

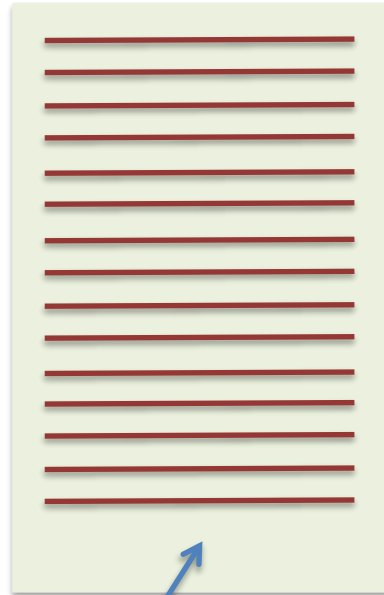
What is the fill factor?

- When we talk about indexes and/or tables, we can't escape knowing about the fill factor.
- We know that each database page (8kb in size) stores one or more rows based on the size of the row. As such, the larger the row size, the fewer rows each page can have.
- The fill-factor value determines **the percentage of space on each leaf-level page that can be filled with data**, reserving the remainder on each page as free space for future growth.
- For example, specifying **a fill-factor value of 80 means 80% of the page will have data and the remaining 20% of each leaf-level page will be left empty**, providing space for **index expansion** as data is added to the underlying table.
Default (100 or 0)

The fill-factor value determines the percentage of space on each leaf-level page to be filled with data, reserving the remainder on each page as free space for future growth. For example, specifying a fill-factor value of 80 means that 20 percent of each leaf-level page will be left empty, providing space for index expansion as data is added to the underlying table. The empty space is reserved between the index rows rather than at the end of the index.

Default (100 or 0)

Page (100 or 0)

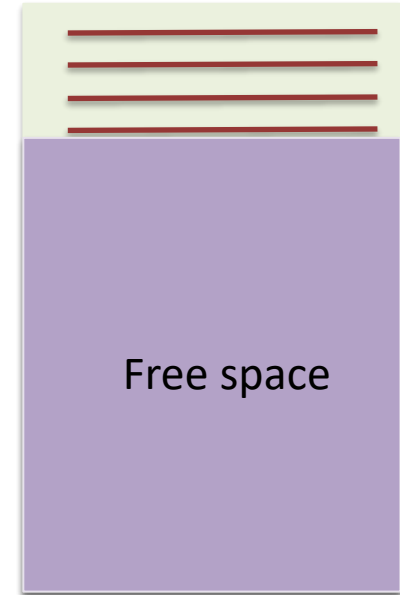


If we insert
2 more
rows, this
page will be
filled



The rows in this page will keep on being inserted until the page is full. As you can see from this example, there are probably two more rows that this page can accommodate, and then it's full

Page (20%)

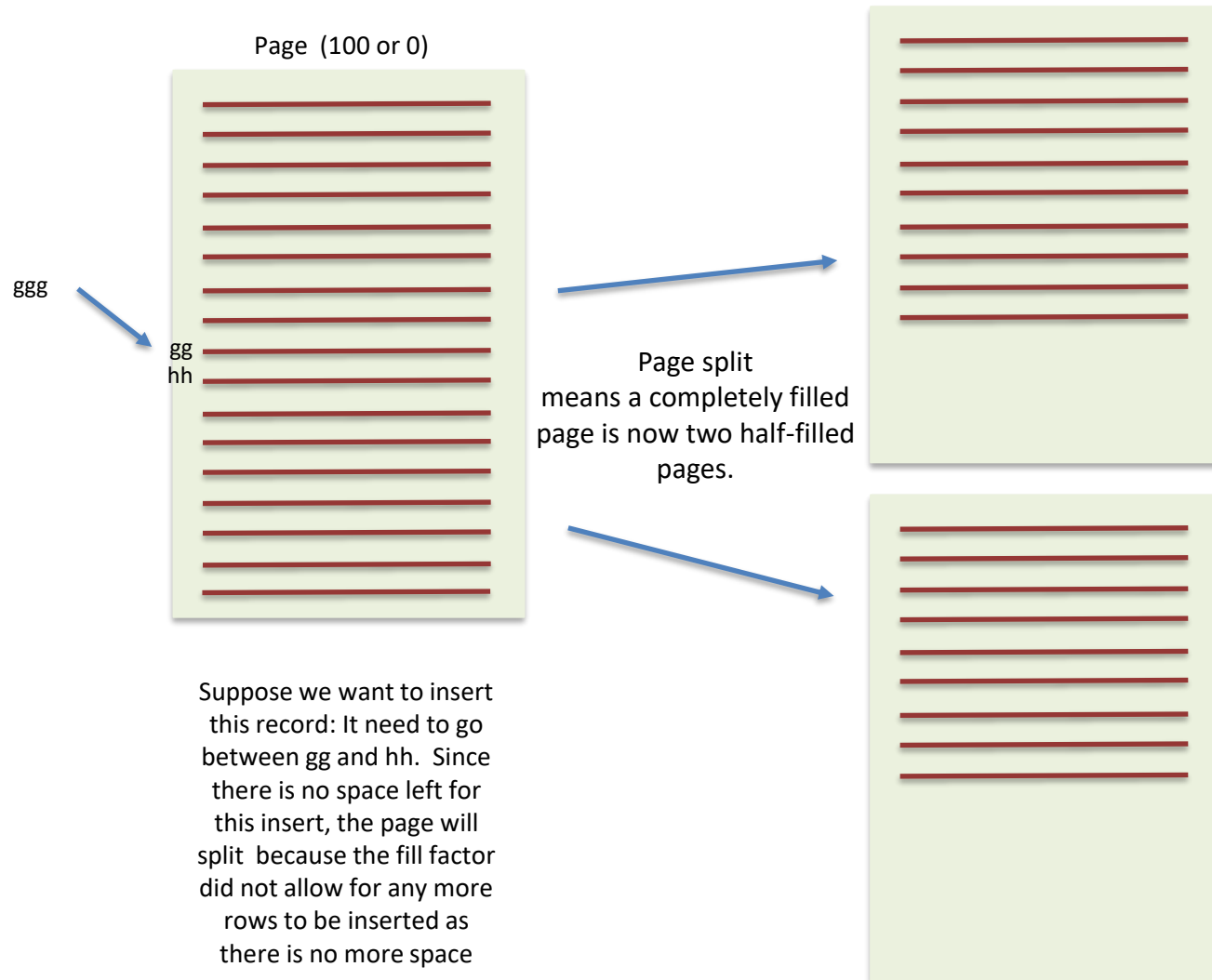


20% fill factor means that after 20% page is full, the next row will be inserted in the next page

What is the significance of understanding the fill factor?

Default (100 or 0)

The significance is this: If the page is completely filled and new data is inserted in the table which belongs to completely filled page, the “page split” event happens to accommodate new data.



The reason why we need to keep an eye on fill factor: I/O

From the last slide we can see that if we had 5000 pages that had the data for a specific index or table and we split the pages in half to accommodate the newly inserted data, the SQL engine now has to read more pages to get the same amount of data, thereby costing a greater I/O consumption.

These added pages and higher I/O can cause performance issues

If we know that a fill factor of 100 can cause page splits, then what is the optimum number we should set for fill factor?

- Depends on your environment.
- If you have a read only database then set the fill factor to 100
- If you have a database with little modification for inserts or updates then set the fill factor to 100
- If your system experiences high OLTP transactions, then consider lowering the value to 85 to 95
- This number will have to be tested (we will cover that in the upcoming videos)