

# 第4章

# 数据库管理系统引论

2012. 03



# 目录 Contents

---

- **4.1 数据库管理系统结构简介**
- **4.2 事务**
- **4.3 DBMS的进程结构**
- **4.4 DBMS的系统结构**
- **4.5 数据目录 (字典)**



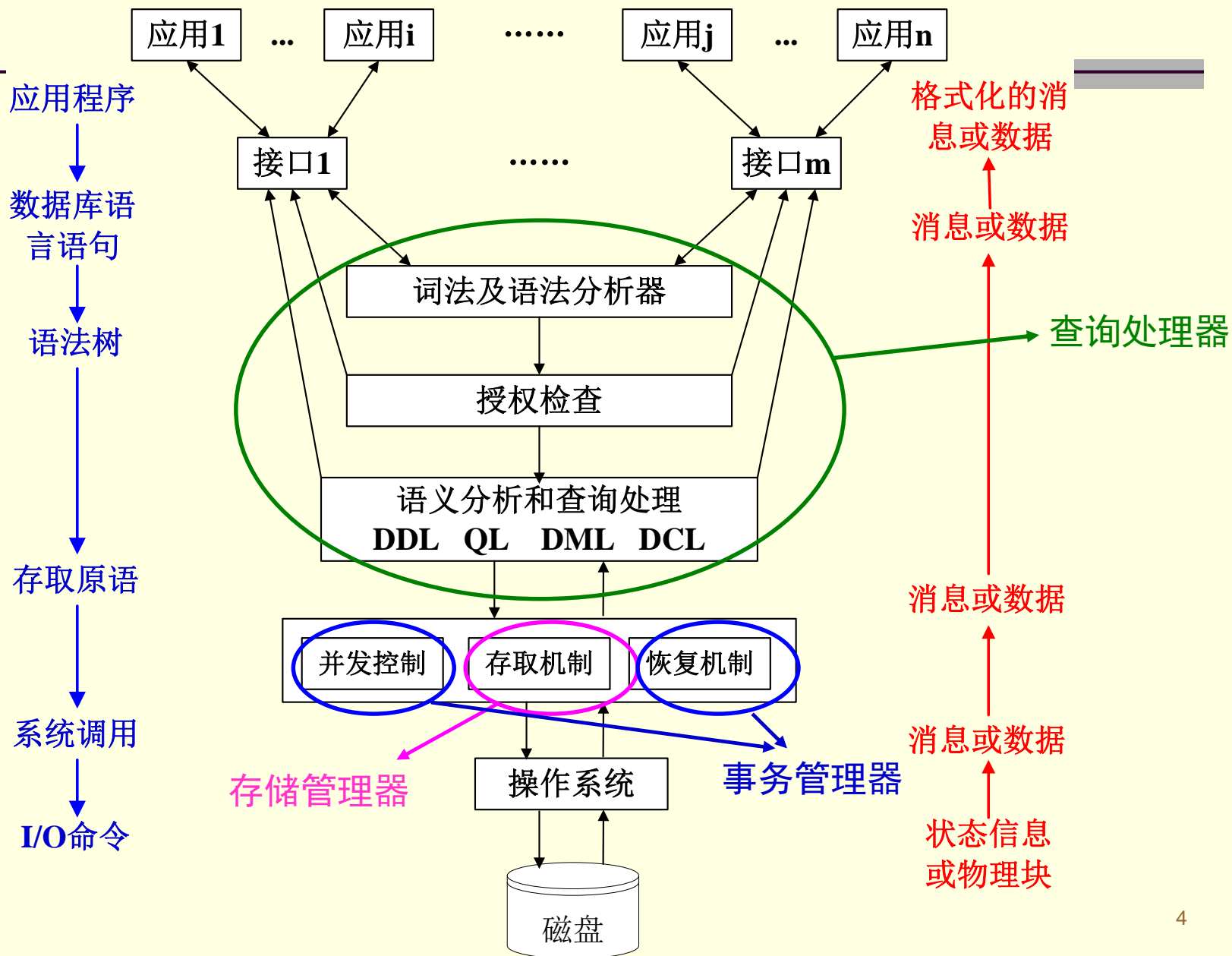
# 4.1 数据库管理系统结构简介

---

- DBMS是数据库系统的核心，对数据库系统的功能和性能有决定性影响。
- DBMS的最基本的功能是正确的、安全、可靠地执行数据库语言的语句。因此，DBMS可以看成数据库语言的一个实现。
- DBMS分为编译和解释两种实现方法。



# 解释执行的RDBMS结构



# 目录 Contents

---

- 4.1 数据库管理系统结构简介
- 4.2 事务
- 4.3 DBMS的进程结构
- 4.4 DBMS的系统结构
- 4.5 数据目录 (字典)



## 4.2 事务

### 一、事务的概念

- **事务(Transaction)**: 是DBMS的（最小、完整的）执行单位，它由某个用户所执行的一个不能被打断的对数据库的操作序列（SQL语句）组成，且必须满足**ACID**性质。
  - ‘事务’是应用程序访问数据库的**基本逻辑工作单位**。
  - ‘事务’通常由一组对于数据库的访问操作组成，在执行过程中**按照预定的次序顺序执行**。
  - 一个‘事务’的执行过程是串行的，它将数据库从一个**旧的一致性**状态转换到一个**新的一致性**状态。在‘事务’的执行过程中，数据库中的数据可能有不一致的现象，但在‘事务’执行结束时，系统将保证数据库中数据的一致性。



## 4.2 事务

- **[例]**银行的转帐业务：根据输入的两个银行存款帐号A和B，以及转帐金额X，将帐号A的金额减去X，帐号B的金额增加X。其处理过程如下（其中**READ(A)**表示将帐号A的金额读入内存变量A，**WRITE(A)**表示将内存变量A的值作为帐号A的金额写入数据库）：

**READ(A);**

IF ( $A \geq X$ )

THEN BEGIN

A := A - X;

**WRITE(A);**

**READ(B);**

B := B + X;

**WRITE(B);**

END

■ 该事务的DB访问操作为：

**READ(A);**

**WRITE(A);**

**READ(B);**

**WRITE(B);**

对该事务而言，数据库中数据的一致性就是指：帐号A和帐号B的总金额之和不变



## 4.2 事务

### ■ 二、事务的性质

- 原子性(Atomicity): 在一个事务中, 所有的数据库访问操作是一个不可分割的操作序列, 事务中的操作**要么全做要么全不做(nothing or all)**
  - e.g. 从ATM机取款时, “发钱登记”与“扣款”操作必须组成原子的事务。
  -
- 一致性(Consistency): 事务在功能上必须使DB从**一致状态(consistent state)变成另一个一致状态**。即DB中的数据必须满足已定义的完整性约束和业务规则。
  - e.g. 银行DB: 某帐号上的收支之差应始终等于余额。





## 4.2 事务

### ■ 二、事务的性质(cont.)

- **隔离性(Isolation)**: 多个事务并发执行时彼此不受影响, 就好象各个事务独立执行一样。
  - e.g. 民航DB: 某航班就剩一张机票, 有两个客户同时提出购买请求, 结果应是一个买到, 一个买不到。
- **持久性(Durability)**: 事务一旦成功执行, 其对DB的影响应是持久的, 即使DB发生故障也应保留这个事务的执行结果。
  - e.g. 银行的存款数据应是持久的。

□ 数据库管理系统通过其**事务管理子系统** (含**并发控制子系统**)、**恢复管理子系统**、**数据完整性保护子系统**来实现事务的**原子性(A)**、**一致性(C)**、**隔离性(I)**和**持久性(D)**。



## 4.2 事务

### ■ 三、事务的两种结束方式

- 提交(Commit): 全做事务中的操作。
- 回滚(Rollback): 全不做事务中的操作（部分已执行的操作要撤消）。
- 提交和回滚可以是显式的、也可以是隐式的：
  - 当发出COMMIT语句/ROLLBACK语句时显式提交/回滚当前事务；
  - 当发出一个DDL语句时，前后均隐式提交一个事务；
  - 当用户撤消对DBMS的连接时，当前事务隐式提交；
  - 当用户进程异常中止时，当前事务隐式回滚。



# 目录 Contents

---

- 4.1 数据库管理系统结构简介
- 4.2 事务
- **4.3 DBMS的进程结构**
- 4.4 DBMS的系统结构
- 4.5 数据目录 (字典)



## 4.3 DBMS的进程结构

### ■ 一、DBMS和操作系统

- DBMS是操作系统的用户。DBMS在操作系统上可以有多种不同的运行方式。
- DBMS需要操作系统的服务：创建和撤销进程，进程通信，读写磁盘，分配内存等。
- 操作系统不同的服务方式：DBMS在实现技术和实现方法上有差异。



## 4.3 DBMS的进程结构

### ■ 二、应用进程与DBMS进程

- **进程(Process)**是OS中的重要基本概念，是指独立程序代码的一次动态执行。不论是用户应用程序还是DBMS代码均作为OS的一个个进程而执行的。
- **应用进程(Application Process)**：也称**用户进程(User Process)**，对应某个应用程序的一次动态执行。
- **DBMS进程(DBMS Process)**：对应DBMS代码的一次动态执行。分为：
  - **核心进程或服务进程(Server Process)**
  - **后台进程(Background Process)**



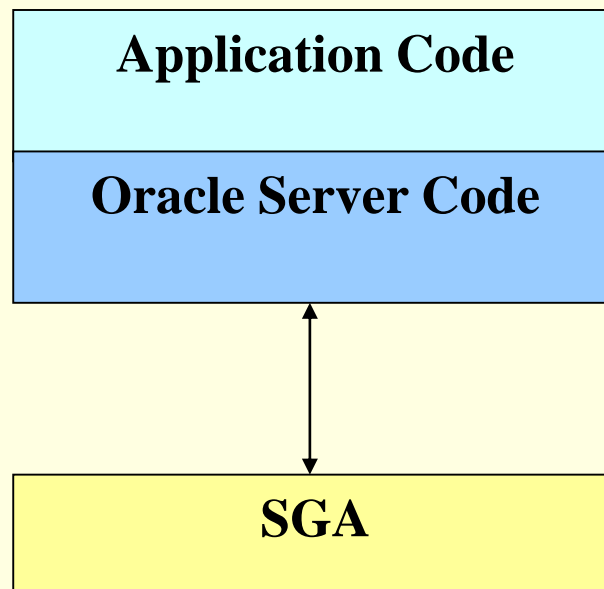
## 4.3 DBMS的进程结构

- 在Oracle中，将DBMS进程和系统全局区（System Global Area, SGA）称为一个Oracle实例(Instance)。
- 当Oracle启动后，称启动了一个Oracle实例。
- SGA是DBMS在内存开辟的一个区域，包括：
  - DB Buffer Cache
  - Redo Log Buffer
  - Shared Pool（共享SQL区、DD存储区，etc.）
  - Other information（队列、进程间通信信息，etc.）



## 4.3 DBMS的进程结构

- 三、几种典型的进程结构实现方案 (以Oracle为背景介绍)
  - 单进程结构 / 单用户结构 / 单用户Oracle
    - 应用代码和DBMS代码结合成单个进程而执行。e.g. MS-DOS上的单用户Oracle。



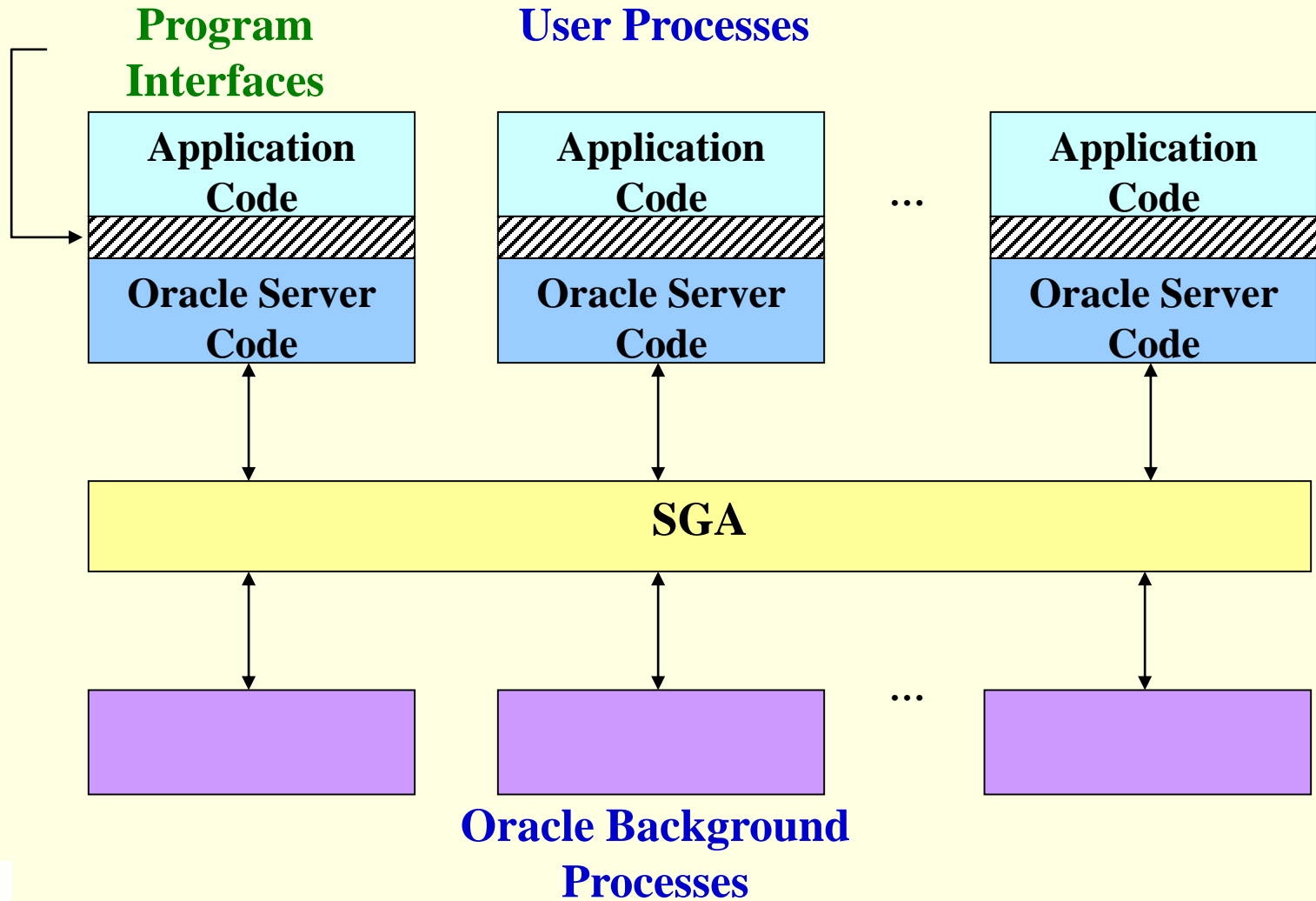
## 4.3 DBMS的进程结构

- 多进程结构 / 多用户结构 / 多用户Oracle
  - 每个连接DBMS的用户应用都对应一个用户进程，且使用多个进程/线程来执行DBMS。
- 1. 应用代码与DBMS代码组成同一个进程 / User/Server相结合的进程结构 / 单任务:
  - 应用代码与DBMS代码在同一个进程（称用户进程）中运行，彼此之间有程序接口维护隔离及传送数据。





## 4.3 DBMS的进程结构

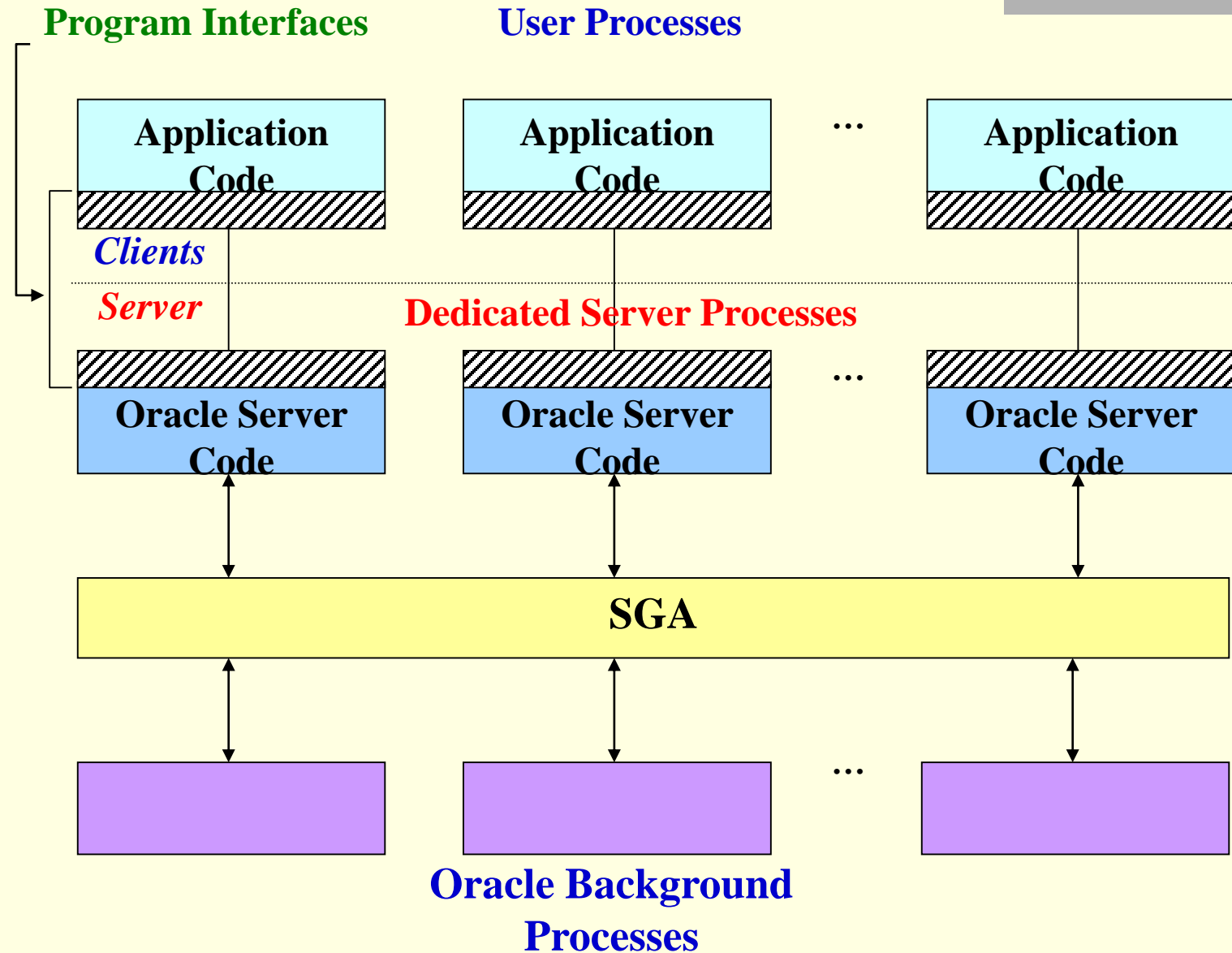


## 4.3 DBMS的进程结构

- 2、一个应用代码对应一个DBMS核心进程 / 使用专用服务器进程的结构 / 两任务Oracle
  - 为每个应用进程建立一个DBMS核心进程，称专用服务器进程（Dedicated Server Process）。



## 4.3 DBMS的进程结构

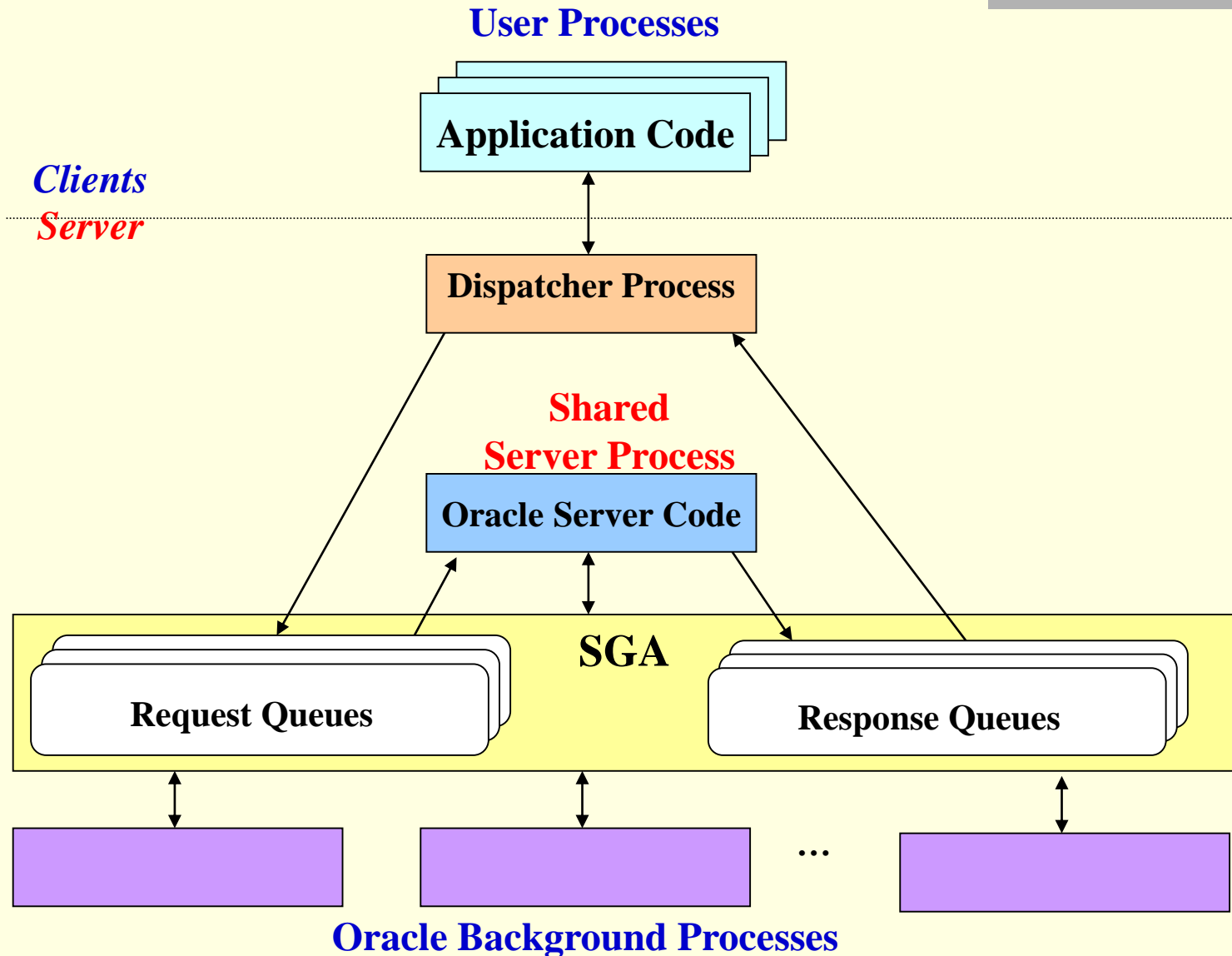


## 4.3 DBMS的进程结构

- 3、单核心进程、多线程的DBMS进程结构 / 使用共享服务器进程的结构
- 线程/线索(Thread): 是现代OS引入的一个新概念, 称轻量进程(Lightweight Process)。
  - 一个进程中可创建多个可切换的线程, 线程共享所属进程的(内存)资源, 具有较少的私有资源, 因此切换开销较小。
  - 进程是资源分配的单位, 而线程是处理机调度的单位。
  - 线程机制是OS的任务粒度(Task Granularity)变小、并发度提高(可实现进程内并发)。
  - 线程机制可在OS核心中(核心态)实现, 也可在OS的用户进程中(用户态)实现。
- 多线程DBMS (Multithreading DBMS): 不使用OS提供的多线程机制, 而由DBMS自己实现多线程机制。
  - e.g. Oracle中, 许多应用进程共同连接到**调度进程(Dispatcher Process)**, 由调度进程将用户请求发送到共享服务器进程。



## 4.3 DBMS的进程结构



# 目录 Contents

---

- 4.1 数据库管理系统结构简介
- 4.2 事务
- 4.3 DBMS的进程结构
- **4.4 DBMS的系统结构**
- 4.5 数据目录 (字典)



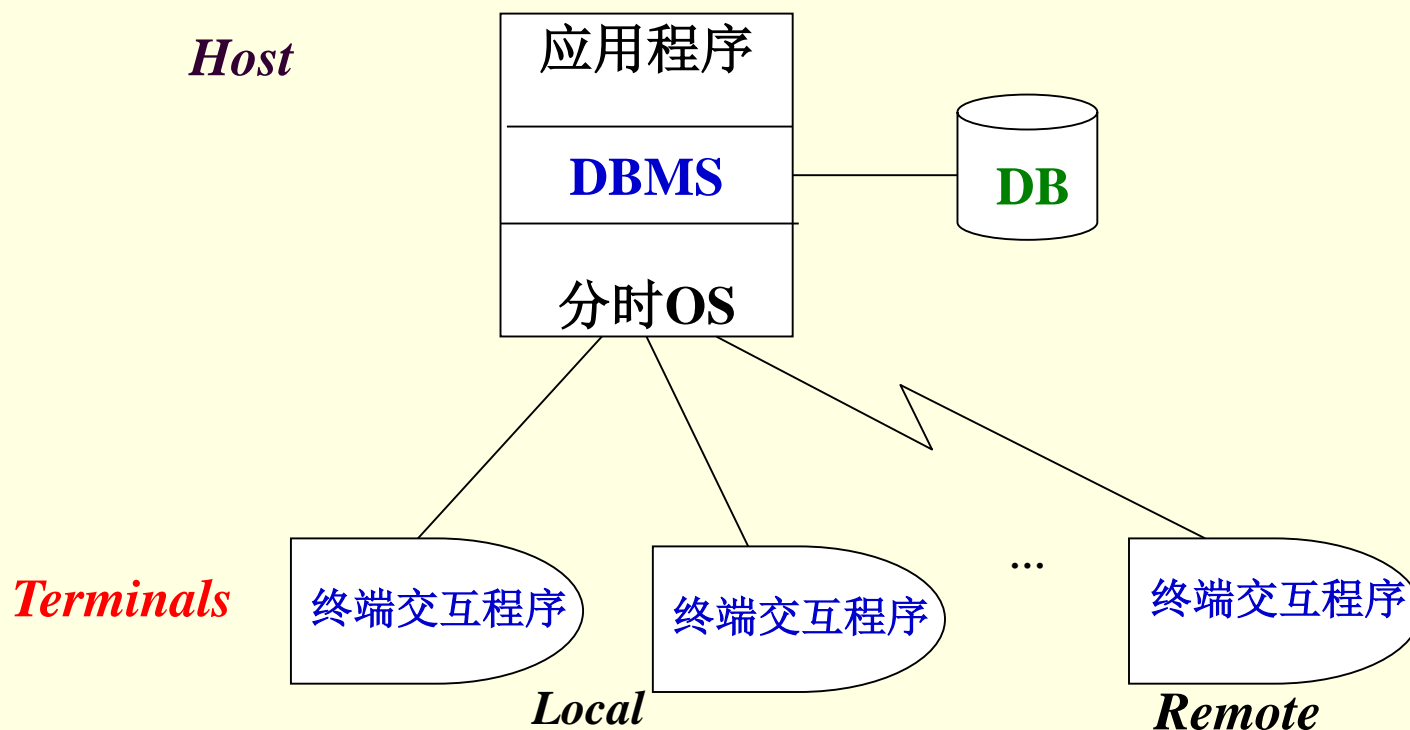
## 4.4 DBMS的系统结构

- 严格地说来应是数据库系统的体系结构  
(DB System Architecture)
- 数据库系统的结构演变与发展的驱动力：
  - 需求：用户的应用需求，市场因素， etc.
  - 技术：DB的运行支撑环境与平台(硬件、软件、网络， etc.)
- 数据库系统的结构可按DB的特点来分类：
  - 集中式数据库：数据集中存储；由DBMS集中管理
  - 分布式数据库：数据分布存储、相互关联；传统上认为应由DDBMS统一管理
    - 物理上分布、逻辑上集中
    - 物理上分布、逻辑上分布



## 4.4 DBMS的系统结构

- 一、集中式数据库系统结构
  - 运行于分时系统环境(即主机/终端系统)。





## 4.4 DBMS的系统结构

- 分时系统环境下的集中式结构
  - 一个主机带多个终端的多用户结构
  - 数据库系统，包括应用程序、DBMS、数据，都集中存放在主机上，所有处理任务都由主机来完成
  - 各个用户通过主机的终端并发地存取数据库，共享数据资源。
  - 特点：数据集中，数据管理集中。



## 4.4 DBMS的系统结构

### ■ 分时系统环境下的集中式结构(cont.)

#### ■ 优点

- 易于管理、控制与维护。

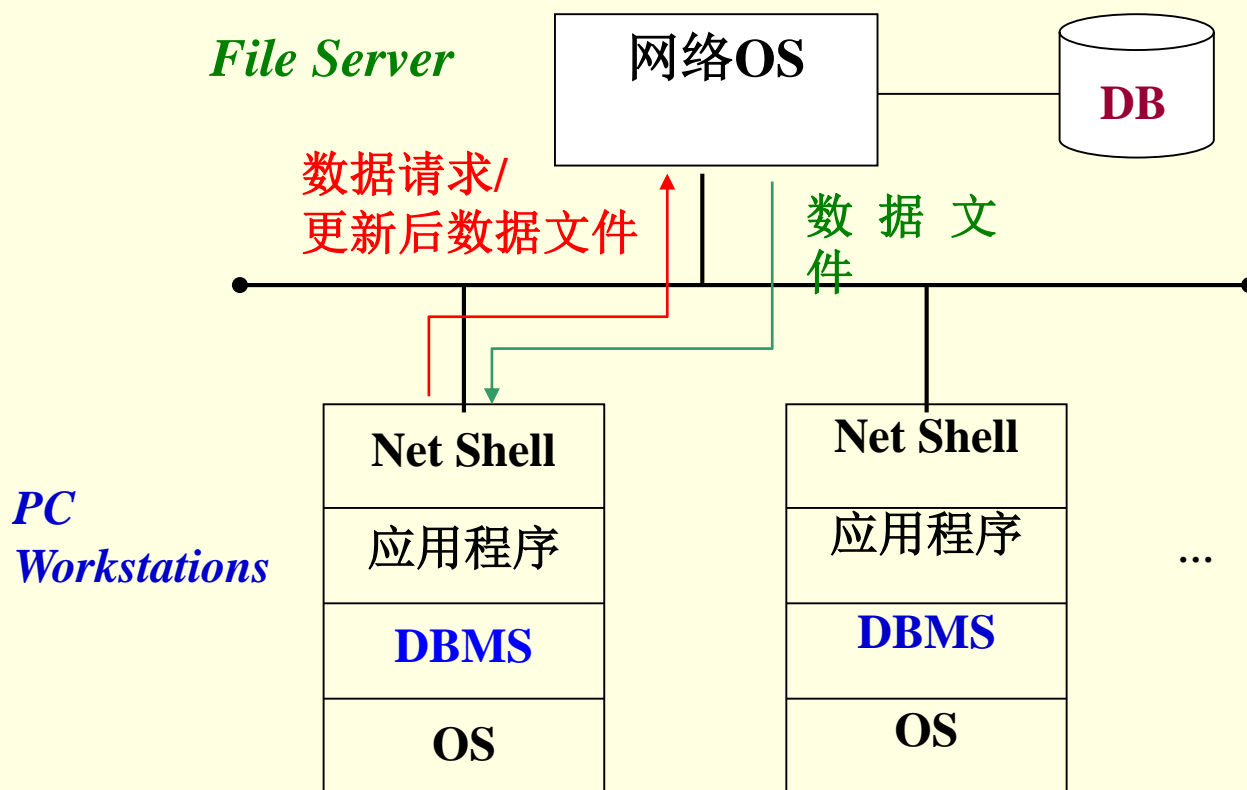
#### ■ 缺点

- 当终端用户数目增加到一定程度后，主机的任务会过分繁重，成为瓶颈，从而使系统性能下降。
- 系统的可靠性依赖主机,当主机出现故障时，整个系统都不能使用。



## 4.4 DBMS的系统结构

- 一、集中式数据库系统结构 (cont.)
  - 运行于PC或PC LAN环境 (单用户版或多用户版)。



## 4.4 DBMS的系统结构

### ■ 运行于PC或PC LAN环境数据库系统结构

#### ■ 特点：数据集中；处理集中

- 整个数据库系统(应用程序、DBMS、数据)装在一台计算机上，为一个用户独占，不同机器之间不能共享数据。

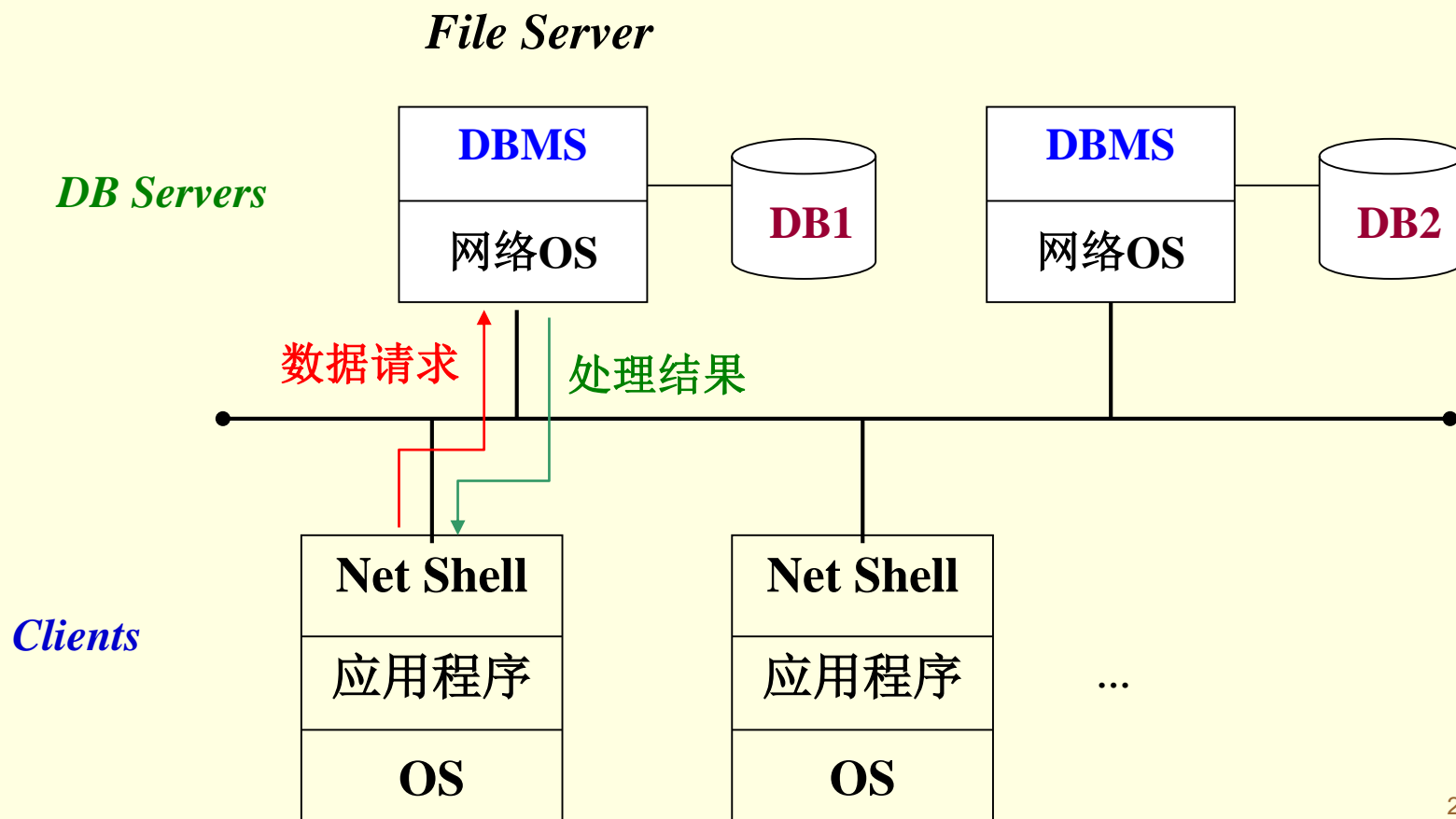
#### ■ 早期的数据库系统（文件系统）

- 例如一个企业的各个部门都使用本部门的机器来管理本部门的数据，各个部门的机器是独立的。由于不同部门之间不能共享数据，因此企业内部存在大量的冗余数据。例如人事部门、会计部门、技术部门必须重复存放每一名职工的一些基本信息（职工号、姓名等）。



## 4.4 DBMS的系统结构

- 一、集中式数据库系统结构 (cont.)
  - 运行于Client/Server环境 (两层)。



## 4.4 DBMS的系统结构

### ■ 运行于Client/Server环境的数据库系统

#### ■ 把DBMS功能和应用分开

- 网络中某个（些）结点上的计算机专门用于执行DBMS功能，称为数据库服务器，简称**服务器**
- 其他结点上的计算机安装DBMS的外围应用开发工具，用户的应用系统，称为**客户机**

#### ■ 客户机与服务器功能划分的原则

- 客户机提供多样化的用户接口，执行应用程序，对服务器提出服务请求等；
- 服务器只完成客户机所委托的公共服务；
- 客户机与服务器间的数据交换量应尽可能地少；
- 消除瓶颈，提高全系统的性能。



## 4.4 DBMS的系统结构

- 运行于Client/Server环境的数据库系统的特点
  - 数据集中；处理分布
  - 客户端的用户请求被传送到数据库服务器，数据库服务器进行处理后，只将结果返回给用户，从而显著减少了数据传输量。
  - 数据库更加开放
    - 客户与服务器一般都能在多种不同的硬件和软件平台上运行
    - 可以使用不同厂商的数据库应用开发工具



## 4.4 DBMS的系统结构

### ■ 运行于Client/Server环境的数据库系统的缺点

#### ■ “胖客户”问题

- 系统安装复杂，工作量大。
- 应用维护困难，难于保密，造成安全性差。
- 相同的应用程序要重复安装在每一台客户机上，从系统总体来看，大大浪费了系统资源。
- 系统规模达到数百数千台客户机，它们的硬件配置、操作系统又常常不同，要为每一个客户机安装应用程序和相应的工具模块，其安装维护代价便不可接受。





## 4.4 DBMS的系统结构

- 一、集中式数据库系统结构 (cont.)
  - 运行于Client/Server环境 (三层)。
    - 三层结构 (Three-tier Application Architecture)



## 4.4 DBMS的系统结构

### ■ 三层结构

#### ■ 客户端

- 浏览器软件、用户界面
- 浏览器的界面统一，广大用户容易掌握
- 大大减少了培训时间与费用。

#### ■ 服务器端

- Web服务器、应用服务器
- 数据库服务器等
- 大大减少了系统开发和维护代价，能够支持数万甚至更多的用户



## 4.4 DBMS的系统结构

---

- 集中式数据库系统的缺点
  - 通信开销大
  - 性能差：容易出现单点失效问题
  - 可用性差
  - 可扩充性差



## 4.4 DBMS的系统结构

### ■ 二、分布式数据库系统结构

#### ■ 物理上分布、逻辑上集中的分布式数据库系统

- 数据库中的数据在逻辑上是一个整体，但物理地分布在计算机网络的不同结点上。
- 网络中的每个结点都可以独立处理本地数据库中的数据，执行局部应用。
- 同时也可以同时存取和处理多个异地数据库中的数据，执行全局应用。
- 特点：有全局数据模式；强调统一管理



## 4.4 DBMS的系统结构

### ■ 优点

- 适应了地理上分散的公司、团体和组织对于数据库应用的需求。

### ■ 缺点

- 数据的分布存放给数据的处理、管理与维护带来困难。
- 当用户需要经常访问远程数据时，系统效率会明显地受到网络传输的制约。



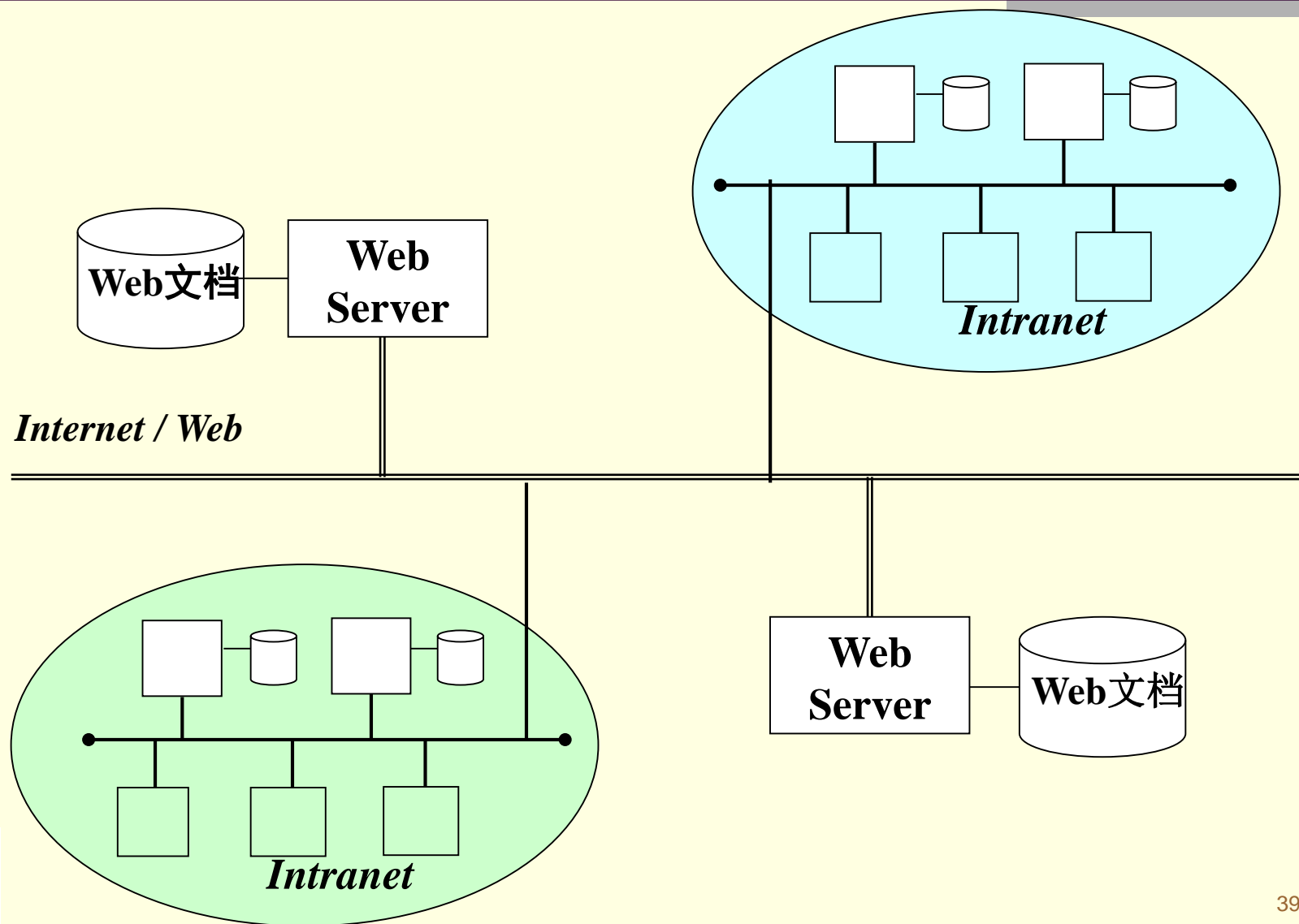
## 4.4 DBMS的系统结构

### ■ 二、分布式数据库系统结构(cont.)

- 物理上分布、逻辑上分布的分布式数据库系统
- 特点：
  - 无全局数据模式；
  - 强调结点自治
- 也称联邦式系统(Federated Distributed System)。
- 随着Web的迅速普及，一个全球信息网络正在形成，使得有可能构成一个全球联邦式系统。
- 一系列新的概念、技术正在探讨、研究中。



## 4.4 DBMS的系统结构



# 目录 Contents

---

- 4.1 数据库管理系统结构简介
- 4.2 事务
- 4.3 DBMS的进程结构
- 4.4 DBMS的系统结构
- 4.5 数据目录 (字典)





## 4.5 数据目录

- **数据目录**(Date directory, DD或Date catalog) 也有称**数据字典**，是一组关于数据的数据（即**元数据** metadata），其中包含了数据库的各种定义信息、描述信息和统计信息。



# 4.5 数据目录

## ■ 一、DD的内容

- DB用户名
- 每个用户所授的特权（Privileges）和角色（Roles）
- 各种模式对象（表、视图、快照、索引、簇集、序列、同义词、过程、触发器、函数、包等）的定义
- 关于完整性约束的信息
- 列的缺省值
- 有关DB中对象的空间分布信息及当前使用情况
- 审计信息
- DB动态性能和统计信息



# 4.5 数据目录

## 二、DD的结构

- 以一组基表存储所有基础信息，这些表有系统自动创建，为DBMS所有、所用。
- 在这组基表上定义了每个用户可存取的一组只读视图，系统自动创建，供用户查询。
- Oracle中，分三类：
  - DBA-前缀视图：e.g. DBA-TABLES----DB中全部表的说明。
  - ALL-前缀视图：e.g. ALL-TABLES----用户可存取的表的说明。
  - USER-前缀视图：e.g. USER-TABLES----用户拥有的表的说明。
- 一组虚表(Virtual Tables)，用于记录当前数据库活动的动态性能，DBA可查询，也可以在这些表上定义视图，授权给用户查询。
- Oracle中，V\$前缀视图：
  - e.g. V\$PROCESS----当前活动进程信息。

