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：
5. 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，假定归并段的总数N小于
6. 查询优化

查询优化是为处理查询找出一个好的策略的过程。一个查询一般有多种方法可以计算出结果，系统负责将用户输入的查询转换成能够更有效执行的等价查询。

复杂的查询操作涉及多级存取磁盘的操作，由于从磁盘中传输数据比从内存中要慢得多，所以有必要进行查询优化，以选择一个能够最小化磁盘存取的方法。

有很多等价规则可供将一个表达式转化成等价表达式，可以使用这些规则系统地产生与所给查询等价的所有表达式。

选择查询处理策略的第一步就是找到一个关系代数表达式，使它与所给的表示是等价并且据估计有更小的执行代价。

数据库系统为执行一个操作所选择的策略依赖于每个关系的大小和列值的分布情况。数据库系统可以为每个关系r存储统计信息，从而能够基于这些可靠消息选择合适的策略。这些统计信息包括：关系r中的元组数；关系r中的一个记录的大小；关系r中某个特定属性中出现的不同取值的数目等。这些统计信息使得我们可以估计各种操作的结果集的大小和执行操作的代价。当处理一个查询的过冲中有多个索引可用于辅助的时候，关系的统计信息特别有用，这些信息对查询处理策略的选择有很大的影响。

对每个表达式，可以用一些等价规则产生多个可选的执行计划，然后从中选择代价最小的执行计划。为了减少需要产生的可选表达式和执行计划的数量，产生了多种优化技术，比如可以使用启发式的方法，来减少优化的代价。用于关系代数查询转换的启发式规则包括“及早执行选择操作”、“及早执行投影操作”、“避免笛卡尔积操作”等。

还可以使用物化视图来加速查询处理。当原关系发生修改时，需要用增量的视图维护来高效地更新物化视图。利用包含一个操作的输入的变化量的代数表达式，能够完成对该操作的变化量的计算。其他与物化视图相关的问题还包括如何借助物化视图进行查询优化和如何选择需要待物化的视图。