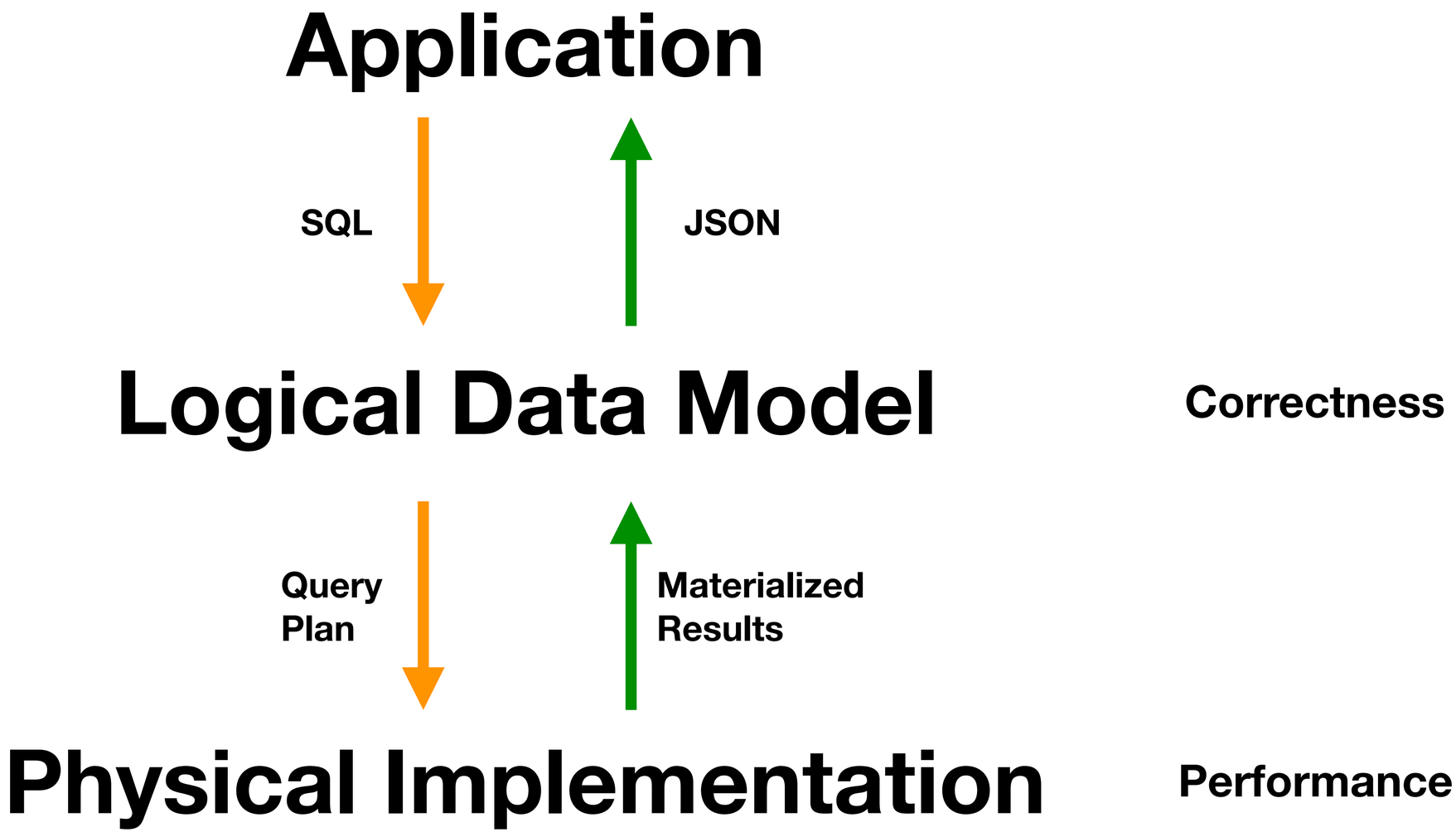
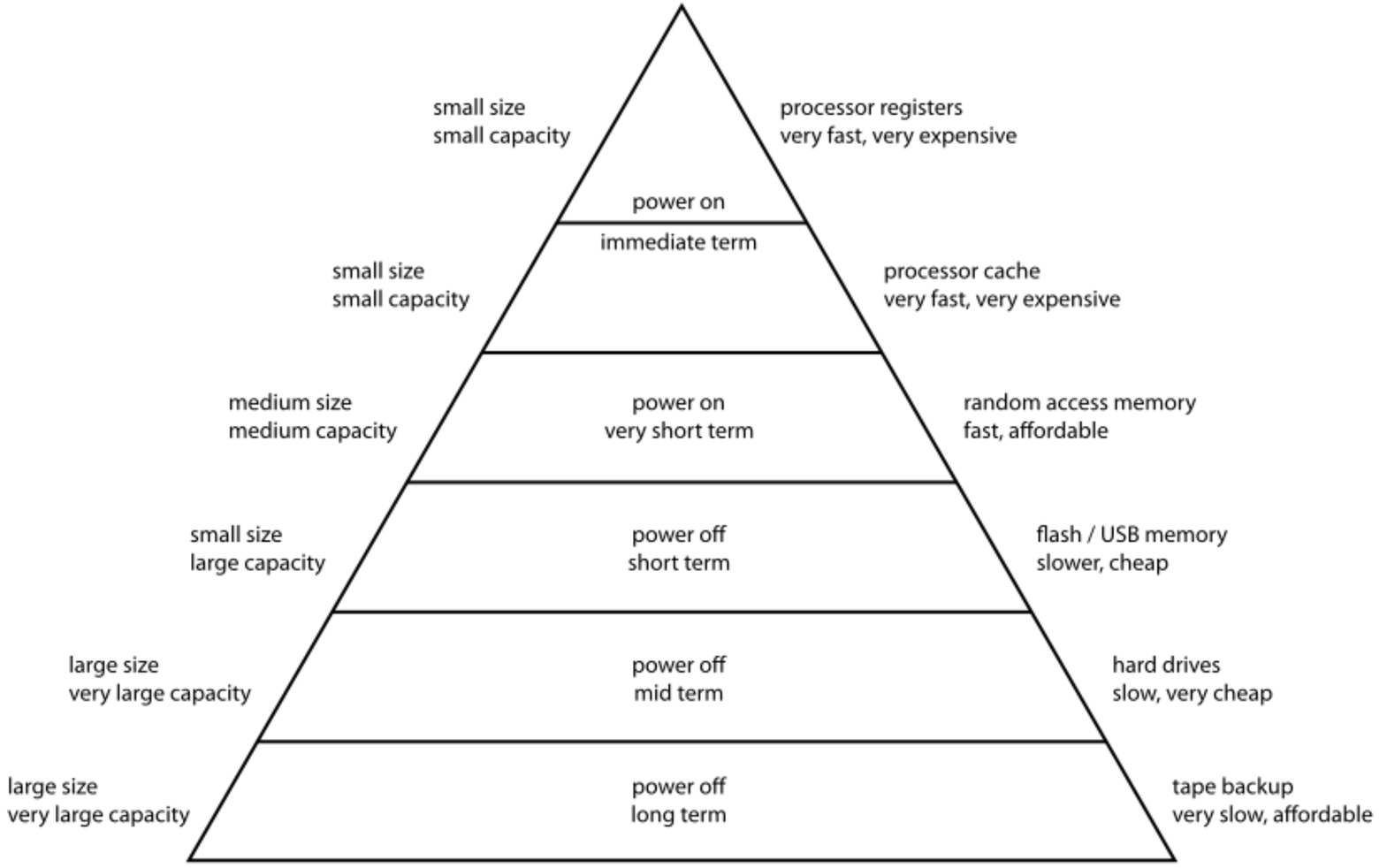
Storage and File Organization

Wherein we start looking inside the database


Application
SQL
JSON
Logical Data Model
Correctness
Query
Plan
Materialized
Results
Physical Implementation
Performance

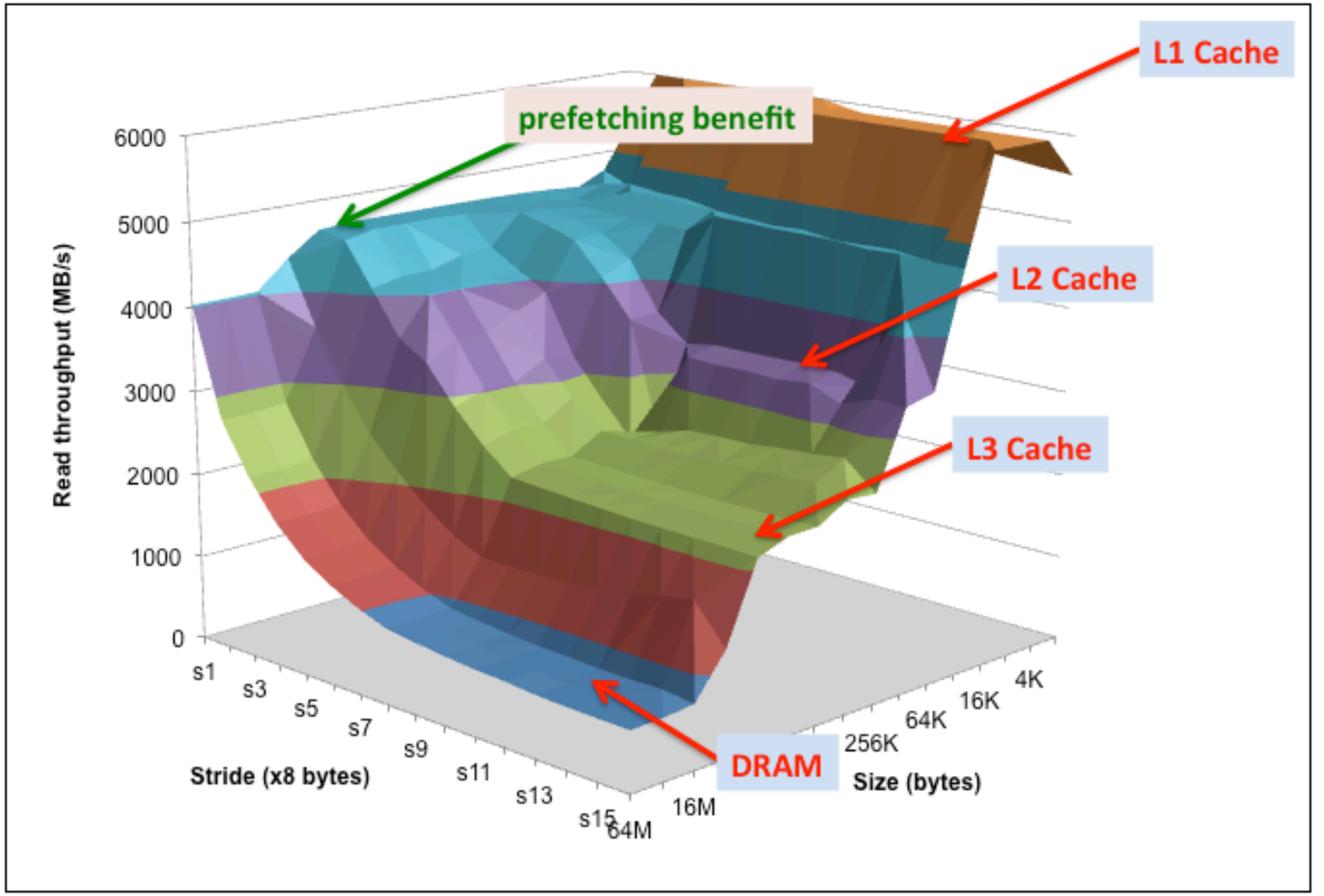

Memory Hierarchy
Performance
Access Patterns

Computer Memory Hierarchy



https://en.wikipedia.org/wiki/Memory_hierarchy#/media/File:ComputerMemoryHierarchy.svg

CSAPP2e: Core i7 (2009 Nehalem)



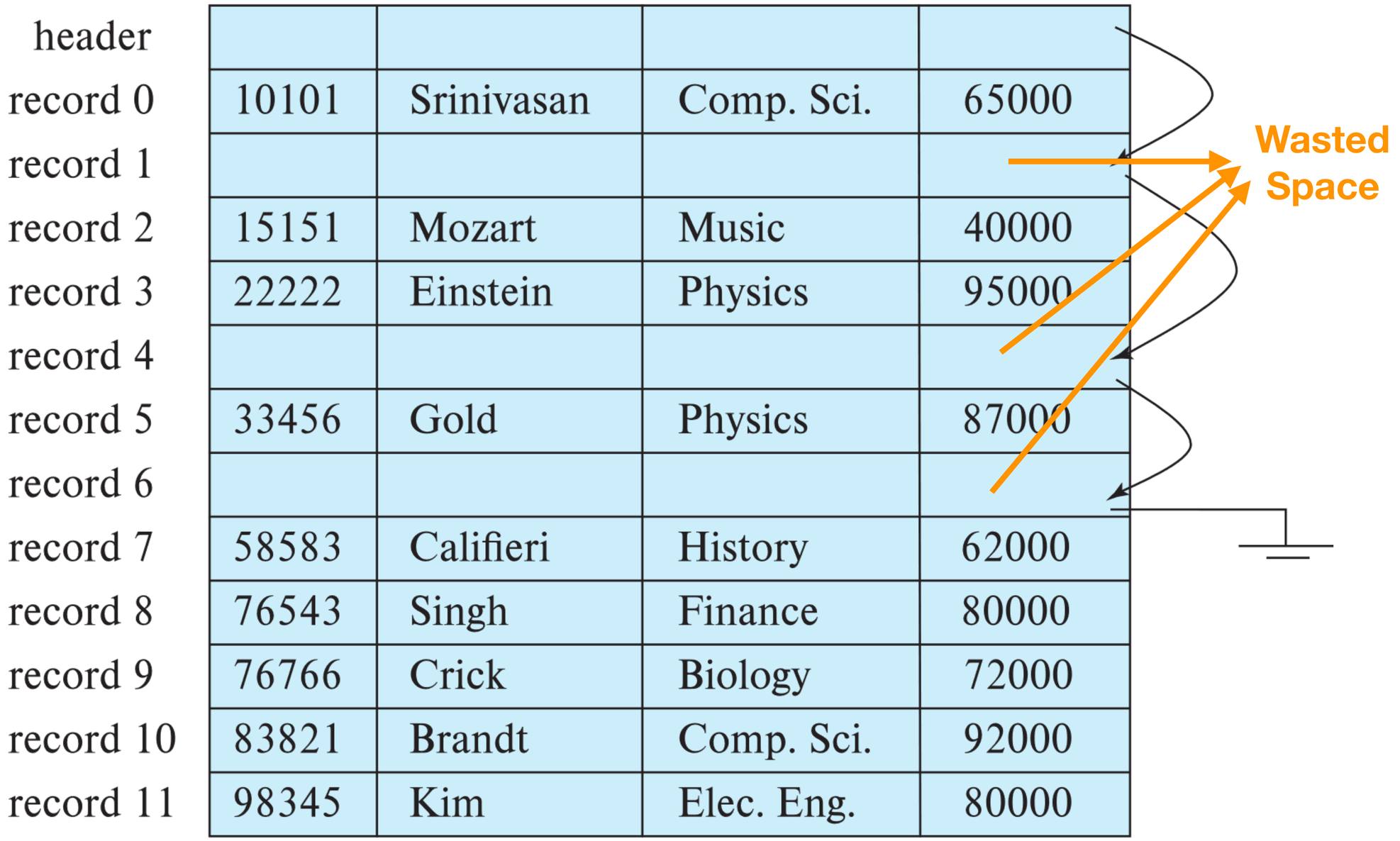
http://csappbook.blogspot.com/2017/05/a-gallery-of-memory-mountains.html

Storing and Organizing Fixed Length Tuples

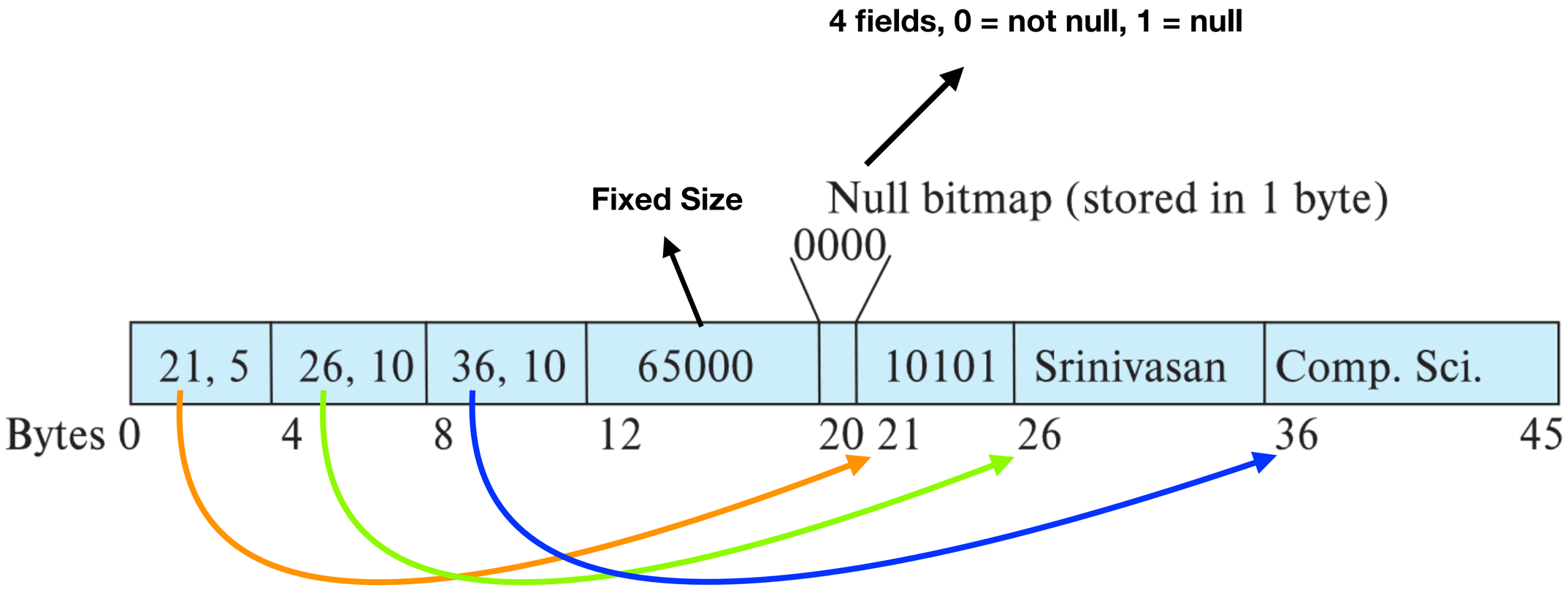
Delete record 3

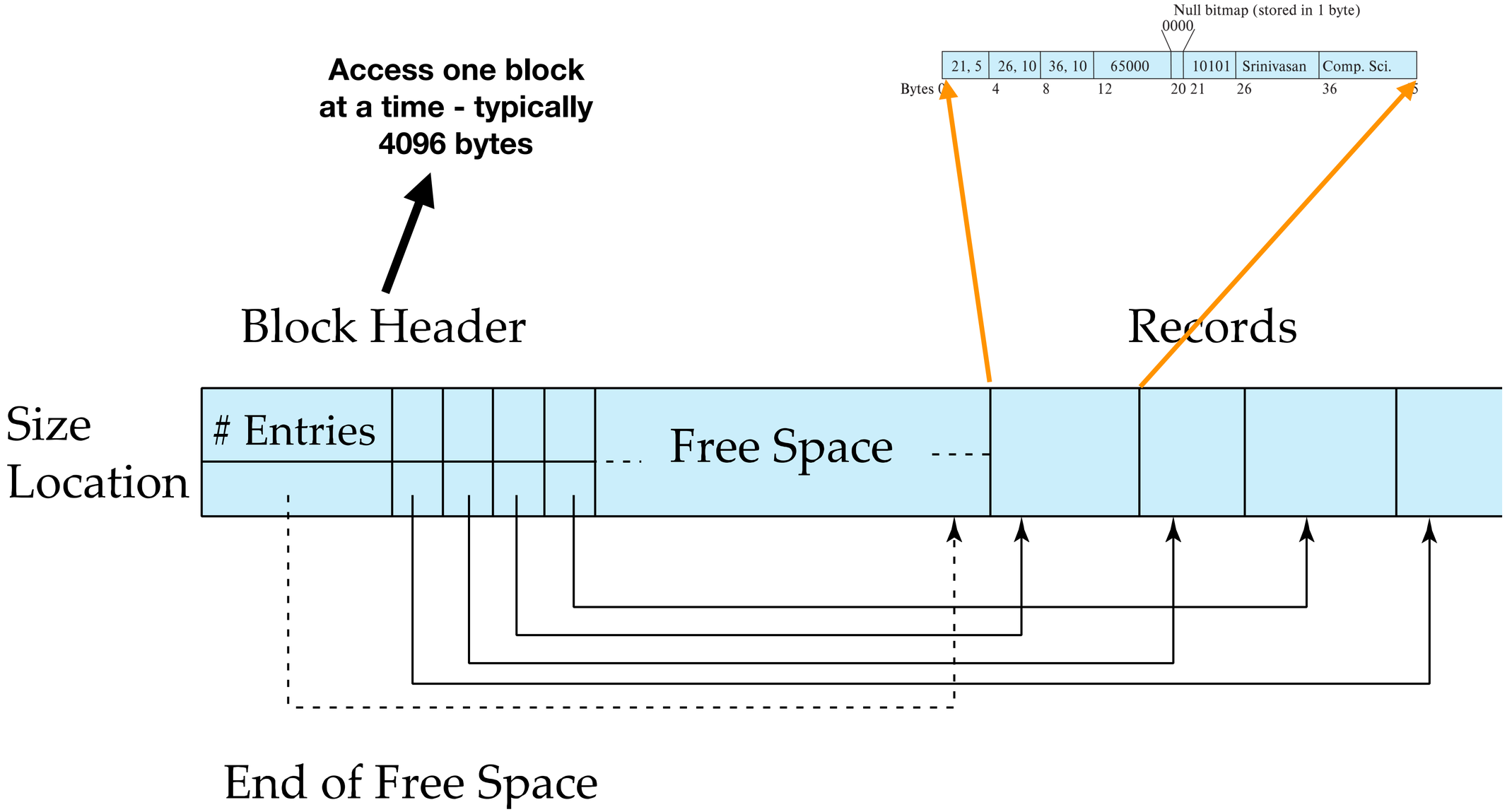
record 0	10101	Srinivasan	Comp. Sci.	65000
record 1	12121	Wu	Finance	90000
record 2	15151	Mozart	Music	40000
~~record 3~~	~~22222~~	~~Einstein~~	~~Physics~~	~~95000~~
record 4	32343	El Said	History	60000
record 5	33456	Gold	Physics	87000
record 6	45565	Katz	Comp. Sci.	75000
record 7	58583	Califieri	History	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

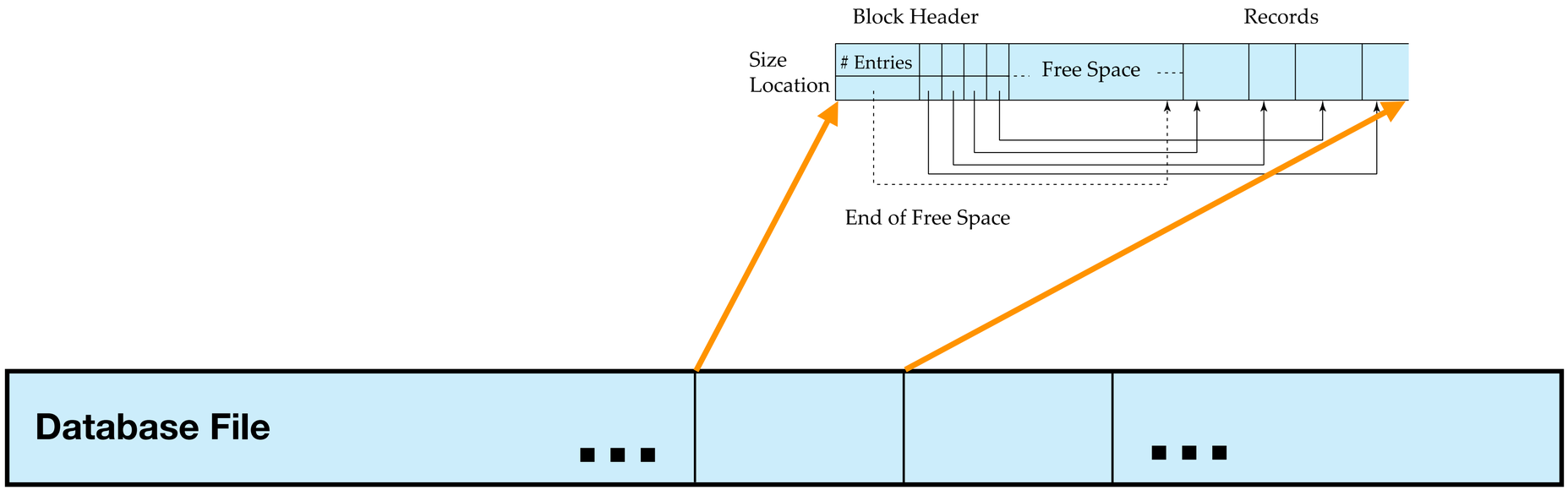
record 0	10101	Srinivasan	Comp. Sci.	65000
record 1	12121	Wu	Finance	90000
record 2	15151	Mozart	Music	40000
record 4	32343	El Said	History	60000
record 5	33456	Gold	Physics	87000
record 6	45565	Katz	Comp. Sci.	75000
record 7	58583	Califieri	History	62000
record 8	76543	Singh	Finance	80000
record 9	76766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


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

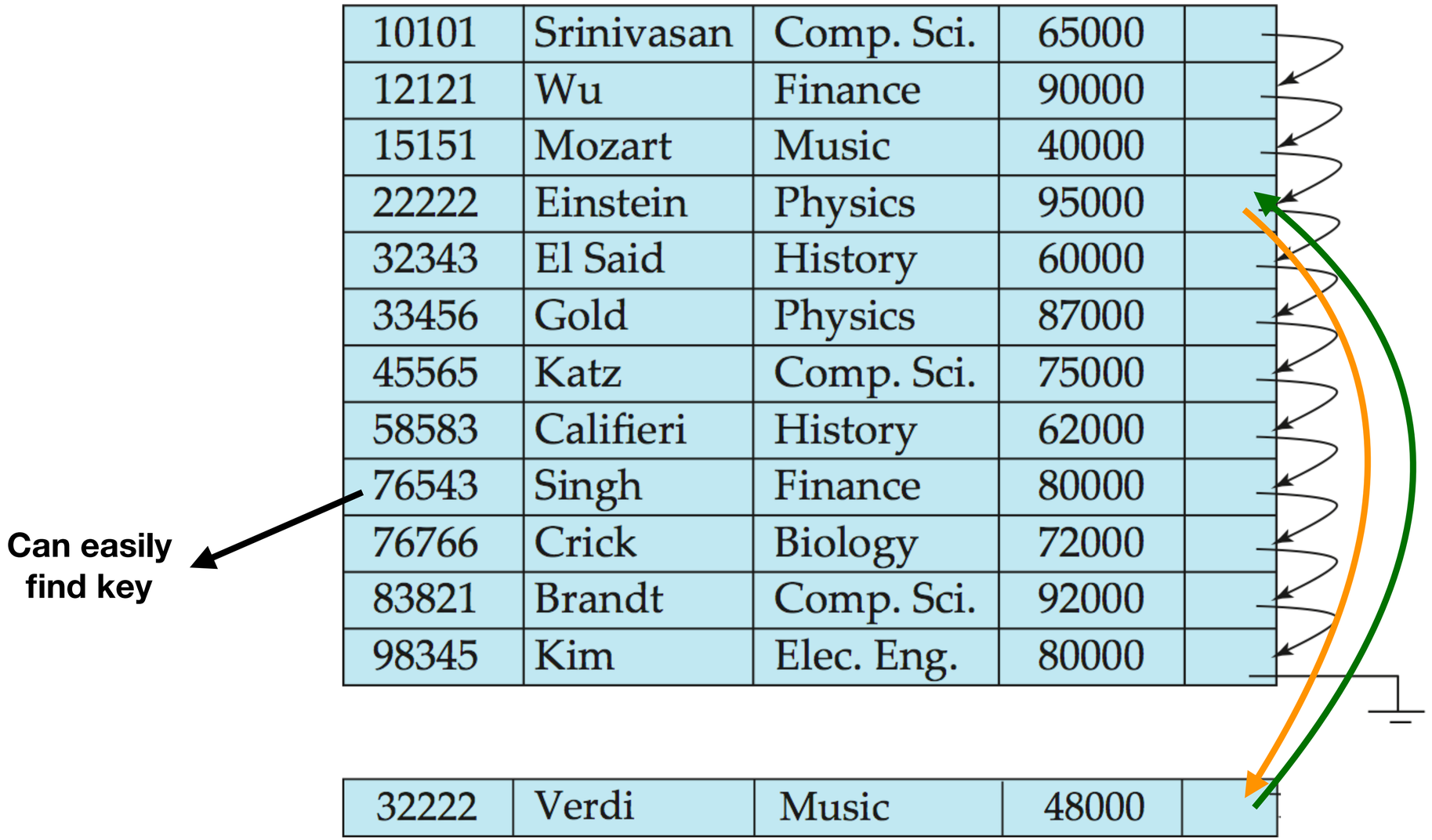
Storing and Organizing Variable Length Tuples


4 fields, 0 = not null, 1 = null
Fixed Size
Null bitmap (stored in 1 byte)
0000
21, 5
26, 10
36, 10
65000
10101
Srinivasan
Comp. Sci.
Bytes 0
4
8
12
20
21
26
36
45


Access one block at a time - typically 4096 bytes
Block Header
Null bitmap (stored in 1 byte)
0000
21, 5
26, 10
36, 10
65000
10101
Srinivasan
Comp. Sci.
Bytes 0
4
8
12
20
21
26
36
Records
Size
Location
Entries
Free Space
End of Free Space


Block Header
Records
Size
Location
Entries
Free Space
End of Free Space
Database File

Order of tuples in a file: Sequential Order


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
Can easily find key
32222 Verdi Music 48000

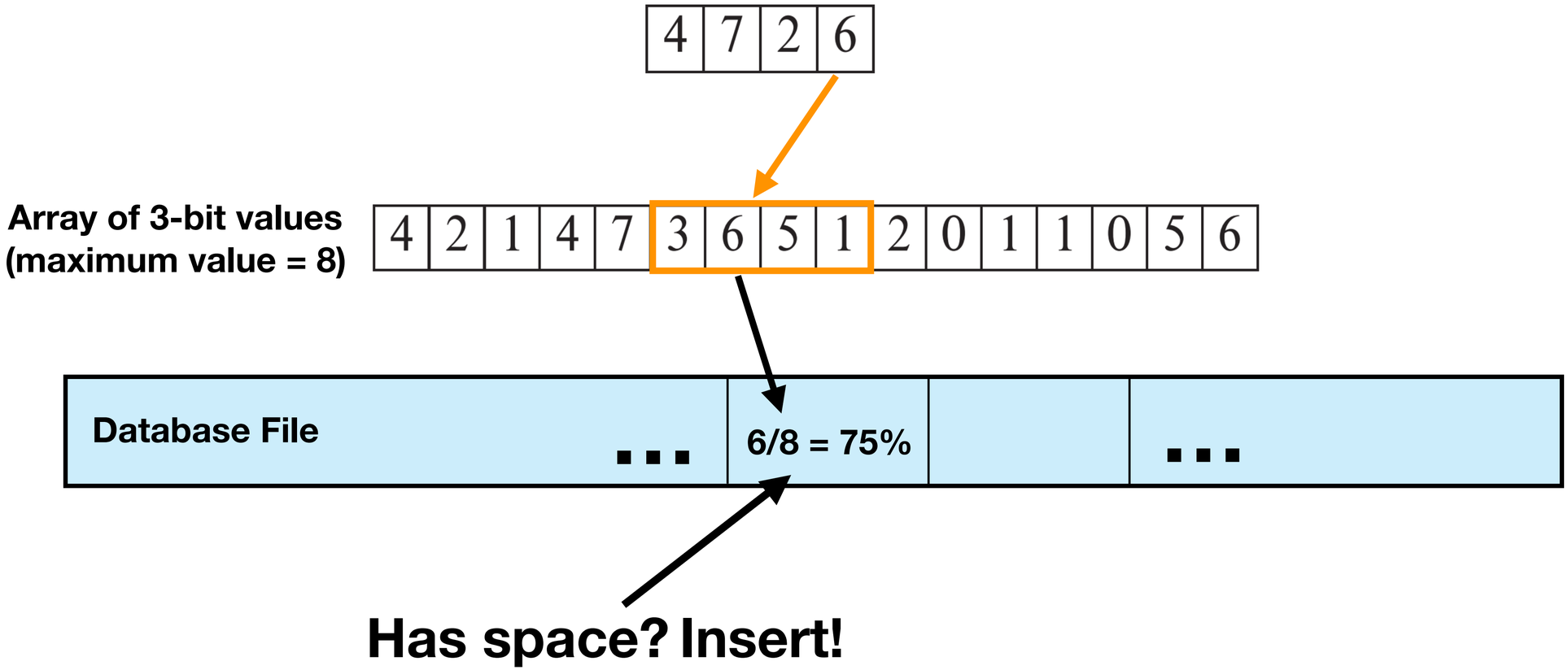
Order of tuples in a file: Multitable Clustering

dept_name	*building*	*budget*
Comp. Sci.	Taylor	100000
Physics	Watson	70000

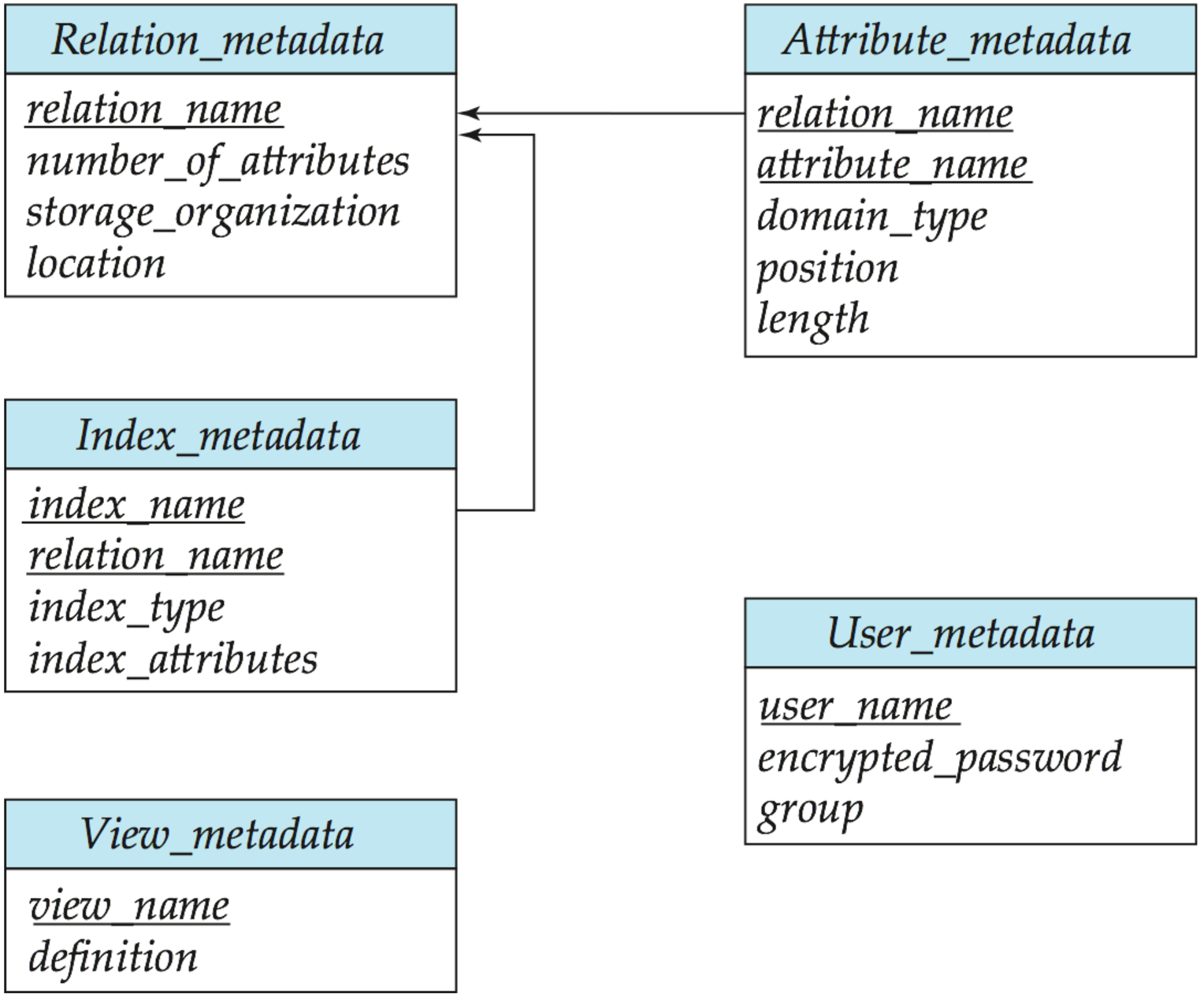
ID	*name*	*dept_name*	*salary*
10101	Srinivasan	Comp. Sci.	65000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
83821	Brandt	Comp. Sci.	92000

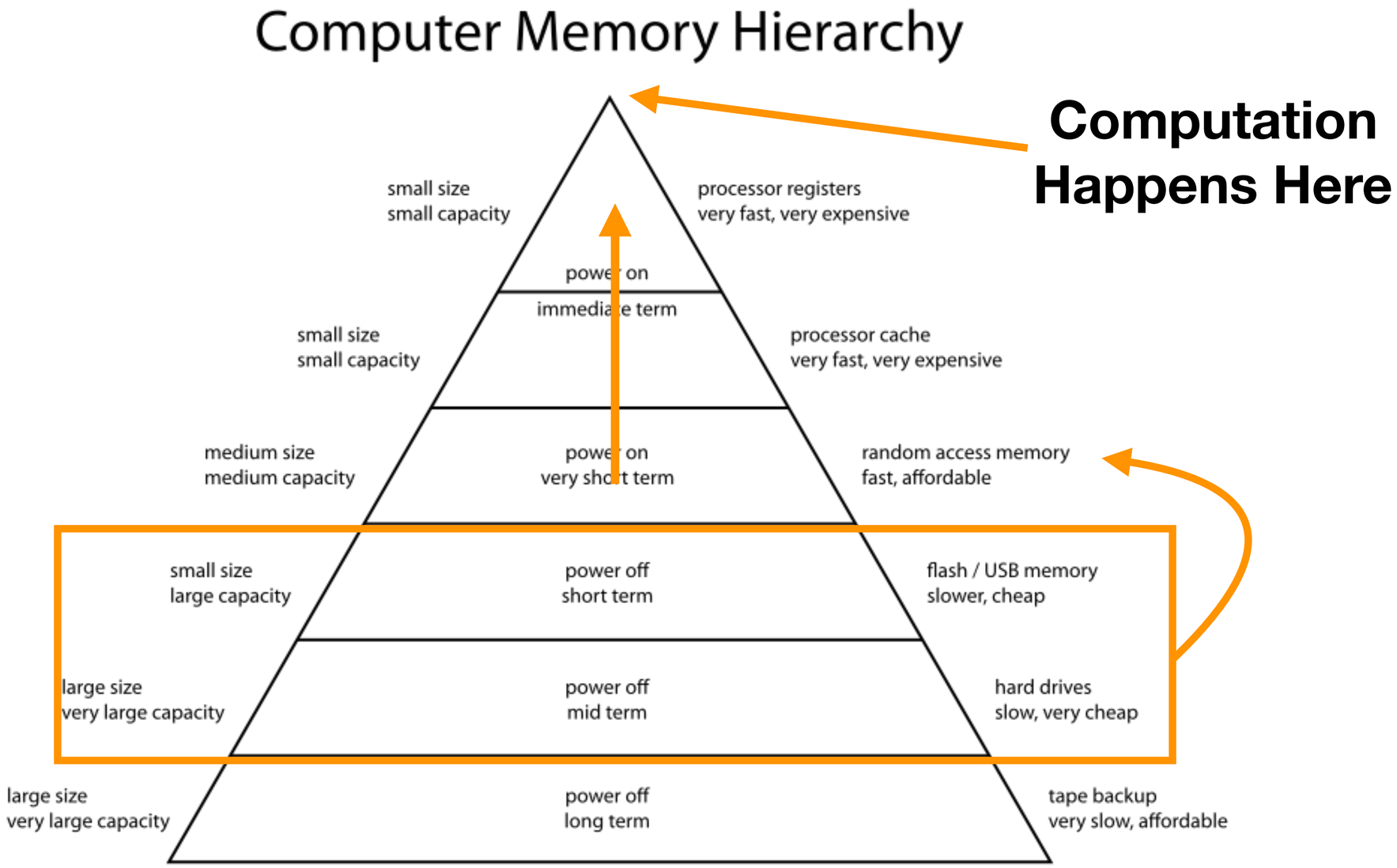
Comp. Sci.	Taylor	100000	
10101	Srinivasan	Comp. Sci.	65000
45565	Katz	Comp. Sci.	75000
83821	Brandt	Comp. Sci.	92000
Physics	Watson	70000	
33456	Gold	Physics	87000

Order of tuples in a file:
Heap File


4 7 2 6
Array of 3-bit values
(maximum value = 8)
4 2 1 4 7 3 6 5 1 2 0 1 1 0 5 6
Database File
...
6/8 = 75%
...
Has space? Insert!

Storing Metadata?
Use what you have! Relations!

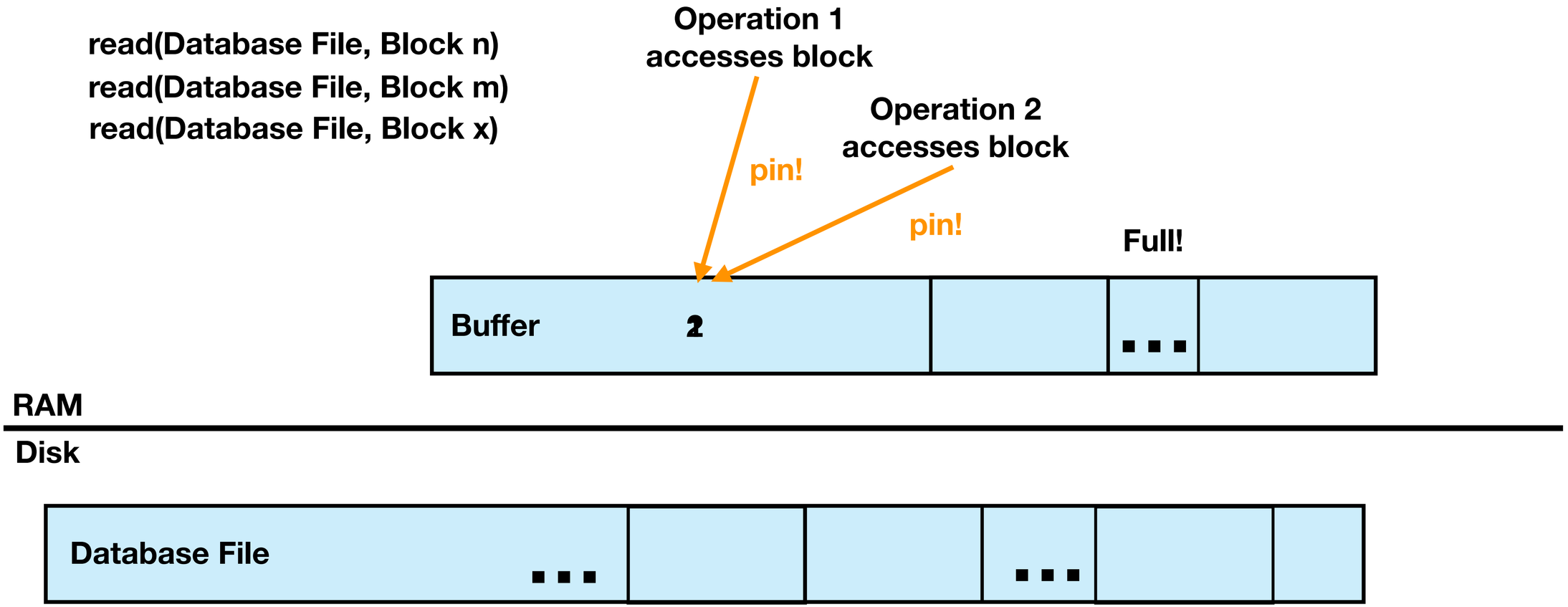

Relation_metadata
relation_name
number_of_attributes
storage_organization
location
Attribute_metadata
relation_name
attribute_name
domain_type
position
length
Index_metadata
index_name
relation_name
index_type
index_attributes
User_metadata
user_name
encrypted_password
group
View_metadata
view_name
definition



https://en.wikipedia.org/wiki/Memory_hierarchy#/media/File:ComputerMemoryHierarchy.svg

Buffer Management

read(Database File, Block n)
read(Database File, Block m)
read(Database File, Block x)



A Modern Alternative

Row Storage

10101	Srinavasan	Comp. Sci.	65000
1212	Wu	Finance	90000
15151	Mozart	Music	40000

Inserts and deletes for each tuple are fast
Scans on a single attribute are slow

Column Storage

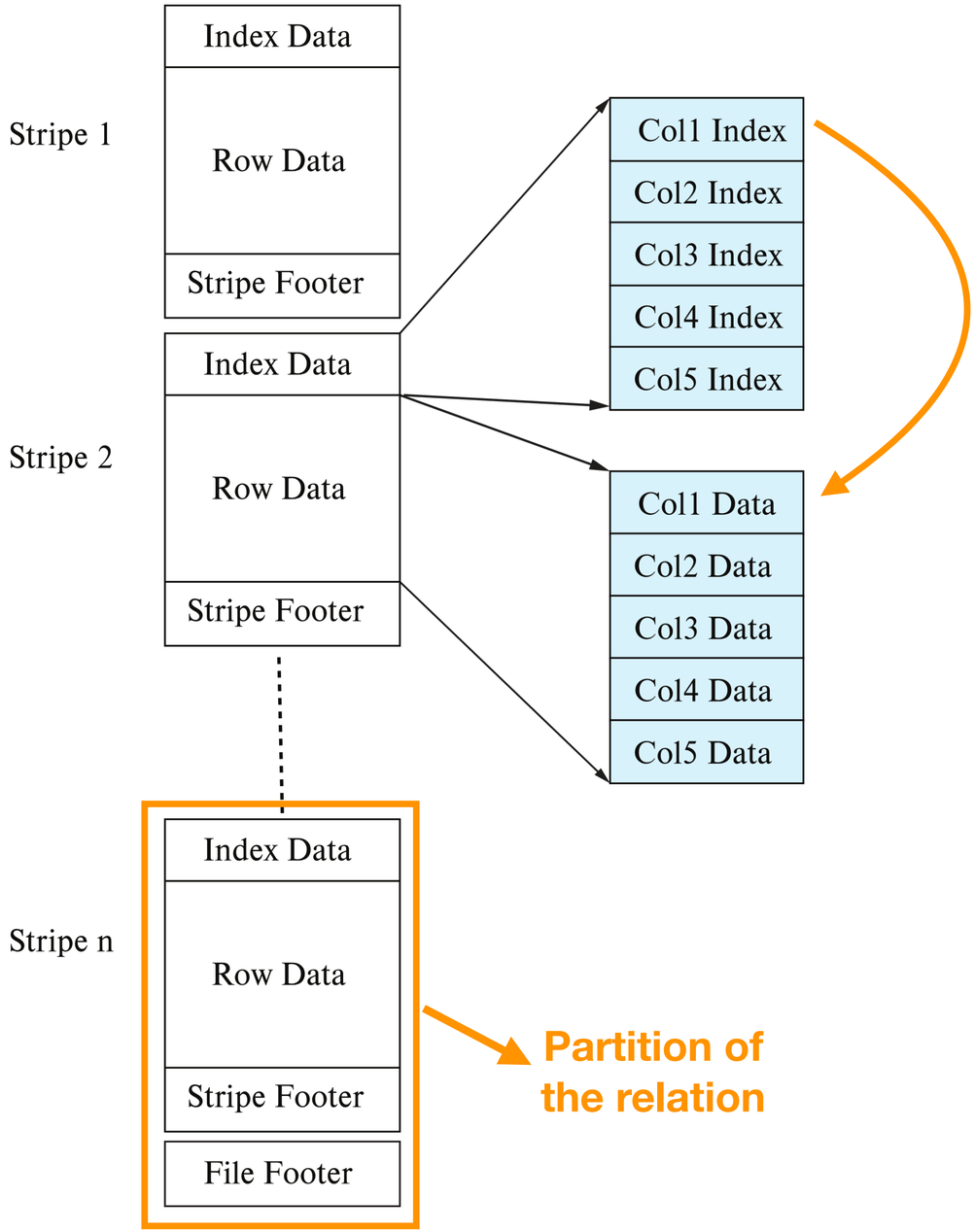
10101	Srinavasan	Comp. Sci.	65000
1212	Wu	Finance	90000
15151	Mozart	Music	40000

Scan on a few attributes are fast
Compression
CPU cache performance

Need to reconstruct the tuple
Decompression cost
Inserting a tuple is random access

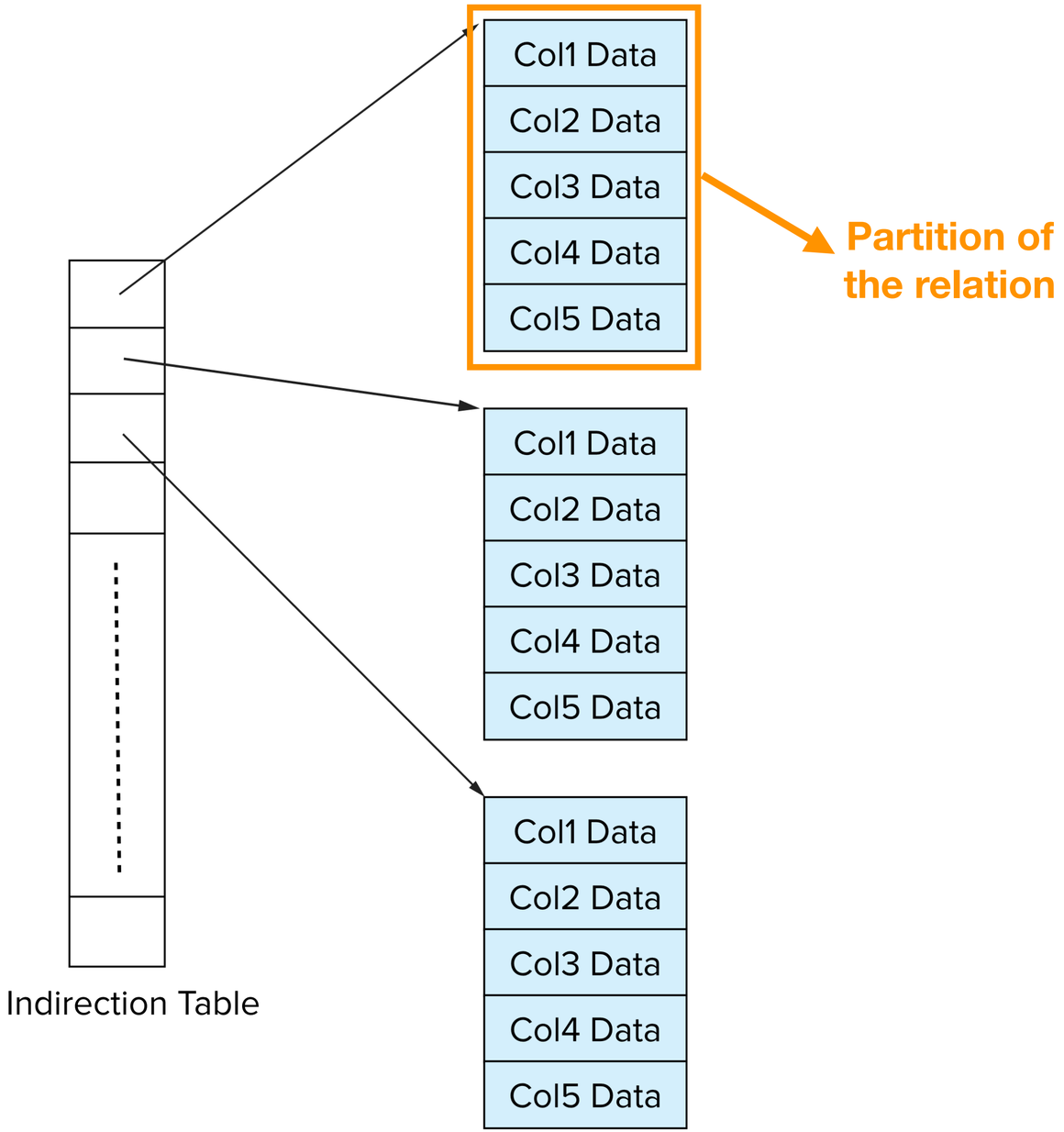
On disc:

- **ORC or Parquet Format**
- **Popular for BIG DATA and Distributed Settings**



In main memory:

- **No buffer manager**
- **Possible due to:**
  - **heavy compression**
  - **larger (TBs) main memory**



Questions?