CSE544
Data Management

Lectures 9
Query Optimization (Part 1)
Announcements

• HW2 is due tomorrow!
• HW3 will be posted on Wednesday
• Review 5 (How good?) due Wednesday
• Mini-project guidelines posted
Query Optimization Motivation

SQL query

Parse & Rewrite Query

Select Logical Plan

Select Physical Plan

Query Execution

Disk

Declarative query

Recall physical and logical data independence

Query optimization

Logical plan

Physical plan
Query Optimization

Goal:

• Given a query plan, find a cheaper (cheapest?) equivalent plan

• Why difficult:
  – Need to explore a large number of plans
  – Need to estimate the cost of each plan
Query Optimization

Three major components:

1. Cardinality and cost estimation
2. Search space
3. Plan enumeration algorithms
Query Optimization

Three major components:

1. Cardinality and cost estimation
2. Search space
3. Plan enumeration algorithms
Cardinality Estimation

Problem: given statistics on base tables and a query, estimate size of the answer

Very difficult, because:
- Need to do it very fast
- Need to use very little memory
Statistics on Base Data

- Number of tuples (cardinality) $T(R)$
- Number of physical pages $B(R)$
- Indexes, number of keys in the index $V(R,a)$
- Histogram on single attribute (1d)
- Histogram on two attributes (2d)

Computed periodically, often using sampling
Assumptions

• Uniformity

• Independence

• Containment of values

• Preservation of values
Size Estimation

Selection: size decreases by selectivity factor $\theta$

$$T(\sigma_{\text{pred}}(R)) = \theta_{\text{pred}} \times T(R)$$
Size Estimation

Selection: size decreases by *selectivity factor* $\theta$

\[
T(\sigma_{\text{pred}}(R)) = \theta_{\text{pred}} \times T(R)
\]

\[
T(R \bowtie_{A=B} S) = \theta_{A=B} \times T(R) \times T(S)
\]
Selectivity Factors

Uniformity assumption

Equality:

\[ T(\sigma_{\text{pred}}(R)) = \theta_{\text{pred}} \times T(R) \]
Selectivity Factors

Uniformity assumption

Equality:

• $\theta_{A=c} = 1/V(R,A)$

$T(\sigma_{\text{pred}}(R)) = \theta_{\text{pred}} \times T(R)$
Selectivity Factors

**Uniformity assumption**

Equality:
- $\theta_{A=c} = 1/V(R,A)$

Range:
- $\theta_{c1<A<c2} = (c2 - c1)/(\max(R,A) - \min(R,A))$
Selectivity Factors

**Uniformity assumption**

Equality:
- $\theta_{A=c} = 1/V(R,A)$

Range:
- $\theta_{c_1<A<c_2} = (c_2 - c_1)/(\max(R,A) - \min(R,A))$

Conjunction
- $\sigma_{A=c\ and\ B=d}(R)$

T($\sigma_{\text{pred}(R)}$) = $\theta_{\text{pred}} \times T(R)$
Selectivity Factors

**Uniformity assumption**

Equality:
- $\theta_{A=c} = 1/V(R,A)$

Range:
- $\theta_{c1<A<c2} = (c2 - c1)/(\max(R,A) - \min(R,A))$

**Conjunction**

**Independence assumption**

- $\theta_{\text{pred1 and pred2}} = \theta_{\text{pred1}} \cdot \theta_{\text{pred2}} = 1/V(R,A) \cdot 1/V(R,B)$

$T(\sigma_{\text{pred}(R)}) = \theta_{\text{pred}} \cdot T(R)$
Selectivity Factors

Join

T(R \bowtie_{A=B} S) = \theta_{A=B} * T(R) * T(S)

R \bowtie_{R.A=S.B} S
Selectivity Factors

Join

- $\theta_{R.A=S.B} = \frac{1}{\text{MAX}(V(R,A), V(S,B))}$

Why? Will explain next...
Selectivity Factors

**Containment of values**: if $V(R,A) \leq V(S,B)$, then the set of A values of R is included in the set of B values of S

- Note: this indeed holds when A is a foreign key in R, and B is a key in S

\[
T(R \bowtie_{R.A=S.B} S) = \theta_{A=B} \ast T(R) \ast T(S)
\]
Selectivity Factors

Assume $V(R,A) \leq V(S,B)$

• Tuple $t$ in $R$ joins with $T(S) / V(S,B)$ tuples in $S$
Selectivity Factors

Assume $V(R,A) \leq V(S,B)$

- Tuple $t$ in $R$ joins with $T(S)/V(S,B)$ tuples in $S$
- Hence $T(R \bowtie_{A=B} S) = T(R) \cdot T(S) / V(S,B)$

$$T(R \bowtie_{A=B} S) = \theta_{A=B} \cdot T(R) \cdot T(S)$$
Selectivity Factors

Assume $V(R,A) \leq V(S,B)$

- Tuple $t$ in $R$ joins with $T(S)/V(S,B)$ tuples in $S$
- Hence $T(R \bowtie_{A=B} S) = T(R) \cdot T(S) / V(S,B)$

In general:
- $T(R \bowtie_{A=B} S) = T(R) \cdot T(S) / \max(V(R,A), V(S,B))$
- $\theta_{R.A=S.B} = 1/ (\max(V(R,A), V(S,B)))$
Final Assumption

Preservation of values:
For any other attribute C:
• \( V(R \bowtie_{A=B} S, C) = V(R, C) \) or
• \( V(R \bowtie_{A=B} S, C) = V(S, C) \)

• This is needed higher up in the plan
Computing the Cost of a Plan

• Estimate cardinalities bottom-up

• Estimate cost by using estimated cardinalities

• Examples next...
Logical Query Plan 1

\[ \sigma_{\text{pno}=2 \land \text{scity}='Seattle' \land \text{sstate}='WA'} \pi_{\text{sname}} \]

\[ \text{SELECT sname} \
\text{FROM Supplier x, Supply y} \
\text{WHERE x.sid = y.sid} \land y.\text{pno} = 2 \land x.\text{scity} = 'Seattle' \land x.\text{sstate} = 'WA' \]

\[ \text{T(Supplier)} = 1000 \\]
\[ \text{B(Supplier)} = 100 \\]
\[ \text{V(Supplier, scity)} = 20 \\]
\[ \text{V(Supplier, state)} = 10 \]

\[ \text{M} = 11 \]
Logical Query Plan 1

\[
\sigma_{pno=2 \land scity='Seattle' \land sstate='WA'} \pi_{\text{sname}}
\]

\[\text{T} = 10000\]

\[\text{SELECT sname FROM Supplier x, Supply y WHERE x.sid = y.sid and y.pno = 2 and x.scity = 'Seattle' and x.sstate = 'WA'}\]

\[T(\text{Supplier}) = 1000\]
\[B(\text{Supplier}) = 100\]
\[V(\text{Supplier, scity}) = 20\]
\[V(\text{Supplier, state}) = 10\]

\[M=11\]
Logical Query Plan 1

Estimated (why?)

\[ \sigma_{\text{pno}=2 \land \text{scity}='Seattle' \land \text{sstate}='WA'} \]

\[ \Pi_{\text{sname}} \]

\[ \text{T} = 10000 \]

Because key / foreign-key

\[ \text{sid} = \text{sid} \]

SELECT \text{sname}
FROM \text{Supplier} \text{x, Supply} \text{y}
WHERE \text{x.sid} = \text{y.sid}
and \text{y.pno} = 2
and \text{x.scity} = 'Seattle'
and \text{x.sstate} = 'WA'

\[ \text{T(Supplier)} = 1000 \]
\[ \text{B(Supplier)} = 100 \]
\[ \text{V(Supplier, scity)} = 20 \]
\[ \text{V(Supplier, state)} = 10 \]

\[ \text{M} = 11 \]
Supply(sid, sname, scity, sstate)
Supplier(sid, pno, quantity)

Logical Query Plan 1

Estimated (why?)

\[ \sigma_{pno=2 \land \text{scity='Seattle'} \land \text{sstate='WA'}} \]

\[ T(Supplier) = 1000 \]
\[ B(Supplier) = 100 \]
\[ V(Supplier, \text{scity}) = 20 \]
\[ V(Supplier, \text{sstate}) = 10 \]

\[ \pi_{sname} \]

\[ \theta = 1/\max(V(Supply, sid) \cdot V(Supplier, sid)) = 1/V(Supplier, sid) \]

Because key / foreign-key

\[ \text{sid} = \text{sid} \]

Select

\[ \text{SELECT sname} \]
\[ \text{FROM Supplier x, Supply y} \]
\[ \text{WHERE x.sid = y.sid} \]
\[ \text{and y.pno = 2} \]
\[ \text{and x.scity = 'Seattle'} \]
\[ \text{and x.sstate = 'WA'} \]

T(Supply) = 10000
B(Supply) = 100
V(Supply, pno) = 2500

M=11
Logical Query Plan 1

\[ \sigma_{pno=2 \land scity='Seattle' \land sstate='WA'}  \]

\[ \pi_{sname} \]

Estimated (why?)

\[ T < 1 \]

\[ T = 10000 \]

\[ \text{M} = 11 \]

T(Supply) = 10000
B(Supply) = 100
V(Supply, pno) = 2500

T(Supplier) = 1000
B(Supplier) = 100
V(Supplier, scity) = 20
V(Supplier, state) = 10
Logical Query Plan 2

\[
\text{SELECT } \text{sname} \\
\text{FROM Supplier } x, \text{ Supply } y \\
\text{WHERE } x.\text{sid} = y.\text{sid} \\
\text{and } y.\text{pno} = 2 \\
\text{and } x.\text{scity} = 'Seattle' \\
\text{and } x.\text{sstate} = 'WA'
\]

\[\begin{align*}
\text{T(Supplier)} & = 1000 \\
\text{B(Supplier)} & = 100 \\
\text{V(Supplier, scity)} & = 20 \\
\text{V(Supplier, state)} & = 10
\end{align*}\]

\[\begin{align*}
\text{T(Supply)} & = 10000 \\
\text{B(Supply)} & = 100 \\
\text{V(Supply, pno)} & = 2500
\end{align*}\]
Logical Query Plan 2

\[
\text{SELECT } sname \\
\text{FROM } Supplier x, Supply y \\
\text{WHERE } x.sid = y.sid \\
\text{and } y.pno = 2 \\
\text{and } x.scity = 'Seattle' \\
\text{and } x.sstate = 'WA'
\]

\[
\text{T(Supplier) = 1000} \\
\text{B(Supplier) = 100} \\
\text{V(Supplier, scity) = 20} \\
\text{V(Supplier, sstate) = 10}
\]

\[
\text{T(Supply) = 10000} \\
\text{B(Supply) = 100} \\
\text{V(Supply, pno) = 2500}
\]

\[
\text{M=11}
\]
Logical Query Plan 2

SELECT sname
FROM Supplier x, Supply y
WHERE x.sid = y.sid
  AND y.pno = 2
  AND x.scity = 'Seattle'
  AND x.sstate = 'WA'

Very wrong! Why?
Logical Query Plan 2

\[
\text{SELECT sname}
\text{FROM Supplier x, Supply y}
\text{WHERE x.sid = y.sid}
\text{and y.pno = 2}
\text{and x.scity = 'Seattle'}
\text{and x.sstate = 'WA'}
\]

Very wrong! Why?

T = 4
T = 5
T = 4
T = 5

\text{T(Supply) = 10000}
\text{B(Supply) = 100}
\text{V(Supply, pno) = 2500}
\text{T(Supplier) = 1000}
\text{B(Supplier) = 100}
\text{V(Supplier, scity) = 20}
\text{V(Supplier, sstate) = 10}

\text{M=11}
Logical Query Plan 2

Select `sname` from `Supplier x, Supply y` where `x.sid = y.sid` and `y.pno = 2` and `x.scity = 'Seattle'` and `x.sstate = 'WA'`.

- **T = 4**: `Sid = Sid`
- **T = 4**: `σ_pno=2`
- **T = 5**: `σ_scity='Seattle' ∧ sstate='WA'`

**Different estimate 😞**

- **T = 4**: `Sid = Sid`
- **T = 5**: `σ_scity='Seattle' ∧ sstate='WA'`

Very wrong! Why?

**M = 11**

**Prime Model**: 11

**Query Cost**: 4

**Cost**: 5

**Wrong**: Very wrong!

**Reason**: Different estimate

**Data Estimation**:
- `T(Supplier) = 1000`
- `B(Supplier) = 100`
- `V(Supplier, scity) = 20`
- `V(Supplier, state) = 10`
- `T(Supply) = 10000`
- `B(Supply) = 100`
- `V(Supply, pno) = 2500`
Physical Plan 1

\[ \pi_{\text{sname}} \]

\[ \sigma_{\text{pno}=2 \land \text{scity}='Seattle' \land \text{sstate}='WA'} \]

\[ \text{T} < 1 \]

\[ \text{T} = 10000 \]

\[ \text{Block nested loop join} \]

Total cost: \[ \frac{100}{10} \times 100 = 1000 \]
Physical Plan 1

\[\pi_{\text{sname}}(\sigma_{\text{pno}=2 \land \text{scity}='Seattle' \land \text{sstate}='WA'}(\text{Supplier}))\]

Scan \rightarrow \text{Block nested loop join} \rightarrow \text{Scan}

Total cost: \[100 + 100 \times 100 / 10 = 1100\]

- \text{T(Supplier)} = 1000
- \text{B(Supplier)} = 100
- \text{V(Supplier, scity)} = 20
- \text{V(Supplier, state)} = 10

\text{M} = 11
Physical Plan 2

\[
\pi_{\text{sname}}(\sigma_{\text{sstate} = 'WA'}(\pi_{\text{scity}}(\sigma_{\text{pno} = 2}(\text{Supply})))))
\]

Cost of \text{Supply}(\text{pno}) = 4
Cost of \text{Supplier}(\text{scity}) = 50
Total cost: 54

\text{M} = 11

\text{T(\text{Supply})} = 10000
\text{B(\text{Supply})} = 100
\text{V(\text{Supply}, \text{pno})} = 2500
Physical Plan 2

\[ \pi_{\text{name}}(\sigma_{\text{sstate} = \text{`WA'}}(\text{Supplier})) \]

Cost of Supplier(\text{scity} = \text{`Seattle'}) = 50
Cost of Supply(\text{pno}) = 4
Total cost: 54

Unclustered index lookup Supplier(\text{scity})
Unclustered index lookup Supply(\text{pno})

T(\text{Supplier}) = 1000
B(\text{Supplier}) = 100
V(\text{Supplier, scity}) = 20
V(\text{Supplier, state}) = 10

M = 11

T(\text{Supply}) = 10000
B(\text{Supply}) = 100
V(\text{Supply, pno}) = 2500

Main memory join

\[ \sigma_{\text{pno} = 2}(\text{Supply}) \]

\[ \sigma_{\text{sid} = \text{sid}}(\text{Supply}, \text{Supplier}) \]

\[ \pi_{\text{name}}(\text{Supplier}) \]
Physical Plan 2

\[ \Pi_{sname} \sigma_{sstate='WA'}(\pi_{sname} \sigma_{pno=2}(\text{Supply})) \]

\[ T(\text{Supply}) = 10000 \]
\[ B(\text{Supply}) = 100 \]
\[ V(\text{Supply}, \text{pno}) = 2500 \]

\[ T(\text{Supplier}) = 1000 \]
\[ B(\text{Supplier}) = 100 \]
\[ V(\text{Supplier}, \text{scity}) = 20 \]
\[ V(\text{Supplier}, \text{sstate}) = 10 \]

\[ M = 11 \]

Cost of \( \text{Supply}(\text{pno}) = 4 \)
Cost of \( \text{Supplier}(\text{scity}) = 50 \)
Total cost: 54

Unclustered index lookup \( \text{Supply}(\text{pno}) \)

Main memory join

Unclustered index lookup \( \text{Supplier}(\text{scity}) \)
Physical Plan 3

\[ \pi_{\text{sname}}(\sigma_{\text{scity} = \text{'Seattle'} \land \text{sstate} = \text{'WA'}}(\text{Supplier} \bowtie \text{Supply}(\text{pno} = 2))) \]

Cost of \( \text{Supply}(\text{pno}) \) = 4
Cost of Index join = 4
Total cost: 8

Unclustered index lookup \( \text{Supply}(\text{pno}) \)

Clustered Index join

\[ \sigma_{\text{scity} = \text{'Seattle'} \land \text{sstate} = \text{'WA'}}(\text{Supplier} \bowtie \text{Supply}(\text{sid} = \text{sid})) \]

\[ \begin{align*}
T(\text{Supplier}) &= 1000 \\
B(\text{Supplier}) &= 100 \\
V(\text{Supplier}, \text{scity}) &= 20 \\
V(\text{Supplier}, \text{sstate}) &= 10
\end{align*} \]

\[ M = 11 \]

\[ \begin{align*}
T(\text{Supply}) &= 10000 \\
B(\text{Supply}) &= 100 \\
V(\text{Supply}, \text{pno}) &= 2500
\end{align*} \]
Physical Plan 3

\[ \Pi_{sname} \sigma_{\text{scity}=\text{'Seattle'} \land \text{sstate}=\text{'WA'}} \]

\[ \sigma_{\text{pno}=2} \]

\( \sigma_{\text{scity}=\text{'Seattle'} \land \text{sstate}=\text{'WA'}} \)∧\( \sigma_{\text{pno}=2} \)

Unclustered index lookup
Supply(pno)

Cost of Supply(pno) = 4
Cost of Index join = 4
Total cost:

M=11
Physical Plan 3

\[ \Pi_{\text{sname}} \sigma_{\text{scity}=\text{Seattle} \land \text{sstate}=\text{WA}} \]

\[
\begin{align*}
T(Supplier) &= 1000 \\
B(Supplier) &= 100 \\
V(Supplier, \text{scity}) &= 20 \\
V(Supplier, \text{sstate}) &= 10
\end{align*}
\]

\[
\begin{align*}
T(Supply) &= 10000 \\
B(Supply) &= 100 \\
V(Supply, \text{pno}) &= 2500
\end{align*}
\]

Cost of Supply(pno) = 4
Cost of Index join = 4
Total cost: 8

Unclustered index lookup
Supply(pno)

Clusteered Index join

T = 4

\[
\begin{align*}
\sigma_{\text{pno}=2}
\end{align*}
\]

\[
\sigma_{\text{scity}=\text{Seattle} \land \text{sstate}=\text{WA}}
\]

M = 11

Supplier(sid, sname, scity, sstate)
Supply(sid, pno, quantity)
Discussion

- We considered only IO cost; real systems need to consider IO+CPU
- Each system has its own hacks
- We assumed that all index pages were in memory: sometimes we need to add the cost of fetching index pages from disk
Histograms

• T(R), V(R,A) too coarse
• Histogram: separate stats per bucket

• In each bucket store:
  – T(bucket)
  – V(bucket,A) – optional
Histograms

Employee(ssn, name, age)

T(Employee) = 25000, V(Employee, age) = 50
\[ \sigma_{\text{age}=48}(\text{Employee}) = ? \]
Histograms

Employee(ssn, name, age)

T(Employee) = 25000, V(Employee, age) = 50
\( \sigma_{\text{age}=48}(\text{Employee}) = ? \)

Estimate: \( T(\text{Employee}) \div V(\text{Employee,age}) = 500 \)
Histograms

Employee(ssn, name, age)

T(Employee) = 25000, V(Employee, age) = 50
\[ \sigma_{age=48}(Employee) = ? \]

Estimate: \( \frac{T(Employee)}{V(Employee, age)} = 500 \)

<table>
<thead>
<tr>
<th>Age:</th>
<th>0..20</th>
<th>20..29</th>
<th>30-39</th>
<th>40-49</th>
<th>50-59</th>
<th>&gt; 60</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>200</td>
<td>800</td>
<td>5000</td>
<td>12000</td>
<td>6500</td>
<td>500</td>
</tr>
</tbody>
</table>
Histograms

Employee(ssn, name, age)

$T(\text{Employee}) = 25000$, $V(\text{Employee, age}) = 50$

$\sigma_{age=48}(\text{Employee}) = ?$

Estimate: $T(\text{Employee}) / V(\text{Employee, age}) = 500$

<table>
<thead>
<tr>
<th>Age:</th>
<th>0..20</th>
<th>20..29</th>
<th>30-39</th>
<th>40-49</th>
<th>50-59</th>
<th>&gt; 60</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T =$</td>
<td>200</td>
<td>800</td>
<td>5000</td>
<td>12000</td>
<td>6500</td>
<td>500</td>
</tr>
</tbody>
</table>

Assume $V = 10$
## Histograms

**Employee**(ssn, name, age)

\[ T(\text{Employee}) = 25000, \quad V(\text{Employee, age}) = 50 \]

\[ \sigma_{\text{age}=48}(\text{Employee}) = ? \]

**Estimate:** \[ T(\text{Employee}) / V(\text{Employee,age}) = 500 \]

<table>
<thead>
<tr>
<th>Age:</th>
<th>0..20</th>
<th>20..29</th>
<th>30-39</th>
<th>40-49</th>
<th>50-59</th>
<th>&gt; 60</th>
</tr>
</thead>
<tbody>
<tr>
<td>( T = )</td>
<td>200</td>
<td>800</td>
<td>5000</td>
<td>12000</td>
<td>6500</td>
<td>500</td>
</tr>
</tbody>
</table>

**Estimate:** \[ 12000/10 = 1200 \]

Assume \( V = 10 \)
Histograms

Employee(ssn, name, age)

T(Employee) = 25000, V(Employee, age) = 50
\( \sigma_{age=48}(Employee) = ? \)

Estimate: \( \frac{T(Employee)}{V(Employee, age)} = 500 \)

<table>
<thead>
<tr>
<th>Age:</th>
<th>0..20</th>
<th>20..29</th>
<th>30-39</th>
<th>40-49</th>
<th>50-59</th>
<th>&gt; 60</th>
</tr>
</thead>
<tbody>
<tr>
<td>T =</td>
<td>200</td>
<td>800</td>
<td>5000</td>
<td>12000</td>
<td>6500</td>
<td>500</td>
</tr>
<tr>
<td>V =</td>
<td>3</td>
<td>10</td>
<td>7</td>
<td>6</td>
<td>5</td>
<td>4</td>
</tr>
</tbody>
</table>

Estimate: \( \frac{12000}{10} = 1200 \)
Histograms

**Employee**(ssn, name, age)

T(Employee) = 25000, \( V(Employee, age) = 50 \)

\( \sigma_{age=48}(Employee) = ? \)

**Estimate:** \( T(Employee) / V(Employee, age) = 500 \)

<table>
<thead>
<tr>
<th>Age:</th>
<th>0..20</th>
<th>20..29</th>
<th>30-39</th>
<th>40-49</th>
<th>50-59</th>
<th>&gt; 60</th>
</tr>
</thead>
<tbody>
<tr>
<td>( T = )</td>
<td>200</td>
<td>800</td>
<td>5000</td>
<td>12000</td>
<td>6500</td>
<td>500</td>
</tr>
<tr>
<td>( V = )</td>
<td>3</td>
<td>10</td>
<td>7</td>
<td>6</td>
<td>5</td>
<td>4</td>
</tr>
</tbody>
</table>

**Estimate:** \( 12000/10 = 1200 \) \( 12000/6 = 2000 \)
Types of Histograms

- Eq-Width
- Eq-Depth
- Compressed: store outliers separately
- V-Optimal histograms
Employee(ssn, name, age)

Histograms

**Eq-width:**

<table>
<thead>
<tr>
<th>Age:</th>
<th>0..20</th>
<th>20..29</th>
<th>30-39</th>
<th>40-49</th>
<th>50-59</th>
<th>&gt; 60</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tuples</td>
<td>200</td>
<td>800</td>
<td>5000</td>
<td>12000</td>
<td>6500</td>
<td>500</td>
</tr>
</tbody>
</table>

**Eq-depth:**

<table>
<thead>
<tr>
<th>Age:</th>
<th>0..32</th>
<th>33..41</th>
<th>42-46</th>
<th>47-52</th>
<th>53-58</th>
<th>&gt; 60</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tuples</td>
<td>1800</td>
<td>2000</td>
<td>2100</td>
<td>2200</td>
<td>1900</td>
<td>1800</td>
</tr>
</tbody>
</table>

**Compressed:** store separately highly frequent values: (48,1900)
V-Optimal Histograms

• Error:
\[
\sum_{v \in \text{Domain}(A)} \left( \left| |\sigma_{A=v}(R)| - est_{Hist}(\sigma_{A=v}(R)) \right|^2 \right)
\]

• Bucket boundaries = \(\text{argmin}_{Hist}(\text{Error})\)
• Dynamic programming
• Modern databases systems use V-optimal histograms or some variations
Discussion

• Cardinality estimation = still unsolved
• Histograms:
  – Small number of buckets (why?)
  – Updated only periodically (why?)
  – No 2d histograms (except db2) why?
• Samples:
  – Fail for low selectivity estimates, joins
• Cross-join correlation – still unsolved