GREENPLUM DATABASE®

# 揭秘Greenplum存储引擎之Heap表

钉钉直播 | 11月18日 20：00 - 21：00

GREENPLUM DATABASE®

# Greenplum中文社区

https://cn.greenplum.org

博文 · 资料 · 文档 · 项目

# 全新的问答论坛

https://cn.greenplum.org/askgp

# GREENPLUM DATABASE®

微信技术讨论群
添加入群小助手：gp_assistant

微信公众号
技术干货、行业热点、活动预告

欢迎访问Greenplum中文社区：cn.greenplum.org

# Greenplum存储引擎之Heap表

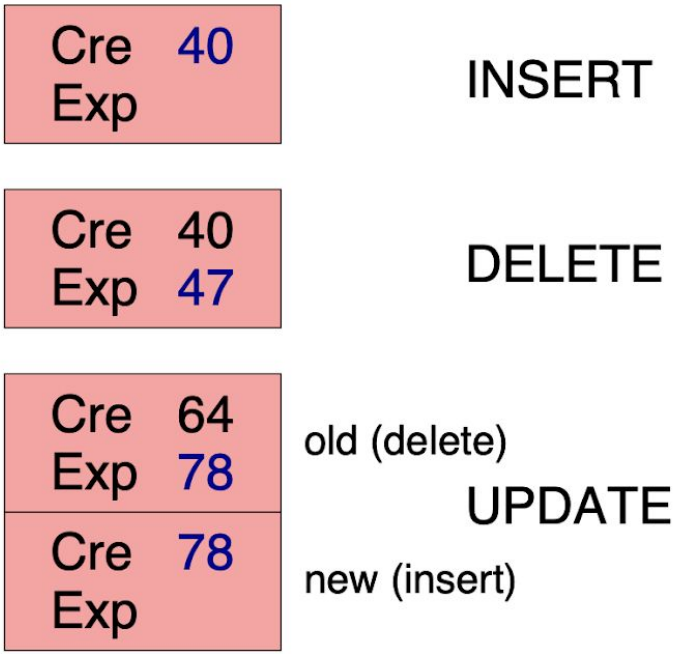
# Agenda

GREENPLUM DATABASE

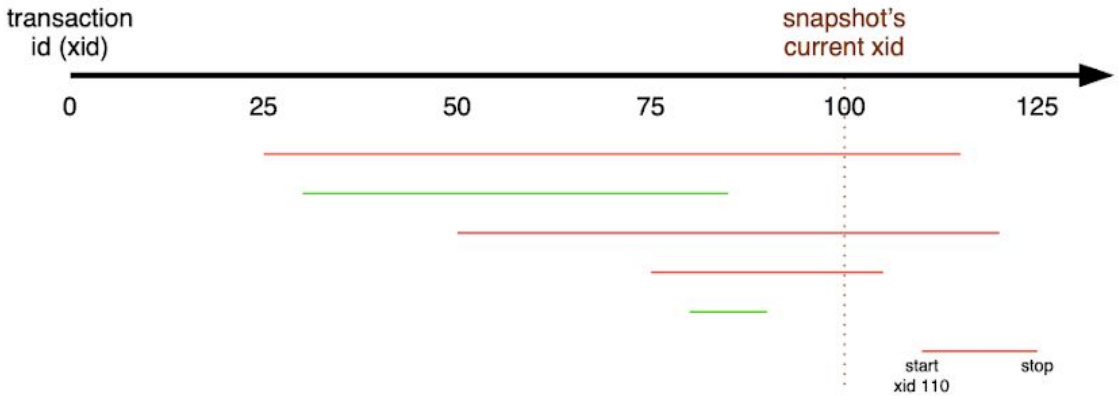
- 
- MVCC机制
- 文件结构
- 页面结构
- 元组结构
- 共享缓冲区管理
- 空闲空间管理
- 页可见性管理
- 元组加锁及LRU管理
- AO/AOCO表

# MVCC机制

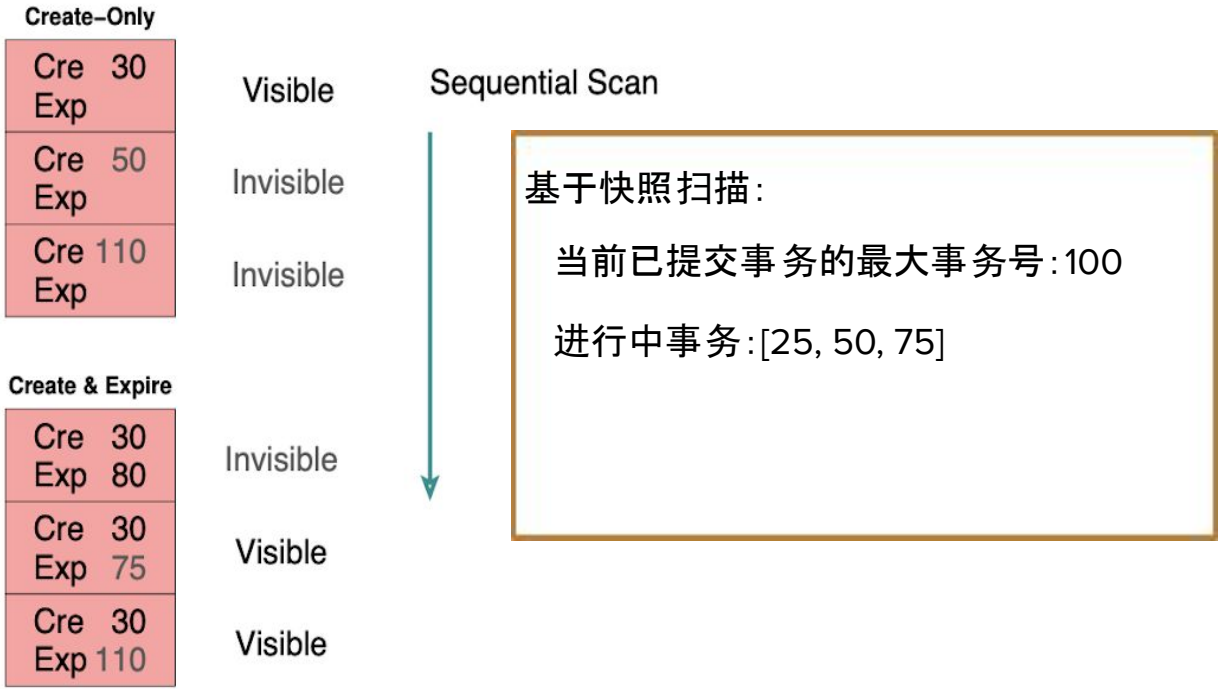
# MVCC: INSERT, DELETE, UPDATE



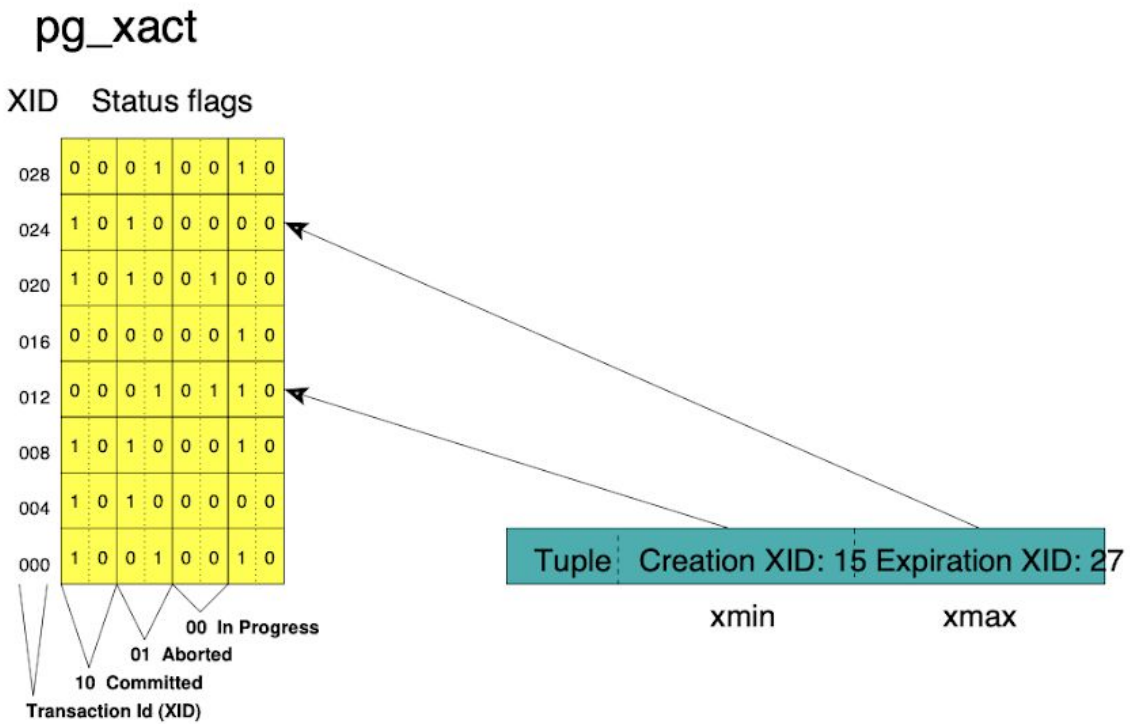
# 事务间可见性


transaction
id (xid)
snapshot's
current xid
0
25
50
75
100
125
start
xid 110
stop

# 数据可见性判断

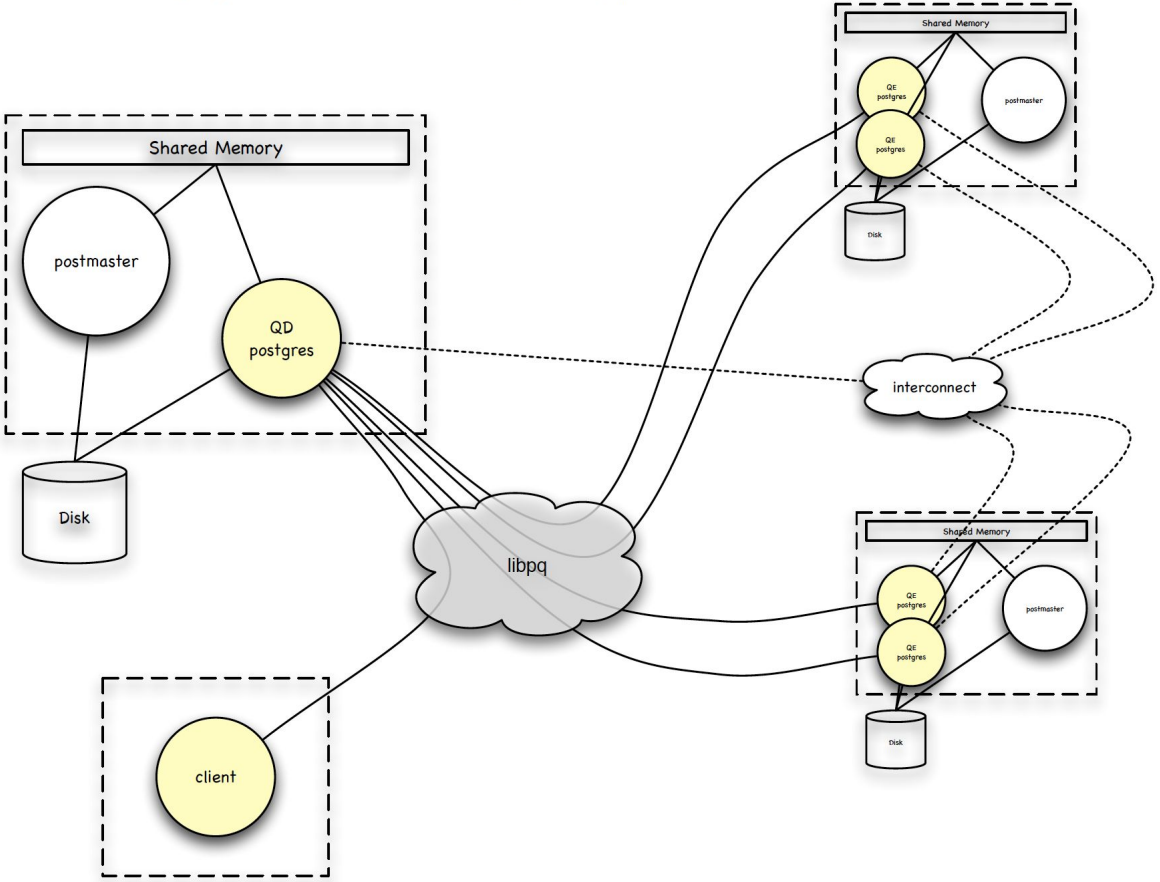

Create-Only
Cre 30
Exp
Cre 50
Exp
Cre 110
Exp
Visible
Invisible
Invisible
Create & Expire
Cre 30
Exp 80
Cre 30
Exp 75
Cre 30
Exp 110
Invisible
Visible
Visible
Sequential Scan
基于快照扫描:
当前已提交事务的最大事务号:100
进行中事务:[25, 50, 75]

# 基于事务状态判断元组可见性

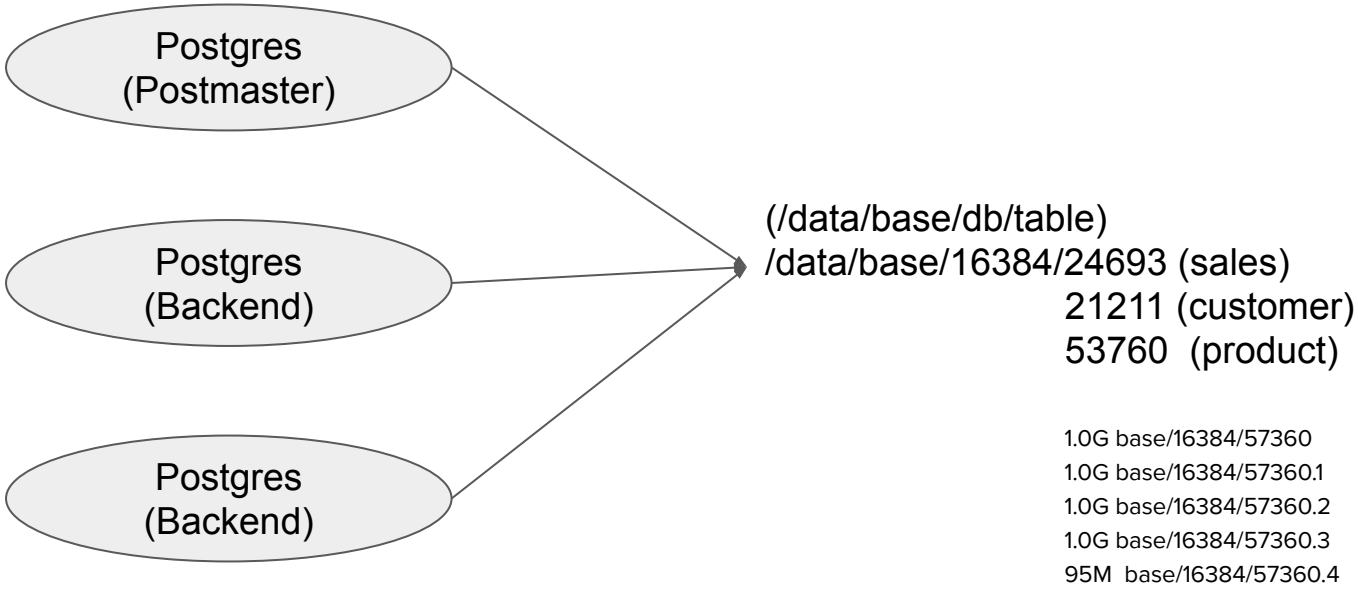


# 文件结构

# Greenplum并行处理架构

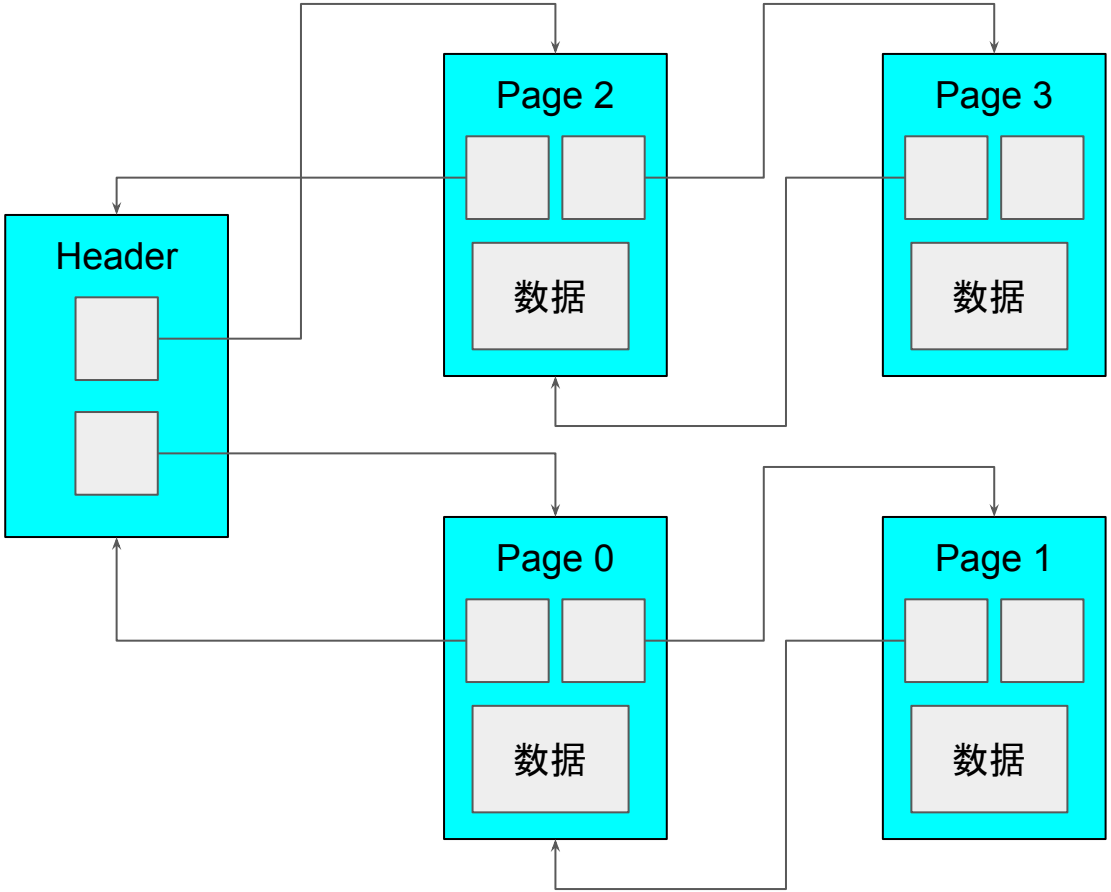


# Greenplum共享文件模型

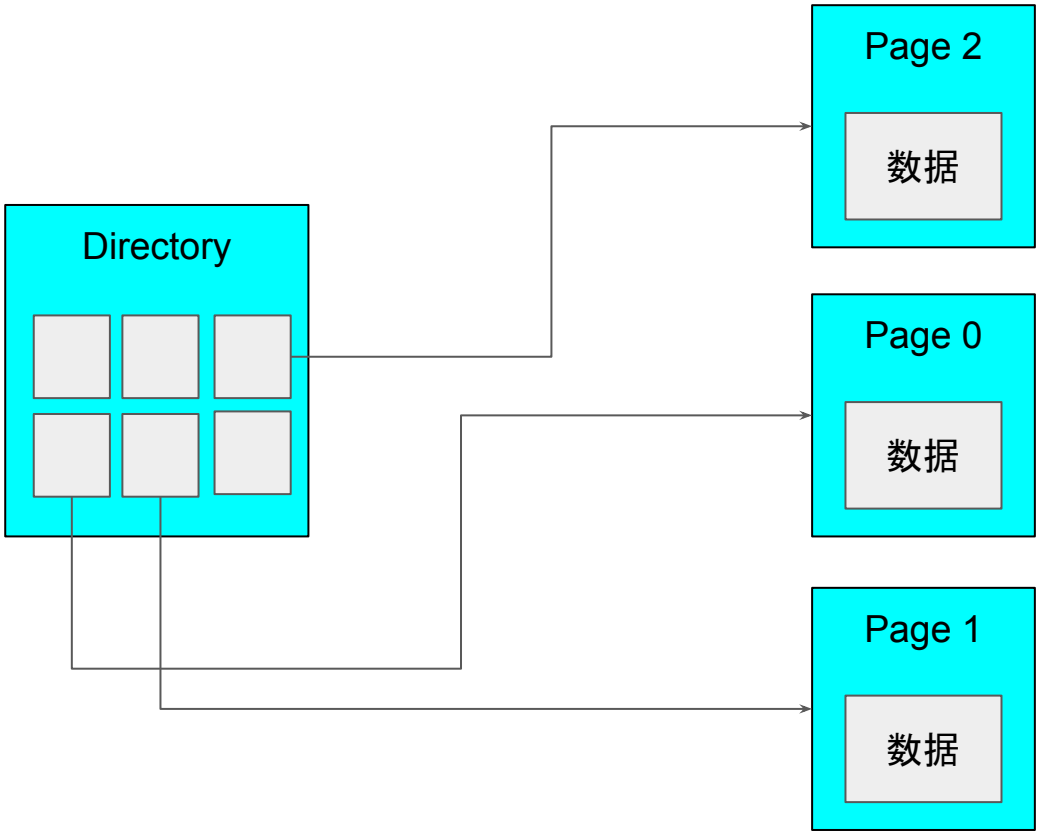


# 文件页组织:链表方式

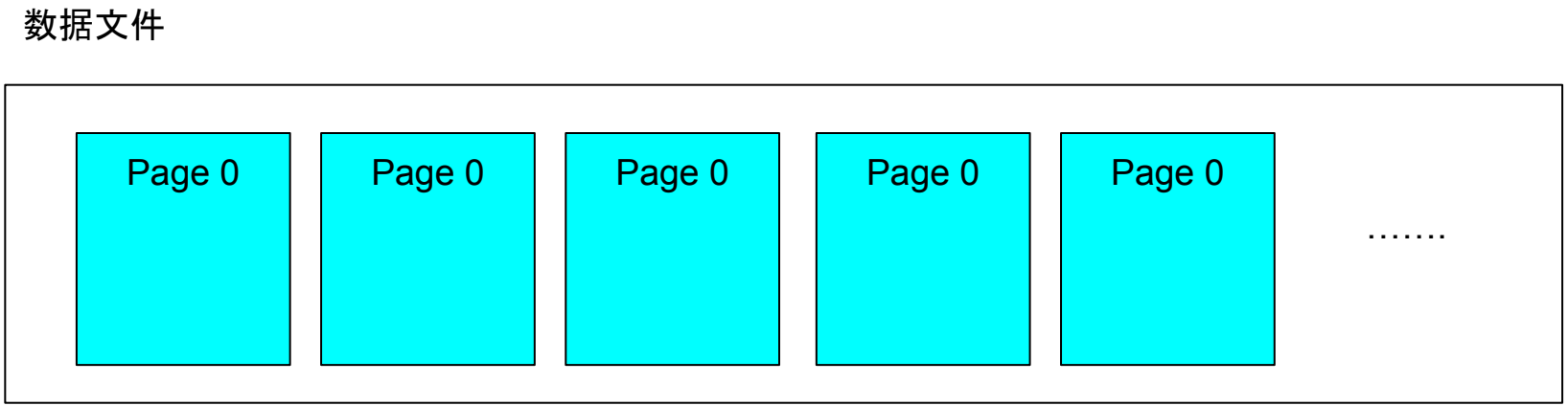
GREENPLUM DATABASE



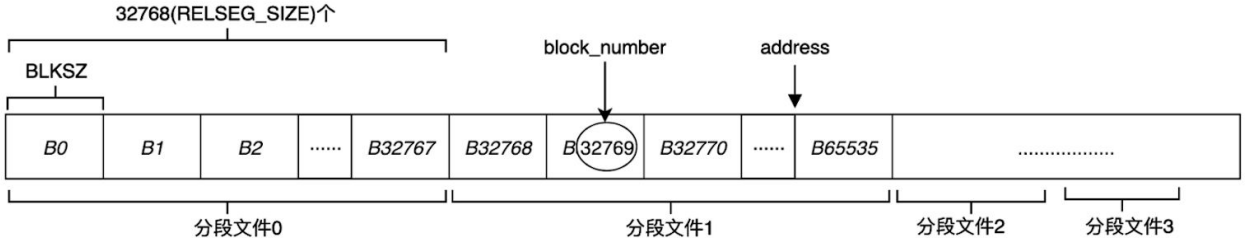
# 文件页组织:目录方式



# 文件页组织:Heap方式(Greenplum)



# Greenplum文件页寻址




相互之间的转换公式为

$$block_number = address \,/\, BLCKSZ$$

$$seg_number = address \,/\, (BLCKSZ * RELSEG_SIZE)$$

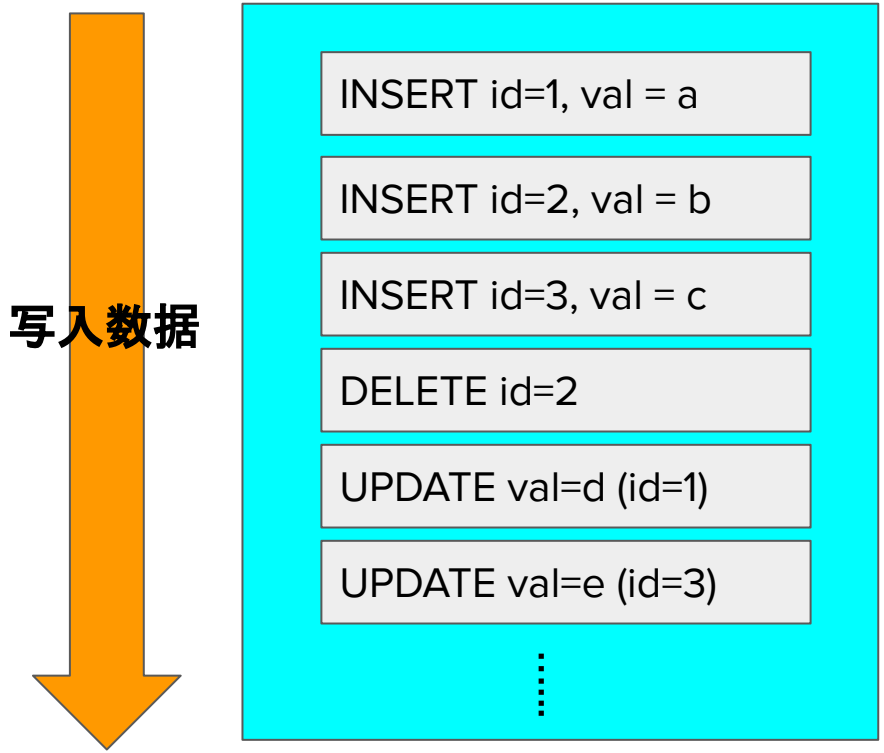
$$seg_number = block_number \,/\, RELSEG_SIZE$$

$$seg_offset = address \,\%\, (BLCKSZ * RELSEG_SIZE)$$

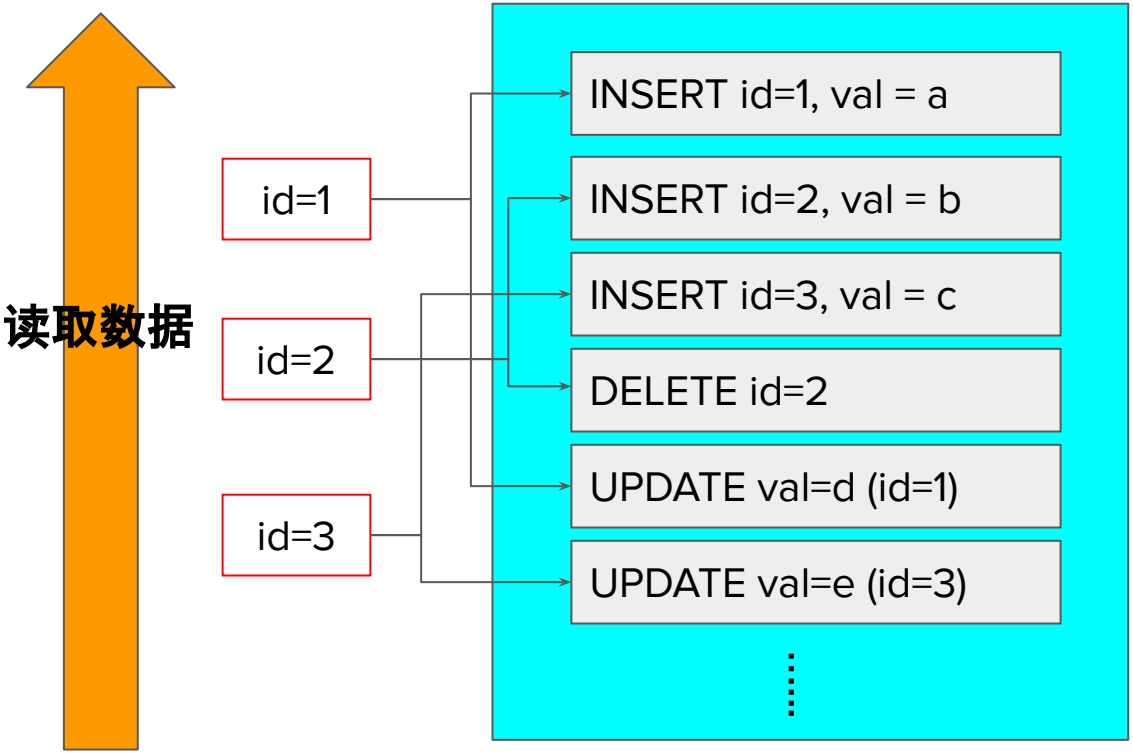
# 页面结构

# 基于日志存储数据

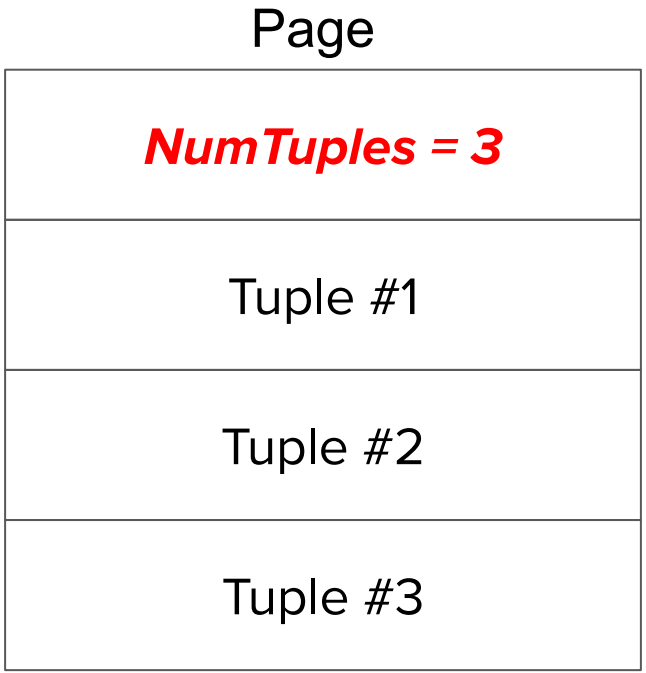
GREENPLUM DATABASE



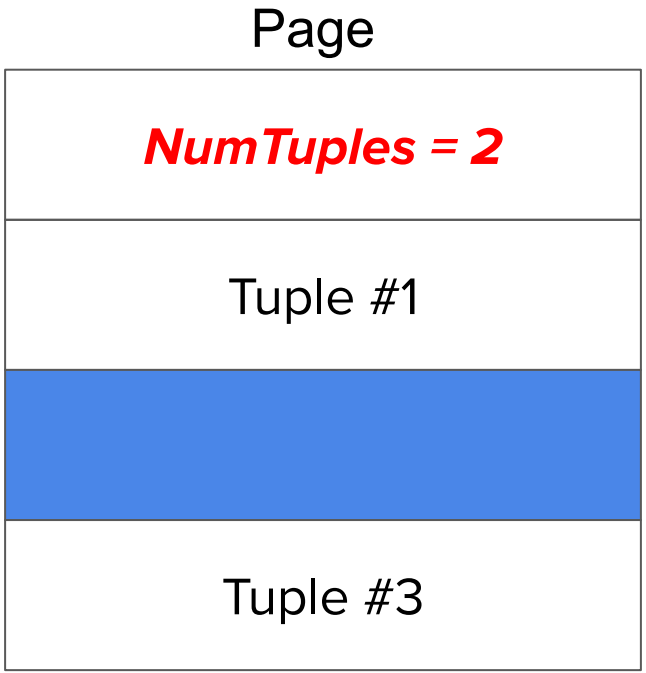
# 基于日志存储数据-续


读取数据
id=1
id=2
id=3
INSERT id=1, val = a
INSERT id=2, val = b
INSERT id=3, val = c
DELETE id=2
UPDATE val=d (id=1)
UPDATE val=e (id=3)

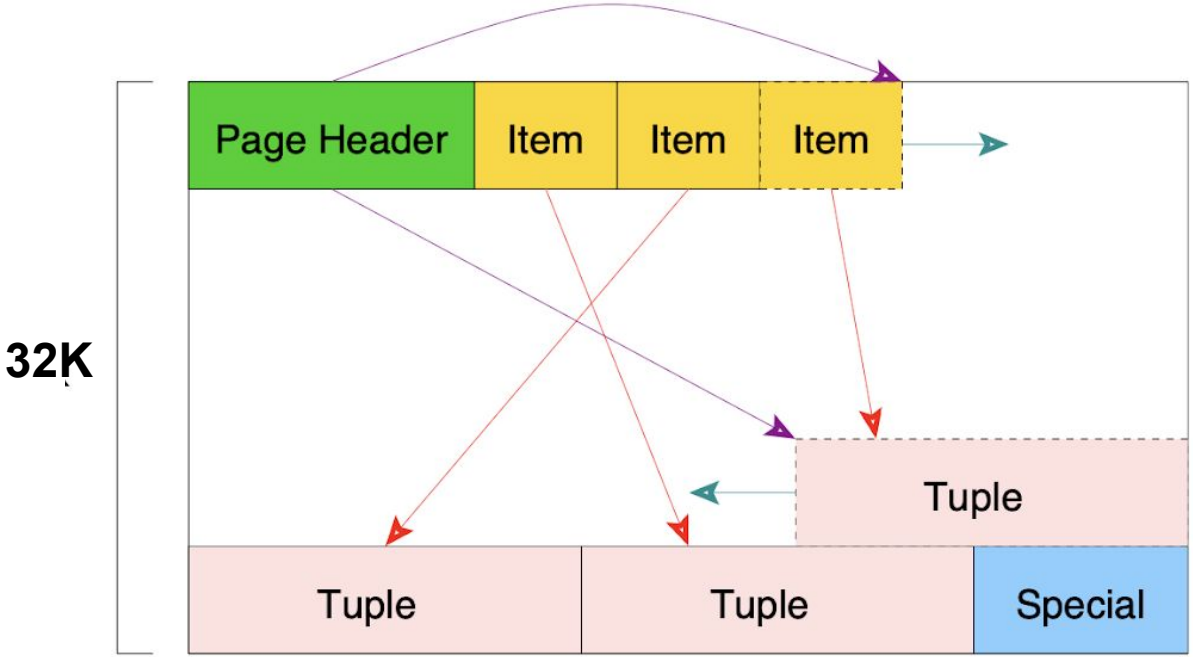
# 页面结构：朴素方式



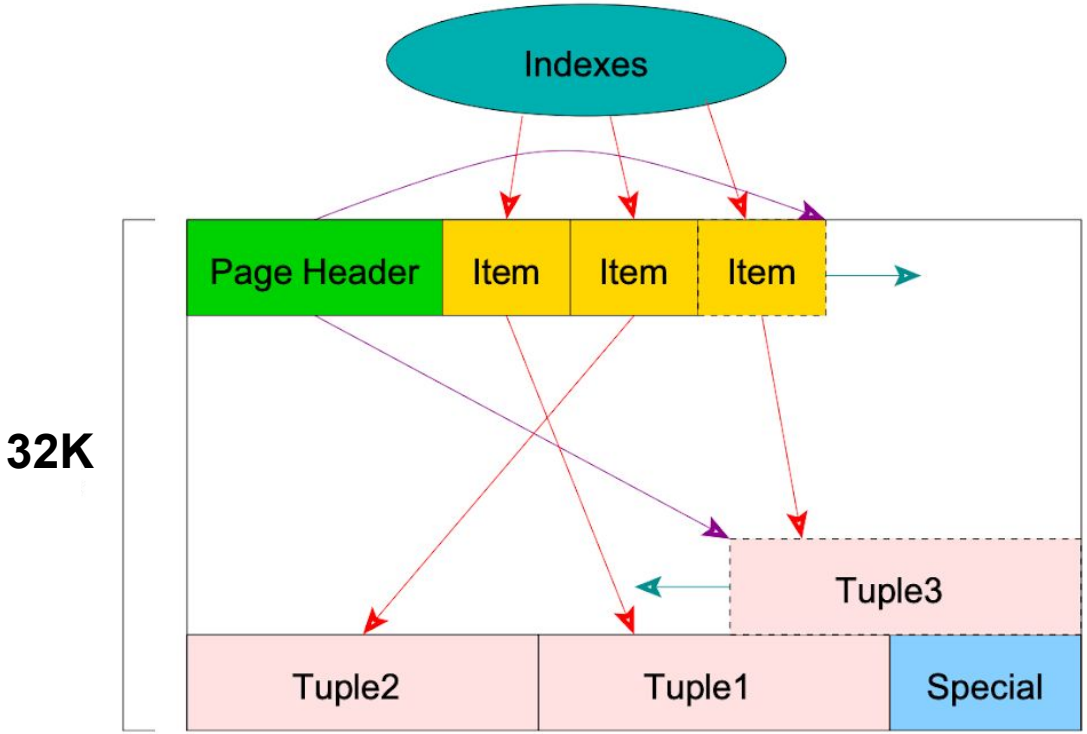
# 页面结构：朴素方式-续



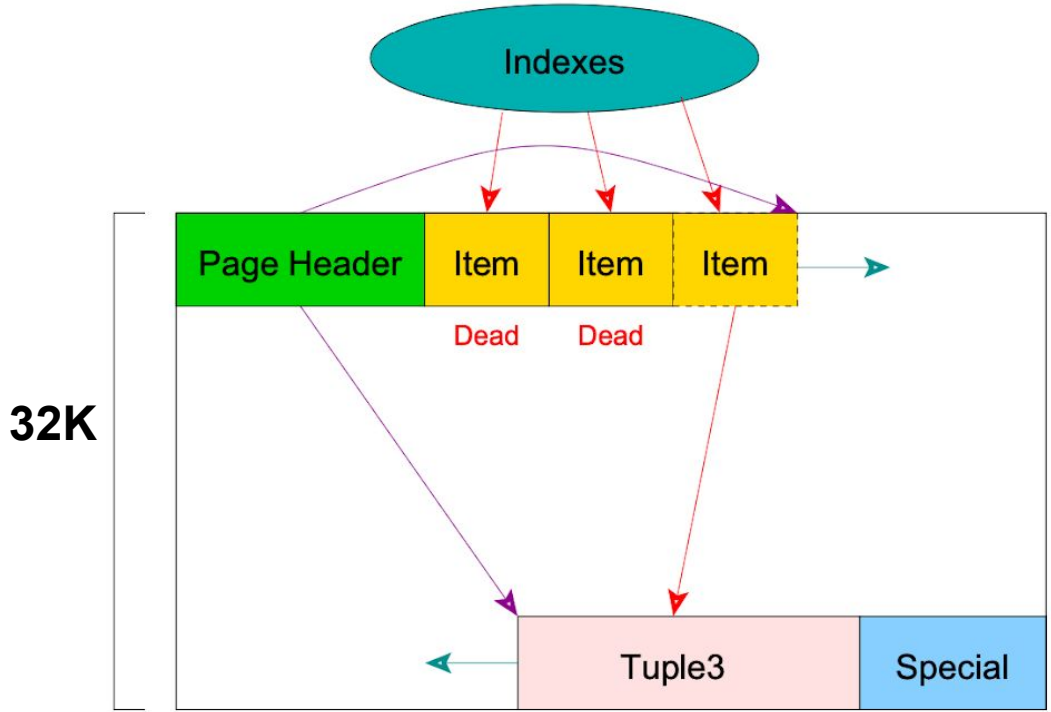
# Heap：Greenplum存储格式



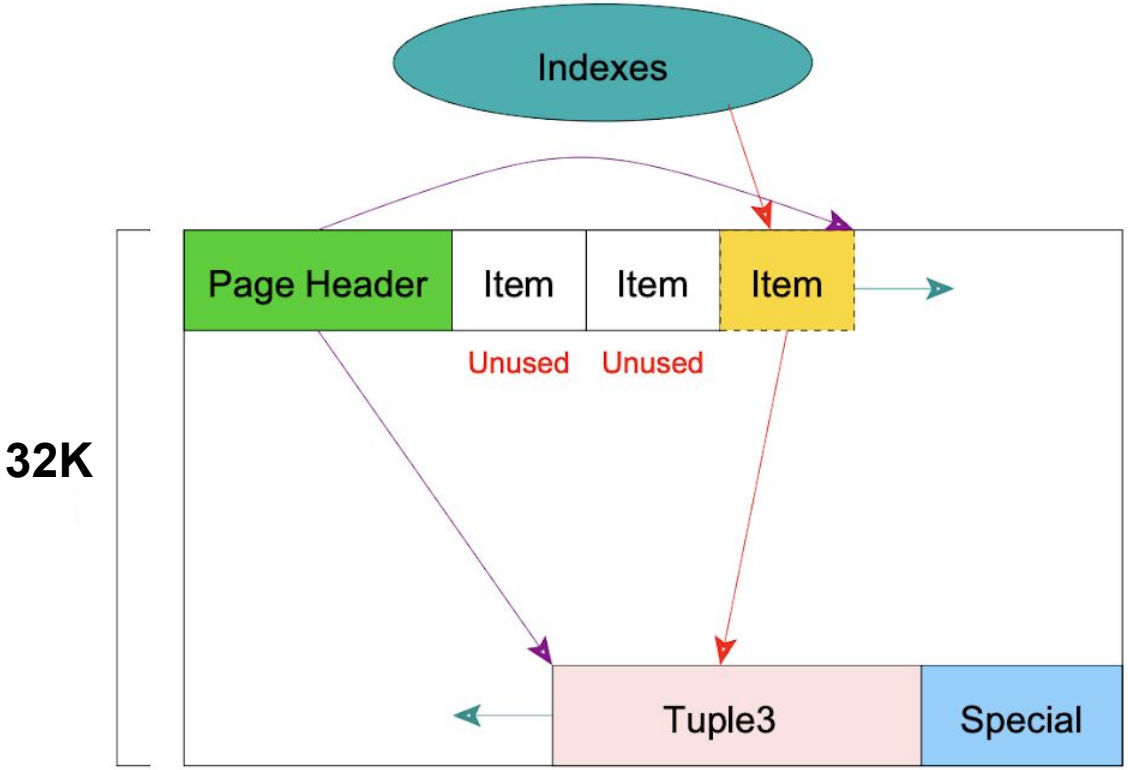
# Heap：索引


Indexes
Page Header
Item
Item
Item
32K
Tuple3
Tuple2
Tuple1
Special

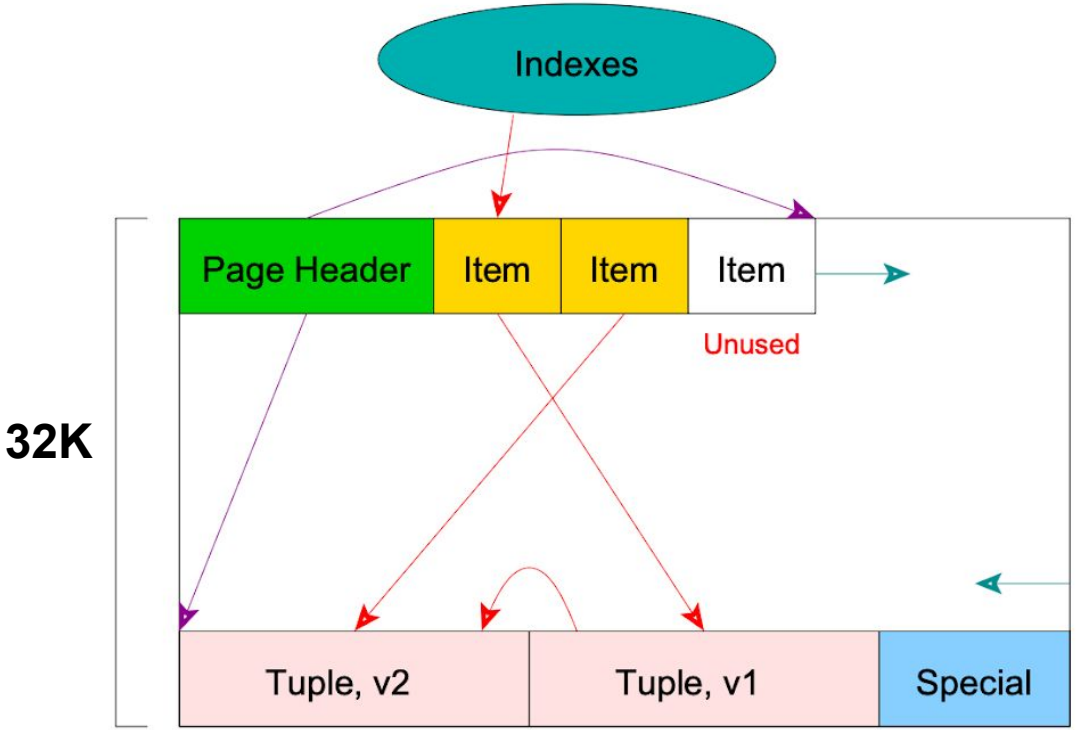
# Heap：删除元组


Indexes
Page Header
Item
Item
Item
Dead
Dead
32K
Tuple3
Special

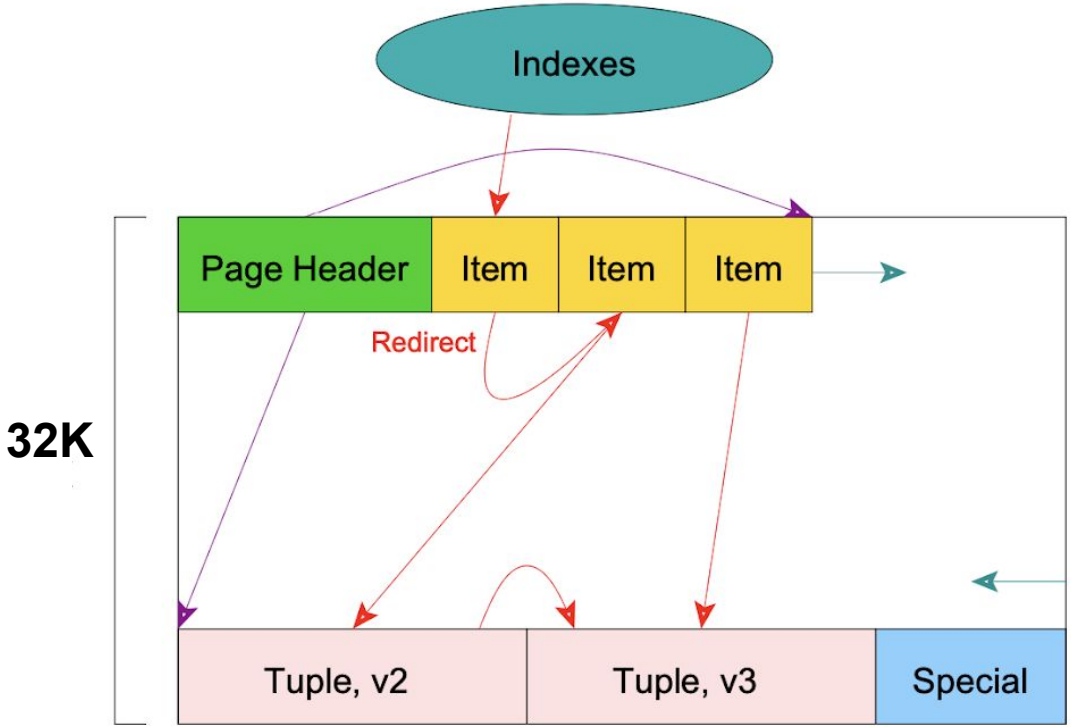
# Heap：索引空间回收


Indexes
Page Header
Item
Item
Item
Unused
Unused
32K
Tuple3
Special

# 减少索引项：Heap Only Tuple（HOT）

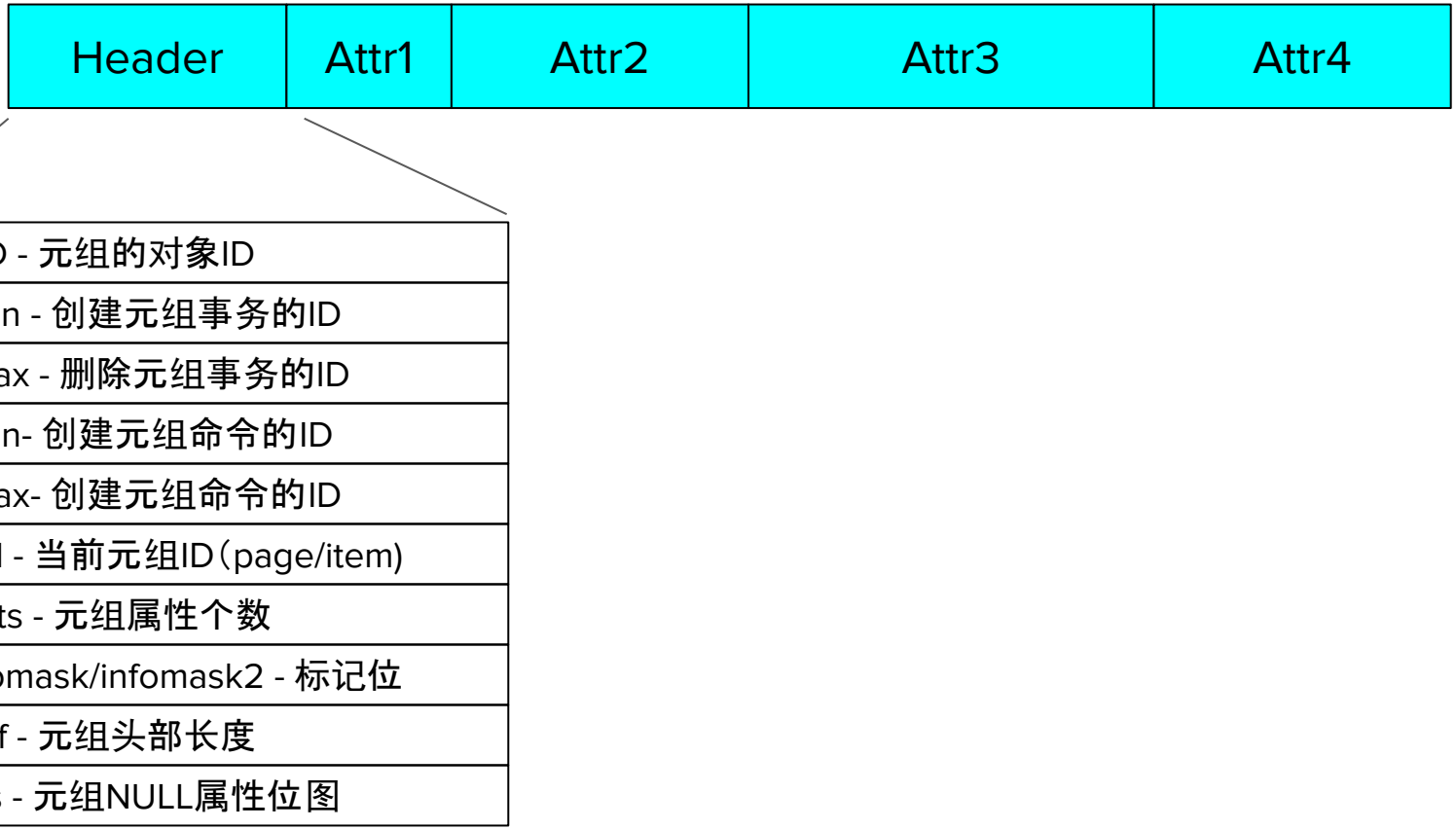


# 减少索引项：Heap Only Tuple（HOT）-续

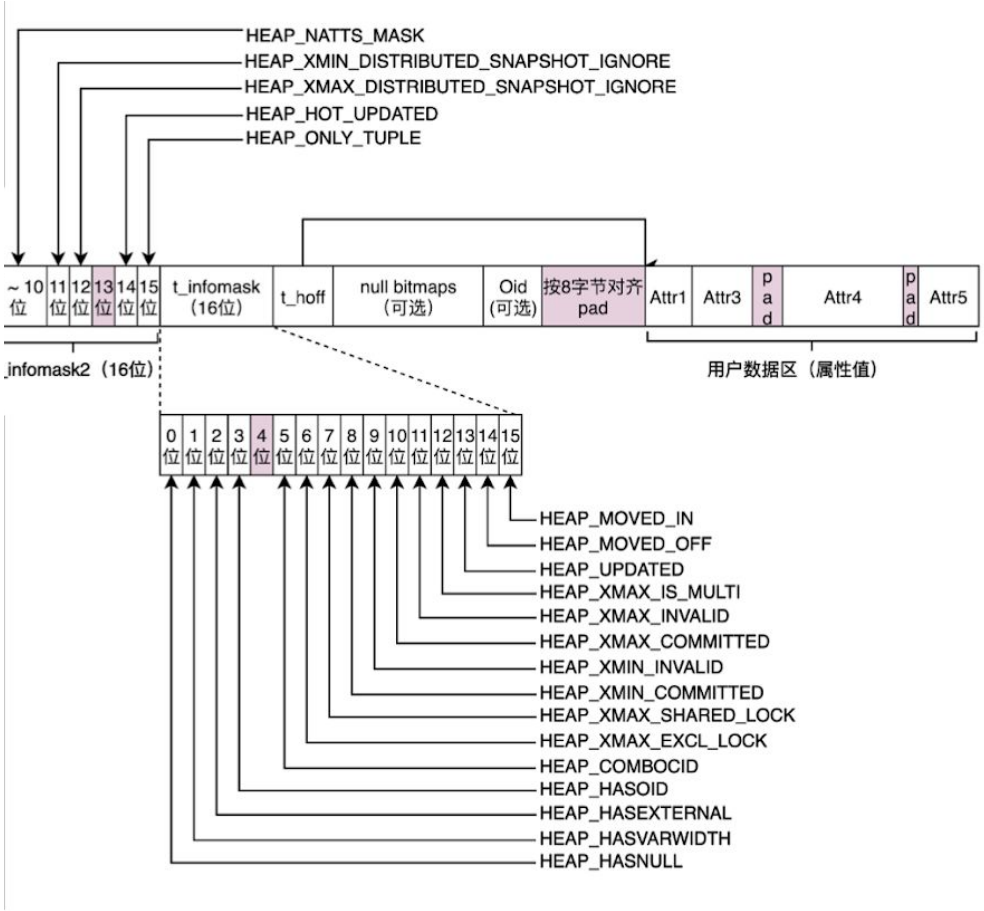


# 元组结构

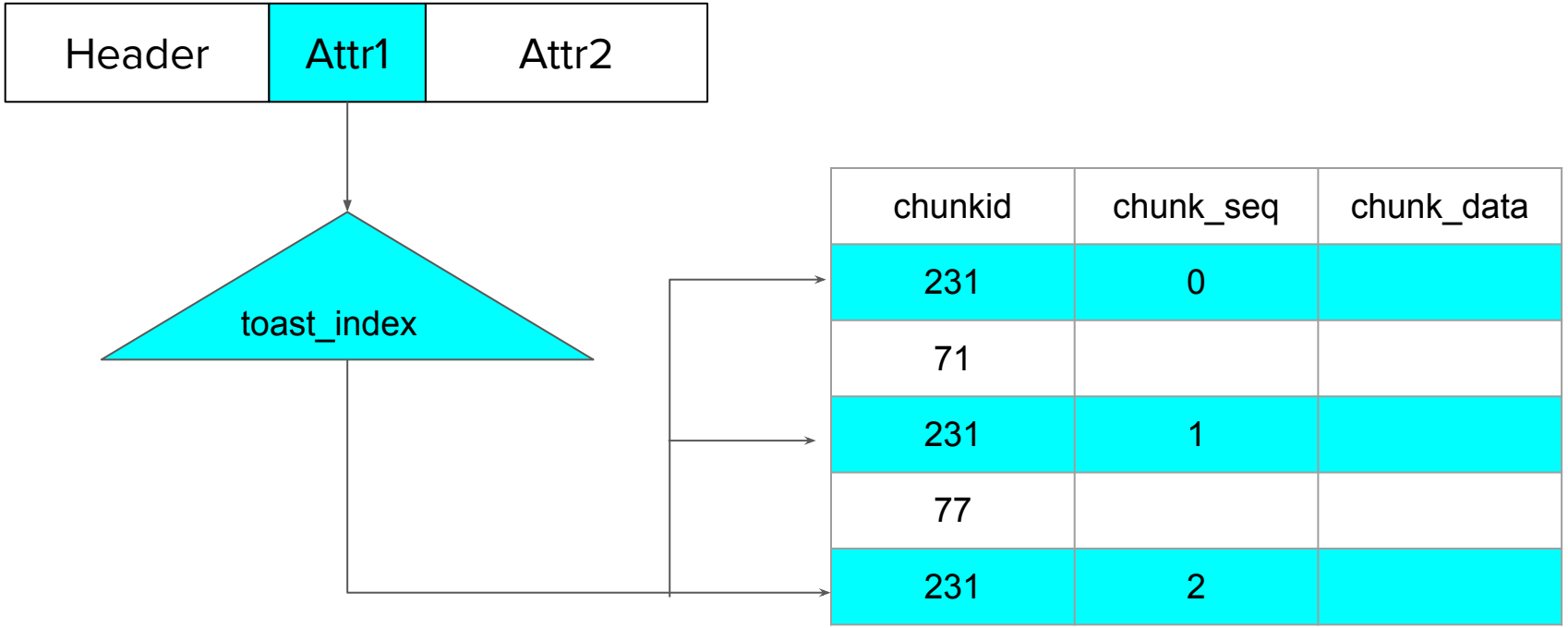
# 元组文件结构


Header
Attr1
Attr2
Attr3
Attr4
OID - 元组的对象ID
xmin - 创建元组事务的ID
xmax - 删除元组事务的ID
cmin- 创建元组命令的ID
cmax- 创建元组命令的ID
ctid - 当前元组ID（page/item)
natts - 元组属性个数
infomask/infomask2 - 标记位
hoff - 元组头部长度
bits - 元组NULL属性位图

# 元组标记位

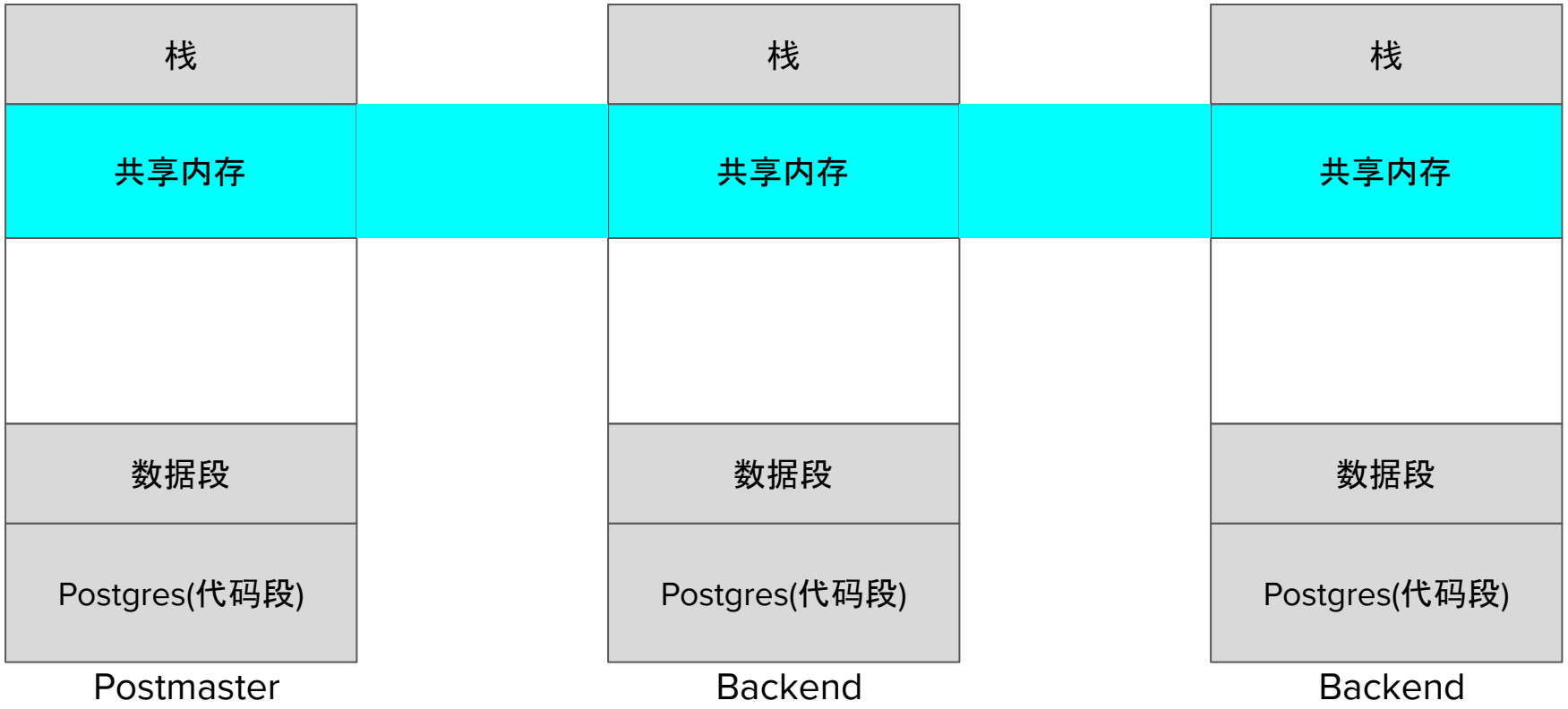

HEAP_NATTS_MASK
HEAP_XMIN_DISTRIBUTED_SNAPSHOT_IGNORE
HEAP_XMAX_DISTRIBUTED_SNAPSHOT_IGNORE
HEAP_HOT_UPDATED
HEAP_ONLY_TUPLE
~ 10 位
11 位
12 位
13 位
14 位
15 位
t_infomask (16位)
t_hoff
null bitmaps (可选)
Oid (可选)
按8字节对齐 pad
Attr1
Attr3
pad
Attr4
pad
Attr5
infomask2（16位）
用户数据区（属性值）
0 位
1 位
2 位
3 位
4 位
5 位
6 位
7 位
8 位
9 位
10 位
11 位
12 位
13 位
14 位
15 位
HEAP_MOVED_IN
HEAP_MOVED_OFF
HEAP_UPDATED
HEAP_XMAX_IS_MULTI
HEAP_XMAX_INVALID
HEAP_XMAX_COMMITTED
HEAP_XMIN_INVALID
HEAP_XMIN_COMMITTED
HEAP_XMAX_SHARED_LOCK
HEAP_XMAX_EXCL_LOCK
HEAP_COMBOCID
HEAP_HASOID
HEAP_HASEXTERNAL
HEAP_HASVARWIDTH
HEAP_HASNULL

# 变长属性:压缩和线外存储



# 共享缓冲区管理

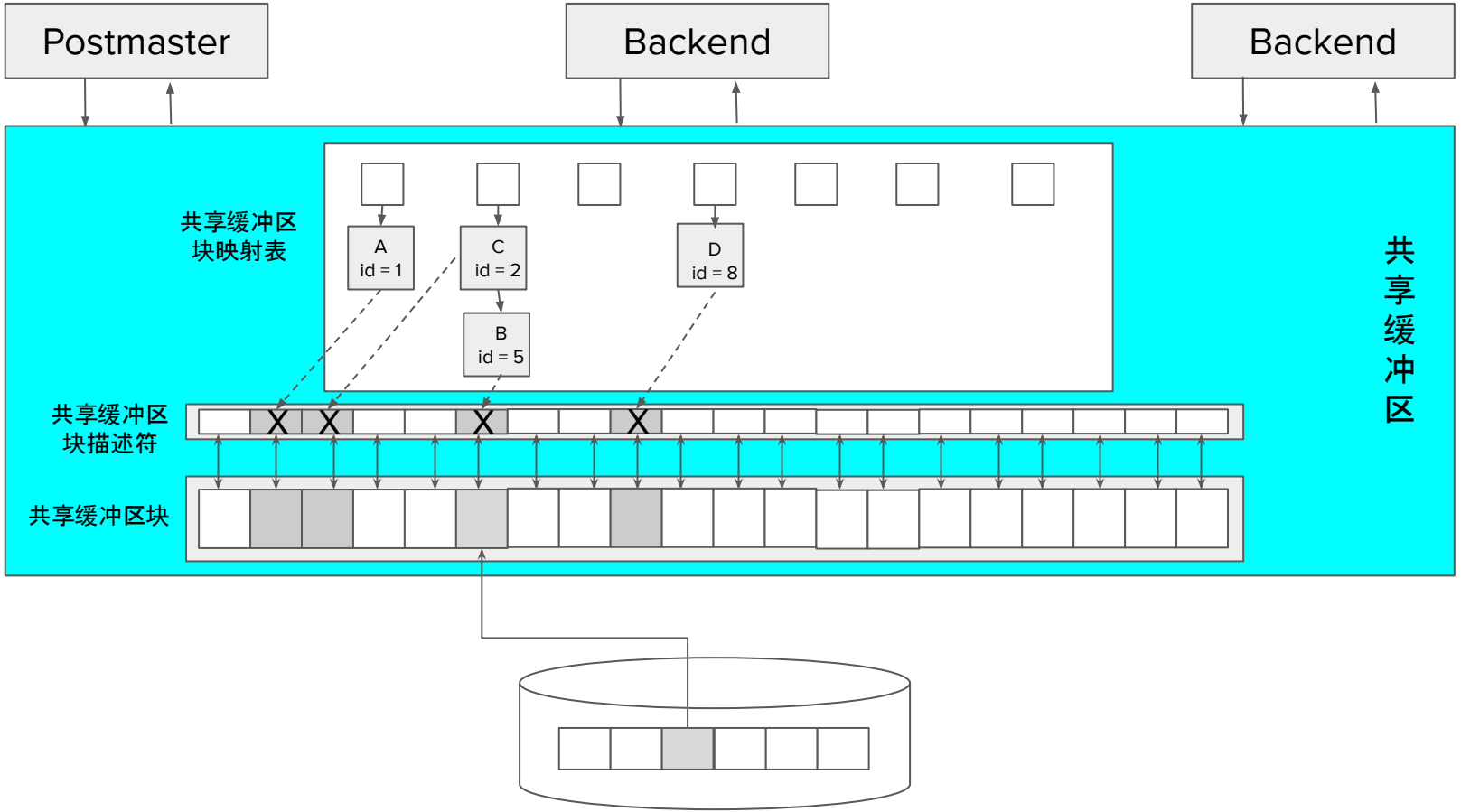
# Greenplum共享内存模型



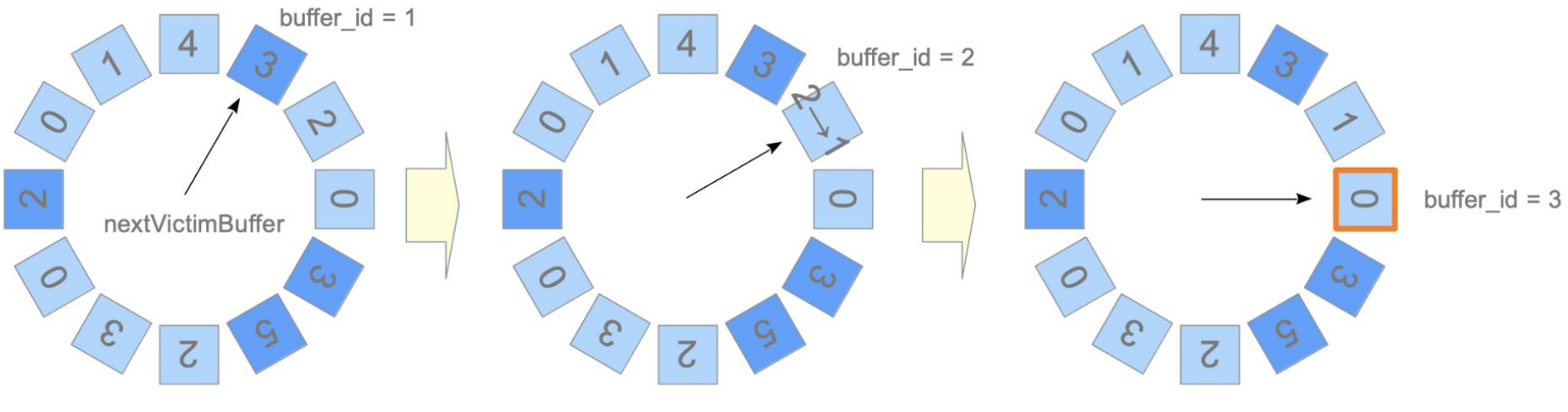
# 共享缓冲区管理

共享内存
共享缓冲区
并发控制（锁，Latch等）
数据交换和管道
（共享消息队列，管道等）
日志系统
进程结构体Proc
流式复制共享内存
WalSnd/WalRcv
同步扫描（SyncScan）
PgStats共享内存
.....

# 共享缓冲区(考虑随机访问与扫描多种访问模式)

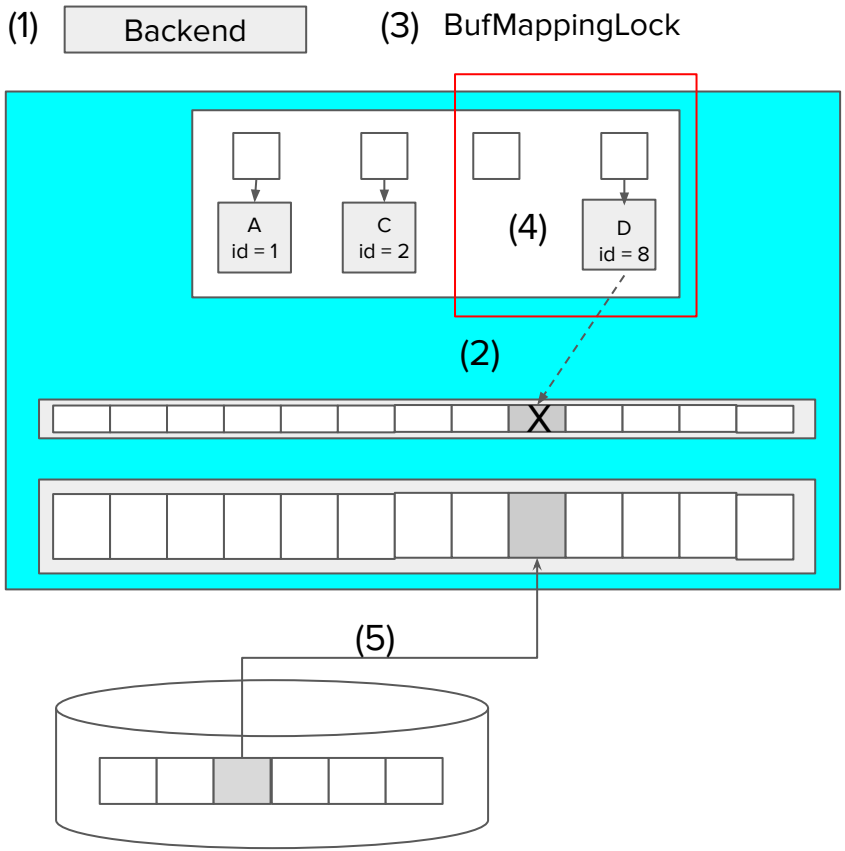


# 缓冲区替换算法：LRU



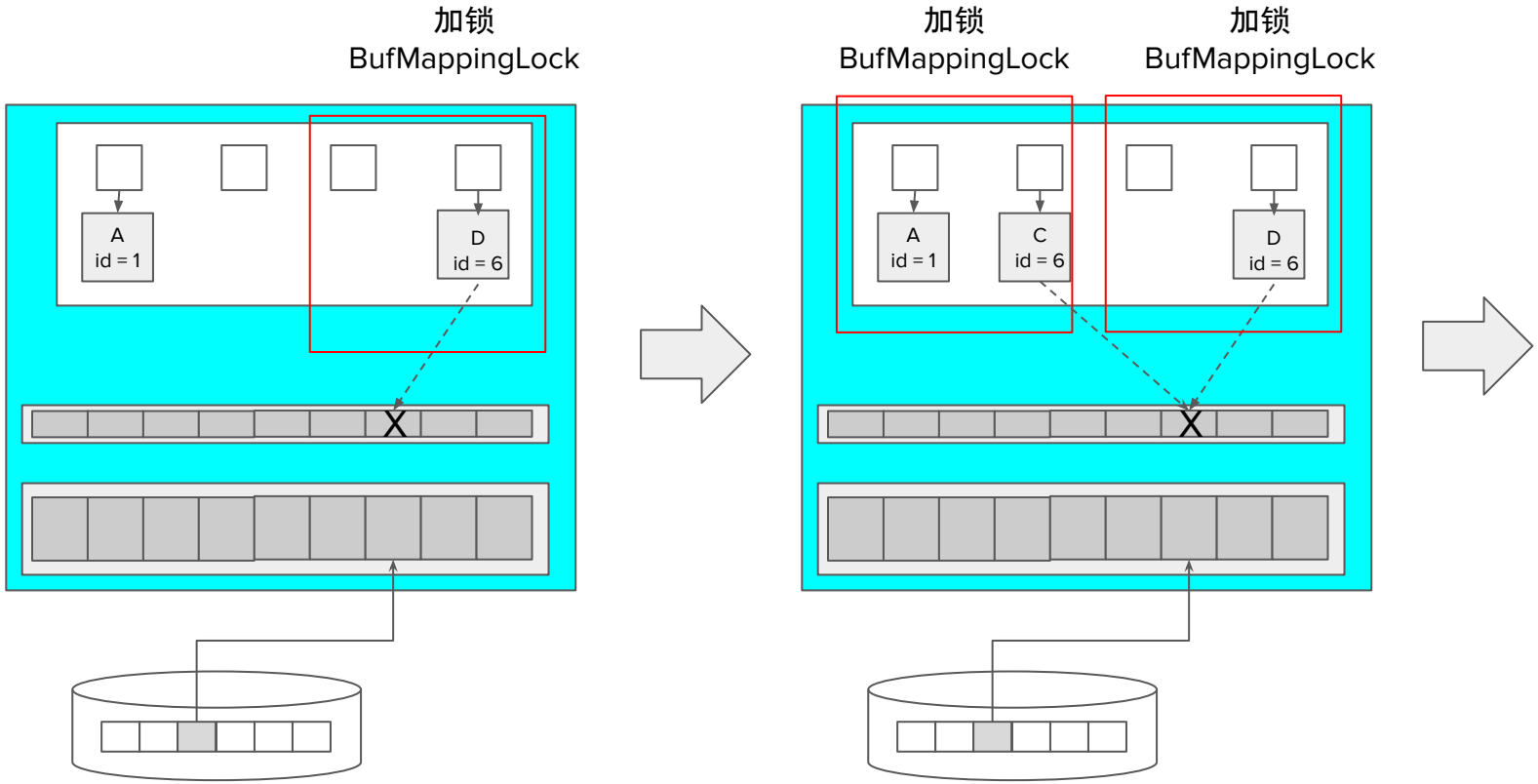

对于顺序扫描/插入数据会导致算法失效？需要更细粒度的缓冲区替换策略！

# 读取文件页(存在空闲缓冲区块)

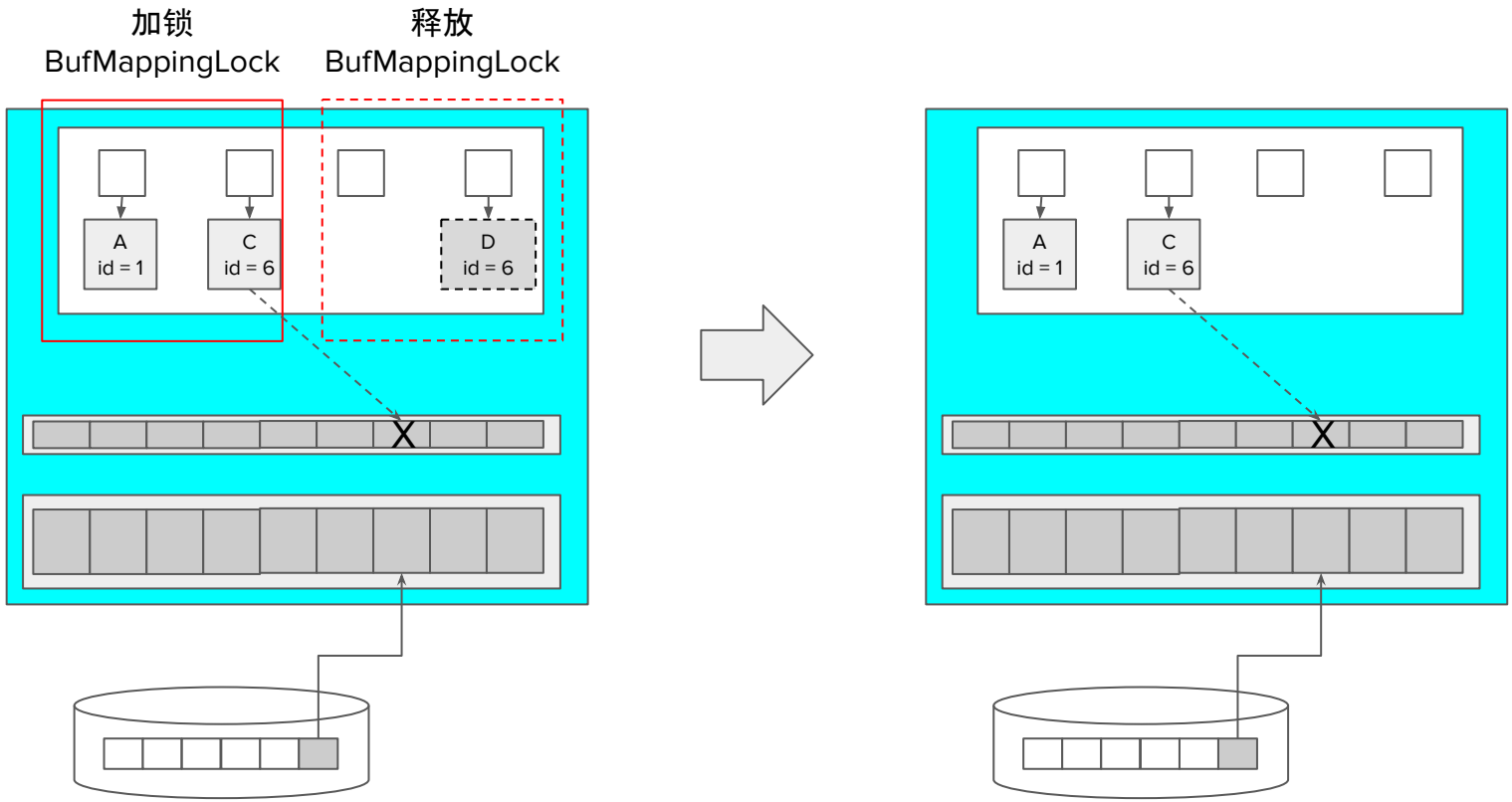



(1)开始读取特定文件块
(2)找到一个空闲缓冲区
(3)对映射表加分区锁，分区锁相比与大锁减少锁的竞争
(4)映射表中插入表项
(5)开始IO读取内容到缓冲区块中

# 读取文件页(替换已有缓冲区块)

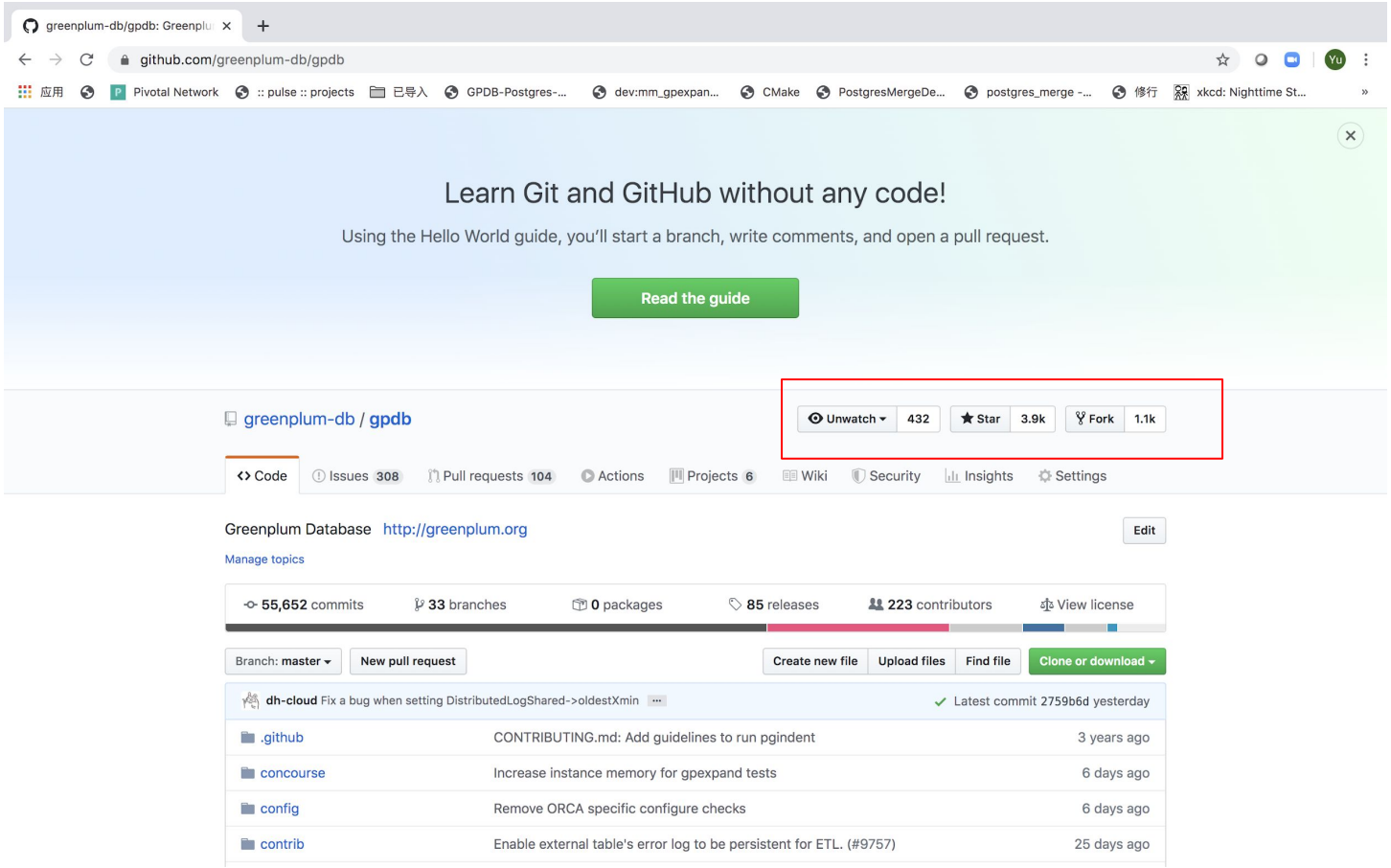


# 读取文件页(替换已有缓冲区块)-续



# 准备工作

# 从源代码开始：下载编译Greenplum源代码



# GREENPLUM DATABASE®

微信技术讨论群
添加入群小助手：gp_assistant

微信公众号
技术干货、行业热点、活动预告

欢迎访问Greenplum中文社区：cn.greenplum.org

# 全新的问答论坛

https://cn.greenplum.org/askgp

GREENPLUM
DATABASE®
Q&A