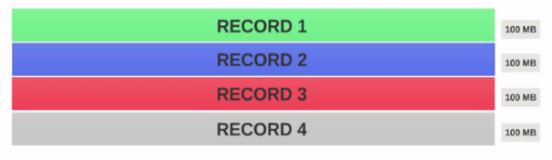
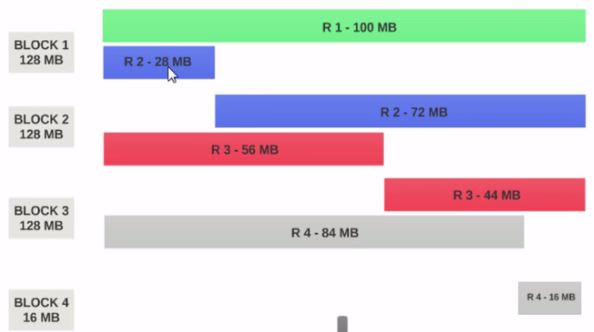
* The answer by **@user1668782** is a great explanation for the question and I'll try to give a graphical depiction of it.
* Assume we have a file of **400MB** with consists of **4 records**(**e.g** : csv file of 400MB and it has 4 rows, 100MB each)

[](https://i.stack.imgur.com/2z6Fc.jpg)

* If the HDFS **Block Size** is configured as **128MB**, then the 4 records will not be distributed among the blocks evenly. It will look like this.

[](https://i.stack.imgur.com/Lc7be.jpg)

* **Block 1** contains the entire first record and a 28MB chunk of the second record.
* If a mapper is to be run on **Block 1**, the mapper cannot process since it won't have the entire second record.
* This is the exact problem that **input splits** solve. **Input splits** respects logical record boundaries.
* Lets Assume the **input split** size is **200MB**

[](https://i.stack.imgur.com/V9xvf.jpg)

* Therefore the **input split 1** should have both the record 1 and record 2. And input split 2 will not start with the record 2 since record 2 has been assigned to input split 1. Input split 2 will start with record 3.
* This is why an input split is only a **logical chunk** of data. It points to start and end locations with in blocks.

Hope this helps.