*** Example 2: Compression of just one column in a concatenated index (with a very selective column second)

*** Note the ID column is very selective (virtually unique in fact) and the OWNER column has relatively few distinct values

```
SQL> CREATE TABLE compression_test2 AS SELECT rownum id, ct.* FROM compression_test ct ORDER BY owner, num_rows;

Table created.


*** Note: COMPRESS 1 meaning only the leading column (owner) is actually compressed


SQL> CREATE INDEX compressed_index2 ON compression_test2(owner, id) PCTFREE 0 COMPRESS 1;

Index created.

SQL> ANALYZE INDEX compressed_index2 VALIDATE STRUCTURE;

Index analyzed.

SQL> SELECT HEIGHT, BR_BLKS, LF_BLKS FROM INDEX_STATS;

    HEIGHT    BR_BLKS    LF_BLKS
---------- ---------- ----------
         2          1        110


*** Note "just" 110 leaf blocks ..


*** Partial index leaf block dump


Leaf block dump
===============
header address 719987292=0x2aea225c
kdxcolev 0
KDXCOLEV Flags = - - -
kdxcolok 0
kdxcoopc 0xa0: opcode=0: iot flags=-C- is converted=Y
kdxconco 3
kdxcosdc 0
kdxconro 524
kdxcofbo 1092=0x444
kdxcofeo 1102=0x44e
kdxcoavs 10
kdxlespl 0
kdxlende 0
kdxlenxt 75545484=0x480bb8c
kdxleprv 0=0x0
kdxledsz 0
kdxlebksz 8036
kdxlepnro 1
kdxlepnco 1
prefix row#0[8025] flag: -P----, lock: 0, len=11      <== Note there is only the one index
prefix entry
col 0; len 8; (8):  41 46 50 31 31 35 35 34
prc 524
row#0[8012] flag: ------, lock: 0, len=13
col 0; len 3; (3):  c2 05 0e
col 1; len 6; (6):  04 80 c5 c5 00 0e
psno 0                                                <== Note all index entries refer to the
same and only prefix entry number
row#1[7999] flag: ------, lock: 0, len=13
col 0; len 3; (3):  c2 05 0f
col 1; len 6; (6):  04 80 c5 99 00 1f
psno 0
row#2[7986] flag: ------, lock: 0, len=13
col 0; len 3; (3):  c2 05 10
col 1; len 6; (6):  04 80 c5 9d 00 06
psno 0
row#3[7973] flag: ------, lock: 0, len=13
col 0; len 3; (3):  c2 05 11
col 1; len 6; (6):  04 80 c5 bd 00 13
psno 0

.
.
.

row#521[1130] flag: ------, lock: 0, len=14
col 0; len 4; (4):  c3 02 1b 4a
```

```
col 1; len 6; (6):  04 80 c5 8b 00 04
psno 0
row#522[1116] flag: ------, lock: 0, len=14
col 0; len 4; (4):  c3 02 1b 4b
col 1; len 6; (6):  04 80 c5 b4 00 08
psno 0
row#523[1102] flag: ------, lock: 0, len=14
col 0; len 4; (4):  c3 02 1b 4c
col 1; len 6; (6):  04 80 c5 b1 00 0c
psno 0                                                    <=== All 524 index entries
refer to prefix entry #0
----- end of leaf block dump -----
End dump data blocks tsn: 21 file#: 18 minblk 48011 maxblk 48011


*** Note that the prefix table only has the one entry in this leaf block meaning all 524
index entries use this one prefix entry

*** Compression has been very successful as we are only storing just the one occurance of the
OWNER column in this particular leaf block



*** Example 3: Compression of just one column (with selective column first)

*** Note this time the very selective ID column is the leading column in the index ...


SQL> CREATE INDEX compressed_index2_2 ON compression_test2(id, owner) PCTFREE 0 COMPRESS 1;

Index created.

SQL> ANALYZE INDEX compressed_index2_2 VALIDATE STRUCTURE;

Index analyzed.

SQL> SELECT HEIGHT, BR_BLKS, LF_BLKS FROM INDEX_STATS;

    HEIGHT    BR_BLKS    LF_BLKS
---------- ---------- ----------
         2          1        192


*** Note that the number of LF_BLKS has increased significantly from 110 to 192 !!


*** Partial index leaf block dump


Leaf block dump
===============
header address 170599004=0xa2b225c
kdxcolev 0
KDXCOLEV Flags = - - -
kdxcolok 0
kdxcoopc 0xa0: opcode=0: iot flags=-C- is converted=Y
kdxconco 3
kdxcosdc 0
kdxconro 292
kdxcofbo 1792=0x700
kdxcofeo 1815=0x717
kdxcoavs 23
kdxlespl 0
kdxlende 0
kdxlenxt 75545740=0x480bc8c
kdxleprv 75545738=0x480bc8a
kdxledsz 0
kdxlebksz 8036
kdxlepnro 292
kdxlepnco 1
prefix row#0[8029] flag: -P----, lock: 0, len=7
col 0; len 4; (4):  c3 04 48 4d
prc 1
prefix row#1[8009] flag: -P----, lock: 0, len=7
col 0; len 4; (4):  c3 04 48 4e
prc 1
prefix row#2[7989] flag: -P----, lock: 0, len=7
col 0; len 4; (4):  c3 04 48 4f
prc 1
prefix row#3[7969] flag: -P----, lock: 0, len=7
col 0; len 4; (4):  c3 04 48 50
prc 1
prefix row#4[7949] flag: -P----, lock: 0, len=7
col 0; len 4; (4):  c3 04 48 51
```

```
.
.
.


prefix row#290[1860] flag: -P----, lock: 0, len=7
col 0; len 4; (4):  c3 04 4b 43
prc 1
prefix row#291[1834] flag: -P----, lock: 0, len=7           <== Note this time we have a
massive 292 prefix index entries
col 0; len 4; (4):  c3 04 4b 44
prc 1
row#0[8016] flag: ------, lock: 0, len=13
col 0; len 3; (3):  53 59 53
col 1; len 6; (6):  04 80 cf 6b 00 05
psno 0                                                      <== Note that all index entries
refer to a different prefix entry value
row#1[7996] flag: ------, lock: 0, len=13
col 0; len 3; (3):  53 59 53
col 1; len 6; (6):  04 80 c7 67 00 18
psno 1
row#2[7976] flag: ------, lock: 0, len=13
col 0; len 3; (3):  53 59 53
col 1; len 6; (6):  04 80 c7 67 00 19
psno 2
row#3[7956] flag: ------, lock: 0, len=13
col 0; len 3; (3):  53 59 53
col 1; len 6; (6):  04 80 c7 67 00 1a
psno 3

.
.
.

row#290[1841] flag: ------, lock: 0, len=19
col 0; len 9; (9):  58 59 54 48 4f 53 5f 44 53
col 1; len 6; (6):  04 80 bb 15 00 0b
psno 290
row#291[1815] flag: ------, lock: 0, len=19
col 0; len 9; (9):  58 59 54 48 4f 53 5f 44 53
col 1; len 6; (6):  04 80 bb 15 00 08
psno 291                                                   <== In fact there are as many
prefix entries as there are index row entries and we can only fit in 292 index entries
----- end of leaf block dump -----
End dump data blocks tsn: 21 file#: 18 minblk 48267 maxblk 48267


*** Note there are 292 prefix entries and 292 index row entries. Therefore, each prefix entry
corresponds to just one index row entry ...

*** In the previous example, we could fit in 524 index entries, that's now been reduced to
just 292 index entries

*** In fact,if you compare it with no compression at all ...


SQL> DROP INDEX compressed_index2_2;

Index dropped.

SQL> CREATE INDEX compressed_index2_2 ON compression_test2(id, owner) PCTFREE 0 NOCOMPRESS;

Index created.

SQL> ANALYZE INDEX compressed_index2_2 VALIDATE STRUCTURE;

Index analyzed.

SQL> SELECT HEIGHT, BR_BLKS, LF_BLKS FROM INDEX_STATS;

    HEIGHT    BR_BLKS    LF_BLKS
---------- ---------- ----------
         2          1        150


*** The number of LF_BLKS drops down to 150 from 192. Compression has actually caused the
index to get bigger not smaller !!


*** Note compressing both columns in the index makes no difference and is highly inefficient
as well as the leading column is not selective

*** All row entries correspond to only the one index prefix row entry


SQL> DROP INDEX compressed_index2_2;
```

```
Index dropped.


*** Both columns are compressed (default is all index columns compressed, unless index is
unique in which case it's indexed columns - 1)

SQL> CREATE INDEX compressed_index2_2 ON compression_test2(id, owner) PCTFREE 0 COMPRESS;

Index created.

SQL> ANALYZE INDEX compressed_index2_2 VALIDATE STRUCTURE;

Index analyzed.

SQL> SELECT HEIGHT, BR_BLKS, LF_BLKS FROM INDEX_STATS;

    HEIGHT    BR_BLKS    LF_BLKS
---------- ---------- ----------
         2          1        192


*** Hummmmm, back to 192 leaf blocks ...


*** Partial index leaf block dump


Leaf block dump
===============
header address 170599004=0xa2b225c
kdxcolev 0
KDXCOLEV Flags = - - -
kdxcolok 0
kdxcoopc 0xa0: opcode=0: iot flags=-C- is converted=Y
kdxconco 3
kdxcosdc 0
kdxconro 292
kdxcofbo 1792=0x700
kdxcofeo 1815=0x717
kdxcoavs 23
kdxlespl 0
kdxlende 0
kdxlenxt 75545740=0x480bc8c
kdxleprv 75545738=0x480bc8a
kdxledsz 0
kdxlebksz 8036
kdxlepnro 292
kdxlepnco 2
prefix row#0[8025] flag: -P----, lock: 0, len=11
col 0; len 4; (4):  c3 04 48 4d
col 1; len 3; (3):  53 59 53
prc 1
prefix row#1[8005] flag: -P----, lock: 0, len=11
col 0; len 4; (4):  c3 04 48 4e
col 1; len 3; (3):  53 59 53
prc 1
prefix row#2[7985] flag: -P----, lock: 0, len=11
col 0; len 4; (4):  c3 04 48 4f
col 1; len 3; (3):  53 59 53
prc 1

.
.
.

prefix row#290[1850] flag: -P----, lock: 0, len=17
col 0; len 4; (4):  c3 04 4b 43
col 1; len 9; (9):  58 59 54 48 4f 53 5f 44 53
prc 1
prefix row#291[1824] flag: -P----, lock: 0, len=17
col 0; len 4; (4):  c3 04 4b 44
col 1; len 9; (9):  58 59 54 48 4f 53 5f 44 53
prc 1
row#0[8016] flag: ------, lock: 0, len=9
col 0; len 6; (6):  04 80 cf 6b 00 05
psno 0
row#1[7996] flag: ------, lock: 0, len=9
col 0; len 6; (6):  04 80 c7 67 00 18
psno 1
row#2[7976] flag: ------, lock: 0, len=9
col 0; len 6; (6):  04 80 c7 67 00 19
psno 2

.
.
.
```

```
row#290[1841] flag: ------, lock: 0, len=9
col 0; len 6; (6):  04 80 bb 15 00 0b
psno 290
row#291[1815] flag: ------, lock: 0, len=9
col 0; len 6; (6):  04 80 bb 15 00 08
psno 291
----- end of leaf block dump -----
End dump data blocks tsn: 21 file#: 18 minblk 48267 maxblk 48267
```

*** Again, one prefix entry for each and every row entry.

*** Compression has been pointless and in fact uses more space than without compression as it needs to store all the index values anyways plus additional overheads such as the pointers in the index row entries