



# 02

# Gitshop

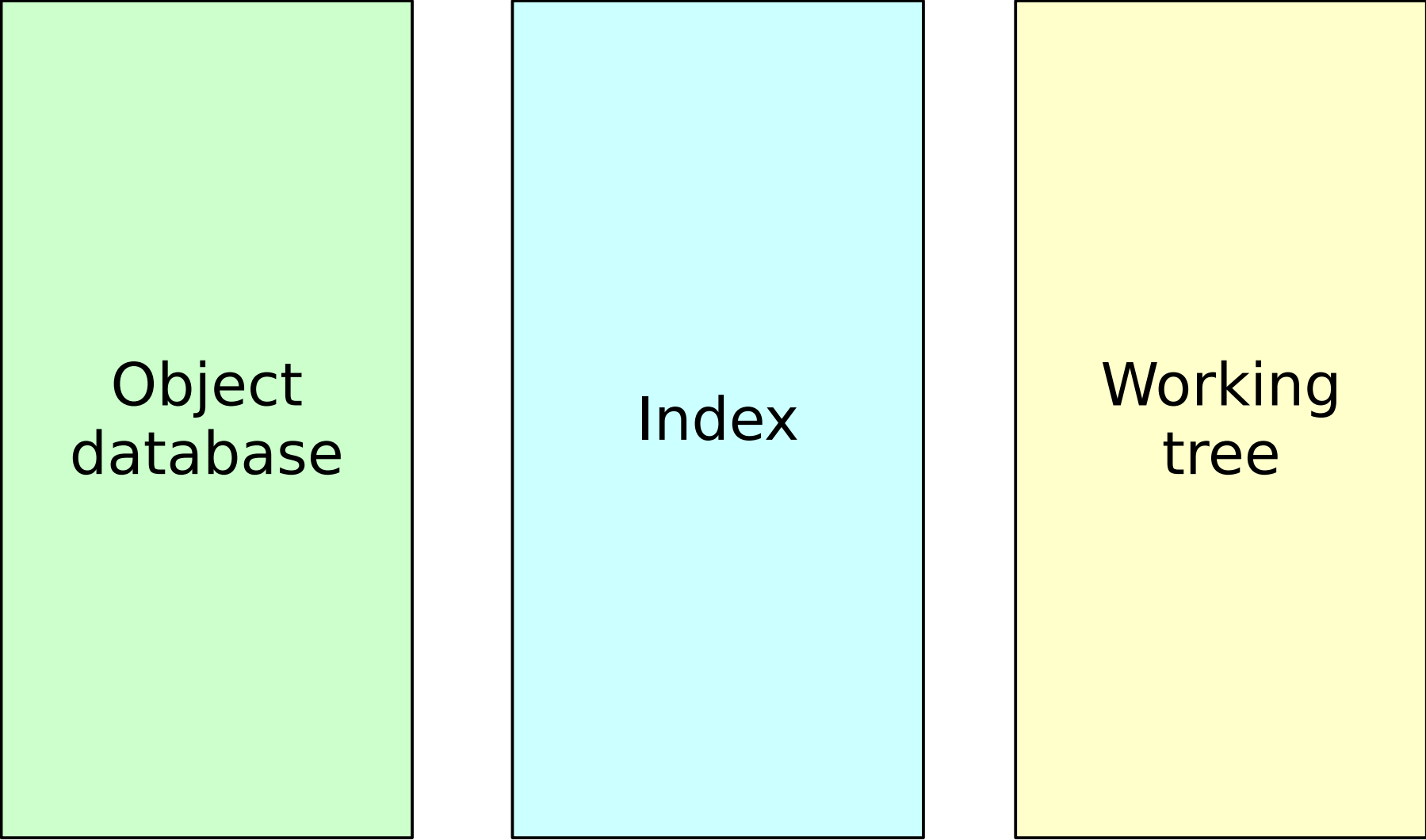
## Git - the index

Jiří Jabůrek (jjaburek)  
Red Hat 2016



# Three Trees of Git

# Three trees of git



The diagram consists of three vertical rectangular boxes arranged horizontally. The leftmost box is light green and contains the text 'Object database'. The middle box is light blue and contains the text 'Index'. The rightmost box is light yellow and contains the text 'Working tree'.

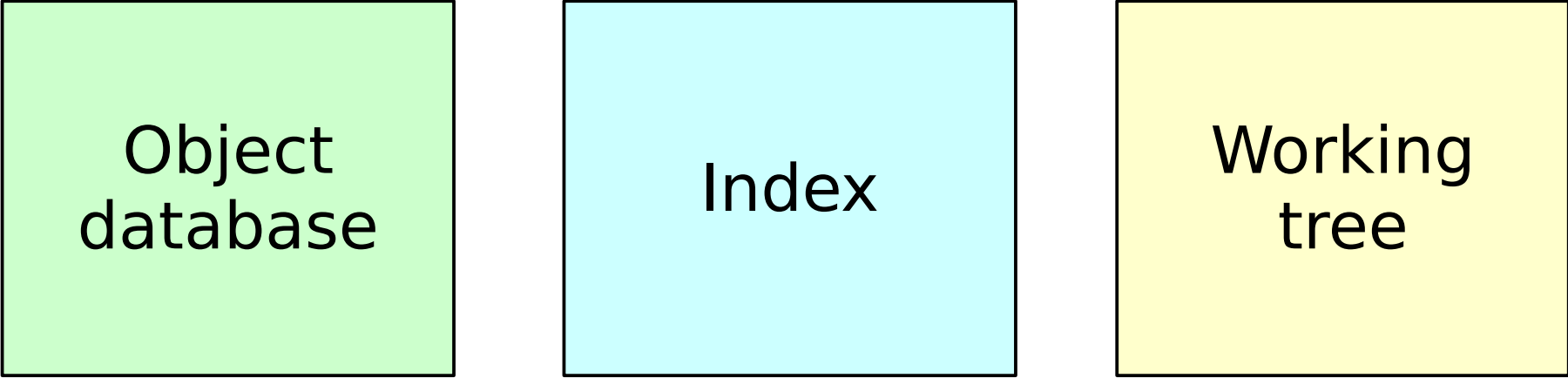
Object  
database

Index

Working  
tree

# Tracked vs Untracked Files

- Unknown file in working tree: untracked
- File in WT matching index: tracked
- My .gitignore is broken!
  - prevents only untracked from becoming tracked



Object  
database

Index

Working  
tree

# The Index (file)

- Bare vs "normal" repositories
  - without/with index+WT
  - metadata directly in dir vs in `.git`

HEAD  
branches/  
config  
description

hooks/  
index  
info/  
logs/

objects/  
packed-refs  
refs/

Object  
database

Index

Working  
tree



# The Index

(as a staging area)

# Staging area

\$

Data  
base

Index

Working  
tree

# Staging area

```
$ echo 1L > water.txt  
$ echo 10g > salt.txt
```

Data  
base

Index

Working  
tree

1L  
water.txt

10g  
salt.txt



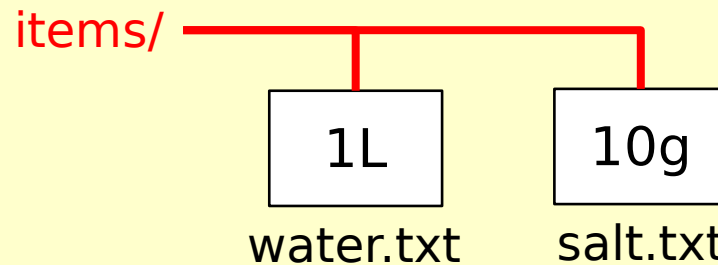
# Staging area

```
$ mkdir items  
$ mv water.txt salt.txt items/
```

Data  
base

Index

Working  
tree



# Staging area

```
$ cat > steps.txt <<EOF  
1. throw in  
2. boil  
EOF
```

Data  
base

Index

Working  
tree

items/

1L

water.txt

10g

salt.txt

1. throw in  
2. boil

steps.txt

# Staging area

\$

Data  
base

Index

Working  
tree

items/

1L

water.txt

10g

salt.txt

1. throw in  
2. boil

steps.txt

# Staging area

```
$ git add steps.txt
```

Data  
base

0672229  
"1. th.."

Index

100644 0672229bb0c392a550290f5b6aafee63469632e8 0 steps.txt

Working  
tree

items/

1L

water.txt

10g

salt.txt

1. throw in  
2. boil

steps.txt

# Staging area

```
$ git add items/
```

Data  
base

0672229  
"1. th.."

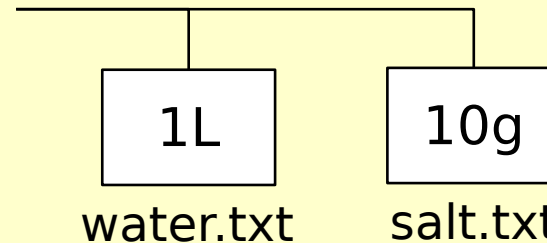
7379a1e  
"1L"  
84f62fb  
"10g"

Index

```
100644 84f62fb07e29ca5ddc8caf3c0b332cd0cc6da1d2 0 items/salt.txt
100644 7379a1e0549f294d33f91e16dcfd7d1721f44736 0 items/water.txt
100644 0672229bb0c392a550290f5b6aafee63469632e8 0 steps.txt
```

Working  
tree

items/



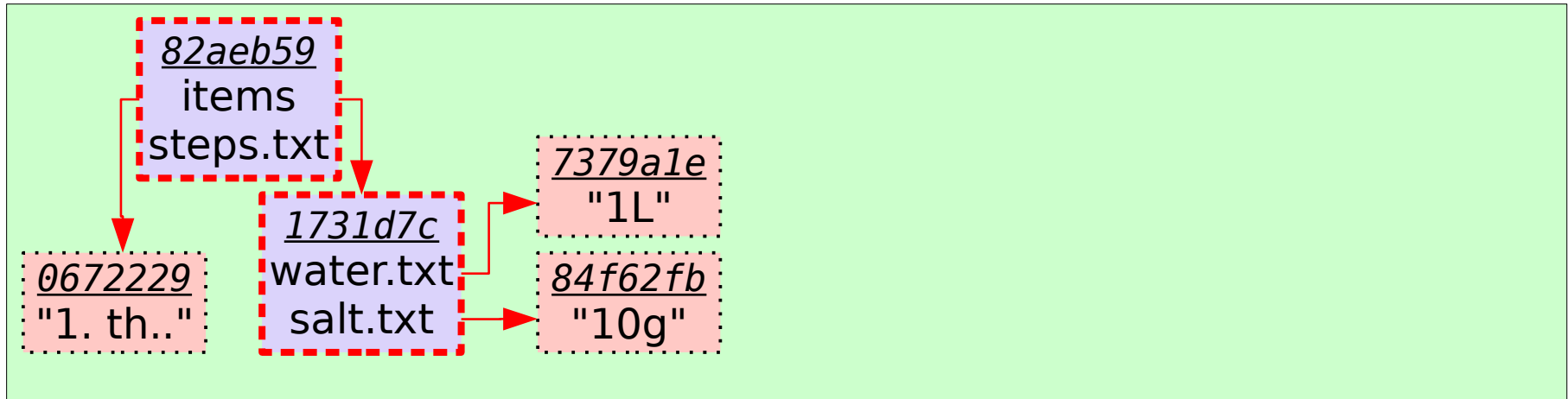
1. throw in  
2. boil

steps.txt

# Staging area

```
$ git write-tree  
82aeb59...
```

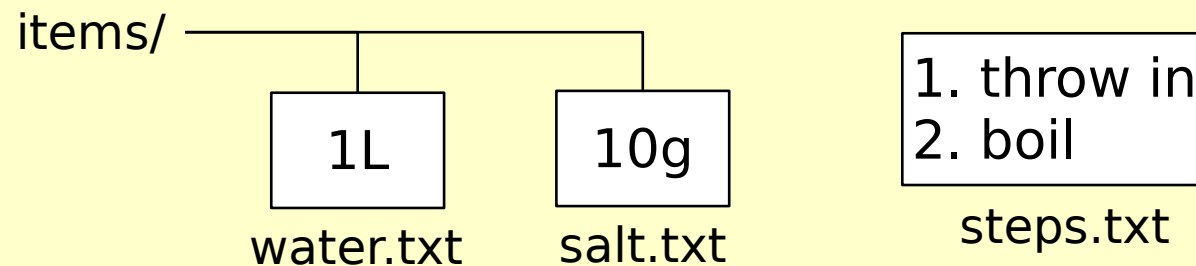
Data  
base



Index

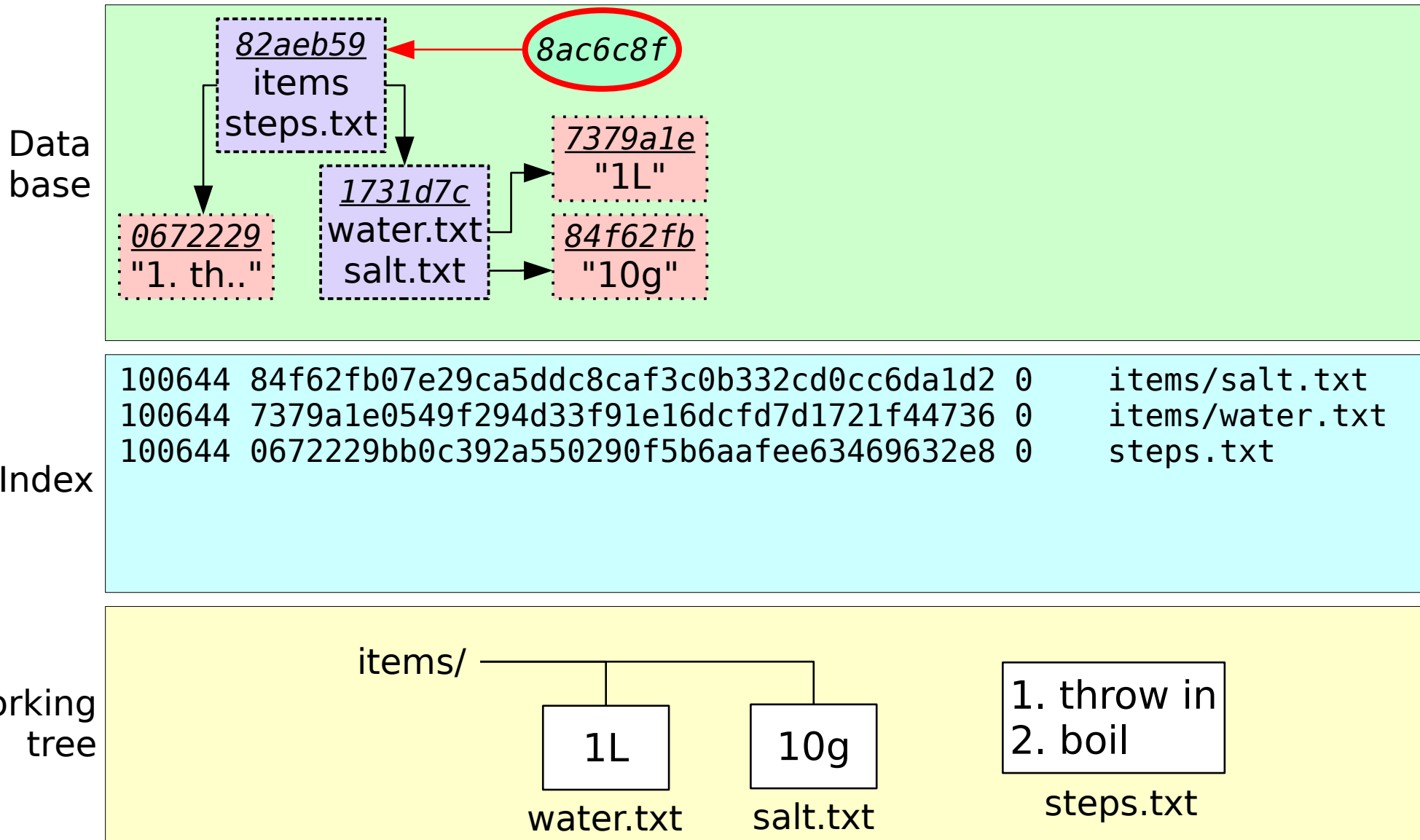
100644	84f62fb07e29ca5ddc8caf3c0b332cd0cc6da1d2	0	items/salt.txt
100644	7379a1e0549f294d33f91e16dcfd7d1721f44736	0	items/water.txt
100644	0672229bb0c392a550290f5b6aafee63469632e8	0	steps.txt

Working  
tree



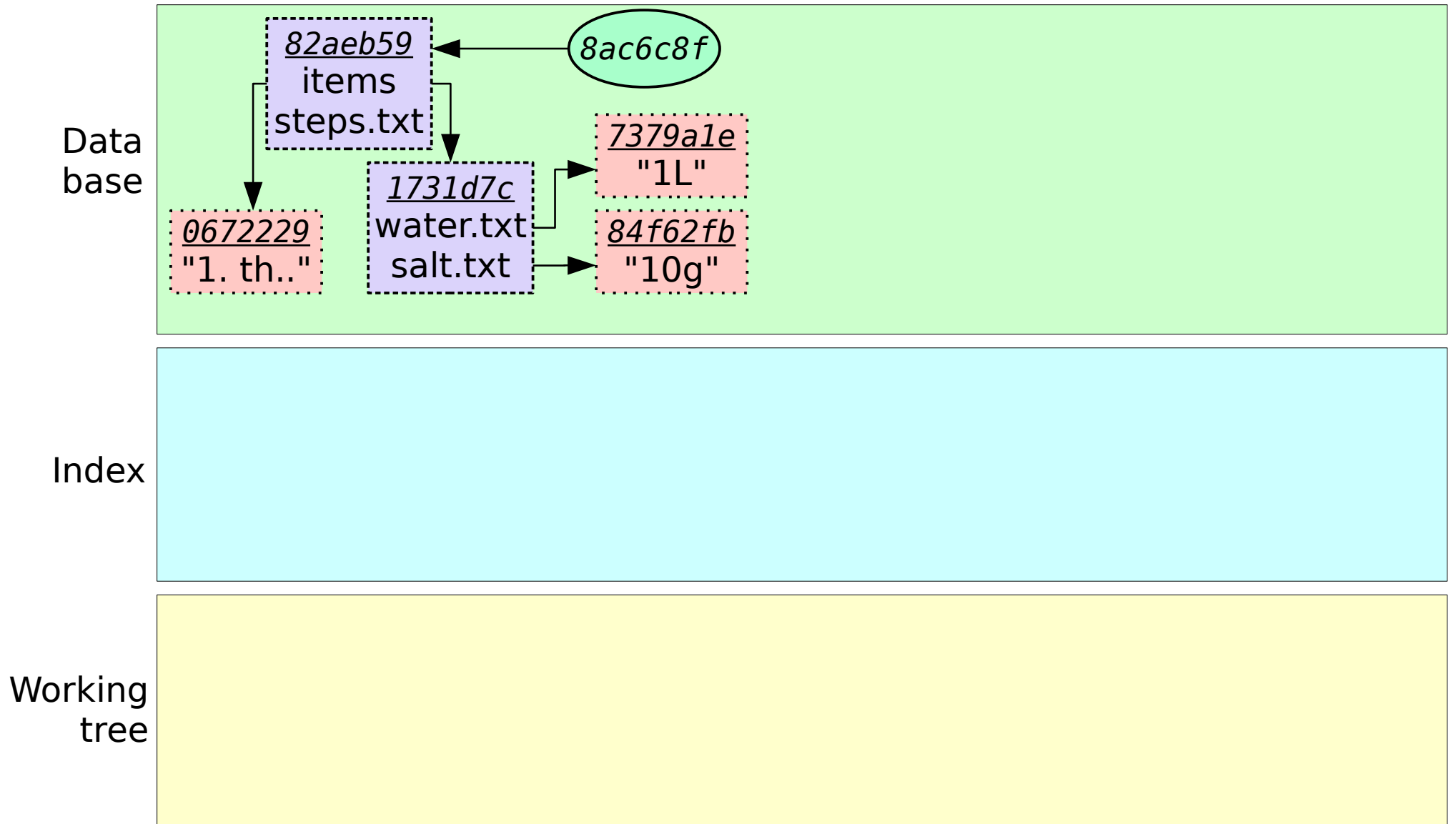
# Staging area

```
$ git commit-tree 82aeb59...  
added hot salt water recipe  
^D  
8ac6c8f...
```



# Staging area

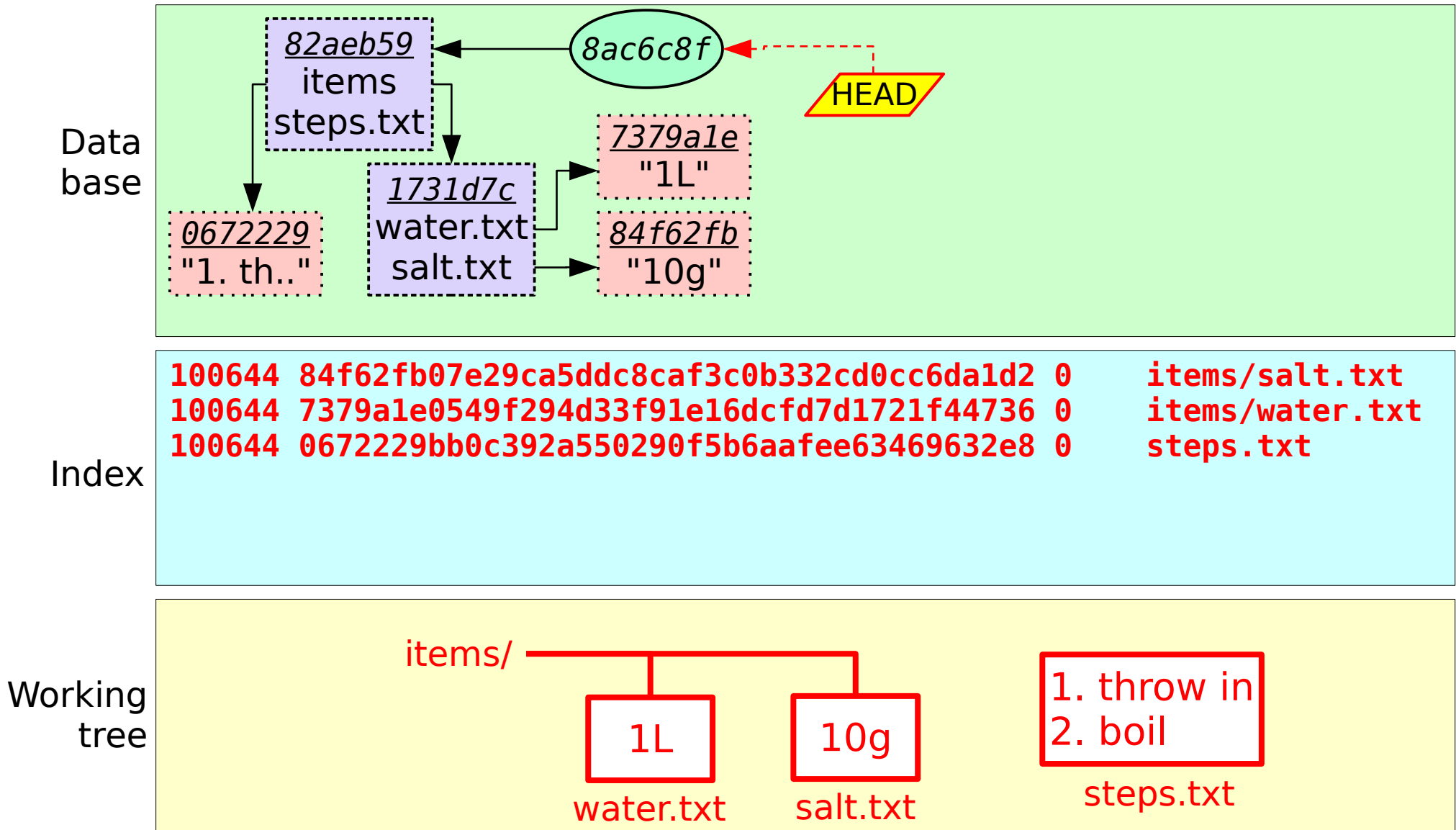
```
$ rm -f .git/index  
$ rm -rf *
```





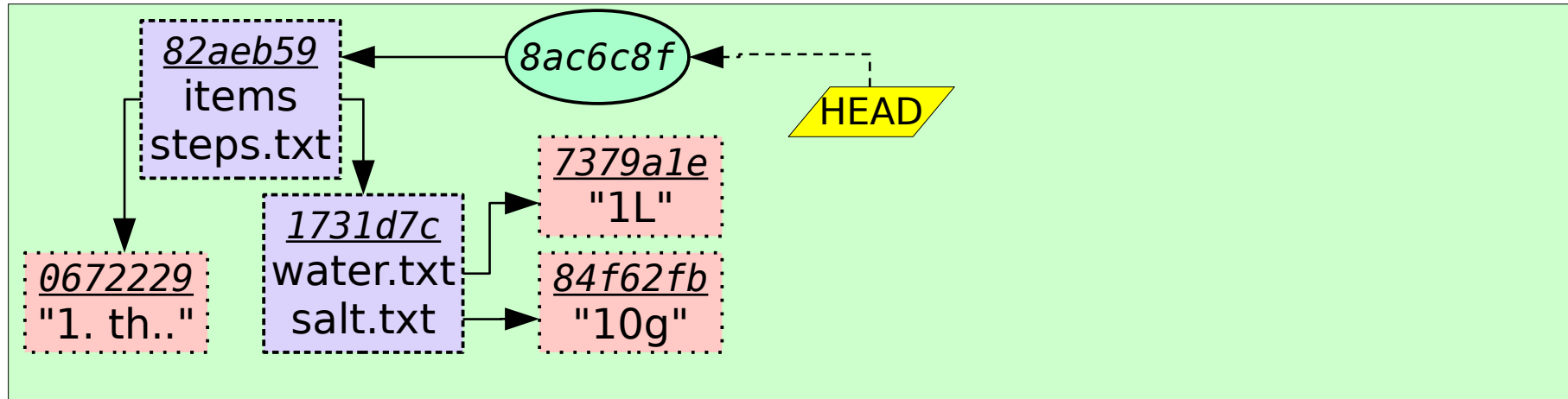
# Staging area

```
$ git checkout -f 8ac6c8f...  
HEAD is now at 8ac6c8f...
```



# Staging area

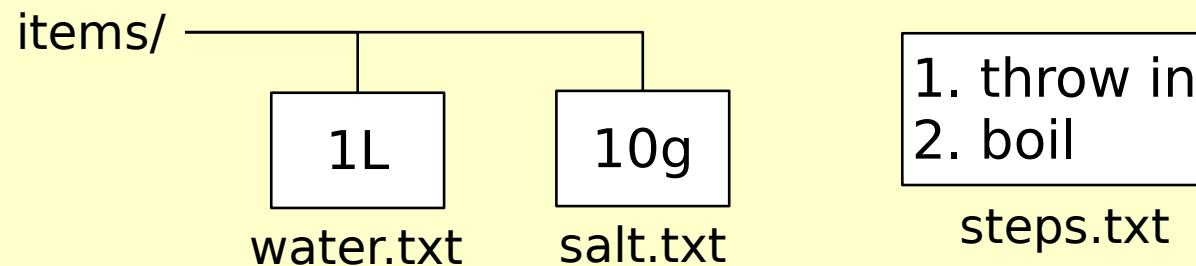
Data  
base



Index

100644	84f62fb07e29ca5ddc8caf3c0b332cd0cc6da1d2	0	items/salt.txt
100644	7379a1e0549f294d33f91e16dcfd7d1721f44736	0	items/water.txt
100644	0672229bb0c392a550290f5b6aafee63469632e8	0	steps.txt

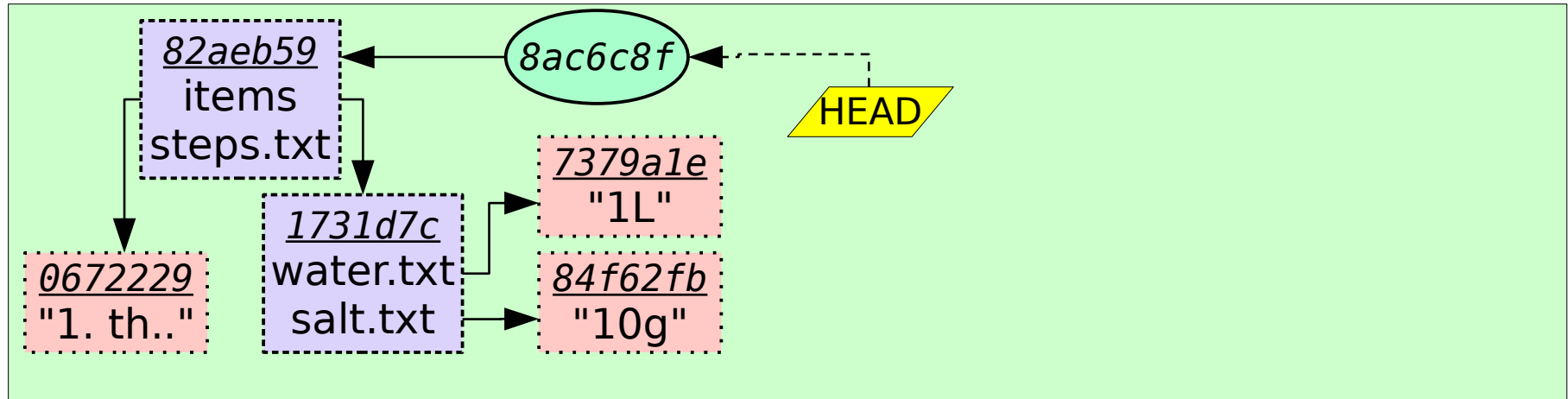
Working  
tree



# Staging area

```
$ echo 2g > items/salt.txt  
$ mv steps.txt guide.txt
```

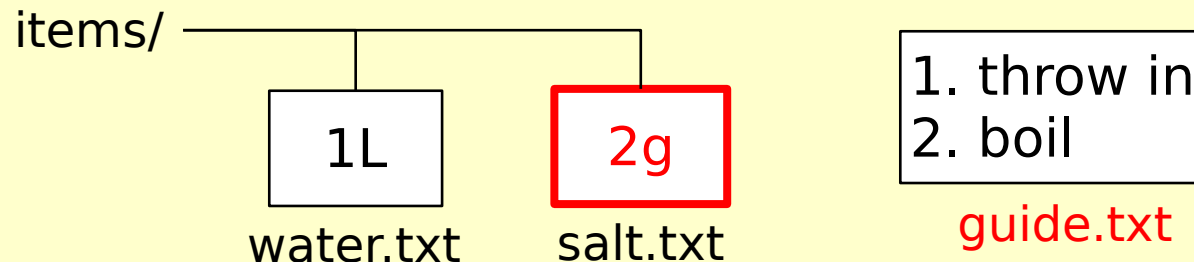
Data  
base



Index

100644	84f62fb07e29ca5ddc8caf3c0b332cd0cc6da1d2	0	items/salt.txt
100644	7379a1e0549f294d33f91e16dcfd7d1721f44736	0	items/water.txt
100644	0672229bb0c392a550290f5b6aafee63469632e8	0	steps.txt

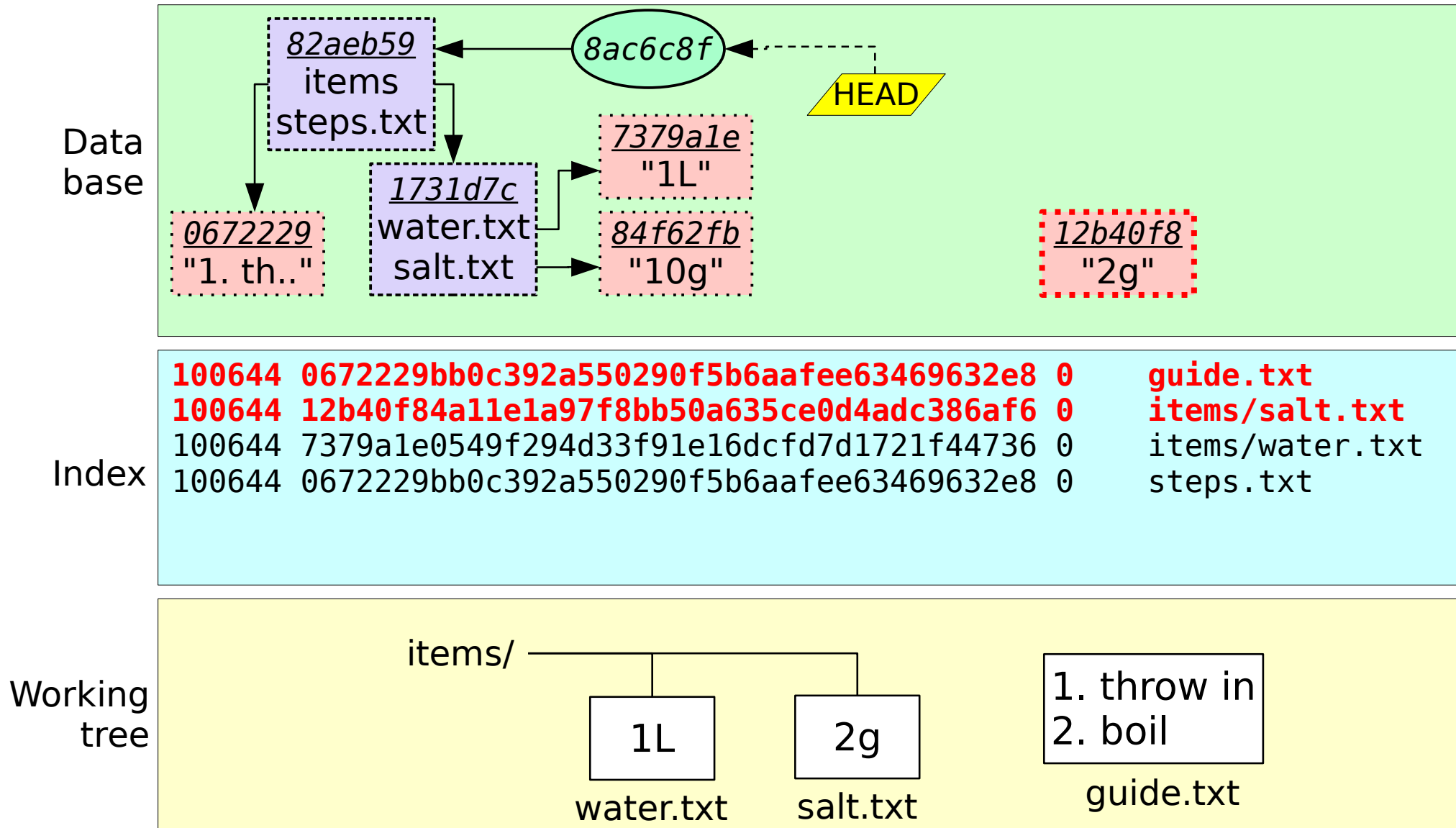
Working  
tree



(pre-2.0 git)

```
$ git add .
```

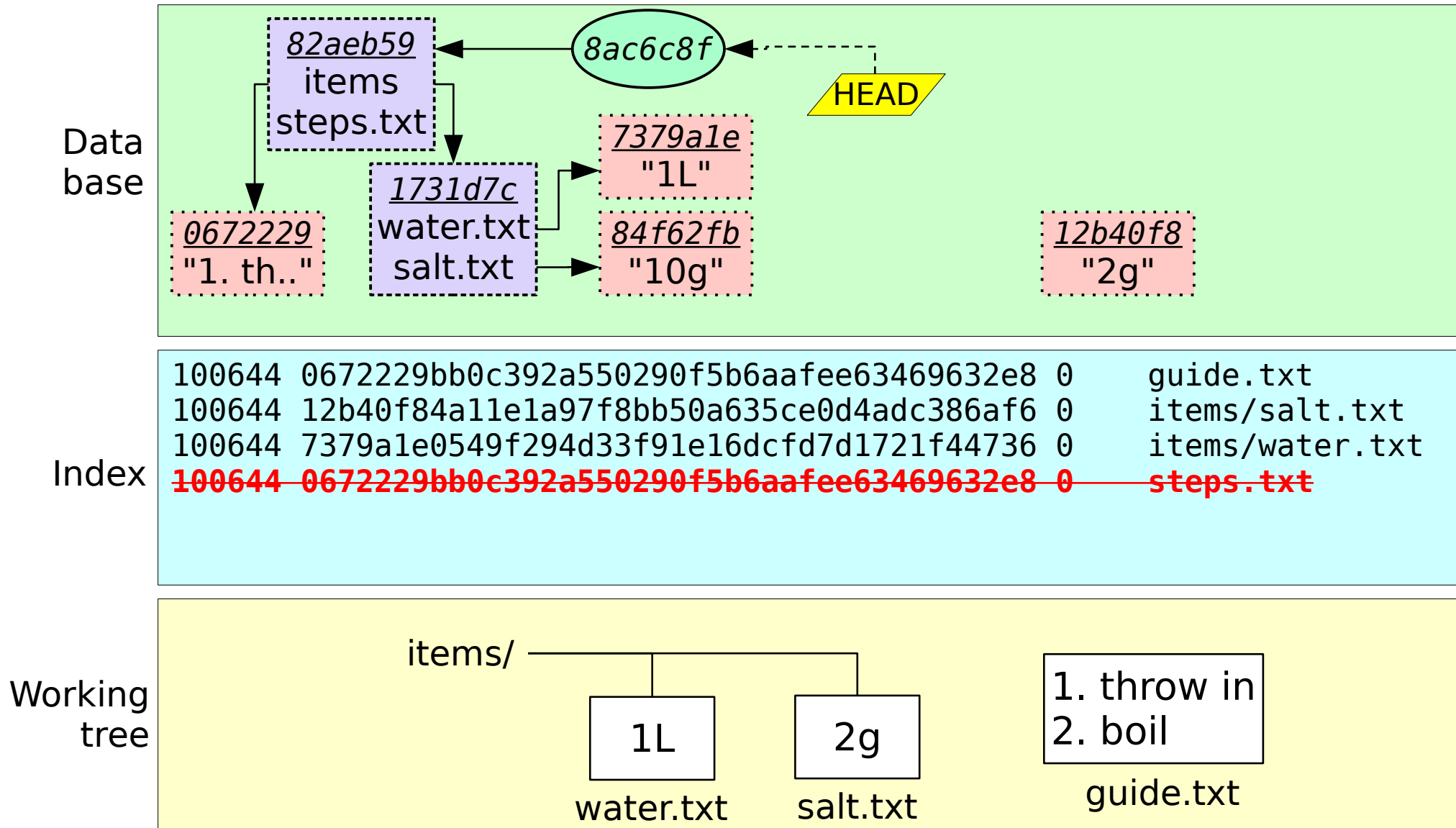
# Staging area



(pre-2.0 git)

```
$ git add -u
```

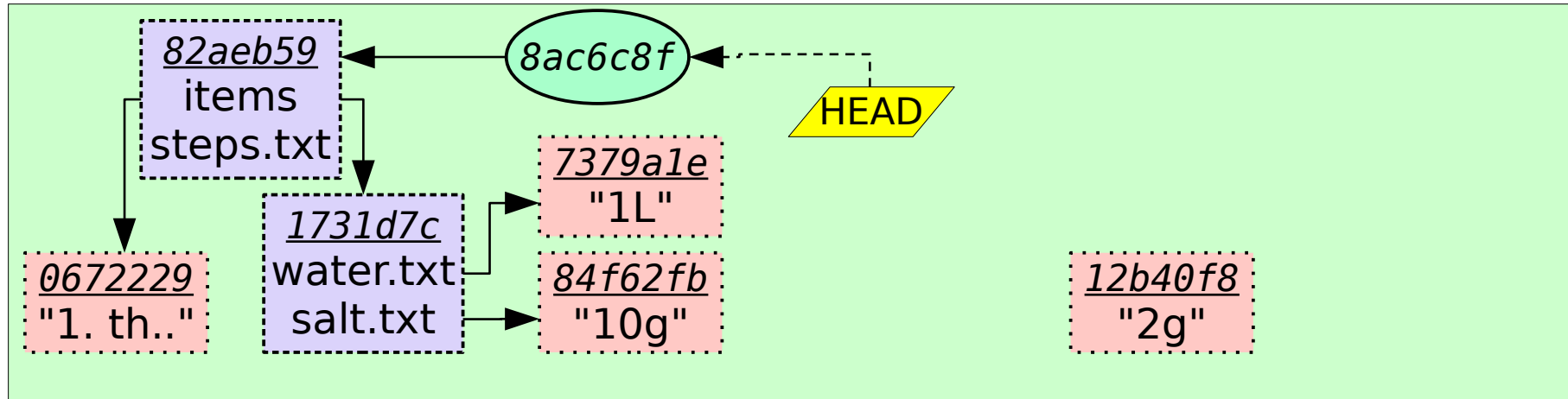
# Staging area



\$

# Staging area

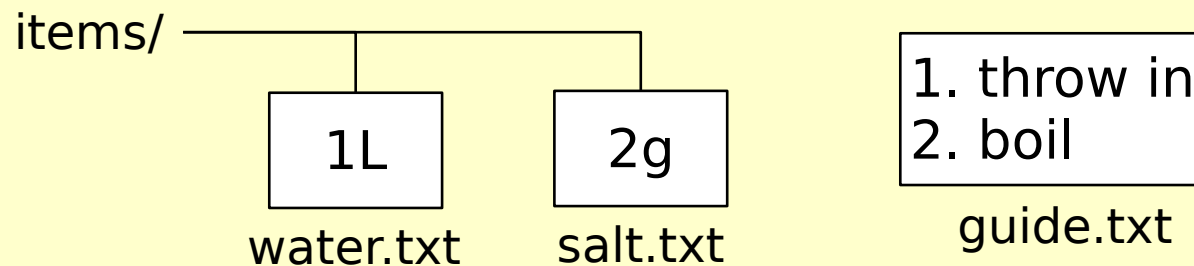
Data  
base



Index

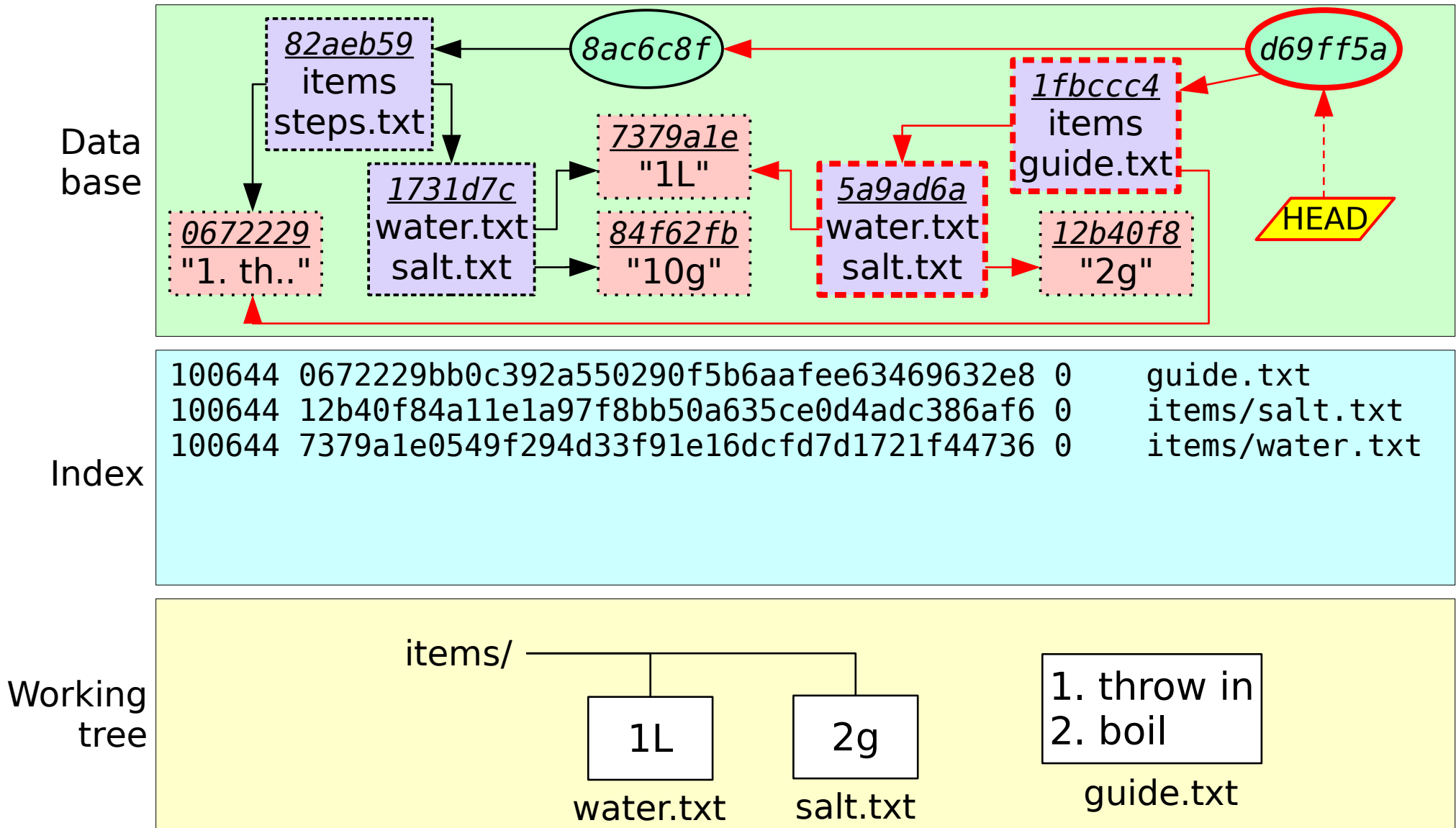
100644	0672229bb0c392a550290f5b6aafee63469632e8	0	guide.txt
100644	12b40f84a11e1a97f8bb50a635ce0d4adc386af6	0	items/salt.txt
100644	7379a1e0549f294d33f91e16dcfd7d1721f44736	0	items/water.txt

Working  
tree

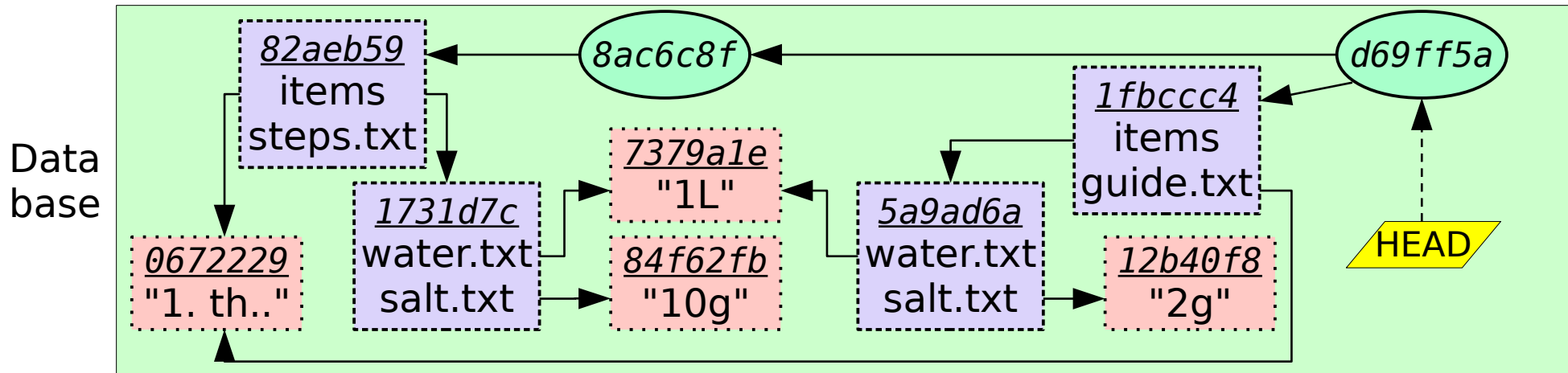


# Staging area

```
$ git commit \  
    -m 'refined the recipe'  
[detached HEAD d69ff5a] ...
```



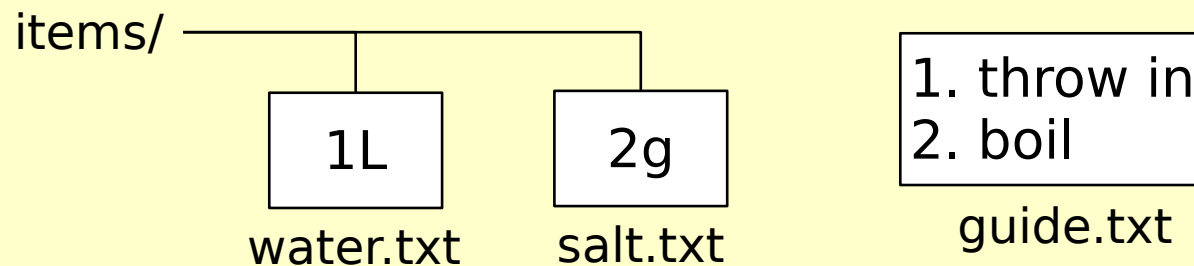
# Staging area



# Index

```
100644 0672229bb0c392a550290f5b6aaf6e63469632e8 0 guide.txt
100644 12b40f84a11e1a97f8bb50a635ce0d4adc386af6 0 items/salt.txt
100644 7379a1e0549f294d33f91e16dcfd7d1721f44736 0 items/water.txt
```

# Working tree





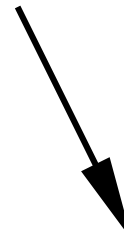


# The Index

(as a merge storage)

# Merge "storage"

stage nr.

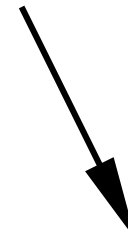


Index

100644	0672229bb0c392a550290f5b6aaf6e63469632e8	0	guide.txt
100644	12b40f84a11e1a97f8bb50a635ce0d4adc386af6	0	items/salt.txt
100644	7379a1e0549f294d33f91e16dcfd7d1721f44736	0	items/water.txt

# Merge "storage"

stage nr.    0 = normal  
              1 = base  
              2 = ours  
              3 = theirs

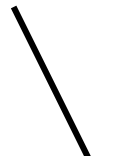


Index

100644	0672229bb0c392a550290f5b6aaf6e63469632e8	0	guide.txt
100644	12b40f84a11e1a97f8bb50a635ce0d4adc386af6	0	items/salt.txt
100644	7379a1e0549f294d33f91e16dcfd7d1721f44736	0	items/water.txt

# Merge "storage"

stage nr.    0 = normal  
                 1 = base  
                 2 = ours  
                 3 = theirs

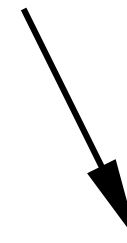


Index

100644	0672229bb0c392a550290f5b6aaf6e63469632e8	0	guide.txt
<b>100644</b>	<b>e8f078cb2339276ef315d05a53bb0ae734ec93b8</b>	<b>1</b>	<b>items/salt.txt</b>
<b>100644</b>	<b>f59dd882f6ec220433c656341880e4e94c32cefc</b>	<b>2</b>	<b>items/salt.txt</b>
<b>100644</b>	<b>db506164cb09d20e2c709d179ed1aa8169e2c11c</b>	<b>3</b>	<b>items/salt.txt</b>
100644	7379a1e0549f294d33f91e16dcfd7d1721f44736	0	items/water.txt

# Merge "storage"

stage nr.    0 = normal  
                 1 = base  
                 2 = ours  
                 3 = theirs



Index

100644	0672229bb0c392a550290f5b6aaf6e63469632e8	0	guide.txt
<b>100644</b>	<b>12b40f84a11e1a97f8bb50a635ce0d4adc386af6</b>	<b>0</b>	<b>items/salt.txt</b>
100644	7379a1e0549f294d33f91e16dcfd7d1721f44736	0	items/water.txt



# The Index

(as a directory cache)

# The Index

- header:
  - 4B: "DIRC"
  - 4B: version (2,3,4)
  - 4B: nr. of index entries
- idx entry:
  - 4B: ctime (seconds) + 4B: ctime (nanoseconds)
  - 4B: mtime (seconds) + 4B: mtime (nanoseconds)
  - 4B: dev id
  - 4B: inode nr.
  - 4B: filemode
  - 4B: uid + 4b: gid
  - 4B: file size
  - 20B: SHA1 hash
  - 2B: flags (incl. 2-bit stage nr.)
  - var: pathname as string
  - 1-8B: padding
- extensions

# The Index

- Extensions
  - Cache tree
  - Resolve undo (`git checkout -m`)
  - Split index
  - Untracked files cache
  - ... ?



# The Index

- Extra (user-facing) features
  - Split index
    - `git update-index --split-index`
  - Untracked file cache
    - `git update-index --untracked-cache`
    - as conf: `core.untrackedCache`
  - "Assume unchanged"
    - `git update-index --assume-unchanged`
    - as conf: `core.ignoreStat`
  - "Skip worktree"
    - `git update-index --skip-worktree`

# The Index

## ■ Sparse checkout

- utilizes "skip worktree" bit
- enable: `core.sparseCheckout`
- set: `echo "some/path" \`  
`> .git/info/sparse-checkout`
- add remote / pull / checkout / ...

# The Index

- Moar features!

- "Intent to add"

- `git add -N`

- Add/update all

- `git add -A [path]`

- (default in v1.8.3+)

- (without path, uses "." on pre-v2.0, entire WT on v2.0+)



# Summary

# Summary

- Repository: bare vs non-bare
  - bare: without working tree, without idx, for ie. hosting
  - non-bare: with working tree, with index, for devel work
- Staging area
  - Used to prepare changes for a commit
    - stage files, diff sections or individual lines!
  - Creates just blobs on the fly
- Directory cache
  - Speeds up status / diff / etc. by caching lstat(2)
- Has additional features
  - sparse checkout, assume-unchanged, etc.



# Questions?

# The End

Thanks for listening



Workshop >>

# Workshop

**Goal:** Create a commit using index, with 2 files 2 directory levels deep.

**Setup:** `echo first > /tmp/first`  
`echo second > /tmp/second`

## 1) Initialize a new repository

- `git init [reponame]`

## 2) Hash content

- `git hash-object -w`

## 3) Populate index

- `git update-index --cacheinfo or --index-info`
  - mode = UNIX mode, see man 2 stat, look for st\_mode
  - check with `git ls-files --stage`

## 4) Create root tree (incl. all subtrees) from index

- `git write-tree`

## 5) Commit the tree

- `git commit-tree`
  - enter commit message (multiline), hit Control-D

## 6) Checkout the commit

- `git checkout -f <sha>`

## 7) See result

- `git log --stat --format=raw ; gitk`
- `ls -R ; tree -C`