

\*\*\* Example 4: Compression of a Unique Index

```
SQL> DROP INDEX compressed_index2_2;
```

Index dropped.

```
SQL> CREATE UNIQUE INDEX compressed_index_uk ON compression_test2(id)
PCTFREE 0 COMPRESS;
CREATE UNIQUE INDEX compressed_index_uk ON compression_test2(id) PCTFREE 0
COMPRESS
```

\*

```
ERROR at line 1:
ORA-25193: cannot use COMPRESS option for a single column key
```

\*\*\* This makes sense as previously shown; compressing an index in which keys don't repeat within a block can cause the index to be less efficient than normal

```
SQL> ALTER TABLE compression_test2 ADD PRIMARY KEY (id) USING INDEX (CREATE
INDEX compressed_index_uk ON compression_test2(id) PCTFREE 0 COMPRESS);
```

Table altered.

\*\*\* However, you still can't prevent stupidity !!

\*\*\* Oracle allows you to create a compressed index on a single column non-unique index used to police a primary key or unique key constraint ...

```
SQL> DROP INDEX compressed_index_uk;
```

Index dropped.

\*\*\* Unique indexes have exactly the same compression issues regarding the importance of column order to the effectiveness of compression

\*\*\* Create a concatenated unique index with a highly selective leading column

\*\*\* Note, only the ID column is compressed

```
SQL> CREATE UNIQUE INDEX compressed_index_uk ON compression_test2(id,
owner) PCTFREE 0 COMPRESS;
```

Index created.

```
SQL> ANALYZE INDEX compressed_index_uk VALIDATE STRUCTURE;
```

Index analyzed.

```
SQL> SELECT HEIGHT, BR_BLKS, LF_BLKS FROM INDEX_STATS;
```

HEIGHT	BR_BLKS	LF_BLKS
2	1	185

\*\*\* Partial index leaf block dump

Leaf block dump

=====

header address 232661596=0xdde225c

kdxcolev 0

KDXCOLEV Flags = - - -

kdxcolok 0

kdxcoopc 0xa0: opcode=0: iot flags=-C- is converted=Y

kdxconco 2

kdxcosdc 0

kdxconro 300

kdxcofbo 1840=0x730

kdxcofeo 1855=0x73f

kdxcoavs 15

kdxlespl 0

kdxlende 0

kdxlenxt 75545740=0x480bc8c

kdxleprv 75545738=0x480bc8a

kdxledsz 6

kdxlebksz 8036

kdxlepno 300

kdxlepno 1

prefix row#0[8029] flag: -P----, lock: 0, len=7

col 0; len 4; (4): c3 04 56 3d

prc 1

prefix row#1[8007] flag: -P----, lock: 0, len=7

col 0; len 4; (4): c3 04 56 3e

prc 1

prefix row#2[7985] flag: -P----, lock: 0, len=7

col 0; len 4; (4): c3 04 56 3f

prc 1

.  
. .  
.

prefix row#298[1886] flag: -P----, lock: 0, len=7

col 0; len 4; (4): c3 04 59 3b

prc 1

prefix row#299[1867] flag: -P----, lock: 0, len=7

col 0; len 4; (4): c3 04 59 3c

prc 1

row#0[8014] flag: -----, lock: 0, len=15, data:(6): 04 80 c6 bc 00 05

col 0; len 6; (6): 50 52 4f 4d 49 53

psno 0

row#1[7992] flag: -----, lock: 0, len=15, data:(6): 04 80 c6 c7 00 08

col 0; len 6; (6): 50 52 4f 4d 49 53

psno 1

row#2[7970] flag: -----, lock: 0, len=15, data:(6): 04 80 c6 f2 00 1a

col 0; len 6; (6): 50 52 4f 4d 49 53

psno 2

.  
. .  
.

row#298[1874] flag: -----, lock: 0, len=12, data:(6): 04 80 c7 5c 00 14

col 0; len 3; (3): 53 59 53

```
psno 298
row#299[1855] flag: -----, lock: 0, len=12, data:(6): 04 80 c7 5c 00 15
col 0; len 3; (3): 53 59 53
psno 299
----- end of leaf block dump -----
End dump data blocks tsn: 21 file#: 18 minblk 48267 maxblk 48267
```

\*\*\* Note that only the first column is (so-called) compressed as the compress length by default is number of indexed columns - 1

\*\*\* Note also that as the leading column is highly selective, compression is very poor (non-existent)

\*\*\* However, swap the columns around and as before and compression is vastly improved as the leading column has many repeated values ...

```
SQL> DROP INDEX compressed_index_uk;
```

Index dropped.

```
SQL> DROP INDEX compressed_index2;
```

Index dropped.

\*\*\* Column OWNER is now the leading column

```
SQL> CREATE UNIQUE INDEX compressed_index_uk ON compression_test2(owner,
id) PCTFREE 0 COMPRESS;
```

Index created.

```
SQL> ANALYZE INDEX compressed_index_uk VALIDATE STRUCTURE;
```

Index analyzed.

```
SQL> SELECT HEIGHT, BR_BLKs, LF_BLKs FROM INDEX_STATS;
```

HEIGHT	BR_BLKs	LF_BLKs
2	1	103

\*\*\* Significantly fewer 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 2

kdxcosdc 0

kdxconro 559

```
kdxcofbo 1162=0x48a
kdxcofeo 1171=0x493
kdxcoavs 9
kdxlespl 0
kdxlende 0
kdxlenxt 75545484=0x480bb8c
kdxleprv 0=0x0
kdxledsz 6
kdxlebksz 8036
kdxlepnro 1
kdxlepnco 1
prefix row#0[8025] flag: -P----, lock: 0, len=11
col 0; len 8; (8): 41 46 50 31 31 35 35 34
prc 559
row#0[8013] flag: -----, lock: 0, len=12, data:(6): 04 80 c5 c5 00 0e
col 0; len 3; (3): c2 05 0e
psno 0
row#1[8001] flag: -----, lock: 0, len=12, data:(6): 04 80 c5 99 00 1f
col 0; len 3; (3): c2 05 0f
psno 0
row#2[7989] flag: -----, lock: 0, len=12, data:(6): 04 80 c5 9d 00 06
col 0; len 3; (3): c2 05 10
psno 0

.
.
.

row#557[1184] flag: -----, lock: 0, len=13, data:(6): 04 80 c5 bb 00 12
col 0; len 4; (4): c3 02 2b 51
psno 0
row#558[1171] flag: -----, lock: 0, len=13, data:(6): 04 80 c5 95 00 0f
col 0; len 4; (4): c3 02 2b 52
psno 0
----- end of leaf block dump -----
End dump data blocks tsn: 21 file#: 18 minblk 48011 maxblk 48011
```

\*\*\* And indeed, the unique index has been successfully compressed