Database Technology

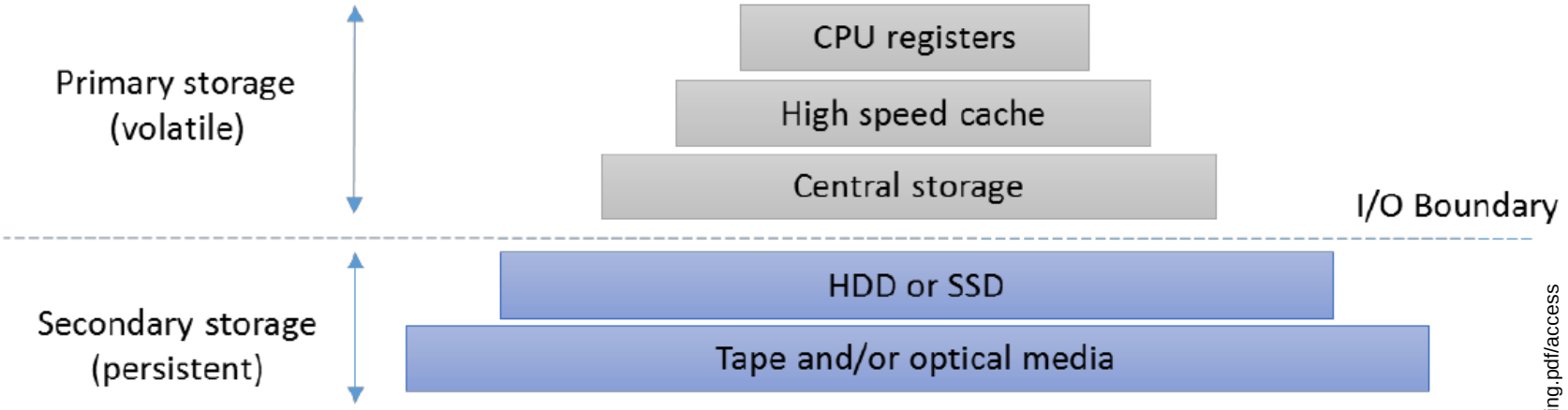
Data Structures for Databases

Olaf Hartig
olaf.hartig@liu.se

LINKÖPING UNIVERSITY

Storage Hierarchy

Quiz



Which of the following statements *is correct*?

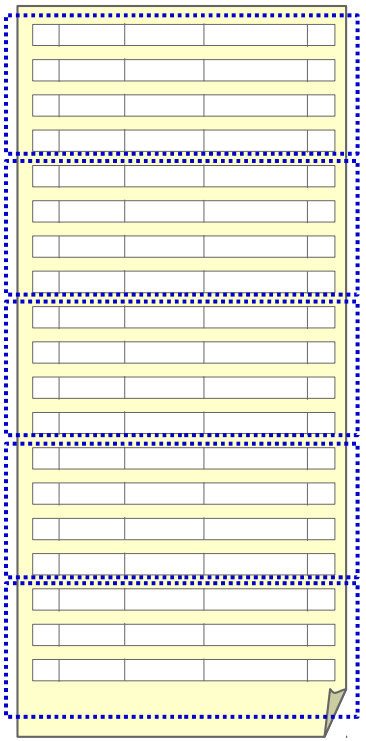
1) Secondary storage devices are usually faster than primary storage devices.
2) Data in a primary storage device may be lost when switching off the power.
3) The CPU may operate directly on data that is in a secondary storage device.
4) A piece of data (e.g., a record) may not be held both in a primary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

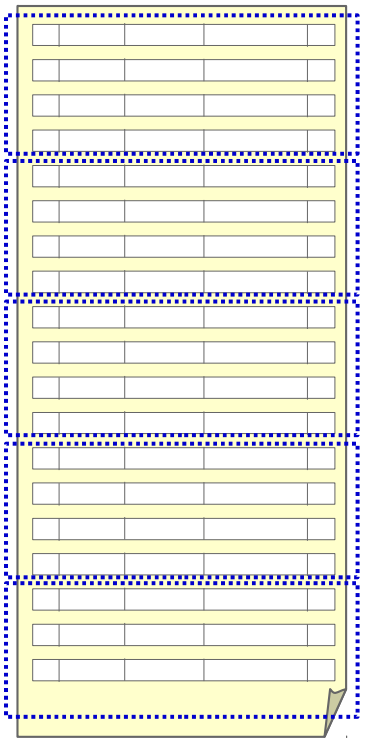


- Assume a file with
  - r = 200,000 records,
  - R = 400 bytes per record, and
  - B = 8,000 bytes per block

- 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

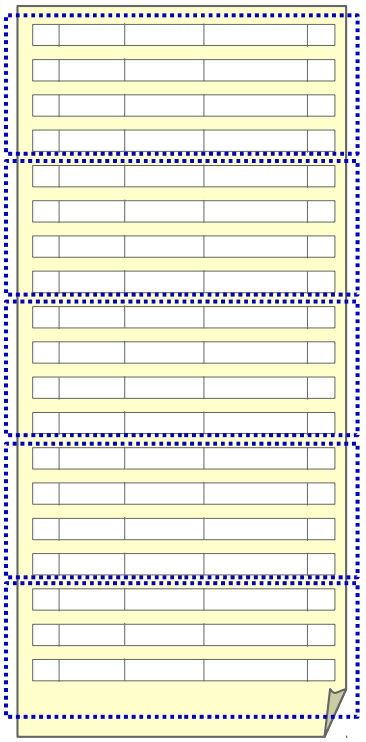
- How many blocks are needed to store the file?

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?

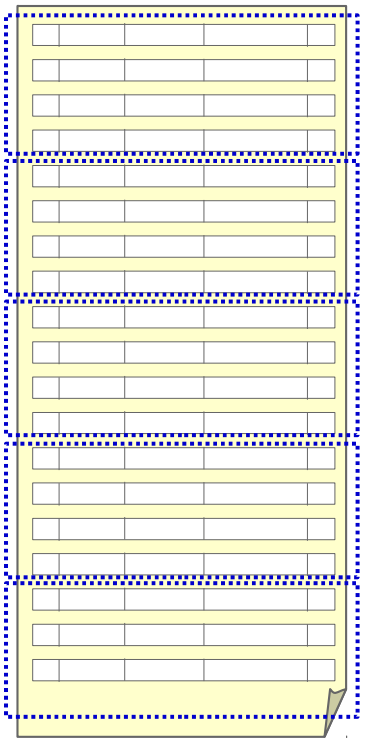
1) ~~b = 1,000~~ 2) ~~b = 2,000~~ 3) ~~b = 8,000~~ 4) b = 10,000

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

- 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

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

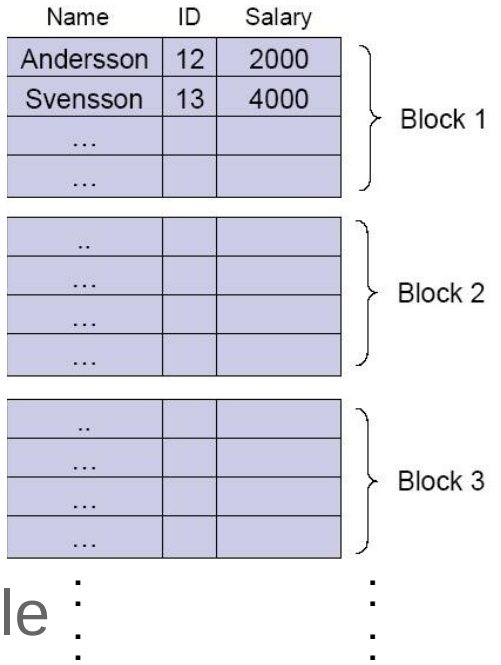
- How much space is wasted per block? $B - bfr * R$

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

File Organization

(Organizing Records in Files)

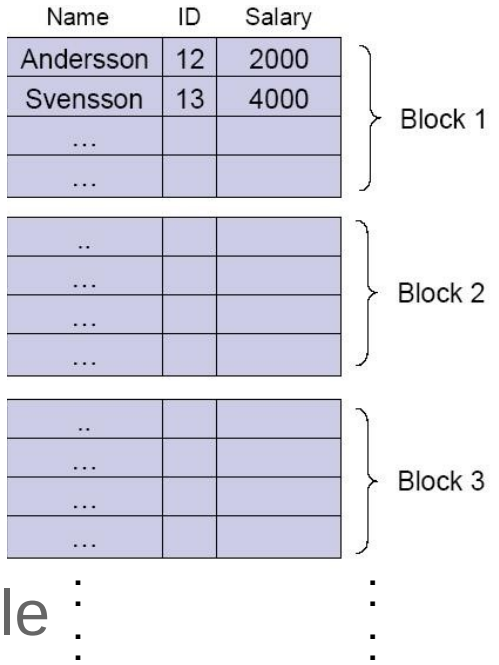
Exercise: Heap File



- 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

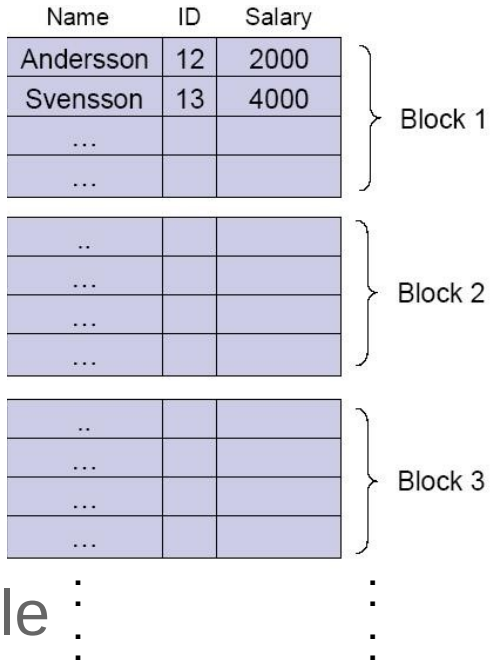


- Assume a file with
  - r = 200,000 records,
  - R = 400 bytes per record, and
  - B = 8,000 bytes per block
- Hence, b = 10,000 blocks needed to store the file
- Assume we organize the file as a heap file
  - i.e., new records are always appended to the end of the file
- How many blocks do we need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

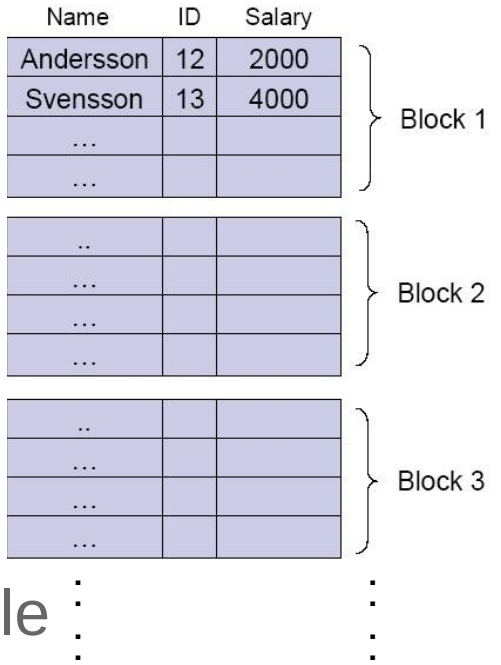


- Assume a file with
  - r = 200,000 records,
  - R = 400 bytes per record, and
  - B = 8,000 bytes per block
- Hence, b = 10,000 blocks needed to store the file
- Assume we organize the file as a heap file
  - i.e., new records are always appended to the end of the file
- How many blocks do we need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

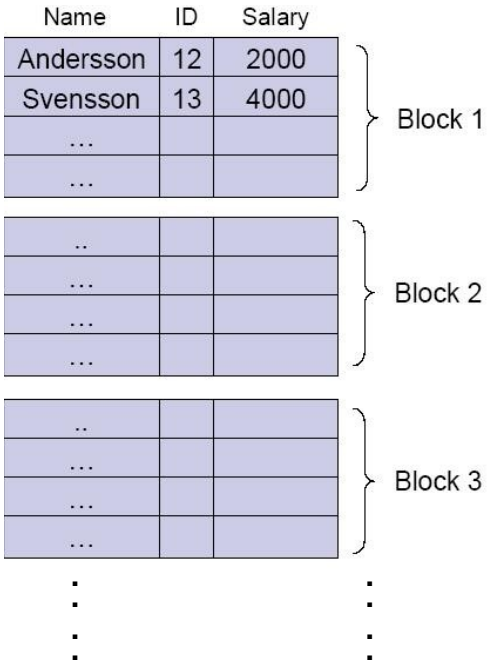
	search field = ID value = 43 (unique)	search field = Name value = Smith (non-unique)
worst case	10,000	10,000
best case	1	10,000
average case	5,000	10,000

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

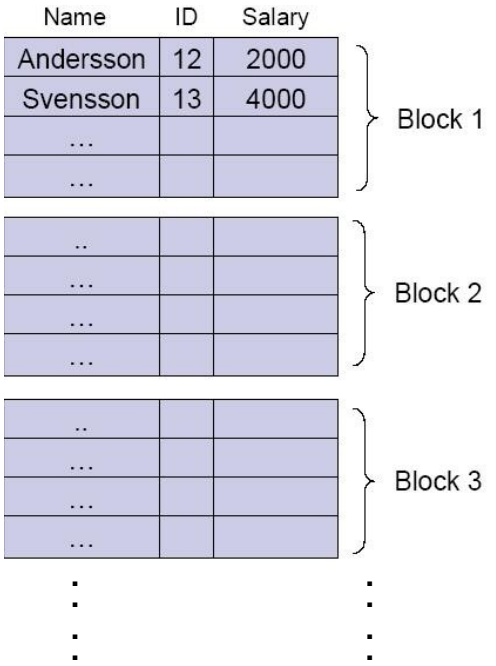


	search field = ID value = 43 (unique)	search field = Name value = Smith (non-unique)
worst case	?	?
best case	?	?
average case	?	?

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

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



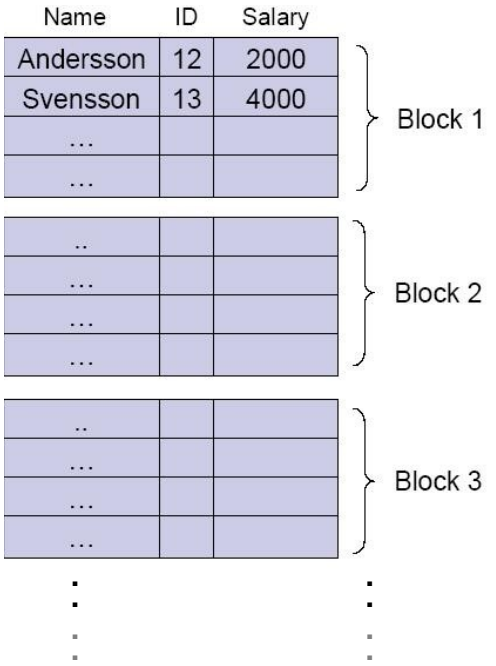
	search field = ID value = 43 (unique)	search field = Name value = Smith (non-unique)
worst case	14	10,000
best case	1	10,000
average case	14	10,000

$\log_2 b$

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

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

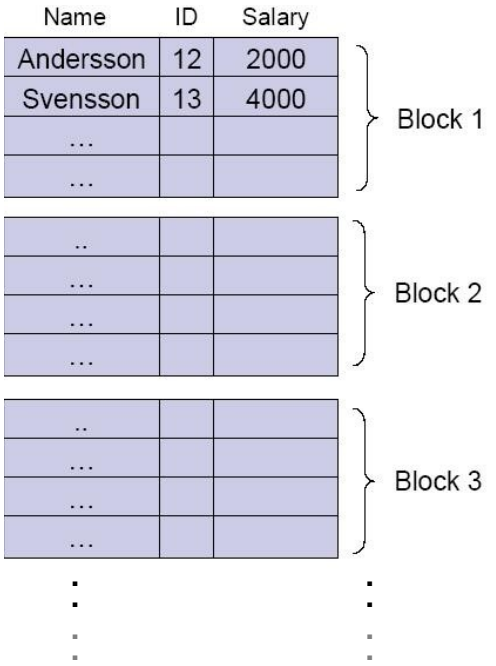
- Assume a file with
  - r = 200,000 records,
  - R = 400 bytes per record, and
  - B = 8,000 bytes per block
- Hence, b = 10,000 blocks needed for the file
- Assume we organize the file as a hash file by using the ID field as the *hash field*
  - i.e., find relevant bucket by applying hash function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

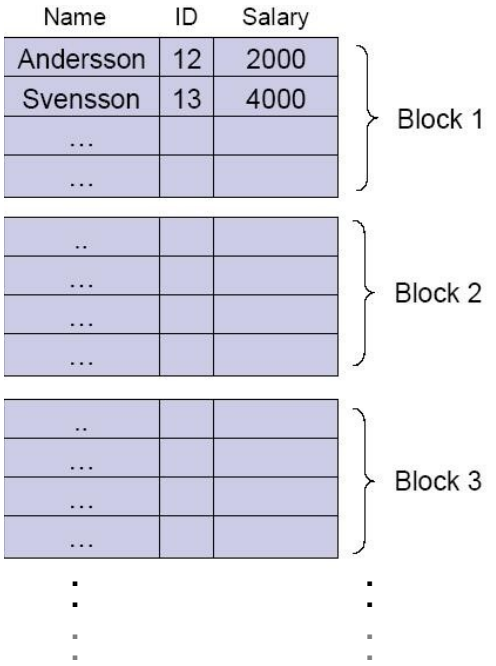


	search field = ID value = 43 (unique)	search field = Name value = Smith (non-unique)
worst case	4	≥ 10,000
best case	1	≥ 10,000
average case	depends	≥ 10,000

scan all non-empty blocks of all buckets

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

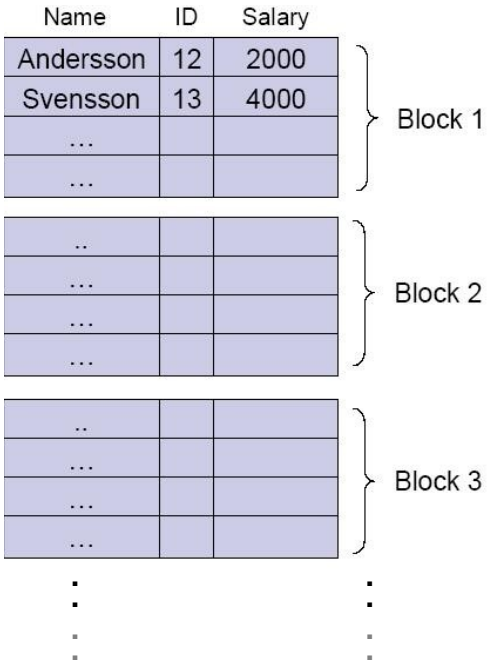
- Assume a file with
  - r = 200,000 records,
  - R = 400 bytes per record, and
  - B = 8,000 bytes per block
- Hence, b = 10,000 blocks needed for the file
- Assume we organize the file as a hash file by using the ID field as the *hash field*
  - i.e., find relevant bucket by applying hash function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

Single-Level Ordered Indexes

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*?

1) Primary index
2) Clustering index
3) Secondary index on a key field
4) Secondary index on a non-key field

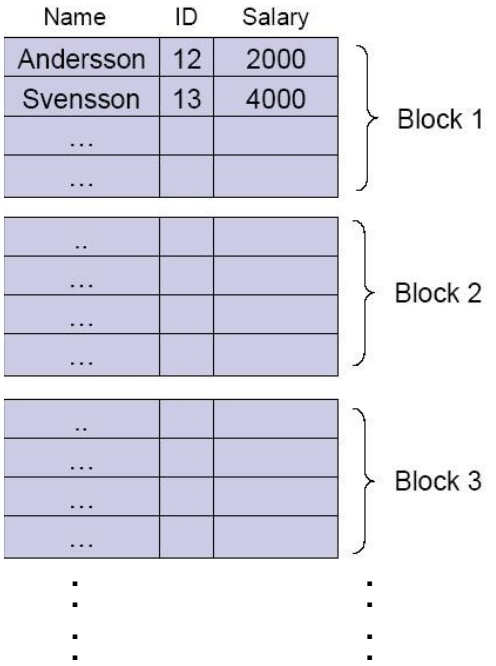
Summary of Single-Level Indexes (cont'd)

	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

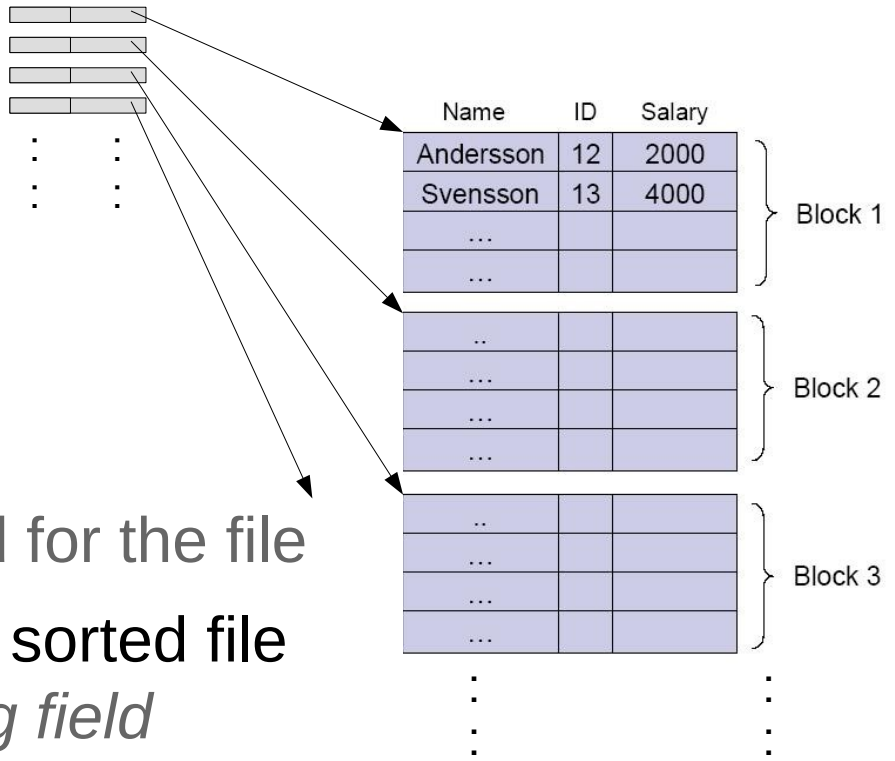
- Assume a file with
  - r = 200,000 records,
  - R = 400 bytes per record, and
  - B = 8,000 bytes per block
- Hence, b = 10,000 blocks needed for the file
- Assume we organize the file as a sorted file by using the ID field as the *sorting field*



- 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} = 200 bytes per index record
- How many index records does this index contain?

  1) 8,000 2) 10,000 3) 20,000 4) 200,000

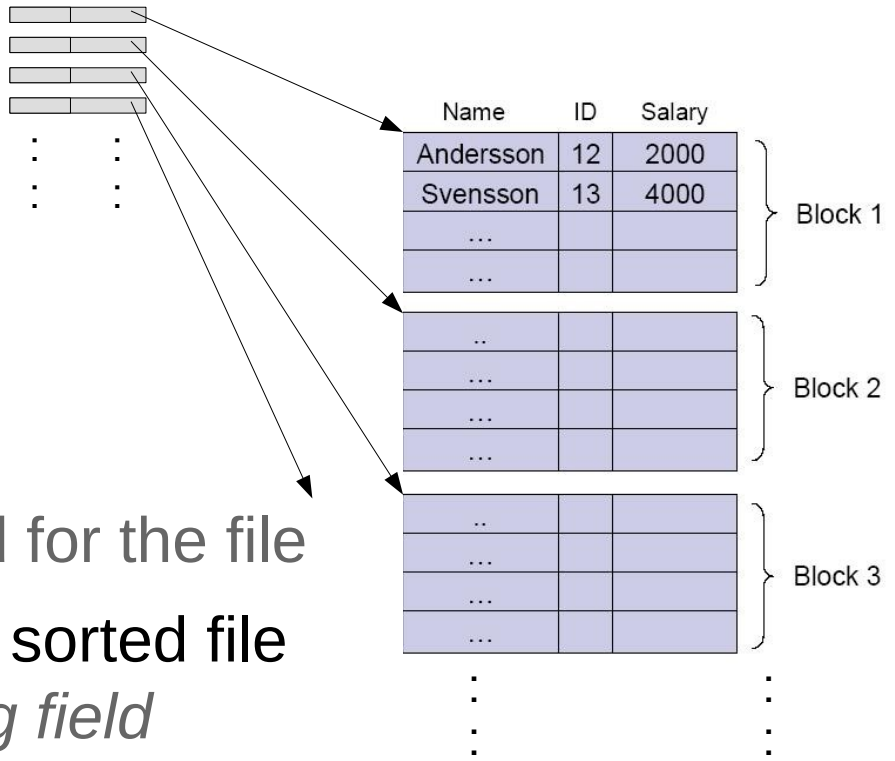
Quiz: Primary Index



- 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} = 200 bytes per index record
- How many index records does this index contain?

1) ~~8,000~~ 2) 10,000 3) ~~20,000~~ 4) ~~200,000~~

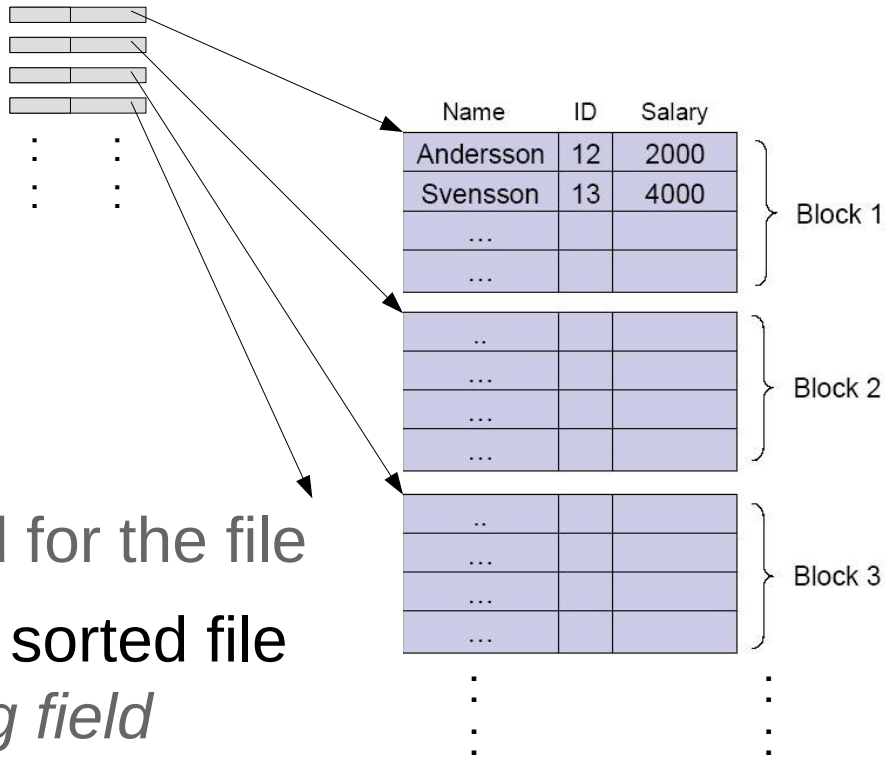
Quiz: Primary Index



- 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} = 200 bytes per index record
- How many blocks does the index file consist of?

  1) 100 2) 250 3) 1,000 4) 8,000

Quiz: Primary Index

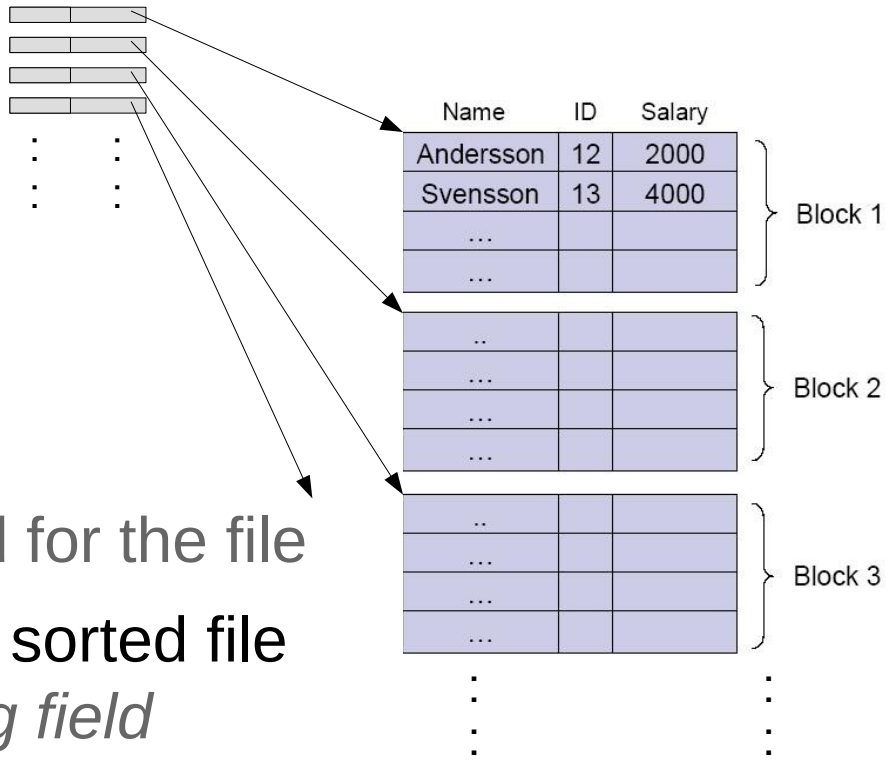


- Assume a file with
  - r = 200,000 records,
  - R = 400 bytes per record, and
  - B = 8,000 bytes per block
- Hence, b = 10,000 blocks needed for the file
- Assume we organize the file as a sorted file by using the ID field as the *sorting field*
- Assume we create a primary index on the ID field
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

1) ~~100~~ 2) 250 3) ~~1,000~~ 4) ~~8,000~~

$$b_{idx} = \left\lceil \frac{r_{idx}}{bfr_{idx}} \right\rceil = \left\lceil \frac{10{,}000}{40} \right\rceil = 250 \qquad bfr_{idx} = \left\lfloor \frac{B_{idx}}{R_{idx}} \right\rfloor = \left\lfloor \frac{8{,}000}{200} \right\rfloor = 40$$

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 250 blocks
- How many blocks do we need to read if we want to retrieve the record with ID = 43?

  1) 7 2) 8 3) 9 4) 10

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

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

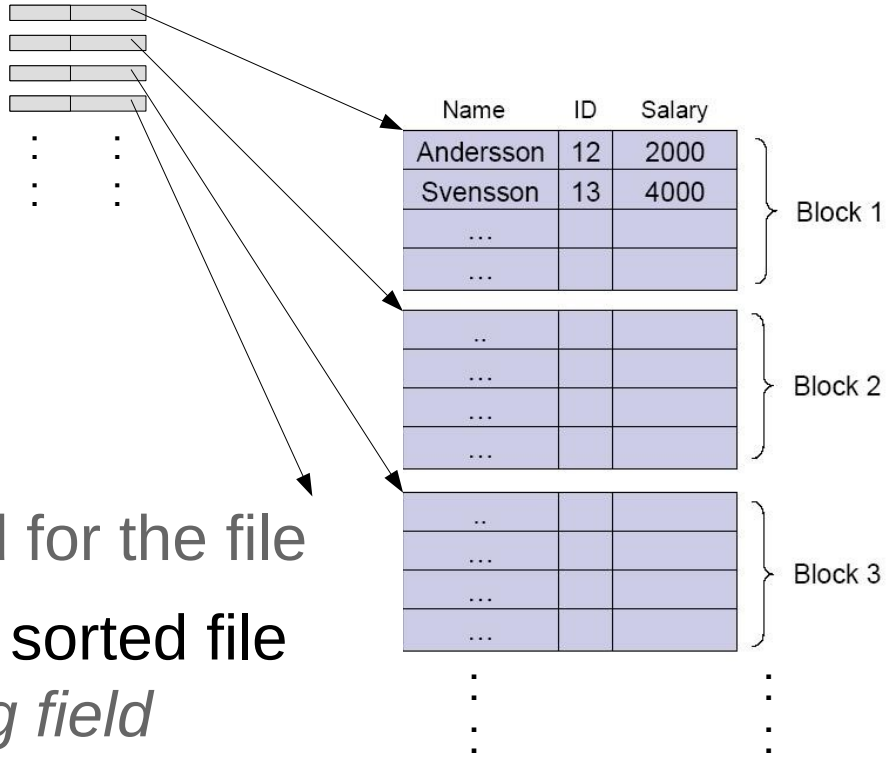
$\log_2(8)=3$ $\log_2(512)=9$

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

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

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

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 250 blocks
- How many blocks do we need to read if we want to retrieve the record with ID = 43?

1) ~~7~~ 2) ~~8~~ 3) 9 4) ~~10~~

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

binary search in the index

read data file block that contains the record

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

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

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

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

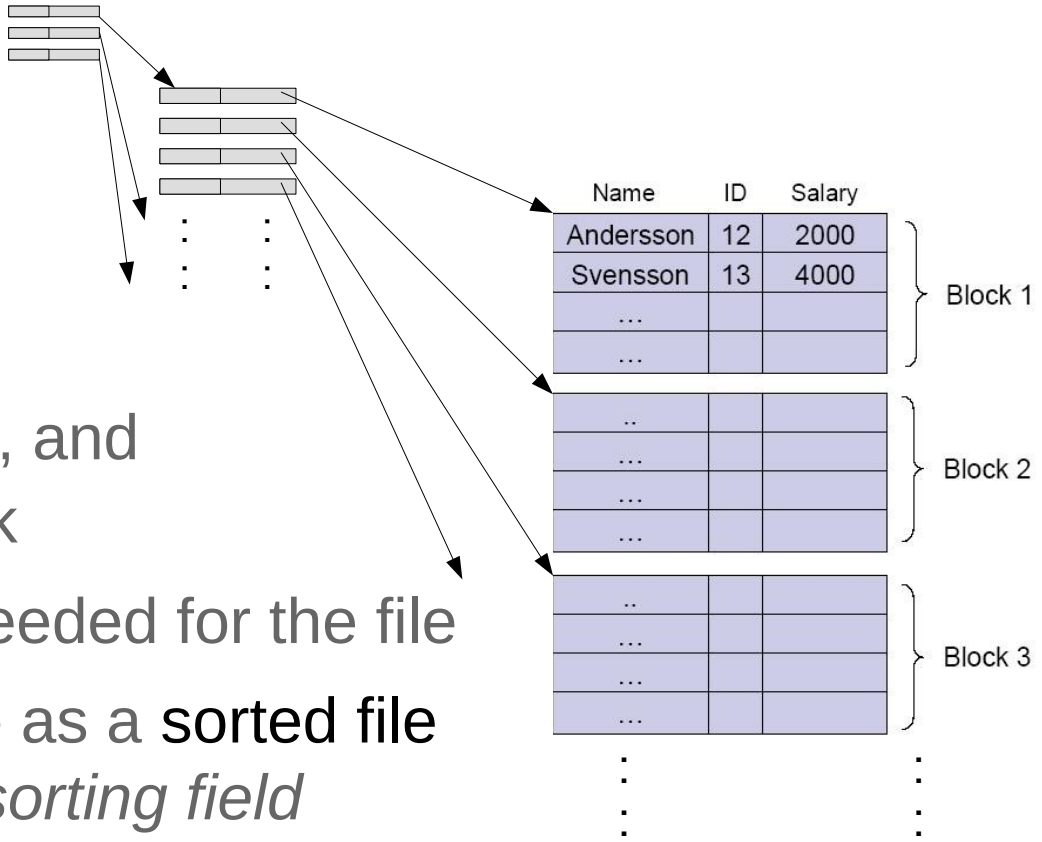
$\log_2(32)=5$ $\log_2(2048)=11$

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

Multilevel Indexes

(Stacking indexes on top of one another)

Multilevel 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*
- Convert the primary index (on ID) into a multilevel index
  - still 250 blocks in the first level
  - second level: 250 index records → 7 blocks (at bfr_{idx} = 40)
  - third level: 7 index records → 1 block
- Retrieving a record by ID: 3 + 1 = 4 block accesses

index lookups (one block access per level)

read data file block that contains the record

www.liu.se

LINKÖPING UNIVERSITY

Binary Search