```
*** Create a simple table and populate with some rows containing some NULL
columns

SQL> CREATE TABLE nulls_table (var1 varchar2(5), var2 varchar2(5));

Table created.

SQL> INSERT INTO nulls_table VALUES ('ABC', 'ABC');

1 row created.

SQL> INSERT INTO nulls_table VALUES ('ABC', 'XYZ');

1 row created.

SQL> INSERT INTO nulls_table VALUES ('XYZ', 'ABC');

1 row created.

SQL> INSERT INTO nulls_table VALUES ('XYZ', 'XYZ');

1 row created.

SQL> INSERT INTO nulls_table VALUES (NULL, 'ABC');

1 row created.

SQL> INSERT INTO nulls_table VALUES (NULL, 'XYZ');

1 row created.

SQL> INSERT INTO nulls_table VALUES ('ABC', NULL);

1 row created.

SQL> INSERT INTO nulls_table VALUES ('XYZ', NULL);

1 row created.

SQL> INSERT INTO nulls_table VALUES (NULL, NULL);

1 row created.

SQL> INSERT INTO nulls_table VALUES (NULL, NULL);

1 row created.

SQL> COMMIT;

Commit complete.


*** Let's select the rows and order them based on the two columns

SQL> SELECT * FROM nulls_table ORDER BY var1, var2;

VAR1  VAR2
----- -----
ABC   ABC
ABC   XYZ
ABC
```

```
XYZ   ABC
XYZ   XYZ
XYZ
      ABC
      XYZ



10 rows selected.
```

*** Note how the NULL values are logically ordered last in the output

*** Now perform a SELECT that dumps the rows into its (in this example) ASCII decimal value

```
SQL> SELECT var1, dump(var1), var2, dump(var2) FROM nulls_table ORDER BY
var1, var2;

VAR1  DUMP(VAR1)                      VAR2  DUMP(VAR2)
----- ------------------------------ ----- ------------------------------
ABC   Typ=1 Len=3: 65,66,67          ABC   Typ=1 Len=3: 65,66,67
ABC   Typ=1 Len=3: 65,66,67          XYZ   Typ=1 Len=3: 88,89,90
ABC   Typ=1 Len=3: 65,66,67                NULL
XYZ   Typ=1 Len=3: 88,89,90          ABC   Typ=1 Len=3: 65,66,67
XYZ   Typ=1 Len=3: 88,89,90          XYZ   Typ=1 Len=3: 88,89,90
XYZ   Typ=1 Len=3: 88,89,90                NULL
      NULL                           ABC   Typ=1 Len=3: 65,66,67
      NULL                           XYZ   Typ=1 Len=3: 88,89,90
      NULL                                 NULL
      NULL                                 NULL

10 rows selected.
```

*** Note the character "A" is represented by 65, "B" by 66 etc.

*** Now create an index on the var1 and var2 columns

```
SQL> CREATE INDEX nulls_table_i ON nulls_table(var1, var2);

Index created.
```

*** Let's determine the block of interest by querying dba_segments

```
SQL> SELECT header_file, header_block FROM dba_segments WHERE segment_name
= 'NULLS_TABLE_I';

HEADER_FILE HEADER_BLOCK
----------- ------------
         18       252681
```

```
*** Note this is not an ASSM tablespace so block 252681 is the index
segment header and block 252682 is our index block of interest


*** Lets dump the index block


SQL> ALTER SYSTEM DUMP DATAFILE 18 BLOCK 252682;

System altered.



*** Block dump follows


Block header dump:  0x0483db0a
 Object id on Block? Y
 seg/obj: 0x13c9c  csc: 0x01.e1e9b54b  itc: 2  flg: -  typ: 2 - INDEX
     fsl: 0  fnx: 0x0 ver: 0x01

 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
0x0001.e1e9b54b

Leaf block dump
===============
header address 83042908=0x4f3225c
kdxcolev 0
KDXCOLEV Flags = - - -
kdxcolok 0
kdxcoopc 0x80: opcode=0: iot flags=--- is converted=Y
kdxconco 3
kdxcosdc 0
kdxconro 8
kdxcofbo 52=0x34
kdxcofeo 7912=0x1ee8
kdxcoavs 7860
kdxlespl 0
kdxlende 0
kdxlenxt 0=0x0
kdxleprv 0=0x0
kdxledsz 0
kdxlebksz 8036
row#0[8019] flag: ------, lock: 0, len=17          <=== 1st index row
entry
col 0; len 3; (3):  41 42 43                       <=== This is the HEX
ASCII representation of column var1 and has a value of "ABC"
col 1; len 3; (3):  41 42 43                       <=== This is the HEX
ASCII representation of column var2 and also has a value of "ABC"
col 2; len 6; (6):  04 83 da 8a 00 00              <=== This represents
the rowid
row#1[8002] flag: ------, lock: 0, len=17          <=== 2nd index row
entry
col 0; len 3; (3):  41 42 43                       <=== var1 with a value
of "ABC"
col 1; len 3; (3):  58 59 5a                       <=== var2 with a value
of "XYZ"
col 2; len 6; (6):  04 83 da 8a 00 01              <=== 2nd rowid
```

```
row#2[7988] flag: ------, lock: 0, len=14          <=== 3rd index row
entry
col 0; len 3; (3):  41 42 43                       <=== var1 with a value
of "ABC"
col 1; NULL                                        <=== Wooohoooo, a NULL
for column var2. Note it has no length associated with it
col 2; len 6; (6):  04 83 da 8a 00 06              <=== 3rd rowid and so
...
row#3[7971] flag: ------, lock: 0, len=17
col 0; len 3; (3):  58 59 5a
col 1; len 3; (3):  41 42 43
col 2; len 6; (6):  04 83 da 8a 00 02
row#4[7954] flag: ------, lock: 0, len=17
col 0; len 3; (3):  58 59 5a
col 1; len 3; (3):  58 59 5a
col 2; len 6; (6):  04 83 da 8a 00 03
row#5[7940] flag: ------, lock: 0, len=14
col 0; len 3; (3):  58 59 5a
col 1; NULL
col 2; len 6; (6):  04 83 da 8a 00 07
row#6[7926] flag: ------, lock: 0, len=14          <=== 7th index row
entry
col 0; NULL                                        <=== var1 has a NULL
col 1; len 3; (3):  41 42 43                       <=== var2 with a value
of "ABC"
col 2; len 6; (6):  04 83 da 8a 00 04            <=== Seventh rowid
row#7[7912] flag: ------, lock: 0, len=14          <=== 8th and LAST index
row entry
col 0; NULL                                        <=== var1 has a value
of NULL and is grouped together with the other NULL value for the var1
column
col 1; len 3; (3):  58 59 5a                       <=== var2 with a value
of "XYZ"
col 2; len 6; (6):  04 83 da 8a 00 05              <=== 8th and last rowid
in the index
----- end of leaf block dump -----
End dump data blocks tsn: 21 file#: 18 minblk 252682 maxblk 252682
```

*** Note each index entry has 3 columns (starting at col 0), where col 0 => var1, col 1 => var2 and col 2 => rowid

*** The first index entry (row#3) has a col 0 => 41 42 43 (Hex) and col 1 => 41 42 43 (Hex)

*** 41 Hex = (4 x 16) + 1 = 65 decimal, 42 Hex = 66 Decimal, 43 Hex = 67 Decimal

*** Going back to our select dump, we note that the character "A" is represented by 65 decimal, "B" is 66 etc., so indeed the first index entry is the ABC, ABC as suggested in the select order by statement

*** Note the third row entry (row#2) has col 0 => ABC and col 1 => NULL is indeed NULL values are indexed and are ordered last

*** Note the 7th (row#6) and the 8th (row#7) both have col 0 => NULL again clearly showing they get ordered last

*** Note there are only 8 index entries for the 10 rows. Both rows with both var1 and var2 set to NULL are NOT indexed as expected.

*** So yes, the index entries are stored in the same order as suggested by the ORDER BY SELECT statement and all NULL index entries are not actually stored in the index.