

面经

计算机组成

中断和程序查询的区别？

程序查询方式是CPU不断的对外设的状态的进行查询，直到外设做好准备，从而控制I/O设备与主机交换信息，CPU会一直被外设所阻塞，不能执行其他的任务。

中断方式，只有在IO设备做好准备并向CPU发送中断请求时，CPU才会对中断进行响应，否则CPU会继续执行自己的任务。

哈佛结构、冯诺依曼结构

哈佛结构是一种将程序指令存储和数据存储分开的存储器结构。哈佛结构是一种并行体系结构，它的主要特点是将程序和数据存储在不同的存储空间中，即程序存储器和数据存储器是两个独立的存储器，每个存储器独立编址、独立访问。

哈佛结构与冯·诺依曼结构处理器相比，处理器有两个明显的特点：使用两个独立的存储器模块，分别存储指令和数据，每个存储模块都不允许指令和数据并存；使用独立的两条总线，分别作为CPU与每个存储器之间的专用通信路径，而这两条总线之间毫无关联。

流水线冒险的种类

结构冒险

是由于多条指令需要同时争用一个功能部件引起的冲突。本质是数据通路中的功能部件数量的匮乏，以至于不足以满足多条指令同时运行时对此类功能部件的需求。

控制冒险

在程序中存在分支、函数调用等需求，所以指令流中会存在导致指令执行方向改变的相关指令，所谓控制冒险，是指，这些指令会改变指令执行方向，但流水线在取下条指令时不知道该从哪个方向取指令。

数据冒险

是因为后进入流水线的指令需要引用先期进入流水线正在执行的指令的计算结果所引起的冲突。数据冒险的本质因为指令之间通过寄存器交换数据。由于寄存器在前后指令之间同时出现，因此数据冒险的本质是数据相关。

操作系统

说出五种进程调度算法，Windows 和 Linux 采用的是哪一种

调度算法：

- 短作业优先
- 先来先服务
- 时间片轮转
- 多级反馈队列
- 高响应比优先

win和linux的调度算法不能指定说是某一种特定的调度算法，在实际的操作系统中调度算法往往需要考虑很多因素，比如进程优先级，响应比，以及进程的本身和重要性，比如linux进程可分为实时进程，普通进程等，操作系统的调度各有特点，win的调度算法拥有CPU亲和性，某些进程只能在某些特定的CPU上运行。

linux的调度算法是将时间片轮转，优先级调度、先进先出调度等多个算法结合在一起的算法。

简要介绍OS实验课做了什么

进程有哪些状态，他们之间如何转换？

创建、就绪态，运行态，阻塞态，终止

进程同步的意义

在操作系统中，多个进程并发执行，不同的进程之间存在着相互制约的关系，为了协调制约关系，实现多个进程在执行次序上的协调，引入了进程同步

Windows / Linux 下进程间通信的手段

低级和高级

低级是PV操作

高级是共享存储，消息传递，管道通信

共享存储，共享访问一个空间。低级的是数据结构的共享，高级的是基于存储区的共享。

消息传递

直接通信：发送方将消息发送到接收方，并挂在接收进程的消息缓冲队列中。

间接通信：消息发送到某个中间实体，接收方从中间实体那里获取消息，中间实体一般称为信箱。

银行家算法

银行家算法的实现思想：允许进程动态地申请资源，系统在每次实施资源分配之前，先计算资源分配的安全性，若此次资源分配安全（即资源分配后，系统能按某种顺序来为每个进程分配其所需的资源，直至最大需求，使每个进程都可以顺利地完成），便将资源分配给进程，否则不分配资源，让进程等待。

Cache替换策略

线程和进程的联系与区别，为什么要引入线程

进程是一个正在执行程序实例，是操作系统资源分配的基本单位，在操作系统中引入进程的目的，是为了使多个程序能并发执行，以提高资源的利用率和系统的吞吐量。

线程是CPU调度和分配的单位，基本上不拥有资源，只有必不可少的少量资源，可以与同进程的其他线程共享进程的所有资源。

目的：减少程序再并发执行时所付出的时空开销，提高操作系统的并发性能。

内存泄露

内存泄漏（Memory Leak）是指程序中已动态分配的堆内存由于某种原因程序未释放或无法释放，造成系统内存的浪费，导致程序运行速度减慢甚至系统崩溃等严重后果。

页式存储管理的优劣

优点：不产生外碎片，产生的内碎片的大小不会超过一个页的大小。

缺点：单纯的分页存储管理，程序需要一次性装入内存，在使用请求分页的管理方式下，需要对应的硬件支持，比如中断处理和地址转换功能。

DMA解决了什么问题

在大数据传输时，即便中断处理方式速度较快，但仍然会产生大量的中断，降低CPU的利用率，DMA在数据传输过程中完全不需要CPU，CPU只需要在开始和结束时进行介入即可。

编译技术

五个基本逻辑部分：

词法分析-语法分析-语义分析并生成中间代码-代码优化-生成目标程序

其中符号表的管理和错误处理贯穿上面的五个部分

编译文法的类型

0 1 2 3

用 C 语言时，函数在不同作用域可以定义同名变量，在编译上如何实现

使用栈式符号表，进入一个函数时，创建符号表，执行结束，符号表便被销毁，从符号表栈中弹出，每次检查当前符号表是否包含此变量，不包含便向外层符号表进行查询。

语法规则的定义和作用

根据语法规则（即语言的文法），分析并识别出各种语法成分，如表达式、各种说明、各种语句、过程、函数等，并进行语法正确性检查。

指针的编译

寄存器的分配

C语言堆和栈内容的区别

堆的内容分配是程序手动分配，并需要程序在是使用后手动释放，但必须及时释放否则会导致内存泄漏。

栈上的内容是在程序运行时自动分配和释放的。

编译执行和解释执行的优劣

编译执行是一次性编译成目标程序，然后机器运行目标程序

解释执行是不直接翻译为机器语言，而是先翻译成中间代码，再由解释器对中间代码进行解释执行。

编译执行执行速度快，因为省略了编译这一步，但灵活性不高，目标程序的运行需要在

解释执行速度慢，灵活性高，可移植性好。

编译文法

0型文法，短语结构文法。

1型文法：上下文有关文法

2型文法：上下文无关文法

3型文法：正则文法。

计算机网络

子网掩码

一个网络地址是否是当前网络中的地址。

计网的七层、五层模型

七层：物理层、数据链路层、网络层、传输层、会话层、表示层、应用层。

五层：物理层、数据链路层、网络层、传输层、应用层。

浏览器输入域名

解析url，获取一系列的信息，生成HTTP请求消息，通过OS进行发送，但需要告诉OS服务器到ip的映射，DNS服务器（UDP），得到IP地址，然后浏览器委托OS，发送请求，OS建立TCP连接（过程），发送http请求报文，接受http响应报文，浏览器展示内容，根据http协议的版本不同，连接建立和断开的方式也不同，1.0无状态，1.1允许持续连接，提高效率。在网页关闭，TCP断开。

TCP拥塞控制

拥塞控制是指避免网络上注入过多的报文，这样可以使网络中的路由器 或链路不致过载。

TCP采用的是基于窗口的拥塞控制。发送方会维持一个拥塞窗口，并根据网络拥塞情况来调整拥塞窗口大小，发送方让发送窗口的大小等于拥塞窗口。具体的算法有四种：慢开始，拥塞避免，快重传，快恢复。

慢开始是发送方采用一种从小到大逐渐增大发送窗口的方式，具体来说就是在发送开始时，由于不知道网络拥塞情况，所以初始设置窗口大小为一个最大报文字段，或两个，之后若网络不发生超时，则每经过一个传输轮次将发送窗口大小扩大一倍，窗口大小超过了慢开始门限，便开始采用拥塞避免算法，此时每经过一个传输轮次，便将发送窗口增加1，然后在接下来的传输中，如果出现传输超时，则将拥塞窗口设置为最小值，并将慢开始门限设置超时时拥塞窗口大小的一半，然后重新执行慢开始和拥塞避免。

快重传算法，是让接收方每收到一个报文便发出确认报文，哪怕不连续，此时若发送方接受到了三个相同的ACK报文，采用快恢复算法，将门限调整为此时拥塞窗口的一半，然后执行拥塞避免算法。

交换机的工作内容

工作在数据链路层，连接多个以太网物理段，隔离冲突域，对以太网帧进行转发，能够自行学习和维护MAC地址信息。

网关协议，路由协议

OSPF, RIP, BGP

TCP面向连接，UDP面向报文

为什么不能手机号实现IP地址，手机号和IP号有什么本质区别

IGP和EGP的区别

IGP：在一个自治系统内部使用的路由选择协议。

EGP：源主机和目的主机处在不同的自治系统中，而两个不同的自治系统内部使用的协议可能不同，当数据报传输到一个自治系统的边界时，就需要使用一种协议将路由选择信息传递到另一个自治系统中。

路由器和交换机的对比

路由器工作在网络层

交换机工作在数据链路层

数据结构

最小生成树

最短路径

稳定排序

假设排序前 $a[i]=a[j]$ ，且 i 的位置在前，而 j 的位置在后，排序后仍然能够保证这样的顺序，那么就成该排序算法是稳定排序算法。

基数排序、冒泡排序、直接插入排序、折半插入排序，归并排序是**稳定排序**。

堆排序、快速排序、希尔排序、直接选择排序是**不稳定排序**。

B树

B树是一种类似二叉平衡树的数据结构，又称多叉平衡搜索树。

从阶的角度对B树进行定义的话：

- M阶B树，树中每一个结点最多含有 m 个孩子。
- 除根节点和叶结点外，其他结点至少拥有 $m/2$ 向正无穷取整的孩子。
- 所有的叶子结点都在同一深度

每一个内部节点的键将节点子树分开。

选择排序

每次从剩下的数中选中最小的换到对应的位置。不稳定排序

插入排序

对于 i ，每次对 $0 \sim i-1$ 中的所有数按照从小到大的顺序进行交换。

数组和链表的优缺点

数组的优点：

- 具备随机访问的能力，根据下标可以直接访问。
- 查找速度快

链表的优点：

- 插入、删除的效率高。
- 可扩展性好
- 内存空间利用率高。

Hash

hash又称散列表，目的是要实现在常数时间内完成对数据的查找、插入、删除等简单操作。具体实现是通过某个算法或者是函数f，我们输入数据到f中，f输出一个对应的index，也即这个数据在散列表中的下标，我们将数据存储到相应位置后，在查找时只需要进行f的计算得到我们查找数据的下标，从而获得数据。由于需要常数次的散列运算，所以要尽可能的保证f对于不同数据所产生的下标不同，也即独一无二，但这是很难的，所以可能会出现两个数据对应同样一个下标的情况，这个时候便是hash碰撞，解决的办法也有很多种：

- 采用链表的方式。也即表中的每一个数据都是一个链表的头，当出现碰撞时，我们可以将数据直接加到下标对应的链表的末尾或者开头。
- 线性探测，当出现碰撞，便顺序从当前下标往后寻找，直到找到一个空闲位置，将其放入。也可以间隔两个进行查找。
- 溢出区。可以按照冲突的前后顺序将冲突的数据放到冲突区，在查找时，如果一次hash找不到，便去溢出区遍历查找。
-

离散数学

笛卡尔积

集合A和B的笛卡尔乘积

$$A \times B = \{ \langle x, y \rangle \mid x \in A \ \& \ y \in B \}$$

关系

图

关系图

关系矩阵

二元关系

- 自反。（关系图中，以下同此）每一个顶点都有自环，（关系矩阵，以下同）主对角线都为1
- 反自反。每一个顶点都无自环，对角线都为0
- 对称。任意两个不同顶点之间：或者无弧，或者有两条相反的弧。R的关系矩阵是对称矩阵。
- 反对称。任意不同顶点之间最多有一条弧。
- 传递。若顶点x到顶点y有一条路径，则必有从x到y的一条边。

关系运算

逆，交并补

关系的合成

定义12 (合成) 设 R 是 X 到 Y 的关系， S 是 Y 到 Z 的关系，则 $R \circ S = \{ \langle x, z \rangle \mid \exists y \in Y \text{ 使得 } x R y \wedge y S z \}$ 为 X 到 Z 的关系，称为 R 和 S 的**合成**。

2 关系的运算——闭包

定义14 设 R 是集合 A 上的关系。

关系 R' 称为 R 的自反（对称、传递）闭包，当且仅当 R' 满足以下三个条件：

(1) R' 是自反的(对称的、传递的)；

包含 R 的最小自反
(对称、传递)关系。

(2) $R \subseteq R'$ ；

(3) 对于 A 上的任何自反（对称、传递）关系 R'' ，

如果 $R \subseteq R''$ ，则 $R' \subseteq R''$ 。

将 R 的自反，对称，传递闭包分别记作 $r(R)$, $s(R)$, $t(R)$ 。

定理： 设 R 为集合 A 上的二元关系，则

(1) R 是自反的 当且仅当 $r(R) = R$ ；

(2) R 是对称的 当且仅当 $s(R) = R$ ；

(3) R 是传递的 当且仅当 $t(R) = R$ 。

66

定理： 设 R 是集合 A 上的关系，则

(1) $r(R) = R \cup I_A$ ；

(2) $s(R) = R \cup R^{-1}$

(3) $t(R) = \bigcup_{n=1}^{\infty} R^n$

次序关系

R 是偏序关系当前仅当 R 是自反的、反对称的和传递的。

定义6.10（全序关系） 设 $\langle A, \leq \rangle$ 是一个偏序结构，
如果对于任意 $x, y \in A$ ，或者 $x \leq y$ ，或者 $y \leq x$ ，
即 x 与 y 可比，则称 \leq 为 A 上的全序或线序，并称
 $\langle A, \leq \rangle$ 为全序结构或链。即

$$(\forall x)(\forall y) (x \in A \wedge y \in A \rightarrow x \leq y \vee y \leq x)$$

R 是严格偏序关系（又称拟序关系）当且仅当 R 是反自反的和传递，并由此可以推断 R 是反对称的。

覆盖



良序



相容关系

集合A上的关系R是**自反**和**对称**的，则称R为A上的**相容关系**。

此时 $r(R) = s(R) = R$ 。

等价关系

集合A上的关系R是**自反**、**对称**、**传递**的，则称R是A上的**等价关系**

等价类



集合的划分



商集



函数

函数是一种特殊类型的二元关系。

部分函数



单射，满射，双射



可逆



集合的势

两个集合A和B，若存在A到B的双射则称A和B对等，或A和B等势。等势关系本身是等价关系。

有限集

任何有限集合都不能与其真子集对等。

图

没有自圈和平行边的图称为**简单图**。

无向图节点的度是边的两倍。

度数为奇数的节点称为奇节点

度数为偶数的节点称为偶节点

度数为0的节点称为孤立节点

度数为1的节点称为断点。

节点都是孤立的点的图称为零图。

一阶零图称为平凡图

所有节点的度均为自然数D的无向图称为d度正则图。

如果n阶简单无向图G是n-1度正则图，则称G为完全无向图。

如果每一个结点的出度和入度都是n-1的n阶简单有向图称为完全有向图。

子图、真子图、生成子图

1、子图、真子图、生成子图

设 $G = \langle V, E, \Psi \rangle$, $G' = \langle V', E', \Psi' \rangle$ 为图。

◆ 如果 $V' \subseteq V$, $E' \subseteq E$, $\Psi' \subseteq \Psi$, 则称**G'是G的子图**, 记为 $G' \subseteq G$, 并称 **G 是 G' 的母图**。

◆ 如果 $V' \subseteq V$, $E' \subset E$, $\Psi' \subset \Psi$, 则称**G'是G的真子图**, 记为 $G' \subset G$ 。

◆ 如果 $V' = V$, $E' \subseteq E$, $\Psi' \subseteq \Psi$, 则称**G'是G的生成子图 (Spanning Subgraph)**。

导出子图：从原图中的取出一部分顶点为V，以所有起点和终点都在V中的边为边集合的G的子图称为由V导出的G的子图，记为G[V]

图的运算

图的可运算：

图的三个运算：

- **交**。即以顶点的交和边的交的集合的图，称为交。
- **并**。以顶点的并，边的并为集合的图，称为并。
- **环和**。以顶点的并，边的异或为集合的图，称为环和。

图运算具有唯一性。即图运算的结果具有唯一性。

补图：一个 n 阶无向图 $G(V, E, *)$ 是 n 阶完全无向图 K 的生成子图，则称 $K-E$ 为 G 的补图

简单无向图都有补图，一个简单无向图的所有补图都同构

路径

路径

定义 7.3.1 设 $n \in \mathbb{N}$ ， v_0, v_1, \dots, v_n 是图 G 的结点， e_1, e_2, \dots, e_n 是图 G 的边，并且 v_{i-1} 和 v_i 分别是 e_i 的起点和终点 ($i=1, 2, \dots, n$)，则称序列 $v_0 e_1 v_1 e_2 \dots v_{n-1} e_n v_n$ 为图 G 中从 v_0 至 v_n 的路径， n 称为该路径的长度。

- (i) 如果 $v_0 = v_n$ ，则称该路径为闭的，否则称为开的。
- (ii) 如果 e_1, e_2, \dots, e_n 互不相同，则称该路径为简单的。
- (iii) 如果 v_0, v_1, \dots, v_n 互不相同，则称该路径为基本的。

◆ 基本路径必为简单路径

边互不相同称为简单路径。

顶点互不相同称为基本路径。

如果两点之间存在路径，那么一定存在基本路径

可达

在图中存在从 v_1 到 v_2 的路径，则称在图中 v_1 和 v_2 可达。

无向图的连通

无向图中的任意两个结点都互相可达，则称图是连通的，否则是非连通的。

有向图的连通

有向图的基础图，将有向图的有向边转变为无向边得到的无向图称为有向图的基础图。

如果有向图中的任意两个结点都相互可达，则称 G 是强连通。

如果对于有向图的任意两个结点，必有一个结点可达另一个结点则称有向图是单向连通的。

如果 G 的基础图是连通的，则称 G 是弱连通的。

极大子图

极大子图、分支

定义7.3.9 设 G' 是图 G 的具有某性质 P 的子图，并且对于 G 的具有该性质的任意子图 G'' ，只要 $G' \subseteq G''$ 就有 $G' = G''$ ，则称 G' 相对于该性质是 G 的极大子图。

定义7.3.10 无向图 G 的极大连通子图称为 G 的分支。

定义7.3.11 设 G 是有向图。

- i) G 的极大强连通子图称为 G 的强分支。
- ii) G 的极大单向连通子图称为 G 的单向分支。
- iii) G 的极大弱连通子图称为 G 的弱分支。

13

连通无向图恰有一个分支。

非连通无向图的分支多于一个。

有向图的每一个结点和每条边恰处于一个若分支中。

强连通（单向连通，弱连通）的有向图恰有一个强分支（单向分支，弱分支）

非强连通（非单向连通，非弱连通）有向图有一个以上强分支（单向分支，弱分支）

半路径

设 G_1 是有向图 G 的基础图，则 G_1 中的路径称为 G 中的半路径。

回路

连通2度正则图称为回路。

基础图是回路的有向图称为半回路。

每一个结点的出度和入度均为1的弱连通有向图称为有向回路。

如果回路（有向回路，半回路） C 是图 G 的子图，则称 G 有回路（有向回路，半回路） C

没有回路的无向图和没有半回路的有向图称为非循环图。

如果一个有向图有子图 G ，使得对于 G 的任意结点 v 都有 v 的入度大于0，则原图有有向回路。

如果一个有向图有子图 G ，使得对于 G 的任意结点 v 都有 v 的出度大于0，则原图有有向回路。

图 G 不是非循环图当且仅当 G 有子图 G_1 ，使得对于 G_1 的任何结点 v 都有 v 的度大于1。

点割集

设无向图 G 是连通图，若有点集 V_1 是 G 点集 V 的真子集，使得

- 图 G 删除了 V_1 的所有结点后，所得的子图是不连通图
- 而删除了 V_1 的任何真子集，所得到的子图仍是连通图

则称 V_1 是 G 的一个点割集。

无向图的点连通度（简称连通度）：连通度表示为了产生一个不连通图所需要删除的点的最少数目。

边割集

设无向图 G 为连通图，若有边集 E_1 是 G 边集 E 的真子集使得

- 图 G 删除了 E_1 的所有边后，所得子图是不连通图
- 而删除了 E_1 的任意真子集后，所得的子图仍是连通图，

则称 E_1 是 G 的一个边割集。

边连通度：是为了产生一个不连通图所需要删除的边的最少数目。

欧拉路径

图 G 包含其所有边的简单开路径称为 G 的欧拉路径

图 G 中包含其所有边的简单闭路径称为 G 的欧拉闭路。

欧拉图

每一个结点都是偶结点的无向图称为欧拉图。

每一个结点的出度和入度都相等的有向图称为欧拉有向图。

欧拉定理

G 是**连通无向图**， G 是欧拉图当且仅当 G 有欧拉回路。

哈密顿回路

如果回路（有向回路） C 是图 G 的生成子图，则称 C 为 G 的哈密顿回路（哈密顿有向回路）

图 G 中包含它所有结点的基本路径称为 G 的哈密顿路径。

有哈密顿回路（有向回路）的图称为哈密顿图（哈密顿有向图）

树

树的定义

定理7.6.1 树定义的等价条件

设 $G = \langle V, E, \Psi \rangle$ 是 n 阶无向图，则以下条件等价：

- i) G 是连通的和非循环的。
- ii) G 无自圈，且当 $v, v' \in V$ 时，皆有唯一的一条从 v 至 v' 的基本路径。
- iii) G 是连通的，且当 $v, v' \in V$ 时， $e \notin E$ ， $\Psi' = \{ \langle e, \{v, v'\} \rangle \}$ 时， $G + \{e\}_{\Psi'}$ 有唯一的一条回路。
- iv) G 是连通的，且当 $e \in E$ 时， $G - e$ 是非连通的。
- v) G 是连通的且 $n(E) = n - 1$ 。
- vi) G 是非循环的且有 $n(E) = n - 1$ 。

32

每一个分支都是树的无向图称为森林。

生成树

如果树 T 是无向图 G 的生成子图，则称 T 为 G 的生成树。

如果森林 F 是无向图 G 的生成子图，则称 F 为 G 的生成森林。

最小生成树：

prim算法：贪心算法，将结点分为最小生成树中的结点和不在最小生成树中的结点，随后每次从树中的结点的所有边中的最小的边所连接的顶点加到树中结点，直到全部结点都被加入。

优化：由于每次都需要在当前顶点的全部边中查找最短的边，耗费时间，所以采用松弛操作，维护一个数组来记录当前顶点到树中结点的最小值，也即树中全部结点的最小值，每次新加入一个结点就对新加入结点所连接的边进行松弛。

Kruskal：贪心算法，将边按照从小到大排序，按照顺序选边，只要不与当前已选择的边构成循环，就选择，直到选择了所有的边。

优化：由于判断循环需要使用dfs或bfs比较耗时，所以采用并查集的形式进行优化。

设 T 是连通无向图 G 的生成树，称 T 的边为枝，而 G 的不属于 T 的边称为弦

设 T 是连通无向图 G 的生成树， G 的只包含一条弦的回路称为基本回路

有向树

一个结点的入度为 0，其余结点的入度均为 1 的弱连通有向图 称为有向树。

概率统计

大数定律

在一定条件下，均值依概率收敛到期望。

中心极限定理

大量独立同分布的变量的和趋向于正态分布。

方差：用于反应数据的离散程度，期望反应数据的聚合情况

协方差用于反映两个维度之间的数据偏离期望值的相关性。

协方差用来研究X与Y之间偏差的关联程度

高数

格林公式用于二元二型曲线积分。

斯托克斯公式用于三元二型曲线积分，两种形式，一种转换为一型曲面积分，一种转换为二型曲面积分。

高斯公式用于二型曲面积分。

函数在某点的梯度是一个向量，它的方向与取得最大方向导数的方向一致

微分是y的变化量中的线性主要部分。

软件工程

瀑布模型

系统需求、软件需求、分析、程序设计、编码、测试、运行。

渐进式模型

MVP模型MBP模型

Minimum Viable Product

最小和最大

敏捷模型

泳道图也称跨职能流程图，由多条泳道框架与相应的流程内容构成。泳道框架能够将流程中所涉及的步骤、流程、职能等信息直观展现出来，是一种特殊的图表。

UML图

- UML图分为结构图和行为图。
- 结构图分为类图、轮廓图、组件图、组合结构图、对象图、部署图、包图。
- 行为图又分活动图、用例图、状态机图和交互图。
- 交互图又分为序列图、时序图、通讯图、交互概览图。

软工三要素

方法、过程和工具

封装、继承、多态、抽象。

封装是将对象的属性和行为结合为一个独立整体，尽可能隐藏细节

super 用来引用直接父类对象

this会返回类的引用，用来访问类中的方法和对象。

重写：函数名和形参都不能改变

重载：函数名不变，参数不能相同。

数据库范式

1NF，表中没有表，都是原子值

2NF，首先是1NF，满足每一个非主属性都完全依赖于码

3NF，首先是2NF，满足每一个非主属性都不传递依赖于任何码

BCNF：所有的非主属性都完全函数依赖于每一个候选码

所有主属性都完全依赖于每一个不包含它的候选码

没有任何属性完全函数依赖于非码的任何一组属性