Database Management Systems

Madhavan Mukund

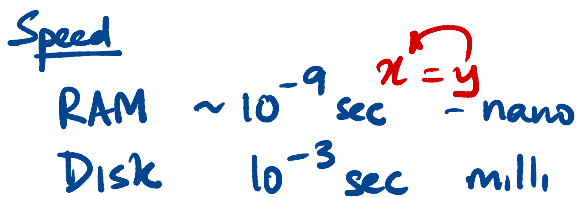
https://www.cmi.ac.in/~madhavan

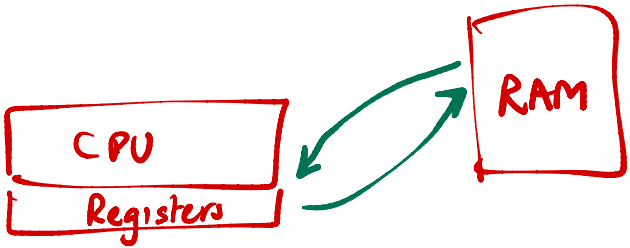
Sai University

Lecture 14, 13 October 2023

Storing data

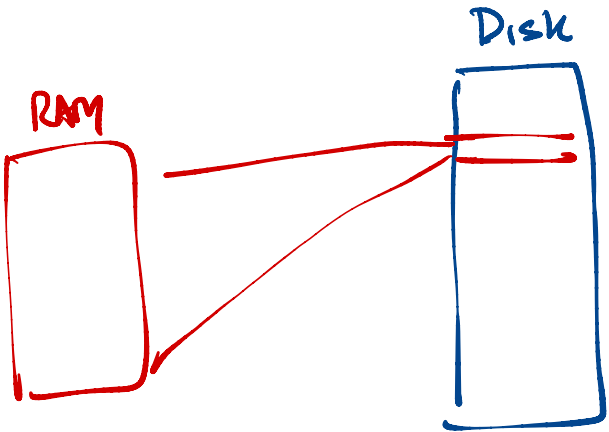
- RAM vs Hard disk vs SSD
- Blocks and latency





Storing data

- RAM vs Hard disk vs SSD
- Blocks and latency



Storing data

- RAM vs Hard disk vs SSD
- Blocks and latency

Hard disk

Locate the data - Seek] Each is 10^{-3}
Fetch it

Seq access - 1 seek + Many fetch
Random access - Many (seek + fetch) - twice as slow

Storing data

- RAM vs Hard disk vs SSD
- Blocks and latency

Seek time

– Unit of transfer

Block ≈ 4KB

Fixed length records

- Blocks and block boundaries

Don't split a row

Record = 1 row in table
= 1 tuple

record 0	10101	Srinivasan	Comp. Sci.	65000
record 1	12121	Wu	Finance	90000
record 2	15151	Mozart	Music	40000
record 3	22222	Einstein	Physics	95000
record 4	32343	El Said	History	60000
record 5	33456	Gold	Physics	87000
record 6	45565	Katz	Comp. Sci.	75000
record 7	58583	Califieri	History	62000
record 8	76543	Singh	Finance	80000
record 9	76766	Crick	Biology	72000
record 10	83821	Brandt	Comp. Sci.	92000
record 11	98345	Kim	Elec. Eng.	80000

Deleting a record

- Compress

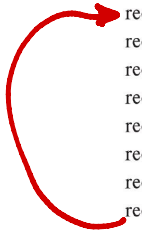
Move rows 4 onwards up one lev.

record 0	10101	Srinivasan	Comp. Sci.	65000
record 1	12121	Wu	Finance	90000
record 2	15151	Mozart	Music	40000
record 4	32343	El Said	History	60000
record 5	33456	Gold	Physics	87000
record 6	45565	Katz	Comp. Sci.	75000
record 7	58583	Califieri	History	62000
record 8	76543	Singh	Finance	80000
record 9	76766	Crick	Biology	72000
record 10	83821	Brandt	Comp. Sci.	92000
record 11	98345	Kim	Elec. Eng.	80000

Deleting a record

- Compress
- Move last record

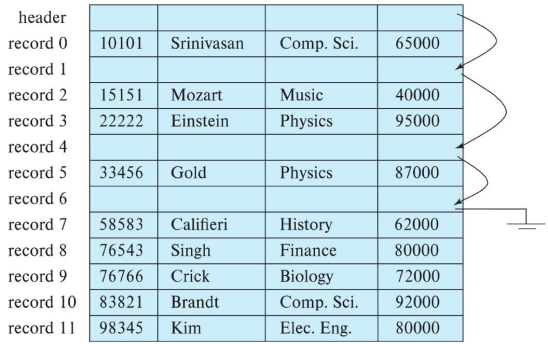
Order disturbed



record 0	10101	Srinivasan	Comp. Sci.	65000
record 1	12121	Wu	Finance	90000
record 2	15151	Mozart	Music	40000
record 11	98345	Kim	Elec. Eng.	80000
record 4	32343	El Said	History	60000
record 5	33456	Gold	Physics	87000
record 6	45565	Katz	Comp. Sci.	75000
record 7	58583	Califieri	History	62000
record 8	76543	Singh	Finance	80000
record 9	76766	Crick	Biology	72000
record 10	83821	Brandt	Comp. Sci.	92000

Deleting a record

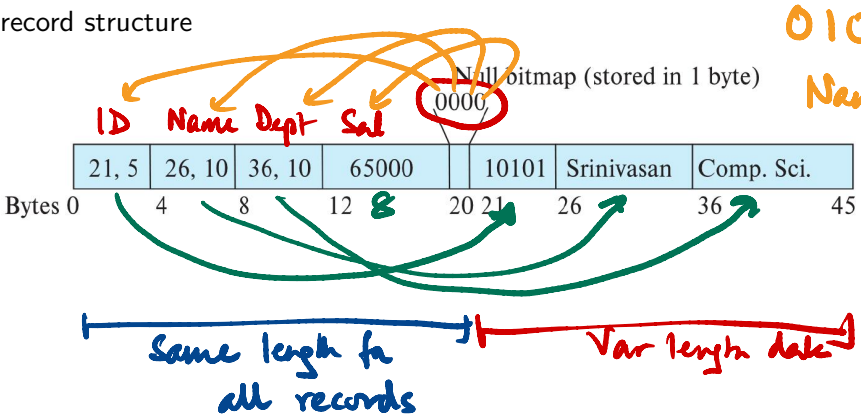
- Compress
- Move last record
- Maintain free list of empty slots



header				
record 0	10101	Srinivasan	Comp. Sci.	65000
record 1				
record 2	15151	Mozart	Music	40000
record 3	22222	Einstein	Physics	95000
record 4				
record 5	33456	Gold	Physics	87000
record 6				
record 7	58583	Califieri	History	62000
record 8	76543	Singh	Finance	80000
record 9	76766	Crick	Biology	72000
record 10	83821	Brandt	Comp. Sci.	92000
record 11	98345	Kim	Elec. Eng.	80000

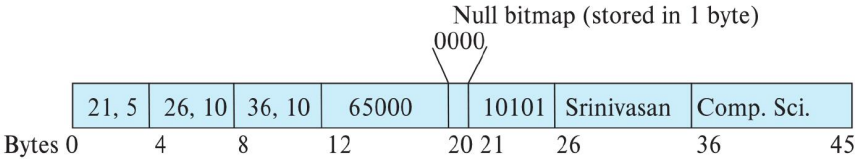
Variable length records

- Single record structure

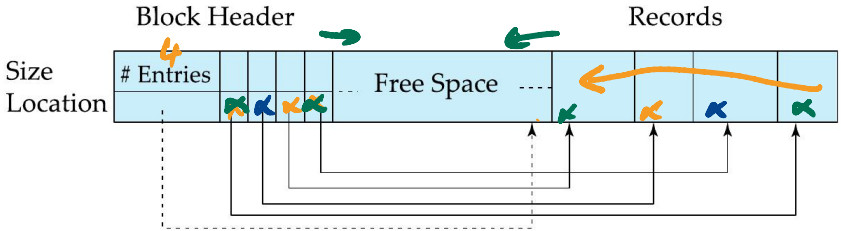


Variable length records

- Single record structure



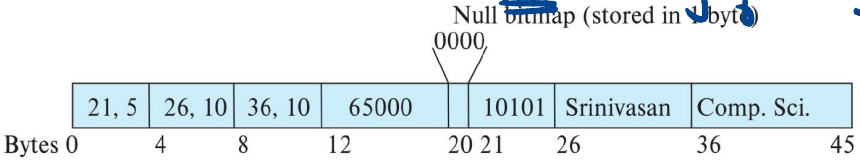
- Slotted page block structure



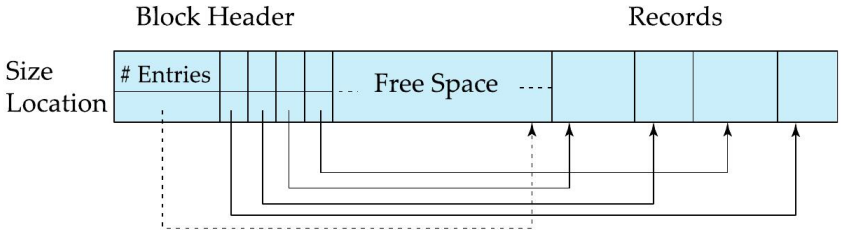
Variable length records

- Single record structure

A block stores many records / rows
A table is a file consisting of many blocks



- Slotted page block structure

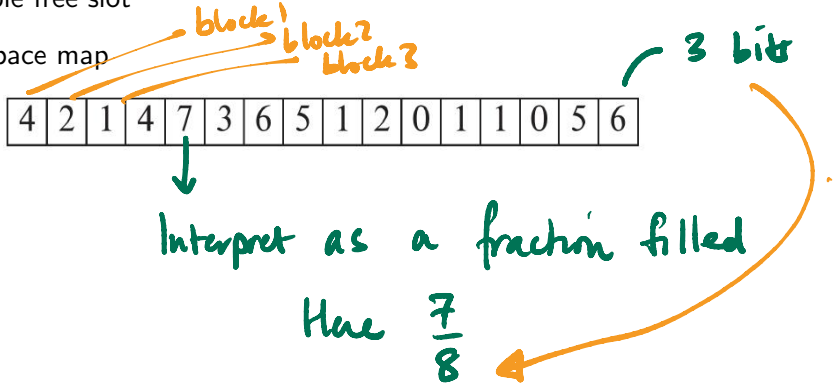


Storing tables — heap file organization

- Use first available free slot

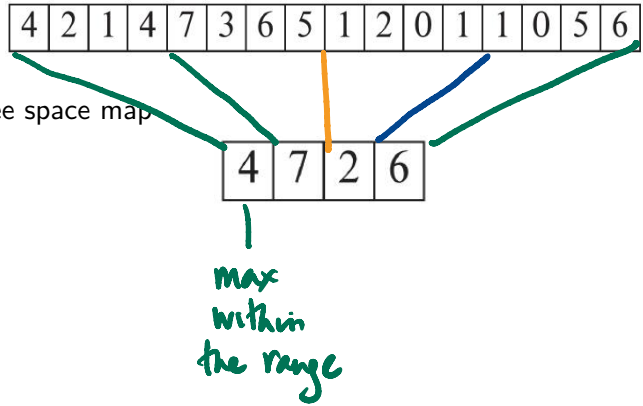
Storing tables — heap file organization

- Use first available free slot
- Maintain free space map



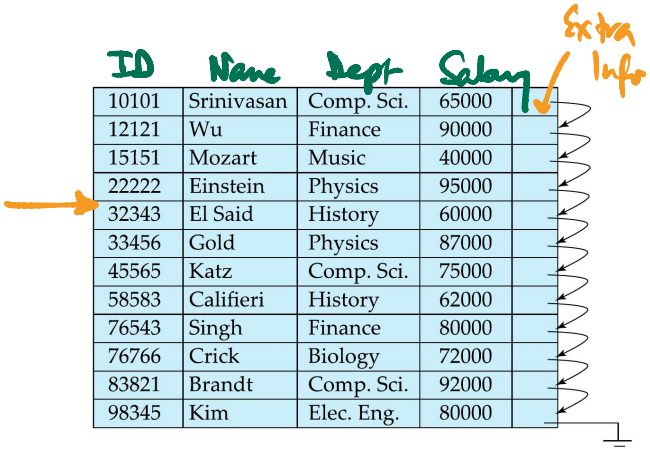
Storing tables — heap file organization

- Use first available free slot
- Maintain free space map
- Second level free space map



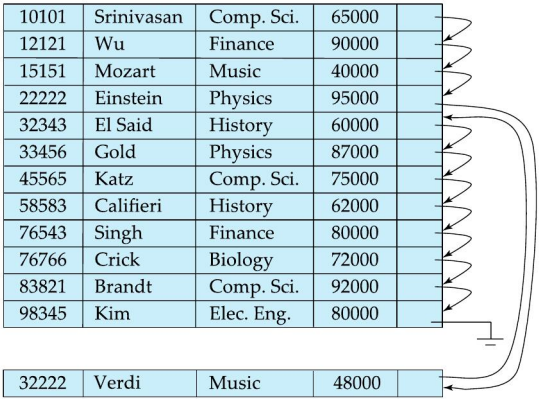
Storing tables — sequential file organization

In some order, here ID



Storing tables — sequential file organization

- Overflow block



10101	Srinivasan	Comp. Sci.	65000	
12121	Wu	Finance	90000	
15151	Mozart	Music	40000	
22222	Einstein	Physics	95000	
32343	El Said	History	60000	
33456	Gold	Physics	87000	
45565	Katz	Comp. Sci.	75000	
58583	Califieri	History	62000	
76543	Singh	Finance	80000	
76766	Crick	Biology	72000	
83821	Brandt	Comp. Sci.	92000	
98345	Kim	Elec. Eng.	80000	

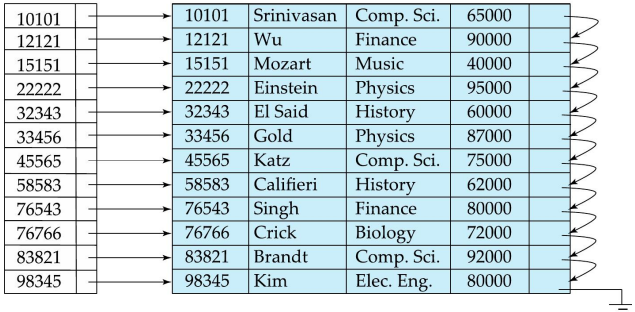
32222	Verdi	Music	48000	

Indexing — dense indices

Every data value is indexed

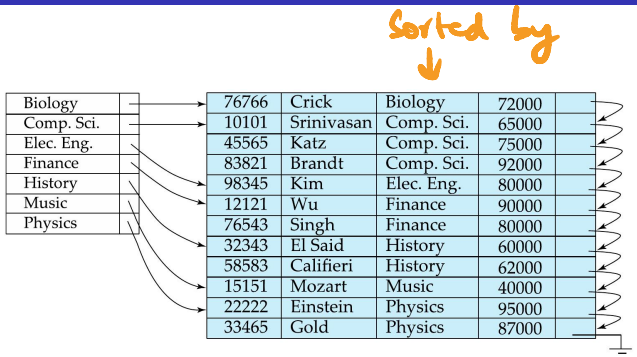
Smaller

Potentially have index for full table in memory at one time

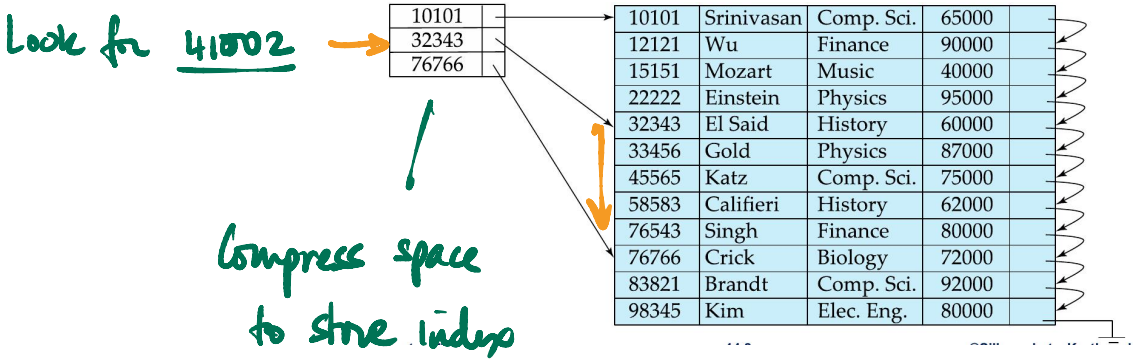


10101	10101	Srinivasan	Comp. Sci.	65000	
12121	12121	Wu	Finance	90000	
15151	15151	Mozart	Music	40000	
22222	22222	Einstein	Physics	95000	
32343	32343	El Said	History	60000	
33456	33456	Gold	Physics	87000	
45565	45565	Katz	Comp. Sci.	75000	
58583	58583	Califieri	History	62000	
76543	76543	Singh	Finance	80000	
76766	76766	Crick	Biology	72000	
83821	83821	Brandt	Comp. Sci.	92000	
98345	98345	Kim	Elec. Eng.	80000	

Indexing — dense indices



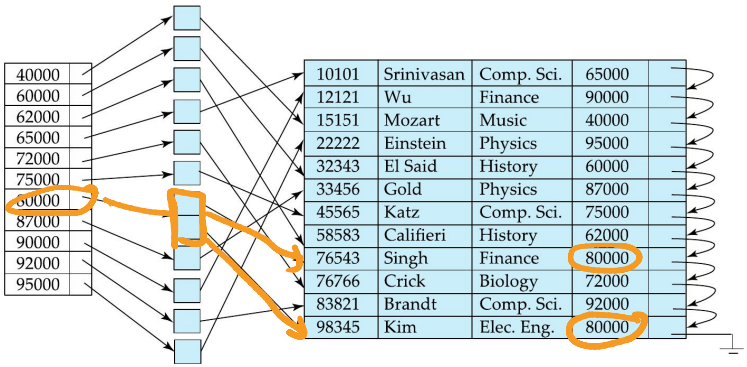
Indexing — sparse indices



Indexing — secondary index

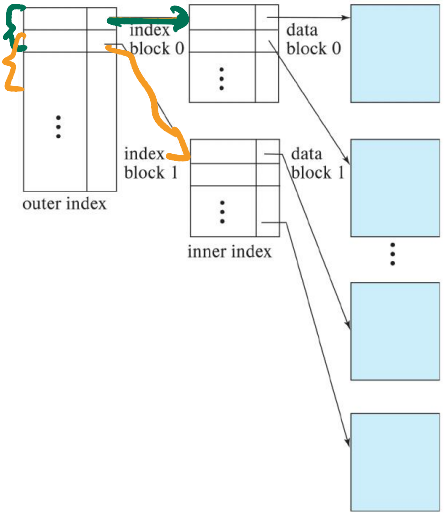
file is not sorted on the given column

One block per value in Salary



Recall library card catalogue

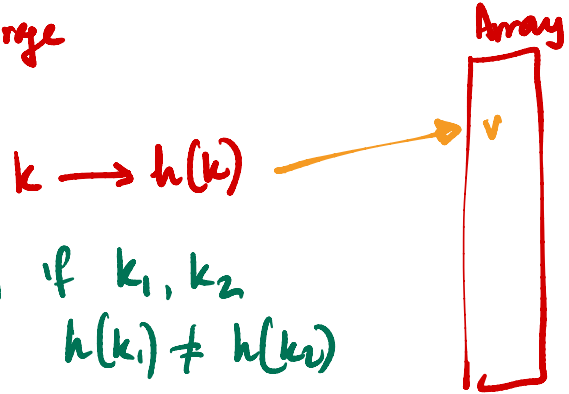
Indexing — multilevel index



Hashing

Dictionary $\{k_1 : v_1, k_2 : v_2, \ldots, k_n : v_n\}$

Storage



$k \longrightarrow h(k)$

Ideally if k_1, k_2
$h(k_1) \neq h(k_2)$

Avoid collisions
$h(k_1) = h(k_2)$

SHA 256 Maps any string to 256 bytes

8^{256} value

2^{768}

Dropbox