Database 2

Lecture II–Part II

Alessandro Artale

Faculty of Computer Science – Free University of Bolzano

Room: 221

artale@inf.unibz.it

http://www.inf.unibz.it/∼artale/

2003/2004 – First Semester

Summary of Lecture II–Part II

- Indexes on Multiple Keys
  - Concatenated keys

Indexes on Multiple Keys

A combination of attributes is frequently used to access the Database. Dedicated index structures are used for efficiently answering such queries.

Example.

`Employee(SSN, DNo, Age, Street, Salary)`

Query: "List all the the employees in **dept.** 4 whose **age** is 59"

Assumption: There is an index on `DNo`, and another on `Age`.

Possible Strategies.

1. Use the index on `DNo` to access the records with `DNo` $= 4$ and then select those with `Age` $= 59$;
2. Analogously, start with the index on `Age`;
3. Use both indexes to find the pertinent pointers and then intersect those pointers in main memory (Best solution).

Indexes on Multiple Keys (cont.)

Disadvantage for all the previous solutions.

If the sets of records that meet each individual condition are large and the intersection is small we need to scan a large number of pointers to produce a small result.

Indexes with Concatenated Keys.

Another strategy is to build a single index by *concatenating* values from each searching attribute to form the search key.

- Let $\{A_1, \ldots, A_n\}$ be the set of searching attributes then the search key values are strings with values $v_1 \otimes v_2 \otimes \ldots \otimes v_n$, with $v_i \in \mathit{Dom}(A_i)$.
- Indexes is sorted considering a lexicographic order on those values.

Any form of previously presented index (i.e., Primary, Secondary, B+Trees, etc.) can be now arranged on concatenated keys as search keys.

Index Definition in SQL

Most SQL implementions support the creation and removal of indexes.

- An index is *created* by the create index command:

  create index ⟨index-name⟩ on ⟨relation-name⟩ (⟨attribute-list⟩)

- To remove an index we use the drop index command:

  drop index ⟨index-name⟩

Example. To define an index with name *branch-index* on the *Account* relation with *branch-name* as the search key, we write:

create index *branch-index* on *Account* (*branch-name*)

If we wish to declare that the search key is also a candidate key:

create unique index *branch-index* on *Account* (*branch-name*)

If *branch-name* is not a candidate key the creation fail. Otherwise, any subsequent attempt to violate the constraint will rise an error.