

数据库系统概述

笔记本: database_theory

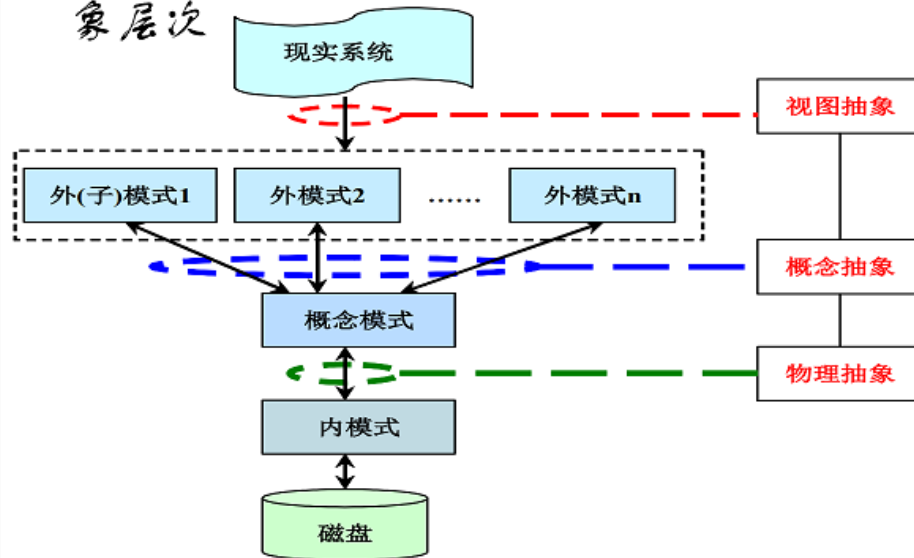
创建时间: 2021/6/25 14:45

更新时间: 2021/7/1 16:34

作者: 134exetj717

- 首先, 我们需要知道数据库为什么会出现?
- 数据库开发以前, 我们一直用的都是基于文件的数据管理系统, 该系统有什么样的缺陷呢?
 - 大容量数据的处理与存储:
 - 内存不够: 基于文件系统进行数据处理时, 一般都是将数据先读到内存的数据链表中, 当面对小数据量时是允许的, 但是当数据量足够庞大时, 我们不能一次将所有数据都读入内存中, 就出现了问题;
 - 32位计算机直接访问的地址为4GB: 在以前的操作系统版本中, 硬盘直接规定了单个文件的大小4GB;
 - 大数据量下的查询速度: 既不能一次读入内存, 也不能放在一个文件中, 直接导致查询速度慢。
 - 多用户并发访问: 基于文件系统一般都是单机的
 - 故障情况下的数据恢复: 文件系统处理的不够好
 - 安全性: 权限问题
 - 数据的完整性: 当对文件进行操作时, 多个文件中的内容不能保持一致, 而数据库中对保持数据的完整性处理的相当好, 既有参照, 也有触发器或者中断
- 当数据库出现以后, 其基本功能有哪些呢?
 - 数据独立性: 用户只管录入数据, 至于其他的数据格式或者存放以及改变问题, 直接交给数据库管理系统进行处理;
 - 并发访问: 多个用户可以同时访问数据库, 其中涉及到了锁;
 - 安全性: 设置了权限, 涉及到数据库的层级关系, 用户、服务器管理员SA (system administrator)、数据库管理员dbo;
 - 数据完整性: 参照与触发器的问题;
 - 故障恢复: 数据库安全处理。
- 当然, 有时候我们不一定需要使用到数据库:
 - 苛刻的实时环境: 对于实时控制的前台应用, 一般都不需要基于磁盘的DBMS开发, 因为增加了中间层, 就增加了响应时间;
 - 操作少、代码要求精炼: 不需要用到DBMS庞大的功能;
 - 操纵的数据是非结构化或半结构化数据:
 - 半结构化: 只有一部分可以离散并结构化, 而其他内容无法用一种标准格式来结构化数据, 比如文档数据中的标题、作者是可以抽取出来的, 因此格式统一, 但是文档内容却无法抽取;
 - 非结构化: 比如音频、图像等。
- 有了数据库以后, 我们又应该如何使用数据库呢?
- 首先, 我们需要知道数据库系统的抽象层次:
 - 视图抽象: 数据提取的过程, 从现实系统中提取数据, 得到多个 外模式;
 - 概念抽象: 从多个 外模式 中总结出一个 概念模式;
 - 物理抽象: 从 概念模式 转化为计算机可实现的 内模式, 即可物理实现的逻辑数据模式。

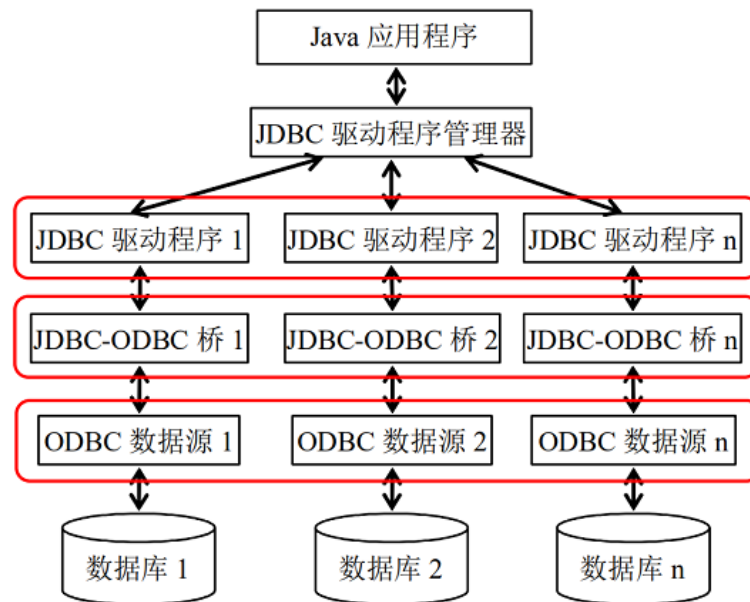
三.从应用系统开发的角度来看待数据库的抽象层次



数据库系统抽象层次

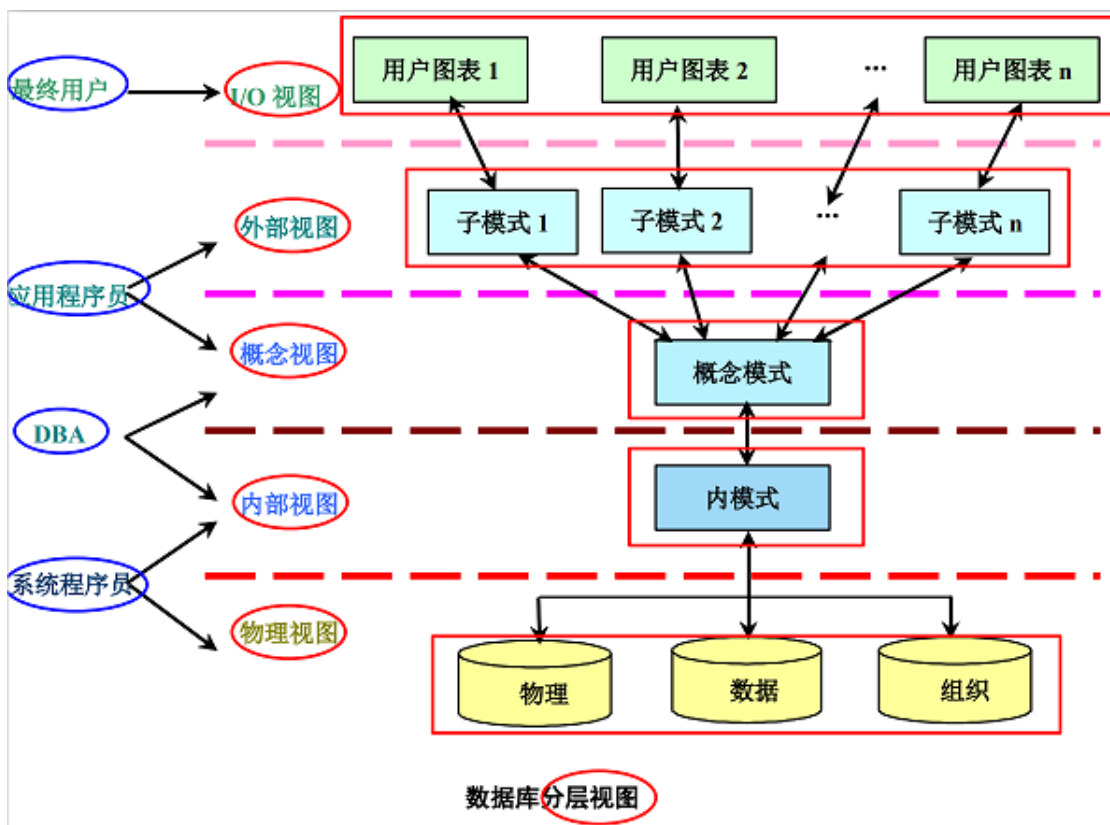
- 了解数据库的抽象层次以后，我们需要了解数据库语言，通过其来让DBMS了解我们创建的具体数据结构
 - 数据定义子语言（DDL），data define language;
 - 数据操纵子语言（DML），manage;（操作
 - 数据控制子语言（DCL），control。（完整性
- 其中，当前最流行的一种数据库语言是：
 - SQL语言，即结构化查询语言，是关系型数据库管理系统（RDBMS）支持的数据库语言，由于RDBMS是目前数据库领域的主流DBMS，因此SQL最流行。
- 有了SQL命令，我们如何与RDBMS进行交互呢？
 - SQL的交互式使用：用户直接在RDBMS控制台上，使用SQL命令交互；
 - 用于通过开发的应用系统与RDBMS交互。
- 随着时代的发展，RDBMS的产品越来越多，那么我们为什么要使用ODBC/JDBC等中间件呢？
 - 原因如下：
 - 不同的RDBMS产品支持的SQL语言在功能、语法上存在一定的差异，为了便于应用程序的移植和互操作，屏蔽不同RDBMS产品在SQL语言上的差异，需要一种遵循标准SQL语句二点、访问DB的中间件。
 - 而ODBC/JDBC则提供了数据库访问的统一、标准的接口函数。
 - 中间件的定义：
 - 位于平台（硬件和操作系统）和应用软件之间的、为应用软件提供运行和开发环境的一种全新软件体系结构。
 - 目前已发展称为与系统软件和应用软件并列的三大软件形式之一。
 - ODBC是什么？
 - 开放数据库连接。
 - JDBC是什么？
 - 执行SQL语句的Java API。
 - JDBC其实就只是提供了一个 JDBC-ODBC桥，成功地允许将JDBC调用转化为ODBC调用。--> 也就是说，最终和数据库连接的都是 ODBC。

JDBC (Java Database Connectivity)



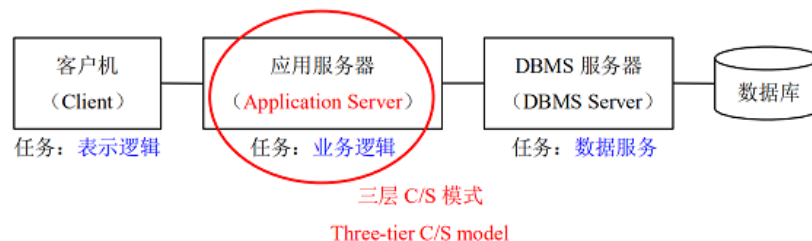
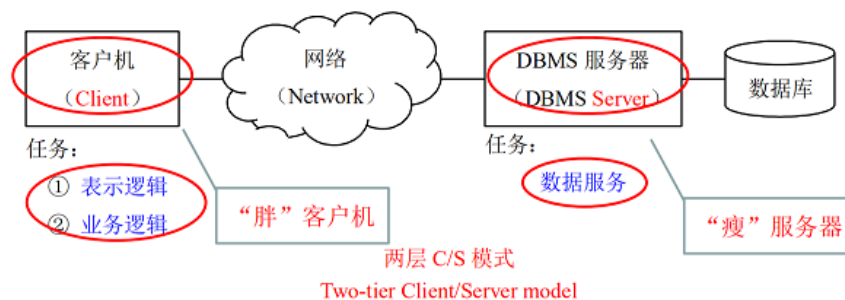
通过 JDBC-ODBC 桥访问数据库

- 有了中间件以后，我们就可以正确使用数据库了，那么此时，我们如何将现实中的信息正确转化为数据库中储存的信息，最后以一种数据结构的方式正确呈现给用户呢？
 - 我们可以将通过视图抽象、概念抽象、物理抽象得到的SQL命令传输给数据库，然后中间件调用数据库，直接或者间接传送给用户观看。
- 了解了数据库的由来，以及数据库的正常运作方式，我们是否需要了解数据库的一些基本概念和关键术语？
 - 数据是什么？
 - 数据是现实世界中各种具体事务或者抽象概念的、可存储并具有明确意义的信息。
 - 数据库：相互关联的数据的集合。
 - 数据库管理系统：通用的软件系统，由一组计算机程序组成。
 - 数据库系统：指一个环境。
 - 数据库应用系统：简称应用系统，主要是指实现业务逻辑的应用程序。
 - 该系统要为用户提供一个友好和人性化的操作数据的图形用户界面。比如：Navicat。
 - 数据字典：DBMS中的一个特殊文件，用于存储数据库中的一些说明信息，这些说明信息称为元数据。
- 随着时代的发展，电脑朝着两个极端发展，一个是巨型化，一般用于大型数据的超高精度计算，服务于政府或国家，代表了硬实力；另一个是微型化，一般用于个人计算机，即PC。此时我们会好奇，技术水平的提升，微型电脑渗透到了家家户户，那么此时我们就没有自己的数据库吗？
 - 答案是有的，大中型RDBMS 和 微型计算机RDBMS之间的区别在于：
 - 功能。微型RDBMS简化或者去除了很多功能，这样就有效避免了数据库过于庞大的问题。
- 数据库展现给用户的，是一个视图，那么视图是怎样的呢？
 - 视图：指一个人看某个物体所得到的图像。
 - 将其延伸到数据库层面，不同的用户对同一个数据库的每一种理解，称为该数据库的一个视图。



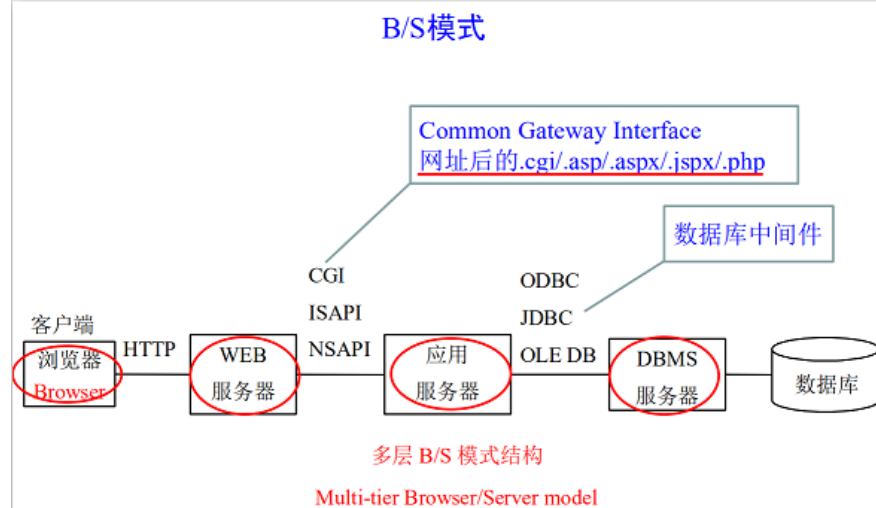
- 要了解视图，我们还需要了解 数据抽象、数据模型、数据模式之间的关系。
 - 数据抽象：指将现实世界中的数据进行抽取变成计算机可以管理的数据的过程；
 - 数据模型：一种数据结构，是数据抽取的工具，数据组织和表达的方式；
 - 数据模型可分为：概念模型、逻辑模型、物理模型
 - 数据模型的三要素：数据结构、数据操作、数据的约束条件
 - 数据模式：通过数据模型实现数据抽象所得到的结果。比如：张三的个人信息表，以一种基本数据结构展现的具体信息。
- 我们已经了解了数据库的内部实现原理，但是我们知道用户和数据库之间具体是怎么建立联系的吗？
 - C/S模式：(client / server)
 - 前端由客户机担任，负责GUI (graphic user interface, 图形用户界面) 处理以及向服务器发送用户请求和接收服务器回送的结果；
 - 后端为服务器，主要承担数据库的管理、按用户请求进行数据处理并会送结果等工作。

C/S模式

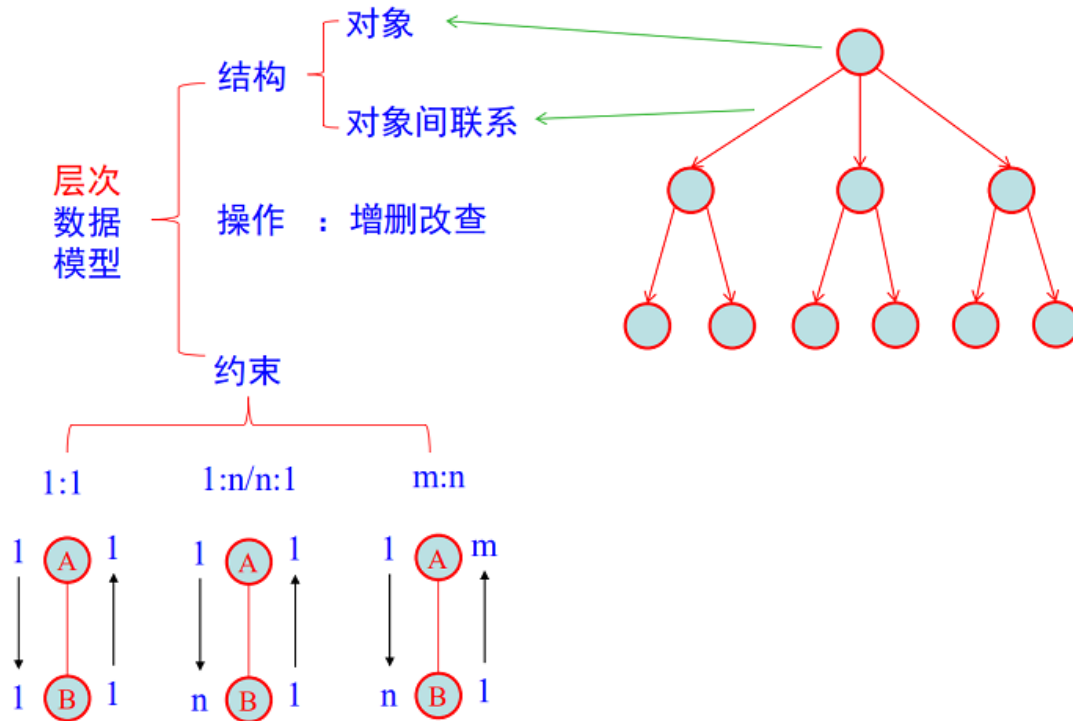


- B/S模式： (browser / server)

■ 该结构中，浏览器与Web服务器之间通过HTTP（超文本传输协议）通信。

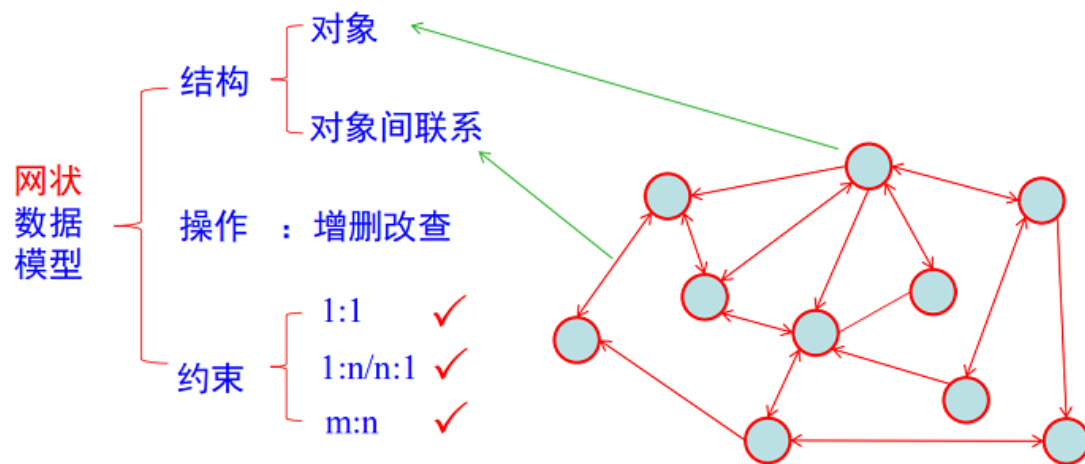


- 后者是前者的进阶版，对于后者来说，不再需要像前者一样下载庞大的客户端软件，而是直接通过浏览器实现云端的登录与应用。
- 了解一个产品，我们往往还需要了解它的过往，这样才能发现当前产品的改进，那么传统的数据模型是怎么样的呢？
 - 层次数据模型：用树形结构表示各类实体以及实体之间的联系



- 缺点：1.只有一个节点，没有双亲结点，即根节点；2.根以外的其他节点有且只有一个双亲结点
- 基本特点：任何一个给定的记录值只有按其路径查看时，才能显出它的全部意义，没有一个子女记录值能够脱离双亲记录值而独立存在。

- 网状数据模型



- 第二代数据库系统：关系数据库管理系统

