Assessment

Assessment: two coursework (each 15%, take-home assignments) + final exam (70%)
Coursework 1: indexing techniques
Coursework 2: web data storage and distributed database
Final exam
    Part A: short answer questions
        20 questions, 2 marks each, in total 40 marks
    Part B: Problem-Solving and Quantitative Questions
        2 'large' question, 30 marks each, in total 60 marks

Teaching Plan:

* Lecture 1a - Introduction to the Module
* Lecture 1b - Data Storage Structures
* Lecture 2a - Indexing Techniques
* Lecture 2b - B+Tree Indexing
* Lecture 3a - Hash Indexing
* Lecture 3b - Advanced Indexing
* Lecture 4a - Introduction to Relational Model
* Lecture 4b - Query Evaluation Basics
* Lecture 5a - Query Evaluation: Selection
* Lecture 5b - Query Evaluation: Join
* Lecture 6a - Query Optimisation 1
* Lecture 6b - Query Optimisation 2
* Lecture 7a - Transaction Management 1
* Lecture 7b - Transaction Management 2
* Lecture 8a - Concurrency Control
* Lecture 8b - Failure Recovery
* Lecture 9a - Object-Oriented Database
* Lecture 9b - Distributed Database
* Lecture 10a - Web Technologies and Data Storage 1
* Lecture 10b - Web Technologies and Data Storage 2
* Lecture 11a - Big Data Storage
* Lecture 11b - Blockchain-based Storage (guest lecture by Prof. Xin Huang)
* Lecture 12 - Data Analytics
* Lecture 13 – Revision

Required textbook:

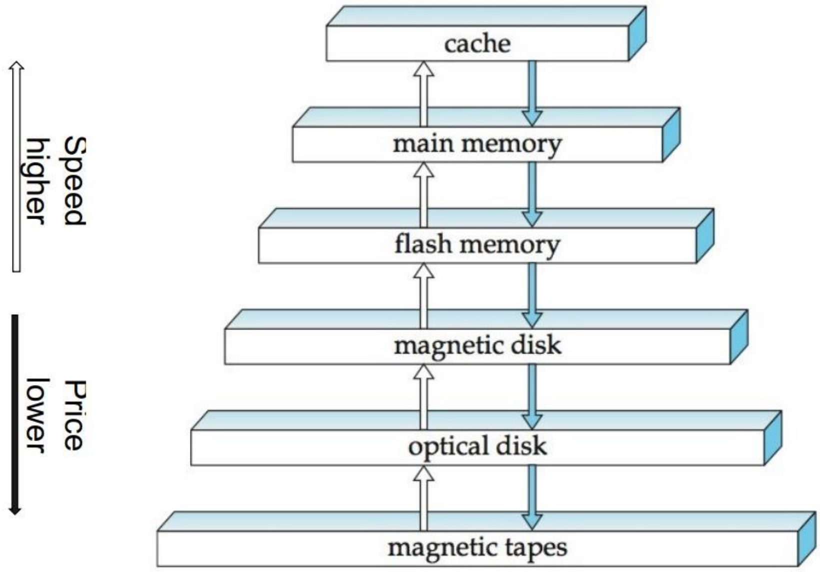
Database System Concept, 7th Ed, A. Silbershatz, H.F. Korth and S. Sudarshen

Data Storage Structures [数据存储结构]:

使用数据模型设计概念性方案例如关系型数据库
数据定义语言(DDL)，数据操作语言(DML)例如 MySQL

Classification of Physical Storage Media [物理储存介质分类]:

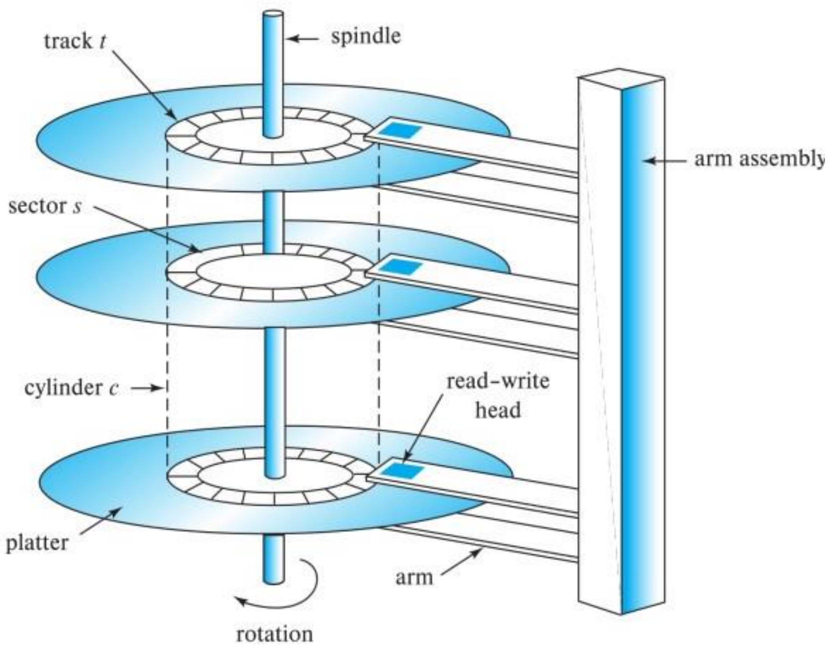
volatile storage [易失性存储器]：在关闭电源时内容会损失
non-volatile storage [非易失性存储器]：包括二级和三级储存，在断电时内容不会损失



Speed：缓存>主内存>闪存>磁盘>光盘>磁带
三级储存概述：
primary storage: fastest media but volatile (cache, main memory)
secondary storage: next levels in hierarchy, non-volatile, moderately fast access time
tertiary storage: lowest levels in hierarchy, non-volatile, slow access time

Magnetic Disks [磁盘概述]:

读写头放在非常接近磁盘表面的地方，用于写入和读取内容；
磁盘呈圆形，表面的每个 track 被划分为单一的 sectors；
sectors 是可读或可写的最小数据单位，Cylinder i 由所有磁盘的第 i 个 track 组成。



Disk Drive Read/Write Operation and Access Time [磁盘读写的操作和对应的访问时间]:

Access time = Seek time + Rotational latency + (Transfer time)

Disk arm swings to position head on the right track;
[磁盘臂旋转到正确的轨道，位置对应的半径]
Average seek time is about 1/2 of the worst case seek time (e.g. from the innermost to the outermost)

Platter spins continually to find the right sector;
[盘旋转让读写头找到正确的区域]
Average latency is about 1/2 of the worst case latency (e.g. nearly 360 degree rotation)

Data is read/written as sector passes under head (Read/Write Need time).
[在读写头经过此区域时进行读写(读写需要时间)]
Average latency is about 1/2 of the worst case latency (e.g. nearly 360 degree rotation)

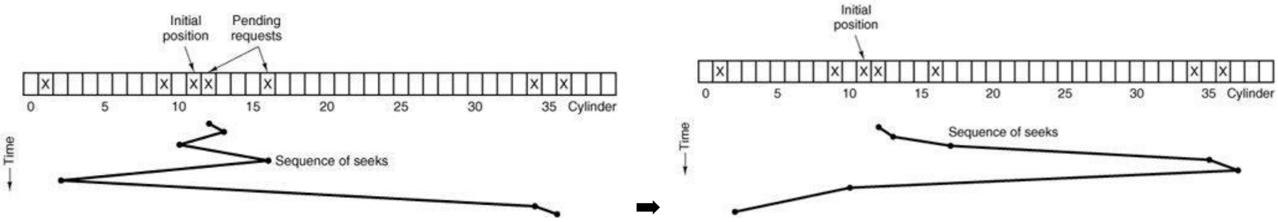
Disk Block [磁盘存储块]:

A contiguous sequence of sectors from a single track, data is transferred between disk and main memory in blocks, sizes range from 512 bytes to several kilobytes.
[单个磁道的连续扇区序列，数据以块的形式在磁盘和主存之间传输，大小从 512 字节到几千字节不等]

Smaller blocks: more transfers from disk
Larger blocks: more space wasted due to partially filled blocks
Typical block sizes today range from 4 to 16 kilobytes

Optimization of Disk-Block Access [优化磁盘块访问]:

The elevator algorithm is a simple algorithm by which a single elevator can decide where to stop, is summarized as follows: Continue traveling in the same direction while there are remaining requests in that same direction.
先到先服务(first come first served)的算法效率太低故转化为电梯(elevator)算法，如下图：



File Organization [文件组织]:

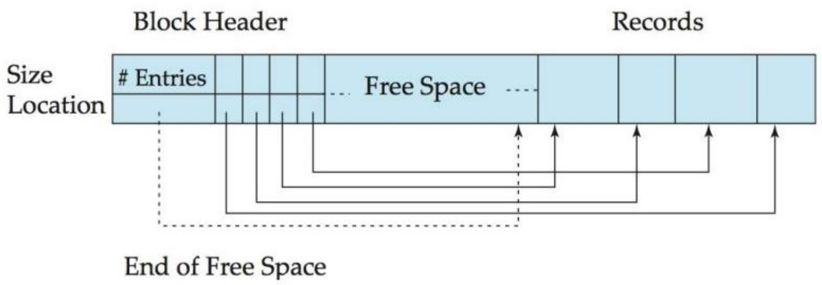
（1）Fixed-length Records
先留一部分空间作为头文件，类似于搞一个链表，增删改通过调整指针指向完成，如果没有可用空间，则在文件末尾添加新空间单元。
（2）Variable-length Records
头文件初始部分为起始位置信息、可变长度属性：(offset, length)

* offset denotes where the data for that attribute begins within the record
* length is the length in bytes of the variable-sized attribute

Slotted-page Structure [分页槽结构体]：

slotted-page structure 属于 Fixed-length Records，用于在一个数据块中组织数据，在每个数据块的起始处有一个记录头，在记录头中通常包含有记录下面三种信息的变量字段：

1. 在这个块中总共包含了多少个记录项(record)
2. 块中的空闲空间结束地址
3. 包含有每个记录项(record) 在系统中存放确切的地址信息和该记录项的长度



(块开始地址)|记录项信息链表的增长方向 --->| (头)空闲空间(不断减少) (尾) |<--- < 记录增长的方向 记录头|(块结尾地址)

Details：CPT201 slotted-page structure - 知乎 (zhihu.com)

Organization of Records in Files [整理数据结构]：

Heap – a record can be placed anywhere in the file where there is space.

[堆——记录可以放在文件中任何有空间的地方]

堆文件组织实现的方式：

1.链表：空闲页和数据页分别组成一个链表，再维护两个指向链表的指针。

2.页目录：目录记录每个页的位置和空闲大小。

Sequential – store records in sequential order, based on the value of the search key of each record.

[顺序——根据每个记录的搜索键的值，按顺序存储记录]

顺序文件组织实现的方式：

使用链表，按照 search-key 进行排序，增删改需要不时地重新调整指针来组织文件以恢复顺序：

10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	62000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000

Multilabel clustering - records of several different relations can be stored in the same file.

[多标签集群——多个不同关系的记录可以存储在同一个文件中]

集群文件组织实现的方式：

在文件中储存多种不同的关系，可以添加链表来连结各种关系对应的数据；

对于查询包括增删改很友好，但是当关系只有一个时略显浪费。

- department

dept_name	building	budget
Comp. Sci.	Taylor	100000
Physics	Watson	70000

- instructor

ID	name	dept_name	salary
10101	Srinivasan	Comp. Sci.	65000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
83821	Brandt	Comp. Sci.	92000

- multitable clustering of *department* and *instructor*

Comp. Sci.	Taylor	100000	
10101	Srinivasan	Comp. Sci.	65000
45565	Katz	Comp. Sci.	75000
83821	Brandt	Comp. Sci.	92000
Physics	Watson	70000	
33456	Gold	Physics	87000

B+-tree - provide efficient ordered access to records even with large number of insert, delete, or update operations (more in next lecture).

[B+树——即使使用大量的插入、删除或更新操作，也提供对记录的高效有序访问(更多在下一讲)]

Hashing – a hash function computed on search key; the result specifies in which block of the file the record should be placed (more in next lecture).

[哈希——在搜索键上计算的哈希函数，结果指定记录应该放在文件的那个块中(更多在在下一讲)]