

Basic Organization Methods

(Sequential indexed files)

Contents of Lecture:

- ❖ Basic file organizations
 - 3) Sequential indexed files
 - ✓ Index
 - ✓ Primary Index
 - ✓ Secondary Index
 - ✓ Basic Operations On simple Index
- ❖ Homework number (1)

3) Sequential indexed files

Index:

- ❖ An **index** is a table containing a list of keys associated with a reference field pointing to the record where the information referenced by the key can be found.
- ❖ An index lets you impose order on a file without rearranging the file.
- ❖ Provide direct access for files with variable length records
- ❖ Eliminates the need to re-organize or sort the file
- ❖ There is two type of index:
 - ✓ Simple/ Primary index.
 - ✓ Multiple/Secondary indexes.
- ❖ A simple index is a list of pairs (key, reference), sorted by key.
 - ✓ Reference: relative record address / block address.
 - ✓ Key: Primary / Secondary

Primary Index:

- ❖ Contains a **primary key** and a pointer to a record in the file.
- ❖ Each entry in the primary index identifies uniquely a single record.
- ❖ Support binary search on the primary key.

<u>Index</u>		<u>Data File</u>			
<u>Pk(st-no)</u>	Ref.	Block address	St-no	St-name	GPA
1	25	0	5	Ahmed	2.5
2	15	15	2	Adam	3.1
3	45	25	1	Mona	3.0
5	0	45	3	Ali	2.2

Indexed File

Secondary Index:

- ❖ Identify multiple records.
- ❖ Support binary search on the primary key.

<u>sk(GPA)</u>	Ref.	<u>sk(st-name)</u>	Ref.	Block address	St-no	St-name	GPA
3.1	15	Adam	15	0	5	Ahmed	2.5
3.0	25	Ahmed	0	15	2	Adam	3.1
3.0	75	Ali	45	25	1	Mona	3.0
2.5	0	Mona	25	45	3	Ali	2.2
2.2	45	Omer	75	75	9	Omer	3.0

Secondary indexes

Basic Operations On simple Index:

1. Create the original empty and data files.
2. Load the index file into memory before using it.
3. Rewrite the index file from memory after using it.
4. Add data records to the data file.
5. Delete records from the data file.
6. Update the index to reflect changes in the data file

1. Create the original empty and data files.

- ❖ First; create the data file with its data.

17	LON 2312 Symphony N.S ...
62	RCA 2626 Quartet in C sharp ...
117	WAR 23699 Adagio ...
152	ANG 3795 Violin Concerto ...

Address of
Record

- ❖ Then; create an empty index file with primary key
 - ✓ For example primary key can be:
primary key = record ID only
 - ✓ or combination of two or more records contain primary key in data file
 - ✓ in above example:
Primary key = company label + record ID (LABEL ID).
 - ✓ Then create an empty index file.

Index :

key	Reference field

2. Load the index file into memory before using it.

- ❖ The index is represented as an array of records. The loading into memory can be done sequentially, reading a large number of index records (which are short) at once.

3. Rewrite the index file from memory after using it.

- ❖ If the index changed the new data must be re-writing.

4. Add data records to the data file.

- ❖ First add data to index file, data must be sorted depend to selected primary key

Index :

key	Reference field
ANG3795	152
LON2312	17
RCA2626	62
WAR23699	117

- ❖ When we add a record, both the data file and the index should be updated.
- ❖ In the data file, the record can be added anywhere. However, the byte-offset of the new record should be saved.
- ❖ Since the index is sorted, the location of the new record does matter: we have to shift all the records that belong after the one we are inserting to open up space for the new record. However, this operation is not too costly as it is performed in memory.

5. Delete records from the data file.

- ❖ We must delete the corresponding entry from the index
- ❖ The deleted record entry is inserted into **AVAIL LIST** (index for available records)
- ❖ Logical deletion
- ❖ Example:

<u>Avail List</u>		<u>Primary Index</u>		<u>Data File</u>			
<u>Pk(st-no)</u>	<u>Ref.</u>	<u>Pk(st-no)</u>	<u>Ref.</u>	<u>Block address</u>	<u>St-no</u>	<u>St-name</u>	<u>GPA</u>
2	15	1	25	0	5	Ahmed	2.5
		3	45	15	2	Adam	3.1
		5	0	25	1	Mona	3.0
				45	3	Ali	2.2

6. Update the index to reflect changes in the data file

There are two cases to consider:

- ❖ changes the value of the key field:
 - ✓ Treat this as a deletion followed by an insertion
- ❖ does not affect the key field
 - ✓ If record size is unchanged, just modify the data record.
 - ✓ If record size changes treat this as a delete/insert sequence.

Homework number (1):

- ❖ If we have the following data file table

Block	Rec	LicencePlate	Manufacturer	Model	Color
B_0	0	ABC 07-07	Ford	Mustang	white
	1	ZP 13-07	Holden	Vectra	red
B_1	2	ABC 13-13	Holden	Astra	blue
	3	XY 13-07	Holden	Commodore	white
B_2	4	AAA 07-07	Ford	Falcon	black
	5	XP 77-07	Nissan	Primera	black
B_3	6	ABA 89-17	Toyota	Corolla	red
	7	ADA 22-22	Ford	Falcon	red
B_4	8	ZA 13-07	Holden	Vectra	green
	9	ZB 13-07	Nissan	Primera	white
B_5	10	ZP 31-00	Ford	Mustang	white
	11	ZX 93-97	Holden	Vectra	Blue
B_6	12	ZY 03-70	Ford	Mustang	white
	13	ABC 31-31	Holden	Commodore	white

- ❖ And we would like to select records from a file according to an attribute (or a set of attributes), which do not form a key. For example
 “Retrieve White cars manufactured by Ford”
- ❖ Discuss the following:
 1. Create the original empty and data files with the field (LicencePlate + Block + Rec)
 2. Add data records to the data file.
 3. Delete records with the **Rec = 8** from the data file.
 4. Changes the color to white in **Rec = 11** in the data file
 5. Changes the LicencePlate to ABC 07-07 in **Rec = 11** in the data file
 6. Write C code to select the above record.