

专业课

计算机

数据库管理

袁礼

华图网校

版权所有 盗版必究

目录

1. 数据库系统概述.....	1
1.1 概念和术语.....	1
1.2 数据管理技术的产生和发展.....	3
1.3 数据库管理技术新进展.....	5
1.4 数据库系统的特点.....	6
1.5 数据库系统结构.....	6
2. 数据模型.....	9
2.1 数据描述.....	9
2.2 数据模型.....	11
3. 关系数据库.....	15
3.1 关系模型.....	15
3.2 关系的完整性.....	18
3.3 关系规范化.....	19
3.4 关系代数.....	19
4. 数据库设计.....	28
1 数据库设计概述.....	28
2 概念模型与 E-R 方法	30
3 关系数据库设计理论.....	31
4、 数据库设计实例.....	35
5. SQL Server 数据库管理	38
5.1 SQL Server 的构成	38
5.2 创建、修改和删除数据库.....	41
5.3 备份和恢复数据库.....	48
6、 SQL 结构化查询语言.....	52
6.1SQL 概述	52
6.2 常见 SQL 语句	52
6.3SQL 查询（重点）	54

本章内容：

- 数据库系统概述
- 数据模型
- 关系数据库
- 数据库设计
- 数据库管理

1. 数据库系统概述

什么是数据库？

数据库技术是最新的数据管理技术，是计算机科学的一个重要分支。在计算机应用的三大领域（科学计算，数据处理和过程控制）中，数据处理占约其中的 70%，而数据库技术就是作为一门数据处理技术发展起来的，是目前应用最广的技术之一。它已成为计算机信息系统的核心技术和重要基础。

1.1 概念和术语

一. 数据（Data）

数据是描述事物的符号记录，是数据库中存储的基本对象。

二. 数据库（DataBase，简称 DB）

数据库，顾名思义，就是存放数据的仓库。只不过这个仓库是在计算机存储设备上，而且数据是按一定的格式存放的。也就是说，数据库是具有统一的结构形式并存放于统一的存储介质内的多种应用数据的集成，并可被各个应用程序所共享。

（一）数据库管理系统（DataBase Management System，简称 DBMS）

数据库管理系统：一种系统软件，负责数据库中的数据组织、数据操纵、数据维护、控制及保护和数据服务等，是数据库系统的核心。

它的主要功能包括以下几个方面：

1. 数据模式定义。
2. 数据存取的物理构建。
3. 数据操纵。
4. 数据的完整性、安全性定义与检查。

5. 数据库的并发控制与故障恢复。

(2) 为完成其基本功能，数据库管理系统提供相应的数据语言，它们是：

1. 数据定义语言（Data Definition Language，简称 DDL)

该语言负责数据的模式定义与数据的物理存取构建。

2. 数据操纵语言（Data Manipulation Language，简称 DML)

该语言负责数据的操纵，包括查询及增删、改等操作。

3. 数据控制语言（Data Control Language，简称 DCL)

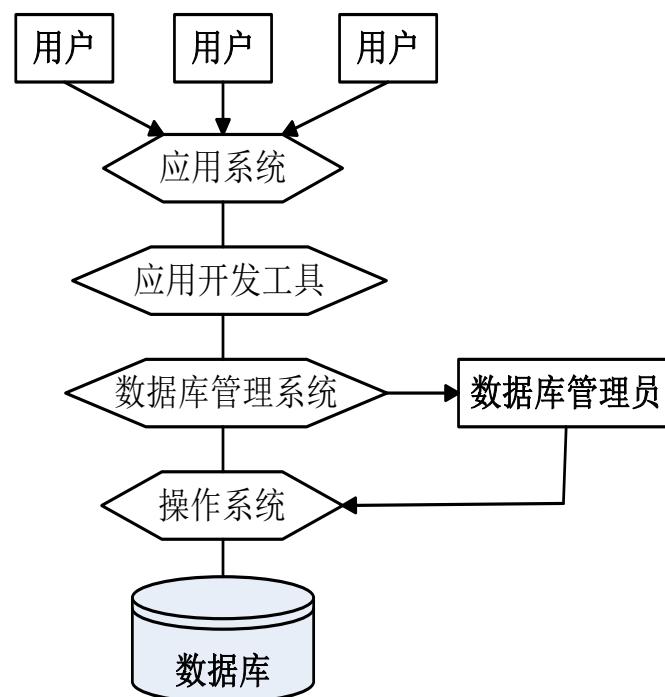
该语言负责数据完整性、安全性的定义与检查以及并发控制、故障恢复等功能。

(二) 数据库系统（DataBase System ，简称 DBS)

数据库系统是指安装和使用数据库技术的计算机系统，一般由数据库、数据库管理系统（及其开发工具）、应用系统和数据库管理员（DataBase Administrator ，简称 DBA）构成。

DBA 的职责：

1. 数据库安装
2. 数据库配置和管理
3. 权限设置和安全管理
4. 监控和性能调节
5. 备份和恢复
6. 解决一般的问题



数据库系统（DataBase System，简称 DBS）

1.2 数据管理技术的产生和发展

随着计算机软件、硬件技术的发展，数据处理量的规模日益扩大，数据处理的应用需求越来越广泛，数据管理技术的发展也不断变迁，经历了从人工管理、文件系统、数据库技术 3 个主要发展阶段。数据库技术在当前网络应用环境下又有了新的进展。

人工管理阶段（20 世纪 50 年代）

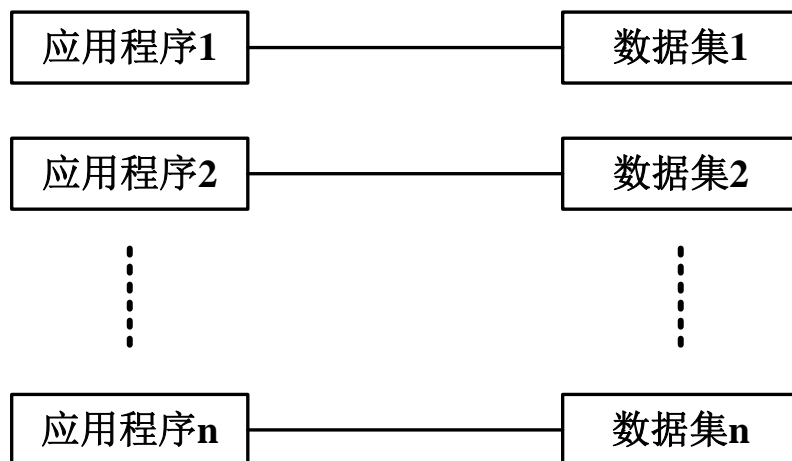
1.20 世纪 50 年代中期以前，计算机主要用于科学计算。

2.外部存储器只有纸带、卡片、磁带，无像磁盘这样的可以随机访问、直接存取的外部存储设备。软件的状况是没有操作系统，没有专门管理数据的软件，数据由计算或处理它的程序自行携带。数据管理任务，包括存储结构、存储方法、输入/输出方式等完全由程序设计者负责。

特点：

1. 数据与程序不具有独立性，一组数据对应一组程序。
2. 数据不能长期保存。
3. 一个程序中的数据无法被其他程序使用。
4. 程序之间存在大量重复数据，数据冗余大。

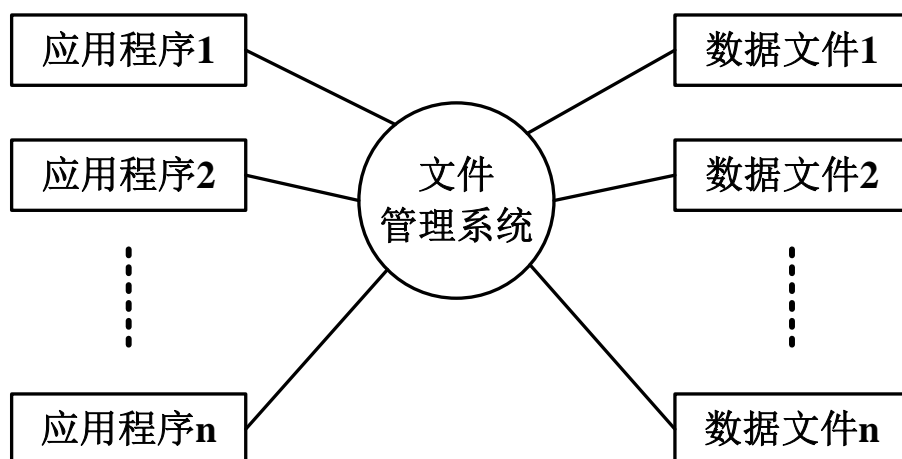
人工管理阶段（20 世纪 50 年代）



文件系统阶段（20 世纪 60 年代）

1.20 世纪 50 年代后期到 60 年代中期。计算机不仅用于科学计算,而且还用于大量的数据处理。直接存储设备，高级语言，操作系统。

2. 程序和数据有一定的独立性，数据文件可以长期保存。
3. 数据和程序相互依赖。数据文件是为满足特定的业务，或某部门的专门需要而设计，服务于某一特定的应用程序。
4. 同一数据项可能重复出现在多个文件中，数据冗余度大。
5. 数据容易造成不一致。



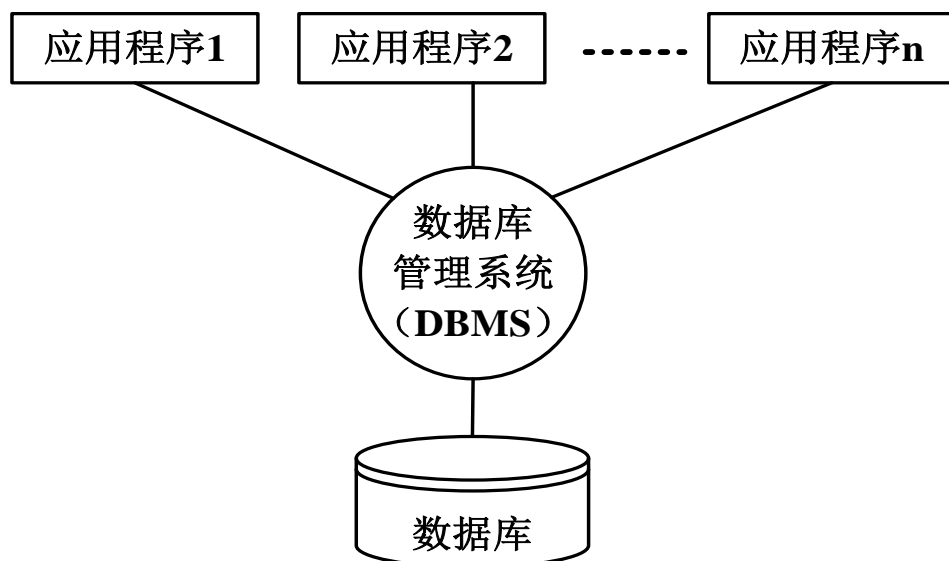
数据库系统阶段（20 世纪 60 年代后期）

1. 20 世纪 60 年代后期。计算机用于管理的规模更加庞大，应用越来越广泛。同时多种应用、多种语言共享数据集合的要求越来越强烈。

2. 标志：1968 年 IBM 的 IMS 是一个层次模型数据库。1969 年美国数据系统语言协会公布的 DBTG 报告，对研制开发网状数据库系统起到了推动作用。自 1970 年 IBM 公司的研究成果奠定了关系数据库理论基础。

3. 数据库技术的主要目的：

有效地管理和存储大量的数据资源，包括：提高数据的共享性，使多个用户能够同时访问数据库中的数据；减少数据冗余，以提高数据的一致性和完整性；提高数据与程序的独立性，从而减少应用程序的开发和维护代价。



1.3 数据库管理技术新进展

1.新型数据库系统带来了一个又一个数据库技术发展的新潮，但对于中、小数据库用户来说，由于很多高级的数据库系统的专业性要求太强，通用性受到一定的限制，在很大程度上推广使用范围也受到约束。

2.而基于关系模型的关系数据库系统功能的扩展与改善，分布式数据库、面向对象数据库、数据仓库等数据库技术的出现，构成了新一代数据库系统的发展主流。

（一）分布式数据库系统

分布式数据库是数据库技术与网络技术相结合的产物。

1.随着传统的数据库技术日趋成熟、计算机网络技术的飞速发展和应用范围的扩充，数据库应用已经普遍建立于计算机网络之上。这时集中式数据库系统表现出它的不足：数据按实际需要已在网络上分布存储，再采用集中式处理，势必造成通信开销大；应用程序集中在一台计算机上运行，一旦该计算机发生故障，则整个系统受到影响，可靠性不高；集中式处理引起系统的规模和配置都不够灵活，系统的可扩充性差。

2.在这种形势下，集中式数据库的“集中计算”概念向“分布计算”概念发展。

3、分布式数据库系统有两种：一种是物理上分布的，但逻辑上却是集中的。另一种在物理上和逻辑上都是分布的，也就是所谓联邦式分布数据库系统。

（二）面向对象数据库系统

1.将面向对象技术与数据库技术结合产生出面向对象的数据库系统。这是数据库应用发展的迫切需要，也是面向对象技术和数据库技术发展的必然结果。

2.面向对象的数据库系统必须支持面向对象的数据模型，具有面向对象的特性。一个面向对象的数据模型是用面向对象的观点来描述现实世界实体（对象）的逻辑组织、对象之间

的限制和联系等的模型。

3.另外，将面向对象技术应用到数据库应用开发工具中，使数据库应用开发工具能够支持面向对象的开发方法并提供相应的开发手段，这对于提高应用开发效率、增强应用系统界面的友好性、系统的可伸缩性、可扩充性等具有重要的意义。

（三）数据仓库

1.随着客户机 / 服务器技术的成熟和并行数据库的发展，信息处理技术的发展趋势是从大量的事务型数据库中抽取数据，并将其清理、转换为新的存储格式，即为决策目标把数据聚合在一种特殊的格式中。随着此过程的发展和完善，这种支持决策的、特殊的数据存储即被称为数据仓库（Data Warehouse）。

2.数据仓库领域的著名学者 W. H. Inmon 对数据仓库的定义是：数据仓库是支持管理决策过程的、面向主题的、集成的、稳定的、随时间变化的数据集合。

1.4 数据库系统的特点

1. 数据的集成性
2. 数据的高共享性与低冗余性
3. 数据独立性
4. 数据统一管理与控制

1.5 数据库系统结构

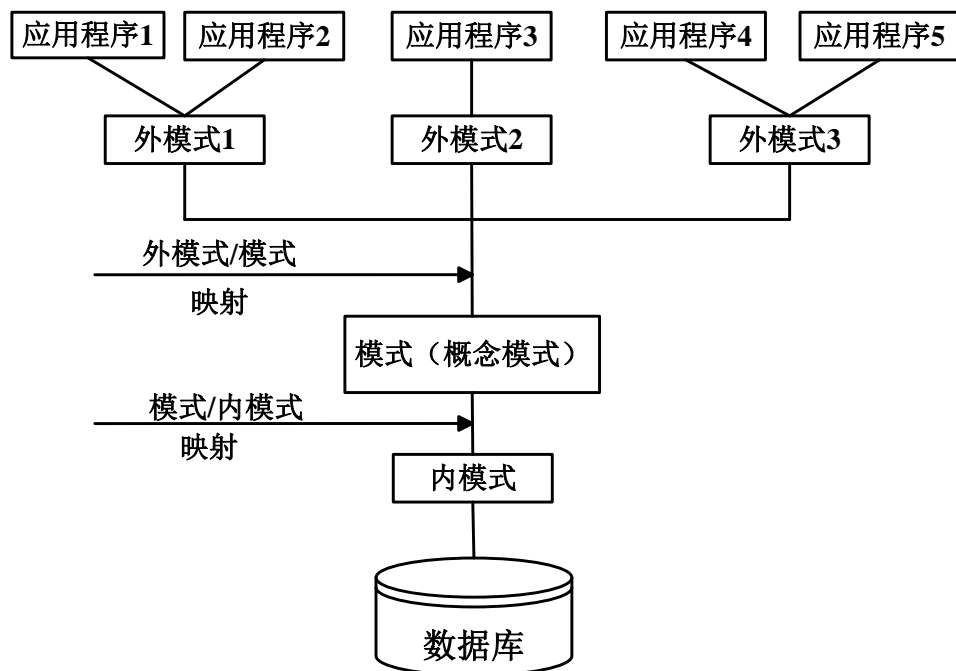
1.考察数据库系统的结构可以有多种不同的层次或不同的角度。

2.从数据库管理系统的角度看，数据库系统通常采用三级模式结构，这是数据库管理系统内部的体系结构。

3.从数据库最终用户角度看，数据库系统的结构分为单用户结构、主从式结构、分布式结构、客户/服务器结构（包括二层、三层或多层结构）等。这是数据库系统外部的体系结构。

1.5.1 数据库系统的三级模式结构

数据库系统的三级模式结构是指数据库系统是由外模式、模式和内模式三级构成。



模式 (Schema)

1. 模式也称逻辑模式或概念模式，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。它是数据库系统模式结构的中间层，不涉及数据的物理存储细节和硬件环境，与具体的应用程序，与所使用的应用开发工具及高级程序设计语言无关。

2. 实际上模式是数据库数据在逻辑级上的视图。一个数据库只有一个模式。数据库模式以某一种数据模型为基础，统一综合地考虑了所有用户的需求，并将这些需求有机地结合成一个逻辑整体。

外模式 (External Schema)

1. 外模式也称子模式或用户模式，它是数据库用户(包括应用程序员和最终用户)看见和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。

2. 外模式通常是模式的子集。一个数据库可以有多个外模式。由于它是各个用户的数据视图，如果不同的用户在不同应用需求、看待数据的方式、对数据保密的要求等方面存在差异，则他们的外模式描述就是不同的。即使对模式中同一数据，在外模式中的结构、类型、长度、保密级别等都可以不同。另一方面，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。

3. 外模式是保证数据库安全性的一个有力措施。每个用户只能看见和访问所对应的外模式中的数据，数据库中的其余数据对他们来说是不可见的。

内模式 (Internal Schema)

1. 内模式也称存储模式，它是数据物理结构和存储结构的描述。是数据在数据库内部的表示方式。例如，记录的存储方式是顺序存储、按照 B 树结构存储还是按 hash 方法存储；

索引按照什么方式组织；数据是否压缩存储，是否加密；数据的存储记录结构有何规定等。一个数据库只有一个内模式。

2.数据模式给出了数据库的数据框架结构，数据是数据库中的真正的实体，但这些数据必须按框架所描述的结构组织，以概念模式为框架所组成的数据库叫概念数据库（Conceptual DataBase），以外模式为框架所组成的数据库叫用户数据库（User's Database），以内模式为框架所组成的数据库叫物理数据库（Physical Database）。这三种数据库中只有物理数据库是真实存在于计算机外存中，其他两种数据库并不真正存在于计算机中，而是通过两种映射由物理数据库映射而成。

3.模式的三个级别层次反映了模式的三个不同环境以及它们的不同要求，其中内模式处于最底层，它反映了数据在计算机物理结构中的实际存储形式，概念模型处于中层，它反映了设计者的数据全局逻辑要求，而外模式处于最外层，它反映了用户对数据的要求。

1.5.2 数据库的二级映像功能

1.数据库系统的三级模式是对数据的三个抽象级别。它把数据的具体组织留给数据库管理系统（DBMS）管理，使用户能逻辑地、抽象地处理数据，而不必关心数据在计算机中的具体表示方式与存储方式。

2.而为了能够在内部实现这三个抽象层次的联系和转换，数据库系统在这三级模式之间提供了两层映像：外模式 / 模式映像和模式 / 内模式映像。正是这两层映射保证了数据库系统中的数据能够具有较高的逻辑独立性和物理独立性。

外模式 / 模式映像

1.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。对应于同一个模式可以有任意多个外模式。对于每一个外模式，数据库系统都有一个外模式 / 模式映像，它定义了该外模式与模式之间的对应关系。

2.当模式改变时，由数据库管理员对各个外模式 / 模式映像作相应改变，也可以使外模式保持不变，因为应用程序是依据数据的外模式编写的，从而应用程序也不必修改，保证了数据与程序的逻辑独立性。

模式 / 内模式映像

1.模式 / 内模式映像定义了数据全局逻辑结构与物理存储结构之间的对应关系。当数据库的存储结构改变时（例如换了另一个磁盘来存储该数据库），由数据库管理员对模式 / 内模式映像作相应改变，可以使模式保持不变，从而保证了数据的物理独立性。

1.5.3 数据库系统的组成

1. 硬件环境
2. 软件环境

3. 数据库

4. 人员

【练习】(C) 是长期存储在计算机内的有组织、可共享的数据集合。

A.数据库管理系统

B.数据库系统

C.数据库

D.文件组织

【练习】在文件系统管理阶段，数据 (B)

A.无独立性

B.独立性差

C.具有物理独立性

D.具有逻辑独立性

【练习】描述数据库全体数据的全局逻辑结构和特性是 (A)

A 模式 B 内模式

C 外模式 D 用户模式

【练习】数据库(DB)、数据库系统(DBS)和数据库管理系统(DBMS)三者之间的关系是(A)

DBS 包括 DB 和 DBMS

DBMS 包括 DB 和 DBS

C. DB 包括 DBS 和 DBMS

D. DBS 就是 DB，也就是 DBMS

2. 数据模型

数据库需要根据应用系统中数据的性质、内在联系，按照管理的要求来设计和组织。数据模型就是从现实世界到机器世界的一个中间层。现实世界的事物反映到人的大脑，人们把这些事物抽象为一种既不依赖于具体的计算机系统又不为某一数据库管理系统支持的概念模型，然后再把概念模型转换为计算机上某一数据库管理系统支持的数据模型。

2.1 数据描述

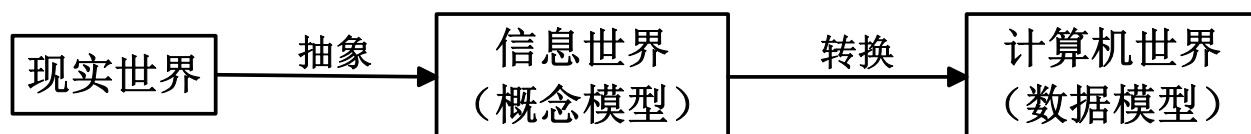
所谓数据描述，就是以数据符号的形式，从满足用户需求出发，对客观事物属性和运动

状态进行描述。

数据的转换

数据的描述既要符合客观现实，又要适应数据库原理与结构，同时也适应计算机原理与结构。进一步说，由于计算机不能够直接处理现实世界中的具体事物，所以人们必须将客观存在的具休事物进行有效的描述与刻画，转换成计算机能够处理的数据，这一转换过程可分为三个数据范畴：现实世界、信息世界和计算机世界。

从客观现实到计算机的描述，数据的转换过程如图所示。



1. 现实世界

用户为了某种需要，需将现实世界中的部分需求用数据库实现，这样，我们所见到的是客观世界中的划定边界的一个部分环境，它称为现实世界。

2. 信息世界

通过抽象对现实世界进行数据库级上的刻画所构成的逻辑模型叫信息世界。信息世界与数据库的具体模型有关，如层次、网状、关系模型等。

3. 计算机世界

在信息世界基础上致力于其在计算机物理机构上的描述，从而形成的物理模型叫计算机世界。现实世界的要求只有在计算机世界中才能得到真正的物理实现，而这种实现是通过信息世界逐步转化得到的。

实体描述

现实世界中存在各种事物，事物之间存在着联系，而且这种联系是客观存在的，是由事物本身的性质所决定的。人们从现实世界抽象各种事物到信息世界（概念模型）时，通常采用实体来描述现实世界中具体的事物或事物之间的联系。

实体

客观存在并可相互区别的事物称为实体。实体可以是具体的人、事、物，也可以是抽象的概念或联系。例如学生、课程、教师都是属于实际存在的事物，而学生选课就比较抽象的事物，是由学生和课程之间的联系而产生的等等。

实体的属性

描述实体的特性称为属性。一个实体可以由若干个属性来刻画，如一个学生实体有学号、

姓名、性别、出生日期等方面的属性。属性有属性名和属性值，属性的具体取值称为属性值。例如，对某一学生的“性别”属性取值“女”，其中“性别”为属性名，“女”为属性值。

实体集和实体型

同类型的实体的集合称为实体集。例如，对于“学生”实体来说，全体学生就是一个实体集。

属性的集合表示一个实体的类型，称为实体型。例如，学生（学号，姓名，性别，出生日期）就是一个实体型。

属性值的集合表示一个实体。例如，属性值的集合（200901001，张三，男，1983-11-23，计算机系）就是代表一个具体的学生。

实体间的的联系及分类

现实世界中事物内部以及事物之间是有联系的，在概念模型中反映为实体内部的联系和实体之间的联系。实体内部的联系通常是指组成实体的各属性之间的联系，而实体之间的联系通常是指不同实体集之间的联系。

实体间的的联系及分类

两个实体之间的联系可分为如下三种类型：

（1）一对一联系（1 : 1）

实体集 A 中的一个实体至多与实体集 B 中的一个实体相对应，反之亦然，则称实体集 A 与实体集 B 之间为一对一的联系，记作 1 : 1。例如，一个学校只有一个校长，一个校长只能管理一个学校。

（2）一对多联系（1 : n）

如果对于实体集 A 中的每一个实体，实体集 B 中有多个实体与之对应，反之，对于实体集 B 中的每一个实体，实体集 A 中至多只有一个实体与之对应，则称实体集 A 与实体集 B 之间为一对多联系，记为 1 : n。例如，学校的一个系有多个专业，而一个专业只属于一个系。

（3）多对多联系（m : n）

如果对于实体集 A 中的每一个实体，实体集 B 中有多个实体与之对应，反之，对于实体集 B 中的每一个实体，实体集 A 中也有多个实体与之对应，则称实体集 A 与实体集 B 之间为多对多联系，记为 m : n。例如，一个学生可以选修多门课程，一门课程可以被多名学生选修。

2.2 数据模型

信息是客观事物在人们头脑中的抽象反映。人们可以从现实世界中获得各种各样的信息，从而了解世界并且相互交流。但是信息的多样化特性使得人们在描述和管理这些数据时往往力不从心，因此人们把表示事物的主要特征抽象地用一种形式化的描述表示出来，模型方法就是这种抽象的一种表示。信息领域中采用的模型通常称为数据模型。

2.2.1 数据模型的概念

1.数据模型是从现实世界到机器世界的一个中间层次。现实世界的事物反映到人的大脑中,人们把这些事物抽象为一种既不依赖于具体的计算机系统又不依赖于具体的 DBMS 的概念模型,然后,再把该概念模型转换为计算机中某个 DBMS 所支持的数据模型。

2.数据模型是实现数据抽象的主要工具。它决定了数据库系统的结构、数据定义语言和数据操纵语言、数据库设计方法、数据库管理系统软件的设计与实现。了解关于数据模型的基本概念是我们学习数据库的基础。

2.2.2 数据模型的组成要素

一般地讲,数据模型是严格定义的概念的集合,这些概念精确地描述系统的静态特性、动态特性和完整性约束条件。因此,数据模型通常由数据结构、数据操作和数据的完整性约束三部分组成。

数据结构

1.数据结构是研究存储在数据库中的对象类型的集合,这些对象类型是数据库的组成部分。数据模型中的数据结构主要描述数据的类型、内容、性质以及数据间的联系等。

2.数据结构是数据模型的基础,数据操作与约束均建立在数据结构上。不同数据结构有不同的操作与约束,因此,一般数据模型的分类均以数据结构的不同而分。

3.数据库系统是按数据结构的类型来组织数据的,因此数据库系统通常按照数据结构的类型来命名数据模型。如层次结构、网状结构和关系结构的模型分别命名为层次模型、网状模型和关系模型。

数据操作

数据操作是指对数据库中各种对象的实例允许执行的操作的集合,包括操作和有关的操作的规则。例如插入、删除、修改、检索、更新等操作,数据模型要定义这些操作的确切含义、操作符号、操作规则以及实现操作的语言等。

数据的完整性约束

1.数据的约束条件是完整性规则的集合,用以限定符合数据模型的数据库状态以及状态的变化,以保证数据的正确、有效和相容。数据模型中的数据及其联系都要遵循完整性规则的制约。

2.另外,数据模型应该提供定义完整性约束条件的机制以反映某一应用所涉及的数据必须遵守的特定的语义约束条件。

2.2.3 数据模型的种类

数据模型按不同的应用层次分成三种模型,

1. 概念数据模型 (Conceptual Data Model)
2. 逻辑数据模型 (Logic Data Model)

3. 物理数据模型（Physical Data Model）

概念数据模型

1.概念数据模型简称概念模型，它是一种面向客观世界、面向用户的模型；它与具体的数据库管理系统无关，与具体的计算机平台无关。

2.概念模型着重于对客观世界复杂事物的结构描述及它们之间的内在联系的刻画。概念模型是整个数据模型的基础。目前，较为有名的概念模型有 E-R 模型、扩充的 E-R 模型、面向对象模型及谓词模型等。

逻辑数据模型

1.逻辑数据模型又称为数据模型，它是一种面向数据库系统的模型，该模型着重于在数据库系统一级的实现。

2.概念模型只有在转换成数据模型后才能在数据库中得以表示。目前，逻辑数据模型也有很多种，较为成熟并先后被人们大量使用过的有：层次模型、网状模型、关系模型、面向对象模型等。

物理数据模型

物理数据模型也简称物理模型，它是一种面向计算机物理表示的模型。物理数据模型给出了数据模型在计算机上物理结构的表示，它是描述数据在储存介质上的组织结构的数据模型。

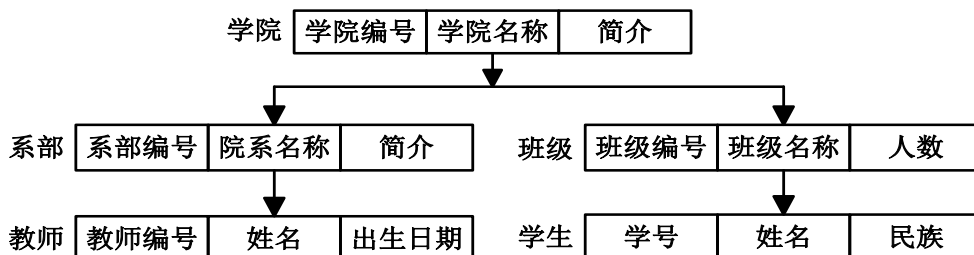
2.2.4 主要的逻辑数据模型

目前，数据库领域中常用的逻辑数据模型有 4 种，它们是：

1. 层次模型（Hierarchical Model）
2. 网状模型（Network Model）
3. 关系模型（Relational Model）
4. 面向对象模型（Object Oriented Model）

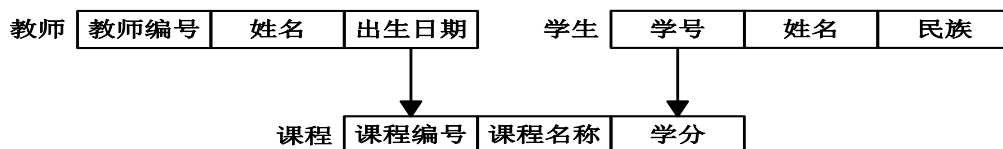
层次模型

层次模型是数据库系统中最早采用的数据模型，它是通过从属关系结构表示数据间的联系，层次模型是有向“树”结构。层次数据库模型的代表是 IBM 公司的 IMS（Information Management System）数据库管理系统。



网状模型

网状模型是层次模型的扩展，它表示多个从属关系的层次结构，呈现一种交叉关系的网络结构，网状模型是有向“图”结构。网状数据模型的典型代表是 DBTG 系统，也称 CODASYL 系统。但它并非实际的数据库管理系统，它所提出的基本概念、方法和技术对于网状数据库系统的发展产生了重大影响。



关系模型

关系数据模型（简称关系模型）以二维表的方式组织数据，如表所示。关系模型建立在严格的数学概念基础之上，自从出现后发展迅速。20 世纪 80 年代以来，几乎所有的数据库系统都是建立在关系模型之上。

学号	姓名	性别	民族	政治面貌	出生日期
200901001	程鑫民	男	锡伯族	团员	1992-1-30
200901002	吴薇	女	壮族	群众	1992-7-9
200901003	李薪	女	汉族	团员	1990-12-3
200901004	张丽	女	白族	团员	1991-2-5
200901005	李斯	男	彝族	团员	1990-8-24

面向对象模型

1.面向对象的概念最初出现在程序设计方法中，由于其优点突出，便于描述复杂的客观现实，因此迅速渗透到计算机领域的众多分支。面向对象模型是面向对象概念与数据库技术相结合的产物，用以支持非传统应用领域对数据模型提出的新要求。

2.面向对象模型最基本的概念是对象（ Object ）和类（ Class ）。在面向对象模型中，

对象是指客观的某一事物，其对对象的描述具有整体性、完整性，对象不仅包含描述它的数据，而且还包含对它进行操作的方法的定义，对象的外部特征与行为是封装在一起的。其中，对象的状态是该对象属性集，对象的行为是在对象状态上操作的方法集。共享同一属性集和方法集的所有对象则构成了类。

【练习】下列数据模型中，数据独立性最高的是（ B ）。

- A.网状数据模型
- B.关系数据模型
- C.层次数据模型
- D.非关系模型

【练习】下列说法中正确的是（ ACD ）。

- A 数据库减少了数据冗余
- B 数据库避免了一切数据的重复
- C 数据库中的数据可以共享
- D 数据库具有较高的数据独立性

【练习】现实世界中事物的特性在信息世界中称为（D）

- A 实体
- B 键
- C 记录
- D 属性

【练习】下列数据模型中，数据独立性最高的是（ B ）。

- A.网状数据模型
- B.关系数据模型
- C.层次数据模型
- D.非关系模型

【练习】下列说法中正确的是（ ACD ）。

- A 数据库减少了数据冗余
- B 数据库避免了一切数据的重复
- C 数据库中的数据可以共享
- D 数据库具有较高的数据独立性

3. 关系数据库

1.关系是数学集合论中的一个重要概念。1970 年，E.F.Codd 发表了题为“大型共享数据库数据的关系模型”的论文，把关系的概念引入了数据库，自此人们开始了数据库关系方法和关系数据理论的研究，在层次和网状数据库系统之后，形成了以关系数据模型为基础的关系数据库系统。

2.关系模型的数学理论基础是建立在集合代数上的，与层次模型、网状模型相比较，是目前广为应用的一种重要的数据模型。关系型数据库的数据组织、管理与检索等，都是基于数学理论的方法来处理数据库中的数据本身和数据之间的联系。

3.1 关系模型

关系模型的用户界面非常简单，一个关系的逻辑结构就是一张二维表。这种用二维表的

形式表示实体和实体间联系的数据模型称为关系数据模型。

关系术语

1. 关系：一个关系就是一张二维表格，每个关系有一个关系名，在 Access 中，一个关系就是一个表对象。
2. 元组：表格中的每一行称为一个元组也，称为记录。
3. 属性：表格中的每一列称为一个属性，给每列起一个名称，该名称就是属性名，如表 1-1 中的学号、姓名、性别、出生日期等。属性也称为字段。
4. 分量：元组中的一个属性值称为分量。关系模型要求关系的每一个分量必须是一个不可分的数据项，即不允许表中还有表。
5. 域：属性的取值范围。从总体上说，以属性分类的若干个元组的集合，构成关系模式中的一个关系，在某种意义上也可以说，关系模式就是一张二维表格，用来描述客观事物以及不同事物间的联系。
6. 候选关键字：关系中的某个属性组（一个属性或几个属性的组合）可以唯一标识一个元组，这个属性组称为候选关键字。
7. 关键字：关键字是指在一个数据表中，若某一字段或几个字段的组合值能够唯一标识一个记录，则称其为关键字（或键），当一个数据表有多个关键字时，可从中选出一个作为主关键字。
8. 外部关键字：如果关系中的一个属性不是本关系的关键字，而是另外一个关系的关键字或候选关键字，这个属性就称为外部关键字。
9. 主属性：包含在任一候选关键字中的属性称为主属性。

关键字		关系（二维表）					
属性 (字段)	→	学号	姓名	性别	民族	政治面貌	出生日期
元组 (记录)	→	200901001	程鑫民	男	锡伯族	团员	1992-1-30
		200901002	吴薇	女	壮族	群众	1992-7-9
		200901003	李薪	女	汉族	团员	1990-12-3
分量 (属性值)		200901004	张丽	女	白族	团员	1991-2-5
		200901005	李斯	男	彝族	团员	1990-8-24

域（取值范围）：男，女

关系的性质

关系是一个二维表，但并不是所有的二维表都是关系。关系应具有以下性质：

1. 每一列中的分量（属性值）是同一类型的数据，来自同一个域。
2. 不同的列要给予不同的属性名。
3. 列的顺序无所谓，即列的次序可以任意交换。
4. 任意两个元组不能完全相同。
5. 行的顺序无所谓，即行的次序可以任意交换。
6. 每一个分量都必须是不可再分的数据项。

由上述可知，二维表中的每一行都是唯一的，而且所有行都具有相同类型的字段。关系模型的最大优点是一个关系就是一个二维表格，因此易于对数据进行查询等操作。

关系之间的联系

表和表之间的关联存在以下 4 种类型：

1. 一对一联系：父表中每一个记录最多与子表中的一个记录相关联，反之也一样。具有一对一关联的两张表通常在创建表时可以将其合并成为一张表。
2. 一对多联系：父表中每一个记录可以与子表中的多个记录相关联，而子表中的每一条记录都只能与父表中的一条记录相关联。一对多关联是数据库中最为普遍的关联。如，院系和学生的关系。

3. 多对一联系：父表中多个记录可以与子表中的一条记录相关联。
4. 多对多联系：父表中的每一条记录都与子表中的多条记录相关联，而子表中的每一条记录又都与父表中的多条记录相关联。多对多关联在数据库中比较难实现，通常将多对多关联分解为多个一对多关联。如学生和课程的关系。

关系数据库

1. 在关系模型中，实体以及实体之间的联系都是用关系来表示的。例如教师实体、学生实体、课程实体等。在一个给定的应用领域中，所有实体以及实体间的联系的关系的集合就构成一个关系数据库。

2. 关系数据库系统是支持关系模型的数据库系统。它是由若干张二维表组成的，包括二维表的结构以及二维表中的数据两部分。Access 就是一个关系型的数据库管理系统，由 Access 所创建的二维表称为数据表。

3.2 关系的完整性

关系模型允许定义三种完整性约束，即实体完整性、参照完整性和用户定义完整性约束。

其中实体完整性约束和参照完整性约束统称为关系完整性约束，是关系模型必须满足的完整性的约束条件，它由关系数据库系统自动支持。用户定义完整性约束是应用领域需要遵循的约束条件。

实体完整性约束

由于每个关系的主键是惟一决定元组的，故实体完整性约束要求关系的主键不能为空值，组成主键的所有属性都不能取空值。

例如，在“学生”关系：学生（学号，姓名，性别，出生日期），其中学号是主键，因此，学号不能为空值。

例如，在“成绩”关系：成绩（学号，课程号，分数），其中**学号**和**课程号**共同构成主键，因此，学号和课程号都不能为空值。

参照完整性约束

1. 参照完整性约束是关系之间相关联的基本约束，它不允许关系引用不存在的元组，即在关系中的外键取值只能是关联关系中的某个主键值或者为空值。

2. 例如：院系编号是“院系（院系编号，名称，简介）”关系的主键、是“学生（学号，姓名，院系编号）”关系的外键。“学生”关系中的“院系编号”必须是“院系”关系中一个存在的“院系编号”的值，或者是空值。

用户定义的完整性约束

1. 实体完整性约束和参照完整性约束是关系数据模型必须要满足的，而用户定义的完整性约束是与应用密切相关的数据完整性的约束，不是关系数据模型本身所要求的。
2. 用户定义的完整性约束是针对具体数据环境与应用环境由用户具体设置的约束，它反映了具体应用中数据的语义要求，它的作用就是要保证数据库中数据的正确性。

例如，限定某属性的取值范围，学生成绩的取值必须是 0 到 100 之间的数值。

3.3 关系规范化

1.在关系数据库中，如果关系模式没有设计好，就会出现数据冗余、数据更新异常、数据删除异常、数据插入异常等问题。关系模式的优良程度直接影响数据库中的数据完整性等方面的性能。

2.关系规范化就是将结构复杂的关系模式分解成结构简单的关系模式，从而使一个关系模式描述一个实体或实体间的一种联系，以达到概念的单一化。关系规范化目的就是要将不好的关系模式转变为好的关系模式。

3. 把关系数据库的规范化过程中为不同程度的规范化要求设立的不同标准称为范式。

4. 由于规范化的程度不同，就产生了不同的范式，如第一范式、第二范式、第三范式、BCNF 范式、第四范式、第五范式等。每种范式都规定了一些限制约束条件。

5.第一范式（1NF）是最基本的规范形式，它要求关系中的每个属性都必须是不可再分割的数据项。在关系数据库中，任何一个关系模式都必须满足第一范式。

【练习】实体完整性要求主属性不能取空值，这通常是通过（ C ）。

- A. 定义外部键来保证
- B. 用户定义的完整性来保证
- C. 定义主键来保证
- D. 关系系统自动保证

【练习】公司中有多个部门和多名职员，每个职员只能属于一个部门，一个部门可以有多个职员，从职员到部门的联系类型是（ B ）。

- A.多对多 B.多对一
- C.一对一 D.一对多

【练习】学生表内容如下:学生(学号, 姓名, 性别, 专业, 出生日期), 它的主键是（ C ）。

- A.姓名 B.(姓名, 性别)
- C.学号 D.(姓名, 专业学号)

3.4 关系代数

关系代数的运算对象是关系，运算结果也是关系。关系代数的基本运算符有：

1. 集合运算符
2. 关系运算符。
3. 传统的集合运算

传统的集合运算都是二目运算。设关系 R 和关系 S 具有相同的属性个数（关系的属性个

数称之为关系的目)，且相应的属性取自同一个域。

并（Union）运算

设关系 R 和关系 S 具有相同的目 n （即两个关系都有 n 个属性），且相应的属性取自同一个域，则关系 R 与关系 S 的并由属于 R 或属于 S 的元组组成。其结果关系仍为 n 目关系。运算结果是将两个关系的所有元组组成一个新的关系，若有完全相同的元组，只留一个。

记作：

$$R \cup S = \{t | t \in R \vee t \in S\}$$

其中 t 代表元组。

关系 R

A	B	C
a1	b1	c1
a1	b2	c2
a2	b2	c1

关系 S

A	B	C
a1	b2	c2
a1	b3	c2
a2	b2	c1

$R \cup S$

A	B	C
a1	b1	c1
a1	b2	c2
a2	b2	c1
a1	b3	c2

差（Difference）运算

设关系 R 和关系 S 具有相同的目 n ，且相应的属性取自同一个域，则关系 R 与关系 S 的差由属于 R 而不属于 S 的所有元组组成。其结果关系仍为 n 目关系。

记作：

$$R - S = \{t | t \in R \wedge t \notin S\}$$

关系 R

关系 S

A	B	C
a1	b1	c1
a1	b2	c2
a2	b2	c1

A	B	C
a1	b2	c2
a1	b3	c2
a2	b2	c1

R-S

A	B	C
a1	b1	c1

交 (Intersection) 运算

设关系 R 和关系 S 具有相同的目 n，且相应的属性取自同一个域，则关系 R 与关系 S 的交由既属于 R 又属于 S 的元组组成。其结果关系仍为 n 目关系。

记作：

$$R \cap S = \{t | t \in R \wedge t \in S\}$$

关系 R

A	B	C
a1	b1	c1
a1	b2	c2
a2	b2	c1

关系 S

A	B	C
a1	b2	c2
a1	b3	c2
a2	b2	c1

$R \cap S$

A	B	C
a1	b2	c2
a2	b2	c1

广义笛卡尔积 (Extended Cartesian product)

关系 R 为 n 目，关系 S 为 m 目，则关系 R 和关系 S 的广义笛卡尔积为 (n+m) 目的元组的集合。元组的前 n 个属性是关系 R 的属性，后 m 个属性是关系 S 的属性。若 R 有 r1

个元组，S 有 s1 个元组，则关系 R 和关系 S 的广义笛卡尔积有 $r1 \times s1$ 个元组。

记作：

$$R \times S = \{trts \mid tr \in R \wedge ts \in S\}$$

广义笛卡尔积

(Extended Cartesian product)

关系 R

A	B	C
a1	b1	c1
a1	b2	c2
a2	b2	c1

关系 S

A	B	C
a1	b2	c2
a1	b3	c2
a2	b2	c1

$R \times S$

R. A	R. B	R. C	S. A	S. B	S. C
a1	b1	c1	a1	b2	c2
a1	b1	c1	a1	b3	c2
a1	b1	c1	a2	b2	c1
a1	b2	c2	a1	b2	c2
a1	b2	c2	a1	b3	c2
a1	b2	c2	a2	b2	c1
a2	b2	c1	a1	b2	c2
a2	b2	c1	a1	b3	c2
a2	b2	c1	a2	b2	c1

专门的关系运算

专门的关系运算包含选择、投影、连接和除运算。这类运算将“关系”看做是元组的集合，其运算不仅涉及到关系的水平方向（表中的行），而且也涉及到关系的垂直方向（表中的列）。

选择（Selection）运算（选记录）

1.选择是根据给定的条件选择关系 R 中的若干元组组成新的关系，是对关系的元组进行筛选。记作： $F(R)$ 。

2.其中 F 是选择条件，是一个逻辑表达式，它由逻辑运算符和比较运算符组成。

3.选择运算也是一元关系运算，选择运算结果往往比原有关系元组个数少，它是原关系的一个子集，但关系模式不变。

例：

关系 R

A	B	C
a1	b1	c1
a1	b2	c2
a2	b2	c1

若 F 为 $A=a1$, $\sigma_{A=a1}(R)$

A	B	C
a1	b1	c1
a1	b2	c2

例：

关系 S

A	B	C
a1	b2	c2
a2	b2	c1

若 F 为 $B=b2$, $\sigma_{B=b2}(S)$

A	B	C
a1	b2	c2
a1	b3	c2
a2	b2	c1

投影（Projection）运算（投影列）

1. 关系 R 上的投影是从关系 R 中选择若干属性列组成新的关系，记作： $\Pi A(R)$ ，A 是 R 中的属性列。它是从列的角度进行操作的。

2. 设 R 是一个 n 目关系， A_{i1} 、 A_{i2} 、 A_{i3} 、 \dots 、 A_{im} 是 R 的第 $i1$ 、 $i2$ 、 $i3$ 、 \dots 、 im ($m \leq n$) 个属性，关系 R 在 A_{i1} 、 A_{i2} 、 A_{i3} 、 \dots 、 A_{im} 上的投影定义为：

$$\Pi_{i1, i2, i3, \dots, im}(R) = \{t | t = (t_{i1}, t_{i2}, t_{i3}, \dots, t_{im}) \wedge (t_{i1}, t_{i2}, t_{i3}, \dots, t_{im}) \in R\}$$

3. 即从关系 R 中按照 $i1$ 、 $i2$ 、 $i3$ 、 \dots 、 im 的顺序取下这 m 列，构成以 $i1$ 、 $i2$ 、 $i3$ 、 \dots 、 im 为顺序的 m 目关系。投影后不仅取消了原关系中的某些列，而且还可能取消某些元组，因为取消某些列后，可能出现重复的元组，应消去这些完全相同的元组。

关系 R

A	B	C
a1	b1	c1
a1	b2	c2
a2	b2	c1

$\Pi_{AB}(R)$

A	B
a1	b1
a1	b2
a2	b2

例

关系 S

A	B	C
a1	b2	c2
a1	b3	c2
a2	b2	c1

$\Pi_{BC}(S)$

B	C
b2	c2
b3	c2
b2	c1

连接 (Join) 运算 (连表格)

1. 连接是从两个关系 R 和 S 的笛卡尔积中选取属性间满足一定条件的元组，连接也称为连接，记作：

$$R \bowtie S = \sigma_{R.A \theta S.B}(R \times S)$$

其中 A 和 B 分别为 R 和 S 上可比的属性组。 θ 是比较运算符， θ 可以是 $>$, \geq , $<$, \leq , $=$, \neq 等符号。连接运算从 R 和 S 的笛卡尔积 $R \times S$ 中选取 (R 关系) 在 A 属性组上的值与 (S 关系) 在 B 属性组上值满足比较关系 θ 的元组，这些元组构成的关系是 $R \times S$ 的一个子集。 θ 为 “=” 的连接运算称为等值连接。它是从关系 R 与 S 的笛卡尔积中选取 A、B 属性值相等的那些元组。

根据条件，R 中第一行与 S 中前两行连接，以此类推

关系 R

A	B	C
a1	b1	a1
a2	b2	a2
a1	b2	a3

关系 S

B	C	D
b1	a1	d3
b2	a1	d2
b3	a2	d2
b2	a3	d1
b2	a4	d1
b1	a2	d2

$$R \bowtie S = \sigma_{R.C=S.C}(R \times S)$$

R. A	R. B	R. C	S. B	S. C	S. D
a1	b1	a1	b1	a1	d3
a1	b1	a1	b2	a1	d2
a2	b2	a2	b3	a2	d2
a2	b2	a2	b1	a2	d2
a1	b2	a3	b2	a3	d1

自然连接 (Natural Join) 运算

1.自然连接是最常用的连接之一，它是指从两个关系的笛卡尔积中选出公共属性值相等的元组所构成的新的关系。

2.定义：设关系 R 和关系 S 具有相同的属性集 U， $U=\{A_1, A_2, \dots, A_k\}$

3.从关系 R 和关系 S 的笛卡尔积中，取满足 $\Pi R.U = \Pi S.U$ 的所有元组，且去掉 $S.A_1, S.A_2, \dots, S.A_k$ ，(去重复)所得的新关系

$$R \bowtie S = \Pi_{i_1, i_2, i_3, \dots, i_k}$$

$$(R.A_1=S.A_1 \wedge R.A_2=S.A_2 \wedge \dots \wedge R.A_k=S.A_k(R \times S))$$

记为关系 R 和关系 S 的自然连接。

自然联接是按照公共属性值相等的条件进行联接，并且消除重复属性。

自然连接 (Natural Join) 运算

关系 R

关系 S

A	B	C
a1	b1	a1
a2	b2	a2
a1	b2	a3

B	C	D
b1	a1	d3
b2	a1	d2
b3	a2	d2
b2	a3	d1
b2	a4	d1
b1	a2	d2

自然连接 $R \bowtie S$ (R 中 B、C 列与 S 中 B、C 列相等的记录)

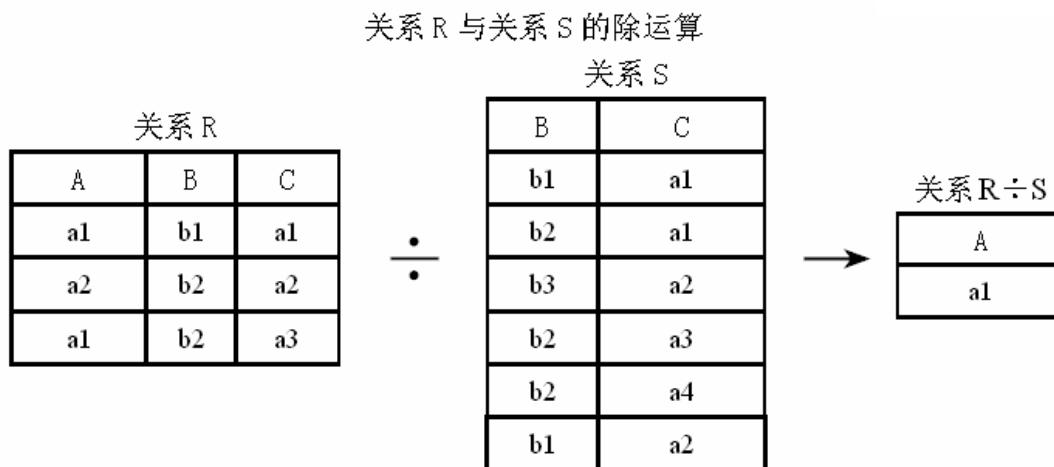
A	B	C	D
a1	b1	a1	d3
a1	b2	a3	d1

除运算 (除去列)

关系 R 与关系 S 的除法运算应满足的条件是：关系 S 的属性全部包含在关系 R 中，关系 R 的一些属性不包含在关系 S 中。

关系 R 与关系 S 的除法运算表示为 $R \div S$ 。除法运算的结果也是关系，而且该关系中的属性由 R 中除去 S 中的属性之外的全部属性组成，元组由 R 与 S 中在所有相同属性上有相等值的那些元组组成。

把 S 包含的列从中 R 除去，并去掉重复记录



【练习】对关系 R 进行投影运算后，得到关系 S，则(C)

- A. 关系 R 的元组数等于关系 S 的元组数
- B. 关系 R 的元组数小于关系 S 的元组数
- C. 关系 R 的元组数大于或等于关系 S 的元组数
- D. 关系 R 的元组数大于关系 S 的元组数

【练习】关系笛卡尔积运算记号 $R \times S$ 中(D)

- A. R 为关系名, S 为属性名
- B. R, S 均为属性名
- C. R 为属性名, S 为关系名
- D. R, S 均为关系名

4. 数据库设计

1.数据库设计是指对于一个给定的应用环境,设计数据库的逻辑模式和物理结构,并据此建立数据库及其应用系统,使之能够有效地存储和管理数据,满足各种用户的应用需求,包括信息管理需求和数据操作需求。

2.大型数据库的设计和应用系统开发是一项庞大的工程,是涉及多学科的综合性技术。数据库建设和一般的软件系统的设计、开发和运行维护有许多相同之处,也有其自身特点。

1 数据库设计概述

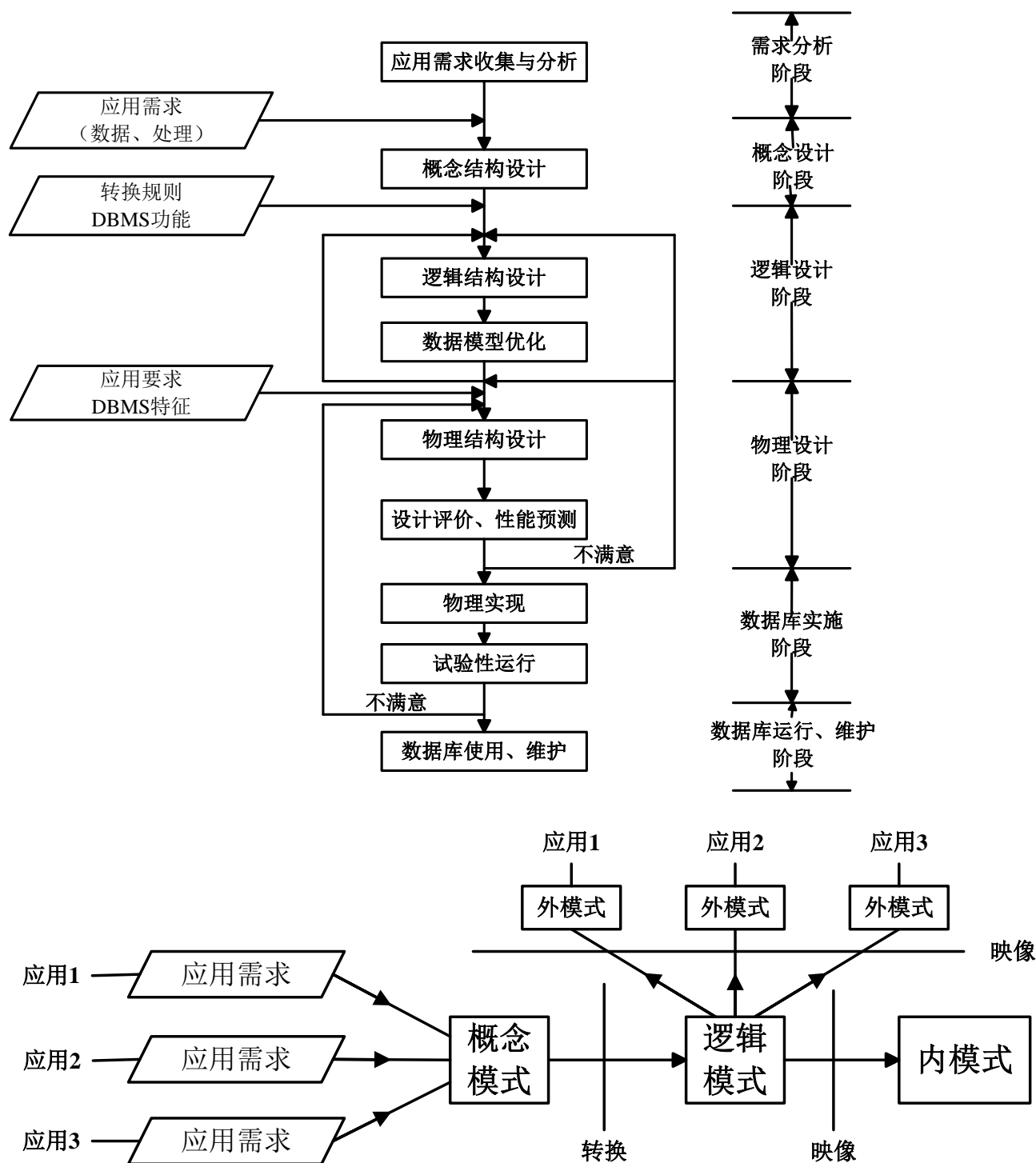
数据库的建设不仅涉及技术,更涉及管理,业界有“三分技术,七分管理”之说,由此可见管理在数据库设计过程中的作用,这是数据库设计的特点之一。

数据库设计应该与应用系统设计相结合,也就是说,整个设计过程要把数据库结构设计和对数据的处理设计紧密的结合起来,这是数据库设计的重要特点。

通过分析、比较与综合各种常用的数据库规范设计方法,将数据库设计分为以下 6 个阶段。

- (1) 需求分析阶段
- (2) 概念结构设计阶段
- (3) 逻辑结构设计阶段
- (4) 物理结构设计阶段
- (5) 数据库实施阶段
- (6) 数据库运行与维护阶段

数据库设计过程



数据库设计过程中的各级模式

1. 需求分析阶段：综合各个用户的应用需求，生成需求说明书。
2. 概念设计阶段：形成独立于机器特点，独立于具体数据库管理系统产品的概念模式(E-R图)。
3. 逻辑设计阶段：首先将 E-R 图转换成具体的数据库管理系统所支持的数据模型，如关系模型，形成数据库逻辑模式；然后根据用户处理的要求、安全性的考虑，在基本表的基础上

再建立必要的视图（View），形成数据的外模式。

4. 物理设计阶段：根据数据库管理系统的特点和处理的需要，进行物理存储安排，建立索引，形成数据库内模式。

2 概念模型与 E-R 方法

数据库概念设计的目的是分析数据间内在的语义关联，在此基础上建立一个数据的抽象模型—概念数据模型。概念数据模型是根据用户需求设计出来的，它不依赖于任何的数据库管理系统。

概念数据模型设计的描述最常用的工具是 E-R 图（实体-联系图）。

概念模型

前面提到，为了把现实世界中的具体事物抽象、组织为某一数据库管理系统支持的数据模型，人们常常首先将现实世界抽象为信息世界，然后将信息世界转换为机器世界。也就是说，首先把现实世界中的客观对象抽象为某一种信息结构，这种信息结构并不依赖于具体的计算机系统，不是某一个数据库管理系统支持的数据模型，而是概念级的模型，称为概念模型。

采用概念数据模型，数据库设计人员可以在设计的开始阶段，把主要精力用于了解和描述现实世界上，而把涉及具体的数据库管理系统的一些技术性的问题推迟到设计阶段去考虑。

实体-联系方法

设计概念模型常用的方法是实体-联系方法，也就是说，描述概念模型的工具是实体-联系模型（E-R 模型）。因此，数据库概念结构的设计就是 E-R 模型的设计。

设计 E-R 模型的步骤如下：

- （1）设计局部 E-R 模型，用来描述用户视图。
- （2）综合各局部 E-R 模型，形成总的 E-R 模型，用来描述数据库全局视图，即用户视图的集成。

实体-联系方法

概念模型是对整个数据库组织的逻辑结构的抽象定义，E-R 模型是用 E-R 图来描述的，即通过 E-R 图来描述实体集、实体属性和实体集之间联系。

其中：

- （1）“矩形”框用于表示实体集；
- （2）“椭圆形”框用于表示实体集中实体的属性；
- （3）“菱形”框用于表示实体集之间的联系。

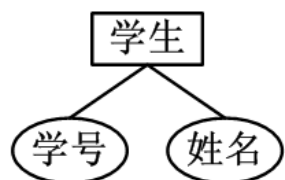
在 E-R 图中，用无向线段或有向线段将实体与其属性、实体与实体之间的联系以及联系与其属性连接起来。

例如：“教学管理”系统中有“学生”和“课程”实体以及相关属性，同时两个实体之

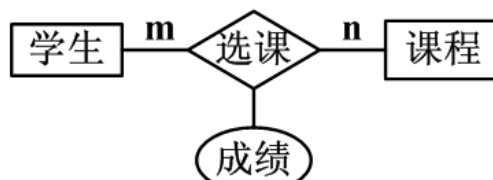
间具有“选课”这样的联系，用 E-R 图可表示为如图所示。



实体与属性之间、联系与属性之间以及实体与实体之间的联系用无向线段连接，线段上所标识的符号数字用以标识实体之间的关系。如图 1-12 所示。



实体与属性之间的联系



实体与实体之间的联系

3 关系数据库设计理论

数据库设计的问题可以简单地描述为：如果要把一组数据存储到数据库中，如何为这些数据设计一个合理的逻辑结构。在关系数据库系统中，就是如何设计一些关系表以及这些关系表中的属性。

关系的规范化

关系模型是建立在严格的数学关系理论基础之上的，通过确立关系中的规范化准则，既可以方便数据库中数据的处理，又可以给程序设计带来方便。在关系数据库设计过程中，是关系满足规范化准则的过程称之为关系规范化（Relation Normalization）。

关系规范化就是将数据库中不太合理的关系模型转化为一个最佳的数据模型，因此它要求对于关系数据库中的每一个关系都要满足一定的规范，根据满足规范的条件不同，可以划分为 6 个范式（Normal Form，简称 NF），分别为：第一范式（1NF）、第二范式（2NF）、第三范式（3NF）、BCNF、第四范式（4NF）、第五范式（5NF）。

模式分解

规范化的基本方法就是模式分解

对关系模式进行分解，要符合“无损连接”和“保持依赖”的原则，使分解后的关系不能破坏原来的函数依赖，保证分解后的所有关系模式中的函数依赖要反映分解前所有的函数依赖。

（1）无损连接：当对关系模式 R 进行分解时，R 元组将分别在相应属性集进行投影而产生新的关系。如果对新关系进行自然连接得到的元组的集合与原关系完全一致，则称无损连接。

（2）保持依赖：当对关系模式 R 进行分解时，R 的函数依赖集也将按相应的模式进行分解，如果分解后的总的函数依赖集与原函数依赖集保持不变，则称为保持函数依赖。

需要特别指出的是，保留适量冗余，达到以空间换时间的目的，也是模式分解的重要原则。

E-R 模型向关系模型的转换

E-R 模型转换成关系模型，就是将实体型和实体型间的联系转换为关系模式，确定关系模式的属性和码，转换过程中要做到不违背关系的完整性约束，尽量满足规范化原则。

【练习】设某学校中，一位教师可讲授多么课程，一门课程可由多位教师讲授，则教师与课程之间的联系是（D）

A 一对一联系 B 一对多联系

C 多对一联系 D 多对多联系

【练习】E-R 图一般用于描述（B）阶段的工作成果

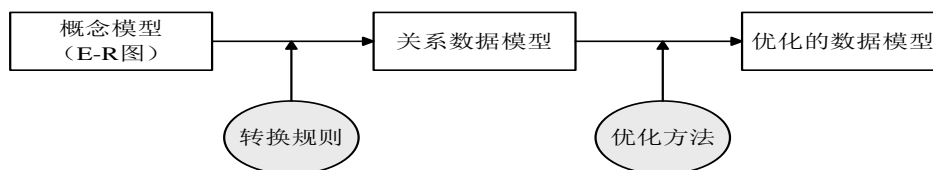
A 需求分析 B 概念结构设计

C 逻辑结构设计 D 物理结构设计

4.3 逻辑结构设计

概念模型是独立于数据库管理系统的概念结构，而逻辑结构设计是根据已设计好的概念模型（E-R 图），将其转换为与某一数据库管理系统支持的数据模型相符的逻辑结构的过程。

逻辑结构设计一般分两步进行，如图所示。



（1）将概念结构转换为关系模型；

（2）对所得到的数据模型进行模型优化。

将 E-R 图转换为关系模型

E-R 图向关系模型的转换要解决的问题是如何将实体和实体型之间联系转换为关系模式，以及如何确定这些关系模式的属性和码。

根据前节所述转换规则，可将教学管理系统的 E-R 图转换为如下关系模式：

- 院系（院系编号、院系名称、院长姓名、院办电话、院系网址）
- 教师（编号、姓名、性别、出生日期、学历、职称、所属院系、办公电话、手机、是否在职、电子邮件）
- 学生（学号、姓名、性别、民族、政治面貌、出生日期、所属院系简历、照片）

- 课程（课程编号、课程名称、课程类别、学时、学分、课程简介）
- 成绩（学号、课程编号、分数）
- 授课（教师编号、课程编号、学期、授课时间、授课地点）

关系视图设计

逻辑设计的另一个重要内容是关系视图的设计，它又称为外模式设计。关系视图是在关系模式基础上所设计的直接面向操作用户的视图，它可以根据用户需求随时创建，一般关系型数据库管理系统都提供关系视图的功能。

关系视图的作用大致有如下几点：

（1）提供数据逻辑独立性：使应用程序不受逻辑模式变化的影响。数据的逻辑模式会随着应用的发展而不断变化，逻辑模式的变化必然会影响到应用程序的变化，这就会产生极为麻烦的维护工作。关系视图则起了逻辑模式与应用程序之间的隔离墙作用，有了关系视图后建立在其上的应用程序就不会随逻辑模式修改而产生变化，此时变动的仅是关系视图的定义。

（2）能适应用户对数据的不同需求：每个数据库有一个非常庞大的结构，而每个数据库用户则希望只知道他们自己所关心的那部分结构，不必知道数据的全局结构以减轻用户在此方面的负担。此时，可用关系视图屏蔽用户所不需要的模式，而仅将用户感兴趣的部分呈现出来。

（3）有一定数据保密功能：关系视图为每个用户划定了访问数据的范围，从而在应用的各用户间起了一定的保密隔离作用。

4.4 物理结构设计

数据库的物理设计是设计数据库的存储结构和物理实现方法。数据库的物理设计主要目标是对数据库内部物理结构作调整并选择合理的存取路径，以提高数据库访问速度以及有效利用存储空间。

目前，在关系数据库中已大量屏蔽了内部物理结构，因此留给用户参与物理设计的任务很少，一般的关系数据库管理系统留给用户参与物理设计的内容大致有索引设计、分区设计等。

数据库物理结构设计主要分为两个方面：

1. 确定数据库的物理结构

在进行设计数据库的物理结构时，要面向特定的数据库管理系统，要了解数据库管理系统的功能，熟悉存储设备的性能。

2. 对物理结构进行评价

在物理结构设计过程中，需要对时间效率、空间效率、维护代价和用户要求进行权衡，设计方案可能有多种，数据库设计人员就要对这些方案进行评价。

4.5 数据库的实施

数据库物理结构设计完成后，就可以进入数据库的具体实施了。

- 数据库实施过程一般步骤如下：
- 定义数据库结构:用某一具体的数据库管理系统提供的数据库定义语言严格描述数据库结构。
- 组织数据入数据库:数据库结构建立完成后，便可以将原始数据载入到数据库中。
- 编写和调试应用程序:编写程序时可用一些模拟数据进行程序调试，待程序编写完成方可正式输入数据。
- 数据库试运行:在这个阶段，最好常对数据库中的数据进行备份的操作，因为，调试期间系统不稳定，容易破坏已存在的数据信息。

4.6 数据库的运行和维护

数据库在实施阶段要反复试运行，当数据库试运行结果符合设计目标后，数据库就可以真正投入运行了，这时候数据库应用系统处于一个相对稳定的状态。投入运行并不意味着数据库设计工作全部完成。设计好的数据库在使用中需要不断维护、修改和调整。

对数据库的维护，通常是由数据库管理员（ Database Administrator ， DBA ）完成的。

- 数据库维护的主要工作内容有：
- 数据库转储和恢复；
- 数据库安全性和完整性控制；
- 数据库性能的监督、分析和改进；
- 数据库的重新组织和重新建构。

【练习】假如采用关系数据库系统来实现应用，在数据库设计的（B）阶段，需将 E-R 模型转换为关系数据模型。

- A 概念设计 B 逻辑设计
- C 物理设计 D 运行阶段

【练习】数据库的（B）是指数据的正确性和相容性

- A 安全性 B 完整性
- C 并发控制 D 恢复

4、 数据库设计实例

（一） “教学管理” 系统的需求分析

教学管理是学校各项管理工作的核心,实现教学管理的计算机化,可以简化繁琐的工作模式,提高教学管理的工作效率、工作质量和管理水平。

（1）背景描述

某大学根据自身管理需要,提出开发“教学管理”系统的要求。教学管理人员的主要工作内容包括教师信息管理、学生信息管理、课程信息管理、教师授课管理以及学生选课成绩管理等几项。

（2）系统分析

根据深入调查了解该学校教学管理运行情况,下面是初步归纳给出的教学管理系统有关功能要求和数据存储要求。

①系统功能要求

方便地录入和修改教师信息、课程信息、学生信息和院系信息。

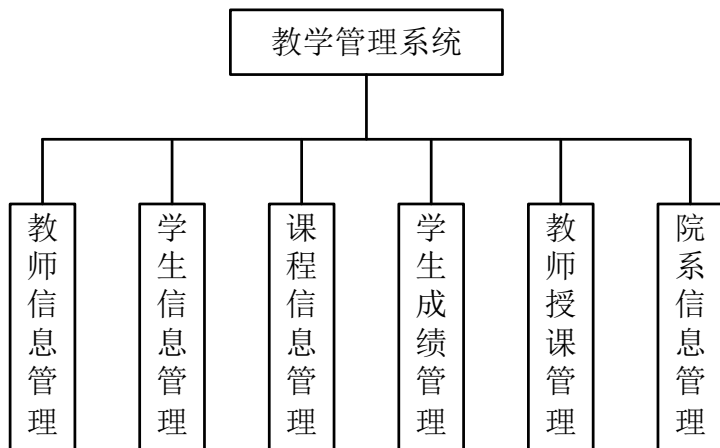
根据教师授课情况,方便地录入和修改教师授课信息。

根据学生选课情况,方便地录入和修改学生成绩信息。

简单快捷地查找相关数据信息。

灵活快捷地统计相关数据信息。

②系统功能模块



③数据信息存储要求

1. 院系信息：包括院系编号、院系名称、院长姓名、院办电话、院系网址等信息。
2. 学生信息：包括学号、姓名、性别、民族、政治面貌、出生日期、所属院系简历、照片等信息。
3. 教师信息：包括编号、姓名、性别、出生日期、学历、职称、所属院系、办公电话、手机号码、是否在职、电子邮件等信息。
4. 课程信息：包括课程编号、课程名称、课程类别、学时、学分、课程简介等信息。

5. 教师授课：包括教师编号、课程编号、学期、授课时间、授课地点等信息。
6. 学生选课：包括学号、课程编号、成绩等信息。

（二）概念结构设计

数据库概念设计的目的是分析数据间内在的语义关联，在此基础上建立一个数据的抽象模型——概念数据模型。概念数据模型是根据用户需求设计出来的，它不依赖于任何的数据库管理系统（DBMS）。

1) 概念结构设计的方法和步骤：

设计概念结构通常有自顶向下、自底向上、逐步扩张、混合策略等方法。其中最常用的策略是自底向上方法。即自顶向下地进行需求分析，然后再自底向上地设计概念结构。

自底向上设计概念结构的方法通常分为两步：首先抽象数据并设计局部视图，然后集成局部视图得到全局的概念结构。

概念结构设计描述最常用的工具是 E-R 图，具体的设计步骤如下：

- （1）确定实体；
- （2）确定实体的属性；
- （3）确定实体的主键；
- （4）确定实体间的联系类型；
- （5）画出 E-R 图。

2) 教学管理系统的 E-R 图

根据教学管理系统的需求分析，该系统存在 4 个实体（院系、学生、教师、课程）。

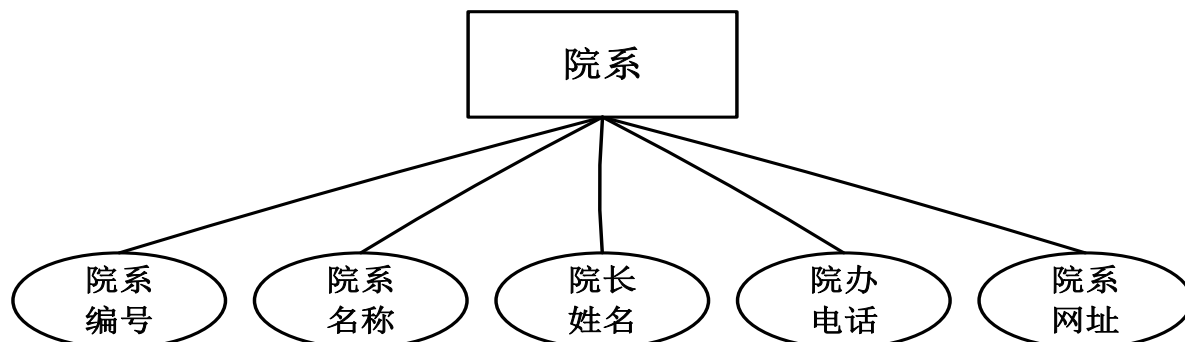
院系：{院系编号、院系名称、院长姓名、院办电话、院系网址}

教师：{编号、姓名、性别、出生日期、学历、职称、所属院系、办公电话、手机、是否在职、电子邮件}

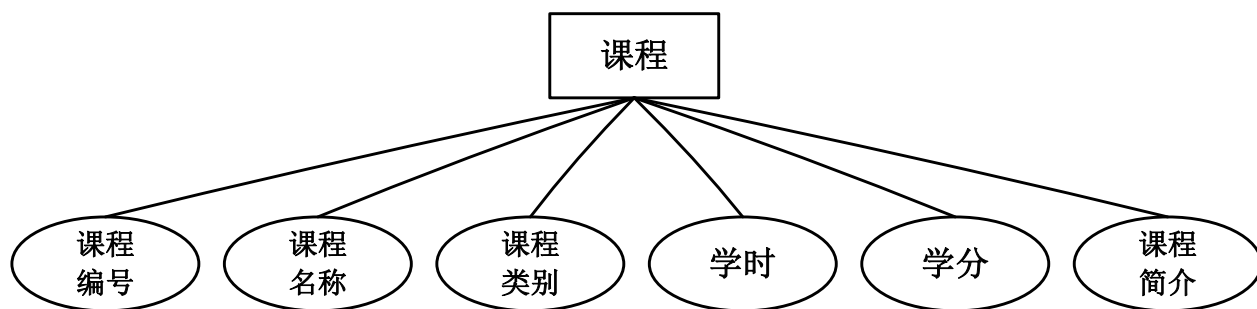
学生：{学号、姓名、性别、民族、政治面貌、出生日期、所属院系简历、照片}

课程：{课程编号、课程名称、课程类别、学时、学分、课程简介}

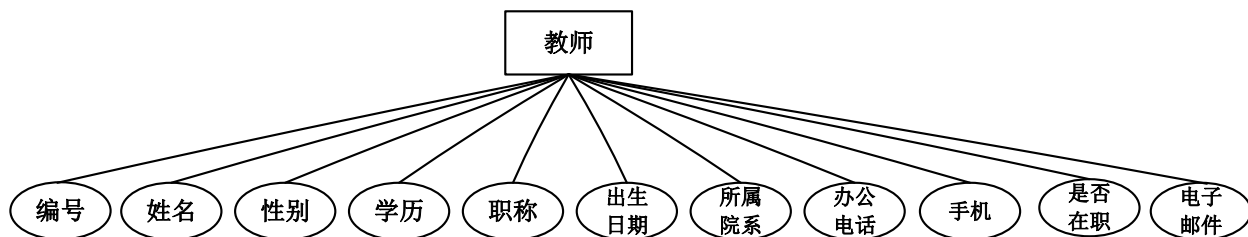
院系实体及其属性 E-R 图：



课程实体及其属性 E-R 图：

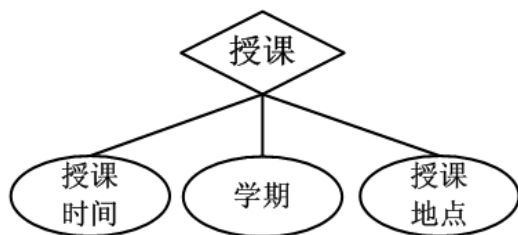


教师实体及其属性 E-R 图



联系及其属性 E-R 图

其中两个联系“授课”和“选课”具有自身属性，如图所示。



“授课”联系及其属性



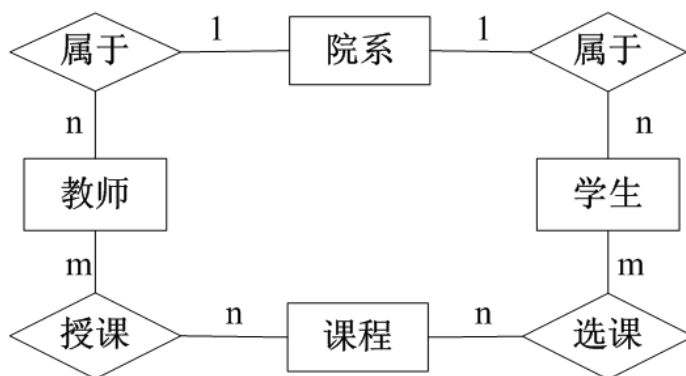
“选课”联系及其属性

联系及其属性图

全局 E-R 图

集成所有实体及其相互联系，得到系统的全局 E-R 图，如图所示。

集成所有实体及其相互联系，得到系统的全局 E-R 图，如图所示。



全局 E-R 图

【练习】在概念模型中，一个实体相对于关系数据库中一个关系中的一个(B)

- A. 属性
- B. 元组
- C. 列
- D. 字段

【练习】下列不是 ER 图中的基本成分的是(D)

- A. 菱形框
- B. 椭圆形框
- C. 直线
- D. 多边形

5. SQL Server 数据库管理

SQL Server 数据库系统的特点

1. 与 Internet 的紧密结合
2. 可扩展性和可用性
3. 企业级数据库
4. 简单、友好的操作方式
5. 数据仓库支持

5.1 SQL Server 的构成

SQL Server 包括：

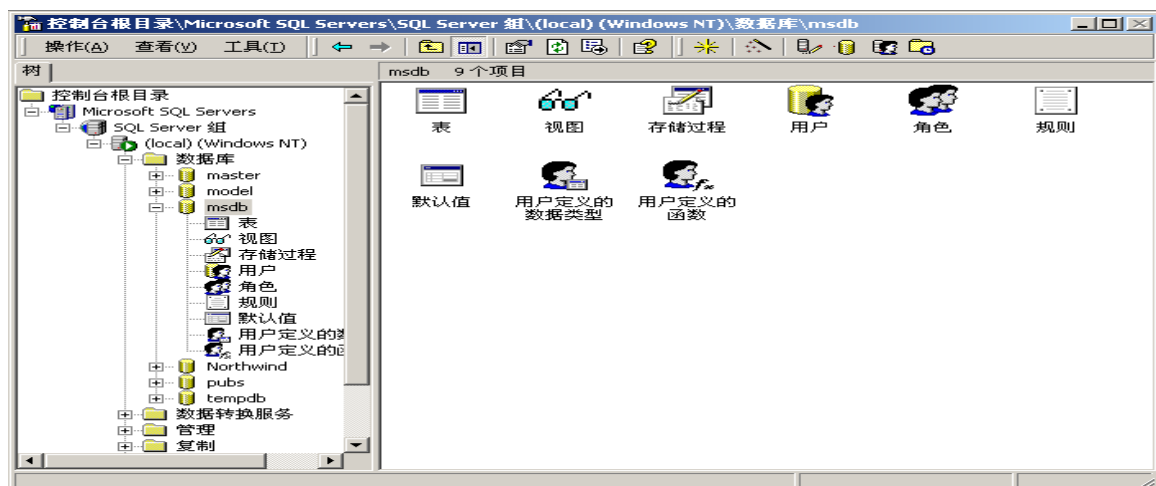
- 服务管理器

- 查询分析器
- 企业管理器
- 事件探查器

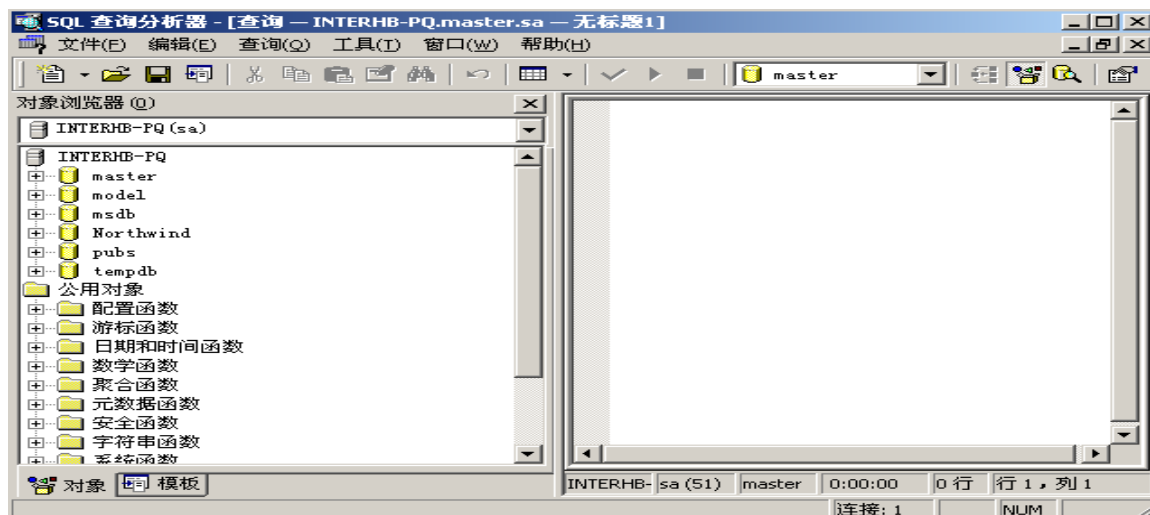
1、服务管理器



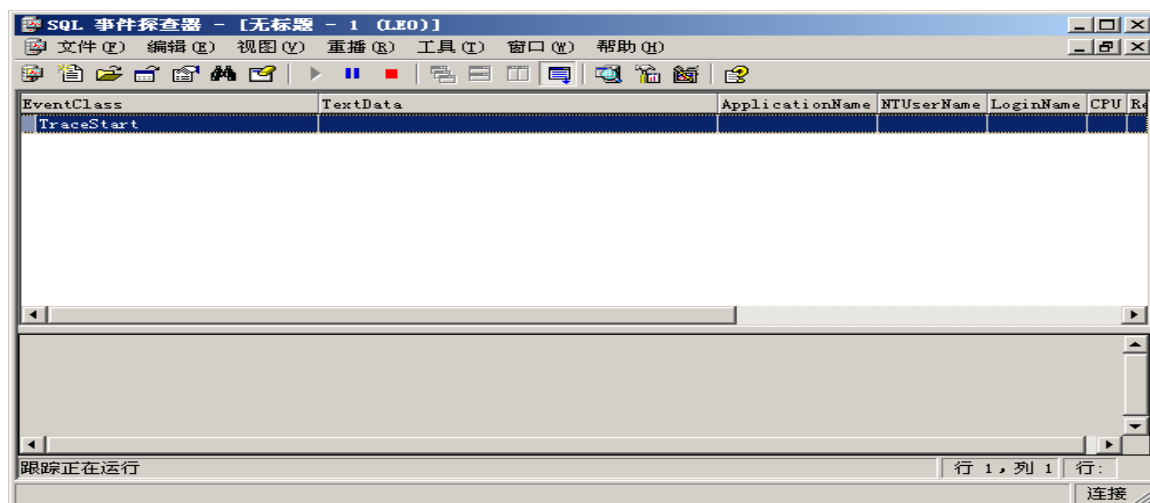
2、企业管理器



3、查询分析器

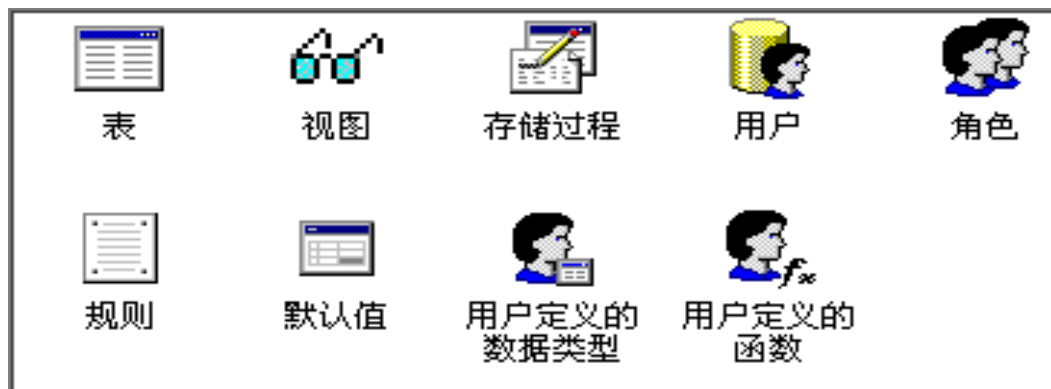


4、事件探查器



数据库对象

在 SQL Server 2000 中，数据库由一些表和其他对象组成的，通过数据库可以为执行与数据有关的活动提供支持。



5.2 创建、修改和删除数据库

在创建之前，必须确认数据库的名称、大小，以及用于存储该数据库的文件和文件组。

创建数据库的两种方法：

1. 使用企业管理器
2. 用 CREATE DATABASE 语句创建

创建数据库

1. 企业管理器创建

右键单击“数据库”或数据库文件，选择“新建数据库”



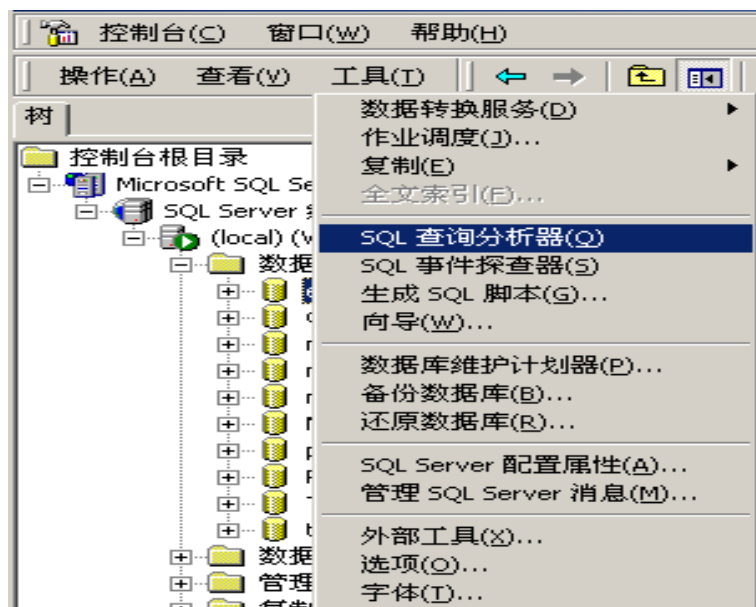
在“名称”文本处写上“Students”，再点击“确定”按钮。



2. 用 CREATE DATABASE 语句创建

单击“工具”，选择“SQL 查询分析器”

create database Students



修改数据库

在创建数据库后，可以对其原始定义进行更改。

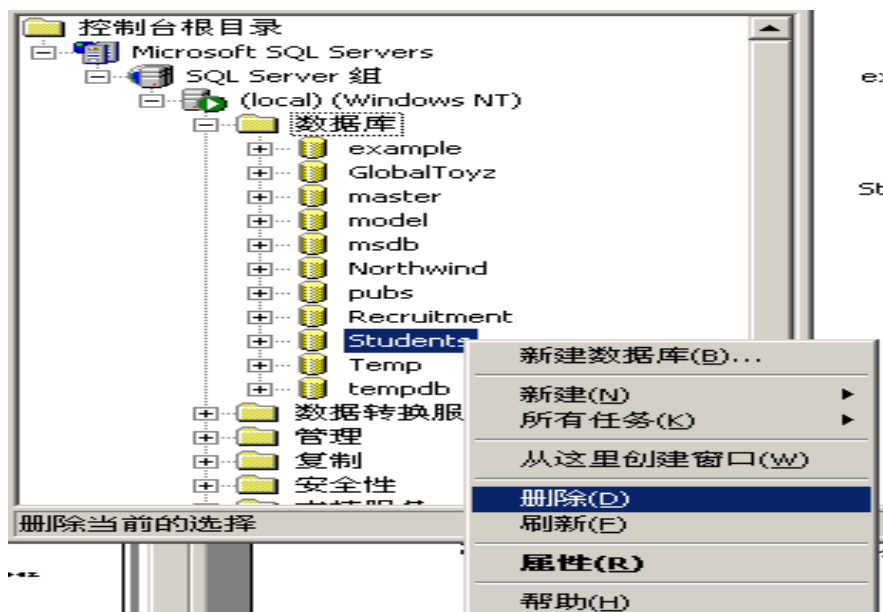
修改数据库可分以下几种

1. 扩充数据库
2. 收缩数据库
3. 扩展事务文件
4. 分离和附加数据库

删除数据库

1. 用企业管理器
2. 用 DROP DATABASE 语句删除数据库

使用企业管理器删除数据库



用 **DROP DATABASE** 语句删除数据库

在 SQL 查询分析器中写 SQL 语句如下：

```
DROP DATABASE Students
```

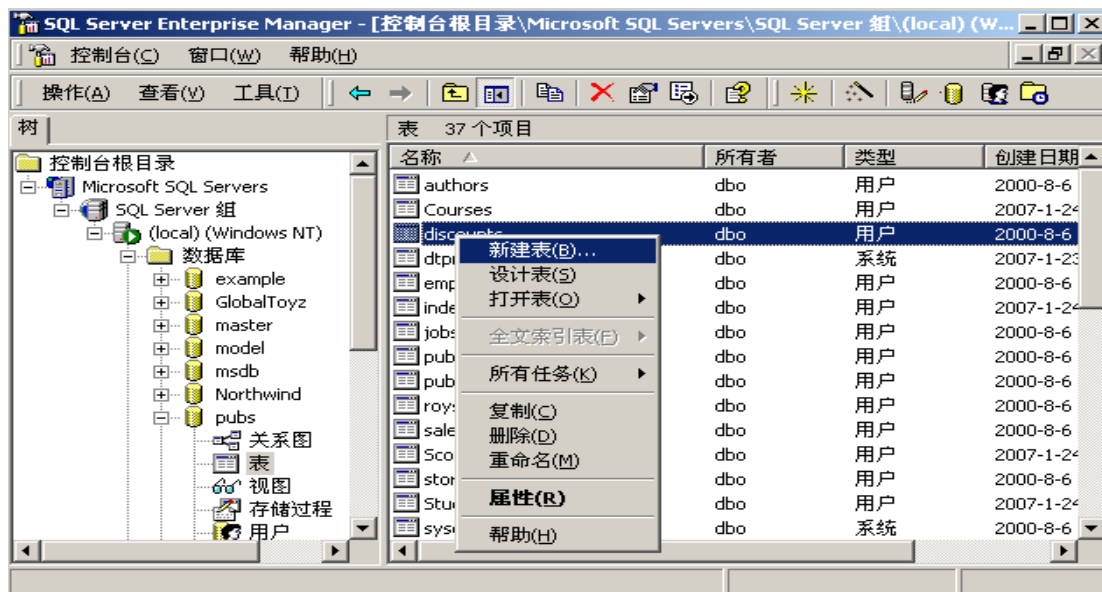
5.3 表的创建和管理

表的创建

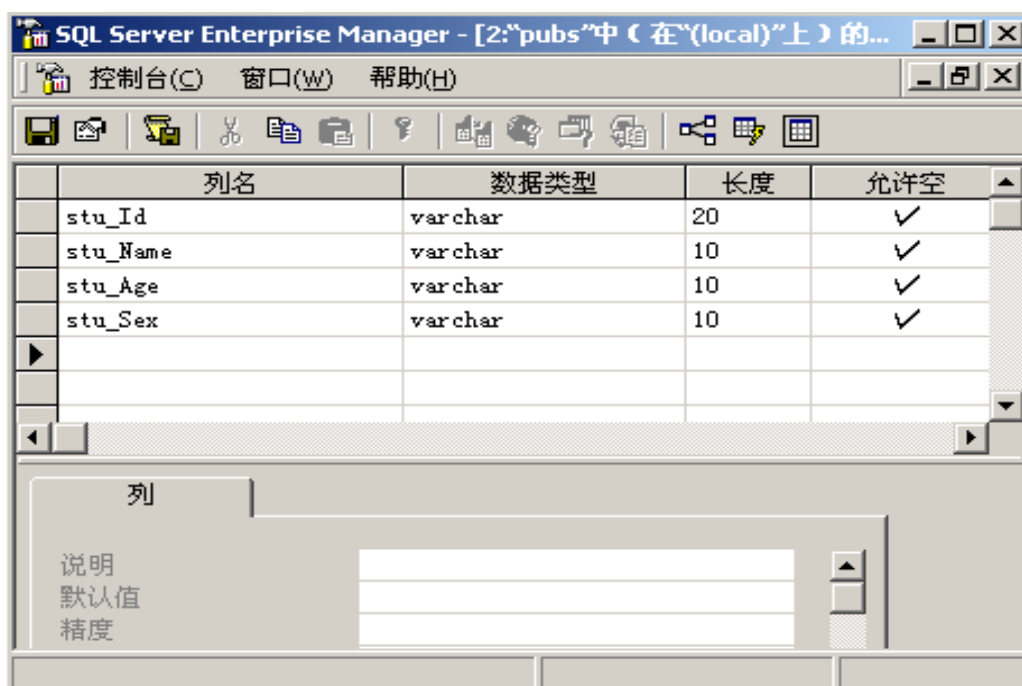
1. 使用企业管理器创建表

2. 使用 **CREATE TABLE** 语句创建表

- 1. 使用企业管理器创建表
- 在数据库里，单击“表”，然后在右边任一处单击右键，选择“新建表”



在“列名”中写上需要的字段，然后根据需求，设置好数据类型以及长度



单击保存，然后弹出如下界面，输入表的名称，再点击“确定”即可。



添加表

使用 **CREATE TABLE** 语句创建表

Use pubs

Create table Students

```
(  
    sut_Id nvarchar (20),  
        sut_Name nvarchar(10),  
        stu_Age nvarchar(10),  
        Stu_Sex nvarchar(10)  
)
```

修改表

使用企业管理器修改表

使用 **ALTER TABLE** 语句修改表，把 **stu_Id** 字段的数据类型改为 **char**,长度改为 10。

Use pubs

Alter table students

Alter column stu_Id char (10)

删除表

使用企业管理器删除表

使用 **DROP TABLE** 语句删除表

右键单击要删除的表，选择“删除”



使用 DROP TABLE 语句删除表

Use pubs

Drop table Students

【练习】SQL server 数据库文件有 3 类，其中主数据文件的后缀为（C）

- A .ndf B .ldf
- C .mdf D .idf

【练习】SQL Sever 2000 是一个（C）的数据库系统

- A 网状型 B 层次型
- C 关系型 D 以上都不是

【练习】SQL Sever2000 提供了一整套管理工具和实用程序，其中负责启动、暂停和停止 SQL Sever 的 4 种服务的是（D）

- A 企业管理器 B 导入和导出数据
- C 事件探索器 D 服务管理器

简单查询语句

- 在学习查询之前，要知道表是由很多字段(列)组成，查询时也只是查询表中的字段内容。
- SQL 是标准化的查询语言。通过语句完成查询任务。语句一般较为复杂。为了方便理解，常常分解成语句段进行讲解。

什么是 SQL?

SQL 是 Structured Query Language(结构化查询语言)的缩写。SQL 是专为数据库而建立的操作

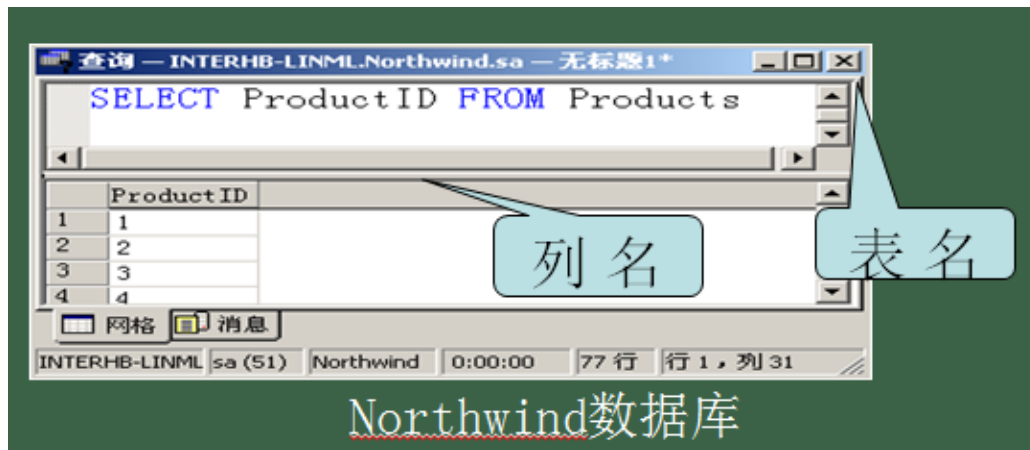
命令集，是一种功能齐全的数据库语言。

在使用它时，只需要发出“做什么”的命令，“怎么做”是不用使用者考虑的。

SELECT 语句

最常用的语法：

SELECT 列名 FROM 表名 例如：



从上例中，可以看到，在关键字“SELECT”后面有一个列名(字段名)“ProductID”，运行后得到的结果是这个列的所有信息。也就是说在关键字“SELECT”后面的就是要查询的字段。

在关键字“FROM”后的“Products”是表的名字，作用是告诉电脑要查询的字段是在哪个表中。

如果要查询“Products”表中的所有信息，可以写成：

```
SELECT ProductID , ProductName,  
SupplierID , CategoryID.....
```

```
FROM Products
```

将所有列名都写出来这样很繁琐，可以用“*”字符代替。写法如下：

```
SELECT * FROM Products
```

SQL 对大小写不敏感，可以将上面查询语句改写成：

```
select * from products
```

WHERE 子句

以关键字“WHERE”引导的语句叫 WHERE 子句。它作用是为查询进行限制，也就是为查询设置条件。

例：查询一个叫“Chang”的产品。

```
SELECT ProductName , Unitprice
```

```
From Products
```

```
WHERE ProductName = 'Chang'
```

在 WHERE 子句中，可以使用“>”、“>=”、“<”、“<=”、“!=”、“NOT”、“AND”等逻辑运算符。

例：查询“Products”表中，产品编号小于 “3”的产品总计金额(单价×存货数量)。

```
SELECT ProductID , ProductName,  
        Unitprice * UnitsInStock AS 总计  
FROM Products  
WHERE NOT ProductID >= 3
```

5.3 备份和恢复数据库

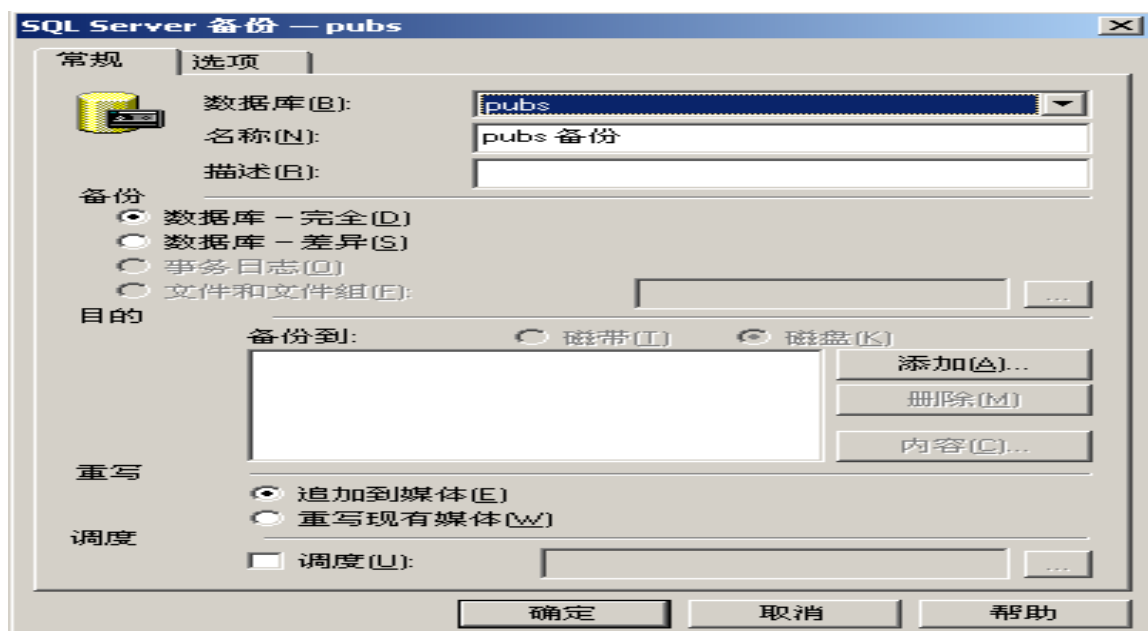
备份数据库

在 SQL Server 2000 中，备份和恢复为存储在 SQL Server 数据库中的关键数据提供了重要的保护手段

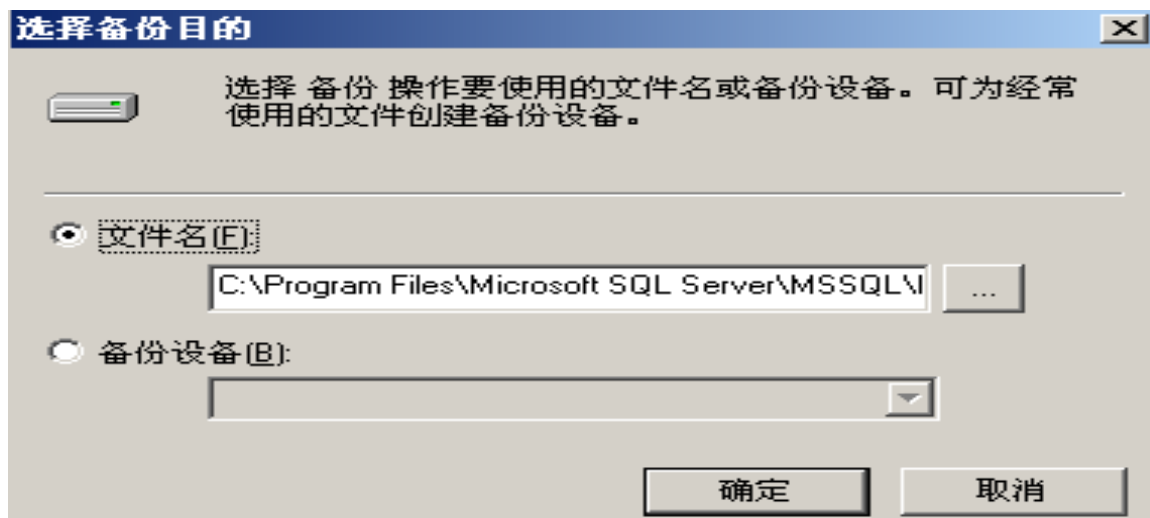
1. 使用企业管理器备份数据库

步骤：

点击“操作”->“所有任务”->“备份数据库”



把备份数据库名命名好之后，点击“确定”，再确定，就完成了。



2. 使用 SQL 语句创建数据库备份

用 BACKUP 语句执行备份操作。对整个数据备份进行备份时，BACKUP 语法格式可以简化为：

BACKUP DATABASE 数据库名 TO 备份设备名

以下程序使用 SQL 语句创建一个名为 pubs01 的备份设备，并将 pubs 示例数据库备份到该设备上。

```
exec sp_addumpdevice 'disk','pubs01','f:\Stud.dat'
```

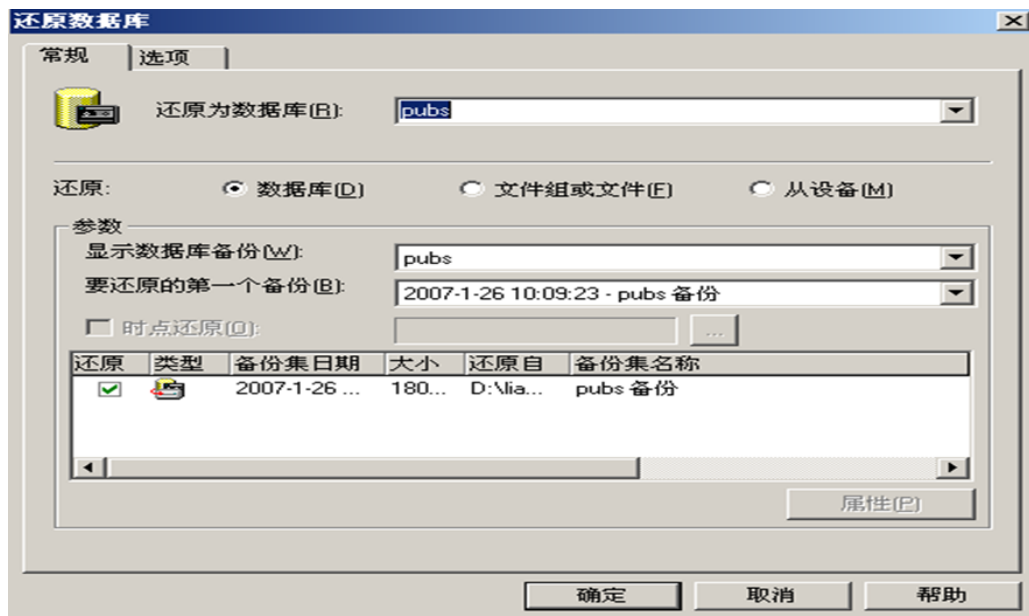
```
backup database pubs to pubs01
```

1. 恢复数据库

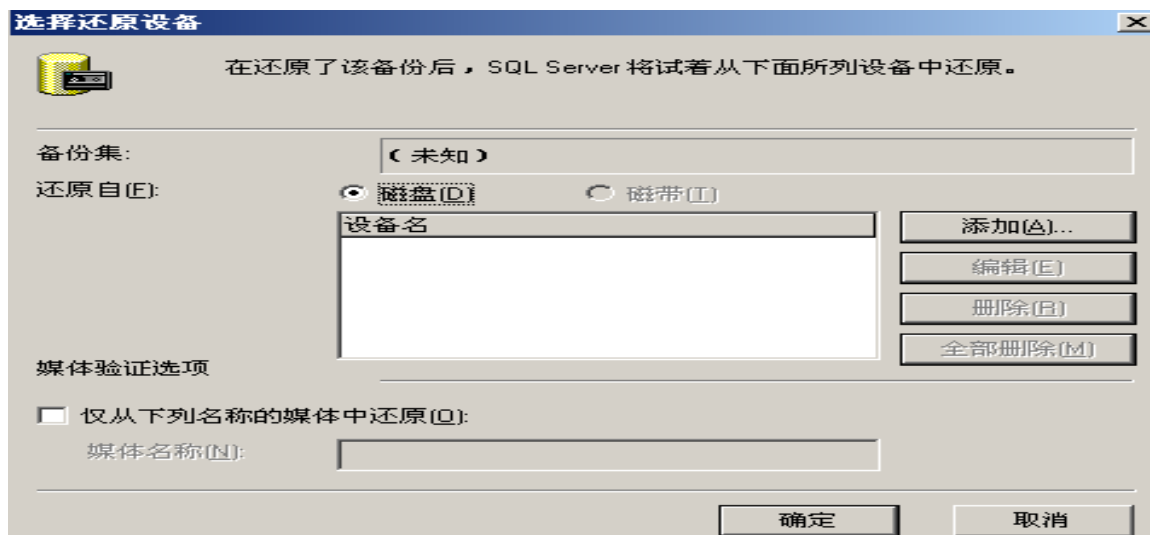
使用企业管理器恢复数据库

步骤：

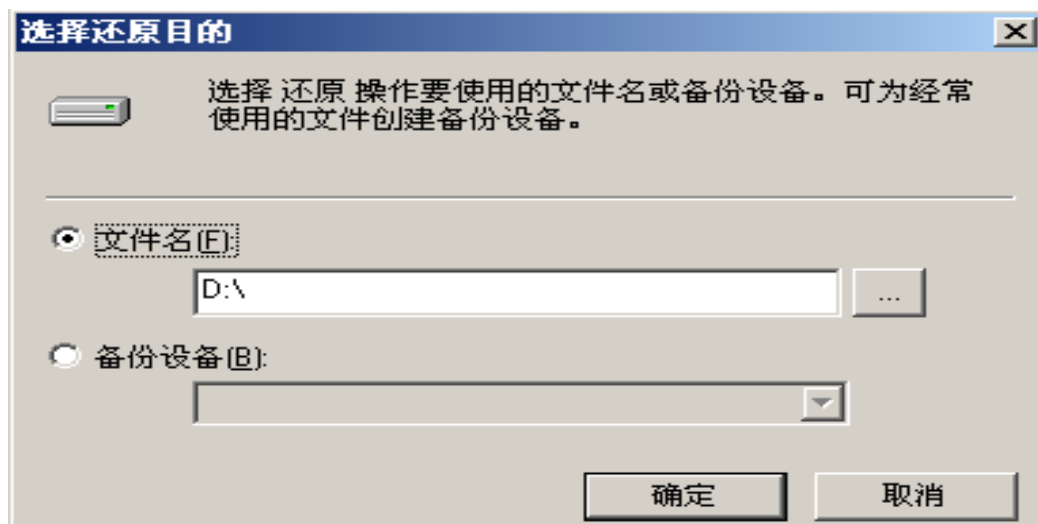
点击“操作”->“所有任务”->“还原数据库”



选择“从设备” ->点击“选择备份”



点击“添加”，再选择刚才的备份文件路径即可



2. 使用 SQL 语句恢复数据库备份

通过执行 RESTORE 语句可以完成还原数据库备份的任务。

格式如下：

RESTORE DATABASE <数据库名> FROM <备份设备名>

restore database pubs from pubs01

【练习】在 SQL 语言的 SELECT 语句中，实现投影操作的是（ A ）子句。

- A. SELECT B. FROM
- C. WHERE D. GROUP BY

【练习】SQL 是（D）的缩写形式。

- A Selected Query Language B Procedured Query Language
- C Standard Query Language D Structured Query Language

【练习】下列 SQL 语言的语句中，最常用的语句是（D）

- A 插入语句 B 删除语句
- C 创建语句 D 查询语句

【练习】SELECT 查询中，要把结果中的行按照某一列的值进行排序，所用到的子句是（A）

- A ORDER BY B WHERE
- C GROUP BY D HAVING

6、SQL 结构化查询语言

数据库模式的建立和删除

表结构的建立、修改和删除

表内容的插入、修改和删除

SQL 查询

6.1 SQL 概述

SQL 具有数据定义(DDL)、数据操纵(DML)、数据控制(DCL)等功能。SQL 的数据定义功能能够定义数据库的三级模式结构，即外模式-视图(View)、全局模式-模式(Schema)或数据库(Database)，内模式由系统根据数据库模式自动实现，一般无需用户过问。

在 SQL 中，每个关系又叫做基本表或表(Table)，每个关系中的属性又叫字段或列，元组又叫行。一个数据库由若干基本表组成，通常一个基本表对应存储在外存数据库空间的一个存储文件中。

每个视图也是一个关系，它由基本表产生出来，有自己独立的结构定义，但没有独立的数据存在，它的数据来自基本表，间接地来自存储基本表的数据存储文件，所以又称视图为虚表，而基本表为实表。

SQL 的数据操纵功能包括对基本表和视图的数据查询、插入、删除和修改，特别是具有很强的数据查询功能。

SQL 的数据控制主要是对用户的访问权限加以控制，以保证系统的安全性。由 DBA 负责进行控制和管理。

6.2 常见 SQL 语句

数据库模式的建立

建立数据库模式的命令：

```
CREATE {SCHEMA|DATABASE} <数据库名>  
[AUTHORIZATION <所有者名>]
```

例：create database SC //建立 SC 数据库

语句说明：标识符/大小写等效

数据库的所有者必须为数据库系统的合法用户，且具有建立数据库的权限

打开数据库：USE<数据库名>

数据库模式的删除

删除数据库模式的命令：

```
DROP {SCHEMA|DATABASE} <数据库名>
```

例： DROP database SC //删除 SC 数据库

在实际操作中建立和删除数据库模式通常是通过窗口界面实现的

表的相关操作

```
CREATE TABLE <表名>
```

```
ALTER TABLE <表名>
```

```
DROP TABLE [...] <表名>
```

表内容的插入

单行插入

```
INSERT [INTO] <表名>(<列名>,...)VALUES(<列值>,...)
```

多行插入

```
INSERT [INTO] <表名>(<列名>,...) <SELECT 子句>
```

表内容的修改

```
UPDATE [...] <表名> SET <列名>=<表达式>,... [FROM<源表名>,...][WHERE<逻辑表达式>]
```

语句说明：FROM 选项(给出非当前表)，WHERE 选项(若省略该选项，则修改目的表中的所有记录)

表结构的数据类型

数据类型

1、char(n)：定长字符型

2、int/integer：整型/整数型，占用 4 个字节

3、float：浮点型/实数型，占用 4 或 8 个字节

4、date/datetime：日期型，占用 4 或 8 个字节，格式为 yyyy-mm-dd 或 yyyy/mm/dd

字符数据和日期数据在书写时用单引号括起来

表内容的删除

DELETE [FROM] <表名> [FROM<源表名>...][WHERE<逻辑表达式>]

语句功能：删除表中满足条件的所有行

语句说明：FROM 选项(给出非当前表)，WHERE 选项(给出删除记录的条件，若省略则删除表中的所有记录)

6.3SQL 查询（重点）

SELECT 查询语句具有丰富的数据查询功能，能够实现关系运算中的大多数运算，如选择、投影、连接、并等，并且还带有分组、排序、统计等数据处理功能。

SELECT 查询语句的结果有多种可能，有可能为空、单值元组、一个多值元组等，若为单值元组时，此查询可以作为一个数据项出现在任何表达式中。

SELECT 语句可以作为一个语句成分（即子查询）出现在各种语句中，若在 SELECT 语句的 WHERE 选项中仍使用一个 SELECT 语句，则称为 SELECT 语句的嵌套。

SQL 查询只对应一条语句，即 SELECT 语句。该语句带有丰富的选项（子句），每个选项都由一个特定的关键字标识，后跟一些需要用户指定的参数。

SELECT 语句格式

SELECT ...	规定结果中的列等
INTO...	将查询结果写入到一个基本表中
FROM...	指定要查询的基本表或视图
WHERE...	指定查询的条件或源表的连接条件
GROUP BY...	指定将查询结果分组的依据
HAVING...	筛选出符合条件的分组统计信息
ORDER BY...	将查询结果进行排序

SELECT 选项

规定在查询结果中每一行(即查询结果中的每一条记录)所包含的列

规定在查询结果中是否允许出现重复的行(即内容完全相同的记录)

Select<表达式>中允许使用函数

SELECT 语句中使用的列函数

COUNT([all | distinct] <列名> | *) 个数

MAX(列名) 最大值

MIN(列名) 最小值

AVG(列名) 平均值

SUM(列名) 和

FROM 选项

提供用于查询的基本表和视图，并可以为每个基本表取一个别名。

别名的作用：作为列名的前缀限定符，以区别于其他源表中相同的列名。如果源表的表名较长，取个简单的别名方便使用。

做“连接”运算

例：... from 商品表 1 as x, 商品表 2 as y...

例：...from 学生 x, 课程 y, 选课 z where x.学生号=z.学生号

WHERE 选项

该子句的功能是指定对记录的筛选条件和源表之间的连接条件。

格式：where <逻辑表达式>

做“选择”运算

例：...where 单价>2000...

例：...where 商品表 1.商品号=商品表 2.商品号

WHERE 选项_逻辑表达式

用于查询语句中的逻辑表达式有两种：一般比较式，专门比较式

一般比较式：<表达式 1> <比较符> <表达式 2>

专门比较式：又叫判断式，它实现单值与集合数据之间的比较，常用有六种格式（ALL、ANY|SOME、[NOT] BETWEEN、[NOT] EXISTS、[NOT] IN、[NOT] LIKE）

WHERE 选项_一般比较式

格式

<表达式 1> <比较符> <表达式 2>

<表达式 1>和<表达式 2>是由列名、常量、变量、函数等通过算术运算符连接而成的式子，最简单的表达式是一个列名或常量

<比较符>是六种比较运算符，也称为 θ 运算

GROUP BY 选项

格式：

group by <分组列名 1>[, <分组列名 2>]...

功能：使查询结果只包含按指定列的值进行分组的统计信息，即按指定的列名对投影或选择后得到的所有元组进行分组

HAVING 选项

格式：having <逻辑表达式>

功能：通常用于 GROUP BY 子句之后，用来从分组统计结果中筛选出符合<逻辑表达式>的部分统计结果

<逻辑表达式>通常带有字段函数

ORDER BY 选项

格式：order by <排序列名 1> [asc | desc]...

功能：该选项用于对查询结果按<排序列名 1>的值进行升序(asc)或降序(desc)排列，若不指定排序方式，则默认按升序(asc)排列

INTO 选项

格式：into <表名>

此选项能够根据查询结果自动建立一个基本表，通常是作为临时表使用，可以根据需要任意建立和删除。

使用临时产生的表，同使用其它基本表一样，没有任何区别，当不再需要时，可使用 DROP TABLE 语句删除掉。

查询综合实例

假设学生选课数据库有三个表即学生表 S、课程表 C 和学生选课表 SC，它们的结构如下所示，请根据所给的每种功能写出相应的查询语句。

S (S# , SN , SEX , AGE , DEPT)

C (C# , CN)

SC (S# , C# , GRADE)

其中：S#为学号，SN 为姓名，SEX 为性别，AGE 为年龄，DEPT 为系别，C#为课程号，CN 为课程名，GRADE 为课程成绩。

学生表

学生号	姓名	性别	年龄	系别
0101001	王明	男	20	计算机
0102005	刘芹	女	21	电子
0202003	张鲁	男	20	电子
0303001	赵红	女	19	电气
0304006	刘川	男	20	通信

课程表

课程号	课程名
C001	C++语言
C004	操作系统
E002	电子技术
X003	信号原理

选课表

学生号	课程号	成绩
0101001	C001	78
0101001	C004	62
0102005	E002	73
0202003	C001	94
0202003	C004	65
0202003	X003	80
0303001	C001	76
0304006	E002	72

1、查询所有姓王的学生的姓名和性别

```
SELECT    SN, SEX
FROM      S
WHERE     SN LIKE '王*'
```

2、统计学生选课数据库中开出的课程总数

```
SELECT    COUNT (*)    AS 课程总数
FROM      C
```

3、查询每个学生选修每门课程的有关数据（姓名、课程名和成绩等）

```
SELECT    S. SN, C. CN, SC. GRADE
FROM      S, C, SC
WHERE     S. S#=SC. S# AND C. C#=SC. C#
```

4、从学生选课库中查询出被 3 名以上学生选修的所有课程的信息

```
SELECT    *
FROM      C
WHERE     EXISTS
(
    SELECT  C#
    FROM    SC
    WHERE   C.C#=SC.C#
    GROUP  BY C#
    HAVING  COUNT (*) > 3
)
```

5、从学生选课库中查询出最多选修了 1 门课（含未选任何课程）的全部学生信息

```
SELECT    *
FROM      S
WHERE     S# IN
(
    SELECT  S#
    FROM    SC
    GROUP  BY S#
    HAVING  COUNT (*) =1
)
```

)

OR NOT EXISTS

(SELECT *

FROM SC

WHERE S.S#=SC.S#

)

6、查询所有与“王明”同年出生的学生姓名、年龄和性别（假设库中只有一个学生的姓名为“王明”）

SELECT SN, AGE, SEX

FROM S

WHERE AGE=

(SELECT AGE

FROM S

WHERE SN="王明")

7、从学生选课库中查询出每门课程被选修的学生人数，并按所选人数的降序排列出课程号和选课人数

SELECT C. C#, COUNT(C. C#) AS 人数

FROM C, SC

WHERE C. C#=SC. C#

GROUP BY C. C#

ORDER BY 人数 DESC

【练习】SQL 语言的语句中，最核心的语句是 (D)

A 插入语句

B 删除语句

C 创建语句

D 查询语句

【练习】在 SQL 中，基本表的撤消(从数据库中删除表)可以用 (B)

A. DROP SCHEMA 命令

B. DROP TABLE 命令

C. DROP VIEW 命令

D. DROP INDEX 命令

【练习】在标准 SQL 中，建立数据库结构(模式)的命令为(A)

A. CREATE SCHEMA 命令

B. CREATE TABLE 命令

C. CREATE VIW 命令

D. CREATE INDEX 命令

【练习】有关系 R(sno,sname,age)，下列关于空值的查询语句中，不能产生正确结果的是(A)

A. SELECT sname FROM R WHERE age=NULL

B. SELECT sname FROM R WHERE age IS NULL

C. SELECT sname FROM R WHERE NOT(age IS NULL)

D. SELECT sname FROM R WHERE age IS NOT NULL

■ 华图网校介绍

华图网校（V.HUATU.COM）于2007年3月由华图教育投资创立，是华图教育旗下的远程教育高端品牌。她专注于公职培训，目前拥有遍及全国各地500万注册用户，已成为公职类考生学习提高的专业门户网站。

华图网校是教育部中国远程教育理事单位。她拥有全球最尖端高清录播互动技术和国际领先的网络课程设计思想，融汇华图教育十余年公职辅导模块教学法，凭借强大师资力量与教学资源、利用教育与互联网的完美结合，真正为考生带来“乐享品质”的学习体验，通过“高效学习”成就品质人生。

华图网校课程丰富多元，涵盖公务员、事业单位、招警、法院、检察院、军转干、选调生、村官、政法干警、三支一扶、乡镇公务员、党政公选等热门考试、晋升及选拔。同时，华图网校坚持以人为本的原则，不断吸引清华、北大等高端人才加入经营管理，优化课程学习平台，提升用户体验，探索网络教育新技术和教学思想，力争为考生提供高效、个性、互动、智能的高品质课程和服务。

华图网校将秉承“以教育推动社会进步”的使命，加快网站国际化进程，打造全球一流的网络学习平台。

我们的使命：以教育推动社会进步

我们的愿景：德聚最优秀人才，仁就基业长青的教育机构

我们的价值观：诚信为根、质量为本、知难而进、开拓创新。

- 咨询电话：400-678-1009
- 听课网址：v.huatu.com（华图网校）