CSE544
Data Management
Lecture11
Query Optimization – Part 2
Announcements

• Project:
  – Feedback on proposals posted
  – Milestone due on Monday 2/26
• Wednesday 2/14: review 3 due
• Monday 2/19: no lecture (holiday)
• Wednesday 2/21: no lecture
• Friday 2/23: makeup lecture, HW3 due
Query Optimization

Three major components:

1. Search space  
   - last week

2. Cardinality and cost estimation  
   - today

3. Plan enumeration algorithms  
   - today+Wed.
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
Cardinality Estimation

• Traditional method:
  – Simple probability space on the set of possible tuples

• Alternate methods:
  – Based on ML
  – Based on sampling
Traditional Cardinality Estimation
Statistics on Base Data

- Number of tuples (cardinality) \( T(R) \)
- Number of physical pages \( B(R) \)
- Indexes, number of keys in the index \( V(R,a) \)
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
Probability Space

```
SELECT *                  --- attrs A1, A2, …, An
FROM R1, R2, …, Rm
WHERE condition
```
Probability Space

SELECT * --- attrs A1, A2, …, An
FROM R1, R2, …, Rm
WHERE condition

Probability space:
• Outcomes: $R_1 \times R_2 \times \cdots \times R_m$
• Probability: $p(A_1, A_2, \ldots, A_n)$

Cardinality estimate: $Est = p(condition) \cdot T(R_1) \cdots T(R_m)$
Probability Space

Probability space:
- Outcomes: $R_1 \times R_2 \times \cdots \times R_m$
- Probability: $p(A_1, A_2, \ldots, A_n)$

Cardinality estimate: $Est = p(condition) \cdot T(R_1) \cdots T(R_m)$

Goal: compute $p(condition)$
Assumptions

• Uniformity

• Independence

• Containment of values

• Preservation of values
Selectivity Factors

\( p(\text{pred}) \) is called \textit{selectivity factor} \( \theta \)

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

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

**Uniformity assumption**

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

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

**Independence assumption**
• $\theta_{\text{pred1 and pred2}} = \theta_{\text{pred1}} \times \theta_{\text{pred2}} = 1/V(R,A) \times 1/V(R,B)$
Example

SELECT * 
FROM Supplier 
WHERE scity = ‘Seattle’ 

Est = ????

T(Supplier) = 100000 
V(Supplier, scity) = 2000 
V(Supplier, sstate) = 50
Example

```
SELECT *
FROM Supplier
WHERE scity = 'Seattle'
```

```
T(Supplier) = 100000
V(Supplier, scity) = 2000
V(Supplier, sstate) = 50
```

Est = $\frac{1}{2000} \times 100000 = 50$
Example

```
SELECT * 
FROM Supplier 
WHERE scity = 'Seattle'
```

Est = $\frac{1}{2000} \times 100000 = 50$

```
SELECT * 
FROM Supplier 
WHERE sstate = 'WA'
```

Est = $\frac{1}{50} \times 100000 = 2000$

$T(Supplier) = 100000$

$V(Supplier, scity) = 2000$

$V(Supplier, sstate) = 50$
Example

```sql
SELECT * FROM Supplier WHERE scity = 'Seattle'
```

Est = \( \frac{1}{2000} \times 100000 = 50 \)

```sql
SELECT * FROM Supplier WHERE sstate = 'WA'
```

Est = \( \frac{1}{50} \times 100000 = 2000 \)

```sql
SELECT * FROM Supplier WHERE scity = 'Seattle' and sstate = 'WA'
```

Est = ????
Example

Supplier(sid, sname, scity, sstate)

SELECT *
FROM Supplier
WHERE scity = 'Seattle'

SELECT *
FROM Supplier
WHERE sstate = 'WA'

SELECT *
FROM Supplier
WHERE scity = 'Seattle'
and sstate = 'WA'

T(Supplier) = 100000
V(Supplier, scity) = 2000
V(Supplier, sstate) = 50

Est = 1/2000 * 100000 = 50

Est = 1/50 * 100000 = 2000

Est = 1/2000 * 1/50 * 100000 = 1

Independence
Example

Supplier($id$, $name$, $city$, $state$)

\[ T(Supplier) = 100000 \]
\[ V(Supplier, city) = 2000 \]
\[ V(Supplier, state) = 50 \]

\[
\text{SELECT *}
\text{FROM Supplier}
\text{WHERE city = 'Seattle'}
\]

\[ \text{Est} = \frac{1}{2000} \times 100000 = 50 \]

\[
\text{SELECT *}
\text{FROM Supplier}
\text{WHERE state = 'WA'}
\]

\[ \text{Est} = \frac{1}{50} \times 100000 = 2000 \]

\[
\text{SELECT *}
\text{FROM Supplier}
\text{WHERE city = 'Seattle' and state = 'WA'}
\]

\[ \text{Est} = \frac{1}{2000} \times \frac{1}{50} \times 100000 = 1 \]

Very wrong

Why?

Independence
Selectivity Factors

Join

• \( \theta_{R.A=S.B} = \frac{1}{\operatorname{MAX}(V(R,A), V(S,B))} \)

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
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 $\text{Est}(R \bowtie_{A=B} S) = T(R) T(S) / V(S,B)$
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 $\text{Est}(R \bowtie_{A=B} S) = T(R) \cdot T(S) / V(S,B)$

In general:

- $\text{Est}(R \bowtie_{A=B} S) = T(R) \cdot T(S) / \max(V(R,A),V(S,B))$
- $\theta_{R.A=S.B} = 1/ (\max(V(R,A), V(S,B))$
Example

SELECT *
FROM Supplier x, Supply y
WHERE x.sid = y.sid

What is the exact output size?
Example

```
SELECT *
FROM Supplier x, Supply y
WHERE x.sid = y.sid
```

What is the exact output size? T(Supply)=5000000

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

T(Supplier) = 100000
V(Supplier, sid) = 100000

T(Supply) = 5000000
V(Supply, sid) = 80000
SELECT * 
FROM Supplier x, Supply y 
WHERE x.sid = y.sid 

What is the exact output size? 

\[ \text{Est}(\text{Supplier} \bowtie \text{Supply}) = \frac{T(\text{Supplier})T(\text{Supply})}{\max(V(\text{Supplier}, \text{sid}), V(\text{Supply}, \text{sid}))} \]
Example

```
SELECT *
FROM Supplier x, Supply y
WHERE x.sid = y.sid
```

What is the **exact** output size?

\[
\text{Est}(\text{Supplier} \bowtie \text{Supply}) = \frac{T(\text{Supplier}) T(\text{Supply})}{\max(V(\text{Supplier}, \text{sid}), V(\text{Supply}, \text{sid}))}
\]

T(Supplier) = 100000
V(Supplier, sid) = 100000

T(Supply) = 50000000
V(Supply, sid) = 80000

Why? Always larger
Example

SELECT *
FROM Supplier x, Supply y
WHERE x.sid = y.sid

What is the exact output size?

\[ \text{Est}(\text{Supplier} \bowtie \text{Supply}) = \frac{T(\text{Supplier})T(\text{Supply})}{\max(V(\text{Supplier}, \text{sid}), V(\text{Supply}, \text{sid}))} \]

Always larger

Why?
Example

```
SELECT * 
FROM Supplier x, Supply y
WHERE x.sid = y.sid
```

What is the **exact** output size?

\[
\text{Est}(\text{Supplier} \bowtie \text{Supply}) = \frac{T(\text{Supplier})T(\text{Supply})}{\max(V(\text{Supplier}, \text{sid}), V(\text{Supply}, \text{sid}))}
\]

\[
= \frac{T(\text{Supplier})T(\text{Supply})}{V(\text{Supplier}, \text{sid})} = T(\text{Supply})
\]

Always larger

Why?

T(Supplier) = 100000
V(Supplier, sid) = 100000

T(Supply) = 5000000
V(Supply, sid) = 80000

Supplier(sid, sname, scity, sstate)
Supply(sid, pno, quantity)
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
- Example next...
Logical Query Plan 1

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

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

T(Supplier) = 1000
B(Supplier) = 100
V(Supplier, scity) = 20
V(Supplier, state) = 10

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

M=11
Logical Query Plan 1

Estimated (why?)

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

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

$T = 10000$

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

$M=11$
**Logical Query Plan 1**

### SQL Query

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

### Estimated (why?)

```
T < 1
```

### Join Cardinalities

- \(T(Supply) = 10000\)
- \(B(Supply) = 100\)
- \(V(Supply, pno) = 2500\)
- \(T(Supplier) = 1000\)
- \(B(Supplier) = 100\)
- \(V(Supplier, scity) = 20\)
- \(V(Supplier, state) = 10\)

\(M = 11\)
Logical Query Plan 2

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

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

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

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

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

T(Supplier) = 1000
B(Supplier) = 100
V(Supplier, scity) = 20
V(Supplier, state) = 10

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?

```
M=11
T(Supplier) = 1000
B(Supplier) = 100
V(Supplier, scity) = 20
V(Supplier, state) = 10
T(Supply) = 10000
B(Supply) = 100
V(Supply, pno) = 2500
```

**Logical Query Plan 2**

\[
\pi_{sname}(\sigma_{pno=2}(\sigma_{scity='Seattle' \land sstate='WA'}(\text{Supply}))) \quad \text{T = 5}
\]

\[
\pi_{sname}(\text{Supplier}) \quad \text{T = 4}
\]
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(Supplier) = 1000
B(Supplier) = 100
V(Supplier, scity) = 20
V(Supplier, state) = 10

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

M = 11

Very wrong! Why?
Logical Query Plan 2

\[
\begin{align*}
\Pi_{sname} & \quad \text{Different estimate 😞} \\
\sigma_{pno=2} & \quad T = 4 \quad \text{Supply} \\
\sigma_{scity='Seattle' \wedge sstate='WA'} & \quad T = 5 \quad \text{Supplier} \\
\end{align*}
\]

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

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

\[M = 11\]
Physical Plan 1

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

Total cost:

\[ B(\text{Supply}) + B(\text{Supply})B(\text{Supplier})/(M-1) \]

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

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

\[ M = 11 \]
Physical Plan 1

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

\[ \text{Total cost:} \]
\[ B(\text{Supply}) + \frac{B(\text{Supply})B(\text{Supplier})}{M-1} = 100 + \frac{100 \times 100}{11} = 1100 \]

\[ M = 11 \]
Physical Plan 2

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

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

\[\sigma_{\text{scity}='Seattle'}(\text{Supplier})\]

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

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

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

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

\(\text{M} = 11\)
Physical Plan 2

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

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

\[ \sigma_{\text{scity}='Seattle'}(\text{Supplier}) \]

\[ \text{Cost of Supply(pno)} = 4 \]
\[ \text{Cost of Supplier(scity)} = 50 \]
\[ \text{Total cost: 54} \]
Physical Plan 2

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

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

\[ \sigma_{\text{scity}='Seattle'}(\text{Supplier}) \]

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

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

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

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

\[ T(\text{Supplier}) = 1000 \]  
\[ B(\text{Supplier}) = 100 \]  
\[ V(\text{Supplier}, \text{scity}) = 20 \]  
\[ V(\text{Supplier}, \text{state}) = 10 \]
Physical Plan 3

\[ \Pi_{\text{sname}}(\sigma_{\text{scity}=\text{Seattle} \wedge \text{sstate}=\text{WA}}(\text{Supplier})) \]

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

T(Supplier) = 1000
B(Supplier) = 100
V(Supplier, scity) = 20
V(Supplier, state) = 10

M=11
Physical Plan 3

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

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

Unclustered index lookup
Supply(pno)

\[ T = 4 \]
\[ \sigma_{\text{pno}=2} \]
\[ \text{sid} = \text{sid} \]

Clustered Index join

\[ T = 4 \]

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

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

\[ M = 11 \]
Physical Plan 3

\[
\begin{align*}
\text{\(\Pi_{\text{sname}}\)} & \quad \text{(T = 4)} \\
\sigma_{\text{scity} = 'Seattle' \land \text{sstate} = 'WA'} & \quad \text{(T = 4)} \\
\sigma_{\text{pno} = 2} & \quad \text{(T = 4)} \\
\text{Supply} & \quad \text{Unclustered index lookup} \quad \text{Supply(pno)} \\
\text{Supplier} & \quad \text{Clustered Index join} \\
\end{align*}
\]

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

\[
\begin{align*}
\text{T(Supplier) } &= 10000 \\
\text{B(Supplier) } &= 100 \\
\text{V(Supplier, scity) } &= 20 \\
\text{V(Supplier, state) } &= 10 \\
\text{M} &= 11 \\
\end{align*}
\]

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

• We considered only IO cost; systems also need to estimate the CPU cost

• We assumed that all index pages were in memory: sometimes we need to add the cost of fetching index pages from disk
Histograms

- 1d Histograms mitigate uniformity assumption
- 2d Histograms mitigate independence assumption
1d-Histograms

- Histogram on R.A refines $T(R), V(R,A)$
- Each bucket: $T(\text{bucket}), V(\text{bucket, A})$
1d-Histograms

Employee($ssn$, $name$, $age$)

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

$\sigma_{age=48}(Employee) = ?$
1d-Histograms

Employee(ssn, name, age)

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

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

Estimate: \[ \frac{T(Employee)}{V(Employee, age)} = 500 \]
1d-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>
1d-Histograms

Employee(ssn, name, age)

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

σ_{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>
</tbody>
</table>

Assume V = 10
1d-Histograms

Employee(ssn, name, age)

T(Employee) = 25000, V(Employee, age) = 50
σ_{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>
</tbody>
</table>

Estimate: 12000/10 = 1200

Assume V = 10
1d-Histograms

Employee(ssn, name, age)

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

σ_{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: \( \frac{12000}{10} = 1200 \)
1d-Histograms

Employee(\texttt{ssn}, name, age)

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

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

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

<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>

\textbf{Estimate:} \(\frac{12000}{10} = 1200\) \(\frac{12000}{6} = 2000\)
Types of 1d-Histograms

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

Types of 1d-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 Histogram

- $\text{domain}(A) = \{v_1, ..., v_n\}$

- Have budget of $b+1$ buckets for R.A:
  \[ -\infty < d_1 < d_2 < \cdots < d_b < \infty \]

- Choose $d_1, ..., d_b$ to minimize error of all queries $\sigma_{A=v}(R)$, for $v \in \text{Domain}(A)$
V-Optimal Histogram

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

• Bucket boundaries = \text{argmin}_{\text{Hist}}(\text{Error})

• Dynamic programming
Discussion: 1d-Histograms

• All systems support some forms

• Histograms need to be small to reside in main memory: 1000 – 10000 buckets

• Recomputed periodically, often from samples. E.g Postgres ANALYSE
**2d-Histograms**

```
SELECT *
FROM Supplier
WHERE scity = 'Seattle'
  and sstate = 'WA'
```

Independence only needed within the bucket
Issues with 2d-Histograms

• Limited # of dividers: $\sqrt{1000} - \sqrt{10000}$
• Too many 2d-histograms: $n(n-1)/2$
• Problem: how do we estimate $P(A,B,C)$ given all 1d and 2d-histograms?

• Few systems support 2d-histograms
Yet Another Difficulty

• SQL Queries issued from applications

• Optimized once using `prepare` statement, executed repeatedly

• Constants in the query are not know at optimization time
Jayant Haritsa, ICDE’2019 tutorial

```sql
select
  o_year, sum(case when nation = 'BRAZIL' then volume else 0 end) / sum(volume)
from
(select YEAR(o_orderdate) as o_year,
     l_extendedprice * (1 - l_discount) as volume,
     n2.n_name as nation
from part, supplier, lineitem, orders,
     customer, nation n1, nation n2, region
where p_partkey = l_partkey and s_suppkey = l_suppkey
  and l_orderkey = o_orderkey and o_custkey = c_custkey
  and c_nationkey = n1.n_nationkey
  and n1.n_regionkey = r_regionkey
  and r_name = 'AMERICA'
  and s_nationkey = n2.n_nationkey
  and o_orderdate between '1995-01-01'
  and '1996-12-31'
  and p_type = 'ECONOMY ANODIZED STEEL'
  and s_acctbal ≤ C1 and l_extendedprice ≤ C2 ) as all_nations
group by o_year order by o_year
```

Optimize without knowing C1, C2
Different optimal plans for different C1, C2
Traditional CE -- Summary

• Used by all DBMS

• Assumptions: uniformity, independence,…

• 1d-histograms; sometimes also 2d

• Often underestimate (why?)
Alternate CE Methods: ML and Sampling
ML-Based CE

• A trend in research: use some ML model to do cardinality estimation

• Main hope: remove independence and uniformity assumptions

• Data-driven or Query-driven
Data-Driven Estimators

- Train a generative ML model that represents $p(A,B,C,\ldots)$ from the database instance

- Map query Q to hyperrectangle

- Problem: space is over all attributes
  - “Full outer join”
  - Lots of complications to make this work
Query-Driven Estimators

• Training set is a set of pairs (Q, |Q|)

• All predicates in the WHERE condition need to be featurized, embedded

• Train a discriminative model for Est(Q)
Discussion

Problems with ML-based CE:
• Large model size: 1MB – 1GB
• Inference time varies
• Updates, change in skewness, correlations
• Not explainable

Consensus: not ready for production
Sampling-Based CE

• Main idea: use a sample of the database to estimate the output size

• New probability space: random choices of the sampler v.s. data distribution

• Offline Sampling or Online Sampling
Offline Sampling

- Compute uniform sample $R_{sample} \subseteq R$
- Horwitz-Thompson:

$$Est(Q(DB)) = \frac{|R|}{|R_{sample}|} \cdot \frac{|S|}{|S_{sample}|} \cdots |Q(DB_{sample})|$$

- The missing tuple problem: $Q(DB_{sample}) = 0$
- High variance
Online Sampling

• To estimate $R \bowtie S$:
  – Sample a tuple from $R$
  – Sample a matching tuple from $S$
  – Repeat

• Need to use index on $S$

• WanderJoin: next slides show how to estimate $R \bowtie S \bowtie T$ using random walk
<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td>1</td>
</tr>
<tr>
<td>...</td>
<td>2</td>
</tr>
<tr>
<td>...</td>
<td>1</td>
</tr>
<tr>
<td>...</td>
<td>3</td>
</tr>
<tr>
<td>...</td>
<td>4</td>
</tr>
<tr>
<td>...</td>
<td>2</td>
</tr>
<tr>
<td>...</td>
<td>4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>y</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>y</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>z</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>w</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>y</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>z</td>
<td></td>
<td></td>
</tr>
<tr>
<td>w</td>
<td></td>
<td></td>
</tr>
<tr>
<td>w</td>
<td></td>
<td></td>
</tr>
<tr>
<td>x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>w</td>
<td></td>
<td></td>
</tr>
<tr>
<td>x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>B</td>
<td>R</td>
</tr>
<tr>
<td>----</td>
<td>----</td>
<td>-----</td>
</tr>
<tr>
<td>...</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>4</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>S</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>x</td>
</tr>
<tr>
<td>1</td>
<td>y</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>y</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>z</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>w</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>y</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>T</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>y</td>
<td>...</td>
</tr>
<tr>
<td>y</td>
<td>z</td>
<td>...</td>
</tr>
<tr>
<td>z</td>
<td>w</td>
<td>...</td>
</tr>
<tr>
<td>w</td>
<td>w</td>
<td>...</td>
</tr>
<tr>
<td>w</td>
<td>x</td>
<td>...</td>
</tr>
<tr>
<td>x</td>
<td>w</td>
<td>...</td>
</tr>
<tr>
<td>x</td>
<td>x</td>
<td>...</td>
</tr>
<tr>
<td></td>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>1</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>2</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>1</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>3</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>y</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>y</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>z</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>w</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>y</td>
<td>...</td>
<td></td>
</tr>
<tr>
<td>z</td>
<td>...</td>
<td></td>
</tr>
<tr>
<td>w</td>
<td>...</td>
<td></td>
</tr>
<tr>
<td>w</td>
<td>...</td>
<td></td>
</tr>
<tr>
<td>x</td>
<td>...</td>
<td></td>
</tr>
<tr>
<td>x</td>
<td>...</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>R</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>-----</td>
</tr>
<tr>
<td>A</td>
<td>B</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>4</td>
<td></td>
</tr>
</tbody>
</table>

Need indices on these columns!
Probability of this walk:
\[ p = \frac{1}{|R|} \cdot \frac{1}{2} \cdot \frac{1}{3} \]
### Probability of this walk:

\[ p = \frac{1}{|R|} \cdot \frac{1}{2} \cdot \frac{1}{3} \]

### Estimation:

\[ Est(R \bowtie S \bowtie T) = \frac{1}{p} \]
<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>4</td>
<td></td>
</tr>
</tbody>
</table>

\[ p = \frac{1}{|R|} \cdot \frac{1}{2} \cdot \frac{1}{3} \]

\[ Est(R \bowtie S \bowtie T) = \frac{1}{p} \]

Repeat n random walks, return the avg of the estimates
Discussion

• Sampling based CE have best precision
• However:
  – Offline samples need to be too large
  – Online samples require access to indices

• Some systems use offline sample for single-table predicates only
Query Optimization

Three major components:

1. Search space  
   last week

2. Cardinality and cost estimation  
   today

3. Plan enumeration algorithms  
   today+Wed.
Two Types of Optimizers

• Heuristic-based optimizers

• Cost-based optimizers (next)
Two Types of Plan Enumeration Algorithms

• Dynamic programming *(in class)*
  – Based on System R [Selinger 1979]
  – *Join reordering algorithm*

• Rule-based algorithm *(will not discuss)*
  – Database of rules (=algebraic laws)
  – Usually: dynamic programming
System R Optimizer

For each subquery \( Q \subseteq \{R_1, \ldots, R_n\} \), compute best plan:

- **Step 1:** \( Q = \{R_1\}, \{R_2\}, \ldots, \{R_n\} \)

- **Step 2:** \( Q = \{R_1, R_2\}, \{R_1, R_3\}, \ldots, \{R_{n-1}, R_n\} \)

- ... 

- **Step n:** \( Q = \{R_1, \ldots, R_n\} \)
Details

For each subquery \( Q \subseteq \{R_1, \ldots, R_n\} \) store:

- Estimated Size(\( Q \))
- A best plan for \( Q \): Plan(\( Q \))
- The cost of that plan: Cost(\( Q \))
Details

Step 1: single relations \{R_1\}, \{R_2\}, \ldots, \{R_n\}

- Consider all possible access paths:
  - Sequential scan, or
  - Index 1, or
  - Index 2, or
  - ...

- Keep optimal plan for each “interesting order”
Details

Step $k = 2 \ldots n$:
For each $Q = \{R_{i_1}, \ldots, R_{i_k}\}$
- For each $j = 1, \ldots, k$:
  - Let: $Q' = Q - \{R_{i_j}\}$
  - Let: $Plan(Q') \bowtie R_{i_j}$ $Cost(Q') + CostOf(\bowtie)$
- $Plan(Q), Cost(Q) = $ cheapest of the above
  - Keep separate optimal for “interesting orders”
Discussion

• All database systems implement Selinger’s algorithm for join reorder

• For other operators (group-by, aggregates, difference): rule-based

• Many search strategies beyond dynamic programming
Final Discussion

• Query optimizer = critical part of DBMS
• Search space + Size est + Algorithm
• Ideal: find “optimal” plan
• In practice: avoid “very bad plans”
• Successful because:
  – RA is a set-at-a-time language
  – RA is order-independent
• Next time:
  How good are they?; New approaches