Indexing Schemes

Tom Kelliher, CS 318

Apr. 19, 2002

1 Administrivia

Announcements

Assignment

Read 13.1–3.

From Last Time

PL/pgSQL lab.

Outline

1. What is indexing?
2. Terminology.
3. Multilevel indexing.

Coming Up

Query processing.

2 Indexing

Consider the Transcript schema: StudId, CrsCode, Semester, Grade. How should this be sorted to find:

1. A particular student's transcript.
2. A roster for a course.
3. Number of students enrolled in a semester.

Individually, collectively?

Questions:

1. How do we decide how much performance is necessary?
2. How do we achieve increased performance?
3. What is the cost of such performance?

General idea:

1. Table is sorted on one (or a set) of attributes.
2. Indices sorted on other attributes may be maintained — increasing performance on queries referencing those attributes.

2.1 Example

Consider Big State University's (BSU) Transcript table:

1. 30,000 students, five courses per semester, 10 year's history maintained.
2. 10 records stored per disk block.

3. Transcript sorted on StudId.

4. How many page accesses needed to determine

   (a) Number of courses student 123456789 has taken.

   (b) Number of students enrolled in CS 436, spring 2002.

   Repeat, assuming an index on (CrsCode, Semester). Costs of this index?

3 Terminology

1. Search key.

2. Index integrated with data; separate index file (key values and rids).

3. Clustered vs. unclustered indices.

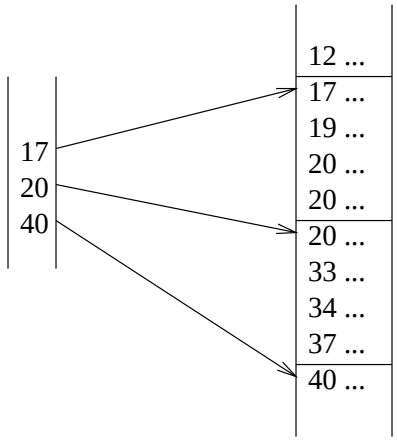
   Is an integrated index clustered on unclustered? What about a separate index? How many clustered indices may be maintained on a table?

4. Inverted, fully inverted tables.

5. Sparse, dense indices.

   Sparse indices must be clustered.

   Illustration of a sparse index:



   Advantages? Disadvantages? Anomalies?

3.1 Search Keys on Multiple Attributes

1. Consider the following index on Transcript: (Semester, CrsCode).
2. How will it help with these queries:
   (a) Ids of all students who took CS 318 S2000.
   (b) Ids of all students who took CS 318 any semester.
   (c) Courses taken in F2001.

   How will it *really* help with this query?

4 Multilevel Indexing

1. What do we do if the index file is large?
2. If indexing is good, why not index the index?
3. Two level indexing: Use a sparse index on the index.
4. Extend to n levels.
5. Terminology:
   (a) Leaf level: If integrated, contains row data. Otherwise, contains key values, rids.
   (b) Separator levels: contain pointers to next level index entries.
   (c) Index level: separator levels *or* leaf level.
   (d) Fan-out: number of separator entries in a page.

4.1 ISAM, B^+ Trees

1. Multilevel indexing schemes.

2. ISAM: Static separator structure, use of overflow buckets. Index may become unbalanced.

3. B^+ trees: Dynamic separator structure. Always balanced.

   How much work is involved in maintaining balance?