1. 查询处理

在从数据库提取数据的过程中，查询处理要做的操作有：语法分析与翻译、优化、评估与执行。

2. 查询代价的度量  
   使用传送磁盘块数(number of block transfers)和搜索磁盘次数(number of disk seeks)来衡量查询的代价。假设磁盘子系统传输一个块的数据需要tT秒，搜索数据需要ts秒，则传送b个块并进行S次磁盘搜索的操作将消耗b\*tT+s\*ts秒。现在磁盘的典型数值为tT =0.1毫秒，ts =4毫秒，假定磁盘块的大小是4KB，传输率为40MB/秒。  
   通过将读操作与写操作区分开可以做出更精细地估算，写操作花费的时间约为读操作的两倍，因为写完数据后，磁盘系统会再次读取该扇区以验证写入是否成功。
3. 选择运算  
   在查询处理中，文件扫描是最低级的存取数据方式，这时一种用于定位、检索满足条件的记录的搜索算法。
   1. 使用文件扫描和索引的选择运算  
      对于所有记录都存储在一个文件中的情况，最直接的选择运算方式是线性搜索(A1、linear search)，这种方法会逐个搜索每个文件块，虽然这会比其他一些算法慢，但它可适用于所有的文件类型，不受文件的组织方式、是否有索引等的影响。还有一些使用索引搜索的索引扫描(index scan)算法：  
      A2(主索引，主键等值比较)：使用索引检索满足相应条件等值条件的唯一记录。  
      A3(主索引，非主键等值比较)：当选择条件是基于非主键属性的等值比较时，可以利用主索引检索到多条记录。  
      A4(辅助索引，等值比较)：  
      A5(主索引，比较)：  
      A6(辅助索引，比较)：  
      \*\*\*\*\*\*\*\*\*\*\*
4. 排序  
   逻辑排序是仅对索引进行处理，使得可以根据索引顺序地读取数据。但逻辑排序有可能造成很多次的磁盘访问，有时有必要进行物理排序。如果待排序数据可以全部放入内存，则可以利用快速排序等标准的排序技术，本书讨论内存无法全部容纳待排序数据的情况。
   1. 外部排序归并算法(external sort-merge)  
      令M表示内存缓存区可以容纳的磁盘块数，算法步骤为：

1.首先建立一些排好序的归并段(run)，每个run只包含所有数据中的一部分，且已经排好序。  
伪代码如下：  
repeat  
 从待排序数据中读入M块数据（剩余不足M条则全部读入），在内存中排序， 将排序结果存入归并段Ri中  
until 到达关系末尾  
2.对归并段进行归并。