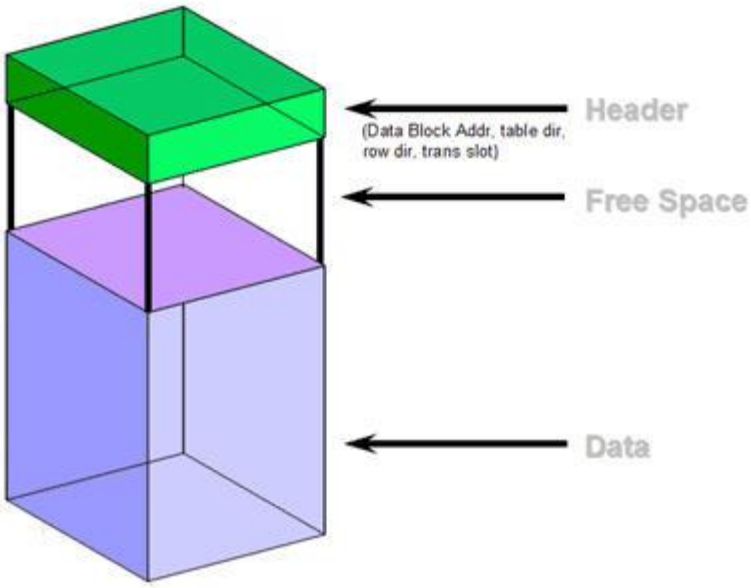
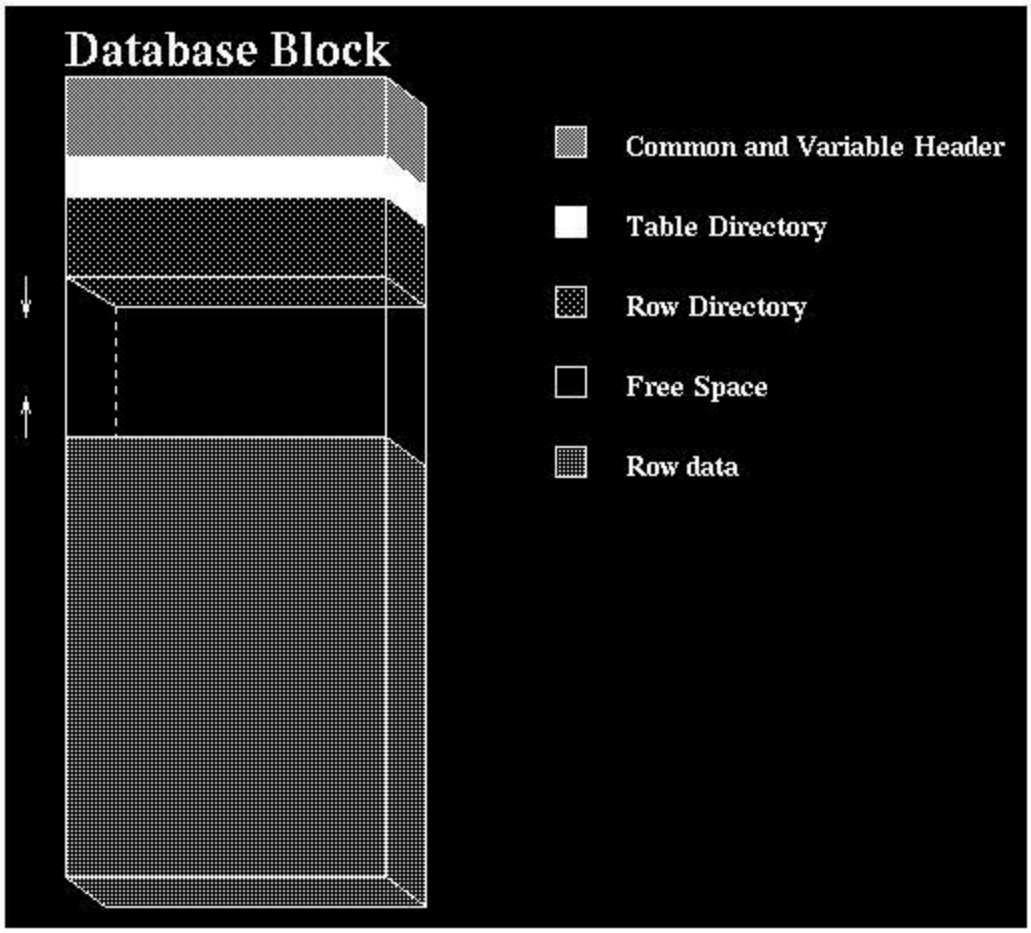
The Database Block (dbblock) has the following structure:



Another Graph with good header info:



1. **Header** (**DBA, Table Directory, Row Directory**) - The header contains general information about the data; i.e. block address and type of segments (table, index, etc.). The header also contains information about the table and the actual row (address) that holds the data.
2. **PCT Free** - Space allocated for future update/insert operations. Generally affected by the values of PCTFREE and PCTUSED parameters.
3. **PCT USED** - Actual row data.

While creating/altering any table/index, Oracle used two storage parameters for space control.

- **PCTFREE:** The percentage of space reserved for future update of existing data.
- **PCTUSED:** The percentage of minimum space used for insertion of new row data. This value determines when the block gets back into Free List table.

Row Migration

Oracle will try to shift the entire row from the current block to another block having 25 (10+15) units of free space. However, it will not remove all the relevant entries for that row from the old block. It will store the new block row ID into the old block.

Now, if I want to view that record, Oracle will internally first check the old block and then from there it will get the new row ID and display the row data from the new block. With this extra amount of I/O operation required, you likely have guessed correctly that it would degrade the performance.

Now the first question that you might ask is what is the use of maintaining the old row ID if the entire row data has been migrated from the old block to the new one? This is because of Oracle's internal mechanism -- for the entire lifespan of a row data, its row ID will never change. That's why Oracle has to maintain two row IDs -- one is because of Oracle's internal mechanism and one is for the current location of the data.

Row Chaining

What we have discussed to this point is the case where we have data in the block and new insertion is not possible into that block, which leads Oracle to go ahead and use a new block.

So what happens when a row is so large that it cannot fit into one free block? In this case, Oracle will span the data into a number of blocks so that it can hold all of the data. The existence of such data results in "Row Chaining".

Row Chaining is the storage of data in a chain of blocks. This primarily occurs in the lob, clob, blob or big varchar2 data types.

The above came from one of the following links.
http://www.akadia.com/services/ora_chained_rows.html
http://ugweb.cs.ualberta.ca/~c391/manual/chapt2.html
http://www.dbasupport.com/oracle/ora9i/rowchaining.shtml
http://www.dbasupport.com/oracle/ora9i/prevent_corruption1_1.shtml

I will be updating the doc as time permits with few cases for our testing. I will keep you updated when I do so.

--
Moid
www.MyDBANotes.com

"Success isn't a result of spontaneous combustion. You must set yourself on fire."