1. 数据库系统概论
2. 数据库系统的地位

答：数据库技术产生于六十年代末，是数据管理的最新技术，是计算机科学的重要分支。

数据库技术是信息系统的核心和基础，它的出现极大地促进了计算机应用向各行各业的参透。

数据库的建设规模、信息量的大小和使用频度已成为衡量一个国家信息化程度的重要标志。

1. 数据（Data）

答：数据是数据库中存储的基本对象

数据的定义：描述事物的符号记录，包括数据的表现形式和数据的解释两部分

数据的表现形式：文字、图形、图像、声音、数字

数据的特点：数据与其语义是不可分的

1. 数据库（DB, DataBase）

答：数据库，是长期存储在计算机内，有组织的，可共享的，大量数据的集合

数据库的基本特征：

组织性（数据按一定的数据模型组织、描述和存储）

共享性、冗余度较小、独立性高、易扩展性

（永久存储、有组织、可共享）

1. 数据库管理系统（DBMS, DataBase Management System）

答：数据库管理系统，位于用户与操作系统之间的一层数据管理软件；基础软件，是一个大型复杂的软件系统。

数据库管理系统（DBMS）的主要功能（P5）：

数据定义功能

数据组织、存储和管理

数据操纵功能

数据库的事务管理和运行管理

数据库的建立和维护功能

其他功能

1. 数据库系统（DBS, DataBase System）

答：数据库系统，是由数据库，数据库管理系统（及其应用开发工具），应用程序和数据库管理员（DBA）组成，是存储、管理、处理和维护数据的系统。

数据库系统的四大特点（P13）：

数据结构化（数据库系统实现整体数据的结构化，这是数据库的主要特征之一，也是数据库系统与文件系统的本质区别）

数据的共享性高、冗余度低且易扩充

数据独立性高

数据由数据库管理系统统一管理和控制

6.管理信息系统（MIS, Management Information System）

答：管理信息系统，是现代管理系统的一个重要组成部分，它是一种在管理中进行资料处理、存储、调用的系统，它从环境和系统内部获取数据，进行筛选、组织和选择，并作出信息而迅速地传递给管理者，各级管理人员借助这些信息进行有效的管理活动。

（MIS，是一个由人、计算机及其他外围设备等组成的能进行信息的收集、传递、存贮、加工、维护和使用的系统）

1. 数据管理技术的产生和发展（三个发展阶段及其特点）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | 人工管理阶段 | 文件系统阶段 | 数据库系统阶段 |
| 背景 | 应用背景 | 科学计算 | 科学计算、  数据管理 | 大规模数据管理 |
| 硬件背景 | 无直接存取存储设备 | 磁盘、磁鼓 | 大容量磁盘、  磁盘阵列 |
| 软件背景 | 没有操作系统 | 有文件系统 | 有数据库管理系统 |
| 处理方式 | 批处理 | 联机实时处理、  批处理 | 联机实时处理、  分布处理、批处理 |
| 特点 | 数据的管理者 | 用户（程序员） | 文件系统 | 数据库管理系统 |
| 数据面向的对象 | 某一应用程序 | 某一应用 | 现实世界（部门、企业、跨国组织） |
| 数据的共享程度 | 无共享，  冗余度极大 | 共享性差，  冗余度大 | 共享性高，  冗余度小 |
| 数据的独立性 | 不独立，完全  依赖于程序 | 独立性差 | 具有高度的物理独立性和一定的逻辑独立性 |
| 数据的结构化 | 无结构 | 记录内有结构、  整体无结构 | 整体结构化，  用数据模型描述 |
| 数据控制能力 | 应用程序自己控制 | 应用程序自己控制 | 有数据库管理系统提供数据安全性、完整性、并发控制  和恢复能力 |

1. 数据库系统的组成

答：硬件平台及数据库、软件、人员

9.数据模型的分类

答：一：概念模型（信息模型）

它是按用户的观点来对数据和信息建模，用于数据库设计

二：逻辑模型和物理模型

逻辑模型主要包括网状模型、层次模型、关系模型、面向对象模型等，按计算机系统的观点对数据建模，用于DBMS实现。

物理模型是对数据最底层的抽象，描述数据在系统内部的表示方式和存取方法，在磁盘或磁带上的存储方式和存取方法。面向计算机系统，物理模型的具体实现是DBMS的任务。

（数据模型是数据库系统的核心和基础）

1. 数据模型的组成要素（P18）

答：数据结构、数据操作、数据的约束条件

1. 概念模型及其相关基本概念（P16）

答：实体：客观存在并可相互区别的事物称为实体。

属性：实体所具有的某一特性称为属性。

码：唯一标识实体的属性集称为码。

实体型：用实体名及其属性名结合来抽象和刻画同类实体，称为实体型。

例：学生（学号，姓名，性别，出生年月，所在院系）

实体集：同一类型实体的结合称为实体集。例：全体学生

联系：实体之间的联系通常是指不同实体集之间的联系

一对一、一对多、多对多

1. 概念模型的一种表示方法（ER图）

答：实体型：矩形

属性：椭圆形

联系：菱形

（菱形框内写联系名，同时在无向边旁标上联系的类型，1:1、1:n、m:n，

联系本身也是一种实体型，可以有属性）

1. 概念模型的用途

答：概念模型用于信息世界的建模

是实现世界到机器世界的一个中间层次

是数据库设计的有力工具

数据库设计人员和用户之间进行交流的语言

1. 常用的数据模型

答：关系模型、层次模型、网状模型

1. 关系模型的基本概念

答：关系模型是最重要的一种数据模型。关系数据库系统采用关系模型作为数据的组织方式。

关系：一个关系对应通常说的一张表

元组：表中的一行即为一个元组

属性：表中的一列即为一个属性，给每个属性起一个名称即属性名

主码：表中的某个属性组，它可以唯一确定一个元组

候选码：当关系中有多个属性组都能唯一确定一个元组，则这些属性组称为候选码，可 以从候选码中选择一个做主码。

全码：

域：属性的取值范围

分量：元组中的一个属性值

关系模式：对关系的描述。例：学生（学号，姓名，年龄，性别，系，年级）

1. 关系模型的数据结构

答：在用户观点下，关系模型中数据的逻辑结构是一张二维表，它是由行和列组成。

1. 关系模型的数据操纵和完整性约束

答：关系模型的数据操纵，包括：查询、插入、删除、更新

关系的完整性约束条件，包括：

实体完整性、（主码的值不能重复，主码的属性非空）

参照完整性、（外码取值必须或者取空值，或者是被参照表中存在的）

用户定义完整性

1. 关系数据模型的优缺点

答：优点：

它是建立在严格的数学概念的基础上的；

关系模式的概念单一；

实体和各类联系都用关系来表示；

对数据的检索结果也是关系；

关系模型的存取路径对用户透明；

具有更高的数据独立性，更好的安全保密性；

简化了程序员的工作和数据库开发建立的工作。

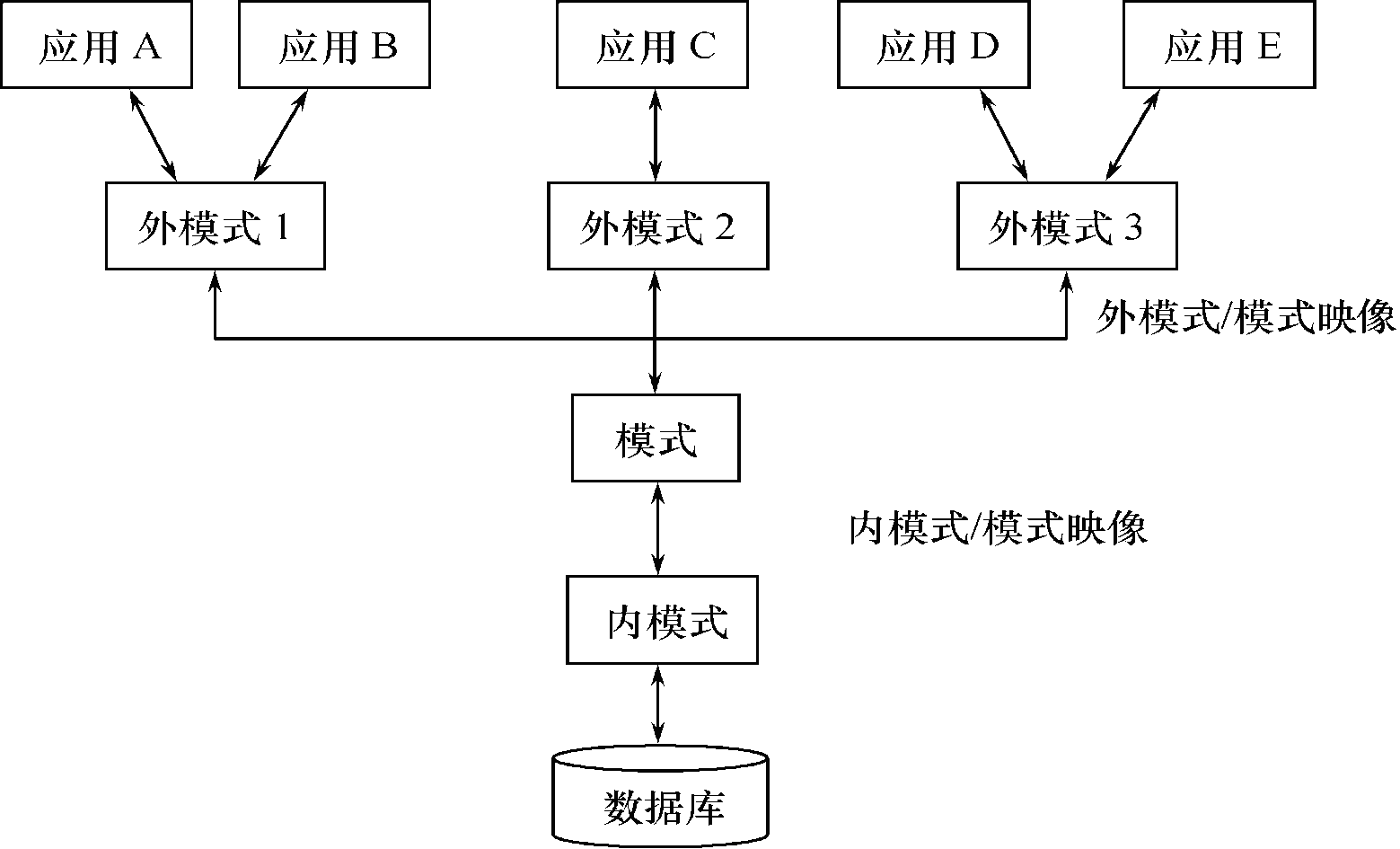
缺点：

存取路径对用户透明导致查询效率往往不如非关系数据模型

为提高性能，必须对用户的查询请求进行优化增加了开发DBMS的难度。

1. 数据库系统的三级模式（每个模式的定义和特点）

答：模式、外模式、内模式



模式：（也称逻辑模式），是数据库中全体数据的逻辑结构和特征的描述，是所有用户的公共数据视图，综合了所有用户的需求。

一个数据库只有一个模式

模式的地位：是数据库系统模式结构的中间层

与数据的物理存储细节和硬件环境无关

与具体的应用程序、开发工具及高级程序设计语言无关

外模式：（也称子模式或用户模式），是数据库用户（包括应用程序员和最终用户）使用的局部数据的逻辑结构和特征的描述；是数据库用户的数据视图，是与某一应用有关的数据的逻辑表示。

外模式的地位：介于模式与应用之间

模式与外模式的关系：一对多

外模式通常是模式的子集

一个数据库可以有多个外模式。反映了不同的用户的应用需求、看待数据的方 式、对数据保密的要求

对模式中同一数据，在外模式中的结构、类型、长度、保密级别等都可以不同

外模式与应用的关系：一对多

同一外模式也可以为某一用户的多个应用系统所使用

但一个应用程序只能使用一个外模式

内模式：（也称存储模式），是数据物理结构和存储方式的描述，是数据在数据库内部的 表示方式。

一个数据库只有一个内模式

1. 两极映像（如何保证独立性P29）

答：外模式/模式映像，模式/内模式映像

外模式/模式映像：

模式：描述的是数据的全局逻辑结构

外模式：描述的是数据的局部逻辑结构

同一个模式可以有任意多个外模式

每一个外模式，数据库系统都有一个外模式/模式映像，定义外模式与模式之间的对应关系

映像定义通常包含在各自外模式的描述中

保证数据的逻辑独立性：

当模式改变时，数据库管理员修改有关的外模式/模式映像，使外模式保持不变

应用程序是依据数据的外模式编写的，从而应用程序不必修改，保证了数据与程序的逻辑独立性，简称数据的逻辑独立性

模式/内模式映像：

模式/内模式映像定义了数据全局逻辑结构与存储结构之间的对应关系

数据库中模式/内模式映像是唯一的

该映像定义通常包含在模式描述中

保证数据的物理独立性：

当数据库的存储结构改变了，数据库管理员修改模式/内模式映像，使模式保持不变

应用程序不受映像。保证了数据与程序的物理独立性，简称数据的物理独立性

1. 数据库的二级映像

答：保证了数据库外模式的稳定；

从底层保证了应用程序的稳定性，除非应用需求本身发生变化，否则应用程序一般不需要修改。

1. 数据的存取由DBMS管理

答：用户不必考虑存取路径等细节

简化了应用程序的编制

大大减少了应用程序的维护和修改