Introduction to Database Systems
CSE 344

Lecture 11:
Basics of Query Optimization and
Query Cost Estimation
Announcements

- HW3 due Wed evening
- HW4 will be out on Wed
- Extra OHs (today and) tomorrow
Review

- What is a disk block? (aka page)

- What is an index?
  - What data structures are used to represent indexes in memory?

- What are clustered/unclustered indexes?
Which Indexes?

- The *index selection problem*
  - Given a table, and a “workload” (big Java application with lots of SQL queries), decide which indexes to create (and which ones NOT to create!)

- Who does index selection:
  - The database administrator DBA
  - Semi-automatically, using a database administration tool
Which Indexes?

• The *index selection problem*
  – Given a table, and a “workload” (big Java application with lots of SQL queries), decide which indexes to create (and which ones NOT to create!)

• Who does index selection:
  – The database administrator DBA
    – Semi-automatically, using a database administration tool

<table>
<thead>
<tr>
<th>ID</th>
<th>fName</th>
<th>lName</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>Tom</td>
<td>Hanks</td>
</tr>
<tr>
<td>20</td>
<td>Amy</td>
<td>Hanks</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
The Index Selection Problem 1

V(M, N, P);

Your workload is this

100000 queries:

SELECT *
FROM V
WHERE N=?

100 queries:

SELECT *
FROM V
WHERE P=?
The Index Selection Problem 1

V(M, N, P);

Your workload is this

100000 queries:

SELECT *
FROM V
WHERE N=?

100 queries:

SELECT *
FROM V
WHERE P=?

What indexes?
The Index Selection Problem 1

V(M, N, P);

Your workload is this

100000 queries:

SELECT *
FROM V
WHERE N=?

100 queries:

SELECT *
FROM V
WHERE P=?

A: V(N) and V(P) (hash tables or B-trees)
The Index Selection Problem 2

V(M, N, P);

Your workload is this
100000 queries:
SELECT * FROM V WHERE N>? and N<?

100 queries:
SELECT * FROM V WHERE P=?

100000 queries:
INSERT INTO V VALUES (?, ?, ?)

What indexes?
The Index Selection Problem 2

V(M, N, P);

Your workload is this

100000 queries:
SELECT * FROM V WHERE N>? and N<?

100 queries:
SELECT * FROM V WHERE P=?

100000 queries:
INSERT INTO V VALUES (?, ?, ?)

A: definitely V(N) (must B-tree); unsure about V(P)
The Index Selection Problem 3

V(M, N, P);

Your workload is this

100000 queries: 1000000 queries: 100000 queries:

SELECT * FROM V WHERE N=?

SELECT * FROM V WHERE N=? and P>?

INSERT INTO V VALUES (?, ?, ?)

What indexes?
The Index Selection Problem 3

V(M, N, P);

Your workload is this

100000 queries: 1000000 queries: 100000 queries:

SELECT * FROM V WHERE N=?

SELECT * FROM V WHERE N=? and P>?

INSERT INTO V VALUES (?, ?, ?)

A: V(N, P)

How does this index differ from:
1. Two indexes V(N) and V(P)?
2. An index V(P, N)?
The Index Selection Problem 4

\[ V(M, N, P); \]

Your workload is this

1000 queries:

\[
\text{SELECT * } \\
\text{FROM } V \\
\text{WHERE } N > ? \text{ and } N < ?
\]

100000 queries:

\[
\text{SELECT * } \\
\text{FROM } V \\
\text{WHERE } P > ? \text{ and } P < ?
\]

What indexes?
The Index Selection Problem 4

V(M, N, P);

Your workload is this
1000 queries:
SELECT * 
FROM V 
WHERE N>? and N<?

100000 queries:
SELECT * 
FROM V 
WHERE P>? and P<?

A: V(N) secondary,  V(P) primary index
Two typical kinds of queries

- **Point queries**
  - What data structure should be used for index?

```sql
SELECT * FROM Movie WHERE year = ?
```

- **Range queries**
  - What data structure should be used for index?

```sql
SELECT * FROM Movie WHERE year >= ? AND year <= ?
```
Basic Index Selection Guidelines

• Consider queries in workload in order of importance

• Consider relations accessed by query
  – No point indexing other relations

• Look at WHERE clause for possible search key

• Consider how each query will be processed
  – Which predicate will be processed first?

• Try to choose indexes that speed-up multiple queries
\begin{verbatim}
SELECT * 
FROM R 
WHERE K>? and K<?
\end{verbatim}

\begin{itemize}
\item Cost
\item Percentage tuples retrieved
\end{itemize}
SELECT * 
FROM R 
WHERE K>? and K<?
SELECT * 
FROM R 
WHERE K>? and K<?
SELECT *  
FROM R  
WHERE K=? and K<?
Choosing Index is Not Enough

• To estimate the cost of a query plan, we still need to consider other factors:
  – How each operator is implemented
  – The cost of each operator
  – Let’s start with the basics
Query Execution
Iterator Interface for Query Operators

- **open()**
  - Initializes operator state
  - Sets parameters such as selection condition

- **next()**
  - Operator invokes get_next() recursively on its inputs
  - Performs processing and produces an output tuple

- **close()**: clean-up state

- (more in 444)
Pipelined Query Execution

\( \sigma_{\text{scity}=\text{Seattle} \text{ and } sstate=\text{WA} \text{ and } pno=2} \)

\( \Pi_{\text{sname}} \)

(On the fly)

(Nested loop)

Suppliers (File scan)

Supplies (File scan)

open()
Pipelined Query Execution

\( \text{(On the fly)} \)

\( \text{(On the fly)} \quad \sigma_{\text{scity}=\text{‘Seattle’} \text{ and } \text{sstate}=\text{‘WA’} \text{ and } \text{pno}=2} \)

\( \text{(Nested loop)} \)

next()

next()

next()

\( \text{π}_{\text{sname}} \)

next()

Suppliers

(File scan)

Supplies

(File scan)
Cost of Reading Data From Disk
Cost Parameters

• Cost = I/O + CPU + Network BW
  – We will focus on I/O in this class

• Parameters:
  – \( B(R) \) = # of blocks (i.e., pages) for relation R
  – \( T(R) \) = # of tuples in relation R
  – \( V(R, a) \) = # of distinct values of attribute a
    • When \( a \) is a key, \( V(R,a) = T(R) \)
    • When \( a \) is not a key, \( V(R,a) \) can be anything \( \leq T(R) \)

• Where do these values come from?
  – DBMS collects statistics about data on disk
Selectivity Factors for Conditions

• $A = c$  \(/* \sigma_{A=c}(R) */\)
  
  Selectivity = $1/V(R,A)$

• $A < c$  \(/* \sigma_{A<c}(R) */\)
  
  Selectivity = $(c - \min(R, A))/\left(\max(R,A) - \min(R,A)\right)$

• $c_1 < A < c_2$  \(/* \sigma_{c_1<A<c_2}(R) */\)
  
  Selectivity = $(c_2 - c_1)/\left(\max(R,A) - \min(R,A)\right)$
Cost of Reading Data From Disk

- Sequential scan for relation $R$ costs $B(R)$

- Index-based selection
  - Estimate selectivity factor $X$ (see previous slide)
  - Clustered index: $X*B(R)$
  - Unclustered index $X*T(R)$

Note: we ignore I/O cost for index pages.