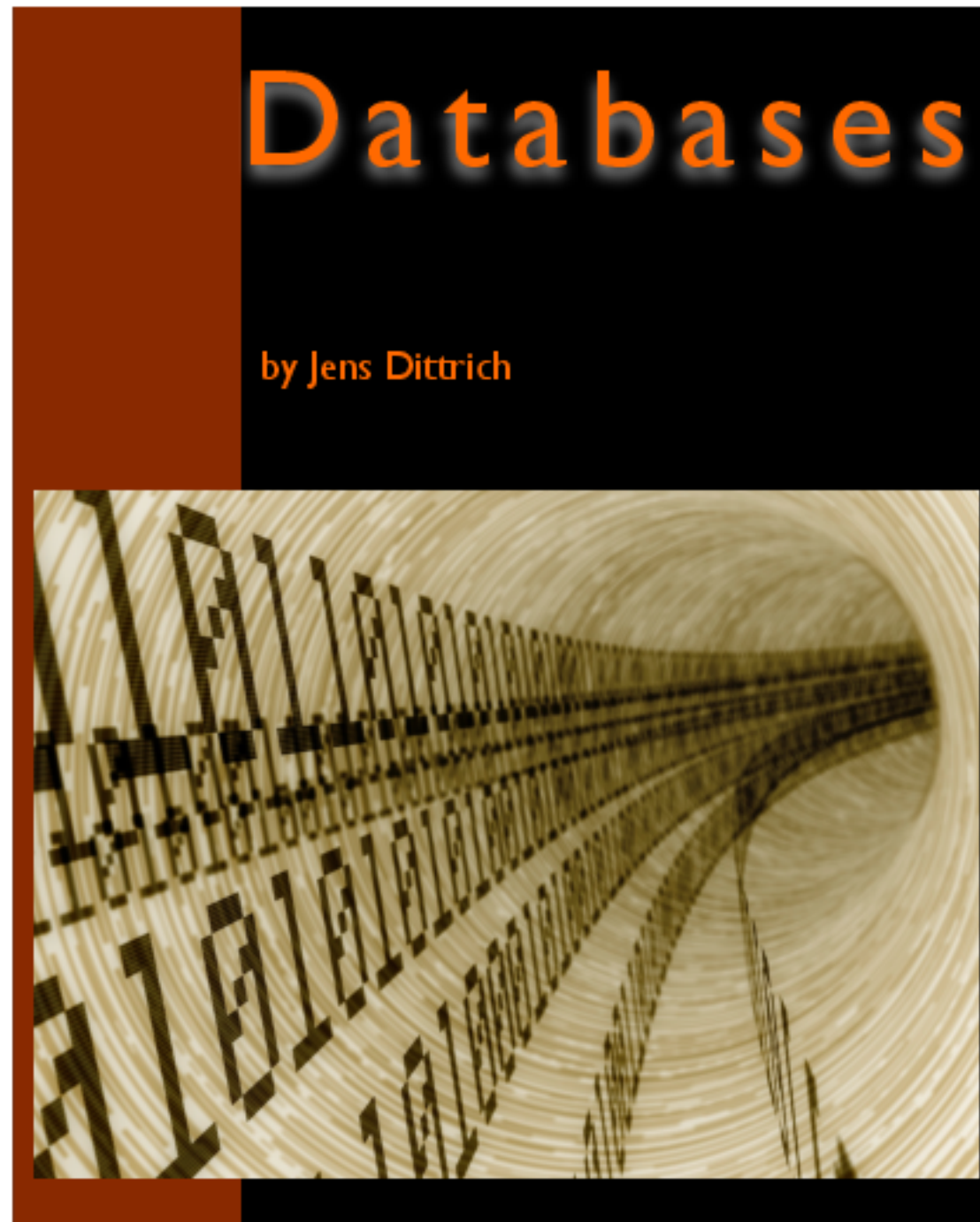


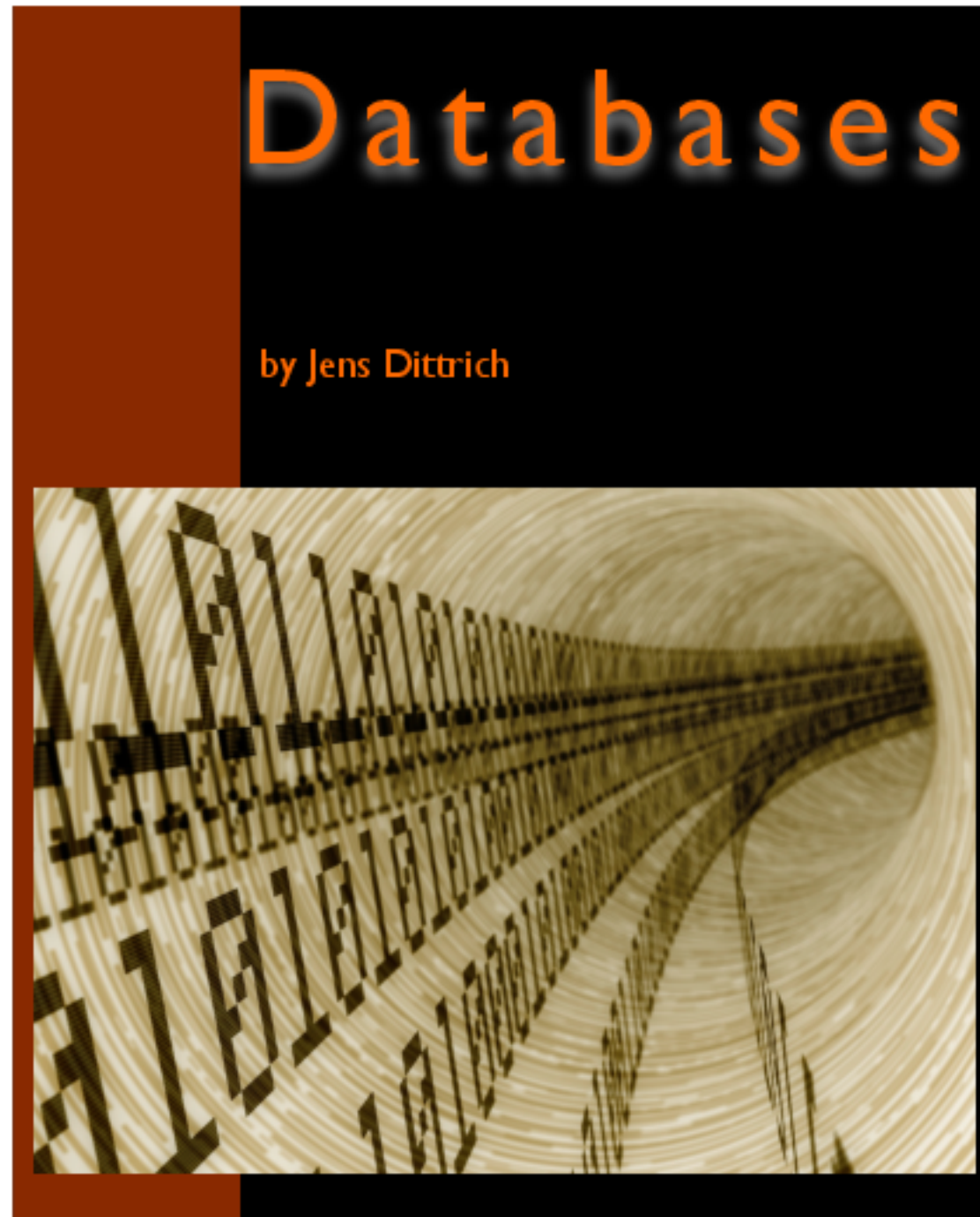
Publishing a Book

1st edition ↙



Publishing a Book

1st edition

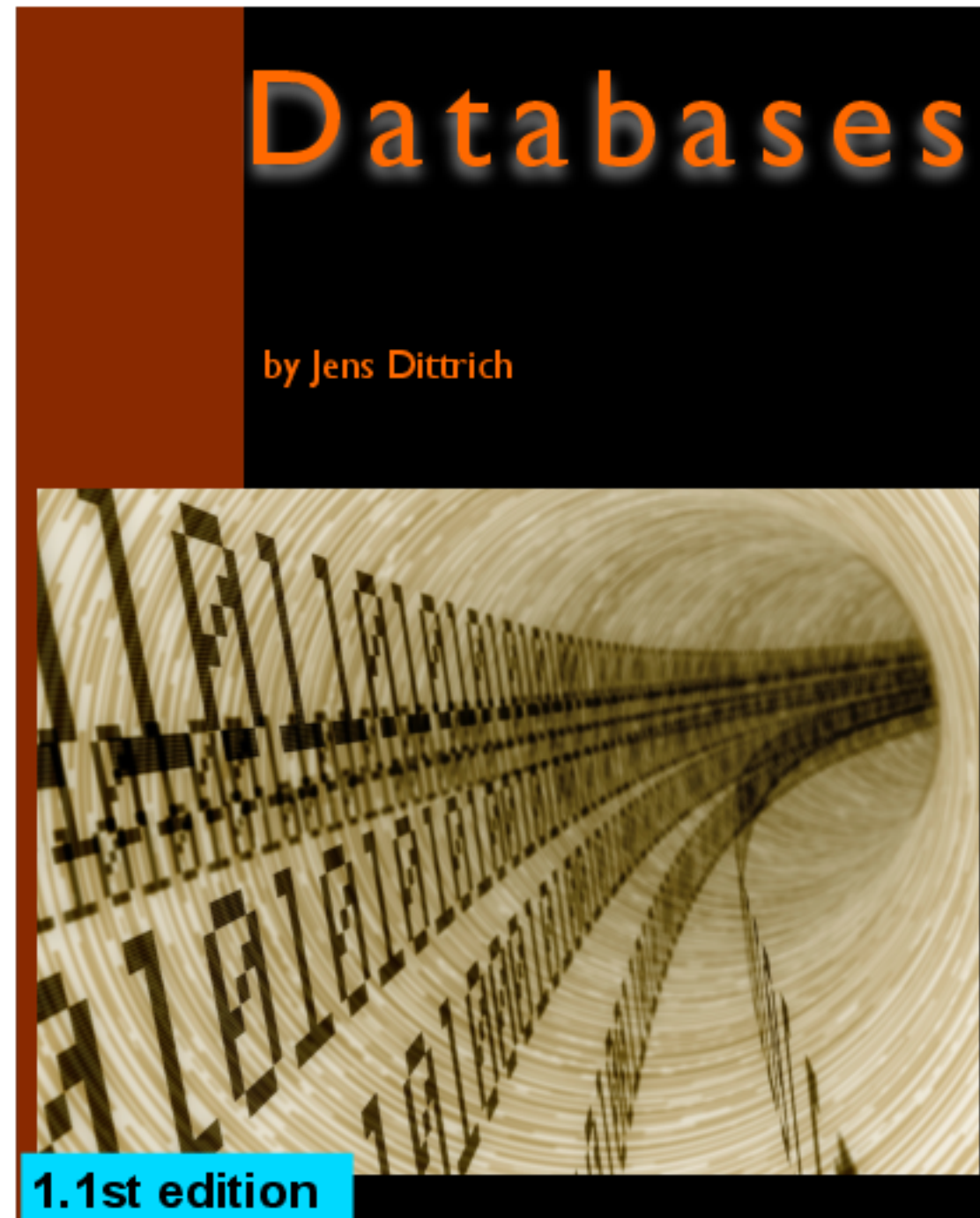


changes:



Publishing a Book

1.1st edition



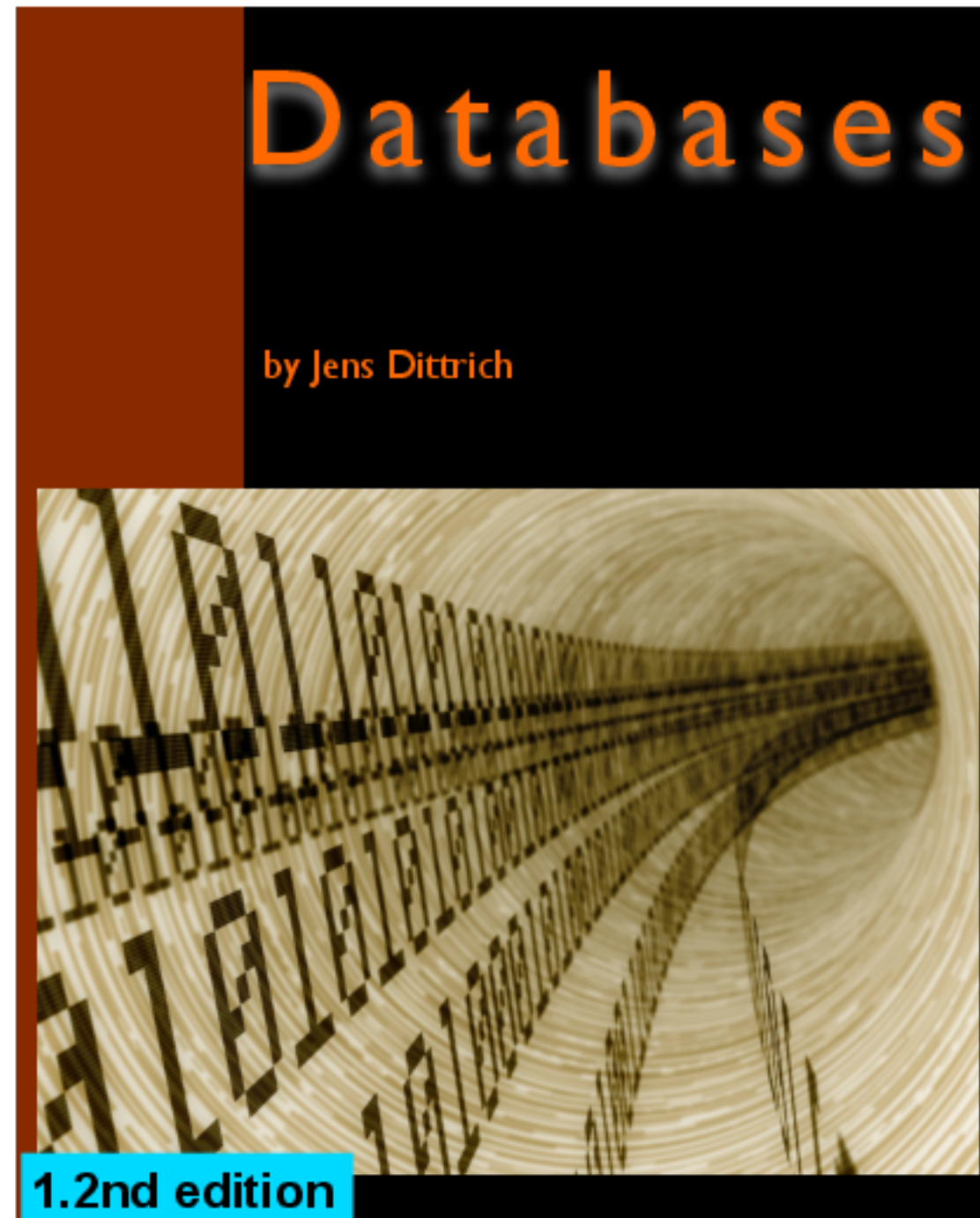
RW

changes:

page 23:
“datbase” → “database”

Publishing a Book

1.2nd edition

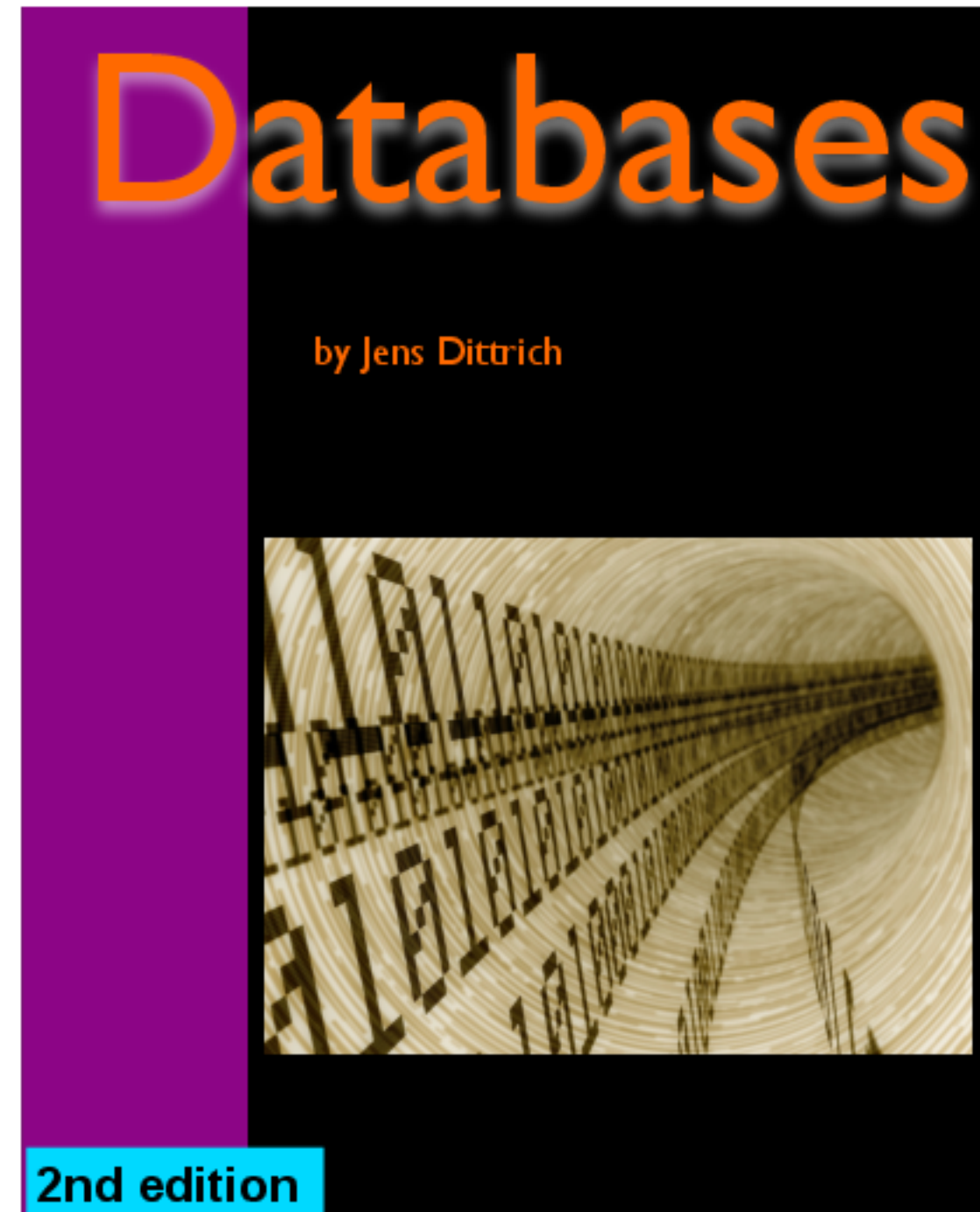


changes:

page 23:
“datbase” → “database”
page 345:
“idex” → “index”

Publishing a Book

2nd edition

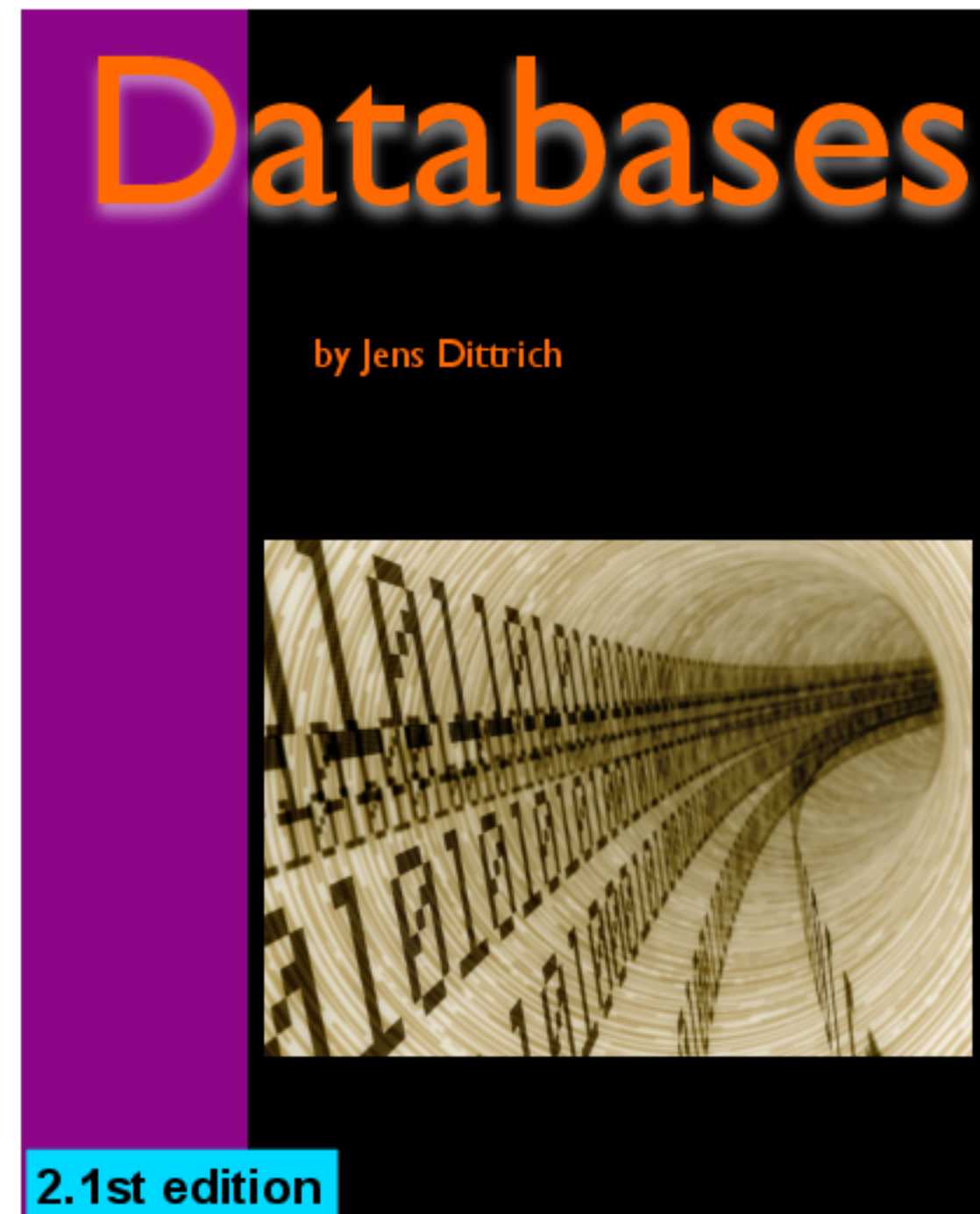


changes:

page 23:
“datbase” → “database”
page 345:
“idex” → “index”
page 77:
“idex” → “index”

Publishing a Book

2.1st edition

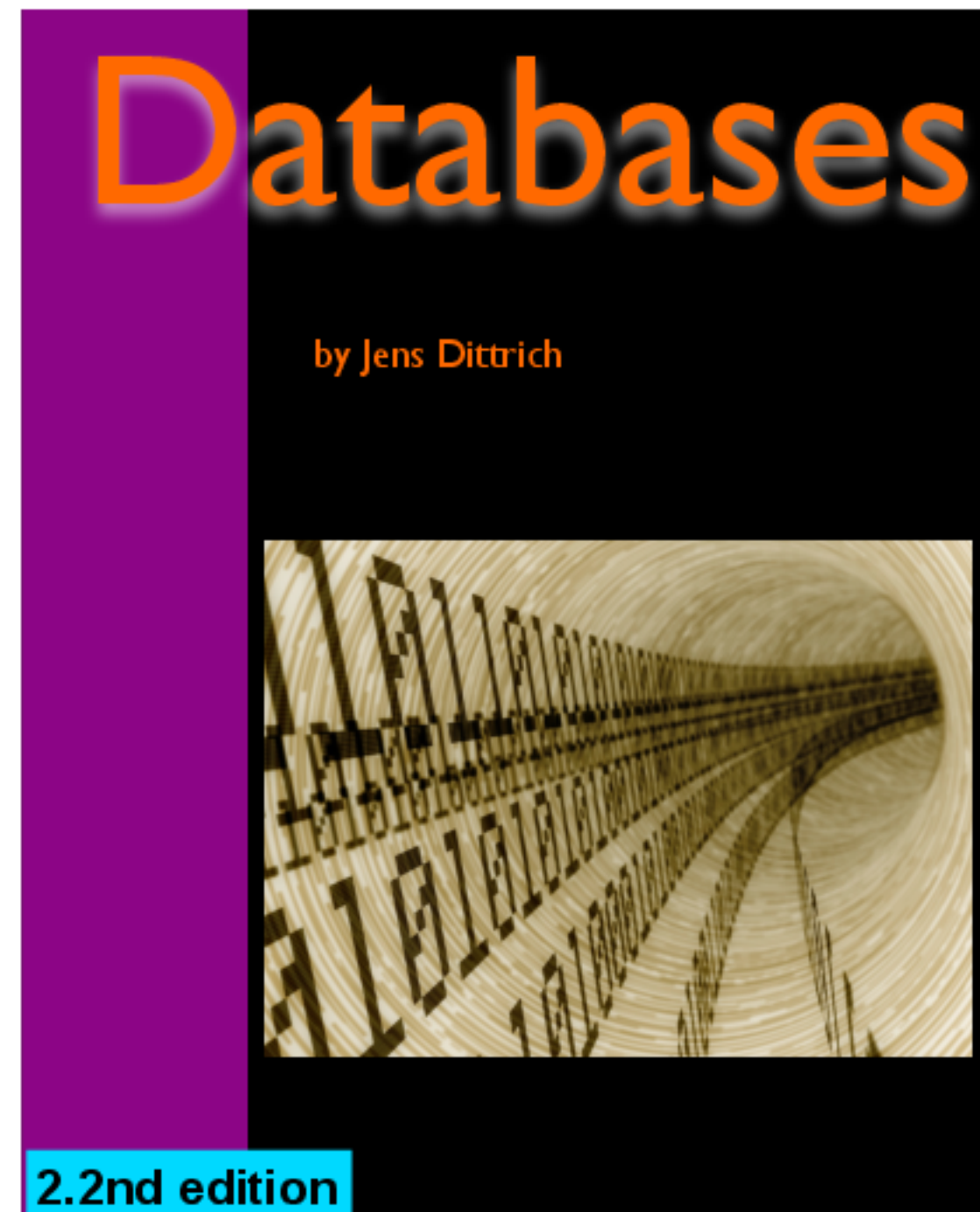


changes:

page 23:
“datbase” → “database”
page 345:
“idex” → “index”
page 77:
“idex” → “index”
page 75:
“kamera” → “camera”

Publishing a Book

2.2nd edition

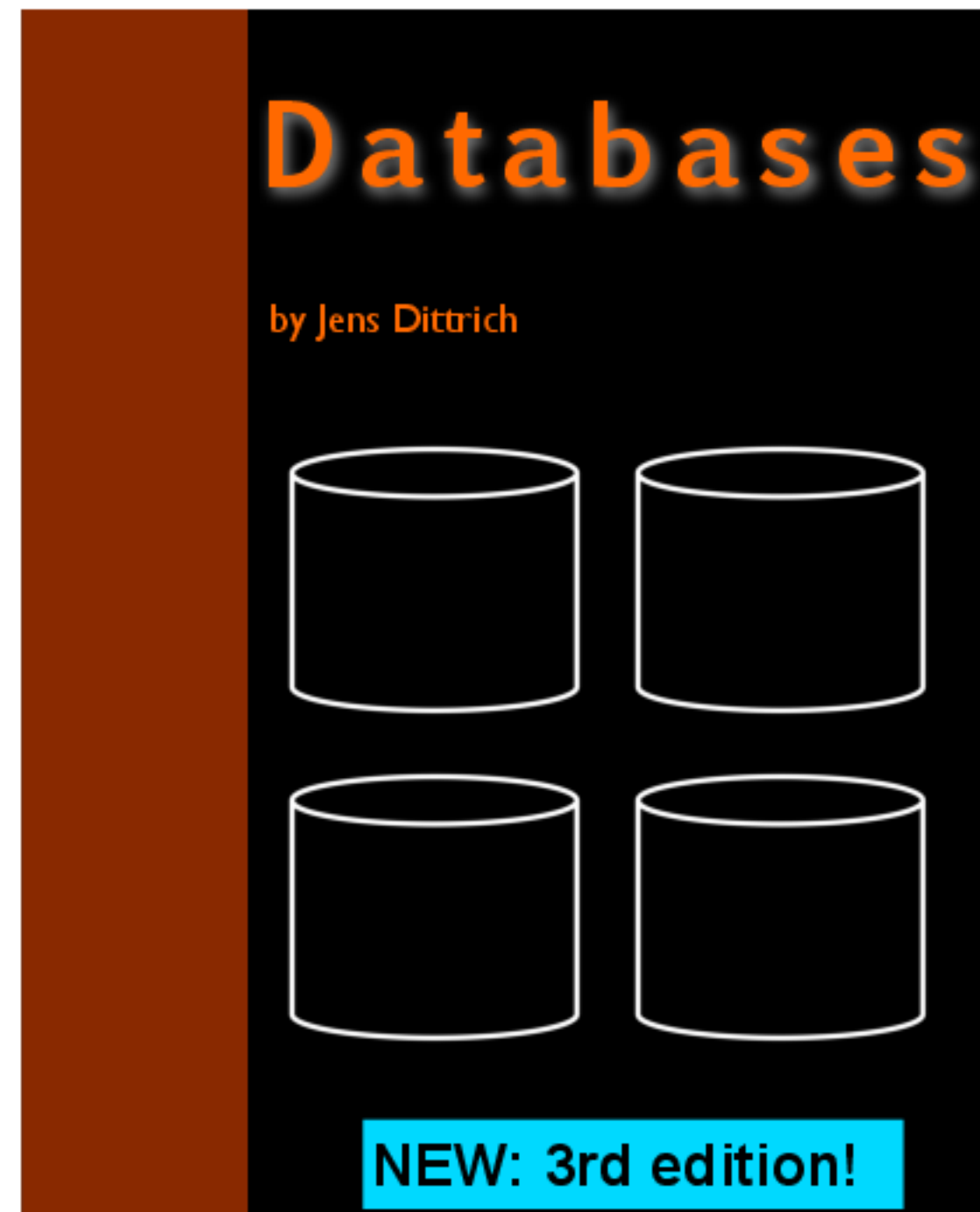


changes:

page 23:
“datbase” → “database”
page 345:
“idex” → “index”
page 77:
“idex” → “index”
page 75:
“kamera” → “camera”
page 143:
“big date” → “big data”

Publishing a Book

3rd edition

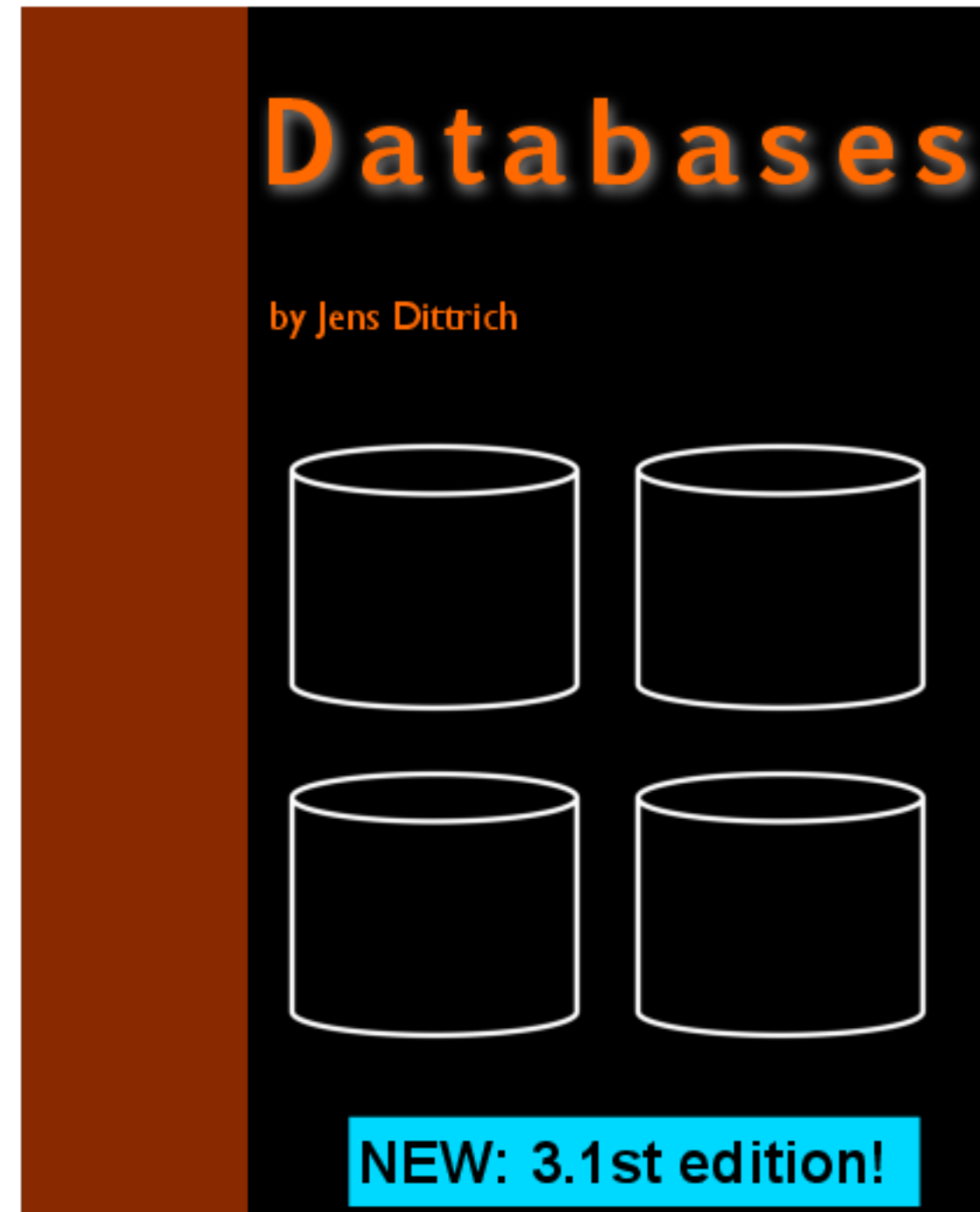


changes:

page 23:
“datbase” → “database”
page 345:
“idex” → “index”
page 77:
“idex” → “index”
page 75:
“kamera” → “camera”
page 143:
“big date” → “big data”
new chapter on “tools”

Publishing a Book

3.1st edition



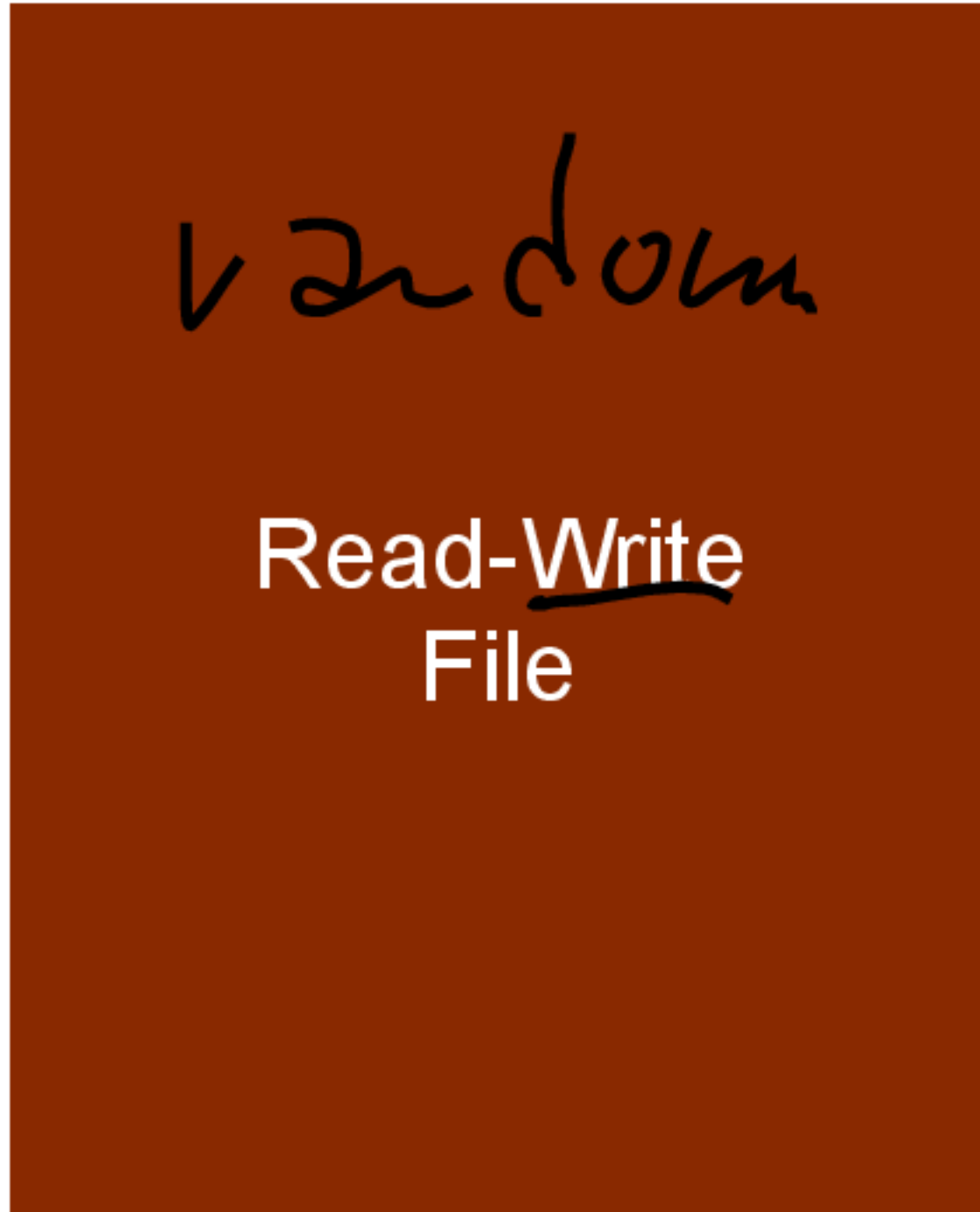
WAL write
 ahead
changes: logging

page 23:
"datbase" → "database"
page 345:
"idex" → "index"
page 77:
"idex" → "index"
page 75:
"kamera" → "camera"
page 143:
"big date" → "big data"
new chapter on "tools"
page 55:
"profi" → "profile"

Logging

DB

current edition



ce

old

Log File

changes:

sequential



= yet another instance of: The Data Redundancy Pattern and The All Levels are Equal Pattern

Advantages

may be applied at almost any level, e.g. entire databases, files, indexes (disk or main memory), tables, blocks, pages, rows, ...

fast sequential write to log

yet: cheap reads (no read of log required)

log corresponds to incremental backup (-> Log Archive, Log Shipping)

data redundancy (recovery)

“truth” = Read-Write File **OR** the log: **“The Log is the Database”**

current

version

Drawbacks

additional storage space for log

still random write I/O (Read-Write File)

log may become large

prune

Drawbacks

additional storage space for log

still random write I/O (Read-Write File)

log may become large

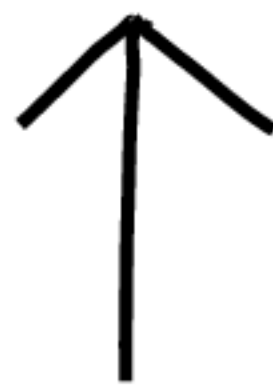
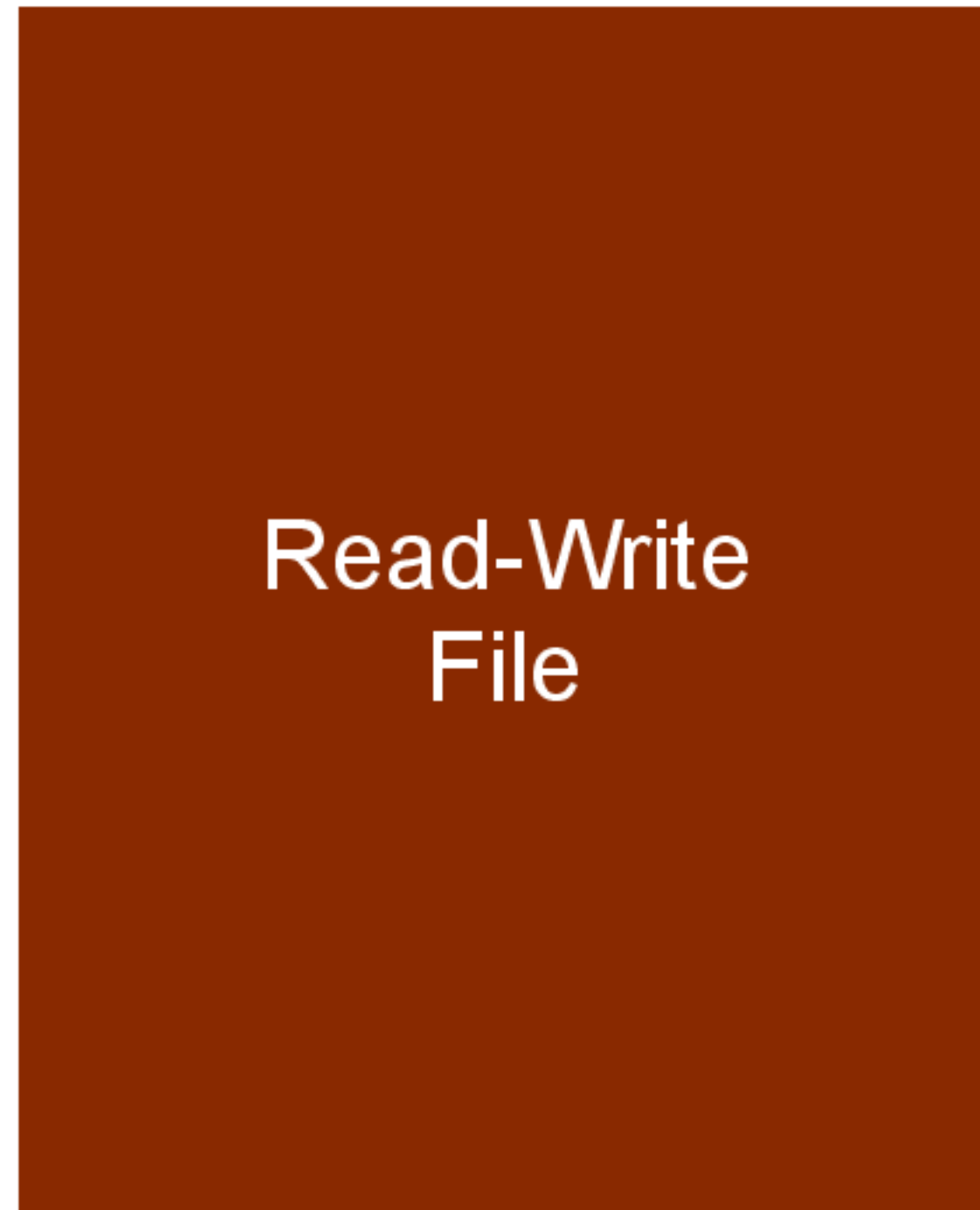
extra storage space

Differential Files vs Logging

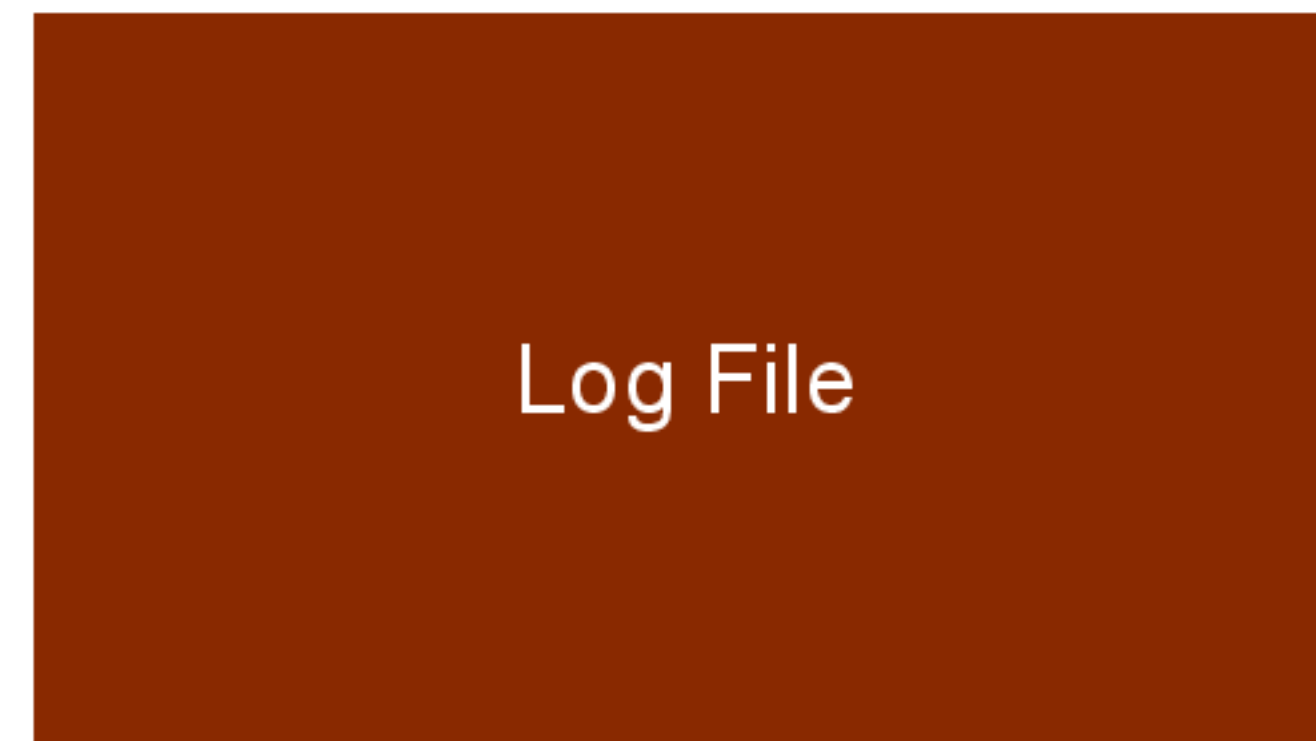
	Differential Files	Logging
main Idea	given a file X: collect changes to X in a separate structure Y	
X	RO	RW
X'	X + DiffFile	X + change
“truth”	X + DiffFile(s)	X OR the log

Example: Combining Differential Files and Logging

current edition

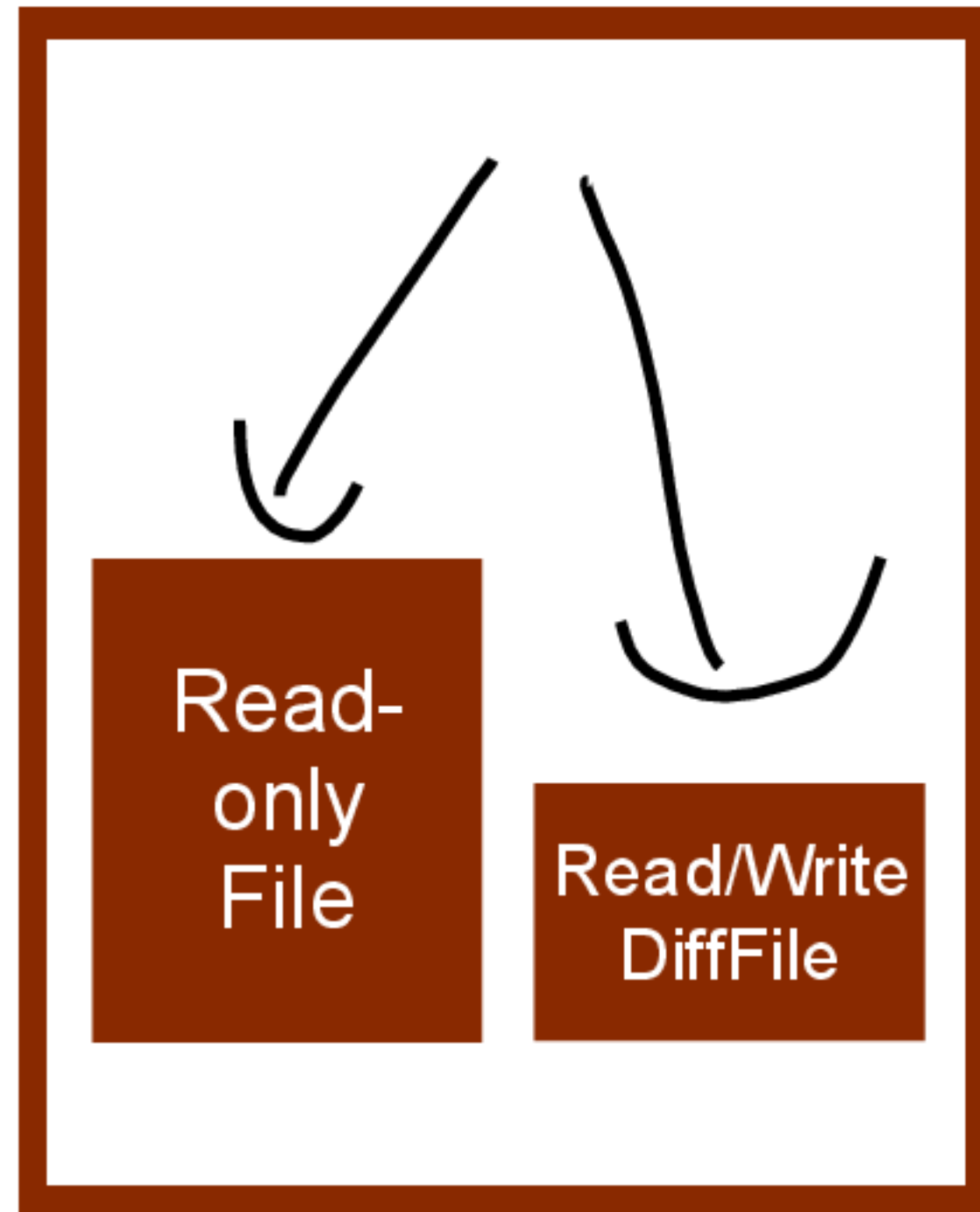


changes:

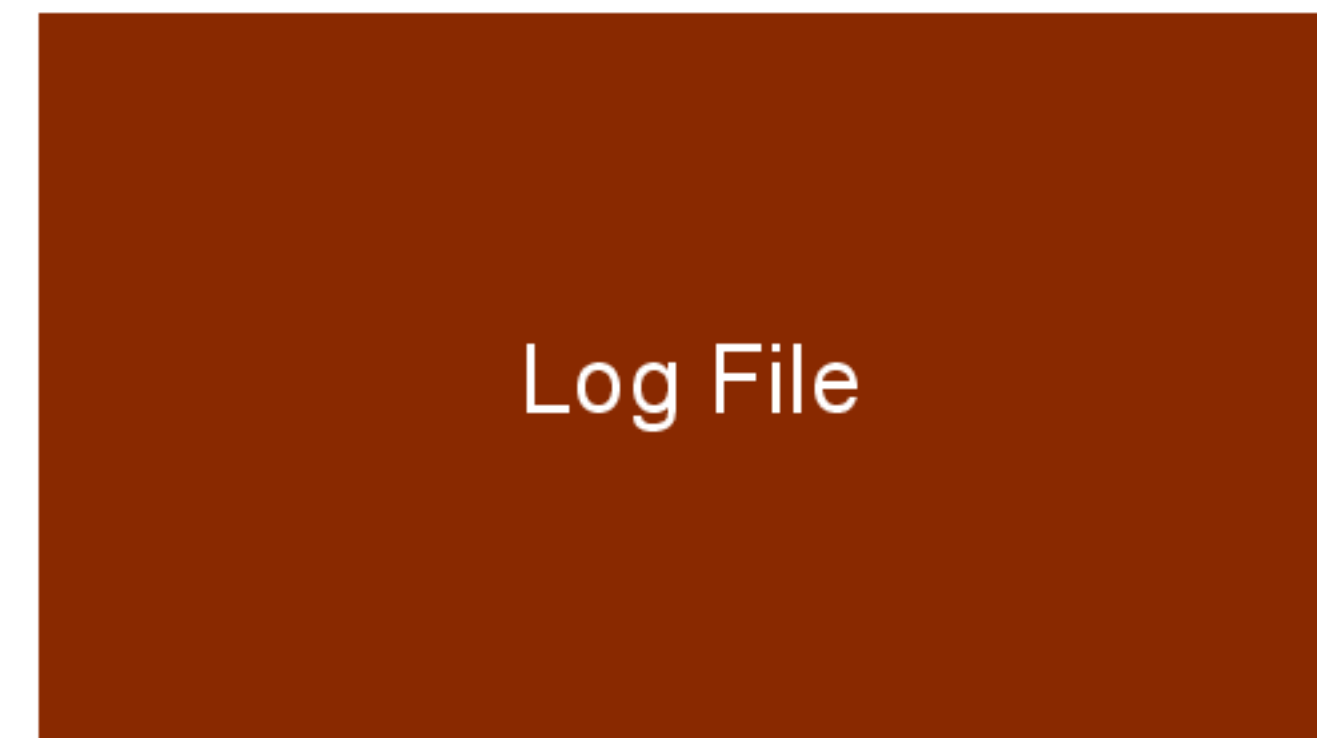


Example: Combining Differential Files and Logging

current edition



changes:



Copyrights

© iStock.com:

voyager624