Lecture 19: Multikey Files

Last few days: Indexed Sequential Files

- Binary & Multiway Trees
- B trees
- B^+ trees

Today: Multikey Files

- Secondary Keys
- Inverted Lists

Folk & Zoellick, ch 6.5 - 6.7

Multikey File Organization

- Direct Files and Indexed Files support efficient data access by a single Field (the key).

- eg, Given an index on Employee Number, the query "Retrieve employee #2317" is fast,

- But, "Retrieve all employees living in Toronto" is slow, since we must scan the entire file.

- Multi key files support fast access by several different fields.

Problem: Given non-key field values, how can we quickly find all records having those field values? (ie, without scanning the entire file.)

Solution: Secondary Key Indices

- A secondary key is a field of a file that is indexed but is not the primary key.

- There are two main secondary index structur
    - Inverted Lists ← today
    - Multilists

Terminology

- The *active values* for a secondary key are the values that the field has in the data file.

Inverted Lists have two parts:

- The *directory* for the secondary key, which has an *entry* for each active value.

- The *accession list* for an active value is a list of pointers to the data file.

Secondary Key Example

loc	E#	NAME	ADDRESS	AGE	SEX	SKILL
1	001	Hicks	Toronto	36	M	Programmer
2	020	McLeod	Montreal	51	M	Analyst
3	023	Lucas	Toronto	25	F	Technician
4	025	Bradley	Ottawa	35	F	Operator
5	030	Date	Montreal	45	M	Operator
6	045	Loomis	Vancouver	45	F	Analyst
7	046	Mader	Edmonton	38	M	Operator
8	048	Wu	Calgary	50	F	Programmer
9	055	Bair	Toronto	28	M	Analyst
10	060	Uhlig	Vancouver	24	M	Technician
11	062	Orilia	Montreal	21	M	Designer
12	070	Fry	Calgary	34	F	Operator
13	075	Riley	Ottawa	40	F	Designer

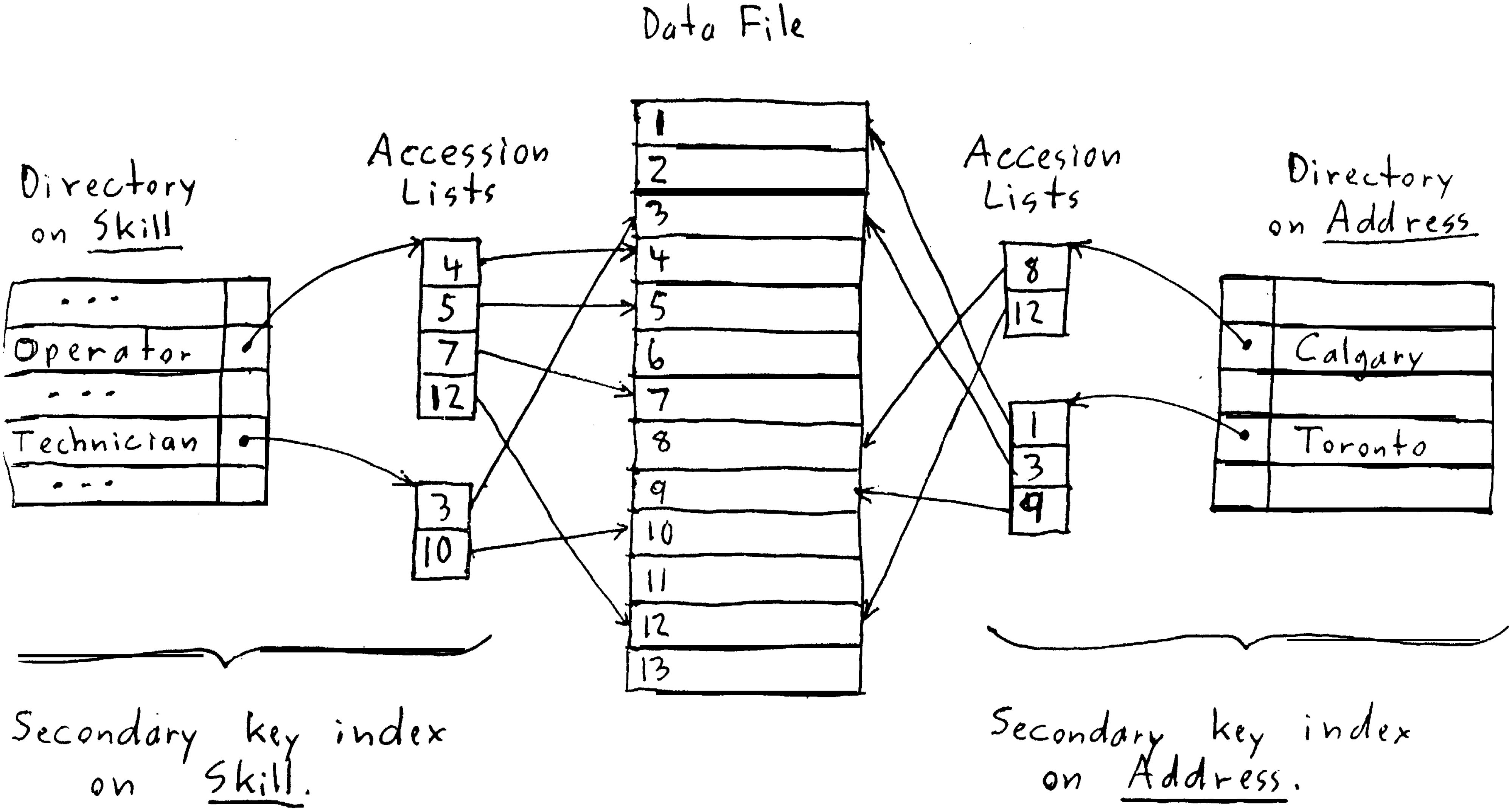
ADDRESS	loc
Calgary	8, 12
Edmonton	7
Montreal	2, 5, 11
Ottawa	4, 13
Toronto	1, 3, 9
Vancouver	6, 10

SKILL	loc
Analyst	2, 6, 9
Designer	11, 13
Operator	4, 5, 7, 12
Programmer	1, 8
Technician	3, 10

Key is E#

Secondary keys are ADDRESS, SKILL

Inverted Lists



Note:

- A secondary key of the data file is a primary key of its directory.

- Directories may be implemented in many ways.

  eg, as arrays in main memory (if small enough),
  as flat files,
  as B trees
  as hash files,
  etc.

Retrieval : Example

- Retrieve all records of operators:

  (1) Find Operator entry in the Skill directory

  (2) Follow the pointer to the accession list.

  (3) Follow each pointer in the accession list to find all operator records in the data file.

- Retrieve all records of employees living in Toronto. (exercise)

Insertion: Example

Insert a record with

E# = 14

Name = Marvin

Address = Toronto

Age = 25

Sex = M

Skill = Technician

Note: All secondary indexes must be updated.

Note: How the record is inserted into the data file depends on how the data file is organized.

- If the data file is serial & unsorted, then just append the new record to the file.
- If the data file is sorted (by E#) then add the record to a differential file
- If there is a B-tree on E#, then insert the record into the B-tree.

After inserting the record into the data file, the secondary indexes must be updated.

Insertion Algorithm

First, insert the new record into the data file.

Then, for each field, F, with a secondary index,

- Let v be the value of field F in the new record
- Insert v into the directory for F (if it is not already there)
- Insert a pointer to the new record into the accession list for v.

Deletion

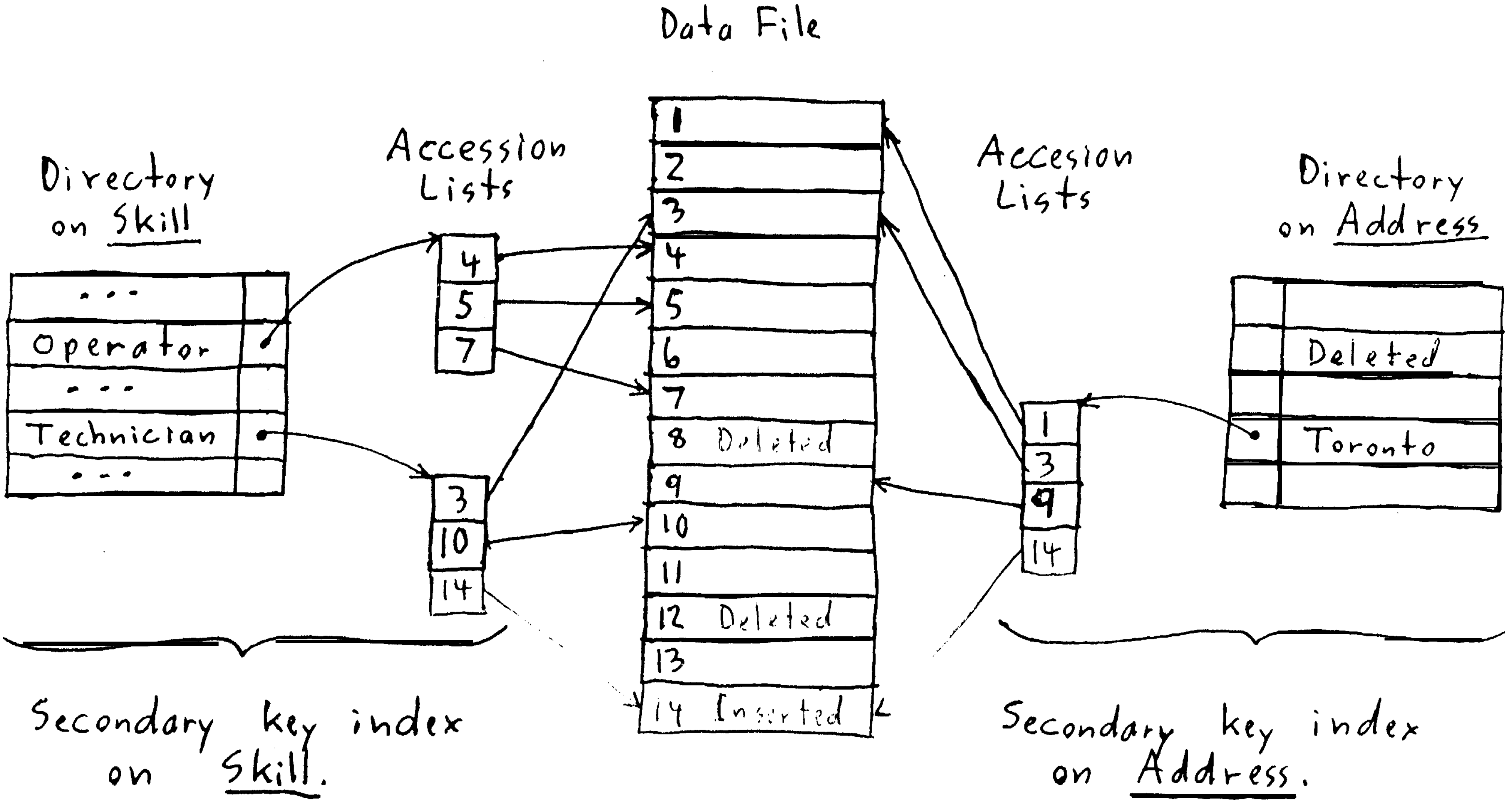
Examples:

- Delete record 12
- Delete record 8 (Accession list for Calgary becomes empty).

Note:

- Again, all secondary indexes must be updated.
- Again, how a record is deleted from the data file depends on how the data file is organized.

Inverted Lists



Final File Organization

Deletion Algorithm

First, delete the record from the data file.

Then, for each field, F, with a secondary index

- Let v be the value of field F in the deleted record.

- Remove the pointer to the deleted record from the accession list for v.

- If the accession list for v is now empty, delete v from the directory for F.