

NJU2012 年计算机科学与技术基础试卷与答案

科目名称：计算机科学与技术基础

一、（10 分）函数 W 递归定义如下：

$$W(n) = cn + w\lfloor n/2 \rfloor; \quad w(1) = 1$$

证明： $W \in O(n)$

解：一个含直接或间接调用本函数语句的函数被称之为递归函数，它必须满足以下两个条件：

- 1) 在每一次调用自己时，必须是（在某种意义上）更接近于解；
- 2) 必须有一个终止处理或计算的准则。

考点：求解递归式 算法导论 P47-53

二、（15 分）

问题 P：任给一无向图 G ，判定它是否含有哈密顿（Hamilton）回路。我们知道问题 P 是 NP-完全问题。

问题 N：任给一带权完全图 H 与一正整数 k ，判定 H 中是否含有总权数为 k 的哈密顿（Hamilton）回路，这就是著名的“旅行推销员问题”，我们知道问题 Q 也是 NP-完全问题。

假设对问题 Q 做如下简化：每条边的权值只能是 1 或 2

1. 简述如何利用问题 P 的 NP-完全性证明简化的问题 Q 仍是 NP-完全问题。
2. 给出具体的证明。

答案来源：算法导论 P624/641

课程网址：http://cs.fjzs.edu.cn/ketang/lssxshort/part4/chapter15/15_02_03_03.htm

在图中找出一条包含所有结点的闭路，并且，出来起点和重点重合外，这条闭路所含结点是互不相同的。对于一个给定的网络，确定起点和终点后，如果存在一条路径，穿过这个网络，我们就说这个网络存在哈密顿路径。

从图中的任意一点出发，路途中经过图中每一个结点当且仅当一次，则成为哈密顿回路。经过图（有向图或无向图）中所有顶点一次且仅一次的通路称为哈密顿通路。经过图中所有顶点一次且仅一次的回路称为哈密顿回路。

具有哈密顿回路的图称为哈密顿图，具有哈密顿通路但不具有哈密顿回路的图称为半哈密顿图。平凡图是哈密顿图。

三、简述软件生命期的瀑布模型。并说明这个模型的优点（10 分）

瀑布模型（Waterfall Model）将软件生命周期划分为系统需求分析、软件需求分析、概要设计、详细设计、编码、测试和运行维护等七个阶段，每一阶段工作的完成需要确认，并且规定了它们自上而下、相互衔接的固定次序，如同瀑布流水，逐级下落，最终得到软件产品，如图 1 所示。瀑布模型核心思想是按工序将问题化简，将功能的实现与设计分开，便于分工协作，即采用结构化的分析与设计方法将逻辑实现与物理实现分开。

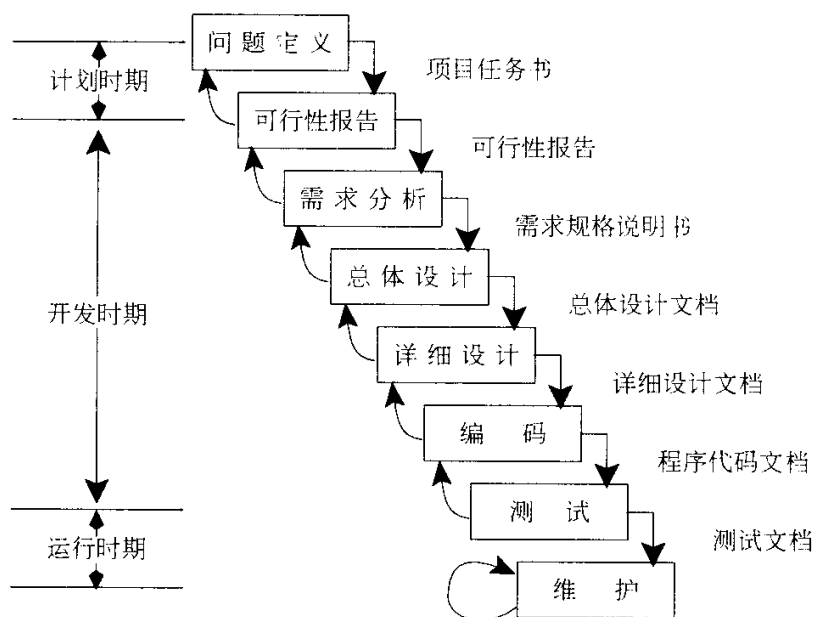


图 1 瀑布模型

在瀑布模型中，软件开发的各项活动严格按照线性方式进行，当前活动接受上一项活动的工作结果，实施完成所需的工作内容。当前活动的工作结果需要进行验证，如果验证通过，则该结果作为下一项活动的输入，继续进行下一项活动，否则返回修改。

瀑布模型的主要特点是：阶段间的顺序性和依赖性。开发过程是一个严格的下导式过程，即前一阶段的输出是后一阶段的输入，每一阶段工作的完成需要确认，而确认过程是严格的追溯式过程，后一阶段出现了问题要从前一阶段的重新确认来解决。因此，问题发现的越晚，解决问题的代价就越高。

瀑布模型有以下优点

- （1）为软件项目提供了按阶段划分的检查点。
- （2）当前一阶段完成后，您只需要去关注后续阶段。
- （3）可在迭代模型中应用瀑布模型。

增量迭代应用于瀑布模型，每次迭代产生一个可运行的版本，同时增加更多的功能。每次迭代必须经过质量和集成测试。

（4）提供了一个模板，这个模板使得分析、设计、编码、测试和支持的方法可以在该模板下有一个共同的指导。

瀑布模型的缺点

瀑布模型强调文档的作用，并要求每个阶段都要仔细验证。但是，这种模型的线性过程太理想化，已不再适合现代的软件开发模式，几乎被业界抛弃，其主要问题在于：

- （1）各个阶段的划分完全固定，阶段之间产生大量的文档，极大地增加了工作量；
- （2）由于开发模型是线性的，用户只有等到整个过程的末期才能见到开发成果，从而增加了开发的风险；
- （3）早期的错误可能要等到开发后期的测试阶段才能发现，进而带来严重的后果。

(4) 通过过多的强制完成日期和里程碑来跟踪各个项目阶段。

(5) 瀑布模型的突出缺点是不适应用户需求的变化。

第一，从认识论上讲，人的认识是一个多次反复的过程，软件开发是人的一个智力认识活动，不可能一次完成，需要多次反复地进行，但瀑布模型中划分的几个阶段，没有反映出这种认识过程的反复性。

第二，软件开发是一个知识密集型的开发活动，需要人们合作完成。因此，人员之间的通讯和软件工具之间的联系，活动之间的并行和串行等都是必需的，但在瀑布模型中也没有体现出这一点。

四、简述模型驱动的体系结构的概念，其中的平台无关模型（Platform Independent Models, PIM）和平台相关模型（Platform Specific Models, PSM）是指什么？（7 分）

参考网址：<http://jpkc.whu.edu.cn/jpkc/dxqyxxxtfgnjg/dzja/dzjc/jc10.htm>

模型驱动体系架构（Model Driven Architecture, MDA）是由 OMG 提出的新的软件方法学，被面向对象技术界预言为未来几年里最重要的软件方法学。模型驱动体系架构(MDA)把建模语言用作一种编程语言而不仅仅是设计语言，并以一种全新的方式将 IT 技术的一系列新的趋势性技术整合到一起。这些技术包括基于组件的开发、设计模式、中间件、说明性约束、抽象、多层系统、企业应用整合以及契约式设计等。

模型(Model)是用某种工具对同类或其他事物的表达方式，它从某一个建模观点出发，抓住事物最重要的方面而简化或忽略其他方面，工程、建筑和其他许多领域中都用到模型概念。软件系统的模型用建模语言来表达，如 UML 是对系统的功能、结构或行为的形式化规范。

(1)模型驱动(Model-driven)是指利用模型来指导系统开发的全过程[6]，包括系统理解、设计、架构、开发、部署、维护、集成等系统生命周期相关的全过程。

(2)软件体系结构(Architecture)为软件总体结构框架，它是构件、连接器及其之间关系的系统规约。MDA 主要使用模型来形式化描述软件体系结构及其风格。

(3)软件平台(Platform)是指用来构建与支撑应用软件的独立软件系统。它是开发与运行应用软件的基础，是任何一个应用软件得以实现与应用的必要条件。软件平台有两个基本要素，即支撑环境和开发体系，其中支撑环境是指应用软件系统开发与运行的基本条件，开发体系是指开发与维护管理应用软件的工具与方法。

平台无关模型(Platform Independent Model, PIM)是一个系统功能和结构的形式化规范，它从与实现技术无关的细节中抽象而来。平台相关模型(Platform Specific Model, PSM)是合成了系统实现平台技术细节的平台模型，它是平台无关模型到具体平台的映射。

平台无关模型（Platform Independent Models, PIM）

PIM 准确的定义应该是/独立于平台的计算模型 0,业务模型和需求模型也是平台独立的"在现阶段的 MDA 开发过程中,平台独立意味着独立于信息格式化技术!3GL!4GL!分布式组件中间件和消息中间件"在这个意义上,平台独立模型要比目前程序员工作的抽象层次更高一级"

平台相关模型（Platform Specific Models, PSM）

平台相关模型是一个同特定的信息格式化技术!编程语言!分布式组件中间件!消息中间件相关的计算模型"

五、简述面向对象语言中是如何实现封装的。（8 分）

封装(encapsulation): 是隐藏对象的属性和实现细节, 仅对外公开接口, 控制应用程序中属性的读取和修改的访问级别。

通过封装, 可以实现对属性的数据访问限制, 同时增加了程序的可维护性。由于取值方法和赋值方法隐藏了实现的变更, 因此并不会影响读取或修改该属性的类, 避免了大规模的修改, 程序的可维护性增强。

实现封装的方法有:

(1) 属性的封装: 将属性私有化, 提供公有的方法访问私有的属性

修改属性的可见性来限制对属性的访问, 并为每个属性创建一对取值 (getter) 方法和赋值 (setter) 方法, 用于对这些属性的访问; 在赋值和取值方法中, 加入对属性的存取的限制。

(2) 类的封装:

(3) 方法的封装:

六、解释题（10 分）

1. 实时与分时

实时是没有时间延迟或是很短, 要求反应快, 尽可能下达任务就能完成, 所以实时系统多是单任务的, 而且要去掉我们平时用的操作系统里冗余的部分, 来达到尽可能快的要求。实时系统一般用在航天、导弹等这些需要快速计算的领域。

分时则是一种实用方式, 指的是多个任务轮流使用一个 CPU, 你用一会, 他用一会, 由于转换的较快, 每个人都认为只有自己用似的。分时系统就很多了, 常见的 UNIX, Linux, Windows 都是分时的。分时不是为了提高效率, 而是为了实现多任务。分时以后, 就可以同时运行好几个任务了。

2. 操作系统的用户接口

操作系统的用户接口, 就是操作系统提供给用户, 使用户可通过它们调用系统服务的手段。

为了方便用户对计算机系统的使用和编程, 操作系统向用户提供了用户与操作系统的接口, 简称为用户接口。通过该接口, 用户可以向操作系统请求特定的服务, 操作系统提供服务的结果。

用户接口可分为三个部分:

(1) 命令接口: 为了便于用户直接或间接控制自己的作业, 操作系统向用户提供了命令接口。命令接口是用户利用操作系统命令组织和控制作业的执行或管理计算机系统。命令是在命令输入界面上输入, 由系统在后台执行, 并将结果反映到前台界面或者特定的文件内。命令接口可以进一步分为联机用户接口和脱机用户接口。

(2) 程序接口: 程序接口由一组系统调用命令组成, 这是操作系统提供给编程人员的接口。用户通过在程序中使用系统调用命令来请求操作系统提供服务。每一个系统调用都是一个能完成特定功能的子程序。如早期的 UNIX 系统版本和 MS-DOS 版本。

(3) 图形接口: 图形用户接口采用了图形化的操作界面, 用非常容易识别的各种图标来将系统各项功能、各种应用程序和文件, 直观、逼真地表示出来。用户可通过鼠标、菜单和对话框来完成对应程序和文件的操作。图形用户接口元素包括窗口、图标、菜单和对话框, 图形用户接口元素的基本操作包括菜单操作、窗口操作和对话框操作等。

3. 进程死锁的必要条件

如果在计算机系统中同时具备下面四个必要条件时, 那么就会发生死锁。换句话说, 只要下面四个条件有一个不具备, 系统就不会出现死锁。

(1) 互斥条件。进程要求对所分配的资源进行排它性控制, 即在一段时间内某资源仅为一个进程所占有。

(2) 不剥夺条件。进程所获得的资源在未使用完毕之前，不能被其他进程强行夺走，即只能由获得该资源的进程自己来释放。

(3) 部分分配条件。进程每次申请它所需要的一部分资源。在等待新资源的同时，进程继续占有已分配到的资源。部分分配条件又称为请求和保持条件。

(4) 环路等待条件。存在一种进程资源的循环等待链，链中每一个进程已获得的资源同时被链中下一个进程所请求。

4. 虚存管理（虚拟内存管理）

当程序的存储空间要求大于实际的内存空间时，就使得程序难以运行了。虚拟存储技术就是利用实际内存空间和相对大的多的外部存储器存储空间相结合构成一个远远大于实际内存空间的虚拟存储空间，程序就运行在这个虚拟存储空间中。能够实现虚拟存储的依据是程序的局部性原理，即程序在运行过程中经常体现出运行在某个局部范围之内的特点。在时间上，经常运行相同的指令段和数据（称为时间局部性），在空间上，经常运行与某一局部存储空间的指令和数据（称为空间局部性），有些程序段不能同时运行或根本得不到运行。虚拟存是把一个程序所需要的存储空间分成若干页或段，程序运行用到页和段就放在内存里，暂时不用就放在外存中。当用到外存中的页和段时，就把它们调到内存，反之就把它们送到外存中。装入内存中的页或段可以分散存放。

虚拟内存管理是一种借助于外存空间，从而允许一个进程在其运行过程中部分装入内存的技术。虚拟内存技术允许执行的进程不必完全在内存中，其优点是程序可以比物理内存大。

七、简答题（15 分）

1. 说明操作系统的作用（3 种以上）。

操作系统（Operating System，简称 OS）是计算机系统中负责支撑应用程序运行环境以及用户操作环境的系统软件，同时也是计算机系统的核心与基石。

作用：控制管理计算机的全部硬软件资源，合理组织计算机内部各部件协调工作，为用户提供操作和编辑界面的程序集合。

(1) 处理器管理：对处理器的分配和运行实施有效的管理。

中央处理机（CPU）是计算机系统中一个举足轻重的资源。用户程序进入内存后，只有获得 CPU，才能运行。为提高 CPU 的利用率，系统必须采用多道程序设计技术，使内存中同时有几个用户作业程序存在，当一个程序因等待某事件的完成而暂时放弃使用 CPU 时，操作系统就把 CPU 分配给其他可运行的作业程序使用，从而提高它的利用率。

处理器管理的主要任务如下：

- 进程和线程的描述与控制
- 处理器调度
- 进程或线程的同步与互斥
- 死锁的检测和预防
- 进程之间及线程之间的通信

(2) 存储器管理：对内存进行分配、保护和补充。

存储器是计算机的记忆装置，分为内存存储器和外存储器两种。操作系统中的存储管理是针对内存而言的。也就是说，存储器管理是指对计算机内存的管理。

存储器管理的主要任务如下：

- 内存规划、分配及地址映射
- 内存保护
- 内存扩充

(3) 设备管理：对系统内的 I/O 设备进行的管理，为用户分配设备，使设备与处理器并行工作，方便使用。

户使用设备。

计算机系统中，除处理器和内存外，都是设备管理的对象，主要是一些输入/输出设备和外存。设备管理是操作系统中最为复杂、庞大的部分。

记住各类设备的使用状态，按各自不同的性能特点进行分配和回收。

为各类设备提供相应的设备驱动程序、启动程序、初始化程序以及控制程序等，保证输入输出操作的顺利完成。

设备管理的主要任务如下：

- 输入/输出设备控制
- 缓冲管理
- 设备独立性
- 设备分配
- 虚拟设备
- 磁盘存储器管理

(4) 文件管理：有效地支持文件的存储、检索和修改等操作，解决文件的共享、保密和保护等问题。

程序与数据以文件的形式存放在外存（如硬盘、光盘）上，是计算机系统的软件资源。用户是通过文件的名称来访问所需要的文件的，这就是所谓的“按名存取”方式。

文件管理指对文件进行组织和访问控制，实现文件的按名存取，为用户提供方便的文件存取访问和可靠的文件共享，实现文件保护。

文件系统的主要任务如下：

- 对文件结构进行组织和目录管理
- 提供文件的存取访问
- 实现文件的存储空间管理
- 实现文件的共享和保护

(5) 作业管理

作业：是用户在一个事务处理过程中要求计算机系统所做工作的集合，包括用户程序、所需的数据及命令等。

从程序运行的角度看，作业由一些相对独立的顺序执行的步骤所组成，这些相对独立的执行步骤通常被称为作业步。从系统管理角度看，作业由程序、数据和作业说明书 3 部分组成。作业包含的程序和数据完成用户所要求的业务处理工作，作业说明书则体现用户的控制意图。作业说明书包含 3 方面的内容，即作业的基本描述、作业控制描述和资源要求描述。

作业管理是指当一个用户的作业提交给系统后，操作系统对其实施的所有管理和控制工作。其主要任务包括建立作业，提供必需的资源系统，控制其开始运行，运行结束后撤销该作业。这些任务可分为两类：

作业控制：主要讨论系统通过何种方式使用户方便地描述和控制解题过程，或者说操作系统为用户提供了什么样的接口，用户又如何利用这些接口使用计算机。也称为操作系统的用户接口

作业调度：包括作业的进入、调度算法的选择、为选中的作业分配系统资源及作业完成后的善后处理等

2. 解释进程通信，列出主要的进程通信方式（3 种）。

进程通信（IPC）是在不同进程之间传播或交换信息，主要的进程通信方式有以下几种：

(1) 共享存储器：

共享内存使得多个进程可以访问同一块内存空间，是最快的可用 IPC 形式。是针对其他通信机制运行效率较低而设计的。往往与其它通信机制，如信号量结合使用，来达到进程间的同步及互斥。

共享内存是被多个进程共享的一部分物理内存。共享内存是进程间共享数据的一种最快的方法，一个进程向共享内存区域写入了数据，共享这个内存区域的所有进程就可以立刻看到其中的内容。

(2) 信号（Signal）：信号是比较复杂的通信方式，用于通知接受进程有某种事件发生，除了用于进程间通信外，进程还可以发送信号给进程本身。

(3) 管道（Pipe）及有名管道（named pipe）：管道可用于具有亲缘关系进程间的通信，有名管道克

服了管道没有名字的限制，因此，除具有管道所具有的功能外，它还允许无亲缘关系进程间的通信。

(4) 报文 (Message) 队列 (消息队列)：消息队列是消息的链接表。有足够权限的进程可以向队列中添加消息，被赋予读权限的进程则可以读走队列中的消息。消息队列克服了信号承载信息量少，管道只能承载无格式字节流以及缓冲区大小受限等缺点。

(5) 套接口 (Socket)：更为一般的进程间通信机制，可用于不同机器之间的进程间通信。

(6) 信号量 (semaphore)：主要作为进程间以及同一进程不同线程之间的同步手段。

3. 简述文件系统的作用。

文件系统是操作系统的一个重要组成部分，通过对操作系统所管理的存储空间的抽象，向用户提供统一的、对象化的访问接口，屏蔽对物理设备的直接操作和资源管理。

文件系统是操作系统用于明确磁盘或分区上的文件的方法和数据结构；即在磁盘上组织文件的方法。也指用于存储文件的磁盘或分区，或文件系统种类。

从用户的角度来看，文件系统主要实现“按名存取”。为了能正确地按名存取，文件系统还具有以下作用 (功能)：

- (1) 实现从逻辑文件到物理文件之间的转换
- (2) 有效地分配文件的存储空间
- (3) 实现对文件的操作管理
- (4) 实现文件信息的共享和保护
- (5) 提供用户接口

八、关系代数与 SQL (第 1 小题 4 分，第 2 小题 6 分，共 10 分)

设有一个公司产品销售数据库，其关系模式如下：

顾 客 C (编号 cid, 姓名 cname, 城市 city, 折扣 discent)

供应商 A (编号 aid, 名称 aname, 城市 city)

商 品 P (编号 pid, 名称 pname, 库存数量 quantity, 单价 price)

订 单 O (编号 ordno, 年份 year, 月份 month, 顾客编号 cid, 供应商编号 aid, 商品编号 pid, 订购数量 qty, 销售金额 dols)

1. 请用关系代数表示下述查询操作

- (1) 查询顾客和供应商位于同一个城市的订单的编号；
- (2) 查询没有销售过商品的供应商的编号和名称。

2. 请用 SQL 语言表示下列查询操作

- (1) 查询没有销售过商品的供应商的编号和名称；
- (2) 统计查询每一个顾客的购买总金额；
- (3) 对于每一种商品，查询其销售金额最高的订单的编号。

九、什么是关系数据库系统中的事务？请简要给出事务的性质 (8 分)

事务：是查询或更新数据库内容 (数据项) 的一个程序执行单元，由有限的数据库数据操作序列构成，

这些操作要么全做要么全不做，是一个不可分割的工作单位；它的执行能够保证数据库从一个正确的状态转移到另一个正确的状态。在关系数据库中，一个事务可以是一条 SQL 语句、一组 SQL 语句或整个程序；一个应用程序可以包含多个事务。事务以 BEGIN TRANSACTION 语句开始，以 COMMIT（提交）语句或 ROLLBACK（回退或撤消）语句结束。

原子性：指事务的所有操作在数据库中要么全部正确完成，要么什么都不做；

一致性：指当事务完成时，必须使数据库中的所有数据都具有一致的状态；

隔离性：指当多个事务并发执行时，一个事务的执行不能被其他事务干扰；

持久性：指一个事务一旦提交，它对数据库中数据的改变应该是永久性的，即使系统可能出现故障，也应保留这个事务执行的痕迹。

十、什么是 NoSQL 数据库？请结合自己的理解，谈一谈数据库兴起的原因（7 分）

NoSQL 是一种与关系数据库截然不同的数据库管理系统，它的数据存储格式可以是松散的、通常不支持 Join 操作，并且支持横向扩展，也可以称之为非关系数据库。

云计算时代对数据库技术提出了新的需求，主要表现在以下几个方面：

（1）海量数据处理：对类似搜索引擎和电信运营商级的经营分析系统这样大型的应用而言，需要能够处理 PB 级的数据，同时应对百万级的流量。

（2）大规模集群管理：分布式应用可以更加简单地部署、应用和管理。

（3）低延迟读写速度：快速的响应速度能够极大地提高用户的满意度。

（4）建设及运营成本：云计算应用的基本要求是希望在硬件成本、软件成本以及人力成本方面都有大幅度的降低。

云计算时代，传统的关系型数据库在应对超大规模和高并发的服务请求方面暴露了许多难以克服的问题，主要表现在以下方面。

（1）高并发读写速度慢

这种情况主要发生在数据量达到一定规模时，由于关系型数据库的系统逻辑非常复杂，使得其非常容易发生死锁等并发问题，导致其读写速度下降非常严重。例如，Web2.0 网站要根据用户个性化信息来实时生成动态页面、提供动态信息，所以基本上无法使用 动态页面静态化技术，因此数据库并发负载非常高，往往要达到每秒上万次读写请求。关系型数据库勉强可以应付上万次 SQL 查询，硬盘 I/O 往往无法承担上万次的 SQL 写数据请求。

（2）支撑容量有限

类似 Facebook、Twitter 这样的 SNS 网站，用户每天产生海量的用户动态，每月会产生几亿条用户动态，对于关系型数据库来说，在一张数亿条记录的表里面进行 SQL 查询，效率是极其低下乃至不可忍受的。

（3）扩展性差

在 基于 Web 的架构当中，数据库是最难进行横向扩展的，当一个应用系统的用户量和访问量与日俱增的时候，传统的关系型数据库却没有办法像 Web Server 那样简单地通过添加更多的硬件和服务节点来扩展性能和负载能力。对于很多需要提供不间断服务的网站来说，对数据库系统进行升级和扩展是非常痛苦的事情，往往需要停机维护和数据迁移，因此迫切需要关系型数据库也能够通过不断添加服务器节点来实现扩展。

（4）建设和运维成本高

企业级数据库的价格很高，并且随着系统的规模增大而不断上升。高昂的建设和运维成本无法满足云

计算应用对数据库的需求。

关系型数据库遇到上述难以克服的瓶颈，与此同时，它的很多主要特性在云计算应用中却往往无用武之地，例如：数据库事务一致性、数据库的写实时性和读实时性、复杂的 SQL 查询特别是多表关联查询。因此，传统的关系型数据库已经无法独立应付云计算时代的各种应用。

关系型数据库越来越无法满足云计算的应用场景，为了解决此类问题，非关系型数据库应运而生。由于在设计上和传统的关系型数据库相比有了很大的不同，所以此类数据库被称为“**NoSQL (Not only SQL)**”系列数据库。与关系型数据库相比，它们非常关注对数据高并发读写和海量数据的存储，在架构和数据模型方面作了简化，而在扩展和并发等方面作了增强。NoSQL 常用数据模型包括以下 3 种：

(1) Column-oriented (列式)

列式主要使用 Table 这样的模型，但是它并不支持类似 Join 这样多表的操作，它的主要特点是在存储数据时，主要围绕着“列 (Column)”，而不是像传统的关系型数据库那样根据“行 (Row)”进行存储，也就是说，属于同一列的数据会尽可能地存储在硬盘同一个页中，而不是将属于同一个行的数据存放在一起。这样做的好处是，对于很多类似数据仓库的应用，虽然每次查询都会处理很多数据，但是每次所涉及的列并没有很多。使用列式数据库，将会节省大量 I/O，并且大多数列式数据库都支持 Column Family 这个特性，能将多个列并为一个小组。这样做的好处是能将相似列放在一起存储，提高这些列的存储和查询效率。总体而言，这种数据模型的优点是比较适合汇总和数据仓库这类应用。

(2) Key-value (键-值)

虽然 Key-value 这种模型和传统的关系型相比较简单，有点类似常见的 HashTable，一个 Key 对应一个 Value，但是它能提供非常快的查询速度、大的数据存放量和高并发操作，非常适合通过主键对数据进行查询和修改等操作，虽然不支持复杂的操作，但是可以通过上层的开发来弥补这个缺陷。

(3) Document (文档)

在结构上，Document 和 Key-value 是非常相似的，也是一个 Key 对应一个 Value，但是这个 Value 主要以 JSON 或者 XML 等格式的文档来进行存储，是有语义的，并且 Document DB 一般可以对 Value 来创建 Secondary Index 来方便上层的应用，而这点是普通 Key-Value DB 所无法支持的。

目前，主流的 NoSQL 数据库包括 BigTable、Cassandra、MongoDB、SimpleDB、CouchDB、HBase、Redis 等。