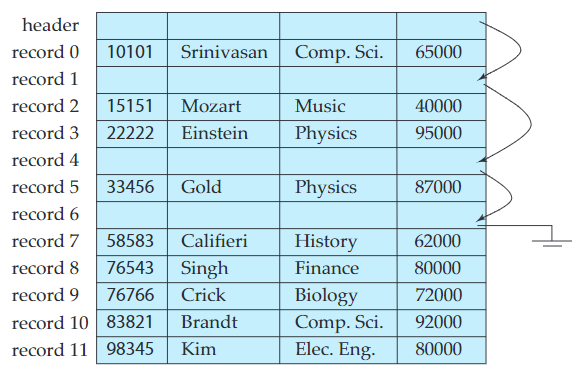
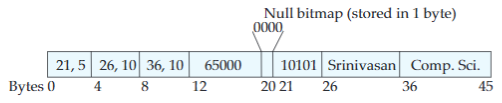
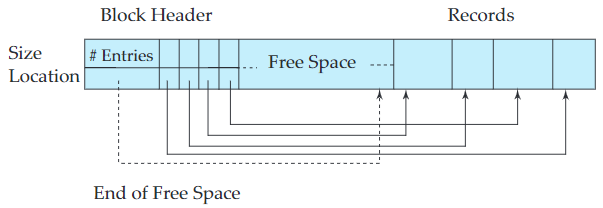
5. File Organization文件组织
   1. 一个数据库被映射到多个不同的文件，这些文件有低层的操作系统来维护。每个文件分成定长的存储单元，称为块(bolck)，块是存储分配和数据传输的基本单元。数据库默认的块在4-8块之间。通常没有记录比块更大（图片音频等大文件先不考虑），此外还要求每条记录保存在单个块中。
      1. 定长记录  
         instructor表的属性有：ID char(5); name char(20)。假设每个字符占用1字节，则每条记录占用25字节，不能将这些记录顺序存储，否则出现跨块的情况；而且删除记录时，将后面的记录依次向前移动开销很大。为了解决跨块的问题，每个块只存储整数条记录，多余的空间暂时不用。而在删除记录时，可以将最后一条记录移动到被删除记录的位置，而不是将所有后面的记录依次向前移动，此外考虑到插入操作往往比删除操作更为频繁，可以将删除记录后留出的空间给下次的插入数据占用。  
         但插入记录时，如何找到被删除记录的位置呢：在文件头部留出一定空间来存储第一条被删除记录的位置，第二条记录被删除后，其位置被记录在第一条删除记录的位置，这样依次进行，形成了图示的情况：  
           
         被删除的记录形成了一条链表(linked list)，这儿也称为空闲列表(free list)。每次新插入数据时，都插入header指向的地址，同时header中的地址更新为下一条被删除记录的地址。
      2. 变长记录
         1. 存储方式  
            数据表中包含变长属性时就会涉及到变长记录的存储。不管何种变长存储技术，都需要解决这两个基本问题：如何描述一条记录，可以方便地获取其中的属性；如何在块中存储一条记录，可以使它被方便地获取到。  
            一条有变长属性的记录通常具有两部分：定长的初始部分，然后是紧随其后的变长部分。定长部分则用(偏移量,长度)这样的格式来表示，偏移量指示了数据的起始位置，长度即为变长部分的长度。变长部分在定长部分之后连续存储。  
              
            上图示例为instructor(ID varchar(5);name varchar(20);dept\_name varchar(20);salary numeric(8,2))的存储，ID name dept\_name是变长的，假定偏移量、长度分别占用2个字节，salary是定长的，所以直接存储。  
            上图也演示了空位图(null bitmap)的使用，如果属性salary是空的，则空位图的第4为会被置1，存储salary的12-19位会被忽略。1字节的空位图可以应对最多8个属性的表结构，如果有更多属性，则需要扩展空位图的长度。这中方式以额外的空位图读取为代价，节约了存储空间，在具有很多属性同时这些属性多为空的表中是很有效的。
         2. 在块中存储变长记录  
            在块中存储变长记录时一般使用分页的槽结构(slotted-page structure),如图所示，每个块(block)中都有header：  
              
            header中包含的信息有：块中条目的个数、空闲空间的结尾处、每条记录的入口(entry,包含记录位置和大小)。记录在块中连续存储，但是是从块的尾部开始存起的。每次插入新的纪录时，就放置在空闲空间的末尾，同时在header中保存这条记录的入口信息；当删除一条记录时，被删除记录前的记录依次先后移动，以占据被删记录的空间。由于每个块的体积在4-8k之间，并不算太大，所以这种移动的开销相对较小。在分页的槽结构中，数据指针不是直接指向数据，而是指向数据的入口(entry)，于是数据在块中可以任意被移动，这可以避免块中碎片的产生。