
Index and redo internals

By

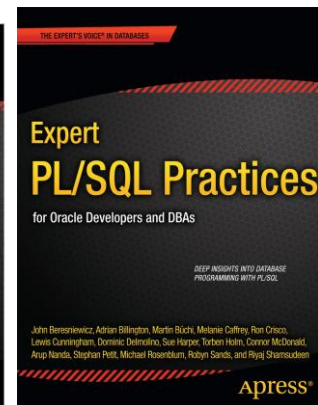
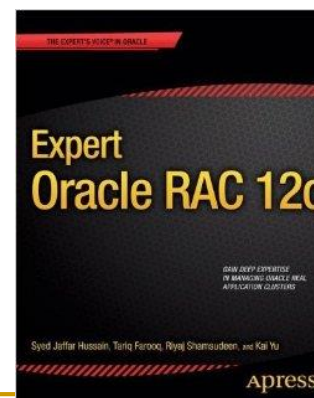
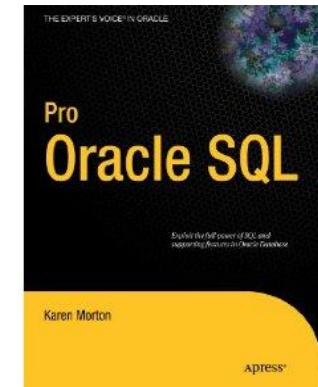
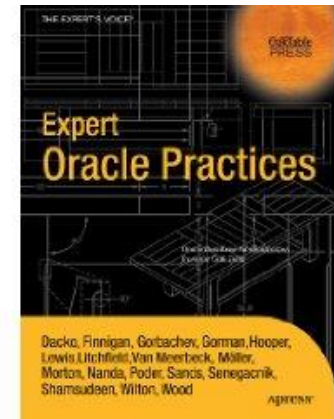
Riyaj Shamsudeen

Me



ORACLE
ACE Director

- 23+ years as DBA/Developer
- OakTable member
- Oracle ACE Director
- Specializes in RAC, performance tuning and Internals.
- Slowly in to BigData
- rshamsud@orainternals.com
- orainternals.wordpress.com
- Web: www.orainternals.com



WARNING

Most of the topics in this presentations are from my research.

Writing about internals have issues:

- a. I completely misunderstood the data and trace files.
- b. Future version changed the feature, so, information is outdated.

Tested in version 12.1.0.2, Linux and Solaris 11 platform.

AGENDA

Index structures

Unique vs Non-unique indexes

Updates to indexed columns

Update to the same value

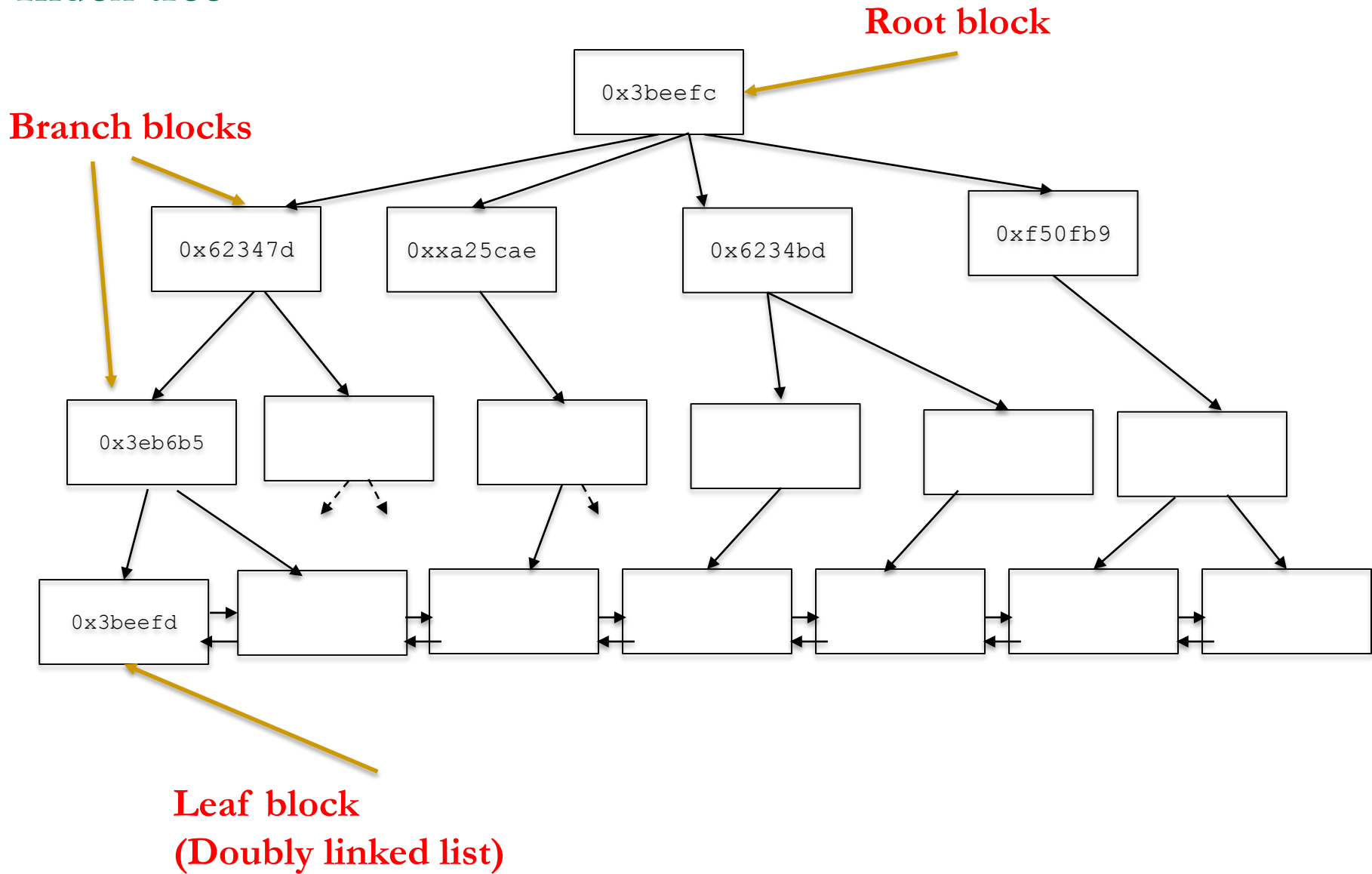
Row chaining and row migration

Other index types

Index splits

Conclusion

Index tree



Trace dump

```
alter session set events 'immediate trace name treedump level <object_id>'
```

```
----- begin tree dump
```

```
branch: 0x3beefc 3927804 (0: nrow: 4, level: 3)
```

```
  branch: 0x62347d 6435965 (-1: nrow: 342, level: 2)
```

```
    branch: 0x3eb6b5 4110005 (-1: nrow: 608, level: 1)
```

```
      leaf: 0x3beefd 3927805 (-1: row:512.512 avs:828)
```

```
      leaf: 0x3beefe 3927806 (0: row:291.291 avs:3922)
```

```
      leaf: 0x48b94f 4766031 (1: row:285.285 avs:4006)
```

```
      leaf: 0x558990 5605776 (2: row:288.288 avs:3964)
```

```
      leaf: 0x64447a 6571130 (3: row:288.288 avs:3964)
```

```
      leaf: 0x72f6ca 7534282 (4: row:288.288 avs:3964)
```

```
      leaf: 0x83961b 8623643 (5: row:288.288 avs:3964)
```

```
      leaf: 0x8fd32a 9425706 (6: row:288.288 avs:3964)
```

```
      leaf: 0xa37c54 10714196 (7: row:288.288 avs:3964)
```

```
      leaf: 0xb42111 11804945 (8: row:286.286 avs:3992)
```

```
      leaf: 0xc9f700 13235968 (9: row:288.288 avs:3964)
```

```
      leaf: 0xdbb023 14397475 (10: row:284.284 avs:4020)
```

```
      leaf: 0xe88a13 15239699 (11: row:284.284 avs:4020)
```

```
      leaf: 0xb40385 11797381 (384: row:276.276 avs:3584)
```

```
      leaf: 0xb404c5 11797701 (385: row:368.368 avs:2108)
```

```
      leaf: 0xb403c5 11797445 (386: row:500.500 avs:4)
```

```
      leaf: 0xb40445 11797573 (387: row:484.484 avs:256)
```

```
...
```

Index root block

```
alter system dump datafile 20 block min 3927804 block max 3927804;
```

```
Block header dump: 0x003beefc
```

```
Object id on Block? Y
```

```
seg/obj: 0xf7b4 csc: 0x1e.ca8b5746 itc: 1 flg: E typ: 2 - INDEX
```

```
brn: 0 bdba: 0x3beef8 ver: 0x01 opc: 0
```

```
inc: 0 exflg: 0
```

| Itl | Xid | Uba | Flag | Lck | Scn/Fsc |
|------|---------------------|--------------------|------|-----|---------------------|
| 0x01 | 0x003c.001.0003aa06 | 0x006526ba.e403.02 | C--- | 0 | scn 0x001e.ca8b5746 |

```
Branch block dump
```

```
=====
```

```
header address 140098665452108=0x7f6b4330124c
```

```
kdxcolev 3
```

```
KDXCOLEV Flags = - - -
```

```
kdxcolok 0
```

```
kdxcoopc 0x80: opcode=0: iot flags=--- is converted=Y
```

```
kdxconco 2
```

```
kdxcosdc 1
```

```
kdxconro 3
```

```
kdxcofbo 34=0x22
```

```
kdxcofeo 8026=0x1f5a
```

```
kdxcoavs 7992
```

kdxcolev: # of index levels (+ leaf)

kdxconco: # of columns indexed

kdxconro: # of entries in this block

Root block dump

kdxbrlmc: left most child

```
...
kdxbrlmc 6435965=0x62347d
kdxbrsno 2
kdxbrbksz 8056
kdxbr2urrc 0
row#0[8036] dba: 10640558=0xa25cae
col 0; len 4; (4): c3 02 17 56
col 1; TERM
row#1[8046] dba: 6436029=0x6234bd
col 0; len 4; (4): c3 03 1d 60
col 1; TERM
row#2[8026] dba: 16060345=0xf50fb9
col 0; len 4; (4): c3 04 17 5a
col 1; TERM
----- end of branch block dump -----
```

Values less than c3 02 17 56 will be in this branch of leaf blocks.

First possible value in this branch of leaf blocks.

Trace file

egrep 'level: 3|level: 2|level: 1' xx.trc

```
branch: 0x3beefc 3927804 (0: nrow: 4, level: 3)
  branch: 0x62347d 6435965 (-1: nrow: 342, level: 2)
    branch: 0x3eb6b5 4110005 (-1: nrow: 608, level: 1)
    branch: 0x8ec2dc 9355996 (0: nrow: 566, level: 1)
    branch: 0x638358 6521688 (1: nrow: 540, level: 1)
  ...
  branch: 0xa25cae 10640558 (0: nrow: 336, level: 2)
    branch: 0x560e23 5639715 (-1: nrow: 499, level: 1)
    branch: 0x8fca6b 9423467 (0: nrow: 455, level: 1)
    branch: 0x48091e 4720926 (1: nrow: 542, level: 1)
  ...
  branch: 0x6234bd 6436029 (1: nrow: 297, level: 2)
    branch: 0x483032 4730930 (-1: nrow: 526, level: 1)
    branch: 0x9c65d1 10249681 (0: nrow: 484, level: 1)
  ...
  branch: 0xf50fb9 16060345 (2: nrow: 278, level: 2)
    branch: 0xa257bd 10639293 (-1: nrow: 425, level: 1)
    branch: 0x3f50dd 4149469 (0: nrow: 488, level: 1)
```

Object and index

```
CREATE TABLE "RS"."TEST1"  
  (  
    "N1" NUMBER,  
    "V1" VARCHAR2(32 BYTE),  
    "V2" VARCHAR2(1000 BYTE),  
    "V3" VARCHAR2(32 BYTE),  
    "V4" VARCHAR2(32 BYTE)  
  );
```

```
CREATE INDEX "RS"."TEST1_V1" ON "RS"."TEST1" ("V1");
```

```
  1* select n1, v1, v3, v4 from rs.test1 where rownum <3  
SQL> /
```

| N1 | V1 | V3 | V4 |
|-----|----------|----------|----------|
| 185 | MERCEDES | MERCEDES | MERCEDES |
| 186 | MARCO | MARCO | MARCO |

Trace dump

```
alter session set events 'immediate trace name treedump level 96090'
```

```
----- begin tree dump
```

```
branch: 0x1848403 25461763 (0: nrow: 13, level: 1)
```

```
  leaf: 0x1848404 25461764 (-1: row:398.398 avs:817)
```

```
  leaf: 0x1848405 25461765 (0: row:393.393 avs:818)
```

```
  leaf: 0x1848406 25461766 (1: row:391.391 avs:831)
```

```
  leaf: 0x1848407 25461767 (2: row:403.403 avs:818)
```

```
  leaf: 0x1848408 25461768 (3: row:390.390 avs:832)
```

```
  leaf: 0x1848409 25461769 (4: row:404.404 avs:819)
```

```
  leaf: 0x184840a 25461770 (5: row:399.399 avs:831)
```

```
  leaf: 0x184840b 25461771 (6: row:402.402 avs:827)
```

```
  leaf: 0x184840c 25461772 (7: row:391.391 avs:827)
```

```
  leaf: 0x184840d 25461773 (8: row:405.405 avs:826)
```

```
  leaf: 0x184840e 25461774 (9: row:399.399 avs:826)
```

```
  leaf: 0x184840f 25461775 (10: row:399.399 avs:818)
```

```
  leaf: 0x1848411 25461777 (11: row:389.389 avs:1143)
```

```
----- end tree dump
```

Index branch block.

Object id on Block? Y

seg/obj: 0x1775a csc: 0x00.1d10560 itc: 1 flg: E typ: 2 - INDEX

brn: 0 bdba: 0x1848400 ver: 0x01 opc: 0

inc: 0 exflg: 0

| Itl | Xid | Uba | Flag | Lck | Scn/Fsc |
|------|---------------------|--------------------|------|-----|---------------------|
| 0x01 | 0xffff.000.00000000 | 0x00000000.0000.00 | C--- | 0 | scn 0x0000.01d10560 |

Branch block dump

=====

header address 140188528960076=0x7f802f78624c

kdxcolev 1

KDXCOLEV Flags = - - -

kdxcolok 0

kdxcoopc 0x80: opcode=0: iot flags=--- is converted=Y

kdxconco 2

kdxcosdc 0

kdxconro 12

kdxcofbo 52=0x34

kdxcofeo 7925=0x1ef5

kdxcoavs 7873

kdxbrlmc 25461764=0x1848404

kdxbrsno 0


kdxbrbksz 8056

kdxbr2urrc 0

Anything less than BAO in this first block (See next slide).

Index branch block.

Lines with +++ added to the trace
file to print ASCII characters



```
row#0[8047] dba: 25461765=0x1848405
col 0; len 3; (3):  42 41 4f
col 0; len 3; (3):  B  A  O +++
col 1; TERM
row#1[8035] dba: 25461766=0x1848406
col 0; len 6; (6):  43 48 41 52 4c 4f
col 0; len 6; (6):  C  H  A  R  L  O +++
col 1; TERM
row#2[8021] dba: 25461767=0x1848407
col 0; len 8; (8):  44 45 4d 45 54 52 49 43
col 0; len 8; (8):  D  E  M  E  T  R  I  C +++
col 1; TERM
row#3[8011] dba: 25461768=0x1848408
col 0; len 4; (4):  45 55 53 54
col 0; len 4; (4):  E  U  S  T +++
col 1; TERM
row#4[8002] dba: 25461769=0x1848409
col 0; len 3; (3):  48 49 50
col 0; len 3; (3):  H  I  P +++
col 1; TERM
row#5[7991] dba: 25461770=0x184840a
col 0; len 5; (5):  4a 4f 52 44 4f
col 0; len 5; (5):  J  O  R  D  O +++
col 1; TERM
```

Index leaf block.

Object id on Block? Y

seg/obj: 0x1775a csc: 0x00.1d10560 itc: 2 flg: E typ: 2 - INDEX

brn: 0 bdba: 0x1848400 ver: 0x01 opc: 0

inc: 0 exflg: 0

| Itl | Xid | Uba | Flag | Lck | Scn/Fsc |
|------|---------------------|--------------------|------|-----|---------------------|
| 0x01 | 0x0000.000.00000000 | 0x00000000.0000.00 | ---- | 0 | fsc 0x0000.00000000 |
| 0x02 | 0xffff.000.00000000 | 0x00000000.0000.00 | C--- | 0 | scn 0x0000.01d10560 |

Leaf block dump

=====

header address 140188528960100=0x7f802f786264

kdxcolev 0

KDXCOLEV Flags = - - -

kdxcolok 0

kdxcoopc 0x80: opcode=0: iot flags=--- is converted=Y

kdxconco 2

kdxcosdc 0

kdxconro 393

kdxcofbo 822=0x336

kdxcofeo 1640=0x668

kdxcoavs 818

kdxlespl 0

kdxlende 0

kdxlenxt 25461766=0x1848406

kdxleprv 25461764=0x1848404

kdxledsz 0

kdxlebksz 8032

kdxlenxt: next leaf block in the chain

kdxleprv: previous leaf block in the chain

Index leaf block – non unique

```
row#0[8019] flag: -----, lock: 0, len=13
col 0; len 3; (3):  42 41 4f
col 0; len 3; (3):   B  A  O +++
col 1; len 6; (6):  01 84 a5 76 00 b4
row#1[8001] flag: -----, lock: 0, len=18
col 0; len 8; (8):  42 41 52 41 42 41 52 41
col 0; len 8; (8):   B  A  R  A  B  A  R  A +++
col 1; len 6; (6):  01 84 a5 76 00 b3
row#2[7987] flag: -----, lock: 0, len=14
col 0; len 4; (4):  42 41 52 42
col 0; len 4; (4):   B  A  R  B +++
col 1; len 6; (6):  01 84 83 e3 00 3e
row#3[7971] flag: -----, lock: 0, len=16
col 0; len 6; (6):  42 41 52 42 41 52
col 0; len 6; (6):   B  A  R  B  A  R +++
col 1; len 6; (6):  01 84 83 eb 00 69
row#4[7954] flag: -----, lock: 0, len=17
col 0; len 7; (7):  42 41 52 42 41 52 41
col 0; len 7; (7):   B  A  R  B  A  R  A +++
col 1; len 6; (6):  01 84 83 ed 00 2c
...
```

← Rowid is a key column internally.

Index leaf block - unique

```
row#58[3301] flag: -----, lock: 2, len=12, data:(6): 01 84 a5 76 00 b4
col 0; len 3; (3): 42 41 4f
col 0; len 3; (3): B A O +++
row#59[3313] flag: -----, lock: 2, len=17, data:(6): 01 84 a5 76 00 b3
col 0; len 8; (8): 42 41 52 41 42 41 52 41
col 0; len 8; (8): B A R A B A R A +++
row#60[5167] flag: ----S--, lock: 2, len=13, data:(6): 01 84 83 e3 00 3e
col 0; len 4; (4): 42 41 52 42
col 0; len 4; (4): B A R B +++
row#61[5180] flag: ----S--, lock: 2, len=15, data:(6): 01 84 83 eb 00 69
col 0; len 6; (6): 42 41 52 42 41 52
col 0; len 6; (6): B A R B A R +++
row#62[5195] flag: ----S--, lock: 2, len=16, data:(6): 01 84 83 ed 00 2c
col 0; len 7; (7): 42 41 52 42 41 52 41
col 0; len 7; (7): B A R B A R A +++
row#63[5211] flag: ----S--, lock: 2, len=16, data:(6): 01 84 83 eb 00 68
col 0; len 7; (7): 42 41 52 42 45 52 41
col 0; len 7; (7): B A R B E R A +++
row#64[4247] flag: -----, lock: 2, len=15, data:(6): 01 84 a5 70 00 23
col 0; len 6; (6): 42 41 52 42 49 45
col 0; len 6; (6): B A R B I E +++
```


Non unique –before insert

```
row#2[7987] flag: -----, lock: 0, len=15  
col 0; len 5; (5): 41 42 42 49 45  
col 0; len 5; (5): A B B I E +++  
col 1; len 6; (6): 01 84 83 e1 00 64
```

```
row#3[7973] flag: -----, lock: 0, len=14  
col 0; len 4; (4): 41 42 42 59  
col 0; len 4; (4): A B B Y +++  
col 1; len 6; (6): 01 84 83 dd 00 41
```

```
row#4[7958] flag: -----, lock: 0, len=15  
col 0; len 5; (5): 41 42 44 55 4c  
col 0; len 5; (5): A B D U L +++  
col 1; len 6; (6): 01 84 83 e2 00 45
```

Non unique after insert of a row

```
row#2[7987] flag: -----, lock: 0, len=15
col 0; len 5; (5):  41 42 42 49 45
col 0; len 5; (5):  A  B  B  I  E +++
col 1; len 6; (6):  01 84 83 e1 00 64
```

Deterministic order of leaf rows
with the addition of rowed.

```
row#3[7973] flag: -----, lock: 0, len=14
col 0; len 4; (4):  41 42 42 59
col 0; len 4; (4):  A B B Y +++
col 1; len 6; (6):  01 84 83 dd 00 41
row#4[1635] flag: -----, lock: 2, len=14
col 0; len 4; (4):  41 42 42 59
col 0; len 4; (4):  A B B Y +++
col 1; len 6; (6):  01 84 84 a1 00 01
```

```
row#5[7958] flag: -----, lock: 0, len=15
col 0; len 5; (5):  41 42 44 55 4c
col 0; len 5; (5):  A  B  D  U  L +++
col 1; len 6; (6):  01 84 83 e2 00 45
```

Update to indexed column

```
select min(v1), max(v1) from test1;
alter system switch logfile;
alter system switch logfile;
@dump_last_log
```

```
1* select min(v1), max(v1) from test1;
```

| MIN(V1) | MAX(V1) |
|---------|---------|
| AARON | ZULMA |

```
SQL> update rs.test1 set v1= 'ZULMA' ||v1 where v1='AARON';
```

```
1 row updated.
```

```
SQL> select min(v1), max(v1) from test1;
```

| MIN(V1) | MAX(V1) |
|---------|------------|
| ABBEY | ZULMAAARON |

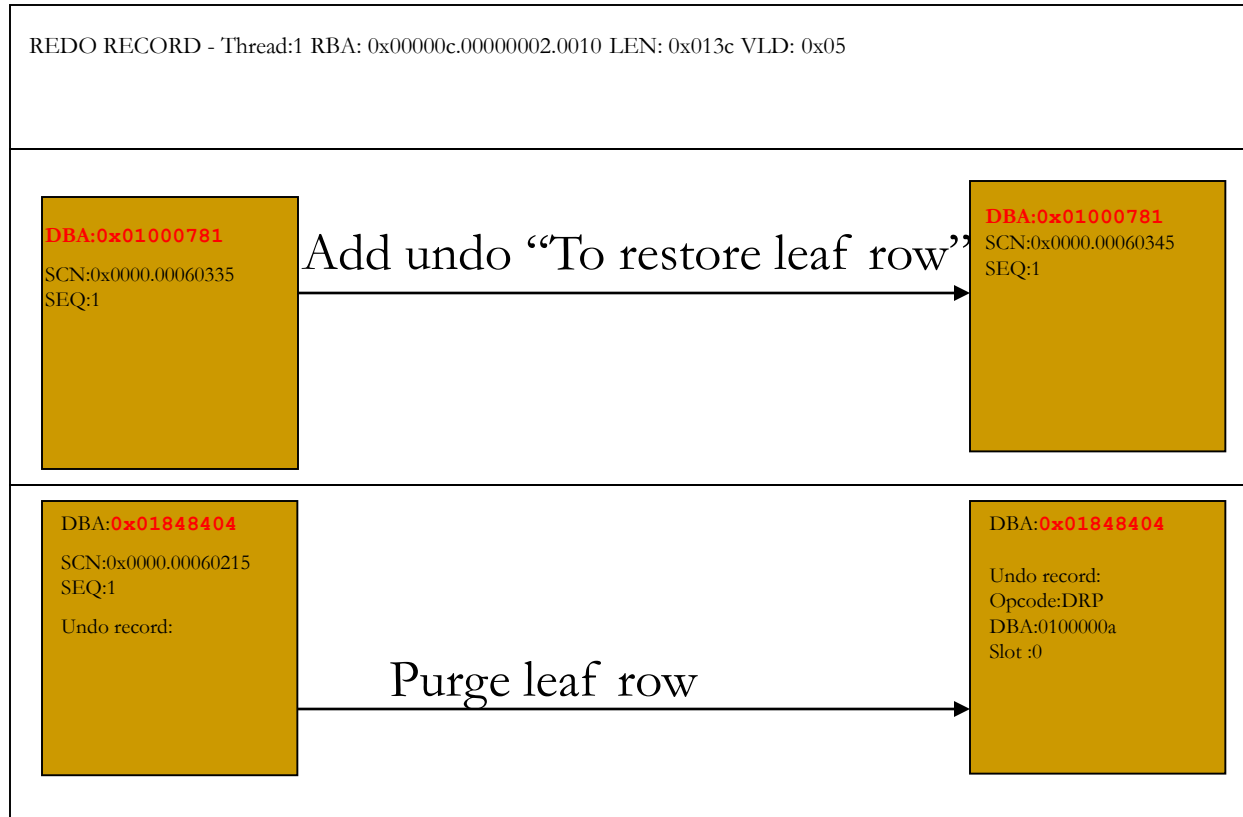
Tree dump

Index entry will move from left most child to right most child.

alter session set events 'immediate trace name treedump level 96090'

```
----- begin tree dump
branch: 0x1848403 25461763 (0: nrow: 13, level: 1)
  leaf: 0x1848404 25461764 (-1: row:398.398 avs:817)
  leaf: 0x1848405 25461765 (0: row:393.393 avs:818)
  leaf: 0x1848406 25461766 (1: row:391.391 avs:831)
  leaf: 0x1848407 25461767 (2: row:403.403 avs:818)
  leaf: 0x1848408 25461768 (3: row:390.390 avs:832)
  leaf: 0x1848409 25461769 (4: row:404.404 avs:819)
  leaf: 0x184840a 25461770 (5: row:399.399 avs:831)
  leaf: 0x184840b 25461771 (6: row:402.402 avs:827)
  leaf: 0x184840c 25461772 (7: row:391.391 avs:827)
  leaf: 0x184840d 25461773 (8: row:405.405 avs:826)
  leaf: 0x184840e 25461774 (9: row:399.399 avs:826)
  leaf: 0x184840f 25461775 (10: row:399.399 avs:818)
  leaf: 0x1848411 25461777 (11: row:389.389 avs:1143)
----- end tree dump
```

Redo record



Update to indexed column

Undo block change

```
REDO RECORD - Thread:1 RBA: 0x0000ef.00000004.00d8 LEN: 0x00fc VLD: 0x01
CON_UID: 0
SCN: 0x0000.01d95084 SUBSCN: 1 12/07/2016 21:15:09
CHANGE #1 CON_ID:0 TYP:0 CLS:18 AFN:4 DBA:0x01000781 OBJ:4294967295
SCN:0x0000.01d95083 SEQ:3 OP:5.1 ENC:0 RBL:0 FLG:0x0000
ktudb redo: siz: 104 spc: 7608 flg: 0x0022 seq: 0x0170 rec: 0x06
          xid: 0x0001.020.00000ac9
ktubu redo: slt: 32 rci: 5 opc: 10.22 objn: 96090 objd: 96090 tsn: 4
Undo type: Regular undo          Undo type: Last buffer split: No
Tablespace Undo: No
          0x00000000
index undo for leaf key operations
KTB Redo
op: 0x04 ver: 0x01
compat bit: 4 (post-11) padding: 1
op: L itl: xid: 0xffff.000.00000000 uba: 0x00000000.0000.00
          flg: C--- lkc: 0 scn: 0x0000.01d10560
Dump kdilk : itl=2, kdxlkflg=0x1 sdc=0 indexid=0x1848402 block=0x01848404
(kdxmlre): restore leaf row (clear leaf delete flags)
key : (13): 05 41 41 52 4f 4e 06 01 84 83 ee 00 05
          ↑           ↑
          value      rowid
```

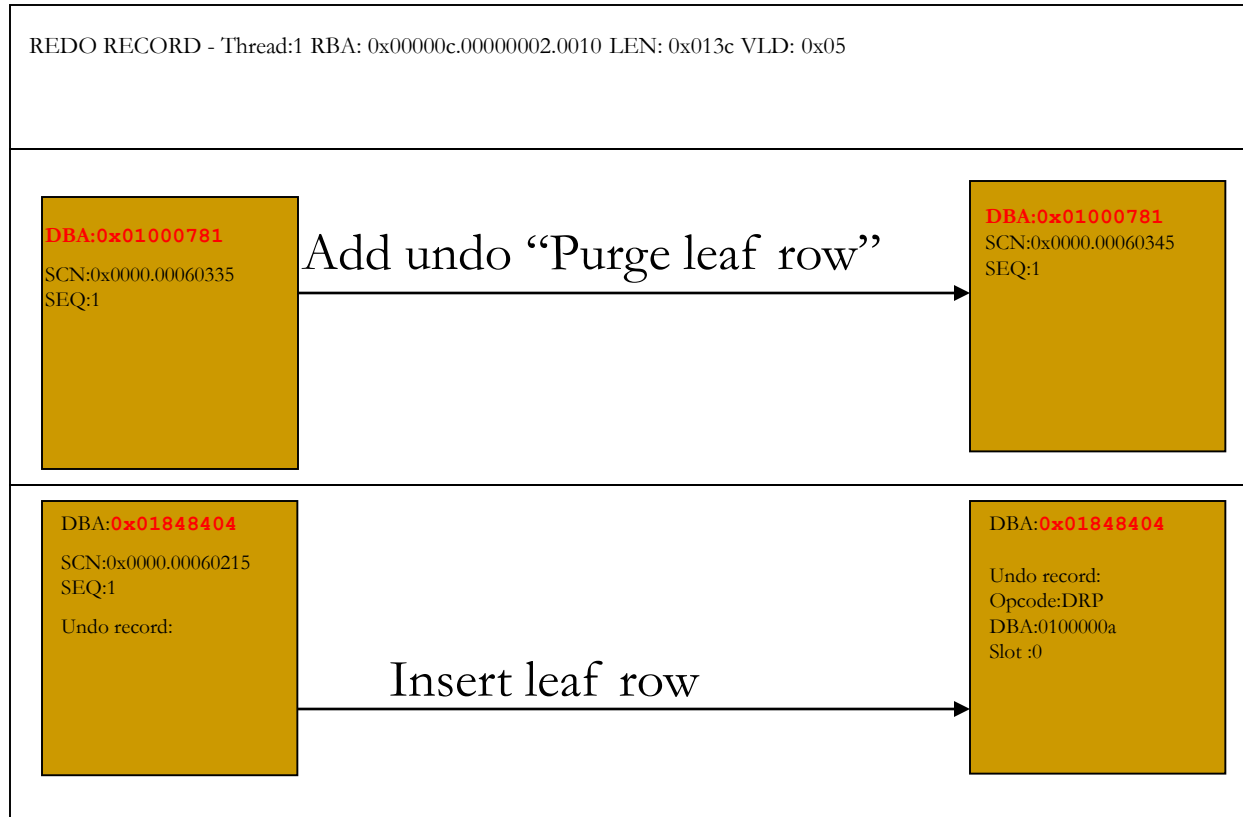
Update to indexed column

Left most child



```
CHANGE #2 CON_ID:0 TYP:0 CLS:1 AFN:6 DBA:0x01848404 OBJ:96090
SCN:0x0000.01d10565 SEQ:2 OP:10.4 ENC:0 RBL:0 FLG:0x0000
index redo (kdxlde): delete leaf row
KTB Redo
op: 0x01 ver: 0x01
compat bit: 4 (post-11) padding: 1
op: F xid: 0x0001.020.00000ac9 uba: 0x01000781.0170.06
REDO: SINGLE / -- / --
itl: 2, sno: 0, row size 17
```

Redo record



Update to indexed column (2)

```
REDO RECORD - Thread:1 RBA: 0x0000ef.00000004.01d4 LEN: 0x0114 VLD: 0x01
CON_UID: 0
SCN: 0x0000.01d95084 SUBSCN: 1 12/07/2016 21:15:09
CHANGE #1 CON_ID:0 TYP:0 CLS:18 AFN:4 DBA:0x01000781 OBJ:4294967295
SCN:0x0000.01d95084 SEQ:1 OP:5.1 ENC:0 RBL:0 FLG:0x0000
ktudb redo: siz: 108 spc: 7502 flg: 0x0022 seq: 0x0170 rec: 0x07
          xid: 0x0001.020.00000ac9
ktubu redo: slt: 32 rci: 6 opc: 10.22 objn: 96090 objd: 96090 tsn: 4
Undo type: Regular undo          Undo type: Last buffer split: No
Tablespace Undo: No
          0x00000000
index undo for leaf key operations
KTB Redo
op: 0x04 ver: 0x01
compat bit: 4 (post-11) padding: 1
op: L itl: xid: 0xffff.000.00000000 uba: 0x00000000.0000.00
          flg: C--- lkc: 0          scn: 0x0000.01d10560
Dump kdilk : itl=2, kdxlkflg=0x1 sdc=32767 indexid=0x1848402 block=0x01848411
(kdxlpu): purge leaf row
key :(18): 0a 5a 55 4c 4d 41 41 41 52 4f 4e 06 01 84 83 ee 00 05
```

Update to indexed column (2)

Right most child



```
CHANGE #2 CON_ID:0 TYP:0 CLS:1 AFN:6 DBA:0x01848411 OBJ:96090  
SCN:0x0000.01d10567 SEQ:2 OP:10.2 ENC:0 RBL:0 FLG:0x0000  
index redo (kdxlin): insert leaf row  
KTB Redo  
op: 0x01 ver: 0x01  
compat bit: 4 (post-11) padding: 1  
op: F xid: 0x0001.020.00000ac9 uba: 0x01000781.0170.07  
REDO: SINGLE / -- / --  
itl: 2, sno: 389, row size 22  
insert key: (18): 0a 5a 55 4c 4d 41 41 41 52 4f 4e 06 01 84 83 ee 00 05
```

All row delete

```
row#0[8017] flag: ---D---, lock: 2, len=15
col 0; len 5; (5): 41 41 52 4f 4e
col 1; len 6; (6): 01 84 83 ee 00 05
row#1[8002] flag: ---D---, lock: 2, len=15
col 0; len 5; (5): 41 42 42 45 59
col 1; len 6; (6): 01 84 83 e6 00 9f
...
row#395[1680] flag: ---D---, lock: 2, len=17
col 0; len 7; (7): 42 41 42 45 54 54 45
col 1; len 6; (6): 01 84 83 eb 00 6a
row#396[1664] flag: ---D---, lock: 2, len=16
col 0; len 6; (6): 42 41 49 4c 45 59
col 1; len 6; (6): 01 84 a5 70 00 24
row#397[1649] flag: ---D---, lock: 2, len=15
col 0; len 5; (5): 42 41 4d 42 49
col 1; len 6; (6): 01 84 83 e6 00 81
```

Update to same value

Updates only the table block, no index block redo.

```
CHANGE #1 CON_ID:0 TYP:0 CLS:34 AFN:4 DBA:0x010005bd OBJ:4294967295
SCN:0x0000.01db1e4e SEQ:1 OP:5.1 ENC:0 RBL:0 FLG:0x0000
ktudb redo: siz: 108 spc: 5538 flg: 0x0022 seq: 0x01be rec: 0x19
           xid: 0x0009.015.00000b50
ktubu redo: slt: 21 rci: 24 opc: 11.1 objn: 96089 objd: 96089 tsn: 4
Undo type: Regular undo           Undo type: Last buffer split: No
Tablespace Undo: No
           0x00000000
KDO undo record:
KTB Redo
op: 0x04 ver: 0x01
compat bit: 4 (post-11) padding: 1
op: L itl: xid: 0x0000.000.00000000 uba: 0x00000000.0000.00
           flg: C--- lkc: 0          scn: 0x0000.00000000
KDO Op code: URP row dependencies Disabled
  xtype: XAxtype KDO_KDOM2 flags: 0x00000080 bdba: 0x0184a57e hdba: 0x018483da
itli: 4 ispac: 0 maxfr: 4858
tabn: 0 slot: 37(0x25) flag: 0x0c lock: 0 ckix: 6
ncol: 4 nnew: 1 size: 0
Vector content:
col 1: [ 5] 41 41 52 4f 4e
```




Table block

Update to same value

```
CHANGE #2 CON_ID:0 TYP:2 CLS:1 AFN:6 DBA:0x0184a57e OBJ:96089
SCN:0x0000.01db1e1b SEQ:1 OP:11.5 ENC:0 RBL:0 FLG:0x0000
KTB Redo
op: 0x11 ver: 0x01
compat bit: 4 (post-11) padding: 1
op: F xid: 0x0009.015.00000b50 uba: 0x010005bd.01be.19
Block cleanout record, scn: 0x0000.01db1e4e ver: 0x01 opt: 0x02, entries
follow...
  itli: 3 flg: (opt=2 whr=1) scn: 0x0000.01db1e1b
KDO Op code: URP row dependencies Disabled
  xtype: XAtype KDO_KDOM2 flags: 0x00000080 bdba: 0x0184a57e hdba:
0x018483da
itli: 4 ispac: 0 maxfr: 4858
tabn: 0 slot: 37(0x25) flag: 0x0c lock: 4 ckix: 6
ncol: 4 nnew: 1 size: 0
Vector content:
col 1: [ 5] 41 41 52 4f 4e
```

Row chaining

```
SQL> select rowid from rs.test1 where v1='AARON';
```

```
ROWID
```

```
-----  
AAAXdZAAGAABIPuAAF
```

```
SQL> update rs.test1 set v2=rpad(v2,1000,v2) where v1='AARON';
```

```
1 row updated.
```

Row chaining

```
declare
    h_rowid_type      number;
    h_object_number   number;
    h_relative_fno    number;
    h_block_number    number;
    h_row_number      number;
    h_rowid           rowid;
begin
    select rowid into h_rowid from rs.test1 where v1='AARON';
    dbms_rowid.rowid_info(h_rowid,h_rowid_type,h_object_number,
                          h_relative_fno, h_block_number, h_row_number);
    dbms_output.put_line('type      ==>'||to_char(h_rowid_type));
    dbms_output.put_line('object #   ==>'||to_char(h_object_number));
    dbms_output.put_line('Rfno#      ==>'||to_char(h_relative_fno));
    dbms_output.put_line('block#     ==>'||to_char(h_block_number));
    dbms_output.put_line('row#      ==>'||to_char(h_row_number));
end;
/
type      ==>1
object #   ==>96089
Rfno#      ==>6
block#     ==>295918
row#       ==>5
```

Row chaining

```
SQL> @dump_block
alter system dump datafile 6 block min 295618 block max 295618
```

```
System altered.
```

```
tab 0, row 4, @0x1ec3
tl: 47 fb: --H-FL-- lb: 0x0 cc: 4
col 0: [ 3] c2 1a 2a
col 1: [ 6] 41 4e 44 52 45 41
col 2: [25]
 6a 57 6d 63 4c 69 67 65 52 73 65 51 43 59 78 78 68 77 73 79 78 6d 47 6c 4b
col 3: [ 6] 41 4e 44 52 45 41
```

```
tab 0, row 5, @0x170
tl: 9 fb: --H----- lb: 0x1 cc: 0
nrid: 0x0184a57e.25
```

```
tab 0, row 6, @0x1e8e
```

Row chaining

No index block need to be updated as the index entry will point to the head row piece.

```
tab 0, row 37, @0x102f
```

```
tl: 1028 fb: ----FL-- lb: 0x4 cc: 4
```

```
hrid: 0x018483ee.5
```

```
col 0: [ 3] c2 1a 2b
```

```
col 1: [ 5] 41 41 52 4f 4e
```

```
col 2: [1000]
```

```
54 64 66 54 42 72 59 76 69 44 41 74 58 77 79 70 52 71 75 44 57 53 79 62 4d
58 75 4b 4f 75 4f 46 46 71 43 54 64 66 54 42 72 59 76 69 44 41 74 58 77 79
70 52 71 75 44 57 53 79 62 4d 58 75 4b 4f 75 4f 46 46 71 43 54 64 66 54 42
72 59 76 69 44 41 74 58 77 79 70 52 71 75 44 57 53 79 62 4d 58 75 4b 4f 75
4f 46 46 71 43 54 64 66 54 42 72 59 76 69 44 41 74 58 77 79 70 52 71 75 44
57 53 79 62 4d 58 75 4b 4f 75 4f 46 46 71 43 54 64 66 54 42 72 59 76 69 44
```

```
...
```


Row chaining

```
CHANGE #2 CON_ID:0 TYP:0 CLS:1 AFN:6 DBA:0x0184a57e OBJ:96089
SCN:0x0000.01db225c SEQ:1 OP:11.5 ENC:0 RBL:0 FLG:0x0000
KTB Redo
op: 0x01 ver: 0x01
compat bit: 4 (post-11) padding: 1
op: F xid: 0x0008.00a.00000bd2 uba: 0x010002f9.0209.1a
KDO Op code: URP row dependencies Disabled
  xtype: XA flags: 0x00000000 bdba: 0x0184a57e hdba: 0x018483da
itli: 4 ispac: 0 maxfr: 4858
tabn: 0 slot: 37(0x25) flag: 0x0c lock: 4 ckix: 0
ncol: 4 nnew: 1 size: 967
col 2: [1000]
 54 64 66 54 42 72 59 76 69 44 41 74 58 77 79 70 52 71 75 44 57 53 79 62 4d
 58 75 4b 4f 75 4f 46 46 71 43 54 64 66 54 42 72 59 76 69 44 41 74 58 77 79
 70 52 71 75 44 57 53 79 62 4d 58 75 4b 4f 75 4f 46 46 71 43 54 64 66 54 42
 72 59 76 69 44 41 74 58 77 79 70 52 71 75 44 57 53 79 62 4d 58 75 4b 4f 75
 4f 46 46 71 43 54 64 66 54 42 72 59 76 69 44 41 74 58 77 79 70 52 71 75 44
```

Reverse key indexes

```
row#0[4475] flag: ----S--, lock: 2, len=15
col 0; len 5; (5): 41 42 45 48 52
col 0; len 5; (5):  A B E H R +++
col 1; len 6; (6): 01 84 a5 72 00 6b
row#1[4490] flag: ----S--, lock: 2, len=15
col 0; len 5; (5): 41 42 45 48 53
col 0; len 5; (5):  A B E H S +++
col 1; len 6; (6): 01 84 a5 72 00 14
row#2[4505] flag: ----S--, lock: 2, len=14
col 0; len 4; (4): 41 42 45 52
col 0; len 4; (4):  A B E R +++
col 1; len 6; (6): 01 84 83 db 00 99
row#3[4519] flag: ----S--, lock: 2, len=14
col 0; len 4; (4): 41 42 4c 41
col 0; len 4; (4):  A B L A +++
col 1; len 6; (6): 01 84 83 e0 00 a3
row#4[4533] flag: ----S--, lock: 2, len=17
col 0; len 7; (7): 41 42 4c 41 53 4f 52
col 0; len 7; (7):  A B L A S O R +++
col 1; len 6; (6): 01 84 83 e3 00 7e
row#5[4550] flag: ----S--, lock: 2, len=14
col 0; len 4; (4): 41 42 4c 45
col 0; len 4; (4):  A B L E +++
col 1; len 6; (6): 01 84 83 e1 00 42
```

In reverse key indexes, values are reversed and stored. Other structures are similar to a normal index.



Bitmap index

```
----- begin tree dump  
leaf: 0x1848473 25461875 (0: row:2.2 avs:6461)  
----- end tree dump
```

Bitmap index - 1 row insert

row#0[7269] flag: -----, lock: 0, len=763

col 0; len 1; (1): 46

col 1; len 6; (6): 01 84 84 63 00 00

col 2; len 6; (6): 01 84 84 6d 00 07

col 3; len 743; (743):

cf a5 d3 45 f1 0f c8 f0 cd cf 46 53 a9 e9 0c 61 bb 78 cf 9f 33 d6 a0 39 b3

1e 3d cf 10 f0 c5 cc bf e9 07 50 cf a9 4f b3 f4 60 77 56 fe cf 8c f5 69 91

6e f9 1c 8d cf 4d 1b e5 a4 85 37 59 29 cf bb 81 7b 8f 0b 05 14 89 c8 1d ff

16 c9 b6 b9 0e 35 e6 60 30 cf b4 de ad e2 68 3d 9e 24 cf 94 8c b9 d8 1b 1e

31 df cf 12 ef 34 a9 58 69 b6 4f cf 2d 92 35 f9 46 a9 df 8b cf c3 ae 09 2a

56 a7 77 3f cf eb 65 bf 2e b5 ec 9b 41 cf 9f f4 20 8d ec 6f 56 4e c8 19 ff

16 fa 54 7b 34 79 61 9b fb cf 65 0c 43 70 c8 fd a9 c5 cf b0 5a 29 14 25 15

...

fa e5 64 c2 62 00 72 cf 43 e3 10 40 d4 5f 69 57 cf 64 6d 67 3a 4d e9 2b af

cf f3 86 8a 36 88 d1 b4 66 cf 8a d9 0d 1a 11 7a dc db cf 41 a0 e0 54 10 14

83 0d cf e0 c4 d0 f5 44 d2 0c e8 cf 01 a2 f6 c4 36 fc 80 1c ff 17 70 5c b8

c8 92 54 b1 94 cf b3 85 bc 47 24 d8 d2 a7 cf 86 79 d6 d1 6d 6f 4c 68 cf 30

3a df 1c a9 ba 84 a6 cf 74 5d 7c 66 82 96 0a 8a cf 93 a0 79 5b 03 d0 a7 e4

cf c7 0a 4c 0d f5 e7 e8 f1 cf ad 53 34 54 66 2b 3f 09 ff 17 21 db 82 a6 1b

2a 6d e3 cf c3 34 dd d6 56 99 24 74 cf 96 c1 ce b6 a1 74 3c 74 cf 3a 39 24

ab d9 4d e7 df cf b8 52 3d ff 28 10 4f b0 cf f3 dc d8 9e 3b f5 cc e0 cf 2d

26 bb b1 60 62 db 97 cf 55 0c 7c 0d 1f c0 fa 28 c0 05

Bitmap index

row#0[7269] flag: -----, lock: 0, len=763

col 0; len 1; (1): 46

col 1; len 6; (6): 01 84 84 63 00 00

col 2; len 6; (6): 01 84 84 **6f** 00 07

col 3; len 743; (743):

cf a5 d3 45 f1 0f c8 f0 cd cf 46 53 a9 e9 0c 61 bb 78 cf 9f 33 d6 a0 39 b3

1e 3d cf 10 f0 c5 cc bf e9 07 50 cf a9 4f b3 f4 60 77 56 fe cf 8c f5 69 91

6e f9 1c 8d cf 4d 1b e5 a4 85 37 59 29 cf bb 81 7b 8f 0b 05 14 89 c8 1d ff

16 c9 b6 b9 0e 35 e6 60 30 cf b4 de ad e2 68 3d 9e 24 cf 94 8c b9 d8 1b 1e

31 df cf 12 ef 34 a9 58 69 b6 4f cf 2d 92 35 f9 46 a9 df 8b cf c3 ae 09 2a

56 a7 77 3f cf eb 65 bf 2e b5 ec 9b 41 cf 9f f4 20 8d ec 6f 56 4e c8 19 ff

16 fa 54 7b 34 79 61 9b fb cf 65 0c 43 70 c8 fd a9 c5 cf b0 5a 29 14 25 15

...

fa e5 64 c2 62 00 72 cf 43 e3 10 40 d4 5f 69 57 cf 64 6d 67 3a 4d e9 2b af

cf f3 86 8a 36 88 d1 b4 66 cf 8a d9 0d 1a 11 7a dc db cf 41 a0 e0 54 10 14

83 0d cf e0 c4 d0 f5 44 d2 0c e8 cf 01 a2 f6 c4 36 fc 80 1c ff 17 70 5c b8

c8 92 54 b1 94 cf b3 85 bc 47 24 d8 d2 a7 cf 86 79 d6 d1 6d 6f 4c 68 cf 30

3a df 1c a9 ba 84 a6 cf 74 5d 7c 66 82 96 0a 8a cf 93 a0 79 5b 03 d0 a7 e4

cf c7 0a 4c 0d f5 e7 e8 f1 cf ad 53 34 54 66 2b 3f 09 ff 17 21 db 82 a6 1b

2a 6d e3 cf c3 34 dd d6 56 99 24 74 cf 96 c1 ce b6 a1 74 3c 74 cf 3a 39 24

ab d9 4d e7 df cf b8 52 3d ff 28 10 4f b0 cf f3 dc d8 9e 3b f5 cc e0 cf 2d

26 bb b1 60 62 db 97 cf 55 0c 7c 0d 1f c0 fa 28 c0 05 **c1 a1 01**

Bitmap index

Undo change vector stores the bitmap in the undo.

REDO RECORD - Thread:1 RBA: 0x000101.00000003.0014 LEN: 0x03e8 VLD: 0x01
CON_UID: 0

SCN: 0x0000.01e98bd0 SUBSCN: 1 01/08/2017 13:12:10

CHANGE #1 CON_ID:0 TYP:0 CLS:20 AFN:4 DBA:0x01001dac OBJ:4294967295

SCN:0x0000.01e98bd0 SEQ:1 OP:5.1 ENC:0 RBL:0 FLG:0x0000

ktudb redo: siz: 852 spc: 1616 flg: 0x0022 seq: 0x01b8 rec: 0x2c
xid: 0x0002.005.00000b7e

ktubu redo: slt: 5 rci: 43 opc: 10.22 objn: 96328 objd: 96328 tsn: 4

Undo type: Regular undo Undo type: Last buffer split: No

Tablespace Undo: No

0x00000000

index undo for leaf key operations

KTB Redo

op: 0x04 ver: 0x01

compat bit: 4 (post-11) padding: 1

op: L itl: xid: 0xffff.000.00000000 uba: 0x00000000.0000.00

flg: C--- lkc: 0 scn: 0x0000.01e98b98

Dump kdilk : itl=2, kdxlkflg=0x1 sdc=0 indexid=0x1848472 block=0x01848473

(kdxlup): update keydata in row

key :(761):

```
01 46 06 01 84 84 63 00 00 06 01 84 84 6d 00 07 82 e7 cf a5 d3 45 f1 0f c8
f0 cd cf 46 53 a9 e9 0c 61 bb 78 cf 9f 33 d6 a0 39 b3 1e 3d cf 10 f0 c5 cc
bf e9 07 50 cf a9 4f b3 f4 60 77 56 fe cf 8c f5 69 91 6e f9 1c 8d cf 4d 1b
...
52 3d ff 28 10 4f b0 cf f3 dc d8 9e 3b f5 cc e0 cf 2d 26 bb b1 60 62 db 97
cf 55 0c 7c 0d 1f c0 fa 28 c0 05
```

Bitmap index

Undo change record

```
CHANGE #1 CON_ID:0 TYP:0 CLS:20 AFN:4 DBA:0x01001dac OBJ:4294967295
SCN:0x0000.01e98bd0 SEQ:2 OP:5.1 ENC:0 RBL:0 FLG:0x0000
ktudb redo: siz: 124 spc: 762 flg: 0x0022 seq: 0x01b8 rec: 0x2d
          xid: 0x0002.005.00000b7e
ktubu redo: slt: 5 rci: 44 opc: 10.37 objn: 96328 objd: 96328 tsn: 4
Undo type: Regular undo          Undo type: Last buffer split: No
Tablespace Undo: No
          0x00000000
index undo for leaf key operations
index change (kdxIndexLogicalNonkeyUpdate): count=6
KTB Redo
op: 0x02 ver: 0x01
compat bit: 4 (post-11) padding: 1
op: C uba: 0x01001dac.01b8.2c
Dump kdilk : len=29 != sizeof(kdilk)=20 :(29):
 26 02 05 c7 72 84 84 01 73 84 84 01 00 00 00 00 00 00 6f c7 00 04 02 02 fd
 ff 00 00 00
itl=2, kdxlkflg=0x5 sdc=0 indexid=0x1848472 block=0x01848473
(kdxIndexLogicalNonkeyUpdate): index logical nonkey update
ncol: 4 nvec: 2 nnew: 2 size: -3
unique key: (9): 01 46 06 01 84 84 63 00 00
logical nonkey columns operation records:
column 2:
  atom same length replace: from offset 0 replace 6 bytes:
  01 84 84 6d 00 07
column 3:
  atom replace: from offset 741 replace 5 bytes with 2 bytes:
  c0 05
```


Bitmap index

CHANGE #2 CON_ID:0 TYP:0 CLS:1 AFN:6 DBA:0x01848473 OBJ:96328

SCN:0x0000.01e98bd0 SEQ:1 OP:10.38 ENC:0 RBL:0 FLG:0x0000

index change (kdxIndexlogicalNonkeyUpdate): count=6

KTB Redo

op: 0x02 ver: 0x01

compat bit: 4 (post-11) padding: 1

op: C uba: 0x01001dac.01b8.2d

ERROR: bad kdxlk len 13

REDO: SINGLE / -- / --

itl: 2, sno: 0, row size 765

ncol: 4 nvec: 2 nnew: 2 size: 3

unique key: (9): 01 46 06 01 84 84 63 00 00

logical nonkey columns operation records:

column 2:

atom same length replace: from offset 0 replace 6 bytes:

| |
|-------------------|
| 01 84 84 6f 00 07 |
|-------------------|

column 3:

atom replace: from offset 741 replace 2 bytes with 5 bytes:

| |
|----------------|
| c0 05 c1 a1 01 |
|----------------|

Bitmap updates

Updates to 10 rows modified the bitmap,
leaving the rowid ranges intact.

```
<  cf a5 d3 45 f1 0f c8 f0 cd cf 46 53 a9 e9 0c 61 bb 78 cf 9f 33 d6 a0 39 b3
---
>  cf ff ff 5f f1 0f c8 f0 cd cf 46 53 a9 e9 0c 61 bb 78 cf 9f 33 d6 a0 39 b3
63c63
< row#1[4201] flag: -----, lock: 2, len=768
---
> row#1[2667] flag: -----, lock: 2, len=768
68c68
<  cf 5a 2c ba 0e f0 37 0f 32 cf b9 ac 56 16 f3 9e 44 87 cf 60 cc 29 5f c6 4c
---
>  cf 00 00 a0 0e f0 37 0f 32 cf b9 ac 56 16 f3 9e 44 87 cf 60 cc 29 5f c6 4c
```

Index splits

Once a root block, always a root block..

```
leaf: 0x1848483 25461891 (0: row:2.2 avs:3970)
```

```
branch: 0x1848483 25461891 (0: nrow: 2, level: 1)
  leaf: 0x1848485 25461893 (-1: row:3.3 avs:1957)
  leaf: 0x1848486 25461894 (0: row:1.1 avs:5983)
```

```
----- begin tree dump
```

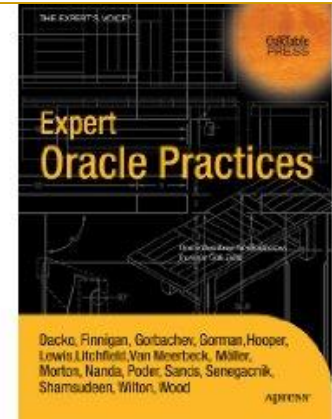
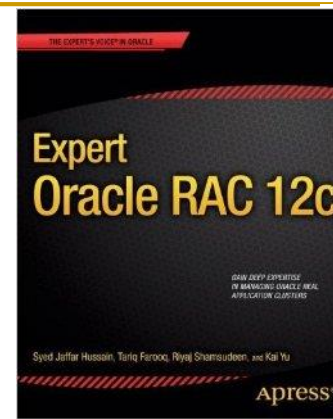
```
branch: 0x1848483 25461891 (0: nrow: 3, level: 2)
  branch: 0x1848496 25461910 (-1: nrow: 4, level: 1)
    leaf: 0x1848485 25461893 (-1: row:2.2 avs:3970)
    leaf: 0x1848493 25461907 (0: row:2.2 avs:3970)
    leaf: 0x1848491 25461905 (1: row:1.1 avs:5983)
    leaf: 0x1848486 25461894 (2: row:3.3 avs:1957)
  branch: 0x1848492 25461906 (0: nrow: 2, level: 1)
    leaf: 0x1848487 25461895 (-1: row:3.3 avs:1957)
    leaf: 0x1848484 25461892 (0: row:3.3 avs:1957)
  branch: 0x1848497 25461911 (1: nrow: 2, level: 1)
    leaf: 0x1848495 25461909 (-1: row:3.3 avs:1957)
    leaf: 0x1848490 25461904 (0: row:3.3 avs:1957)
```

Index splits

Index splits have many steps:

1. Setup ITL in the first ITL entry for recursive transaction.
 2. Lock the block – kdxcolok
 3. Dump the whole block into the redo stream.
 4. Search for free block.
 5. Format the acquired block(s)
 6. Setup leaf blocks
 7. Setup branch blocks
-

THANK YOU



- Email: rshamsud@orainternals.com
- Blog : orainternals.wordpress.com
- Web: www.orainternals.com

