Database Technology

Data Structures for Databases

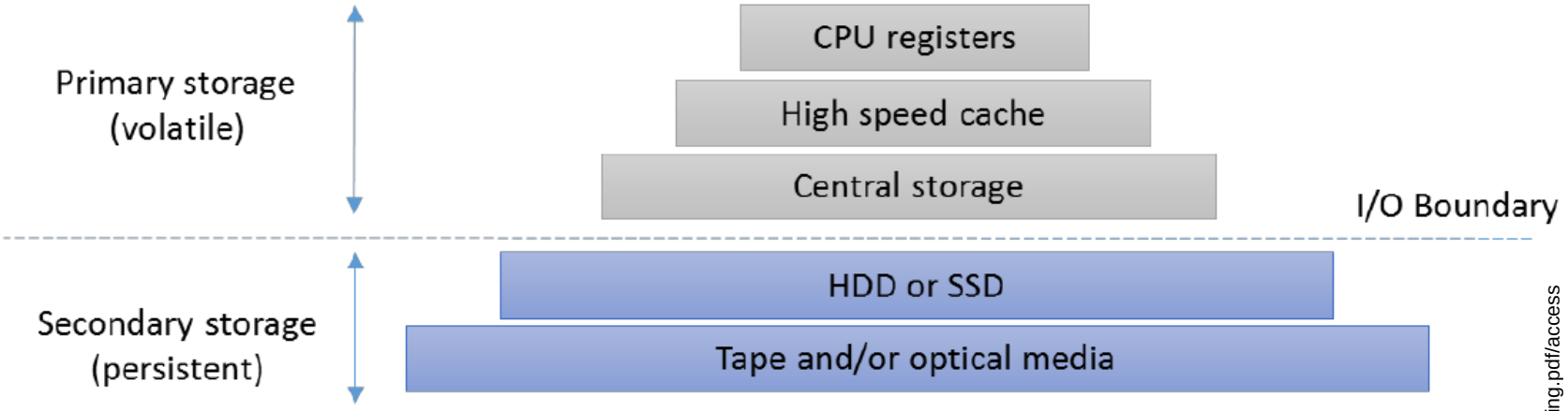
Olaf Hartig

olaf.hartig@liu.se

LINKÖPING UNIVERSITY

Storage Hierarchy

Quiz



Which of the following statements *is correct*?

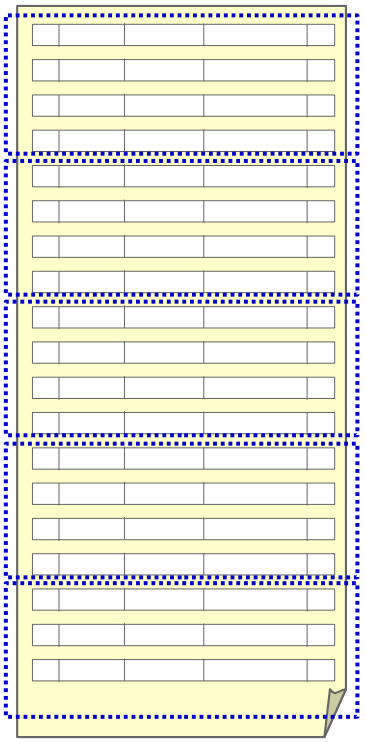
1) Secondary storage devices are usually faster than primary storage devices.
2) Data in a primary storage device may be lost when switching off the power.
3) The CPU may operate directly on data that is in a secondary storage device.
4) A piece of data (e.g., a record) may not be held both in a primary storage device and in a secondary storage device at the same time.

Image source: https://www.pdbmbook.com/vault/Chapter%20Physical%20File%20Organization%20and%20Indexing.pdf/access

Record Allocation

(Allocating Records to File Blocks)

Quiz

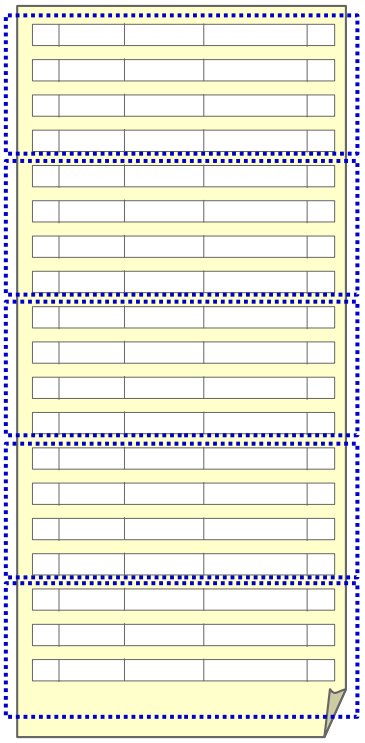


- Assume a file with
  - r = 200,000 records,
  - R = 400 bytes per record, and
  - B = 8,000 bytes per block

- How many blocks are needed to store the file?

1) b = 1,000 2) b = 2,000 3) b = 8,000 4) b = 10,000

Quiz



- Assume a file with
  - r = 200,000 records,
  - R = 400 bytes per record, and
  - B = 8,000 bytes per block

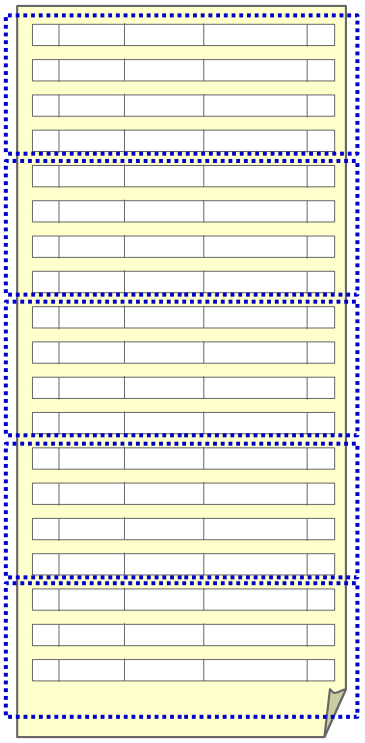
- How many blocks are needed to store the file?

1) ~~b = 1,000~~ 2) ~~b = 2,000~~ 3) ~~b = 8,000~~ 4) b = 10,000

$$bfr = \left\lfloor \frac{B}{R} \right\rfloor = \left\lfloor \frac{8{,}000}{400} \right\rfloor = 20 \qquad b = \left\lceil \frac{r}{bfr} \right\rceil = \left\lceil \frac{200{,}000}{20} \right\rceil = 10{,}000$$

blocking factor

Quiz



- Assume a file with
  - r = 200,000 records,
  - R = 400 bytes per record, and
  - B = 8,000 bytes per block

- How many blocks are needed to store the file?

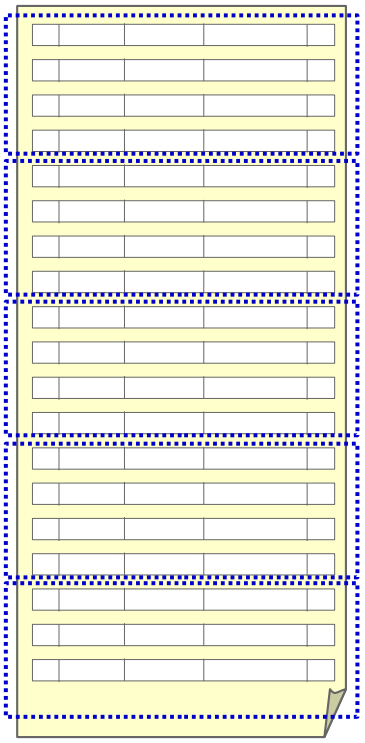
1) ~~b = 1,000~~ 2) ~~b = 2,000~~ 3) ~~b = 8,000~~ 4) b = 10,000

$$bfr = \left\lfloor \frac{B}{R} \right\rfloor = \left\lfloor \frac{8{,}000}{400} \right\rfloor = 20 \qquad b = \left\lceil \frac{r}{bfr} \right\rceil = \left\lceil \frac{200{,}000}{20} \right\rceil = 10{,}000$$

- How much space is wasted per block?

1) 0 bytes 2) 10 bytes 3) 20 bytes 4) 100 bytes

Quiz



- Assume a file with
  - r = 200,000 records,
  - R = 400 bytes per record, and
  - B = 8,000 bytes per block

- How many blocks are needed to store the file?

1) ~~b = 1,000~~ 2) ~~b = 2,000~~ 3) ~~b = 8,000~~ 4) b = 10,000

$$bfr = \left\lfloor \frac{B}{R} \right\rfloor = \left\lfloor \frac{8{,}000}{400} \right\rfloor = 20 \qquad b = \left\lceil \frac{r}{bfr} \right\rceil = \left\lceil \frac{200{,}000}{20} \right\rceil = 10{,}000$$

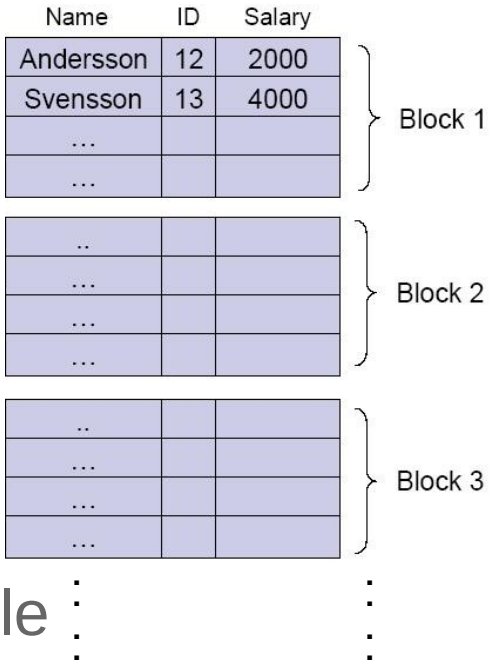
- How much space is wasted per block? $B - bfr * R$

1) 0 bytes 2) ~~10 bytes~~ 3) ~~20 bytes~~ 4) ~~100 bytes~~

File Organization

(Organizing Records in Files)

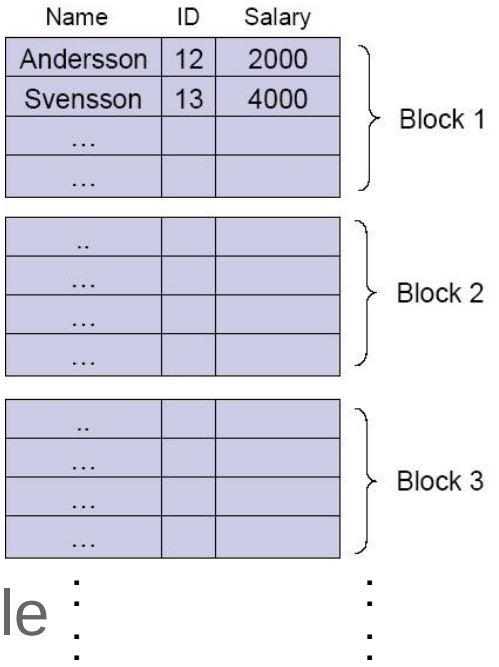
Exercise: Heap File



- Assume a file with
  - r = 200,000 records,
  - R = 400 bytes per record, and
  - B = 8,000 bytes per block
- Hence, b = 10,000 blocks needed to store the file
- Assume we organize the file as a heap file
  - i.e., new records are always appended to the end of the file
- How many blocks do we need to read?

	search field = ID value = 43 (unique)	search field = Name value = Smith (non-unique)
worst case	?	
best case	?	
average case	?	

Exercise: Heap File

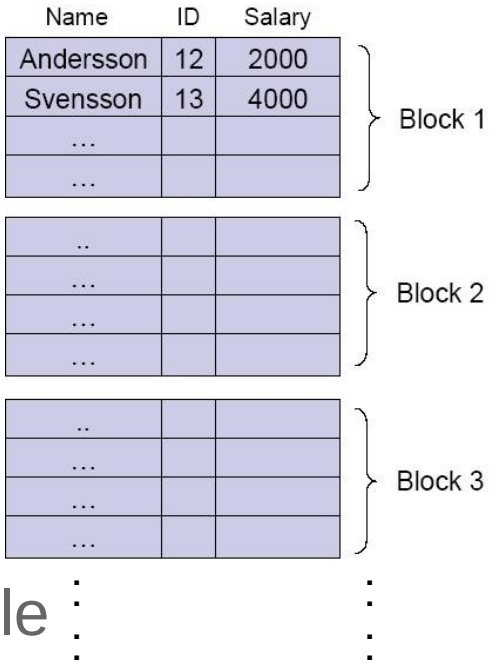


- Assume a file with
  - r = 200,000 records,
  - R = 400 bytes per record, and
  - B = 8,000 bytes per block
- Hence, b = 10,000 blocks needed to store the file
- Assume we organize the file as a heap file
  - i.e., new records are always appended to the end of the file
- How many blocks do we need to read?

	search field = ID value = 43 (unique)	search field = Name value = Smith (non-unique)
worst case	10,000	
best case	1	
average case	5,000	

$\left\lceil \frac{b}{2} \right\rceil$

Exercise: Heap File

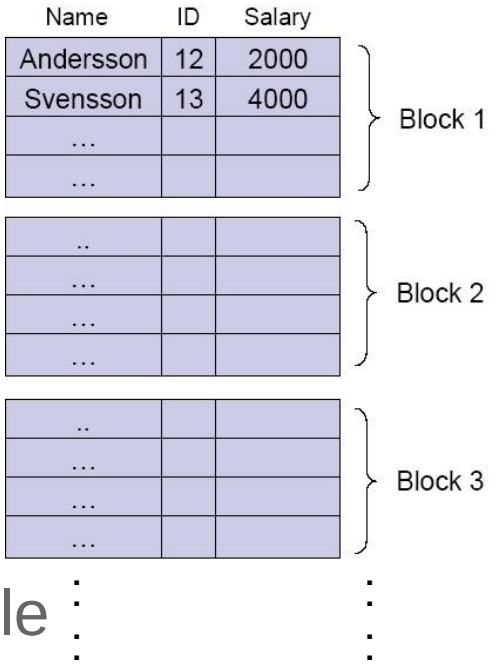


- Assume a file with
  - r = 200,000 records,
  - R = 400 bytes per record, and
  - B = 8,000 bytes per block
- Hence, b = 10,000 blocks needed to store the file
- Assume we organize the file as a heap file
  - i.e., new records are always appended to the end of the file
- How many blocks do we need to read?

	search field = ID value = 43 (unique)	search field = Name value = Smith (non-unique)
worst case	10,000	?
best case	1	?
average case	5,000	?

$\left\lceil \frac{b}{2} \right\rceil$

Exercise: Heap File



- Assume a file with
  - r = 200,000 records,
  - R = 400 bytes per record, and
  - B = 8,000 bytes per block
- Hence, b = 10,000 blocks needed to store the file
- Assume we organize the file as a heap file
  - i.e., new records are always appended to the end of the file
- How many blocks do we need to read?

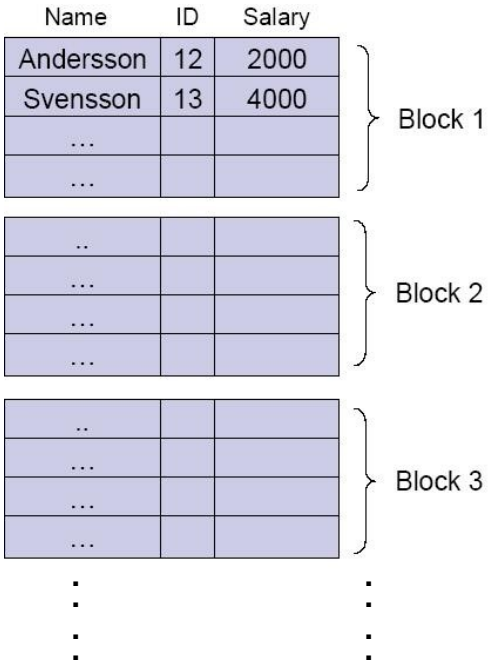
	search field = ID value = 43 (unique)	search field = Name value = Smith (non-unique)
worst case	10,000	10,000
best case	1	10,000
average case	5,000	10,000

linear search until last block

$\left\lceil \frac{b}{2} \right\rceil$

Exercise: Sorted File (a.k.a. Sequential File)

- Assume a file with
  - r = 200,000 records,
  - R = 400 bytes per record, and
  - B = 8,000 bytes per block
- Hence, b = 10,000 blocks needed for the file
- Assume we organize the file as a sorted file by using the ID field as the *sorting field*
  - i.e., records inserted based on their ID value



	search field = ID value = 43 (unique)	search field = Name value = Smith (non-unique)
worst case	?	?
best case	?	?
average case	?	?

$\log_2(2)=1$ $\log_2(256)=8$

$\log_2(4)=2$ $\log_2(512)=9$

$\log_2(8)=3$ $\log_2(1024)=10$

$\log_2(16)=4$ $\log_2(2048)=11$

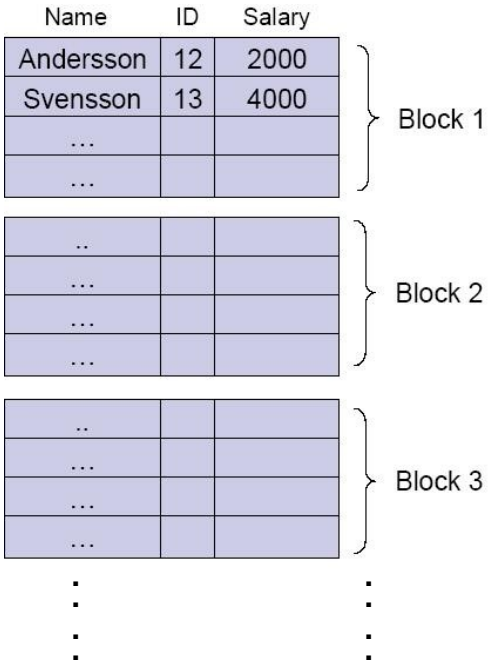
$\log_2(32)=5$ $\log_2(4096)=12$

$\log_2(64)=6$ $\log_2(8192)=13$

$\log_2(128)=7$ $\log_2(16384)=14$

Exercise: Sorted File (a.k.a. Sequential File)

- Assume a file with
  - r = 200,000 records,
  - R = 400 bytes per record, and
  - B = 8,000 bytes per block
- Hence, b = 10,000 blocks needed for the file
- Assume we organize the file as a sorted file by using the ID field as the *sorting field*
  - i.e., records inserted based on their ID value



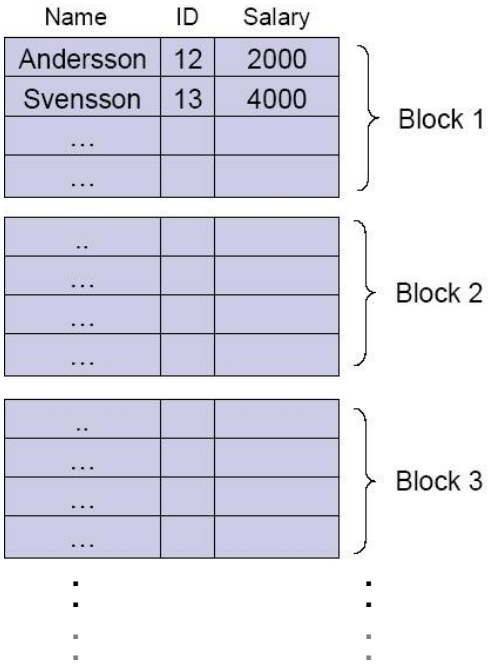
	search field = ID value = 43 (unique)	search field = Name value = Smith (non-unique)
worst case	14	10,000
best case	1	10,000
average case	ca. 14	10,000

$\lceil \log_2 b \rceil$

$\log_2(2)=1$ $\log_2(256)=8$
$\log_2(4)=2$ $\log_2(512)=9$
$\log_2(8)=3$ $\log_2(1024)=10$
$\log_2(16)=4$ $\log_2(2048)=11$
$\log_2(32)=5$ $\log_2(4096)=12$
$\log_2(64)=6$ $\log_2(8192)=13$
$\log_2(128)=7$ $\log_2(16384)=14$

Exercise: Hash File (a.k.a. Random File Orga.)

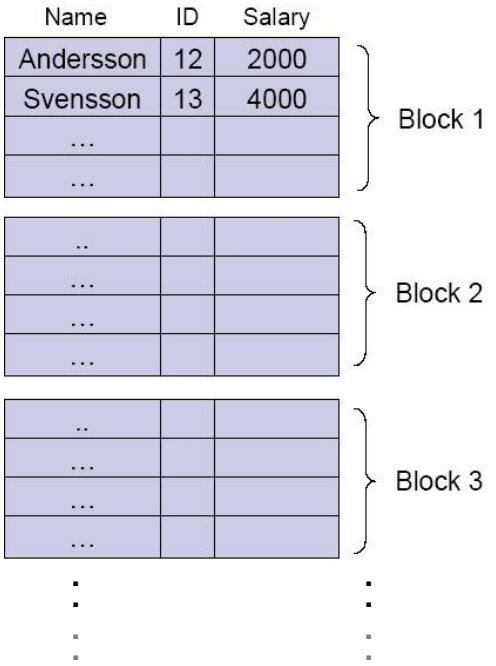
- Assume a file with
  - r = 200,000 records,
  - R = 400 bytes per record, and
  - B = 8,000 bytes per block
- Hence, b = 10,000 blocks needed for the file
- Assume we organize the file as a hash file by using the ID field as the *hash field*
  - i.e., find relevant bucket by applying hash function to the ID value; assume 5,000 buckets with 4 blocks per bucket



	search field = ID value = 43 (unique)	search field = Name value = Smith (non-unique)
worst case	?	?
best case	?	?
average case	?	?

Exercise: Hash File (a.k.a. Random File Orga.)

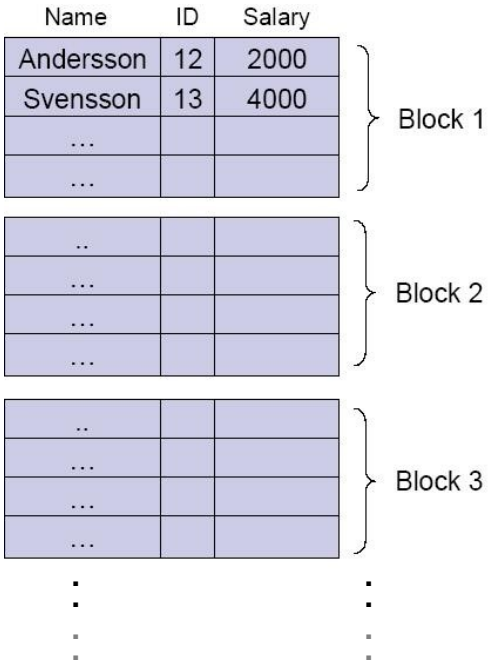
- Assume a file with
  - r = 200,000 records,
  - R = 400 bytes per record, and
  - B = 8,000 bytes per block
- Hence, b = 10,000 blocks needed for the file
- Assume we organize the file as a hash file by using the ID field as the *hash field*
  - i.e., find relevant bucket by applying hash function to the ID value; assume 5,000 buckets with 4 blocks per bucket



	search field = ID value = 43 (unique)	search field = Name value = Smith (non-unique)
worst case	4	≥ 10,000
best case	1	≥ 10,000
average case	depends	≥ 10,000

scan all non-empty blocks of all buckets

Exercise: Hash File (a.k.a. Random File Orga.)

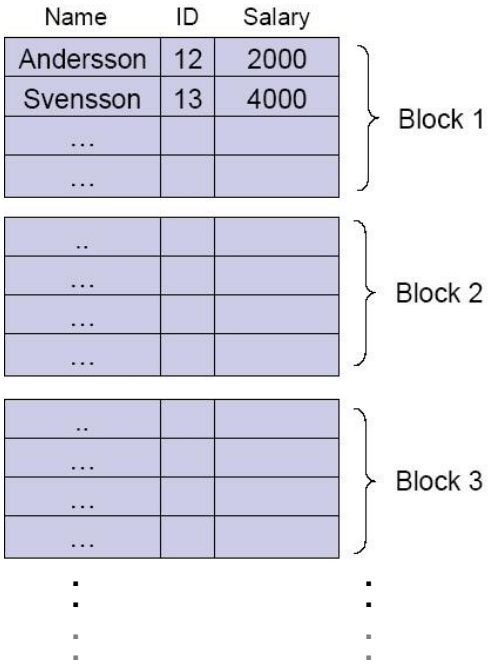


- Assume a file with
  - r = 200,000 records,
  - R = 400 bytes per record, and
  - B = 8,000 bytes per block
- Hence, b = 10,000 blocks needed for the file
- Assume we organize the file as a hash file by using the ID field as the *hash field*
  - i.e., find relevant bucket by applying hash function to the ID value; assume 5,000 buckets with 4 blocks per bucket
- What if we want to retrieve all records with an ID value smaller than 10? (assuming IDs cannot be smaller than 1)

worst case	?
best case	?

Exercise: Hash File (a.k.a. Random File Orga.)

- Assume a file with
  - r = 200,000 records,
  - R = 400 bytes per record, and
  - B = 8,000 bytes per block
- Hence, b = 10,000 blocks needed for the file
- Assume we organize the file as a hash file by using the ID field as the *hash field*
  - i.e., find relevant bucket by applying hash function to the ID value; assume 5,000 buckets with 4 blocks per bucket
- What if we want to retrieve all records with an ID value smaller than 10? (assuming IDs cannot be smaller than 1)

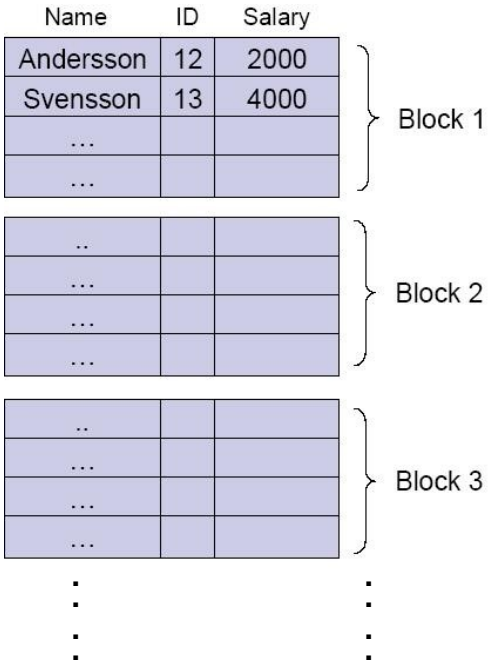


worst case	$9 \cdot 4 = 36$
best case	1 ← unlikely

Single-Level Ordered Indexes

Quiz: Types of Single-Level Ordered Indexes

- Back to the case of a sorted file, sorted on ID
- If we try to speed up finding records with a particular ID value by adding a single-level ordered index, which type would we need?
  A) Primary index
  B) Clustering index
  C) Secondary index on a key field
  D) Secondary index on a non-key field



	search field = ID value = 43 (unique)	search field = Name value = Smith (non-unique)
worst case	14	10,000
best case	1	10,000
average case	ca. 14	10,000

Summary of Single-Level Ordered Indexes

	Index field used for sorting the data records	**Index field *not* used for sorting the data records**
Index field is a key	Primary index	Secondary index (key)
Index field is not a key	Clustering index	Secondary index (non-key)

Quiz

	Index field used for sorting the data records	**Index field *not* used for sorting the data records**
Index field is a key	Primary index	Secondary index (key)
Index field is not a key	Clustering index	Secondary index (non-key)

Which of these four types of indexes has the *smallest number of index records*?

A) Primary index

B) Clustering index

C) Secondary index on a key field

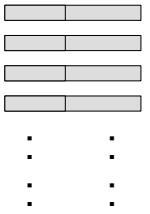
D) Secondary index on a non-key field

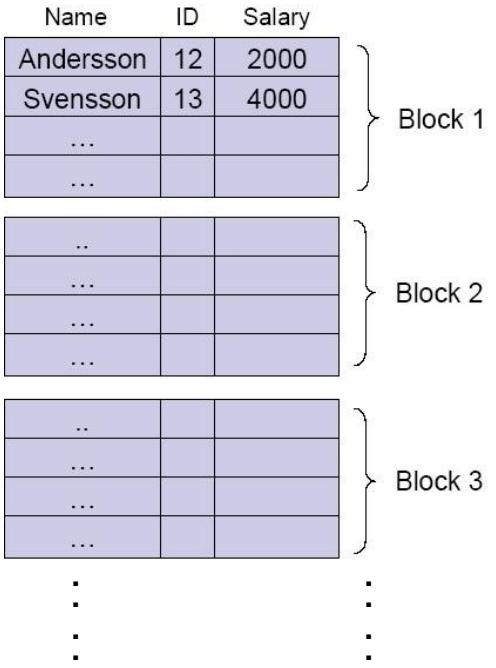
Summary of Single-Level Indexes (cont'd)

	Index field used for sorting the data records	**Index field *not* used for sorting the data records**
Index field is a key	Primary index	Secondary index (key)
Index field is not a key	Clustering index	Secondary index (non-key)

Type of index	**Number of index entries**
Primary	Number of blocks in data file
Clustering	Number of distinct index field values
Secondary (key)	Number of records in data file
Secondary (non-key)	Number of records or number of distinct index field values

Quiz: Primary Index

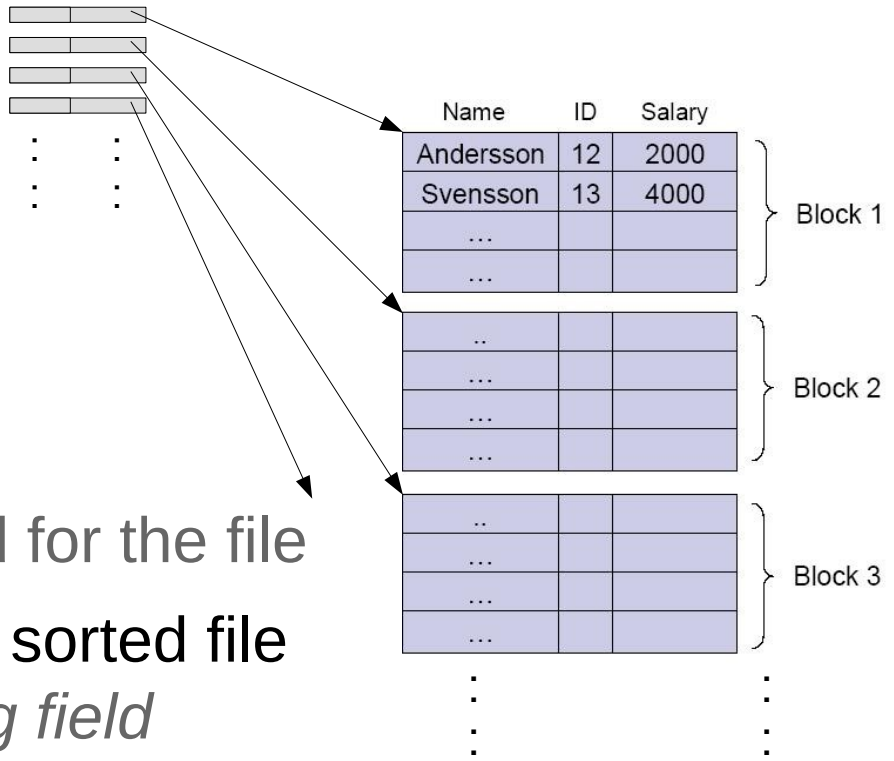




- Assume a file with
  - r = 200,000 records,
  - R = 400 bytes per record, and
  - B = 8,000 bytes per block
- Hence, b = 10,000 blocks needed for the file
- Assume we organize the file as a sorted file by using the ID field as the *sorting field*
- Assume we create a primary index on the ID field
  - same block size for the index file: $B_{idx} = B$ = 8,000 bytes
  - but smaller records: R_{idx} = 100 bytes per index record
- How many index records does this index contain?

  A) 8,000 B) 10,000 C) 20,000 D) 200,000

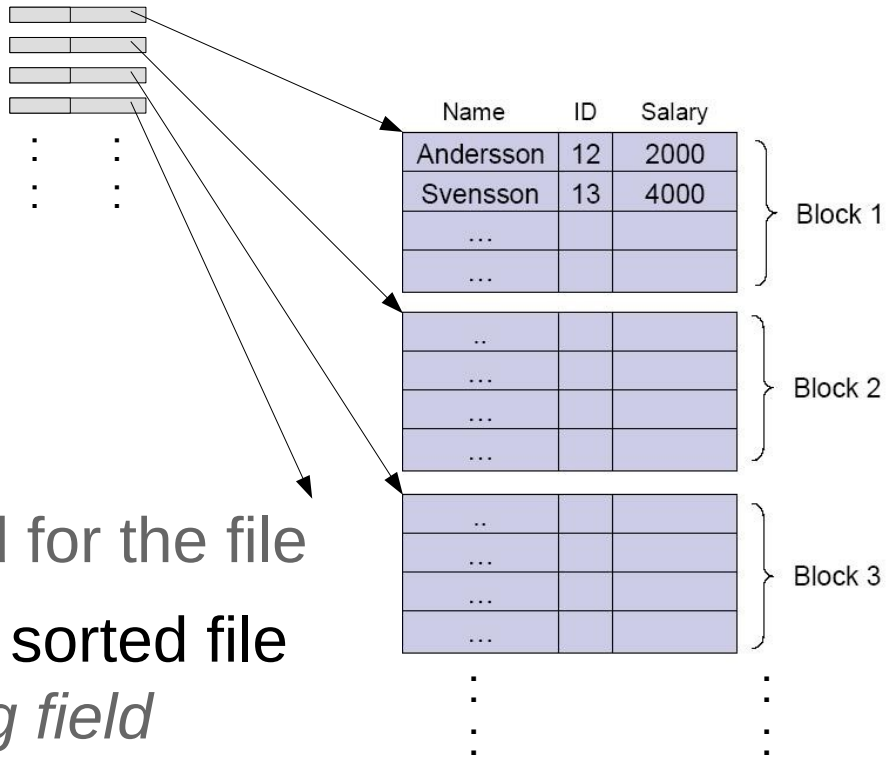
Quiz: Primary Index



- Assume a file with
  - r = 200,000 records,
  - R = 400 bytes per record, and
  - B = 8,000 bytes per block
- Hence, b = 10,000 blocks needed for the file
- Assume we organize the file as a sorted file by using the ID field as the *sorting field*
- Assume we create a primary index on the ID field
  - same block size for the index file: $B_{idx} = B$ = 8,000 bytes
  - but smaller records: R_{idx} = 100 bytes per index record
- How many index records does this index contain?

A) ~~8,000~~ B) 10,000 C) ~~20,000~~ D) ~~200,000~~

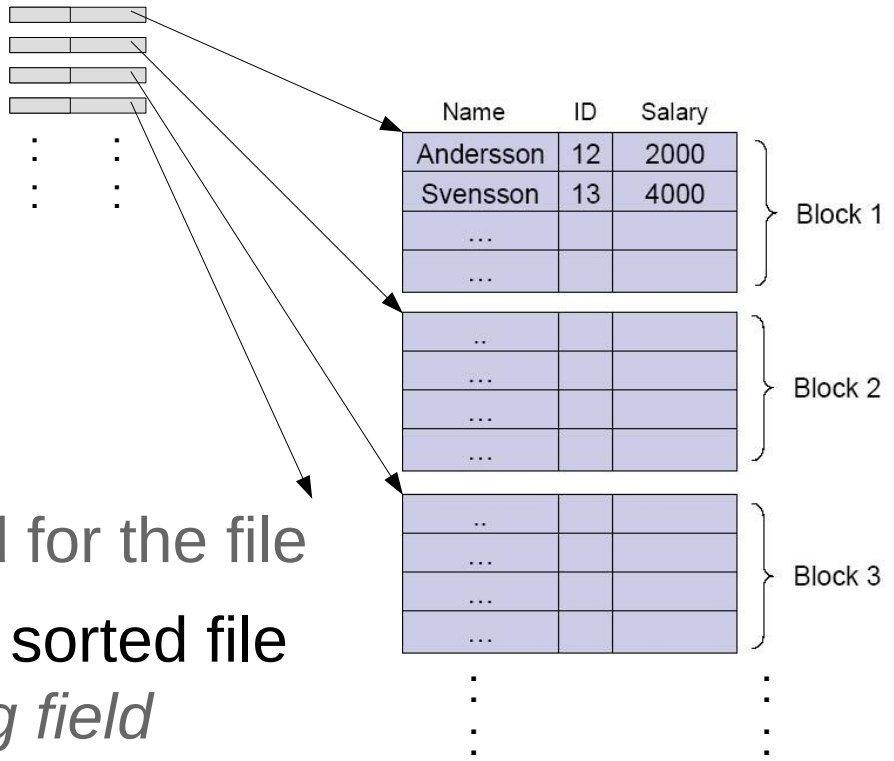
Quiz: Primary Index



- Assume a file with
  - r = 200,000 records,
  - R = 400 bytes per record, and
  - B = 8,000 bytes per block
- Hence, b = 10,000 blocks needed for the file
- Assume we organize the file as a sorted file by using the ID field as the *sorting field*
- Assume we create a primary index on the ID field
  - same block size for the index file: $B_{idx} = B$ = 8,000 bytes
  - but smaller records: R_{idx} = 100 bytes per index record
- How many blocks does the index file consist of?

  A) 125 B) 250 C) 1,000 D) 10,000

Quiz: Primary Index

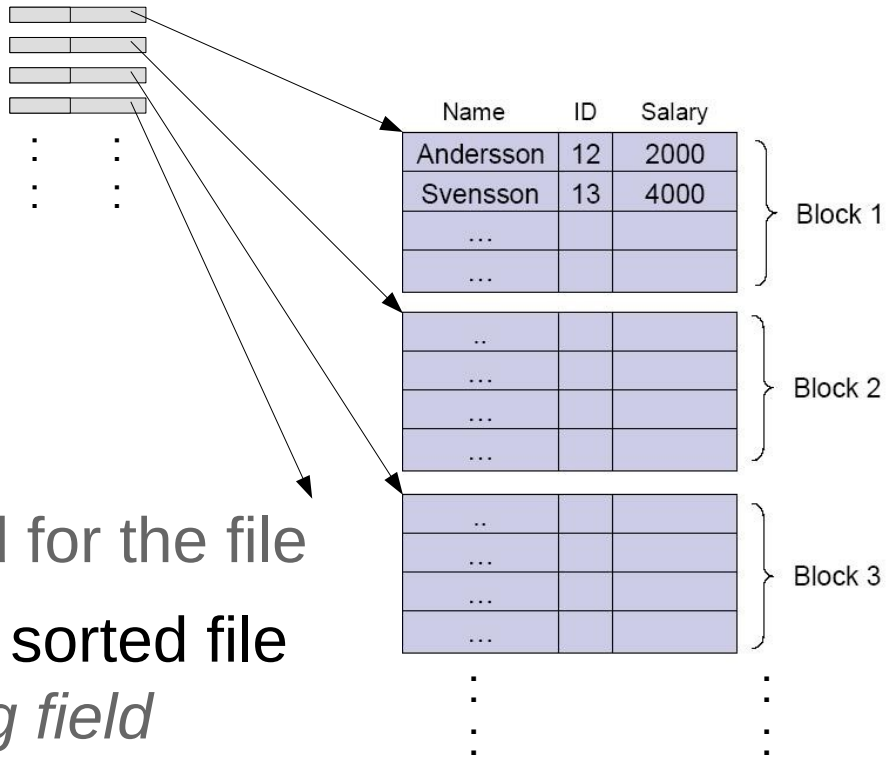


- Assume a file with
  - r = 200,000 records,
  - R = 400 bytes per record, and
  - B = 8,000 bytes per block
- Hence, b = 10,000 blocks needed for the file
- Assume we organize the file as a sorted file by using the ID field as the *sorting field*
- Assume we create a primary index on the ID field
  - same block size for the index file: $B_{idx} = B$ = 8,000 bytes
  - but smaller records: R_{idx} = 100 bytes per index record
- How many blocks does the index file consist of?

  A) 125 B) ~~250~~ C) ~~1,000~~ D) ~~10,000~~

$$b_{idx} = \left\lceil \frac{r_{idx}}{bfr_{idx}} \right\rceil = \left\lceil \frac{10{,}000}{80} \right\rceil = 125 \qquad bfr_{idx} = \left\lfloor \frac{B_{idx}}{R_{idx}} \right\rfloor = \left\lfloor \frac{8{,}000}{100} \right\rfloor = 80$$

Quiz: Primary Index



- Assume a file with
  - r = 200,000 records,
  - R = 400 bytes per record, and
  - B = 8,000 bytes per block
- Hence, b = 10,000 blocks needed for the file
- Assume we organize the file as a sorted file by using the ID field as the *sorting field*
- Assume we create a primary index on the ID field
  - consisting of 125 blocks
- How many blocks do we need to read if we want to retrieve the record with ID = 43?

  A) 6 B) 7 C) 8 D) 9

$\log_2(2)=1$ $\log_2(128)=7$

$\log_2(4)=2$ $\log_2(256)=8$

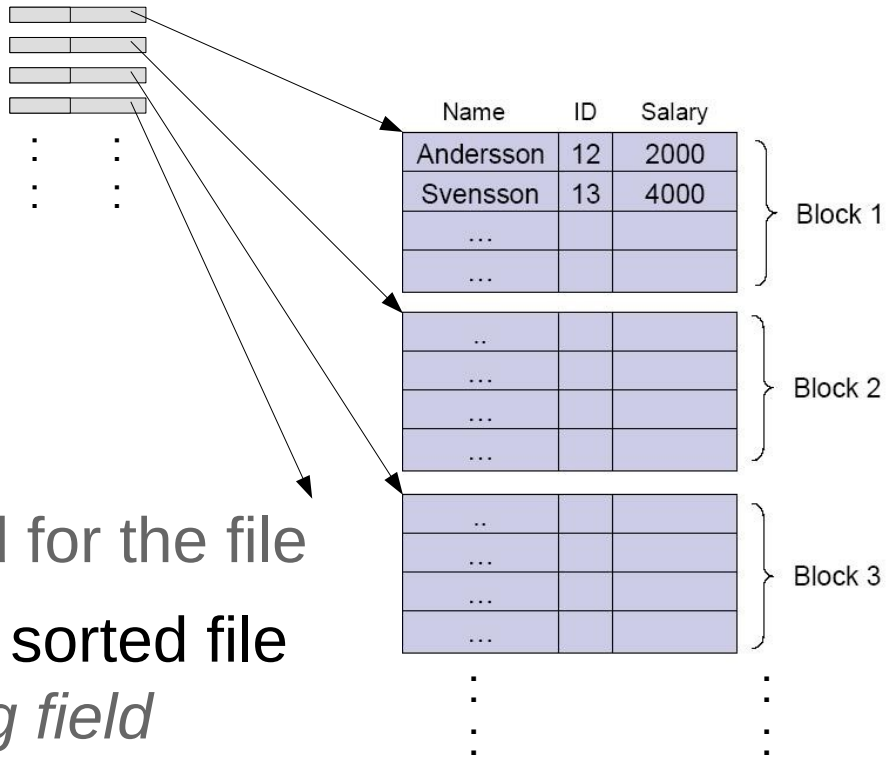
$\log_2(8)=3$ $\log_2(512)=9$

$\log_2(16)=4$ $\log_2(1024)=10$

$\log_2(32)=5$ $\log_2(2048)=11$

$\log_2(64)=6$ $\log_2(4096)=12$

Quiz: Primary Index



- Assume a file with
  - r = 200,000 records,
  - R = 400 bytes per record, and
  - B = 8,000 bytes per block
- Hence, b = 10,000 blocks needed for the file
- Assume we organize the file as a sorted file by using the ID field as the *sorting field*
- Assume we create a primary index on the ID field
  - consisting of 125 blocks
- How many blocks do we need to read if we want to retrieve the record with ID = 43?

A) ~~6~~ B) ~~7~~ C) 8 D) ~~9~~

$$\lceil \log_2 b_{idx} \rceil + 1$$

binary search in the index

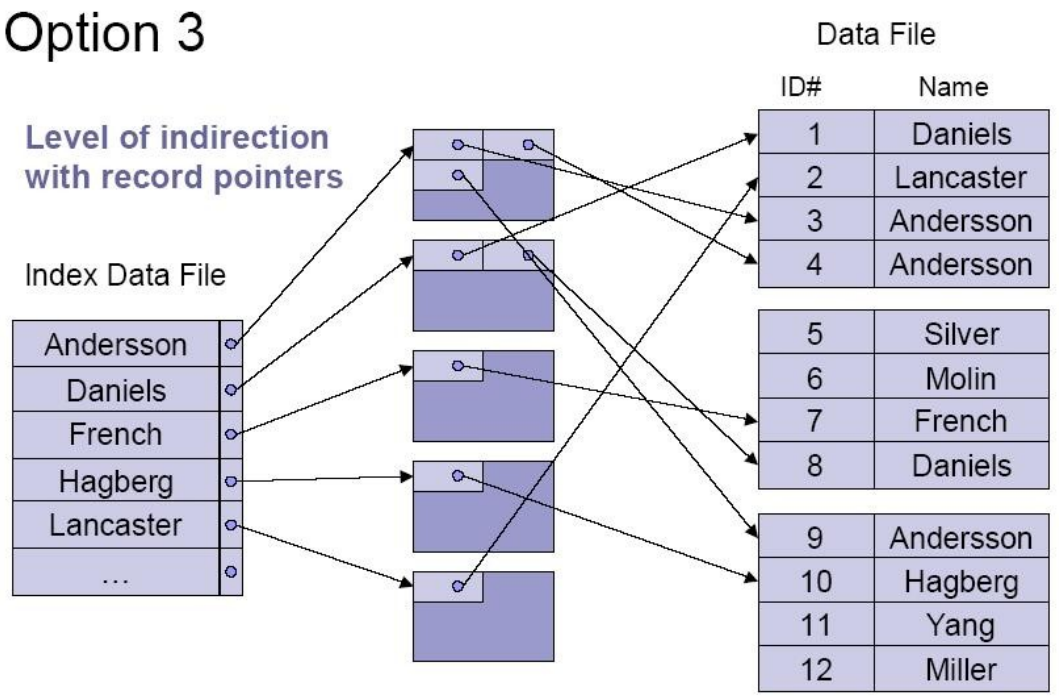
read data file block that contains the record

$\log_2(2)=1$ $\log_2(128)=7$
$\log_2(4)=2$ $\log_2(256)=8$
$\log_2(8)=3$ $\log_2(512)=9$
$\log_2(16)=4$ $\log_2(1024)=10$
$\log_2(32)=5$ $\log_2(2048)=11$
$\log_2(64)=6$ $\log_2(4096)=12$

Quiz: Secondary Index (Non-Key)

- Assume a file 200,000 records in 10,000 blocks
  - the file is not sorted on Name, and Name is not unique
- To speed up finding records with a given Name value, we create a secondary index on the Name field
- Assume the index blocks have a size of B_{idx} = 8,000 bytes, the index records have a size of R_{idx} = 200 bytes, and there 20,000 names
- How many blocks does the index file consist of (incl. indirection blocks)?

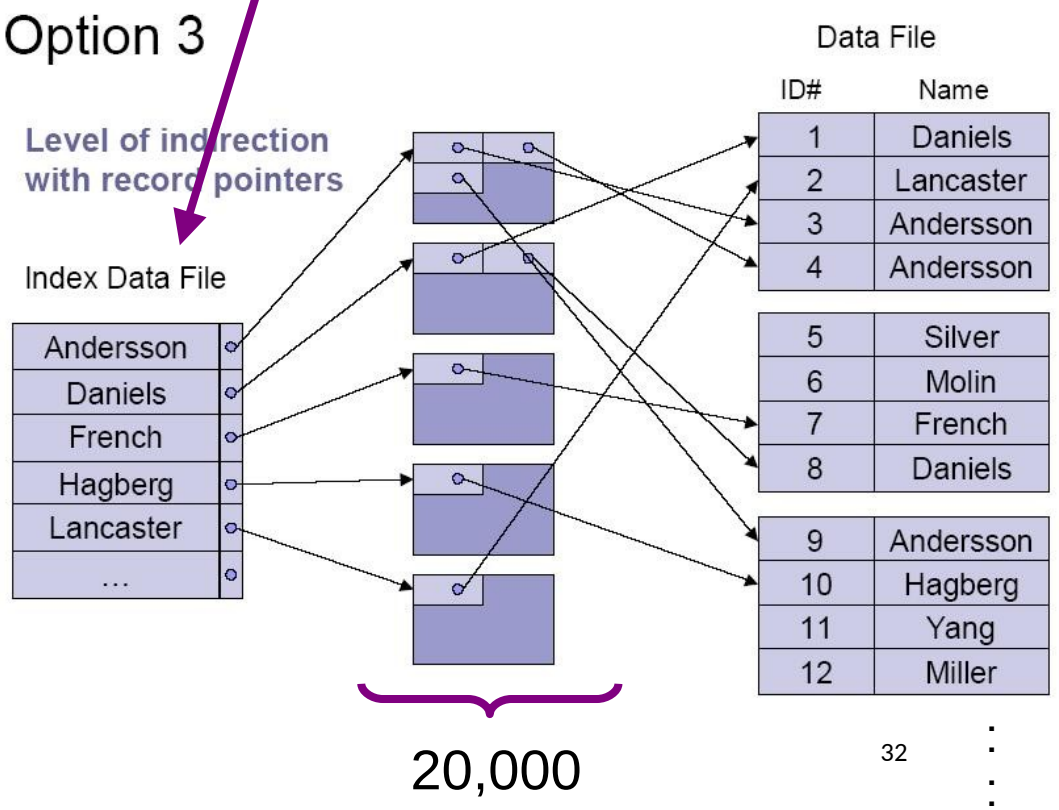
A) 20,250
B) 20,500
C) 22,000
D) 40,000



Quiz: Secondary Index (Non-Key)

- Assume a file 200,000 records in 10,000 blocks
  - the file is not sorted
- To speed up finding re… we create a secondary index on the Name field
- Assume the index blocks have a size of B_{idx} = 8,000 bytes, the index records have a size of R_{idx} = 200 bytes, and there 20,000 names
- How many blocks does the index file consist of (incl. indirection blocks)?

$$b_{idx} = \left\lceil \frac{r_{idx}}{bfr_{idx}} \right\rceil = \left\lceil \frac{20{,}000}{40} \right\rceil = 500 \qquad bfr_{idx} = \left\lfloor \frac{B_{idx}}{R_{idx}} \right\rfloor = \left\lfloor \frac{8{,}000}{200} \right\rfloor = 40$$



A) ~~20,250~~

B) **20,500**

C) ~~22,000~~

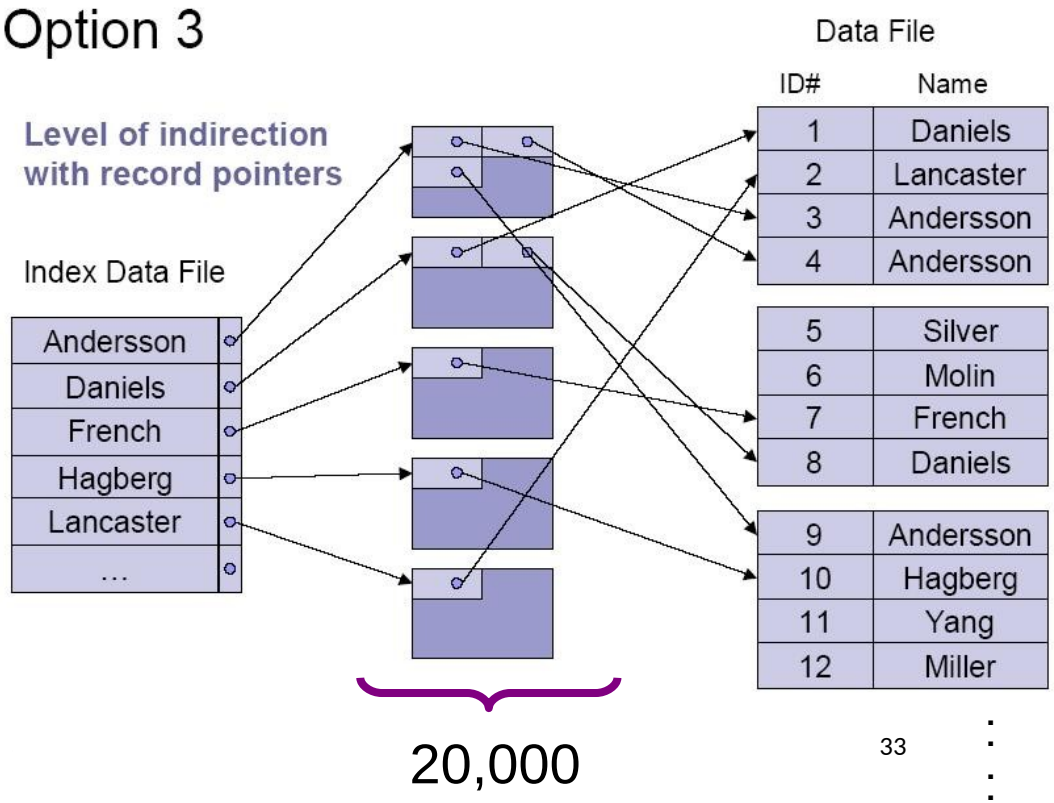
D) ~~40,000~~

Quiz: Secondary Index (Non-Key)

$\log_2(2)=1$ $\log_2(128)=7$
$\log_2(4)=2$ $\log_2(256)=8$
$\log_2(8)=3$ $\log_2(512)=9$
$\log_2(16)=4$ $\log_2(1024)=10$
$\log_2(32)=5$ $\log_2(2048)=11$
$\log_2(64)=6$ $\log_2(4096)=12$

- Assume a file 200,000 records in 10,000 blo
  - the file is not sorted on Name, and Name is r
- To speed up finding records with a given Na
  we create a secondary index on the Name field
- Index file consists of b_{idx} = 500 + 20,000 = 20,500 blocks
- How many blocks do we need to read, *in the best case*, to obtain all records with a particular Name value (e.g., Smith)?

  A) 1
  B) 9
  C) 10
  D) 11



Quiz: Secondary Index (Non-Key)

$\log_2(2)=1$ $\log_2(128)=7$
$\log_2(4)=2$ $\log_2(256)=8$
$\log_2(8)=3$ $\log_2(512)=9$
$\log_2(16)=4$ $\log_2(1024)=10$
$\log_2(32)=5$ $\log_2(2048)=11$
$\log_2(64)=6$ $\log_2(4096)=12$

- Assume a file 200,000 records in 10,000 blo
  - the file is not sorted on Name, and Name is r
- To speed up finding records with a given Na
  we create a secondary index on the Name field
- Index file consists of b_{idx} = 500 + 20,000 = 20,500 blocks
- How many blocks do we need to read, *in the best case*, to obtain all records with a particular Name value (e.g., Smith)?

A) 1

B) ~~9~~

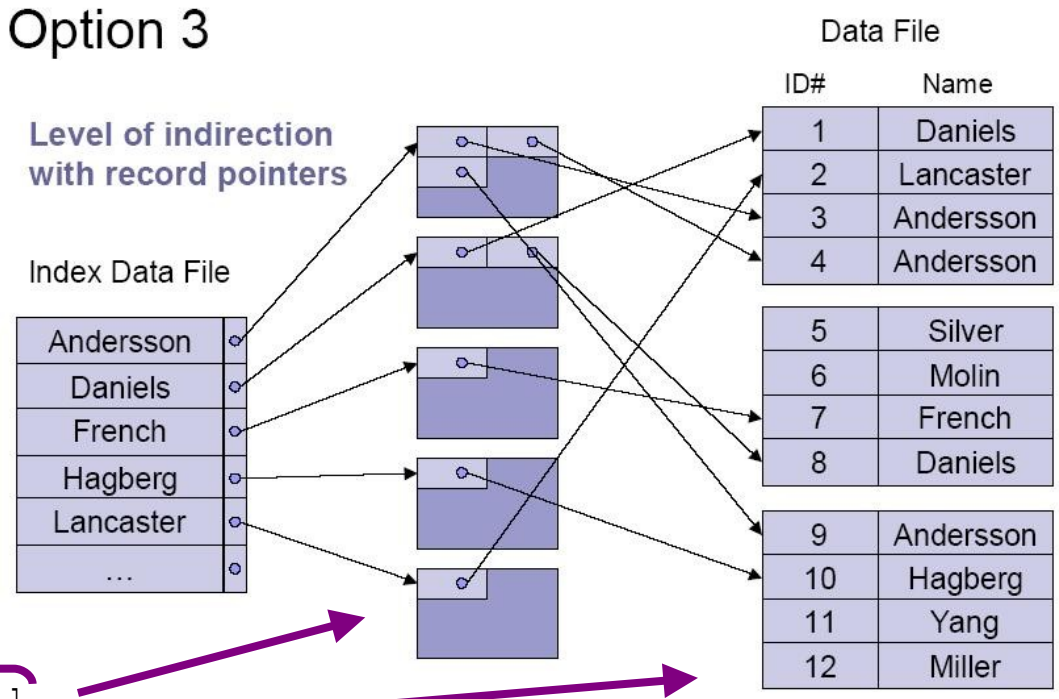
C) ~~10~~

D) **11**

no Smith, but if there was, its index record would be in the first index block

binary search

$\lceil \log_2 b_{idx} \rceil + 1 + 1$

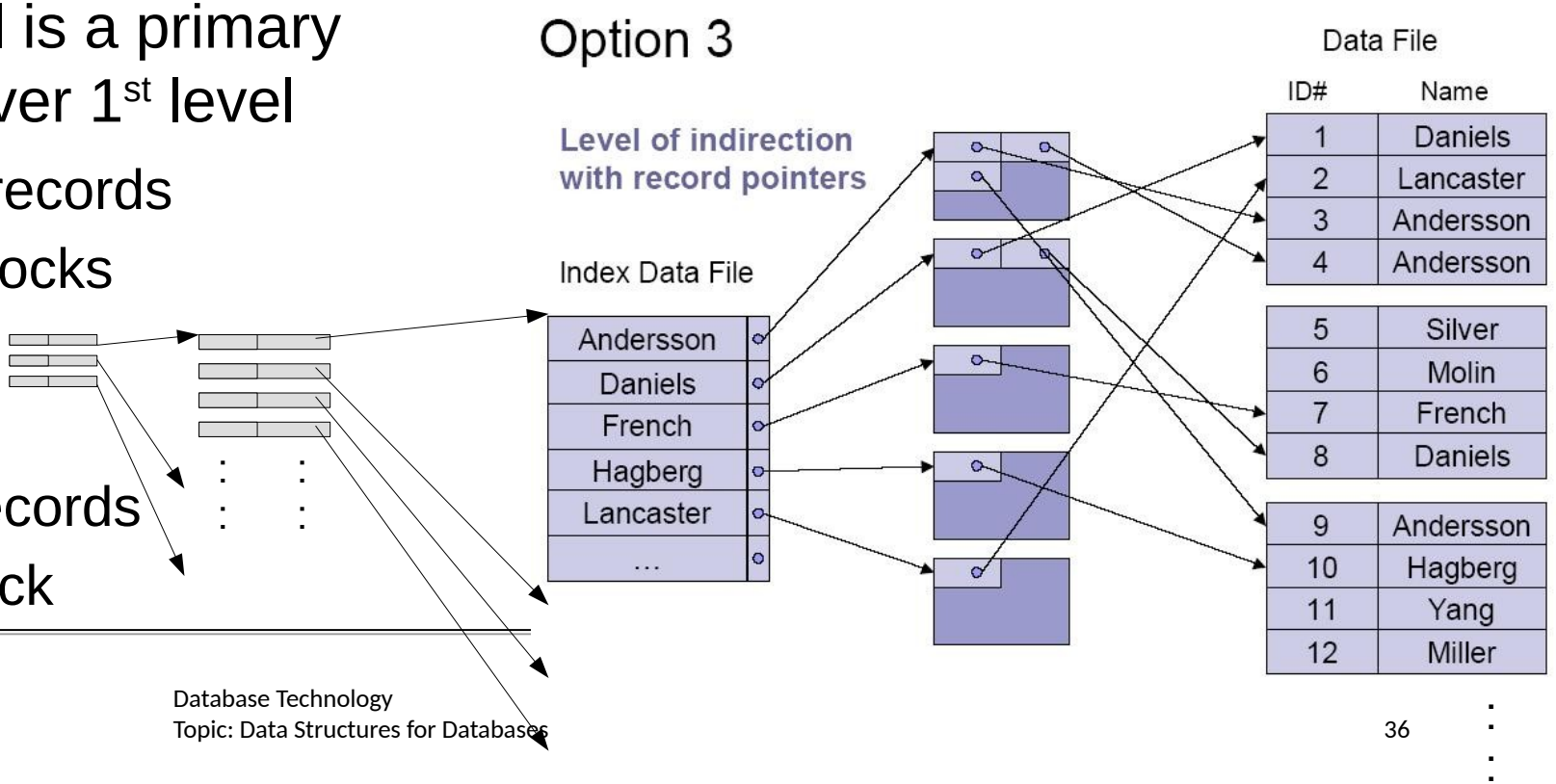


Multilevel Indexes

(Stacking indexes on top of one another)

Multilevel Index

- To speed up finding records with a given Name value, we create a *multilevel* secondary index on the Name field
  - still assuming B_{idx}= 8,000 bytes, R_{idx}= 200 bytes, and bfr_{idx}= 40
- First level is the secondary index, with 20,000 index records in 500 blocks (plus 20,000 indirection blocks)
- 2nd level is a primary index over 1st level
  - 500 records
  - 13 blocks
- 3rdlevel
  - 13 records
  - 1 block



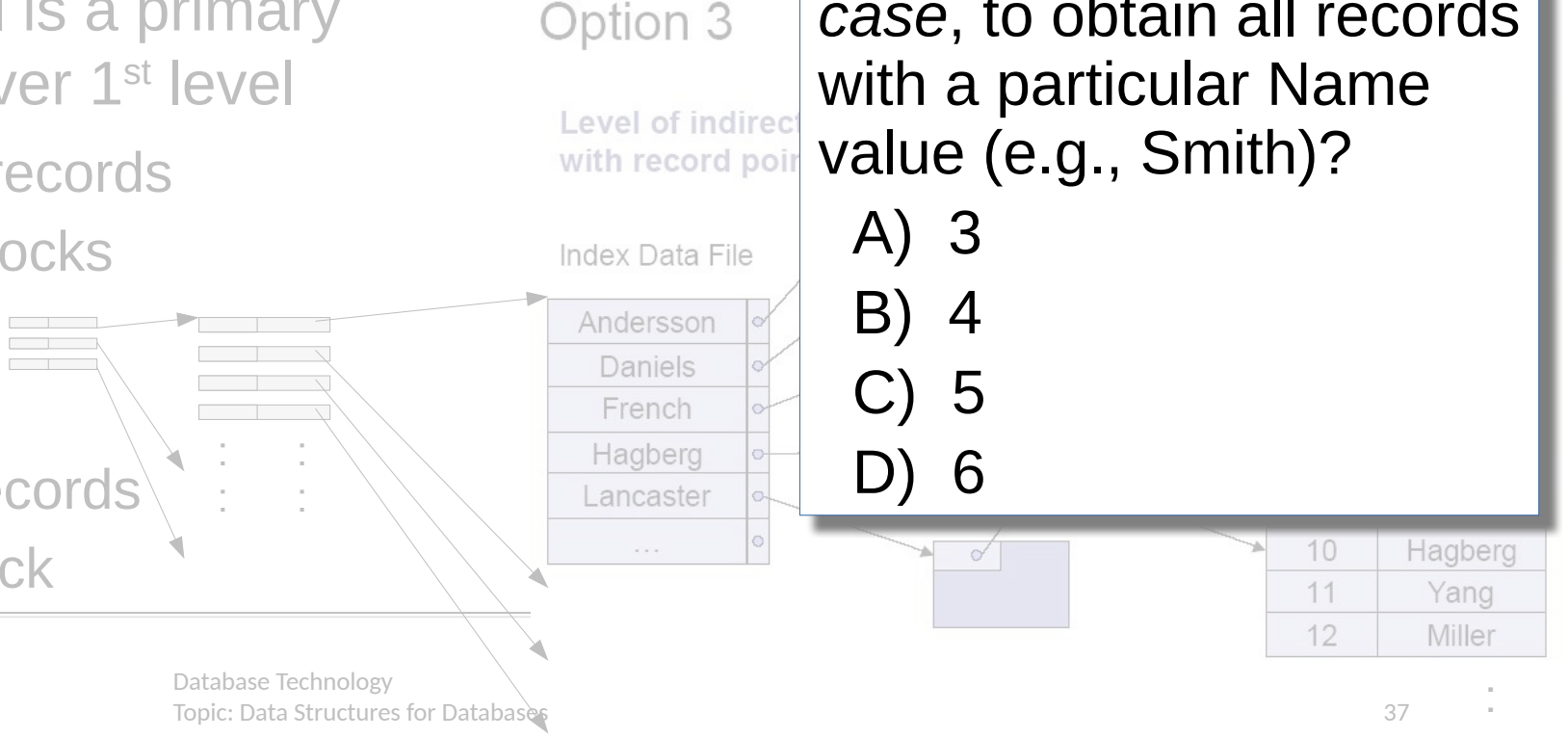
Multilevel Index

- To speed up finding records with a given Nam
  create a *multilevel* secondary index on the N
  - still assuming B_{idx} = 8,000 bytes, R_{idx} = 200 byte
- First level is a secondary index, with
  in 500 blocks (plus 20,000 indirection
- 2nd level is a primary index over 1st level
  - 500 records
  - 13 blocks
- 3rd level
  - 13 records
  - 1 block

$\log_2(2)=1$ $\log_2(128)=7$
$\log_2(4)=2$ $\log_2(256)=8$
$\log_2(8)=3$ $\log_2(512)=9$
$\log_2(16)=4$ $\log_2(1024)=10$
$\log_2(32)=5$ $\log_2(2048)=11$
$\log_2(64)=6$ $\log_2(4096)=12$

How many blocks do we need to read, *in the best case*, to obtain all records with a particular Name value (e.g., Smith)?

A) 3
B) 4
C) 5
D) 6



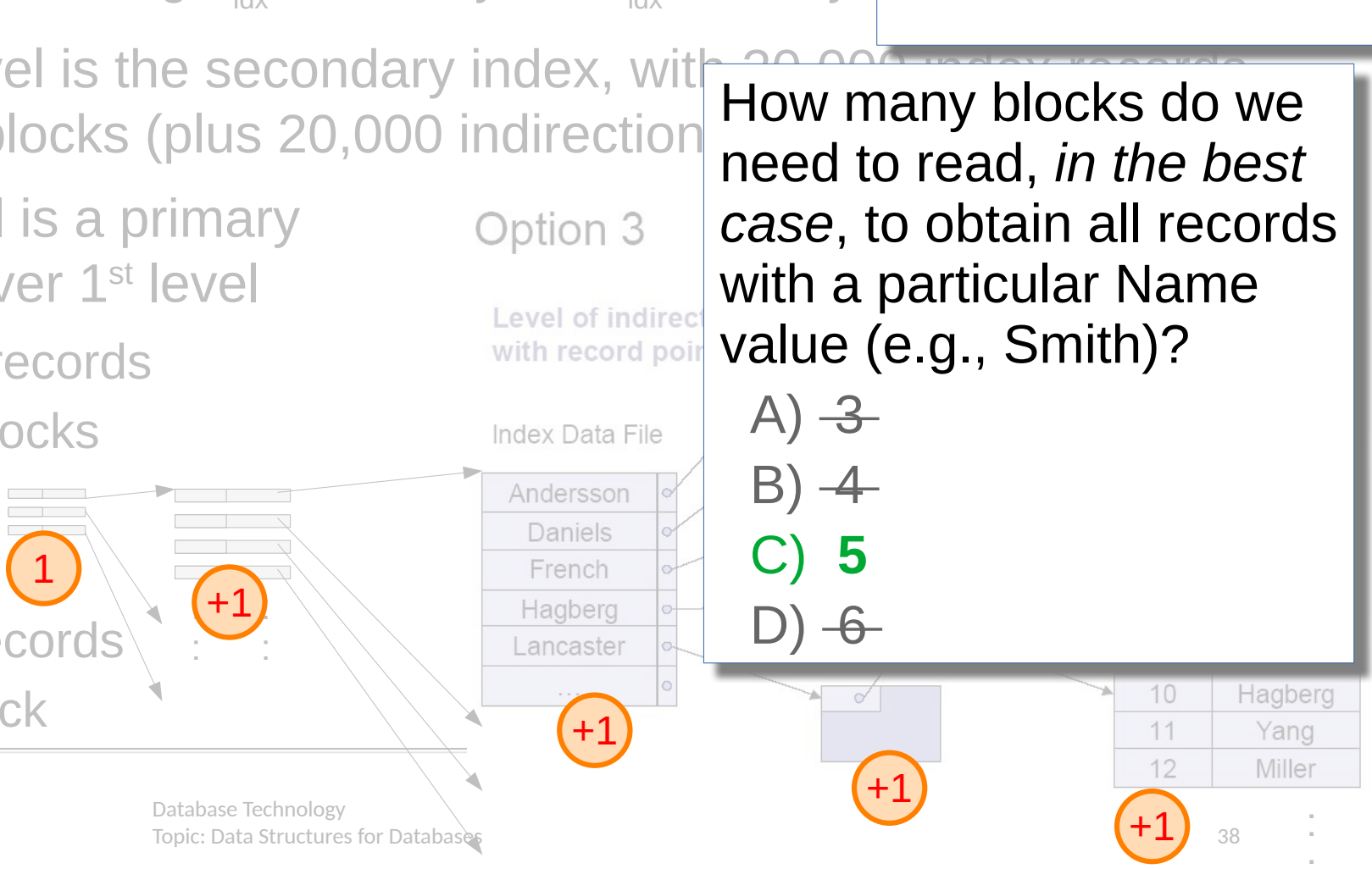
Multilevel Index

- To speed up finding records with a given Nam
  create a *multilevel* secondary index on the N
  - still assuming B_{idx} = 8,000 bytes, R_{idx} = 200 byte
- First level is the secondary index, with
  in 500 blocks (plus 20,000 indirection
- 2nd level is a primary index over 1st level
  - 500 records
  - 13 blocks
- 3rd level
  - 13 records
  - 1 block

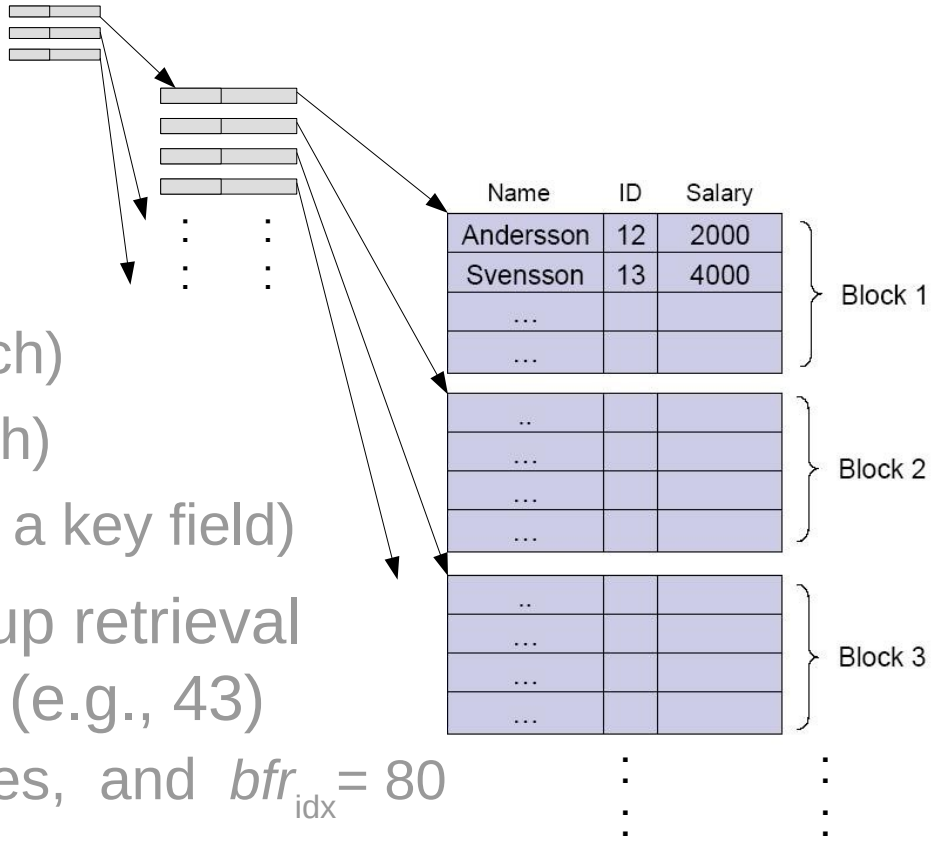
$\log_2(2)=1$ $\log_2(128)=7$
$\log_2(4)=2$ $\log_2(256)=8$
$\log_2(8)=3$ $\log_2(512)=9$
$\log_2(16)=4$ $\log_2(1024)=10$
$\log_2(32)=5$ $\log_2(2048)=11$
$\log_2(64)=6$ $\log_2(4096)=12$

How many blocks do we need to read, *in the best case*, to obtain all records with a particular Name value (e.g., Smith)?

A) ~~3~~
B) ~~4~~
C) **5**
D) ~~6~~

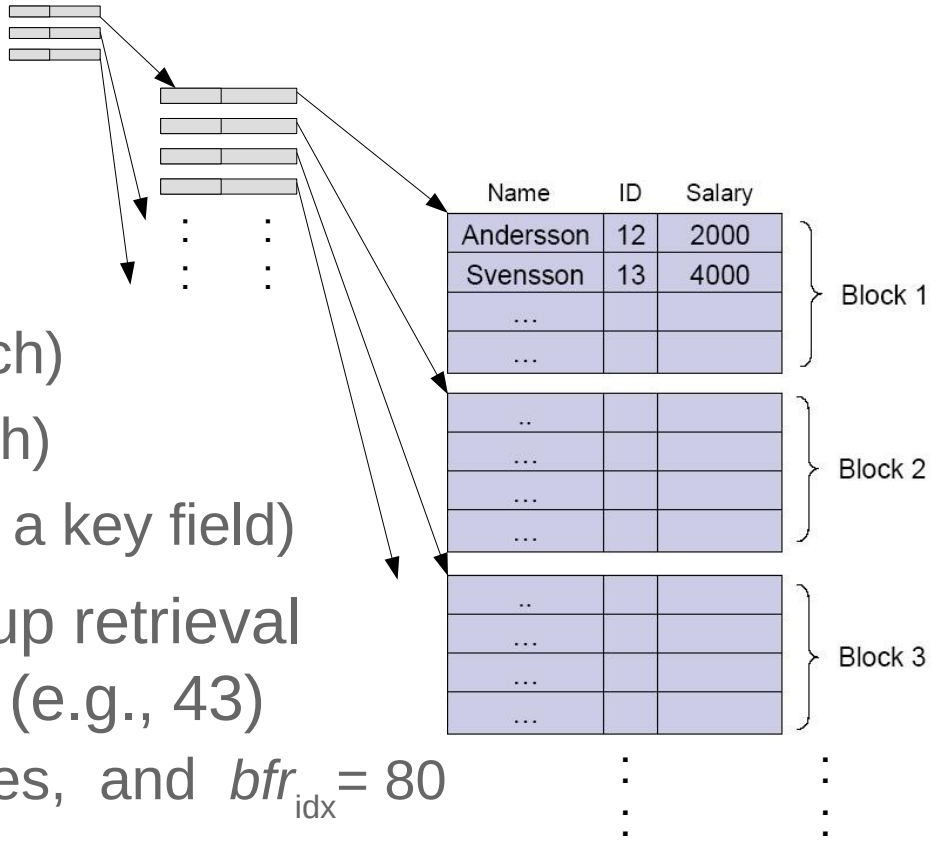


Multilevel Primary Index



- Assume a sorted file with
  - 200,000 records (400 bytes each)
  - 10,000 blocks (8,000 bytes each)
  - ID field as the *sorting field* (and a key field)
- Primary index (on ID) to speed up retrieval of records with a given ID value (e.g., 43)
  - B_{idx} = 8,000 bytes, R_{idx} = 100 bytes, and bfr_{idx} = 80
  - 10,000 index records and, thus, 125 blocks
- Extend this (single-level) primary index into a multilevel index
  - also here: B_{idx} = 8,000 bytes and R_{idx} = 100 bytes
- How many blocks do we need to read to find a record with a given ID value? A) 3 B) 4 C) 5 D) 6

Multilevel Primary Index



- Assume a sorted file with
  - 200,000 records (400 bytes each)
  - 10,000 blocks (8,000 bytes each)
  - ID field as the *sorting field* (and a key field)
- Primary index (on ID) to speed up retrieval of records with a given ID value (e.g., 43)
  - B_{idx} = 8,000 bytes, R_{idx} = 100 bytes, and bfr_{idx} = 80
  - 10,000 index records and, thus, 125 blocks
- Extend this (single-level) primary index into a multilevel index
  - also here: B_{idx} = 8,000 bytes and R_{idx} = 100 bytes
- How many blocks do we need to read to find a record with a given ID value?

A) ~~3~~ B) **4** C) ~~5~~ D) ~~6~~

= 3+1

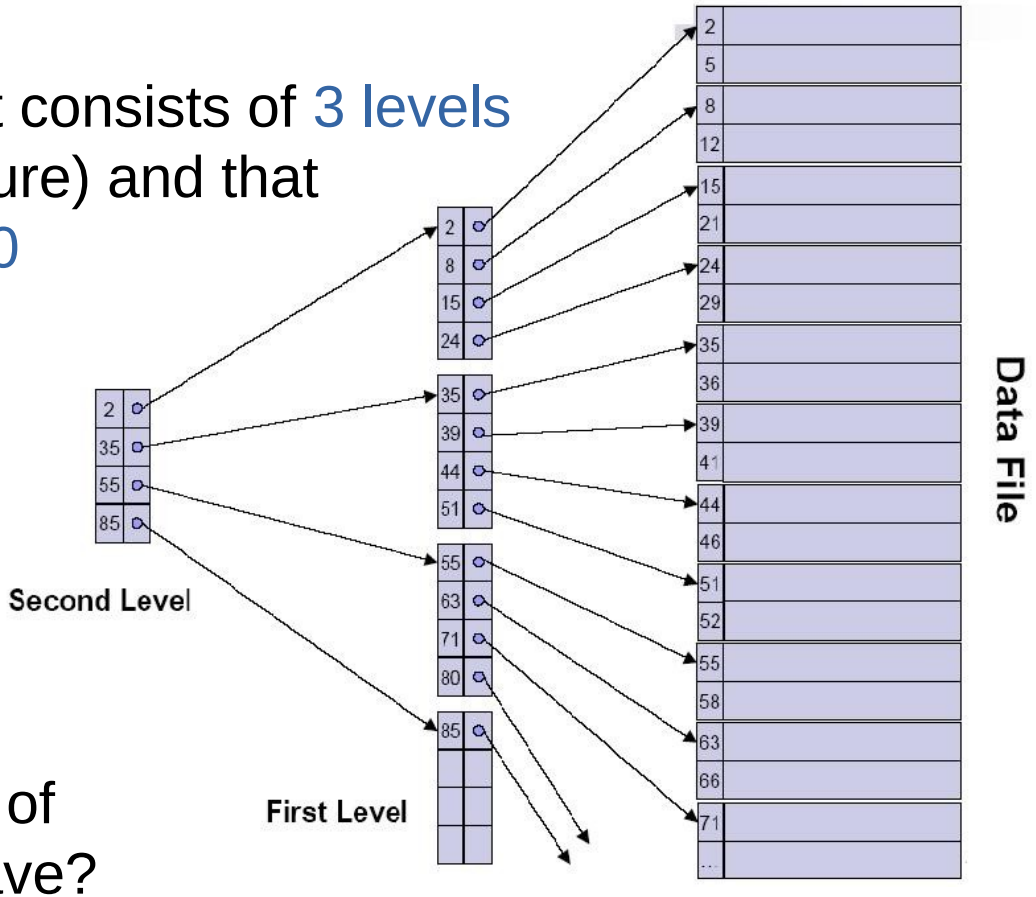
index lookups (one block access per level, 3 levels needed in this case)

read data file block that contains the record

Quiz

- Assume a multilevel index that consists of 3 levels (i.e., one more than in the picture) and that has a blocking factor (*bfr*) of 10 for all index levels
- Assume the first level is a primary index (i.e., the data file is sorted on a key field and the index has been created on this key field)
- What is the maximum number of blocks that the data file can have?

A) 3,000 B) 1,000 C) 300 D) 100

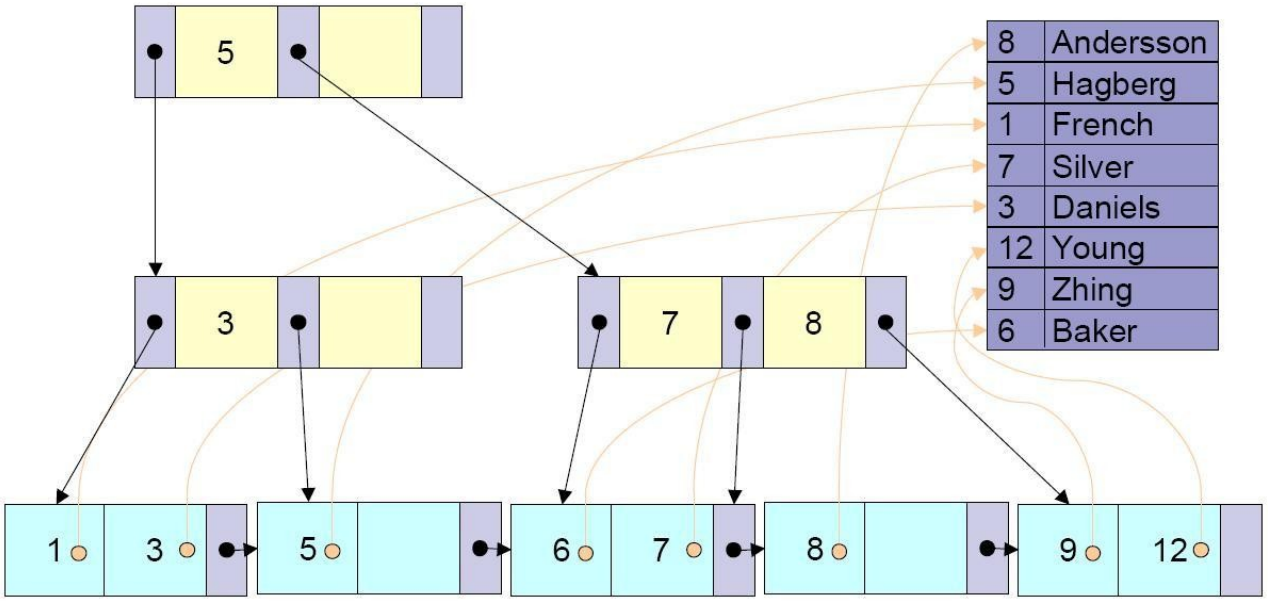


B^+-Trees

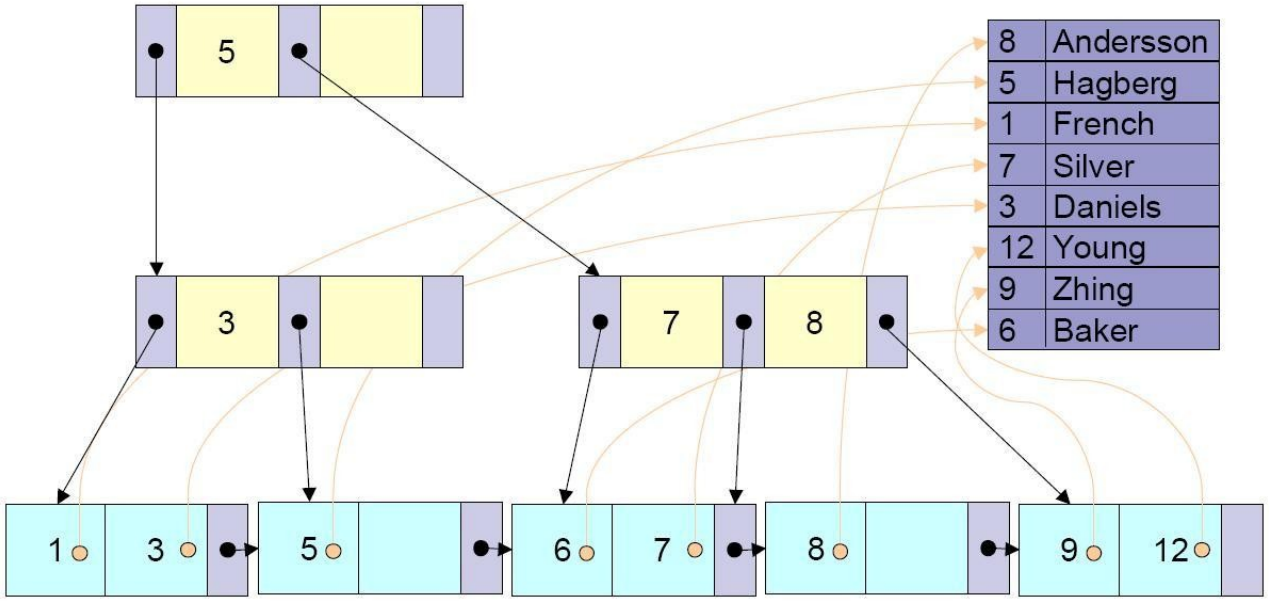
Dynamic Multilevel Indexes

Exercise: Insertion into a B$^+$-Tree

Assume we insert the field (10, Smith) into the data file, which means we need to update the B$^+$-tree accordingly.
How does the tree look after this insertion?



Quiz: Properties of B+-Trees



One of the following statements *is wrong*. Which one?

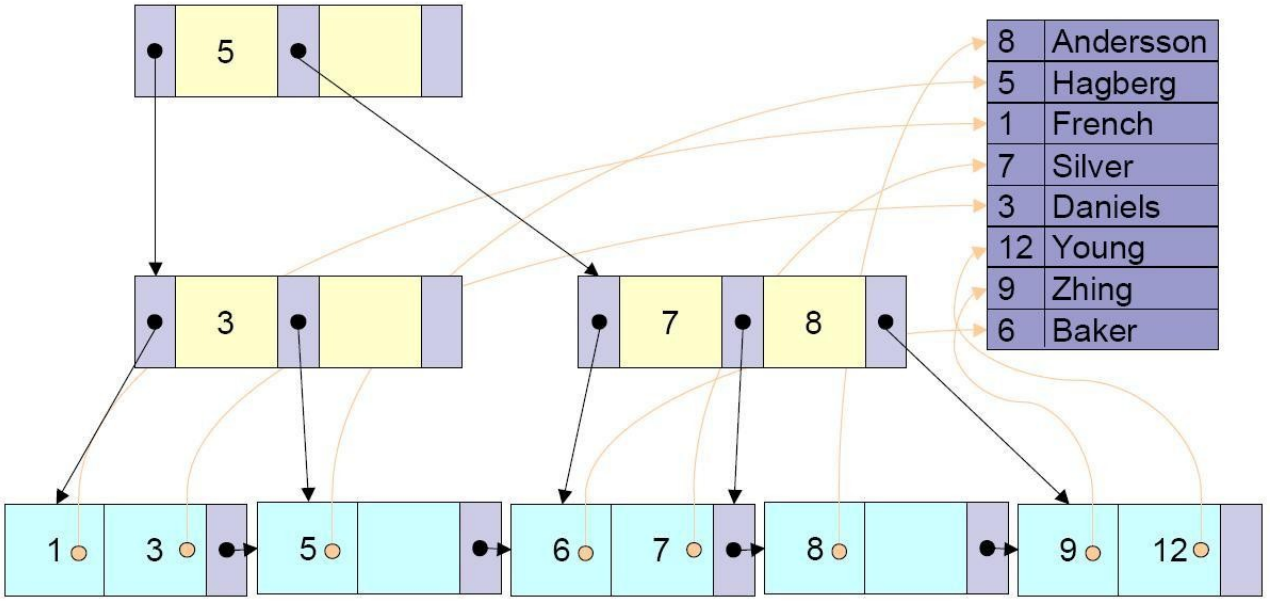
A) The order (*p*) of a B+-tree determines both the maximum and the minimum number of index values that internal nodes can have.

B) In terms of number of block reads during search in a B+-tree, it is best if all internal nodes are filled to the maximum.

C) During insertion, if a leaf node needs to be split, the index value propagated to the parent node is appended to that parent node (assuming there is still space in the parent node).

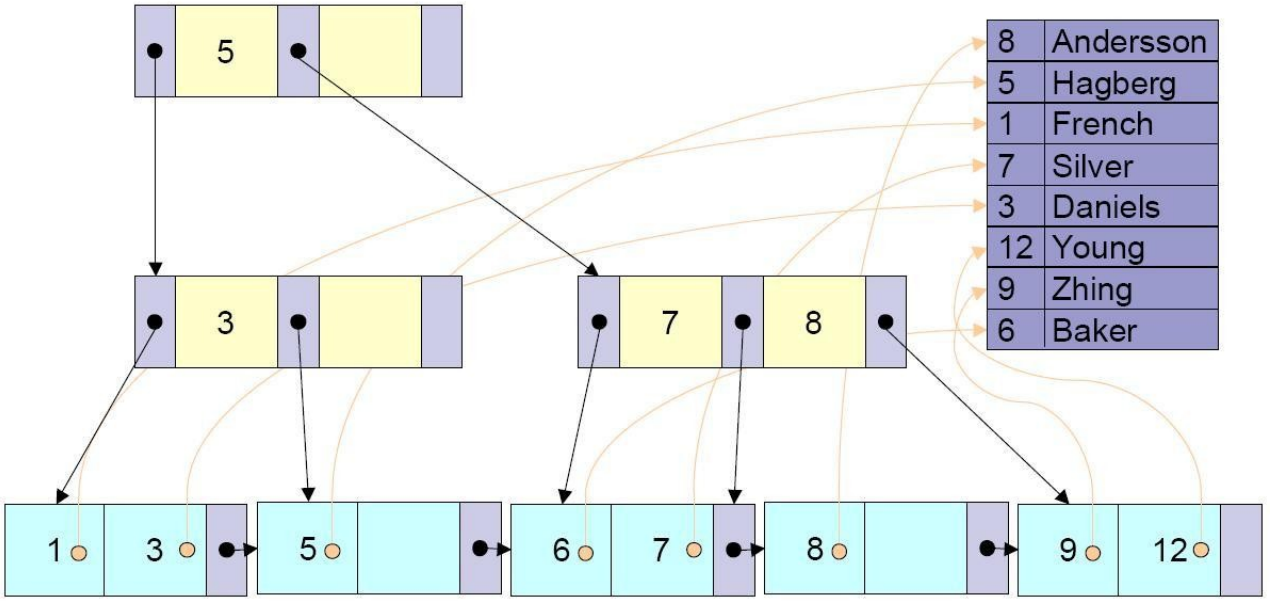
D) There may be cases in which an internal node of a B+-tree contains an index value that is not in any leaf node.

Exercise



- Remember that each node of a B$^+$-tree is stored as a file block
- Assume a case in which:
  - each file block can store 8,000 bytes
  - each sub-tree pointer P_i requires 100 bytes
  - each index value K_i requires 100 bytes
- What is the order (p) of the B$^+$-tree in this case?
- What is the minimum and the maximum number of tree pointers that each internal node may contain?
- What is the minimum and the maximum number of index values that each internal node may contain?

Exercise



- Remember that each node of a B$^+$-tree is stored as a file block
- Assume a case in which:
  - each file block can store 8,000 bytes
  - each sub-tree pointer P_i requires 100 bytes
  - each index value K_i requires 100 bytes
- What is the order (p) of the B$^+$-tree in this case? 40
- What is the minimum and the maximum number of tree pointers that each internal node may contain? 20 and 40
- What is the minimum and the maximum number of index values that each internal node may contain? 19 and 39

www.liu.se

LINKÖPING UNIVERSITY

Binary Search