Monday, November 26, 12

The CouchDB Implementation

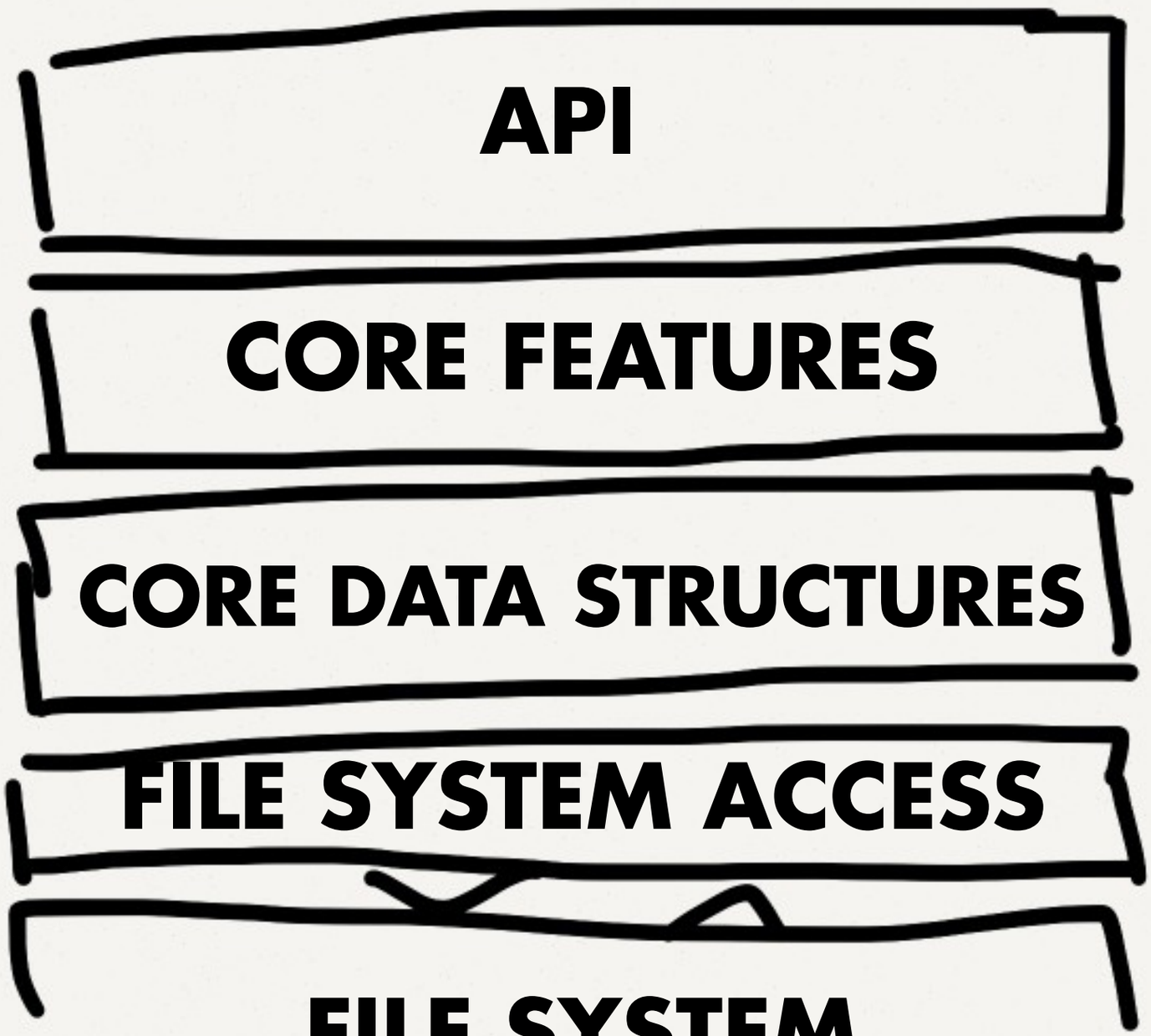
Jan Lehnardt
jan@a.o
@janl
CouchDB PMC Chair
Committer #2

Thanks for the invite
Glad to be here

JSConf EU / BBuzz / JSFAB

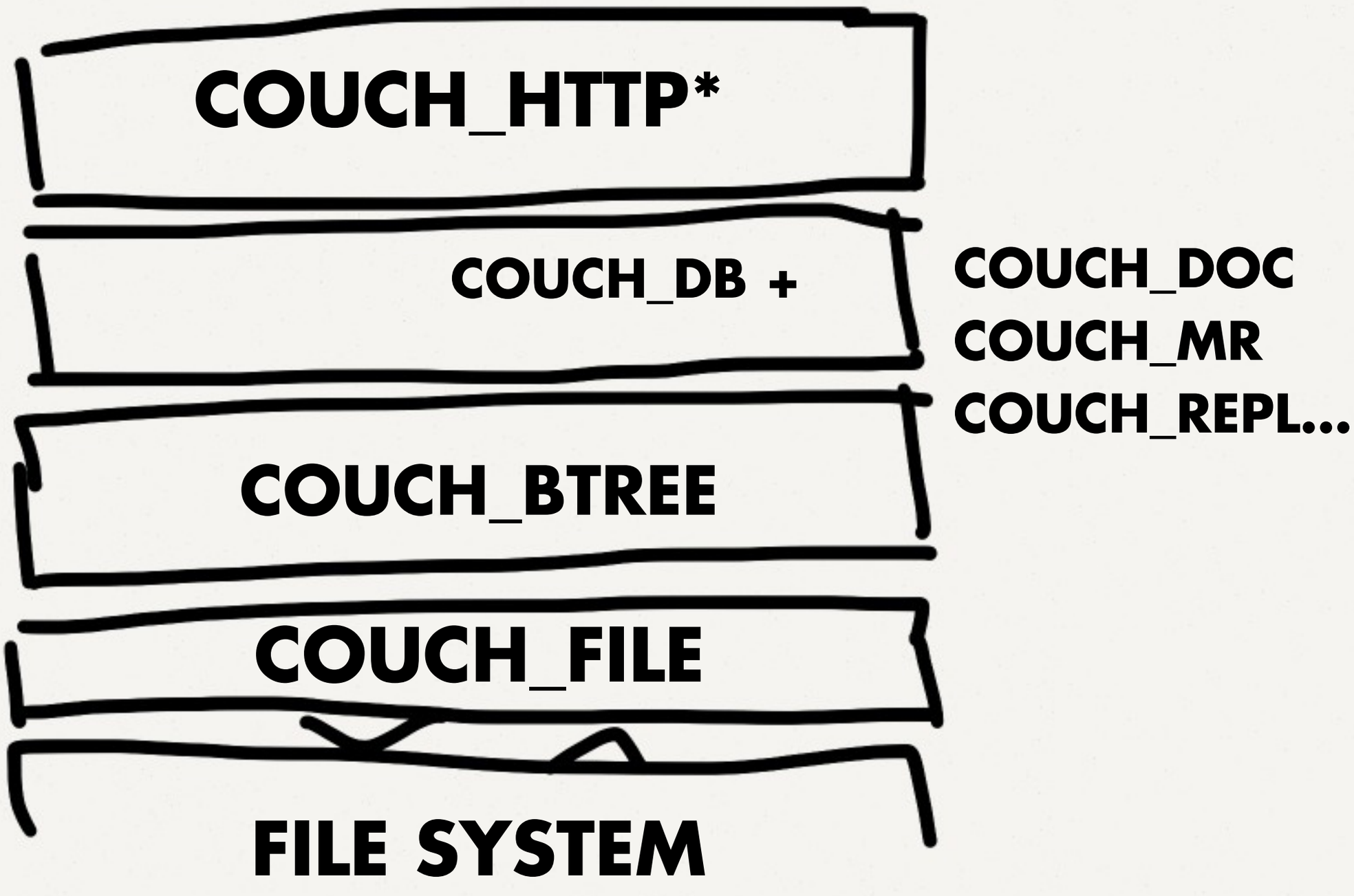
Any Database

- FS integration / raw storage
- core data structures
- core features
- API


API
CORE FEATURES
CORE DATA STRUCTURES
FILE SYSTEM ACCESS
FILE SYSTEM

CouchDB is no different

- couch_file
- couch_btree
- couch_db / couch_doc / couch_mr / couch_replicator / etcpp
- couch_httpd*


COUCH_HTTP*
COUCH_DB +
COUCH_DOC
COUCH_MR
COUCH_REPL...
COUCH_BTREE
COUCH_FILE
FILE SYSTEM

Core

Datastructures

- 2 x b+tree & data interleaved
- append only, mvcc
- full fsync control

- Can answer:
  - Data for $key
  - What happened $since

- Used for core data storage
- As well as indexes

- Everything else is built on top

Behold the

b+tree

Monday, November 26, 12

by-id

Monday, November 26, 12

A
B
C
D
E
F
G
H
I
J
K
L
...

A
B
C
D
E
F
DOC_G
H
I
J
K
L
...

A
B
C
DOC_D
E
F
DOC_G
H
I
J
K
L
...

A
B
C
DOC_D
E
F
DOC_G
H
I
J
DOC_K
L
...

DOC_D
DOC_G
DOC_K

by-seq
or
“what happened since?”

Monday, November 26, 12

1. DOC_G

2. DOC_D

3. DOC_K

The CouchDB File Format

- 2 x b+tree & data interleaved
- append only, mvcc
- full fsync control

- Can answer:
  - Data for $key
  - What happened $since

- Used for core data storage
- As well as indexes

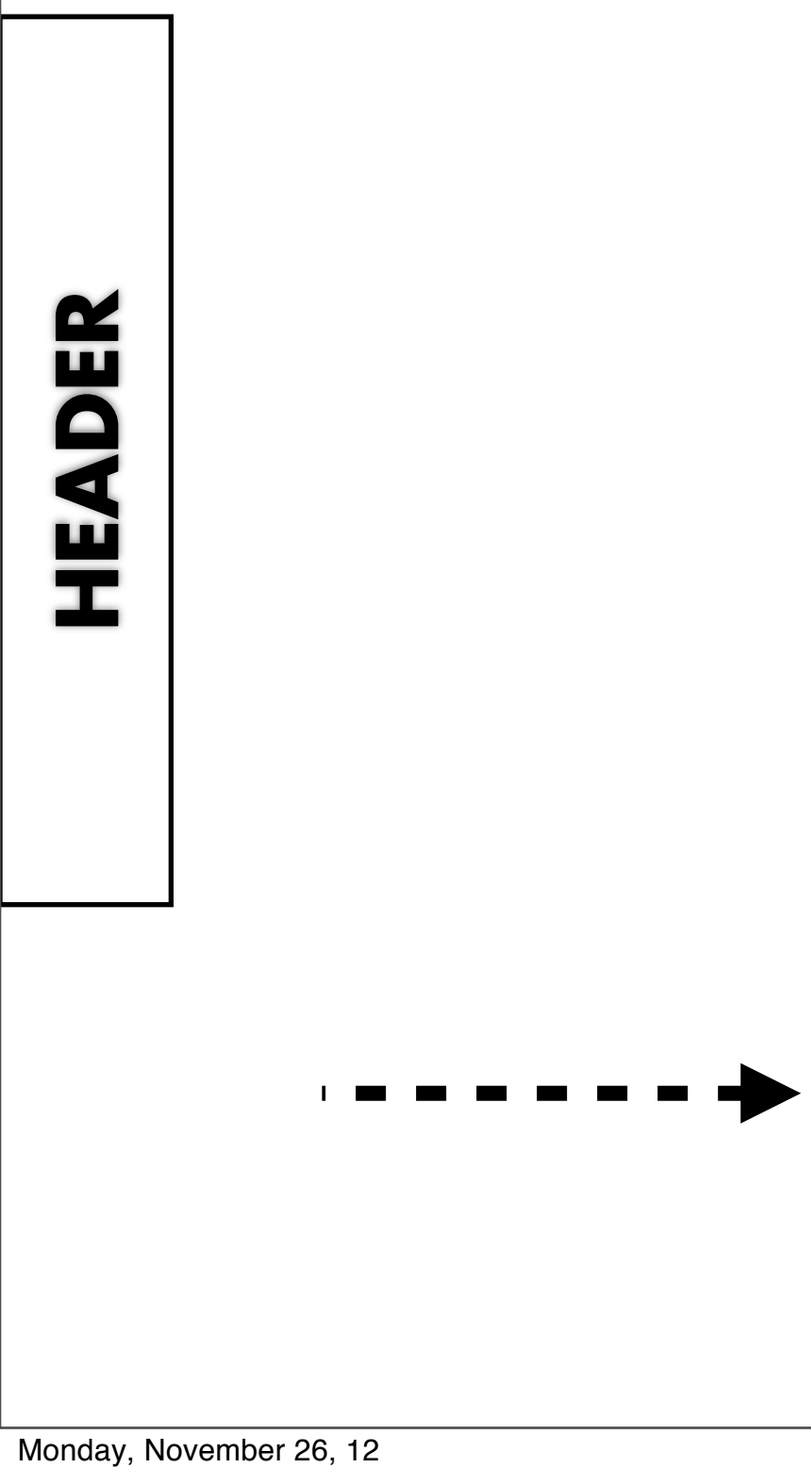
- Everything else is built on top

- 2 x b+tree & data interleaved
- append only, mvcc
- full fsync control

- Can answer:
  - Data for $key
  - What happened $since

- Used for core data storage
- As well as indexes

- Everything else is built on top

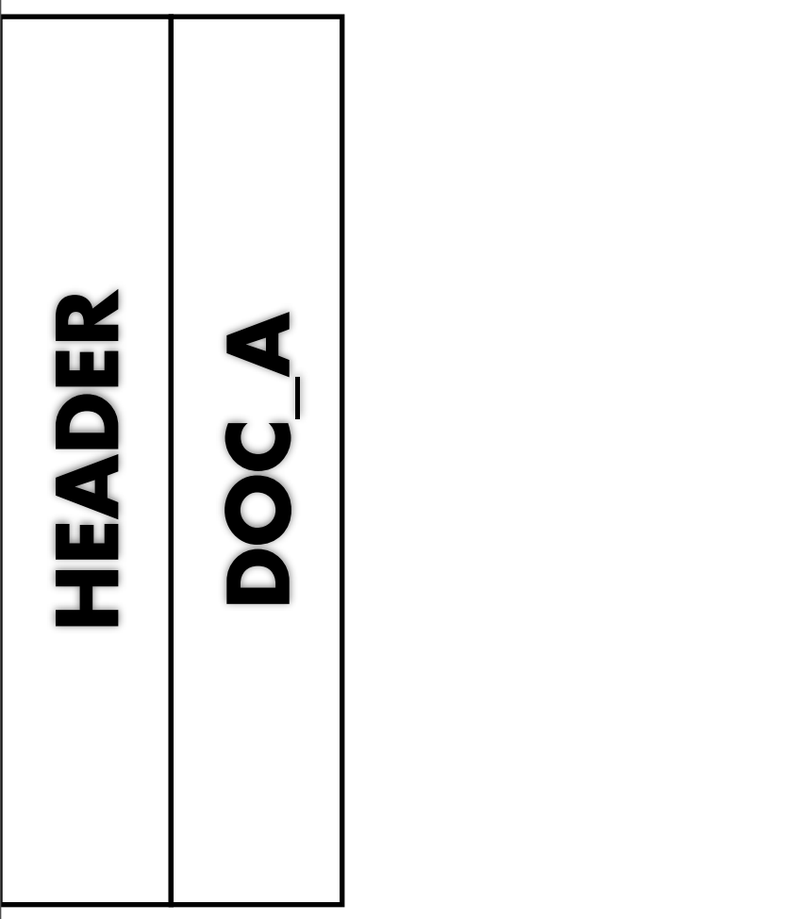


- 2 x b+tree & data interleaved
- append only, mvcc
- full fsync control

- Can answer:
  - Data for $key
  - What happened $since

- Used for core data storage
- As well as indexes

- Everything else is built on top

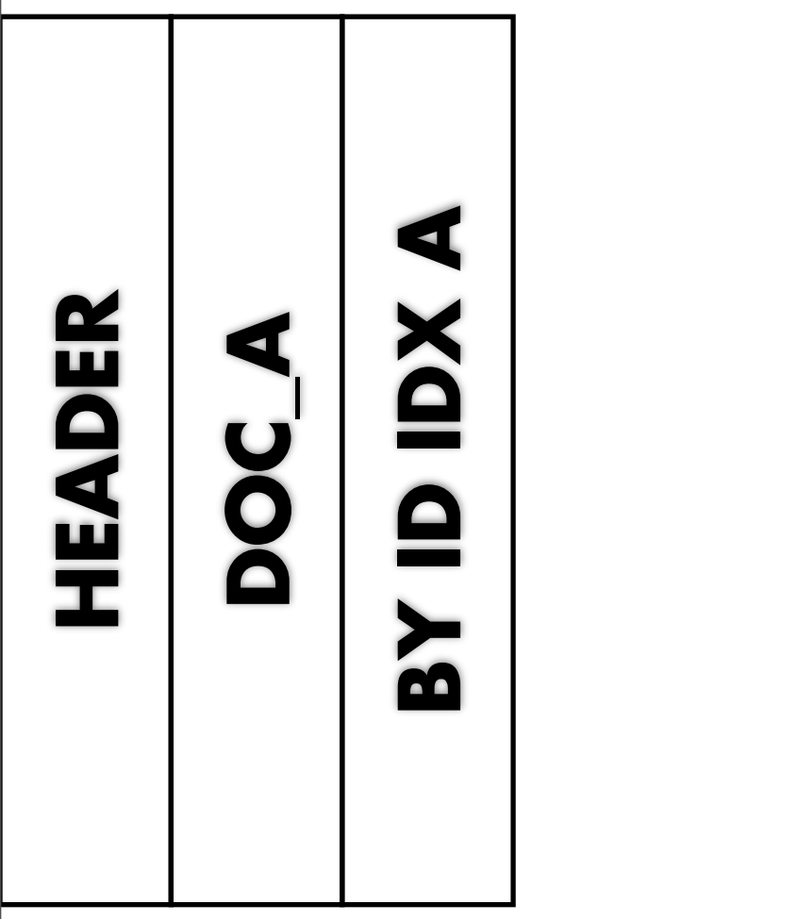


- 2 x b+tree & data interleaved
- append only, mvcc
- full fsync control

- Can answer:
  - Data for $key
  - What happened $since

- Used for core data storage
- As well as indexes

- Everything else is built on top

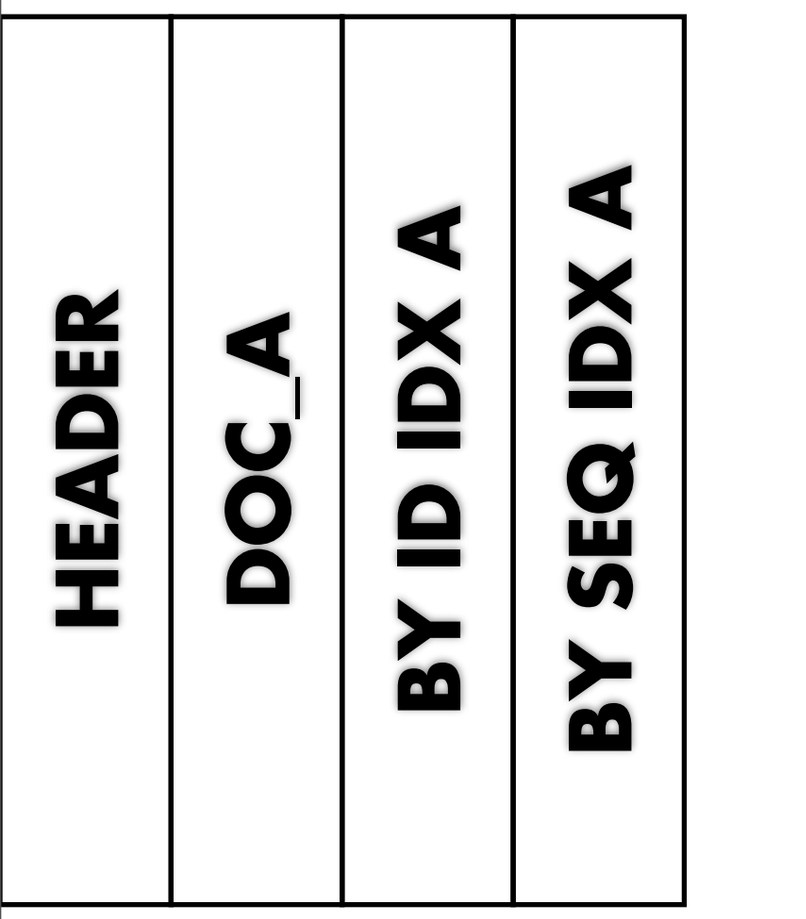


- 2 x b+tree & data interleaved
- append only, mvcc
- full fsync control

- Can answer:
  - Data for $key
  - What happened $since

- Used for core data storage
- As well as indexes

- Everything else is built on top

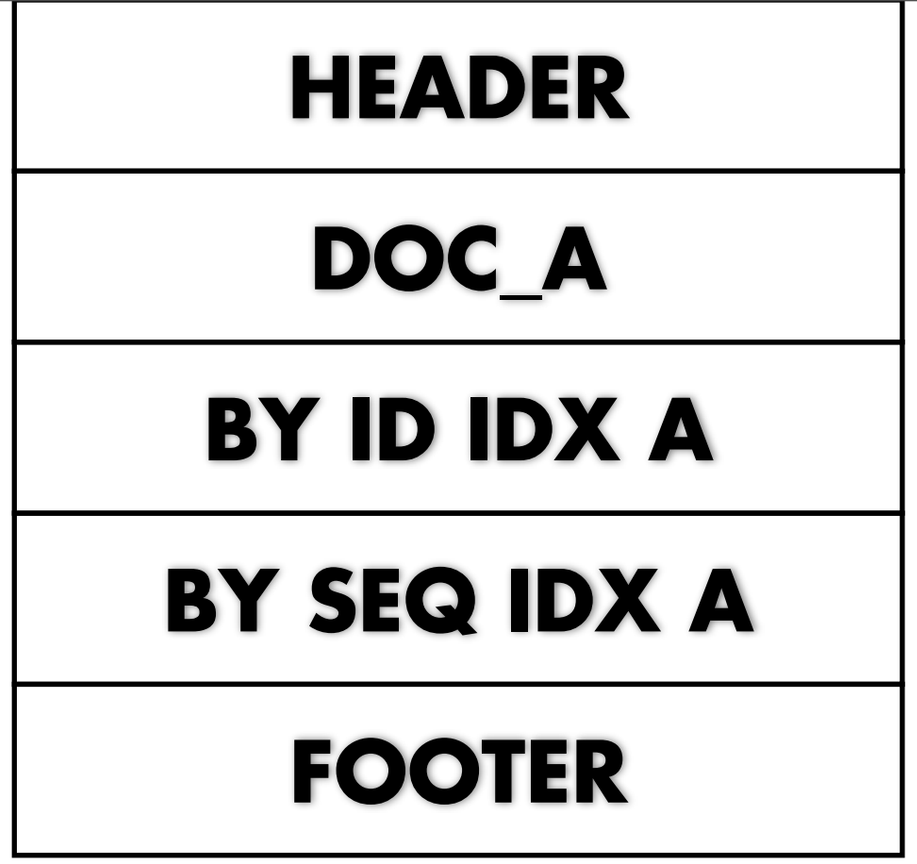


- 2 x b+tree & data interleaved
- append only, mvcc
- full fsync control

- Can answer:
  - Data for $key
  - What happened $since

- Used for core data storage
- As well as indexes

- Everything else is built on top

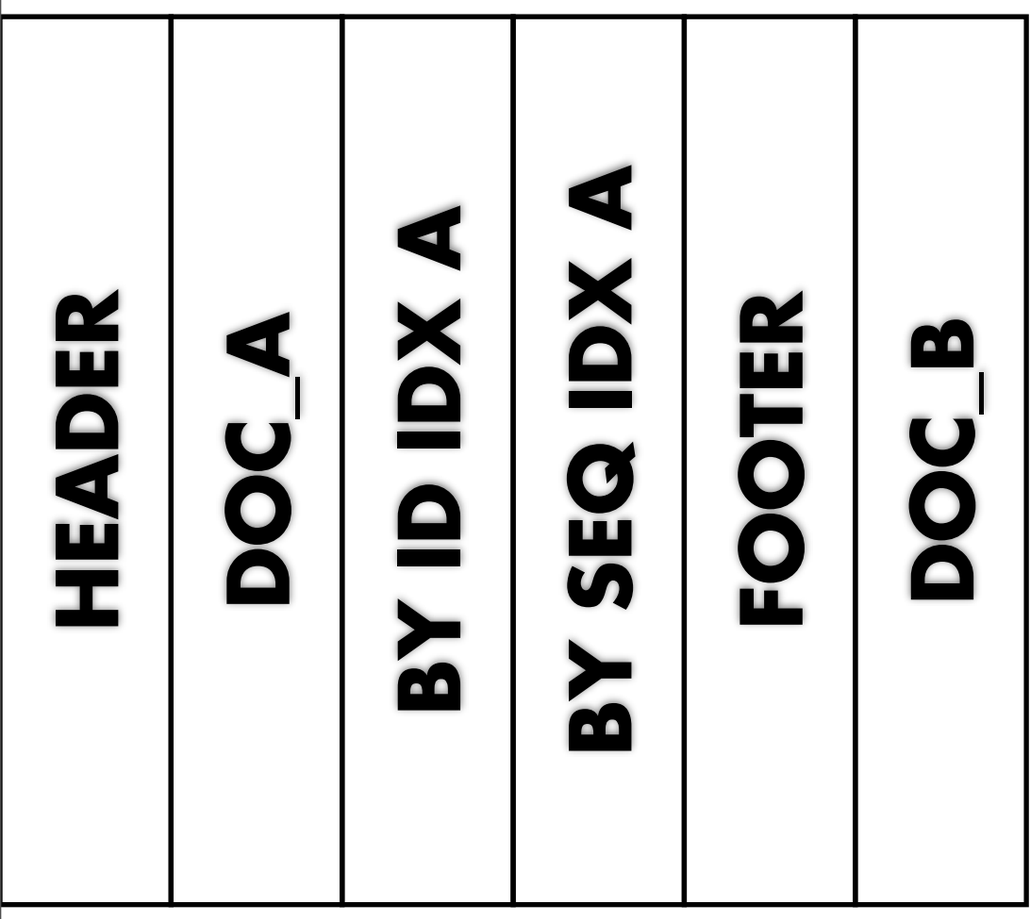


- 2 x b+tree & data interleaved
- append only, mvcc
- full fsync control

- Can answer:
- Data for $key
- What happened $since

- Used for core data storage
- As well as indexes

- Everything else is built on top

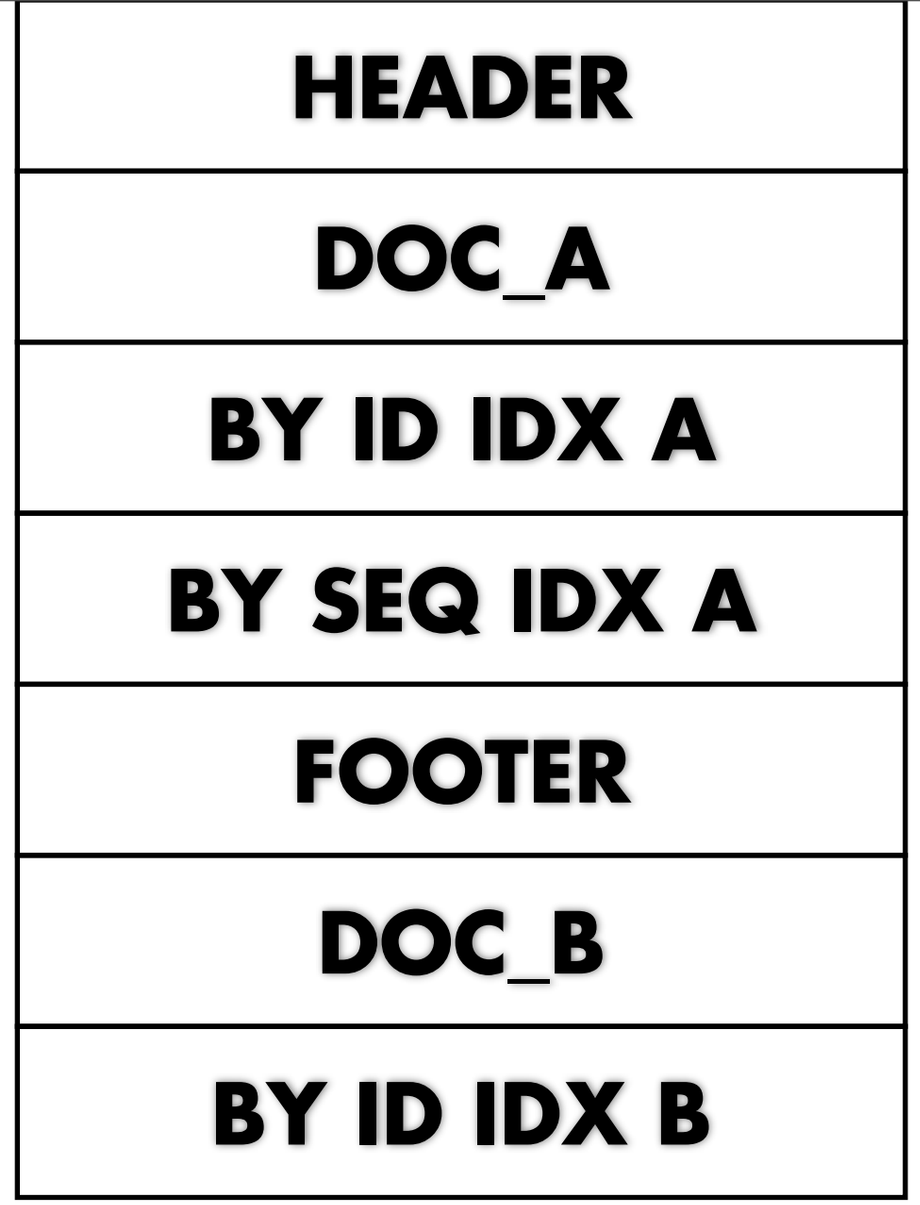


- 2 x b+tree & data interleaved
- append only, mvcc
- full fsync control

- Can answer:
  - Data for $key
  - What happened $since

- Used for core data storage
- As well as indexes

- Everything else is built on top



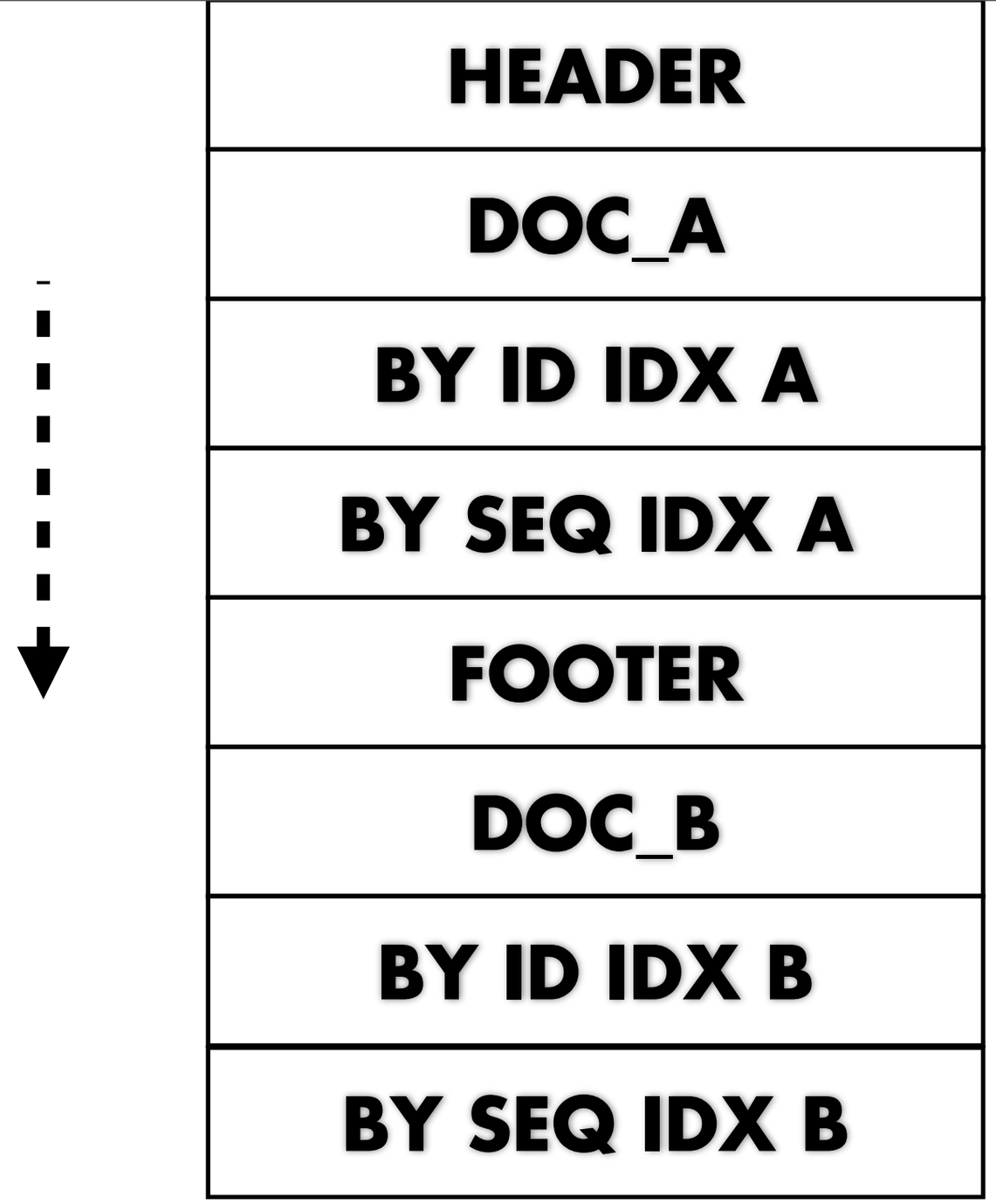
Monday, November 26, 12

- 2 x b+tree & data interleaved
- append only, mvcc
- full fsync control

- Can answer:
- Data for $key
- What happened $since

- Used for core data storage
- As well as indexes

- Everything else is built on top



- 2 x b+tree & data interleaved
- append only, mvcc
- full fsync control

- Can answer:
- Data for $key
- What happened $since

- Used for core data storage
- As well as indexes

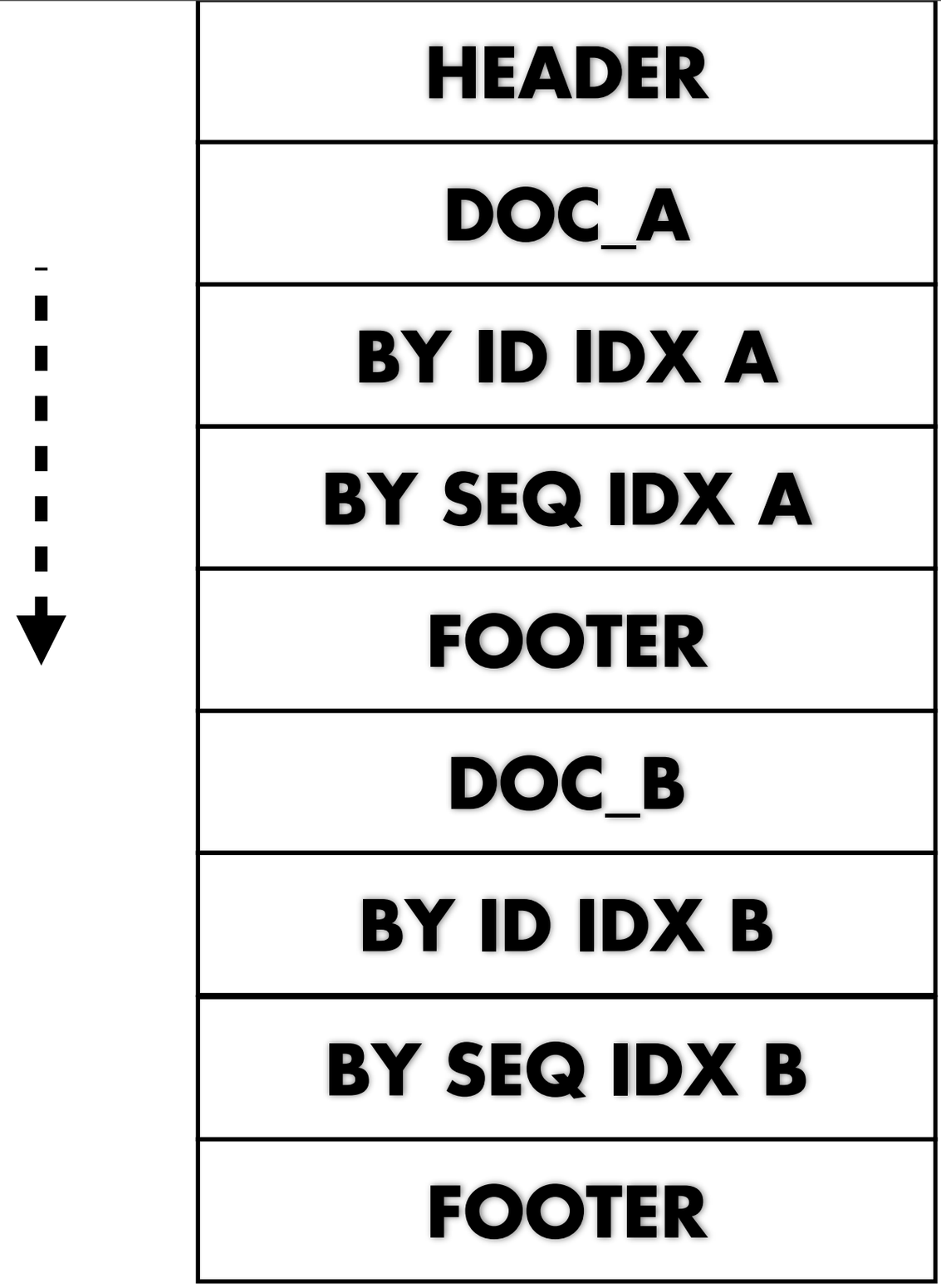
- Everything else is built on top

- 2 x b+tree & data interleaved
- append only, mvcc
- full fsync control

- Can answer:
- Data for $key
- What happened $since

- Used for core data storage
- As well as indexes

- Everything else is built on top



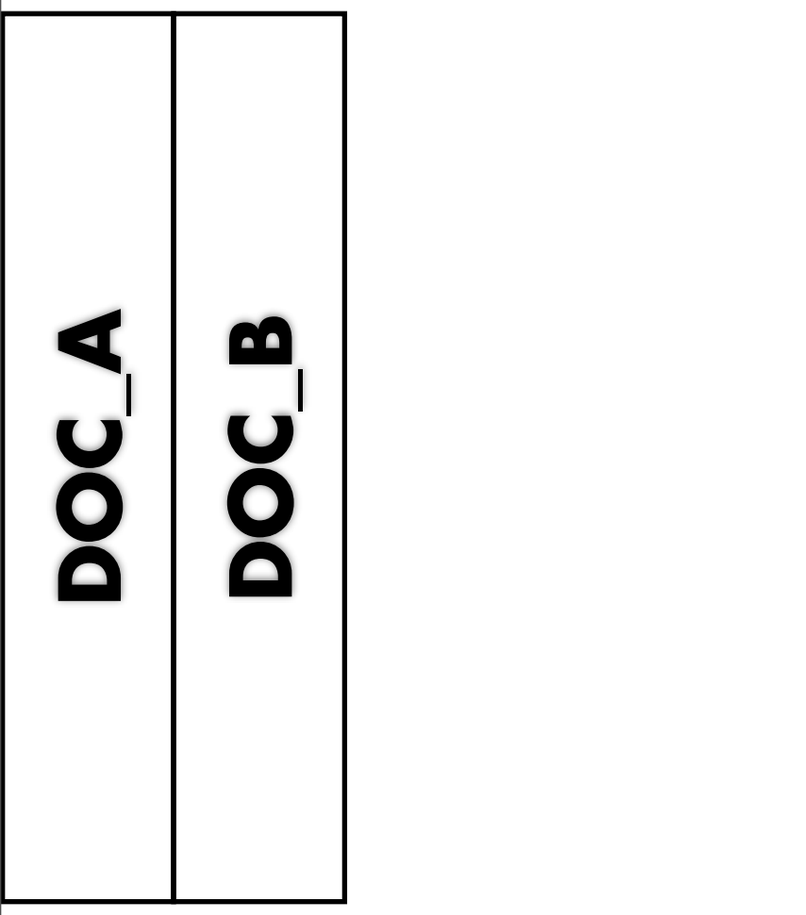
Monday, November 26, 12

Bulk add + Delete

DOC_A

Monday, November 26, 12

Bulk add + Delete

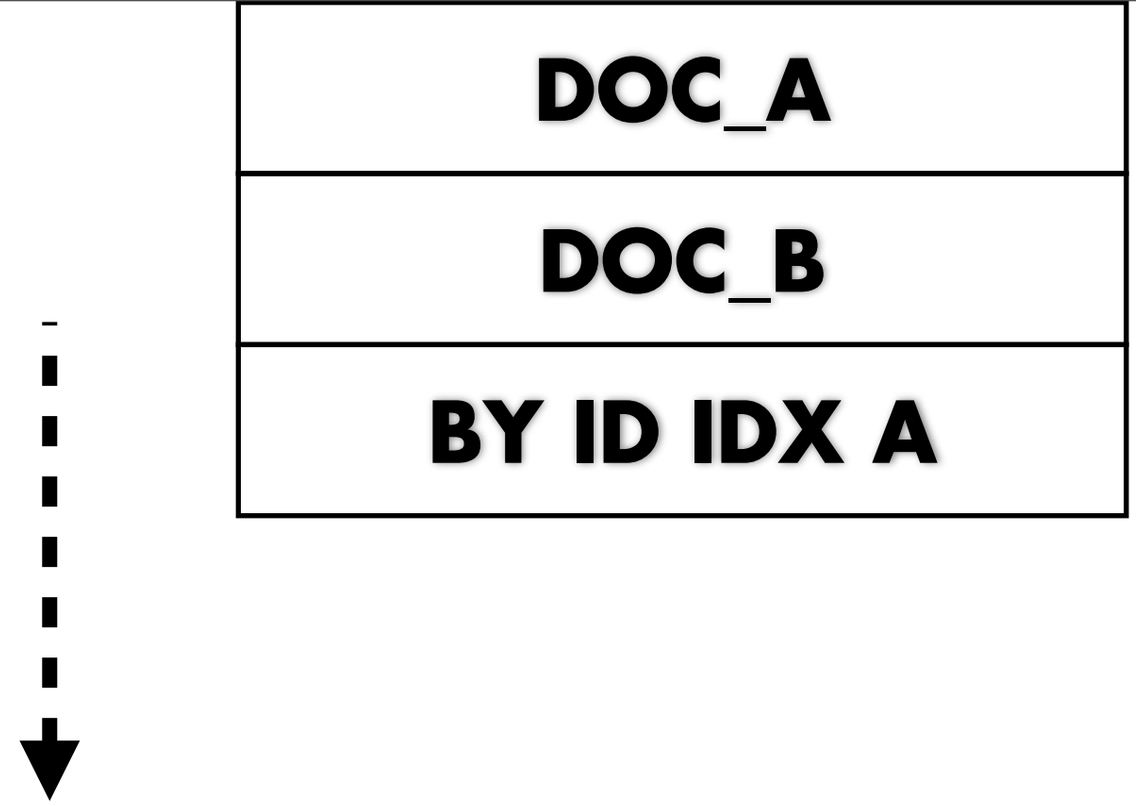


Monday, November 26, 12

Bulk add + Delete

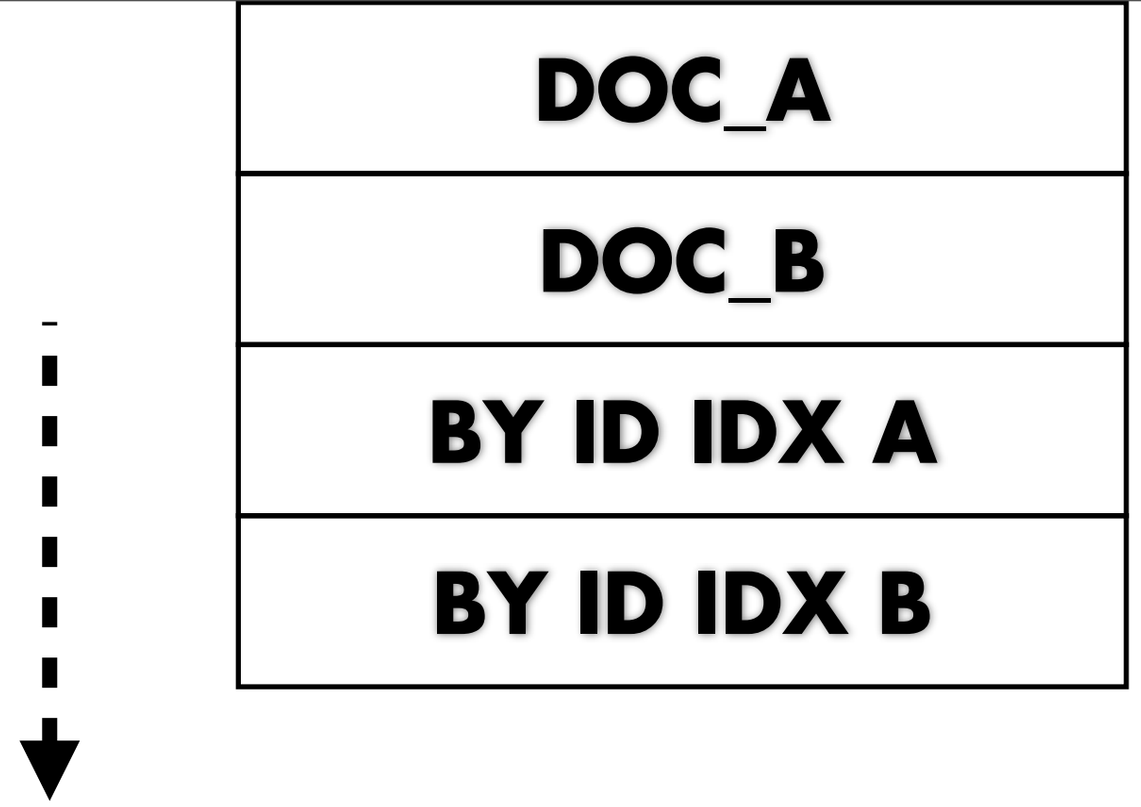
Monday, November 26, 12

Bulk add + Delete

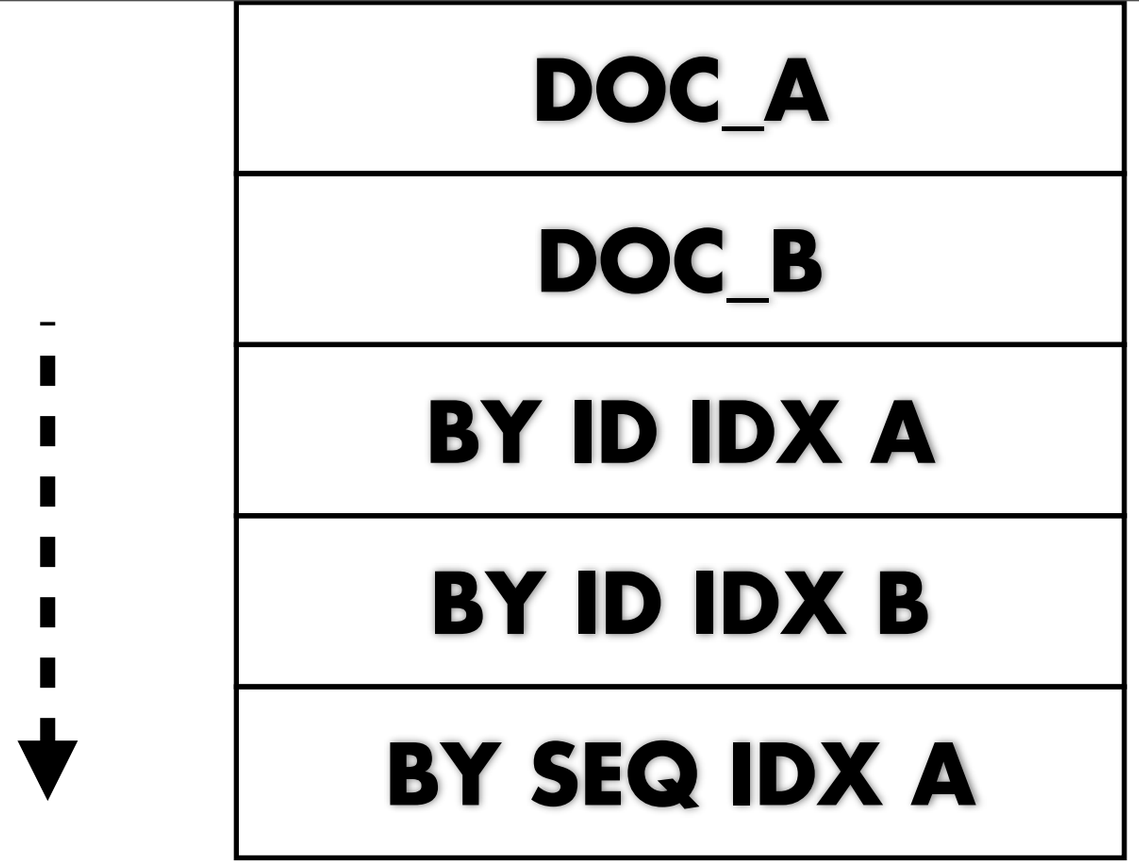


Monday, November 26, 12

Bulk add + Delete

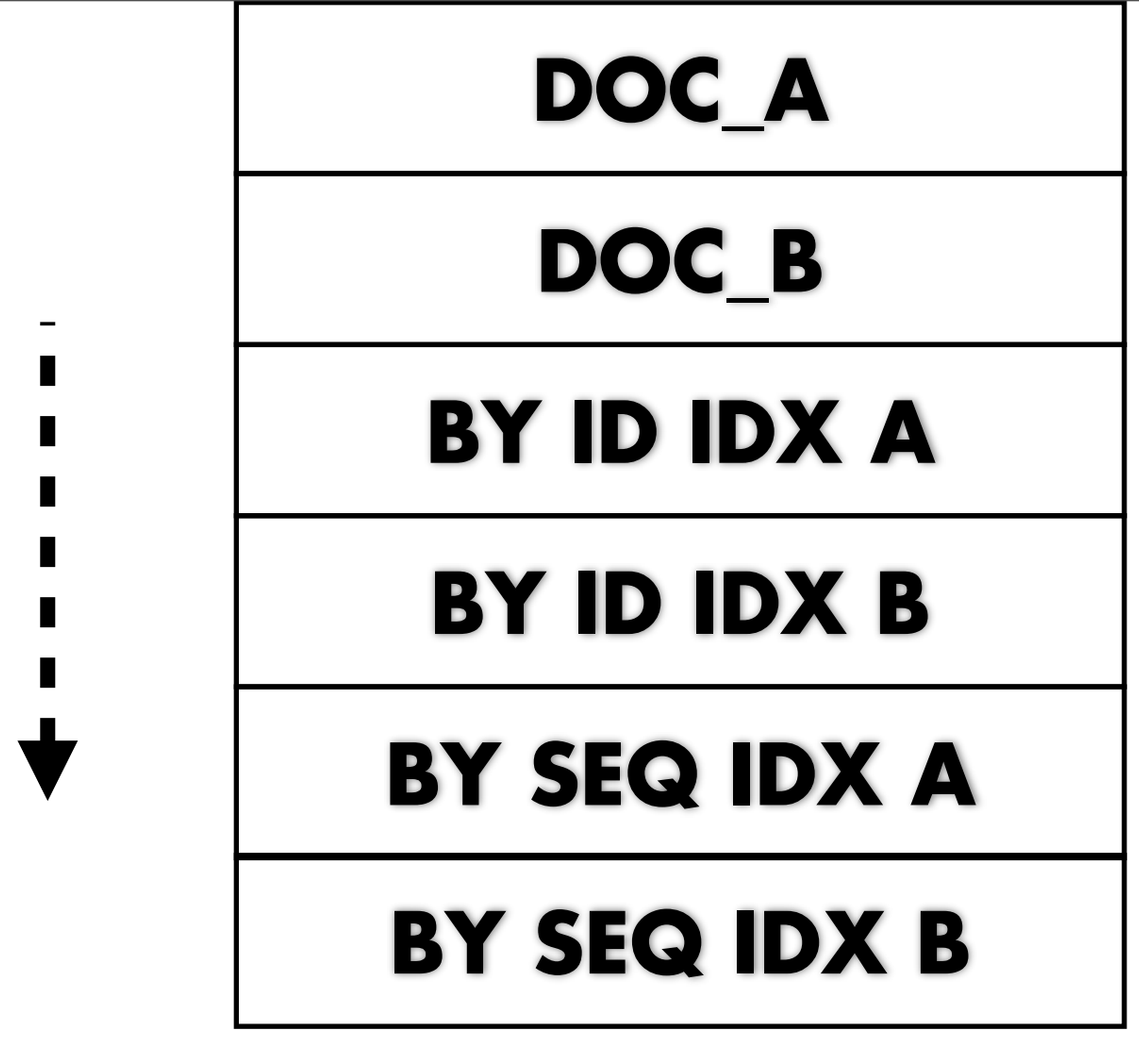


Bulk add + Delete



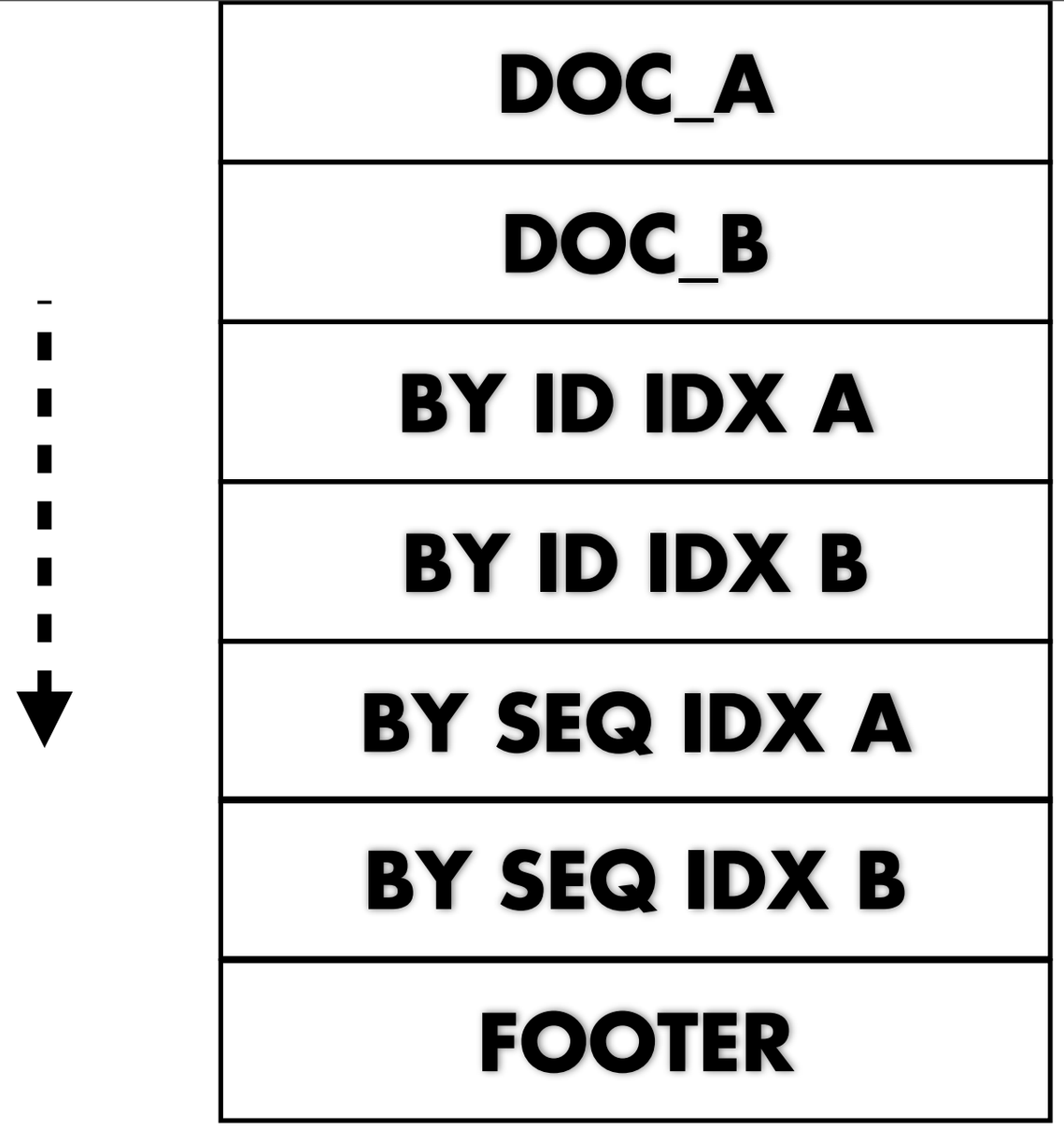
Monday, November 26, 12

Bulk add + Delete

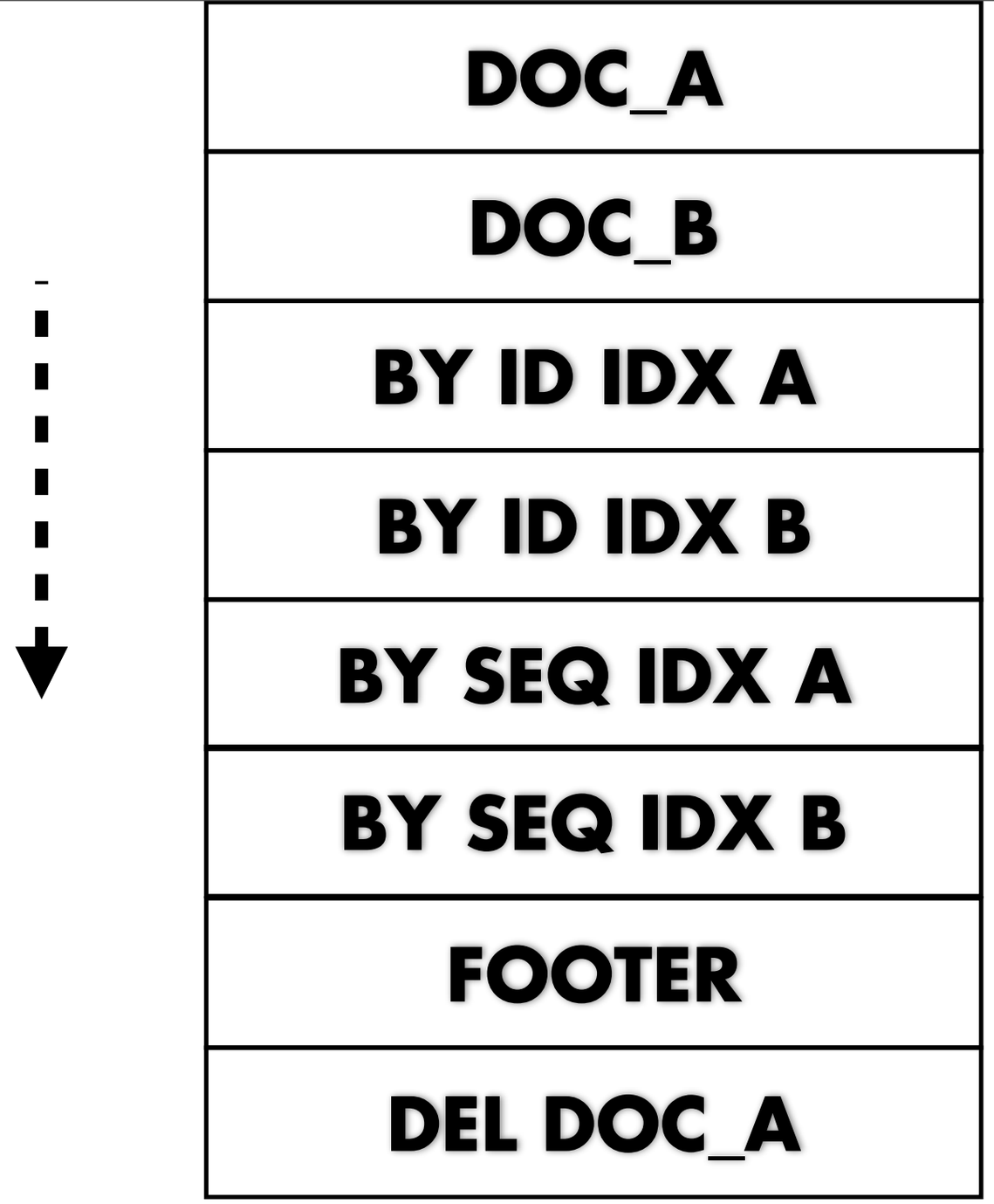


Monday, November 26, 12

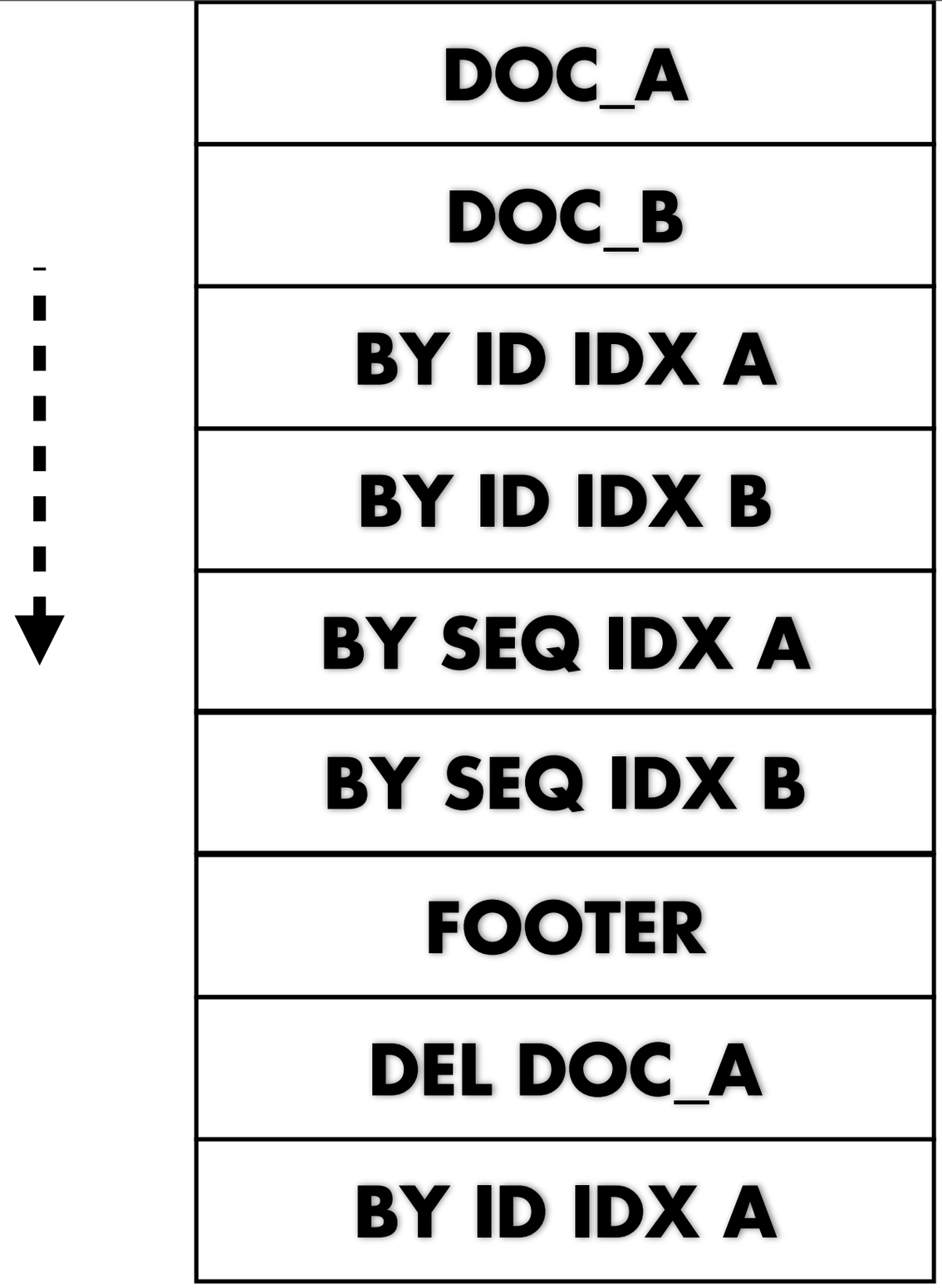
Bulk add + Delete



Bulk add + Delete



Bulk add + Delete

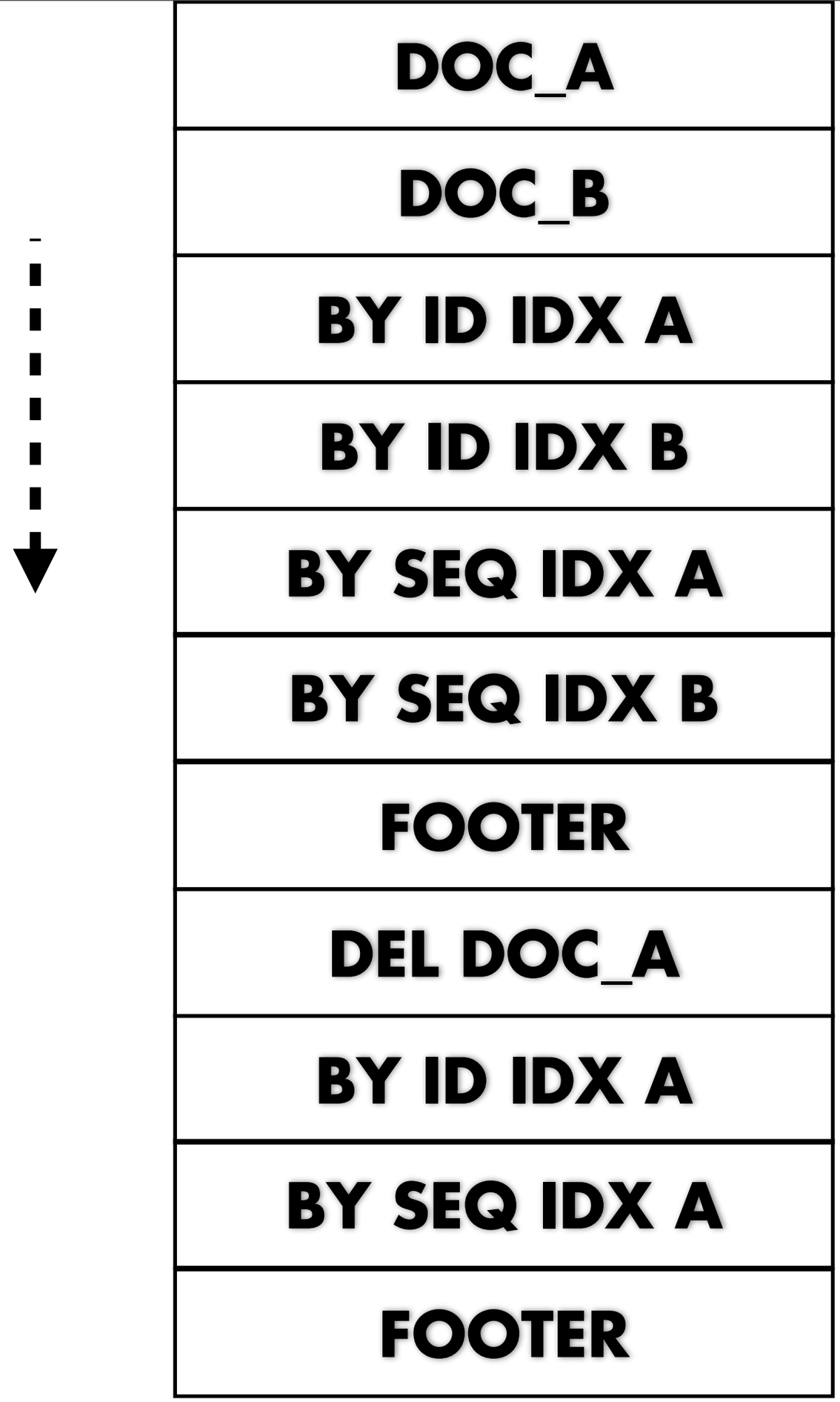


Monday, November 26, 12

Bulk add + Delete

DOC_A
DOC_B
BY ID IDX A
BY ID IDX B
BY SEQ IDX A
BY SEQ IDX B
FOOTER
DEL DOC_A
BY ID IDX A
BY SEQ IDX A

Bulk add + Delete



Operational Consequences

- efficient on spinning disk, “tape”
- btree = wide, upper layers in disk cache
- backup with cp $a $b

- compaction hurts

Core Features

(using by-seq)

- Replication
- Indexing / Views / GeoCouch / Lucene / ES etc.
- /_changes
- Compaction

Monday, November 26, 12

Replication

DATABASE A

Replication

1
DATABASE A

Replication

Monday, November 26, 12

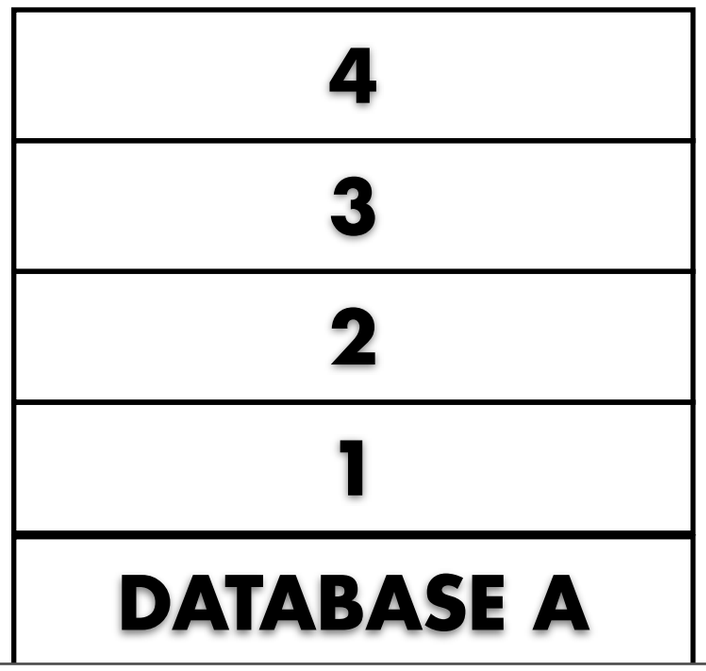
Replication

Monday, November 26, 12

Replication

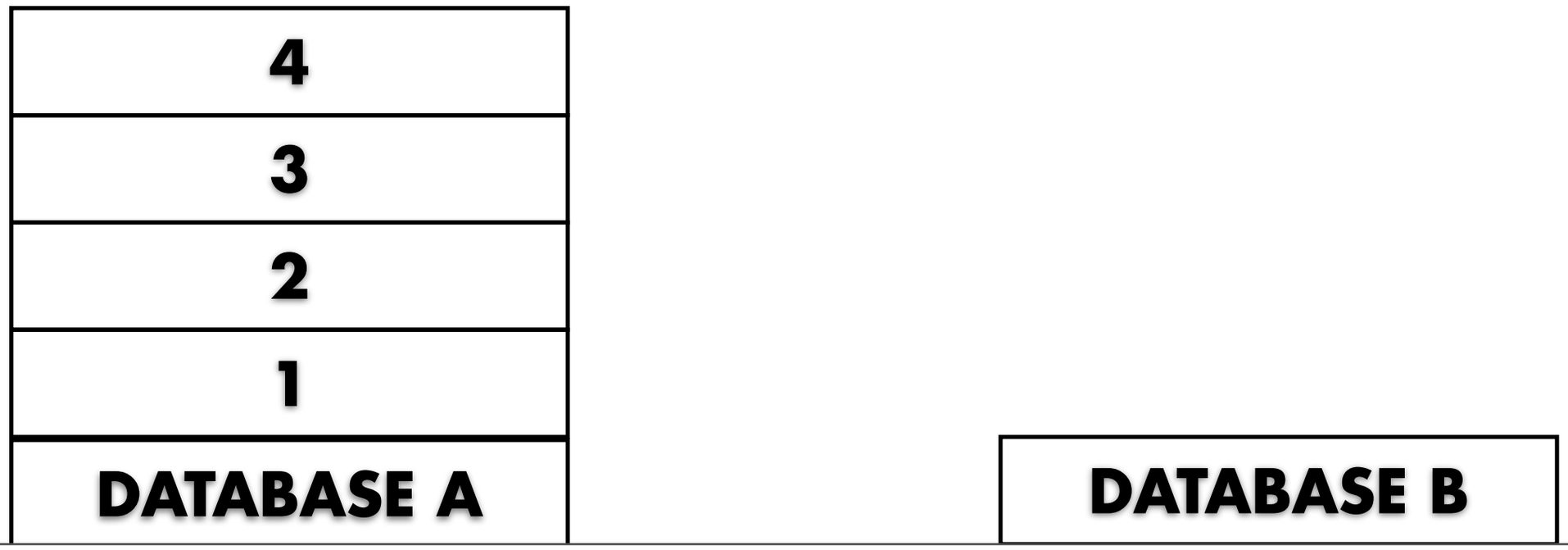
Monday, November 26, 12

Replication



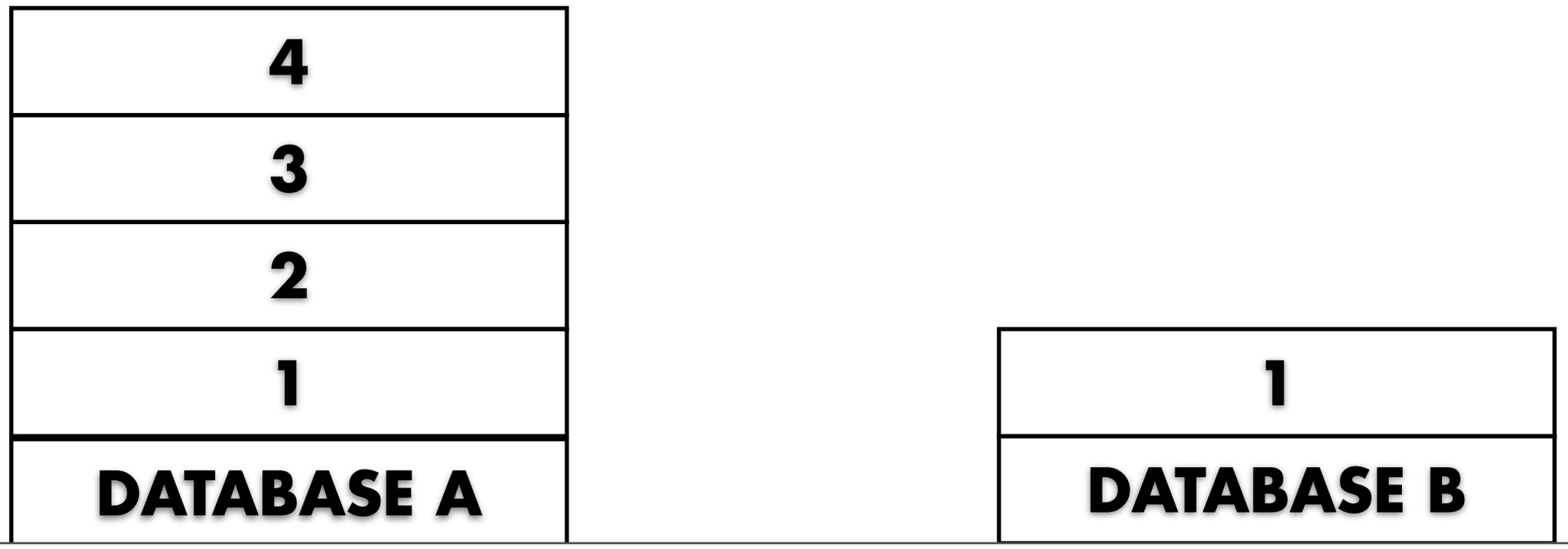
Monday, November 26, 12

Replication



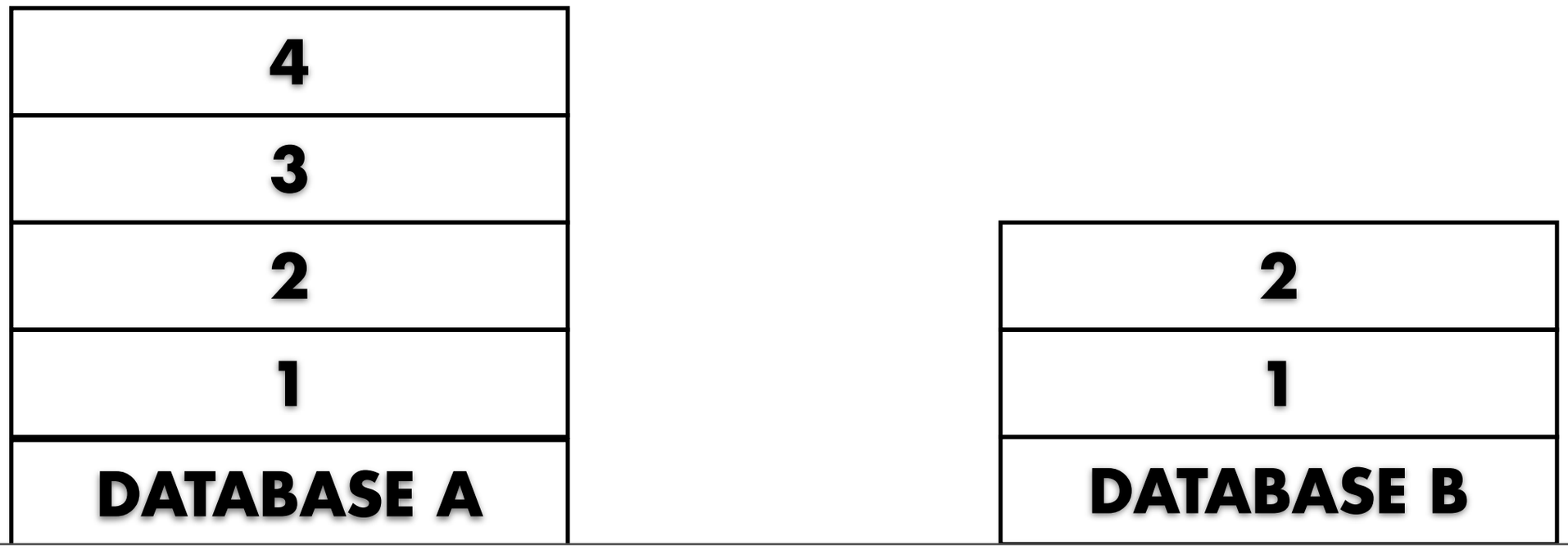
Monday, November 26, 12

Replication



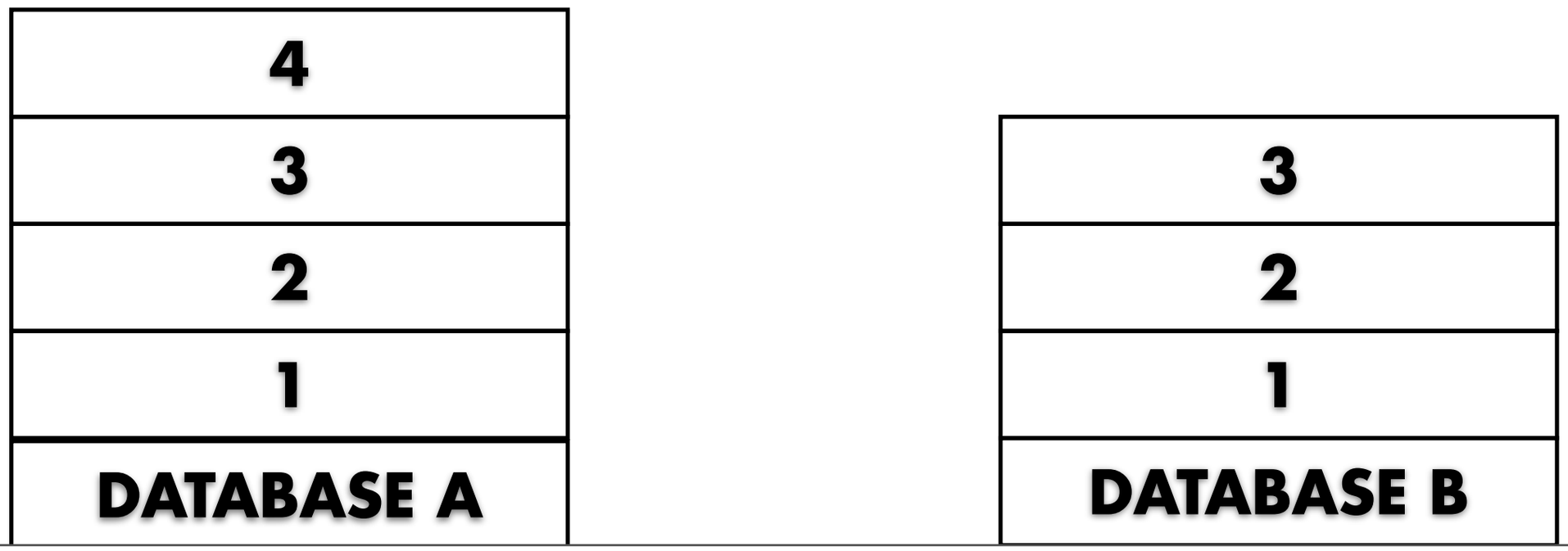
Monday, November 26, 12

Replication



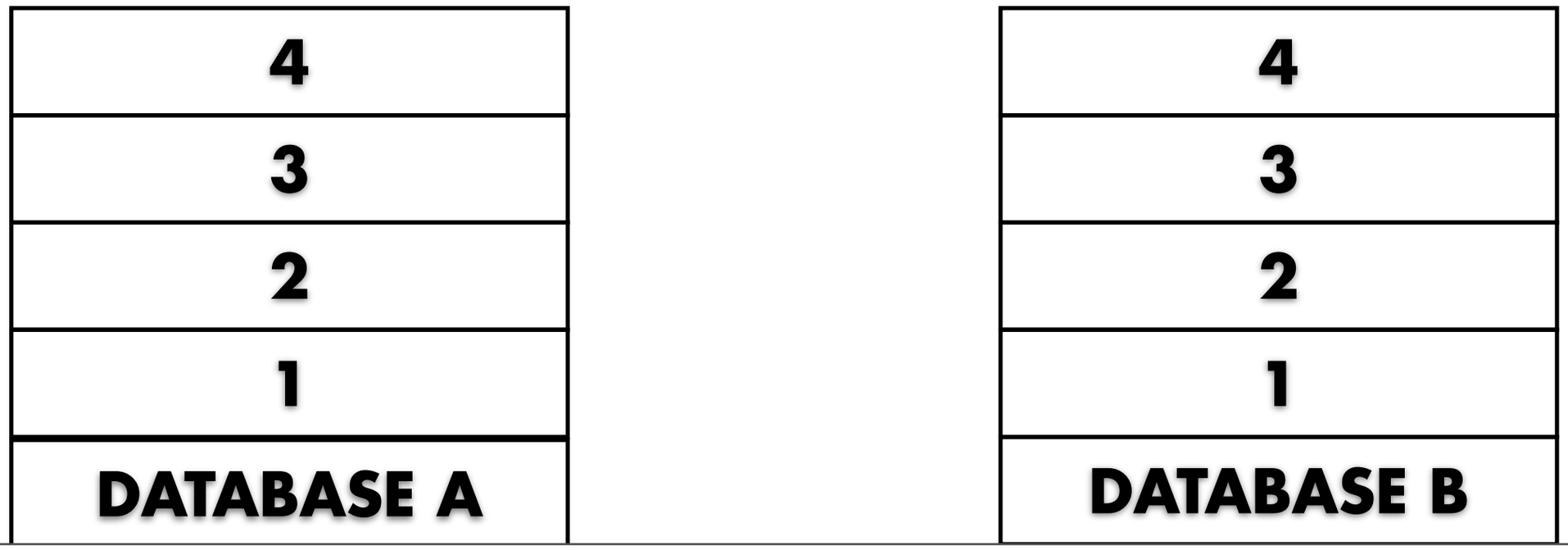
Monday, November 26, 12

Replication



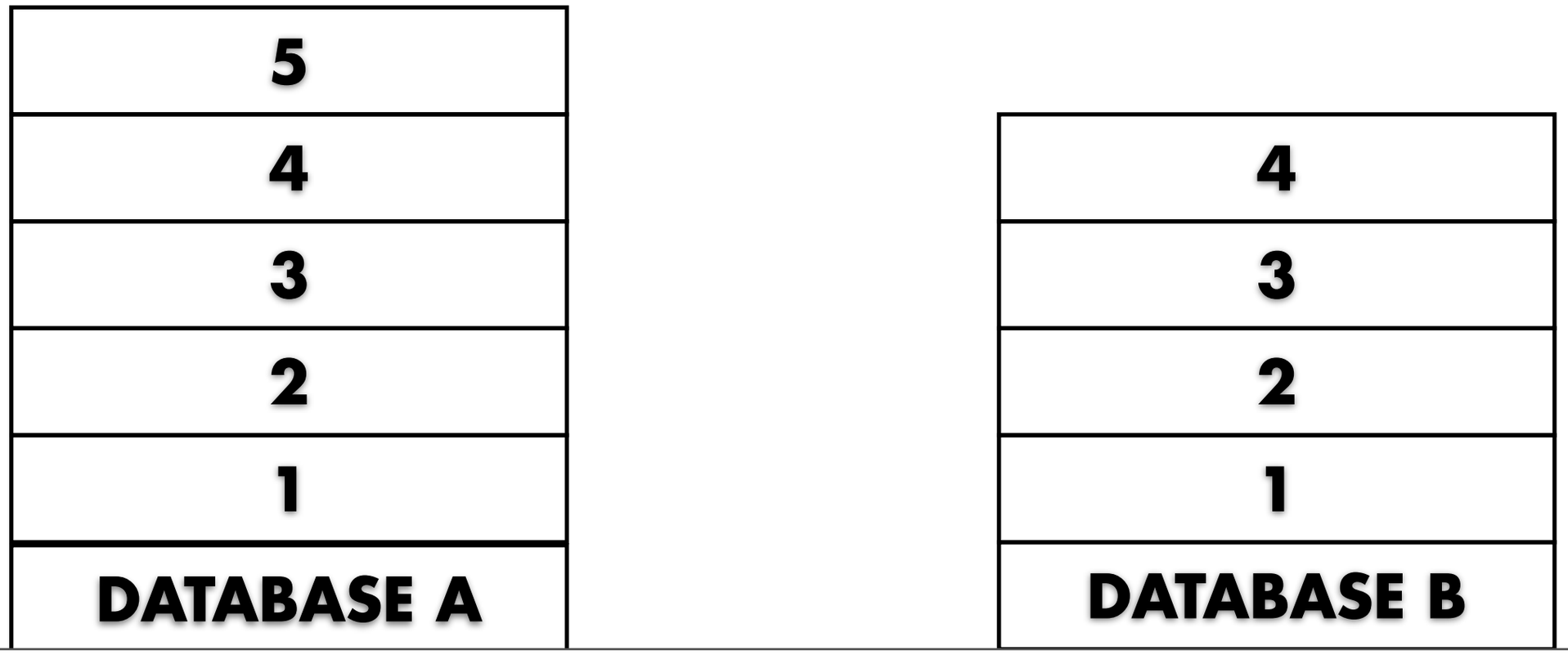
Monday, November 26, 12

Replication



Monday, November 26, 12

Replication

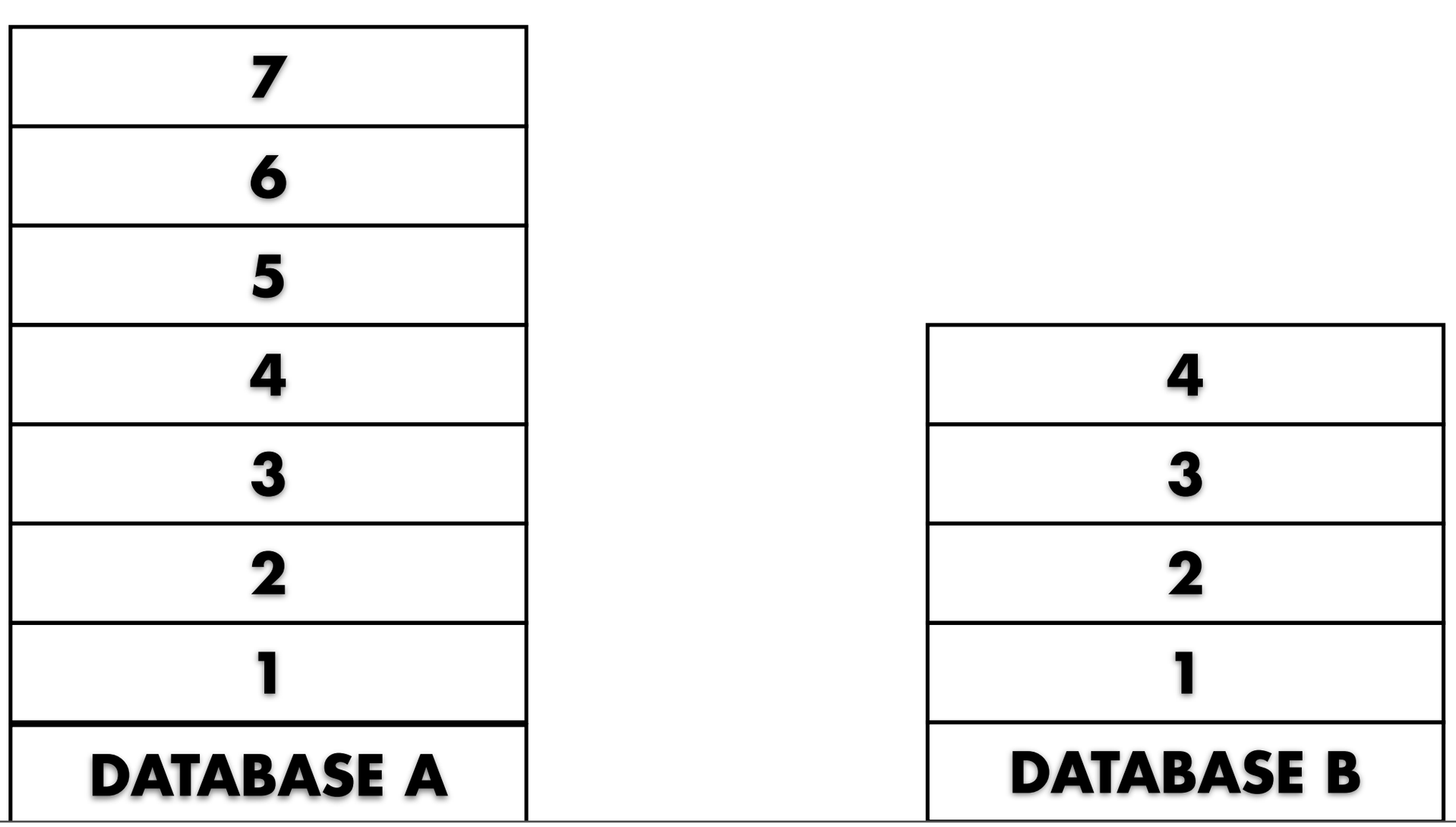


6
5
4
3
2
1
DATABASE A

4
3
2
1
DATABASE B

Monday, November 26, 12

Replication

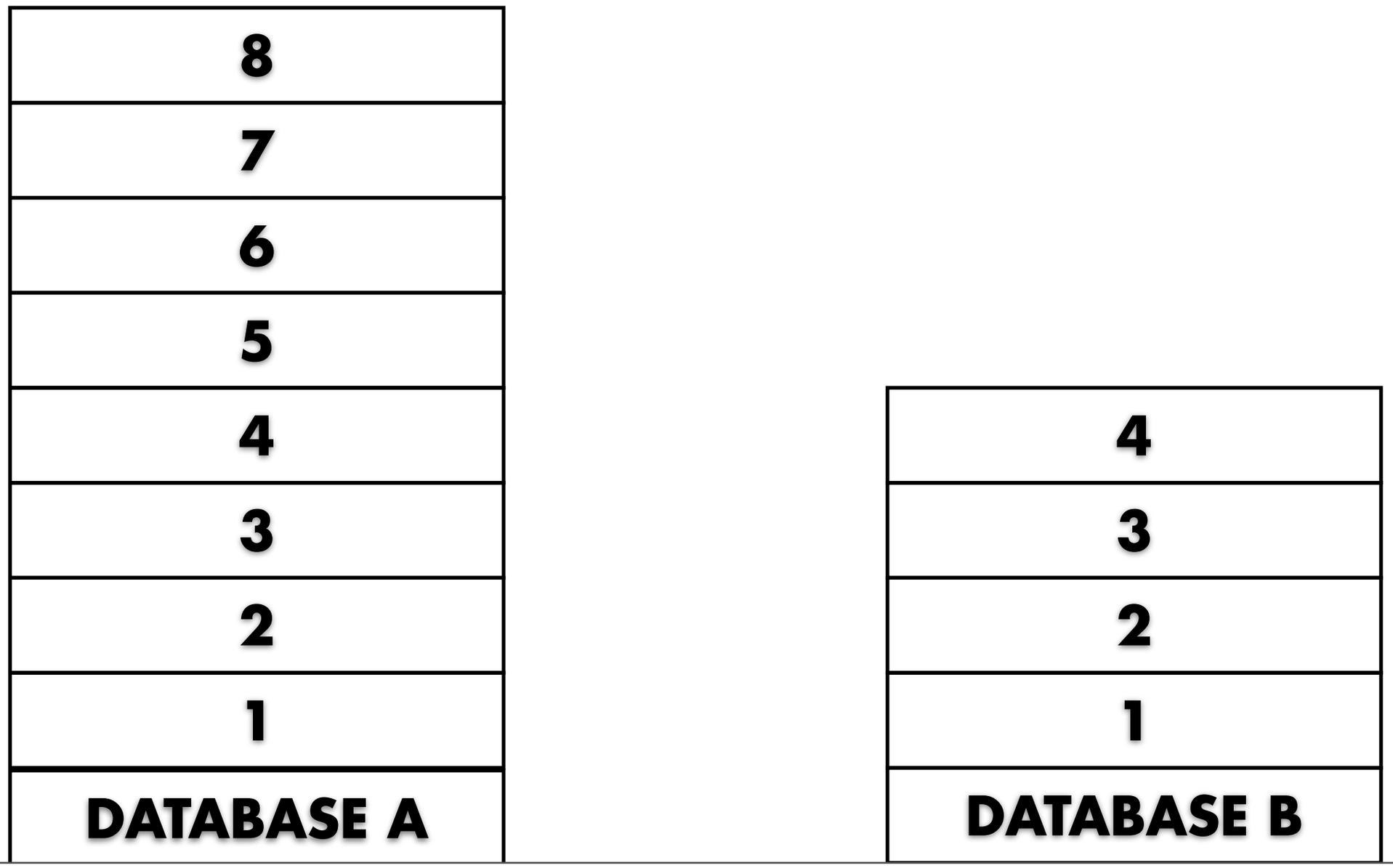


Monday, November 26, 12

Replication

Monday, November 26, 12

Replication



8
7
6
5
4
3
2
1
DATABASE A

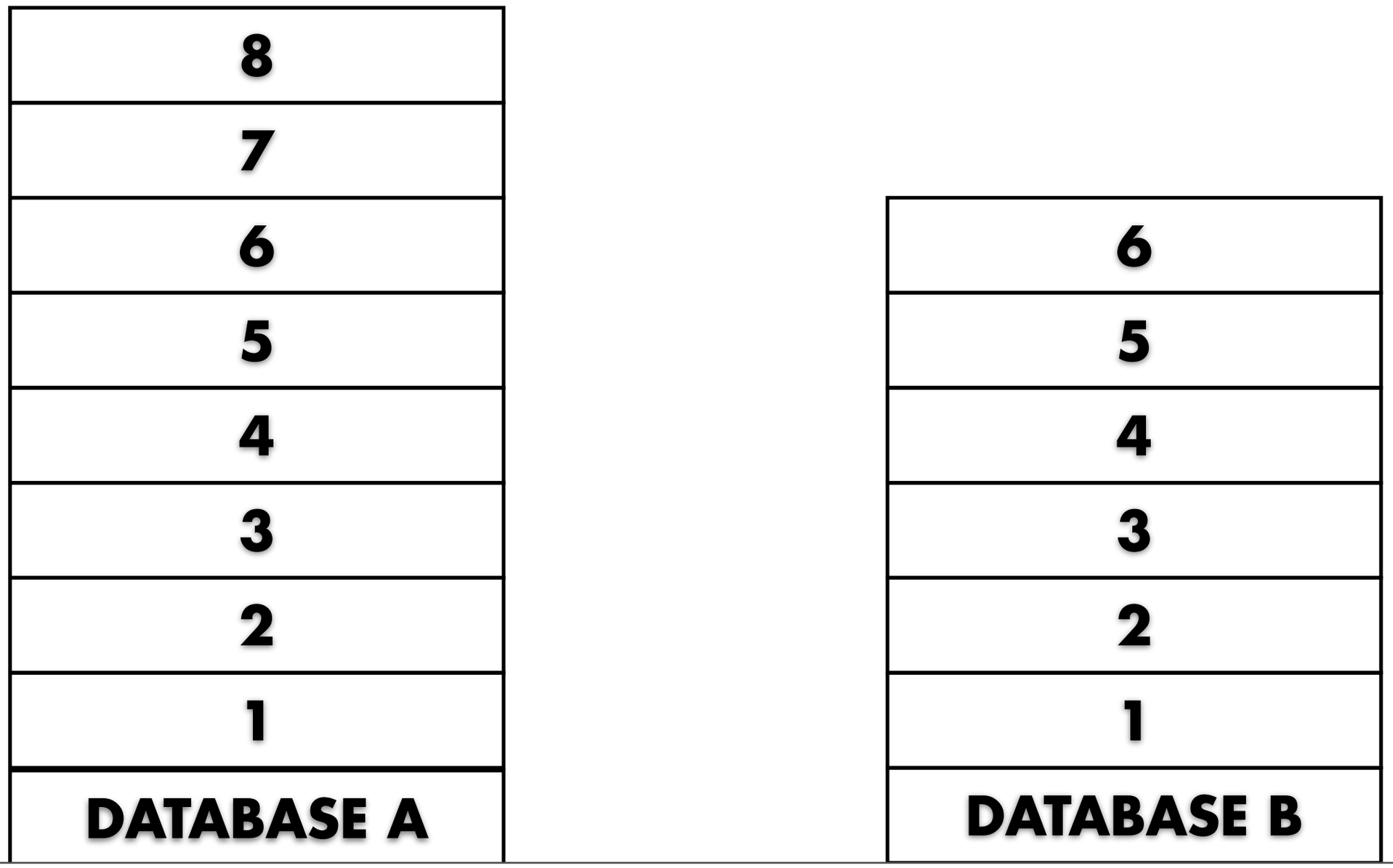
5
4
3
2
1
DATABASE B

Monday, November 26, 12

Replication

Monday, November 26, 12

Replication

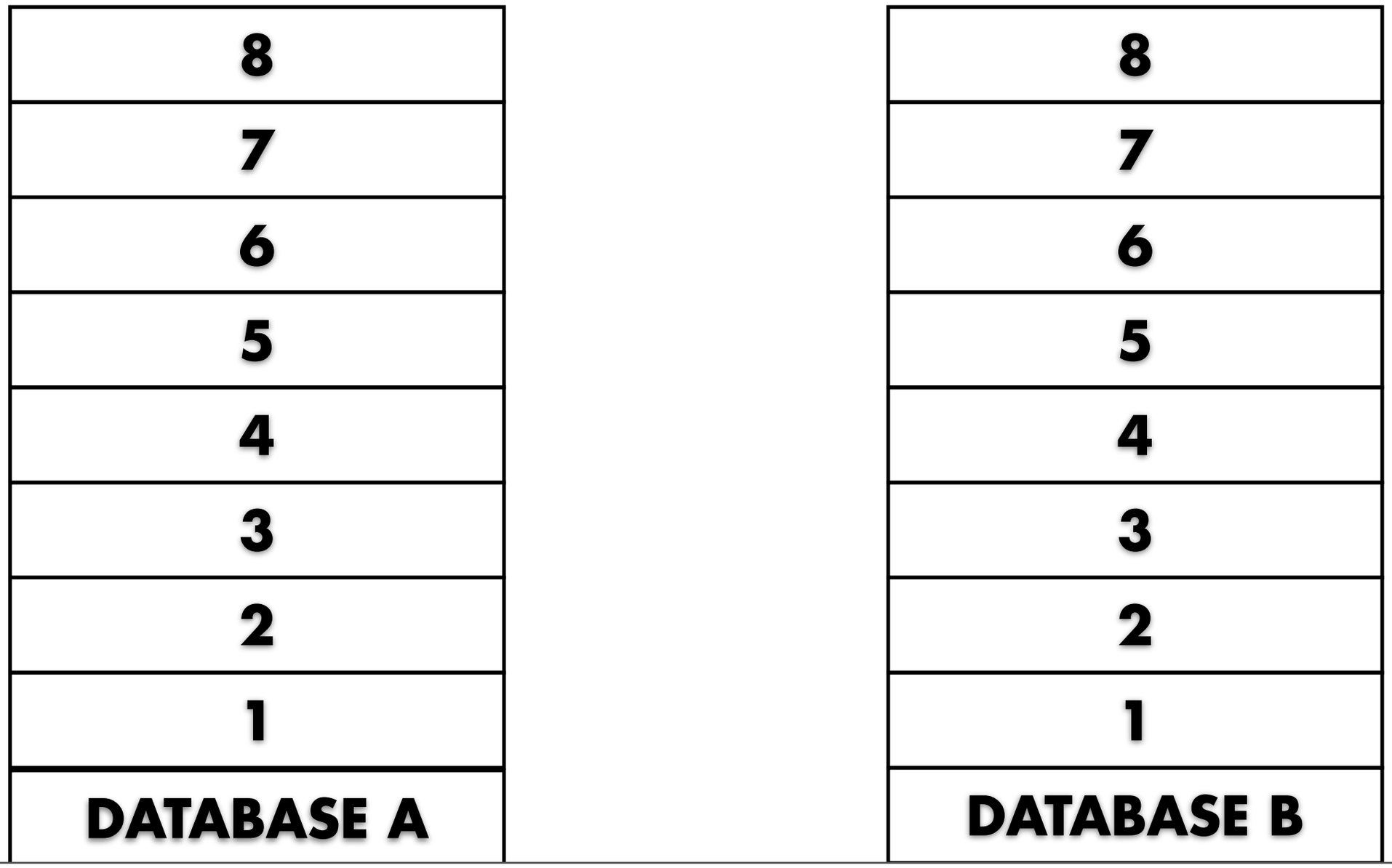


8
7
6
5
4
3
2
1
DATABASE A

7
6
5
4
3
2
1
DATABASE B

Monday, November 26, 12

Replication



Monday, November 26, 12

Replication

Monday, November 26, 12

Indexing

DATABASE A

Monday, November 26, 12

Indexing

1
DATABASE A

Monday, November 26, 12

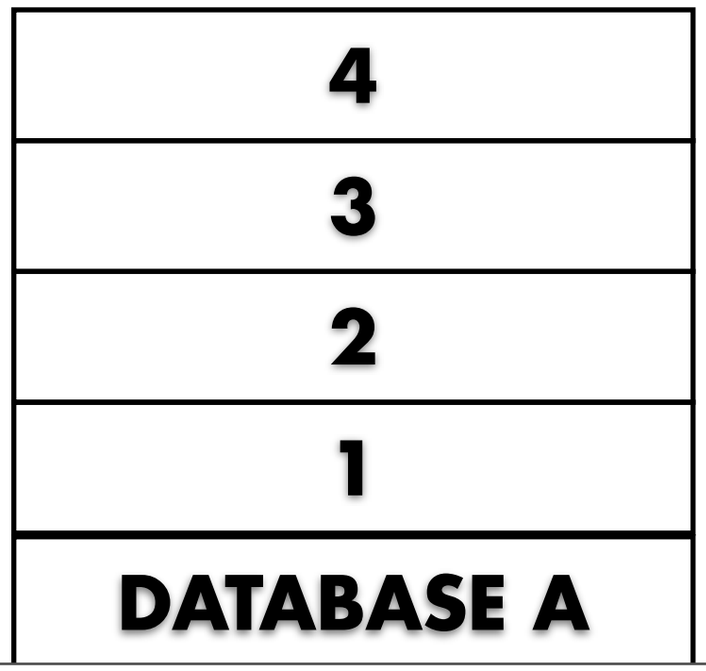
Indexing

Monday, November 26, 12

Indexing

Monday, November 26, 12

Indexing

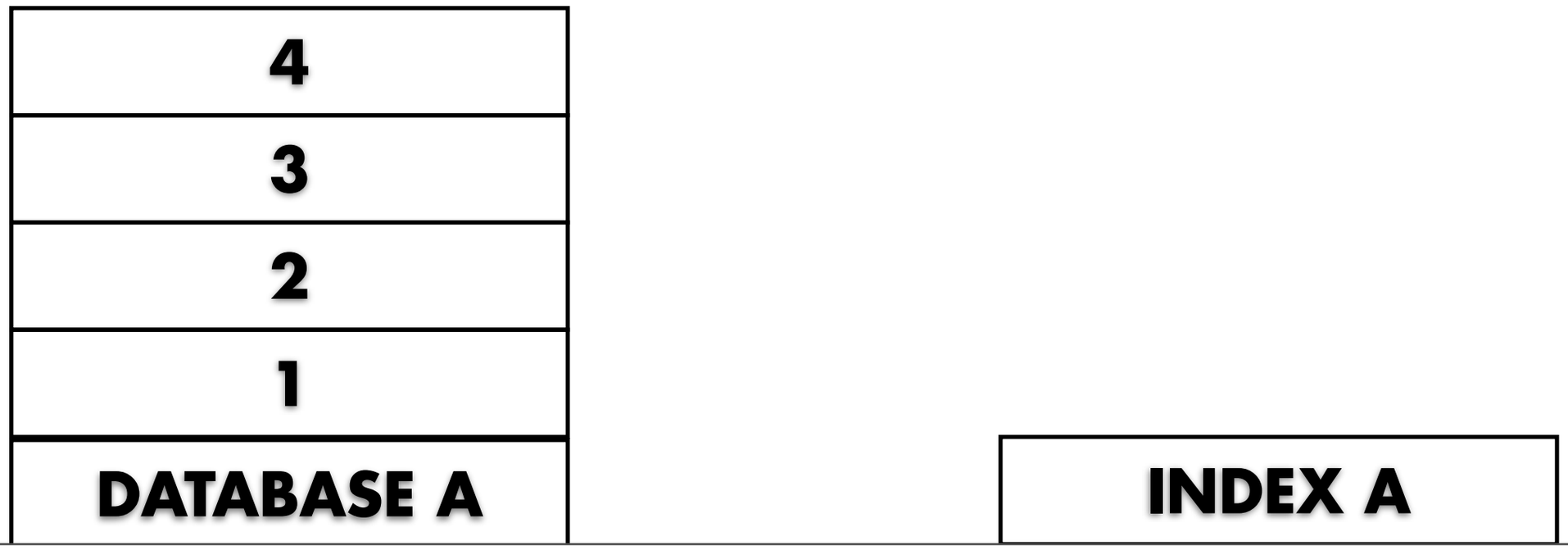


Monday, November 26, 12

Indexing

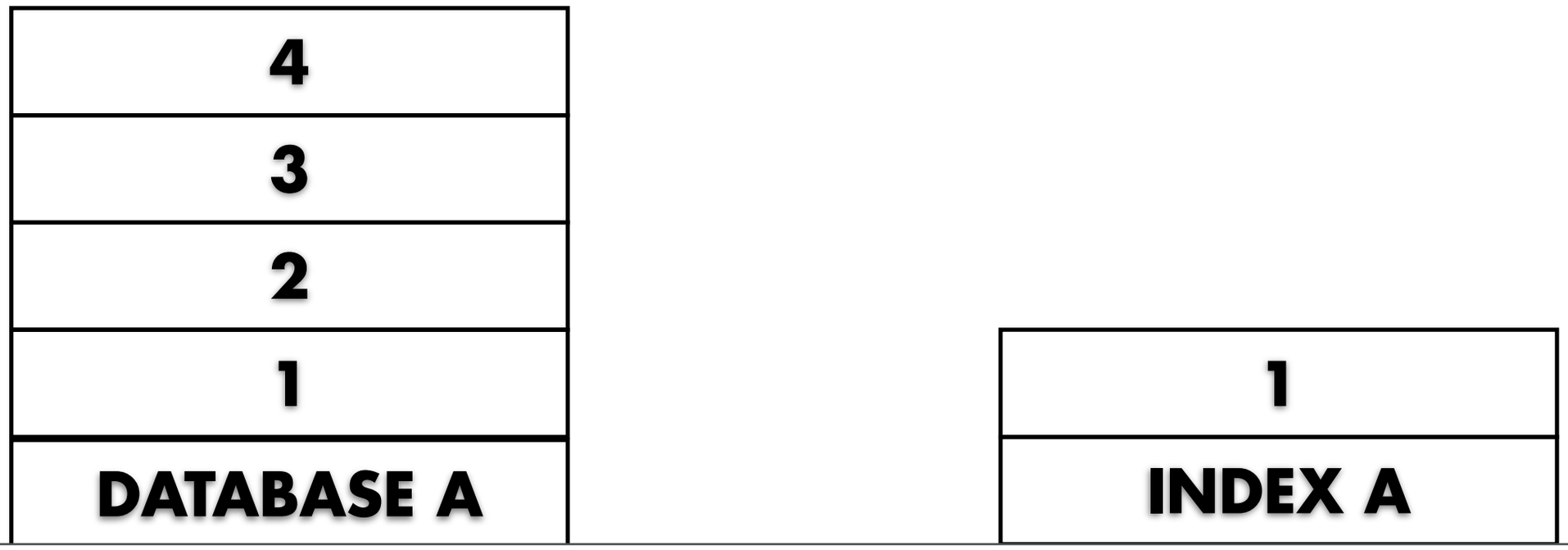
Monday, November 26, 12

Indexing



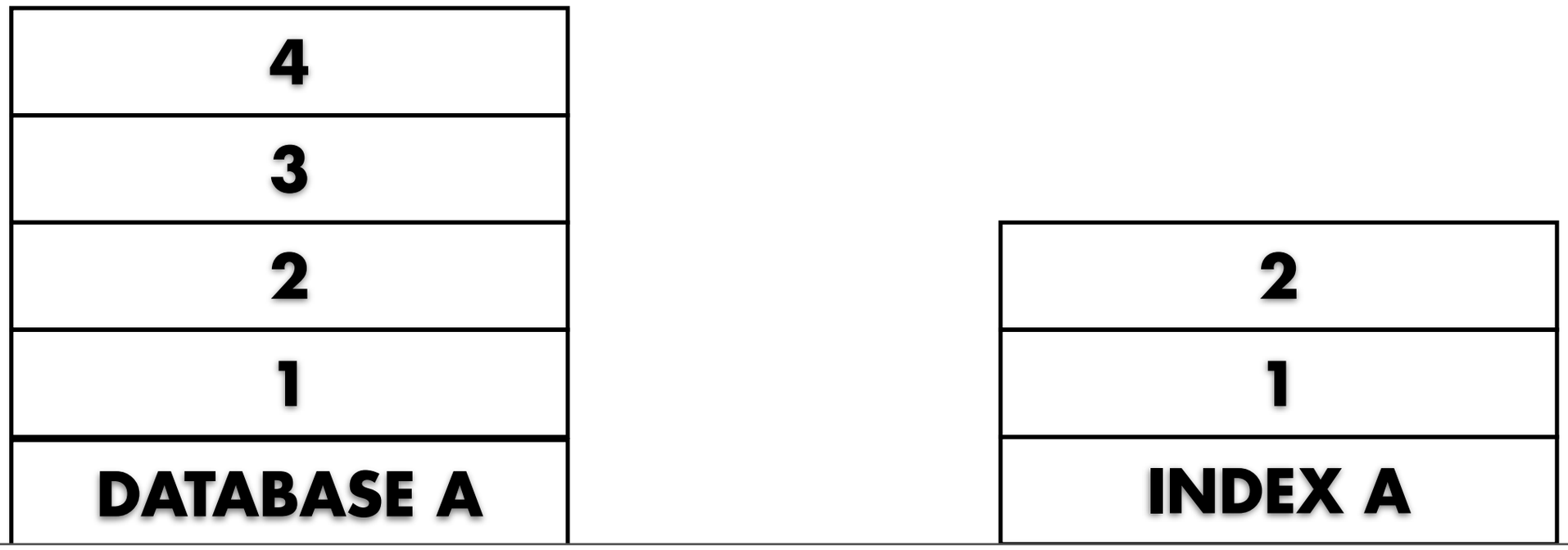
Monday, November 26, 12

Indexing



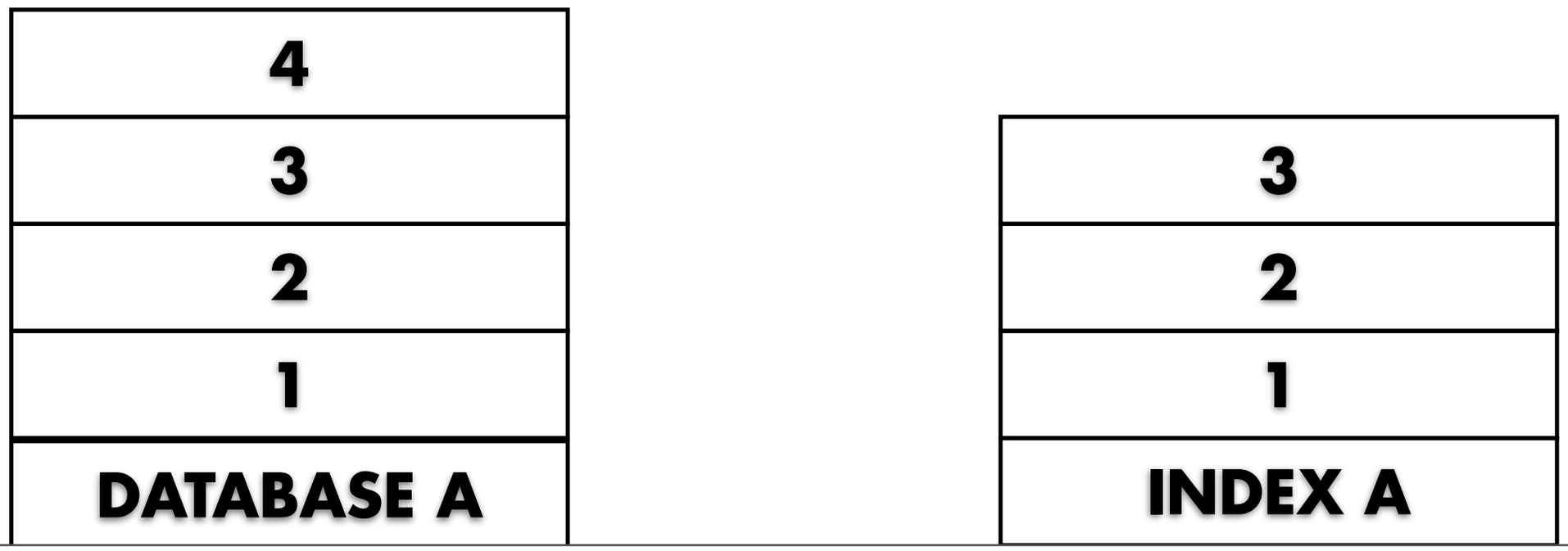
Monday, November 26, 12

Indexing



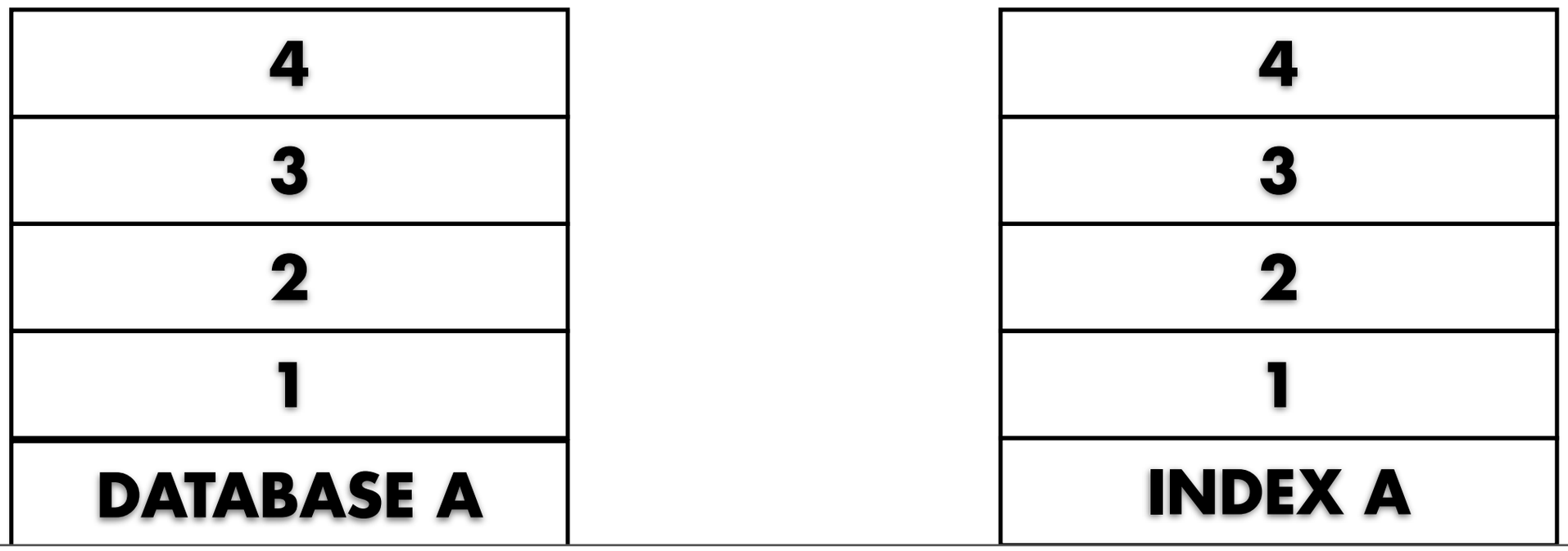
Monday, November 26, 12

Indexing



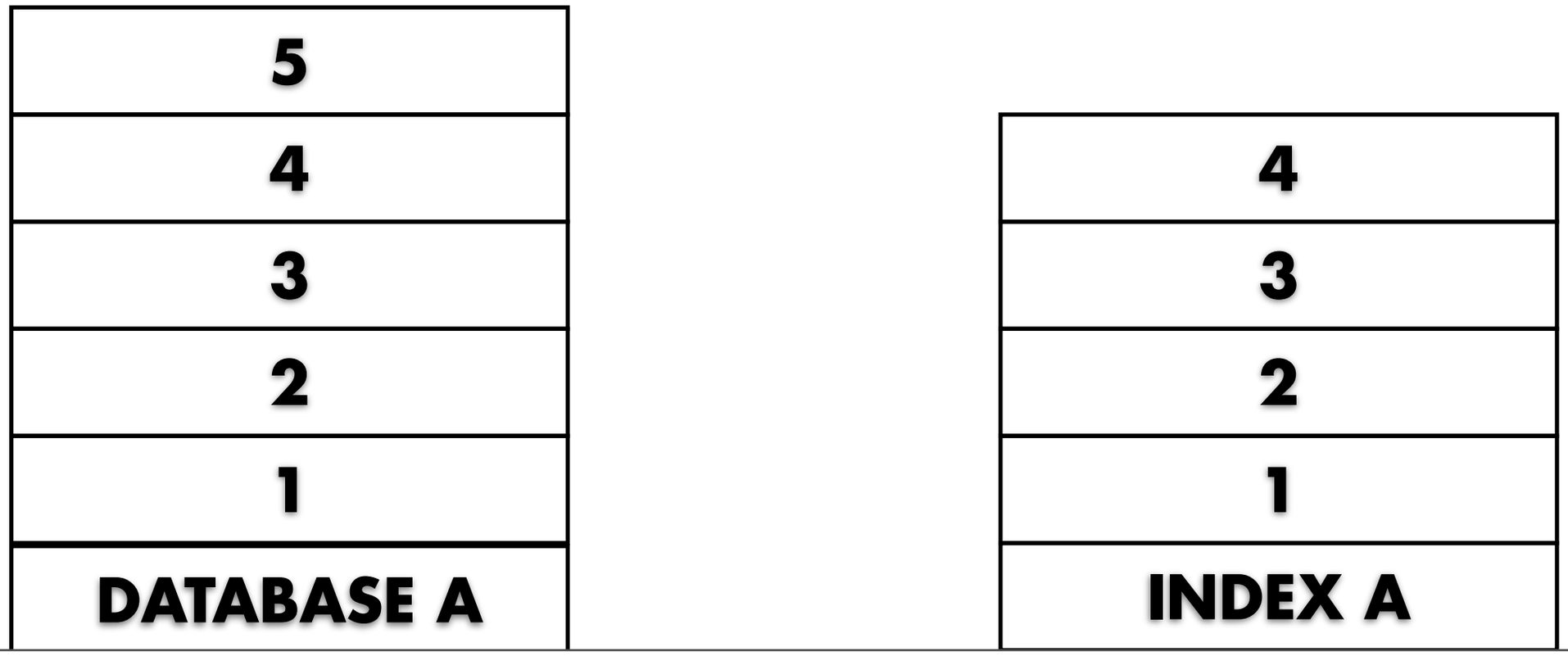
Monday, November 26, 12

Indexing



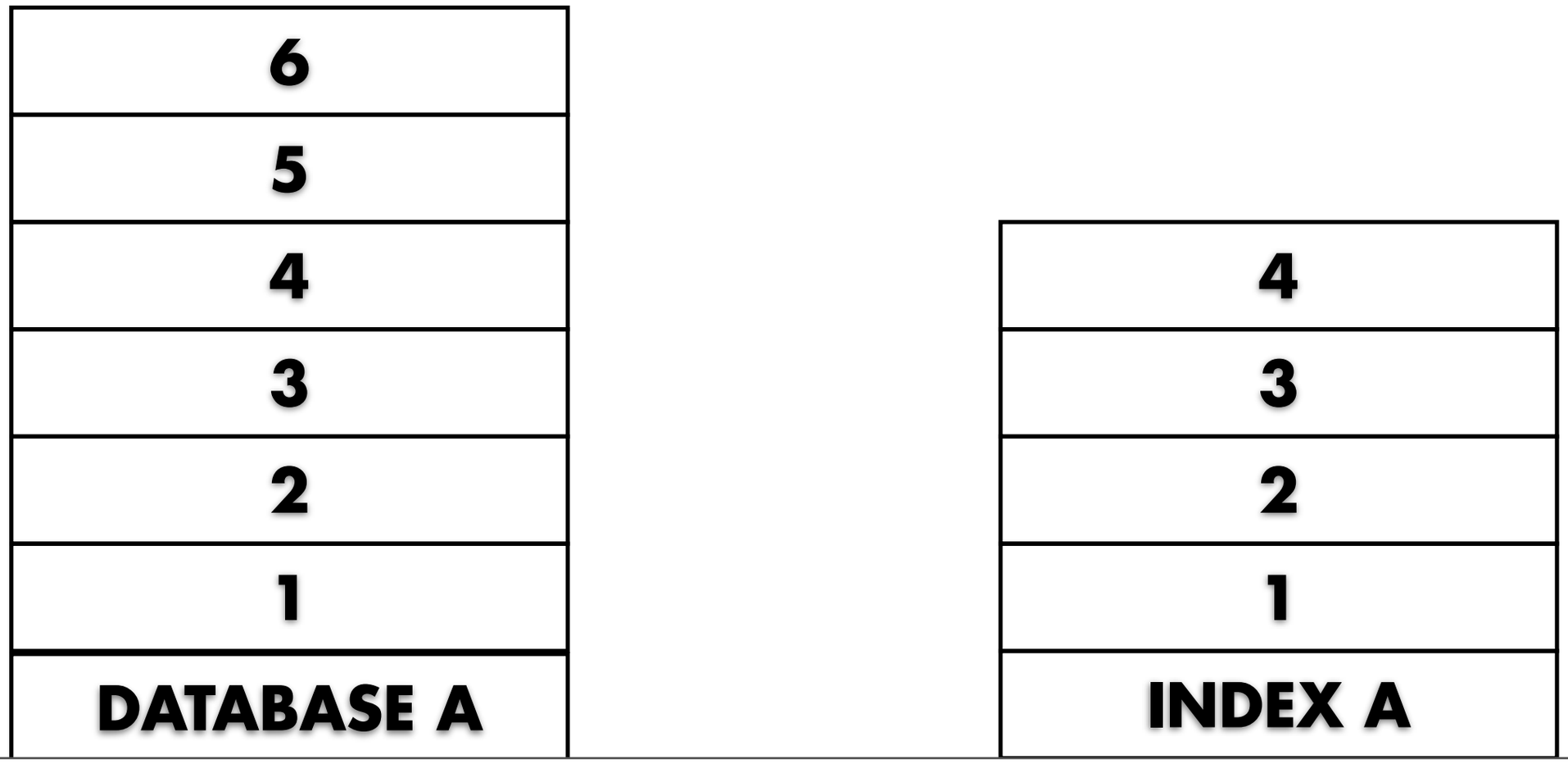
Monday, November 26, 12

Indexing



Monday, November 26, 12

Indexing



7
6
5
4
3
2
1
DATABASE A

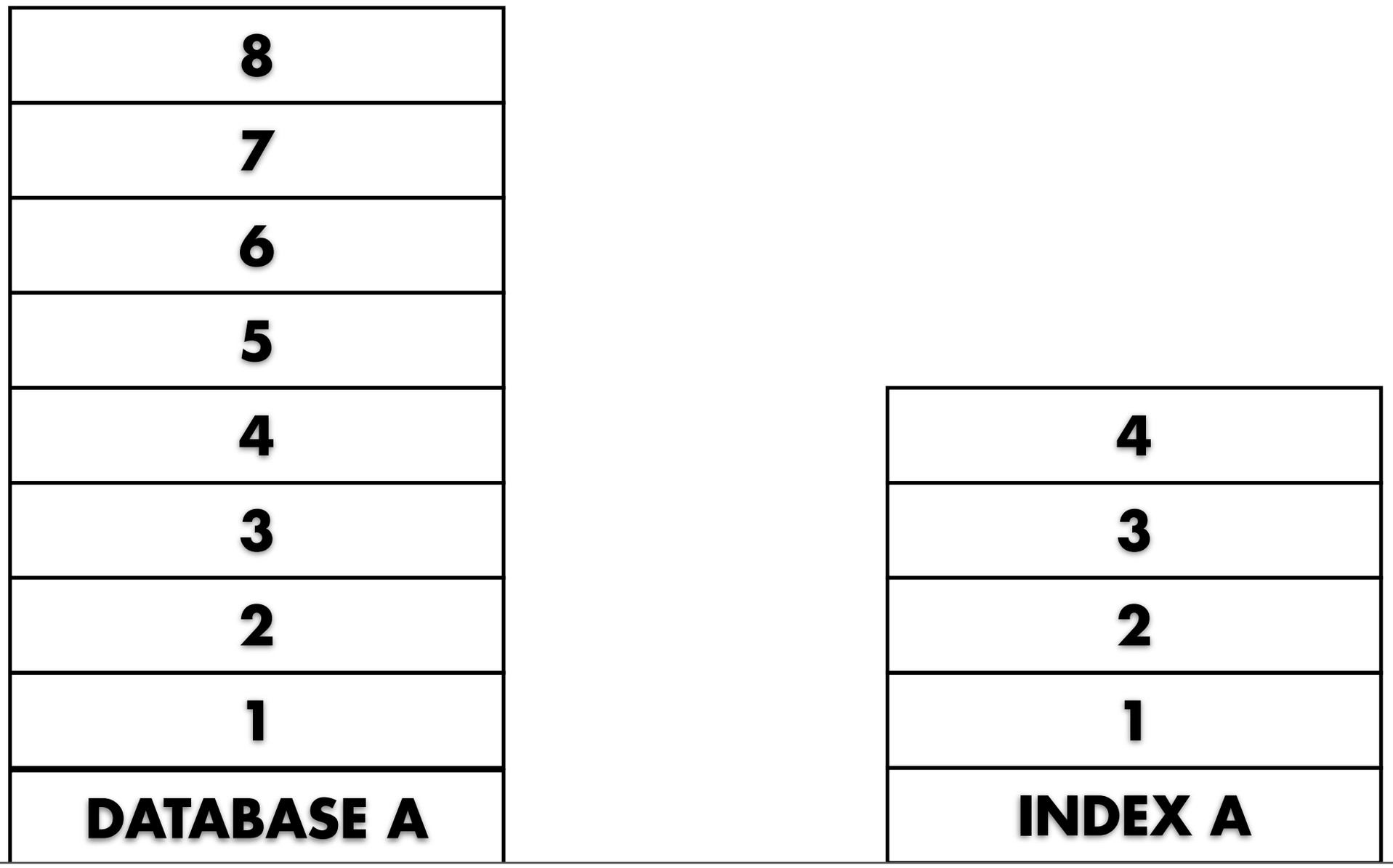
4
3
2
1
INDEX A

Monday, November 26, 12

Indexing

Monday, November 26, 12

Indexing



Monday, November 26, 12

Indexing

8
7
6
5
4
3
2
1
DATABASE A

5
4
3
2
1
INDEX A

8
7
6
5
4
3
2
1
DATABASE A

6
5
4
3
2
1
INDEX A

Monday, November 26, 12

Indexing

8
7
6
5
4
3
2
1
DATABASE A

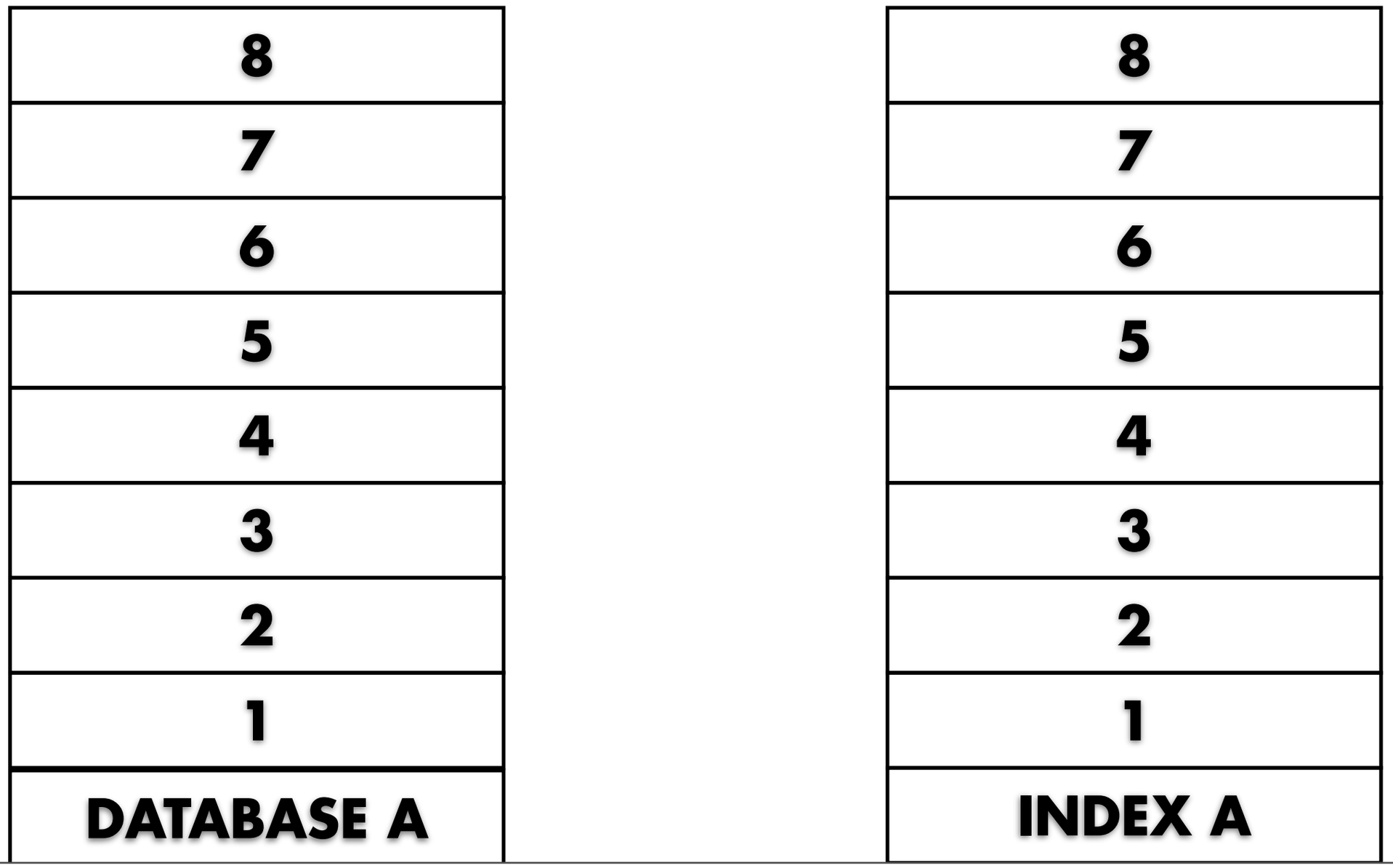
7
6
5
4
3
2
1
INDEX A

Monday, November 26, 12

Indexing

Monday, November 26, 12

Indexing



/_changes

DATABASE A

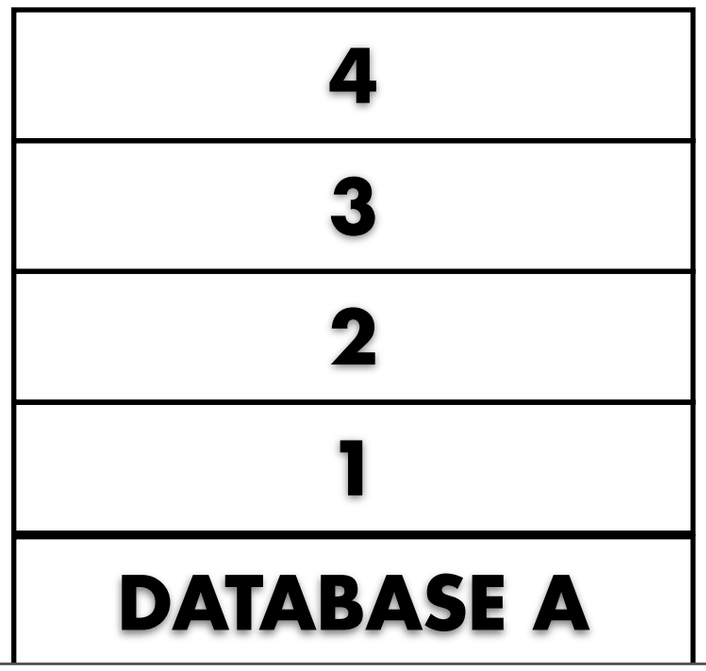
/_changes

/_changes

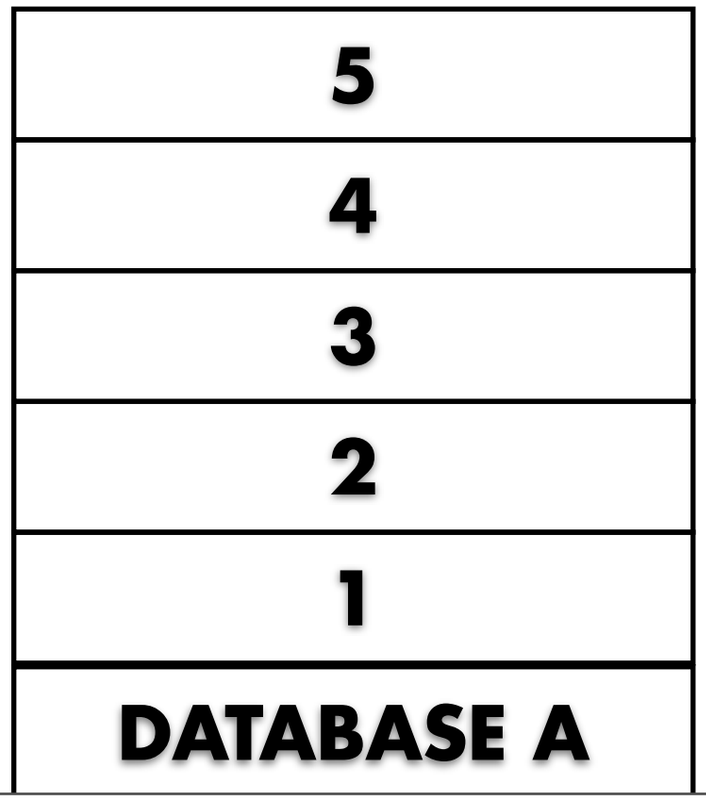
1
DATABASE A

/_changes

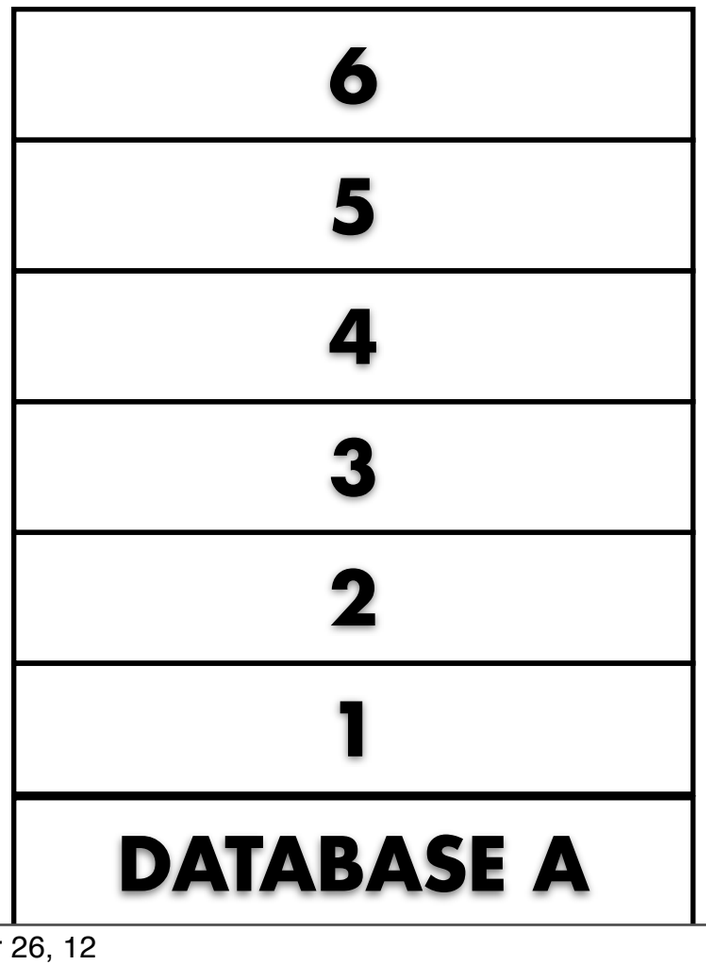
/_changes



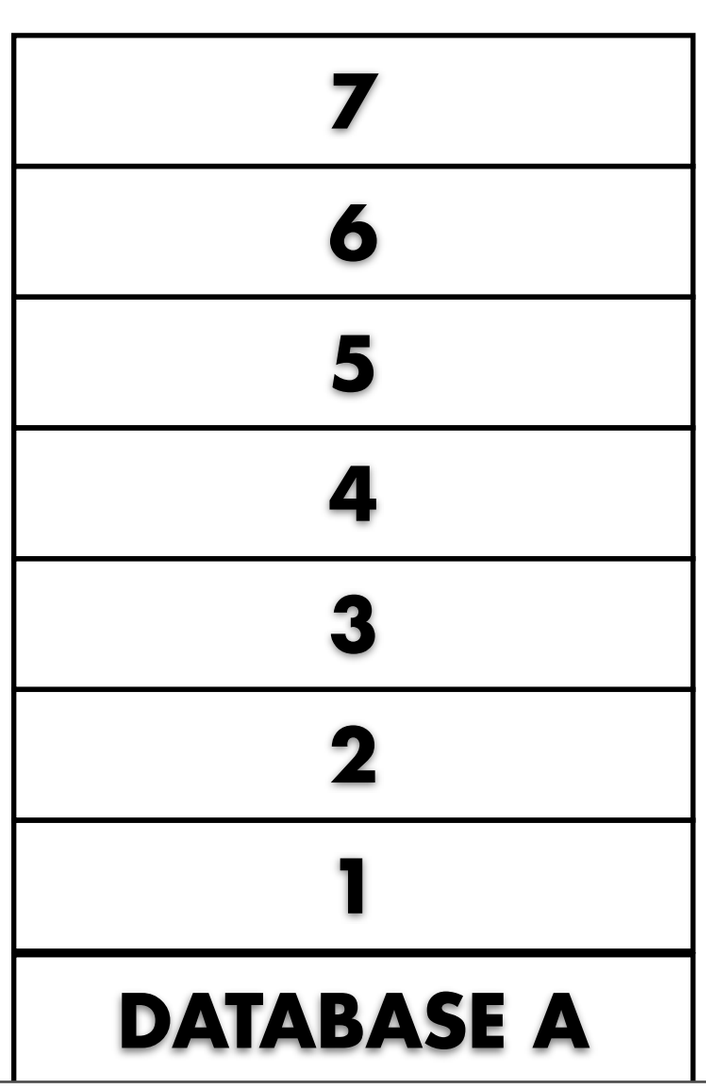
/_changes



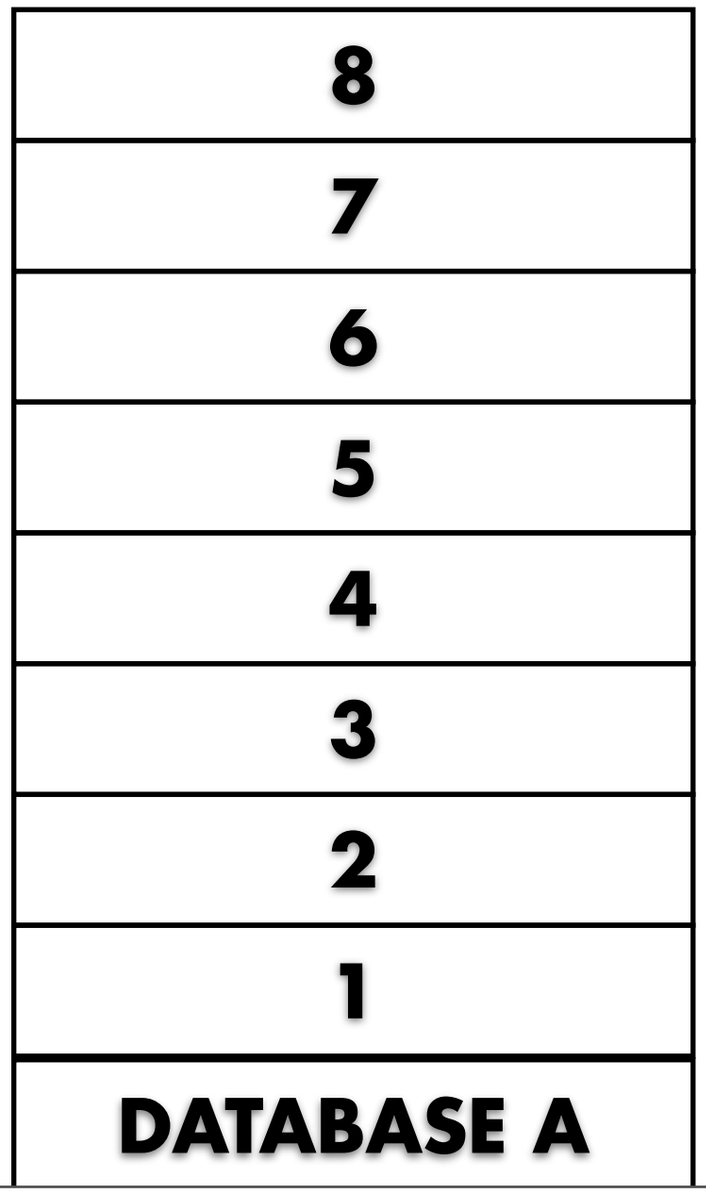
/_changes



/_changes



/_changes



/_changes

Monday, November 26, 12

Compaction

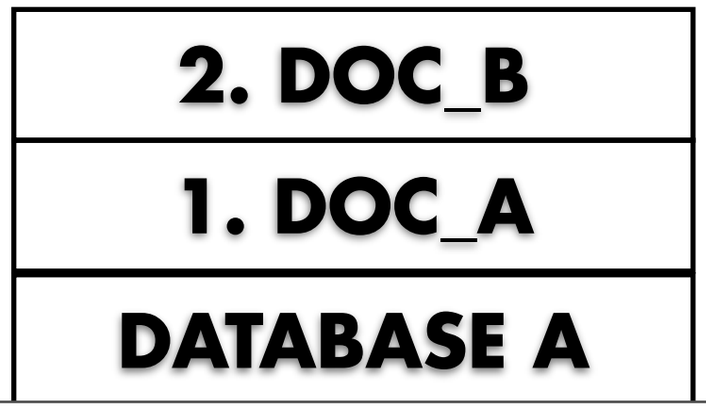
DATABASE A

Compaction

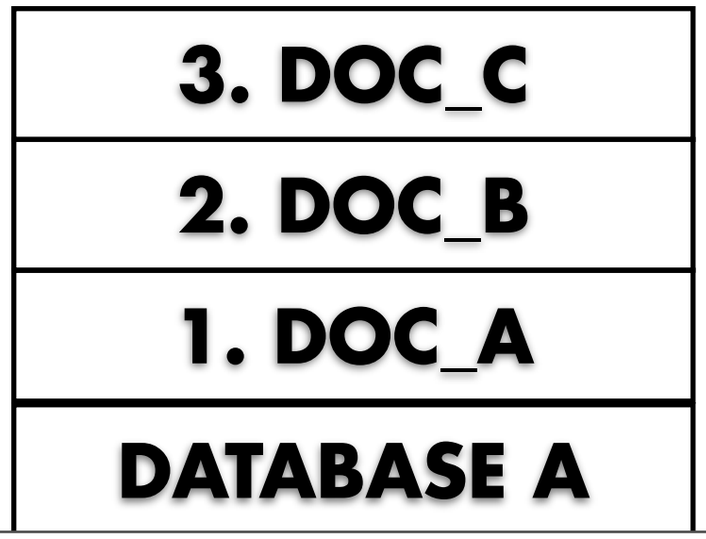
Monday, November 26, 12

Compaction

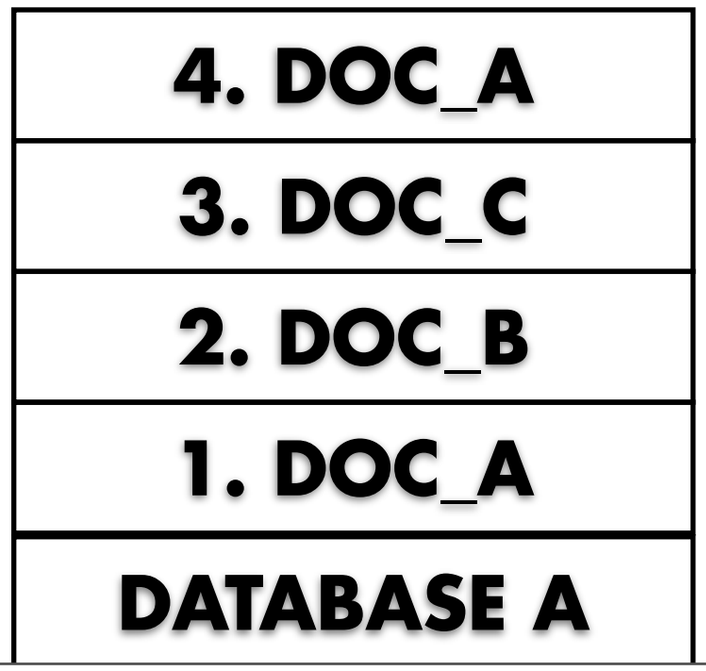
Compaction



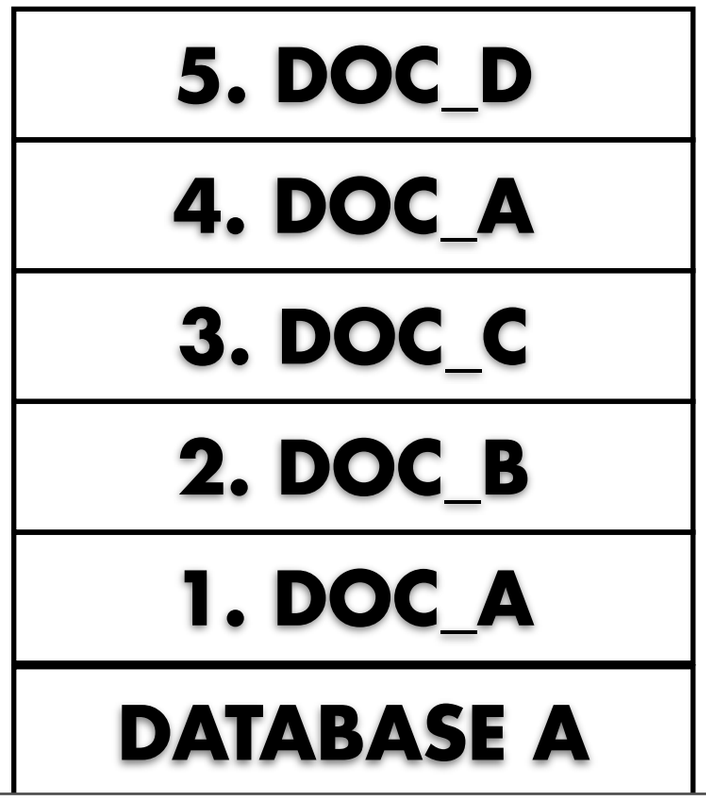
Compaction

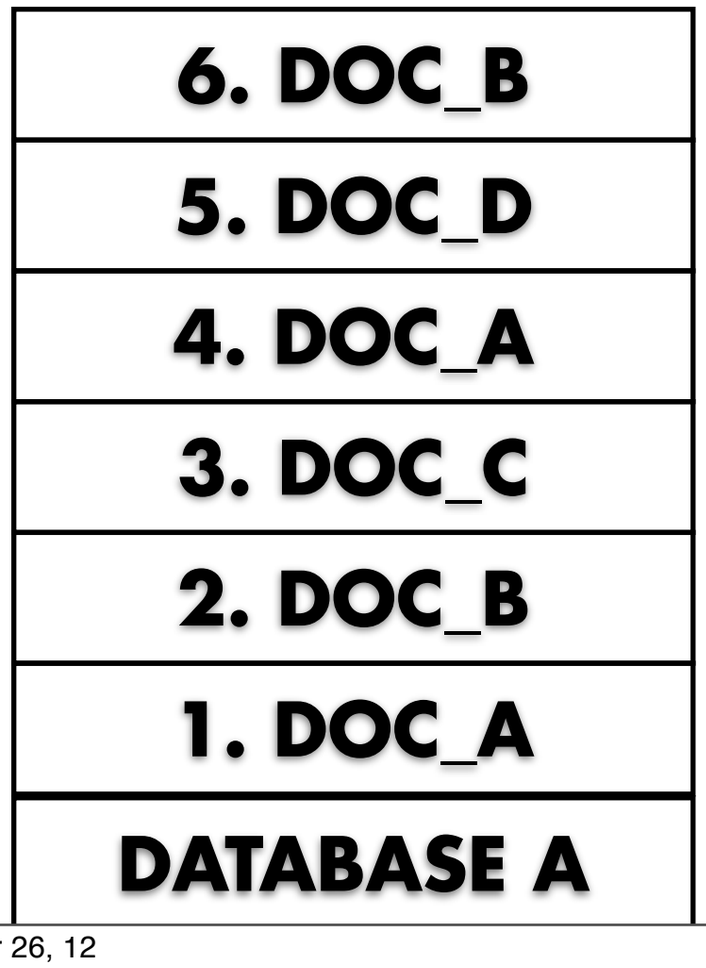


Compaction

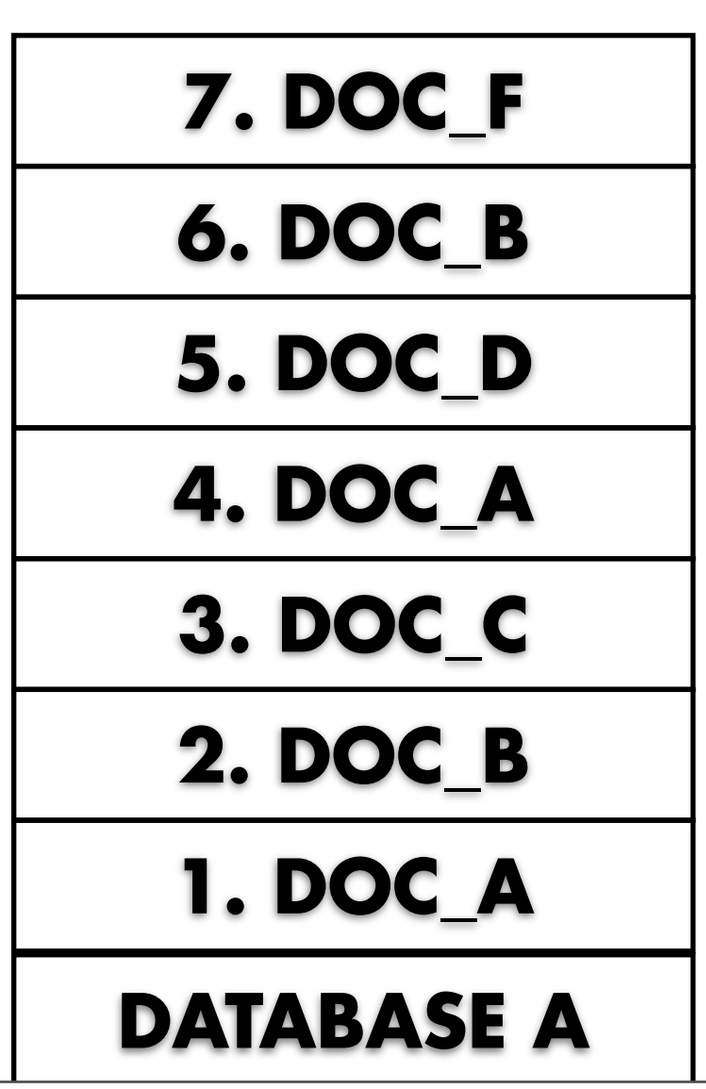


Compaction

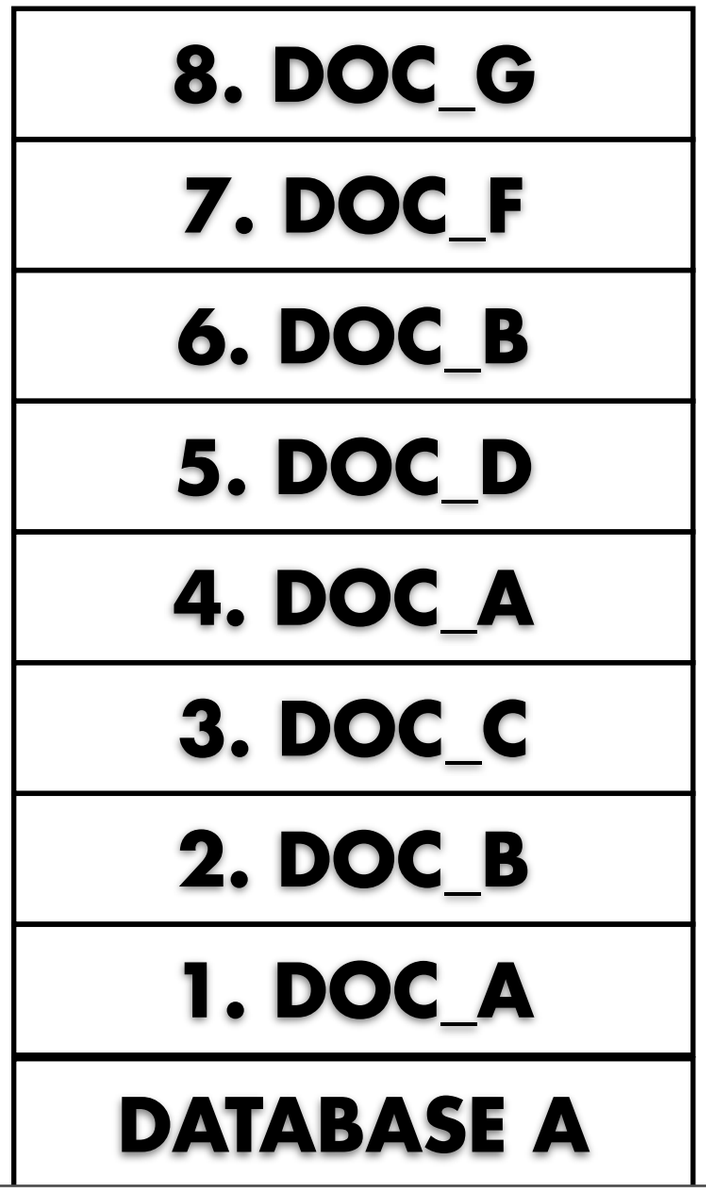




Compaction



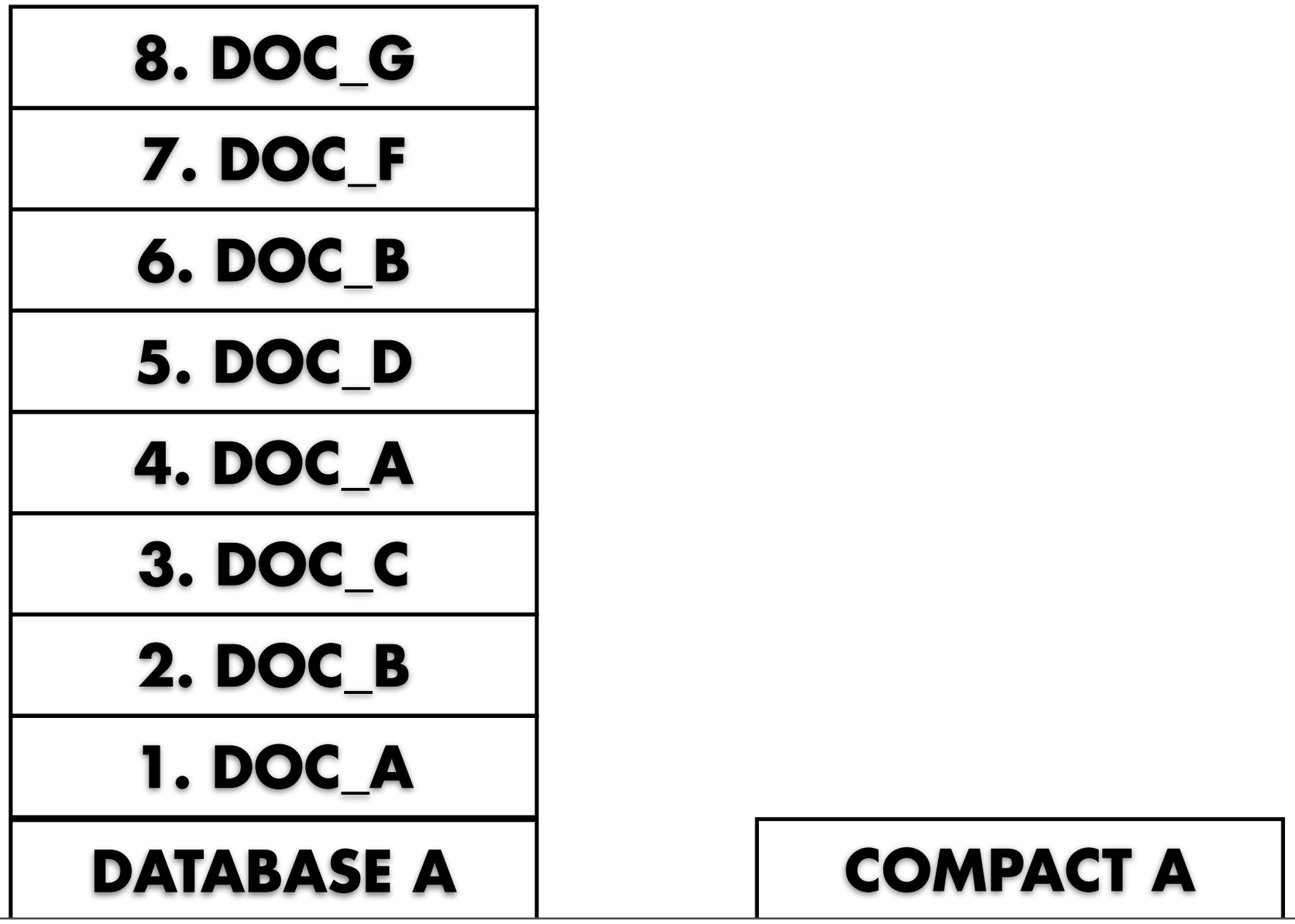
Compaction



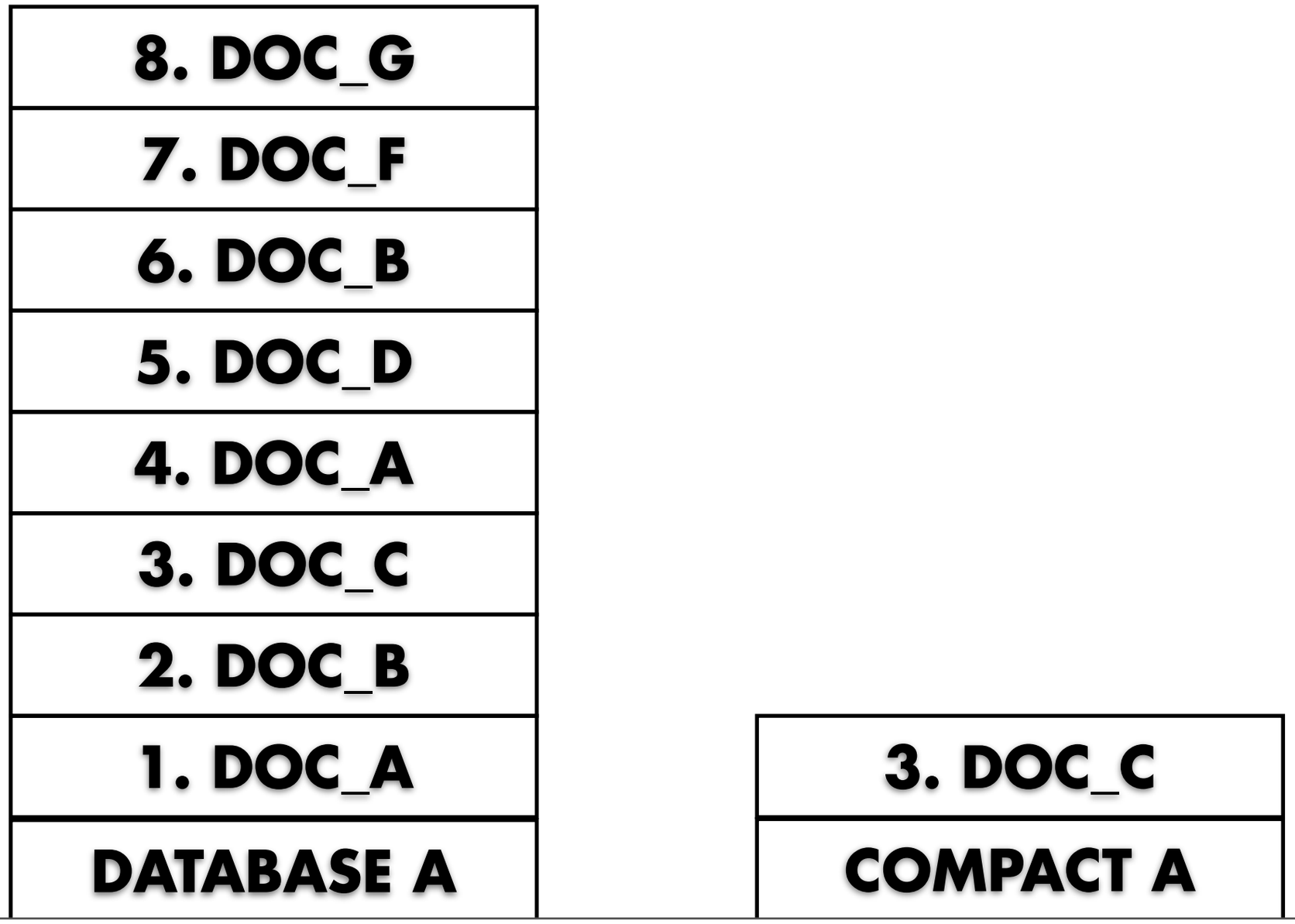
Monday, November 26, 12

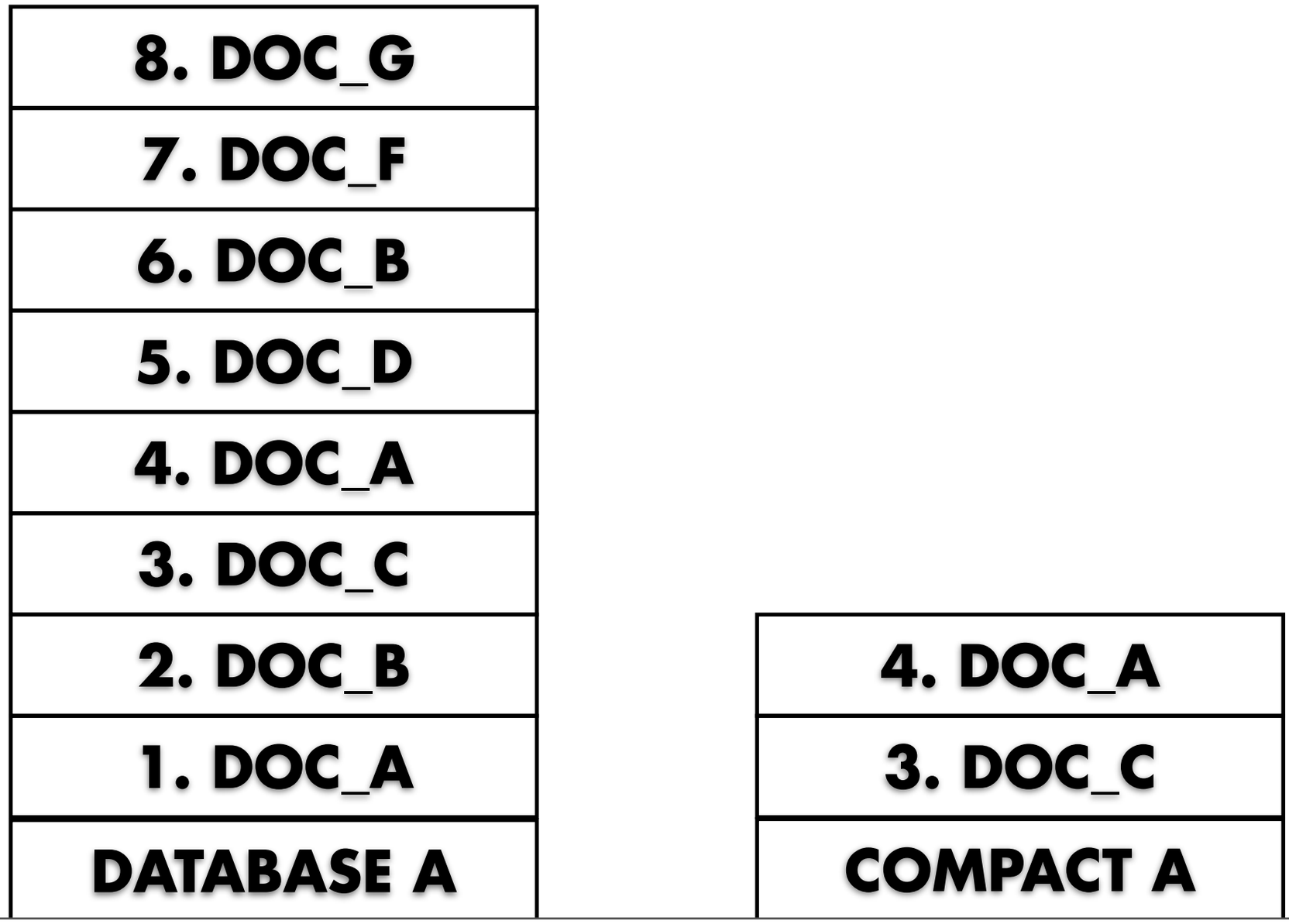
Compaction

Compaction

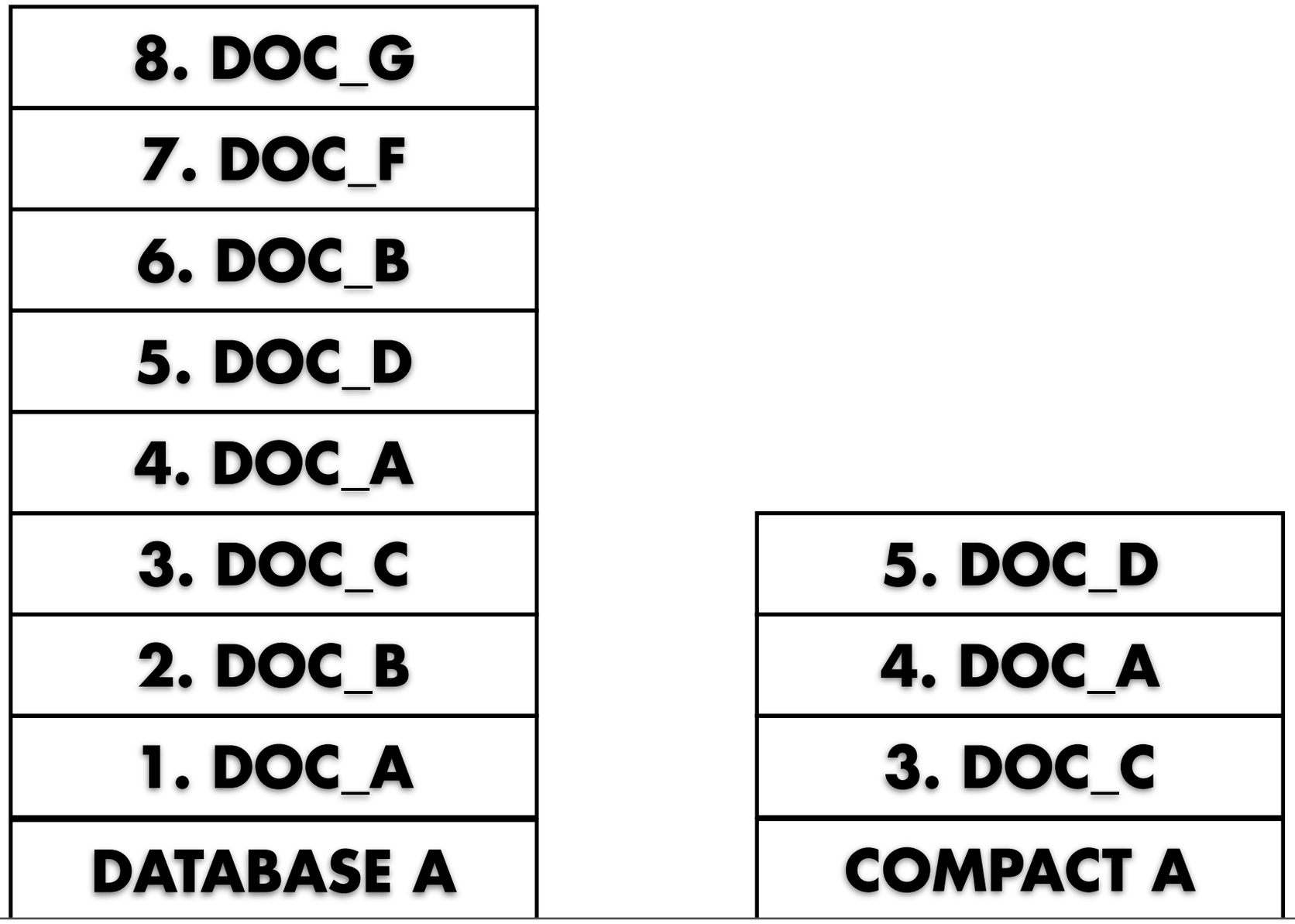

8. DOC_G
7. DOC_F
6. DOC_B
5. DOC_D
4. DOC_A
3. DOC_C
2. DOC_B
1. DOC_A
DATABASE A
COMPACT A

Compaction

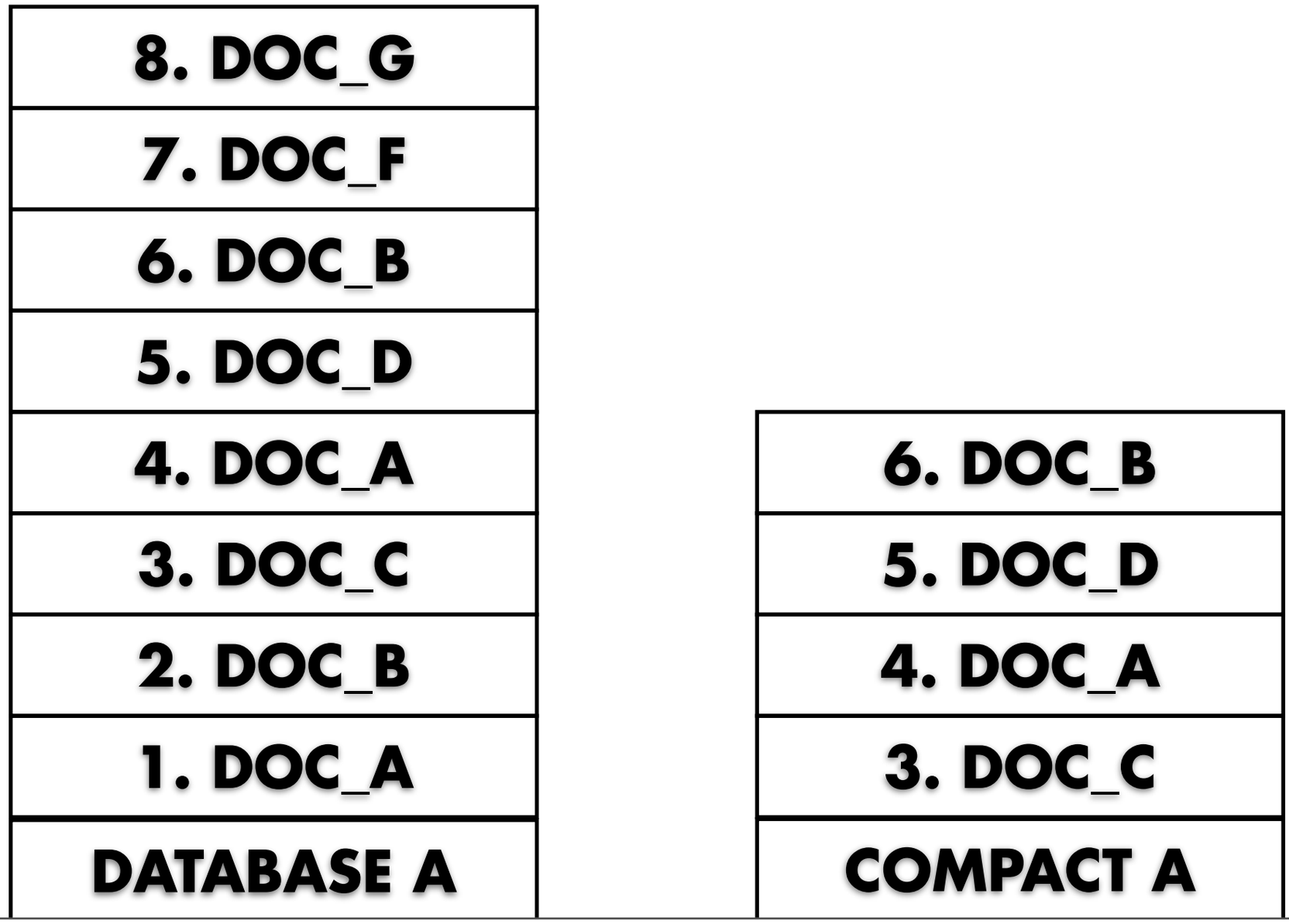

8. DOC_G
7. DOC_F
6. DOC_B
5. DOC_D
4. DOC_A
3. DOC_C
2. DOC_B
1. DOC_A
DATABASE A
3. DOC_C
COMPACT A


8. DOC_G
7. DOC_F
6. DOC_B
5. DOC_D
4. DOC_A
3. DOC_C
2. DOC_B
1. DOC_A
DATABASE A
4. DOC_A
3. DOC_C
COMPACT A

Compaction

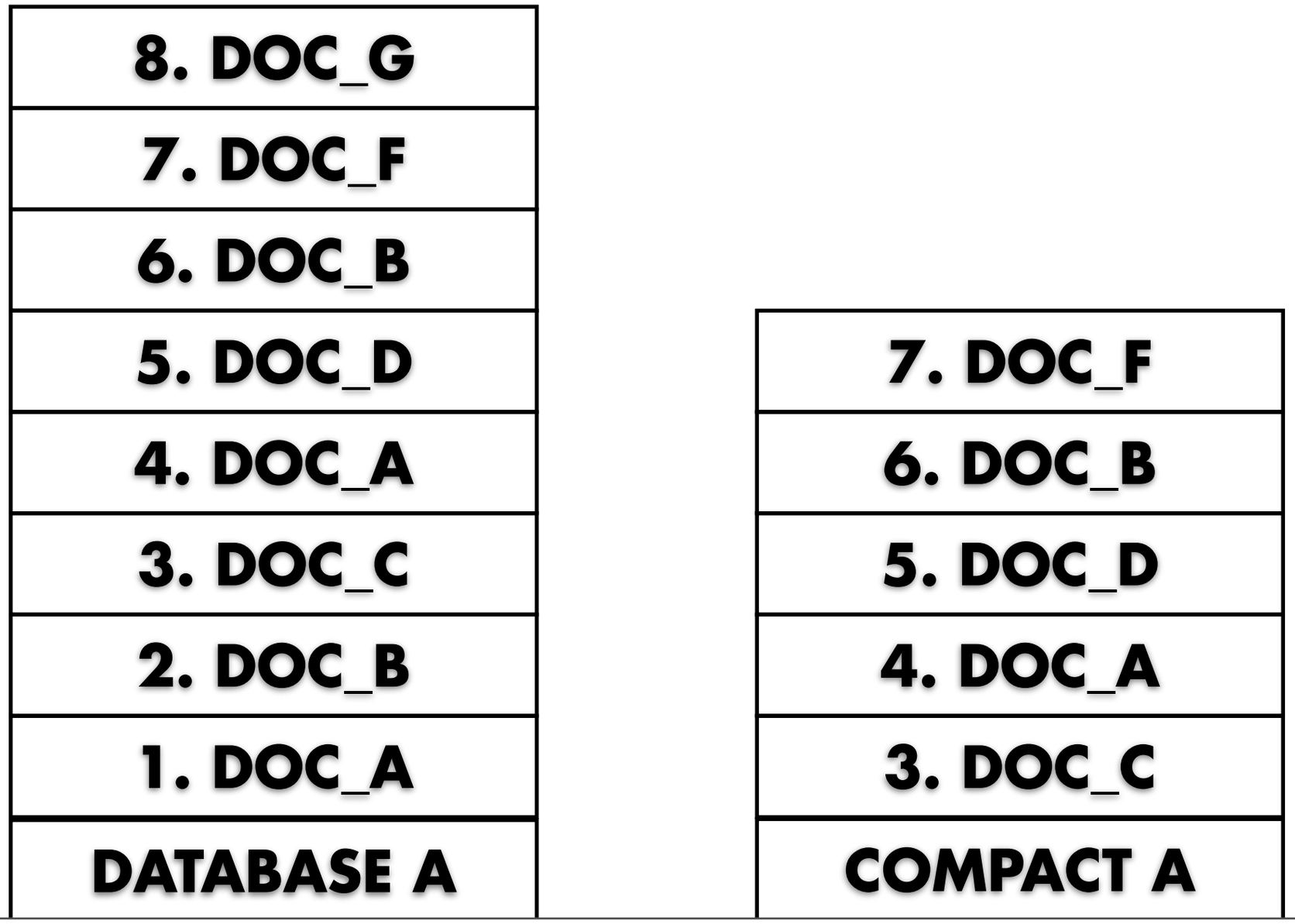

8. DOC_G
7. DOC_F
6. DOC_B
5. DOC_D
4. DOC_A
3. DOC_C
2. DOC_B
1. DOC_A
DATABASE A
5. DOC_D
4. DOC_A
3. DOC_C
COMPACT A

Compaction



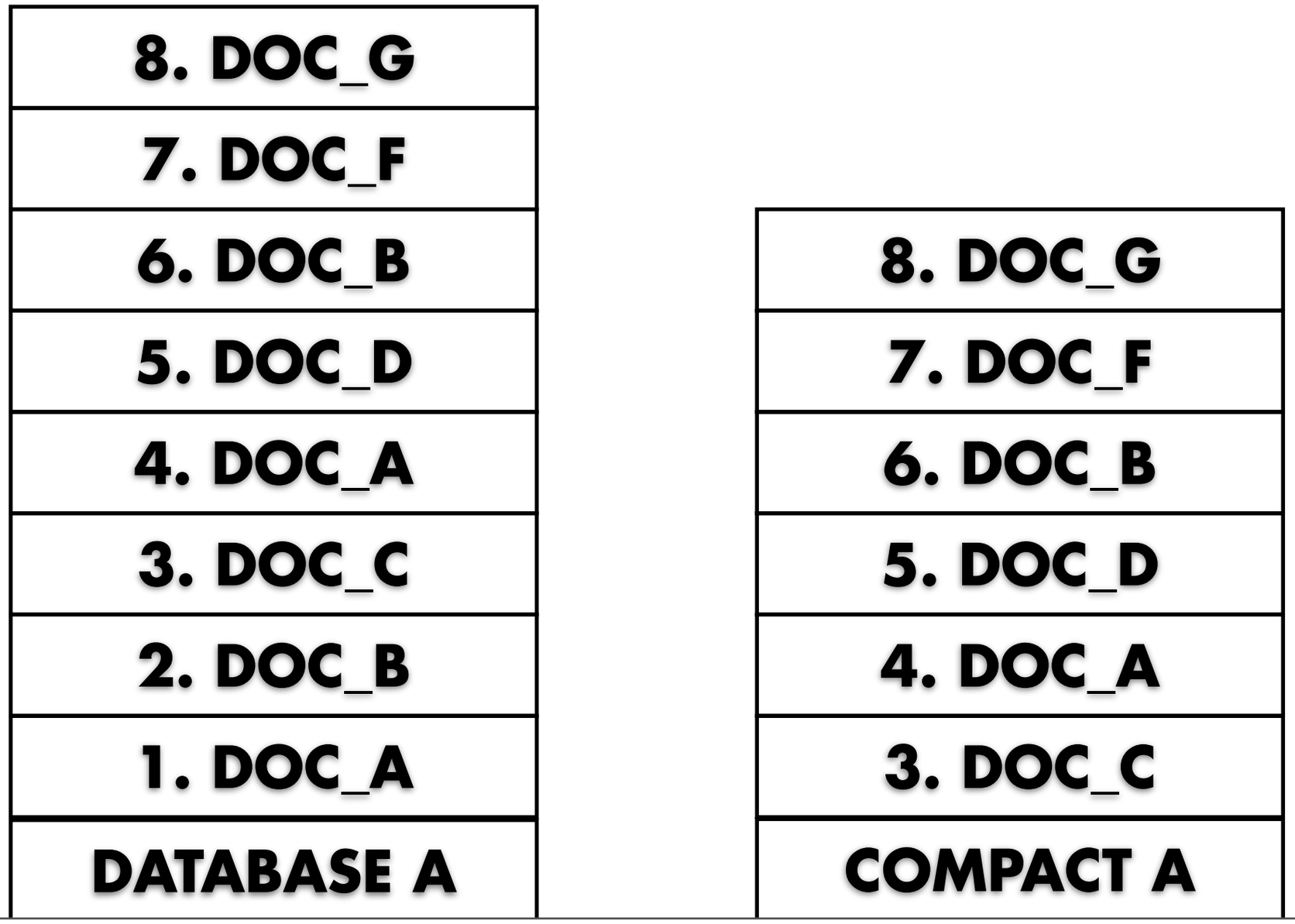
Monday, November 26, 12

Compaction



Monday, November 26, 12

Compaction



Monday, November 26, 12

Compaction

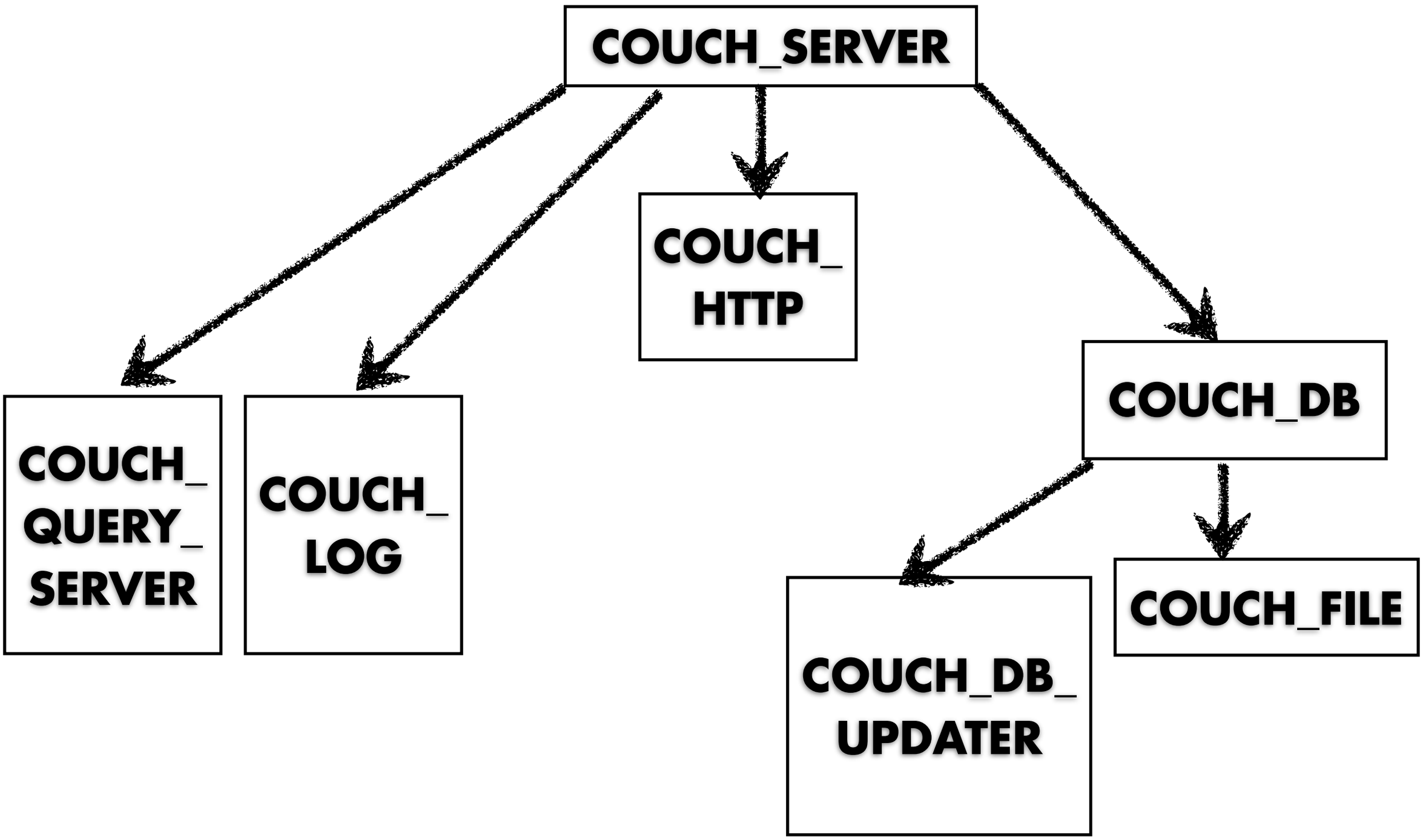
Erlang

- Small codebase
- Efficient in small teams

- Supervision tree
- Isolated processes
- Concurrency

- Portable runtime

- Hard to recruit for
- Steep ramp-on



- Small codebase
- Efficient in small teams

- Supervision tree
- Isolated processes
- Concurrency

- Portable runtime

- Hard to recruit for
- Steep ramp-on

Potential Improvements

- Smarter compactor
- Smarter file-storage
- Less custom HTTP handling
- More indexers

The End

Thanks!

Monday, November 26, 12