CS522 Advanced Database Systems
Query Optimization

Chengyu Sun
California State University, Los Angeles

---

SQL Query Example

```
select B, D
from R, S
where R.A = c and S.E = 2 and R.C = S.C;
```

---

Parse Tree

```
<SPARQL>
<Query>
  SELECT <SelList> FROM <FromList> WHERE <Condition>
  <Attribute> <SelList> <RefName> <FromList> <Conditions> AND <Condition>
  <Attribute> R B S D
  <RefName> S
  <Conditions> AND <Condition>
  <Attribute> S.E 2 R.C S.C
```

---

Logical Query Plan

- Operators
  - R, S, =
- Arguments (Operands)
  - relations
    - R and S
  - Parameters
    - non-relations
      - B, D
      - A=2 AND E=2 AND R.C=S.C
- Parameters can be "applied" to each tuple in the relation(s)

```
π_{B,D}(σ_{A=2 \& E=2 \& \text{R.C}=\text{S.C}}(R \bowtie S))
```

---

Subqueries in Conditions

```
select A, B from R where C in (select C from S);
```

- notational complications
- expensive to evaluate

```
π_{A,B}(σ_{C\in S}(R \bowtie S))
```

---

Two-argument Selection

```
π_{A,B}(σ_{<\text{condition}>}(R \bowtie \text{IN}(\text{C}, S)))
```

---

1
Convert to Relational Algebra Selection

- Have to come up rules on a case-by-case basis
  - IN, EXISTS, ANY, ALL ...
  - correlated or uncorrelated
- In some rare cases we can leave the two-argument selection as part of the logical query plan

Example: Uncorrelated IN

Simple Algebraic Laws for Transformation

- Commutative law
  - \( R \cup S = S \cup R \)
  - \( R \cap S = S \cap R \)
  - \( R \times S = S \times R \)
  - \( R \bowtie S = S \bowtie R \)
- Associative law
  - \( (R \cup S) \cup T = R \cup (S \cup T) \)
  - \( (R \cap S) \cap T = R \cap (S \cap T) \)
  - \( (R \times S) \times T = R \times (S \times T) \)
  - \( (R \bowtie S) \bowtie T = R \bowtie (S \bowtie T) \)

Bags vs. Sets

- Consider the distributive laws
  - \( R \cap (S \cup T) = (R \cap S) \cup (R \cap T) \)
  - \( R \cup (S \cap T) = (R \cup S) \cap (R \cup T) \)

Proofs HOWTO

- Prove set equivalence
  - \( t \in A \Rightarrow t \in B \)
  - \( t \in B \Rightarrow t \in A \)
- Prove bag equivalence
- Disprove
Selection Laws ...

**Splitting**
- \( \sigma_{a_1 \text{ AND } a_2}(R) = \sigma_{a_1}(\sigma_{a_2}(R)) \)
- \( \sigma_{a_1 \text{ OR } a_2}(R) = \sigma_{a_1}(R) \cup \sigma_{a_2}(R) \)
- \( \cup \text{ or } \cup \? \)

**Pushing**
- \( \sigma_{\rho(S)} \cup \sigma_2 \cap \sigma_3 \times \sigma_4 \times \sigma_5 \times \sigma_6 \)
- push to both arguments
  - \( \sigma_{\rho(S)} \cap \sigma_1(S) \)
  - \( \text{when??} \)
- push to one of the arguments
  - \( \sigma_{\rho(S)} \cap \sigma_1(S) \)
  - \( \text{when??} \)

---

Examples: Pushing Selections

**R(a,b) and S(b,c)**
- \( \sigma_{a=1 \text{ OR } a=3}(R \bowtie S) \)
- \( \sigma_{b=1}(R) \bowtie S \)

---

Projection Laws

**Adding projections**
- In general, we can project out attributes that are not used later on

**Examples:**
- \( R(a,b,c) \text{ and } S(c,d,e) \)
  - \( \pi_{a>1 \rightarrow a>3}(R \bowtie S) \)
  - \( \pi_{a>b \rightarrow c>d \rightarrow e}(R \bowtie S) \)
  - Union, intersection, difference??

---

Project Law Examples

**Prove**
- \( \pi_1(R \cup B S) = \pi_1(R) \cup \pi_1(S) \)

**Disprove**
- \( \pi_1(R \cap B S) = \pi_1(R) \cap \pi_1(S) \)
- \( \pi_1(R \setminus B S) = \pi_1(R) \setminus \pi_1(S) \)
- \( \pi_1(R \setminus S) = \pi_1(R) \setminus \pi_1(S) \)

---

Some Other Laws

**Duplicate elimination**
- \( \delta(R) = R \text{ if } ...? \)
- \( \delta(R \times S) = \delta(R) \times \delta(S) \)
- \( \delta(R \bowtie S) = \delta(R) \bowtie \delta(S) \)
- \( \delta(\sigma_1(R)) = \sigma_1(\delta(R)) \)

**Group by**
- Duplicate-impervious aggregations
About Algebraic Laws
- There are too many to remember
- You can come up with more (as long as you can prove)
- Beware of the different semantics of sets and bags

Cost-based Query Optimization
- Choose the best logical or physical query plan
- What influence the "cost" of a query?
  - Choice of operators
  - Order of operators
  - Interaction between operators

Selectivity Estimation
- Selectivity = |ResultSet| / |DataSet|
- Cost estimation for logical query plans
  - All equivalent plans produce the same final result set
  - The plan which produces the smallest intermediate result set wins
- Provide information for choosing physical query plans

Selectivity Estimation with Simple Statistics
- T(R) – number of tuples in R
- V(R,a) – number of distinct values of attribute a
- V(R, [a_1,a_2,…,a_n])

Estimating Selection Selectivity
- a=x: 1/V(R,a)
- a>x: 1/2 or 1/3
- a<x: ??
- c_i AND c_j: ??
- c_i OR c_j: ??
- Example: R(a,b)
  - T(R) = 10000, V(R,a) = 50
  - Estimate | σ_{a=10 or b=20}(R) |

Estimating Join Size ...
- Very hard problem even with more sophisticated methods
- Consider natural join of R(X,Y) and S(Y,Z)
  - 0
  - |R| or |S|
  - |R|^*|S|
... Estimating Join Size ...

- Simplifying assumptions
  - Containment of value sets
    - if $V(R,Y) \subseteq V(R,Y)$, then every $Y$-value of $R$ is a $Y$-value of $S$
  - Preservation of value sets
    - if $A$ is an attribute of $R$ but not a join attribute, then $V(S \bowtie R, A) = V(R,A)$
  - When do these assumptions hold?

- Estimating Other Operators
  - Projection
  - Union, intersection, difference
    - Usually the average of max and min
  - Duplicate elimination and group by
    - $V(R, \{a_1, a_2, \ldots, a_n\})$

... Estimating Join Size

- $|R \bowtie S| \approx ?$
- Example:
  - $R(a,b)$: $T(R) = 1000$, $V(R,b) = 20$
  - $S(b,c)$: $T(S) = 2000$, $V(S,b) = 50$, $V(S,c) = 100$
  - $U(c,d)$: $T(U) = 5000$, $V(U,c) = 500$
- Join on multiple attributes where $Y = \{y_1, y_2, \ldots, y_n\}$

Example: Plan Selection

- $R(a,b)$
  - $T(R) = 5000$, $V(R,a) = 50$, $V(R,b) = 100$
- $S(b,c)$
  - $T(S) = 2000$, $V(S,b) = 200$, $V(S,c) = 100$

More Statistics

- Criteria
  - Small storage footprint
  - Low computation overhead
  - Accurate estimation
- General techniques
  - Histogram
    - Works very well for low-dimensional data
  - Sampling
    - Works better for high-dimensional data

About Histograms

- Construction
  - Sampling
  - Buckets
- Maintenance
  - Incremental
  - Periodically re-build
- Usage
  - Uniform assumption
### Equi-width Histogram

- Construction??
- Maintenance??
- Usage??

### Equi-depth Histogram

- Construction??
- Maintenance??
- Usage??

### Examples

- Estimate | $\sigma_{\text{score}=60}$ |
- Estimate | $\sigma_{\text{score} \leq 60 \text{ and score} > 50}$ |

### Join Order

- How many different ways can we join $R, S, T$?
- How about $R_1, R_2, \ldots, R_n$?
- Number of tree shapes:
  
  $$ T(1) = 1 $$
  
  $$ T(n) = \sum_{i=1}^{n-1} T(i) T(n - i) $$

### Select Join Order

- Consider only left-deep trees: we still have $n!$ choices
  - Dynamic programming
  - Greedy

### Dynamic Programming

- Diagram showing the dynamic programming approach for join orders.
Greedy

Histogram for Spatial Data

Euler Histogram

Physical Query Plans

Readings

- Stanford book: Chapter 16
- [Euler histogram paper]