is R but ordered according to the list L

Outer Join Operator

- Inner join leaves out certain tuples
- Outer join includes them with null values added

Outer Join Operator

- Example

R

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

S

B	C	D
2	3	10
2	3	11
6	7	12

$R \overset{o}{\bowtie} S$

Outer Join Operator

- Example

R

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

S

B	C	D
2	3	10
2	3	11
6	7	12

$R \overset{o}{\bowtie} S$

A	B	C	D
1	2	3	10
1	2	3	11
4	5	6	NULL
7	8	9	NULL
NULL	6	7	12

Outer Join Operator

- Left outer join:
  - Only dangling tuples in the left relation are padded with NULL and added to the relation
- Right outer join:
  - Only dangling tuples in the right relation are padded with NUMM and added to the relation

Outer Join Operator

- Example

R

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

S

B	C	D
2	3	10
2	3	11
6	7	12

$R \overset{o}{\bowtie}_l S$

Outer Join Operator

- Example

R

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

S

B	C	D
2	3	10
2	3	11
6	7	12

$R \overset{o}{\bowtie}_l S$

A	B	C	D
1	2	3	10
1	2	3	11
4	5	6	NULL
7	8	9	NULL

Outer Join Operator

- Example

R

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

S

B	C	D
2	3	10
2	3	11
6	7	12

$R \overset{o}{\bowtie}_r S$

A	B	C	D
1	2	3	10
1	2	3	11
NULL	6	7	12

Outer Join Operator

- Can also be extended to theta joins