Introduction to Data Structure (Data Management) Lecture 8

Felipe P. Vista IV

Reminder

- Everybody, make sure that your name in ZOOM is in the following format:
  - Ex: 202054321 Juan Dela Cruz

    Not changing your name to this format

    * you might be marked Absent * → absent?

- Our class will still be **online/Zoom** starting Monday 19 Oct 2020

INTRO TO DATA STRUCTURE

RELATIONAL ALGEBRA (CH 2.4 & 5.1)

Where are we now

- Motivation in using DBMS for managing data
- SQL:
  - Declaring schema for data (CREATE TABLE)
  - Insert data one record at a time (INSERT) or in bulk (.import)
  - Modify schema (ALTER TABLE) and updating data (UPDATE)
  - Query data (SELECT)

- Next-steps: More knowledge on how DBMSs works
  - Client-server architecture
  - Relational algebra and query execution

Query Evaluation Steps

SQL Query

Parse & Check Query

Translate query string into internal representation

Check syntax, access control, table names, etc…

What is best way to answer query:
Query optimization

(RA)

Logical plan → physical plan

Query execution

Return results

Query Evaluation

WHAT and HOW

- SQL = WHAT we want to get from the data
- Relational Algebra = HOW to get the data we want

SQL RA

- Moving from WHAT → HOW is query optimization
  - SQL ~> Relational Algebra ~> Physical Plan
  - Relational Algebra = Logical Plan

- Sets vs Bags

- Relational Algebra Operators

Sets vs Bags

- Sets : {a,b,c}, {a,d,e,f},{},…
  - *unordered* collection of elements *without duplicates*
- Bags: {a, a, b, c}, {b, b, b, b, b}, …
  - *unordered* collection of elements *with duplicate*

Relational Algebra has two semantics

- Set semantics = standard Relational Algebra
- Bag semantics = extended Relational Algebra

Relational Algebra Operators

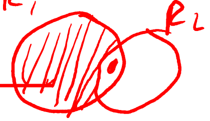
- union ($\cup$), intersection ($\cap$), difference ($-$)
- selection (σ)
- projection ($\pi, \prod$)
- cartesian product ($\times$), join ($\bowtie$)
- rename (ρ)
- duplicate elimination (δ)
- grouping and aggregation (γ)
- sorting (τ),

RA

Extended RA

Union and Difference

$R_1 \cup R_2$



$R_1 - R_2$

$R_1 \Rightarrow$ TABLE1

$R_2 \Rightarrow$ TABLE2

set × ~~$R_1 = \{a, b, c, d, d\}$~~

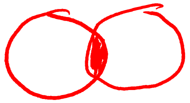
$R_2 = [e, f, a, b]$

R1 ∪ R2
R1 – R2

What does these mean over bags?

→ bags

$R_1 \cap R_2$



sets

$R_1 \cup R_2 = (a\ b\ c\ d\ d\ e\ f\ a\ b)$

$R_1 - R_2 = \{a, b, c, d, d\} \{e\} f, a, b\}$

$= \{a, b, c, d\}$

$R_1 \cap R_2 = \{d, e\}$

What about Intersection?

- Derived operator using minus

$$\boxed{R1 \cap R2 = R1 - (R1 - R2)}$$

- Derived operator using join (explain more later)

$$\boxed{R1 \cap R2 = R1 \bowtie R2}$$

Selection

- Return all tuples that satisfy given condition "c"

$$\boxed{\sigma_c(R)}$$

- Examples
  - $\sigma_{\text{salary} > 40000}$ (Employee)
  - $\sigma_{\text{name} = \text{"Mikki"}}$ (Employee)
- The condition "*c*" can be =, <, ≤, >, ≥, <> combined with AND, OR , NOT

Selection

- Employee

EmpID	Name	Salary
1234567	Mikki	20000
2345678	Nwabisa	60000
3456789	Patricia	50000
4567890	Janin	40000

- $\sigma_{salary > 40000}$ (Employee)

EmpID	Name	Salary
2345678	Nwabisa	60000
3456789	Patricia	50000

Projection

- Eliminate column(s)

$$\boxed{\pi_{A1,\ldots,An}(R)}$$

$\pi = \prod$

- Example: project Employee ID num and names
  - $\prod_{EmpID,\ Name}$ (Employee)
  - Answer(EmpID, Name)

Different semantics over sets or bags! Why?

Projection

- Employee

~~EmpID~~	Name	Salary
1234567	Divan	20000
2345678	Divan	60000
3456789	Divan	20000

$\Pi_{Name,\ Salary}$ (Employee)

Name	Salary
Divan	20000
Divan	60000
Divan	20000

Name	Salary
Divan	60000
Divan	50000

Bag semantics Dup ✓

Set semantics Dup X

Which is more efficient?

Composing RA Operators

- Patient

Num	Name	Zip	Disease
1	p1	54896	Flu
2	p2	54896	Heart
3	p3	54001	Lung
4	p4	54001	Heart

$\Pi_{\text{Zip, Disease}}$ (Patient)

Zip	Disease
54896	Flu
54896	Heart
54001	Lung
54001	Heart

$\sigma_{\text{disease="Heart"}}$(Patient)

Num	Name	Zip	Disease
2	p2	54896	Heart
4	p4	54001	Heart

oper 1

$\Pi_{\text{Zip, Disease}}(\overbrace{\sigma_{\text{disease="Heart"}}(\text{Patient})}^{\text{oper 1}})$

Zip	Disease
54896	Heart
54001	Heart

Cartesian Product

- Each tuple on R1 with each tuple in R2

rec rec

$$\boxed{R1 \times R2}$$

- Rare in practice; mainly used to express joins

Cross-Product Examples

Employee

Name	EmpID
Khan	2222222
Matt	4444444

Dependent

DepEmpID	DepName
2222222	Emily
4444444	Davis

Employee × Dependent

NamedemEm	EmpID	DepEmpID	DepName
Khan	2222222	2222222	Emily
Khan	2222222	4444444	Davis
Matt	4444444	2222222	Emily
Matt	4444444	4444444	Davis

Renaming

- Change the schema, not the instance

$$\boxed{\rho_{B1,\ldots,Bn}(R)}$$

- Example:
  - $\rho_{N,\,S}(\text{Employee}) \rightarrow \text{Answer}(N, S)$

Not really used by systems, but needed on paper

Natural Join

$$\boxed{R1 \bowtie R2}$$

- Meaning: $R1 \bowtie R2 = \prod_A(\sigma_\theta(R1 \times R2))$

- Where:
  - Selection σ checks equality of all common attributes (attributes with the same names)
  - Projection $\prod$ eliminates duplicate common attributes

    removes

A | B | B | C

Natural Join Example

R × S = ?
4 · 3 = 12

R

A	B
X	Y
X	Z
Y	Z
Z	V

S

B	C
Z	U
V	W
Z	V

$R \bowtie S$
$= \prod_A(\sigma_\theta(R \times S))$
12

A	B	C
X	Z	U
X	Z	V
Y	Z	U
Y	Z	V
Z	V	W

$\sigma_G(R \times S)$

A B B C

$\Pi_A(\sigma_G(R \times A))$

3

Natural Join Example #2

Anonymous Patient P

age	Zip	Disease
56	54896	Heart
23	54001	Flu

Voters V

Name	Age	Zip
P1	56	54896
P2	23	54001

$P \bowtie V$

Natural Join Example #2

Anonymous Patient P

age	Zip	Disease
56	54896	Heart
23	54001	Flu

Voters V

Name	Age	Zip
P1	56	54896
P2	23	54001

$P \bowtie V$

Age	Zip	Disease	Name
56	54896	Heart	P1
23	54001	Flu	P2

Natural Join

- Given schemas R(A, B, C, D), S(A, C, E); what is the schema of $R \bowtie S$?

Natural Join

- Given schemas R(A, B, C, D), S(A, C, E); what is the schema of R ⋈ S ?
  - (A, B, C, D, E) through join on (A, C)
- Given R(A, B, C), S(D, E); what is R ⋈ S ?
  - (A, B, C, D, E) through cross product
- Given R(A, B), S(A, B); what is R ⋈ S ?
  - (A, B) through cross intersection

Natural Join

- Given schemas R(A, B, C, D), S(A, C, E); what is the schema of R ⋈ S ?
  - (A, B, C, D, E) through join on (A, C)
- Given R(A, B, C), S(D, E); what is R ⋈ S ?

Natural Join

- Given schemas R(A, B, C, D), S(A, C, E); what is the schema of R ⋈ S ?
  - (A, B, C, D, E) through join on (A, C)
- Given R(A, B, C), S(D, E); what is R ⋈ S ?
  - (A, B, C, D, E) through cross product
- Given R(A, B), S(A, B); what is R ⋈ S ?
  - (A, B) through cross intersection

Natural Join

- Given schemas R(A, B, C, D), S(A, C, E); what is the schema of R ⋈ S ?
  - (A, B, C, D, E) through join on (A, C)
- Given R(A, B, C), S(D, E); what is R ⋈ S ?
  - (A, B, C, D, E) through cross product
- Given R(A, B), S(A, B); what is R ⋈ S ?
  - (   )

Natural Join

- Given schemas R(A, B, C, D), S(A, C, E); what is the schema of R ⋈ S ?
  - (A, B, C, D, E) through join on (A, C)
- Given R(A, B, C), S(D, E); what is R ⋈ S ?
  - (A, B, C, D, E) through cross product
- Given R(A, B), S(A, B); what is R ⋈ S ?
  - (A, B) through cross intersection

```
AnonPatient(age,zip,disease)
Voters(name,age,zip)
```

Theta Join

- A join that involves a predicate

$$\boxed{R1 \bowtie_{\theta} R2 = \sigma_{\theta}(R1 \times R2)}$$

- Here θ can be any condition
- For the voters/patients example:

$$P \bowtie_{P.zip = V.zip \text{ AND } P.age \geq V.age - 1 \text{ AND } P.age \leq V.age + 1} V$$

Equijoin

- A theta join where θ is an equality predicate
- By far the most used variant of join in practice

Equijoin Example

Anonymous Patient P

age	Zip	Disease
56	54896	Heart
23	54001	Flu

Voters V

Name	Age	Zip
P1	56	54896
P2	23	54001

$$P \bowtie_{P.age=V.age} V$$

Equijoin Example

Anonymous Patient P

age	Zip	Disease
56	54896	Heart
23	54001	Flu

Voters V

Name	Age	Zip
P1	56	54896
P2	23	54001

$$P \bowtie_{P.age=V.age} V$$

P.Age	P.Zip	P.Disease	V.Name	V.Zip	V.Age
56	54896	Heart	P1	54896	56
23	54001	Flu	P2	54001	23

Join Summary

- Theta-join: $R \bowtie_\theta S = \sigma_\theta(R \times S)$
  - Join of R and S with a join condition θ
  - Cross product followed by selection θ
- Equijoin: $R \bowtie_\theta S = \sigma_\theta(R \times S)$
  - Join condition θ consists only of equalities
- Natural join: $R \bowtie S = \prod_A(\sigma_\theta(R \times S))$
  - Equijoin
  - Equality on all fields with same name in R and in S
  - Projection $\prod_A$ drops all redundant attributes

duplicates

R (A, B, C) S(A, D)
(A, B, C, D)

So which join is it?

- When we write $R \bowtie_{\theta} S$, we usually mean an equijoin, but we often omit the equality predicate when it is clear from the context

More Joins

- Outer join
  - Include tuples with no matches in the output
  - Use NULL values for missing attributes
  - Does not eliminate duplicate columns

- Variants
  - Left outer join
  - Right outer join
  - Full outer join

Outer Join Example

Anonymous Patient P

age	Zip	Disease
56	54896	Heart
23	54001	Flu
34	54001	Lung

Anonymous Job J

Job	Age	Zip
Explorer	56	54896
Diver	23	54001

LOJ ⟕ ⟖ ⟗

$P \bowtie J$

Outer Join Example

Anonymous Patient P

age	Zip	Disease
56	54896	Heart
23	54001	Flu
34	54001	Lung

Anonymous Job J

Job	Age	Zip
Explorer	56	54896
Diver	23	54001

Null N? Null

P ⟕ J

P.Age	P.Zip	P.Disease	J.Job	J.Age	J.Zip
56	54896	Heart	Explorer	56	54896
23	54001	Flu	Diver	23	54001
34	54001	Lung	null	null	null

More Examples

```
Supplier(sno, sname, scity, sstate)
Part(pno, pname, psize, pcolor)
Supply(sno, pno, qty, price)
```

σ π

- Name of supplier of parts with size greater than 10:

More Examples

```
Supplier(sno, sname, scity, sstate)
Part(pno, pname, psize, pcolor)
Supply(sno, pno, qty, price)
```

* Name of supplier of parts with size greater than 10:

$\prod_{sname}(Supplier \bowtie Supply \bowtie (\sigma_{psize>10}(Part))$

* Name of supplier of red or parts with size greater than 10:

$\prod_{sname}(Supplier \bowtie Supply \bowtie (\sigma_{psize>10}(Part) \cup \sigma_{pcolor=\text{'red'}}(Part)))$

More Examples

```
Supplier(sno, sname, scity, sstate)
Part(pno, pname, psize, pcolor)
Supply(sno, pno, qty, price)
```

* Name of supplier of parts with size greater than 10:

select

$\Pi_{sname}(Supplier \bowtie Supply \bowtie (\sigma_{psize>10}(Part))$

(supply.pno = Part.pno AND Supply.sno = Supplier.sno)

* Name of supplier of red or parts with size greater than 10:

More Examples

```
Supplier(sno, sname, scity, sstate)
Part(pno, pname, psize, pcolor)
Supply(sno, pno, qty, price)
```

* Name of supplier of parts with size greater than 10:

$\Pi_{sname}(Supplier \bowtie Supply \bowtie (\sigma_{psize>10}(Part))$

color

* Name of supplier of red or parts with size greater than 10:

$\Pi_{sname}(Supplier \bowtie Supply \bowtie (\sigma_{psize>10}(Part) \cup \sigma_{pcolor='red'}(Part)))$

Thank you.