

初识Oracle 11g



数据库基础

Oracle基础



数据库概述

- ✧ 数据库的基本概念
- ✧ 三个世界与两个模型
- ✧ E-R图的使用
- ✧ 关系数据库规范化的方法和步骤
- ✧ SQL的功能和语法格式



如果你是一个班主任，为及时掌握班级动态，提高班级管理效率，就应该对班级的各种信息进行及时收集，分析、处理。

问1：你认为应该分析处理那些数据？

（基本信息、成绩信息、档案信息）

问2：没有计算机你会怎么处理数据？

（手工表、计算器）

问3：有了计算机你会怎么处理数据？

WORD不便于统计排序, EXCEL可以统计排序，但表间数据的组合困难

问4：用什么办法解决？

用数据库系统可以处理复杂数据。（学生综合管理系统）



其他实例：

✎ 进销存数据库

✎ 电子病历

✎ Access示例数据库Northwind（罗斯文示例数据库）



1.1 数据库简介

1.1.1 数据库的基本概念

✎ **数据**：是数据库系统研究和处理的对象，本质上讲是描述事物的符号记录。文字、图形、图像、学生的档案记录、货物的运输情况等都是数据。

✎ **数据库**：长期存储在计算机内、有组织、可共享的数据集合。数据库中的数据按一定的数据模型组织、描述和储存，具有较小的冗余度，较高的数据独立性和易扩展性，并可为各种用户共享。是数据库系统的核心和管理对象。



✎ **数据库管理系统（DBMS）：**位于用户与操作系统之间的数据库管理软件，是系统软件，数据库在建立、运用和维护时由DBMS统一管理、统一控制。DBMS使用户能方便的定义数据和操纵数据，并能够保证数据的安全性、完整性、多用户对数据的并发使用及发生故障后的系统恢复。

✎ **数据库系统（database system）**通常是指带有数据库的计算机应用系统。一般由数据库、数据库管理系统（及其开发工具）、应用系统、数据库管理员（DBA）和用户组成。

1.1.2数据库技术的发展

数据管理：是指如何对数据分类，组织，编码，储存，检索和维护，是数据处理的中心问题。

数据管理经历了人工管理，文件系统和数据库系统三个阶段。

1. 人工管理阶段

在20世纪50年代中期以前，计算机主要用于科学计算。当时的硬件状况是，外存只有纸带、卡片、磁带，没有磁盘等直接存取的存储设备；软件状况是，没有操作系统，没有管理数据的软件；数据处理方式是批处理。



人工管理数据的特点：

- ✎ 数据不保存，主要用于科学计算，计算某一课题时输入数据，用完就撤走。
- ✎ 应用程序管理数据。应用程序不仅要设计数据的逻辑结构，还要设计物理结构，包括存储结构、存取方法、输入方式等。
- ✎ 数据不共享。一组数据只能对应一个程序。多个程序涉及相同的数据时，必须各自定义，冗余较大。
- ✎ 数据不具有独立性，数据结构变化后，必须对应用程序做相应的修改。



2 . 文件系统阶段

20世纪50年代后期到60年代中期，计算机的应用范围逐渐扩大，计算机不仅用于科学计算，而且还大量用于管理。这时硬件上已有了磁盘、磁鼓等直接存取存储设备；软件方面，操作系统中已经有了专门的数据管理软件，一般称为文件系统；处理方式上不仅有了文件批处理，而且能够联机实时处理。



文件系统管理数据的特点：

✎ 数据可以长期保存。

✎ 由文件系统管理数据，文件系统把数据组织成相互独立的数据文件，利用“按文件名访问，按记录进行存取”。数据的最小存取单位是记录，不能细到数据项。程序和数据之间由文件系统提供存取方法进行转换，使应用程序和数据之间有了一定的独立性。

✎ 数据共享性差，冗余度大。不同应用程序具有部分相同数据时，也必须建立各自的文件。

✎ 数据独立性差。应用程序的修改，将引起文件数据结构的改变，反之亦然。



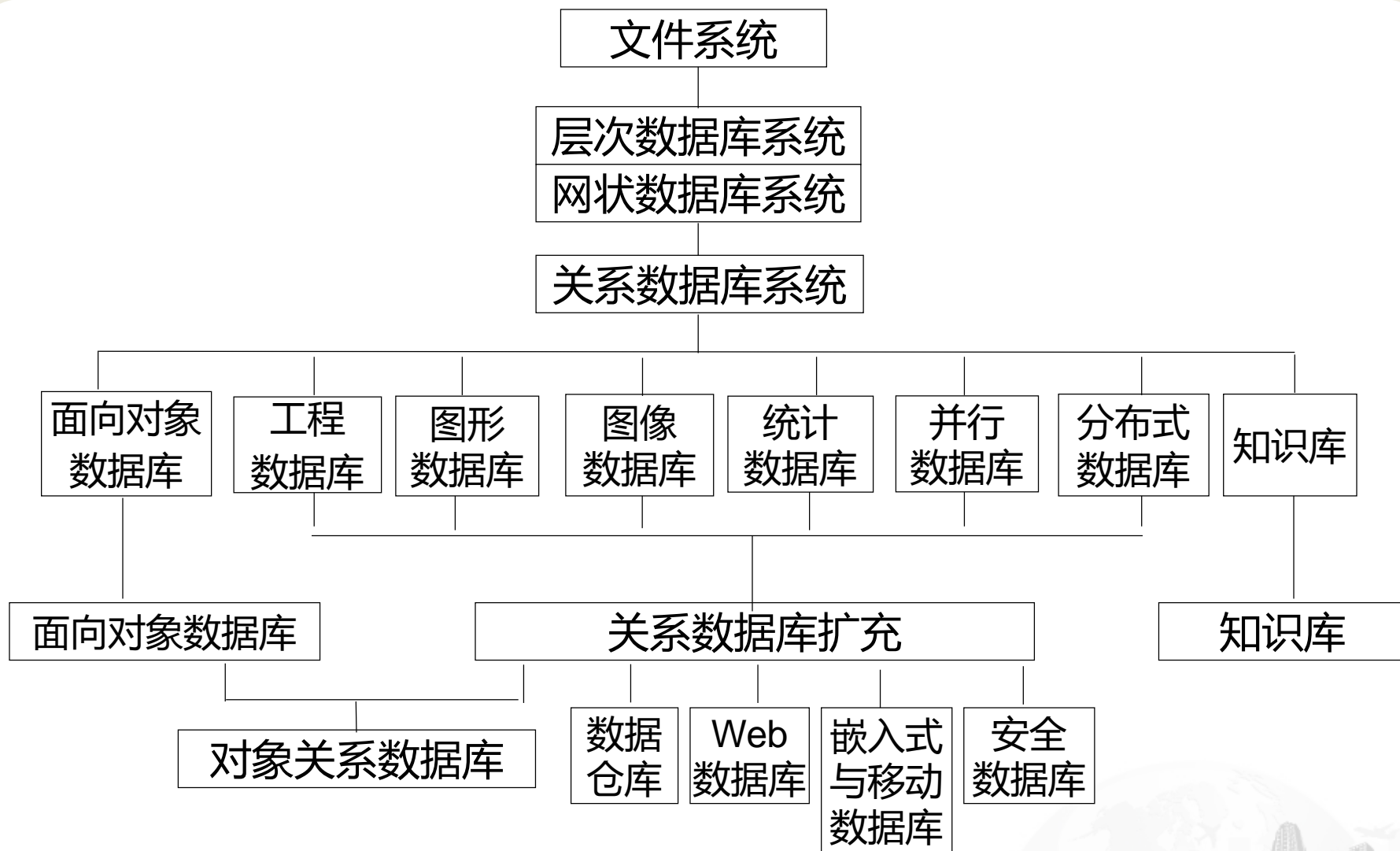
3 . 数据库系统阶段

二十世纪六十年代后期，硬件价格下降，软件价格上升，联机实时处理要求更多，并开始提出和考虑分布处理。在这种背景下，以文件系统作为数据管理手段已经不能满足应用的需求，于是为解决多用户、多应用共享数据的需求，使数据为尽可能多的应用服务，出现了数据库技术，出现了统一管理数据的专门软件系统 - 数据库管理系统。



- 1、60年代末第一代数据库系统：网状、层次数据库
- 2、70年代第二代数据库系统：关系数据库系统
- 3、第三代数据库系统
- 4、数据库技术的新进展





数据库系统发展简图

1.1.3 数据库技术的研究领域

1.数据库管理系统软件的研制：研制DBMS本身以及以其为核心的一系列相互联系的软件系统。目标是扩大功能、提高性能和提高用户的生产率。

2.数据库设计：主要研究方向包括数据库设计方法、设计工具和设计理论的研究，数据模型和数据建模的研究，计算机辅助数据库设计方法及其软件系统的研究。

3.数据库理论：主要集中于关系的规范化理论、关系数据理论等。如数据库逻辑演绎、知识推理、并行算法等理论研究，以及知识库系统和数据仓库的研制都已成为新的研究方向。



1.2 数据模型

数据模型就是现实世界的模拟，在数据库中用数据模型这个工具抽象、表示和处理现实世界中的数据和信息。

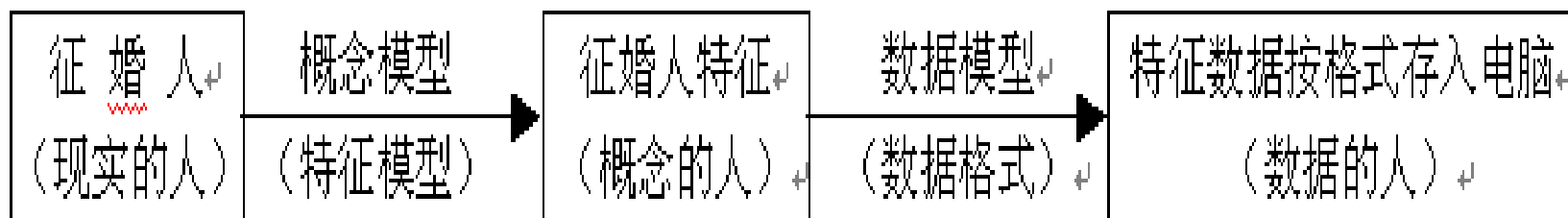
根据模型应用的不同目的，模型可以分为两类：

概念模型：也称信息模型，它是按用户的观点对数据和信息建模。

数据模型：主要包括网状模型、层次模型、关系模型等，它是按计算机系统的观点对数据建模。



✎ **课题引入：**假如你是一个婚介所的经理，要使用电脑对征婚人进行管理，那么实现的过程应该是：



现实世界

概念世界

数据世界



1.2.1 概念模型

1.三个世界

- 1) 现实世界：人们管理的对象存在于现实世界，现实世界的事物以及事物之间的联系是普遍存在的，是由事物本身的性质决定的。
- 2) 概念世界：概念世界是现实世界在人们头脑中的反映，是对客观事物及其联系的一种抽象描述。
- 3) 数据世界：将概念世界中的事物数据化存入计算机系统。



2.概念模型的主要概念

实体：客观存在并相互区别的事物及其事物之间的联系。例如，一个学生、一门课程、学生的一次选课等都是实体。

属性：实体所具有的某一特性。例如，学生的学号、姓名、出生年份、系、入学时间等。

码：唯一标识实体的属性集。例如，学号。

域：属性的取值范围。例如，性别的域为（男，女），分数的域为（0，100）



- ✎ **实体型**：用实体名及其属性名集合来抽象和刻画同类实体，称为实体型。例如，学生（学号，姓名，性别，出生年份，系，入学时间）
- ✎ **实体集**：同型实体的集合。例如，全体学生就是一个实体集。
- ✎ **联系**：实体与实体之间以及实体与组成它的各属性间的关系。联系有三种情况：一对一、一对多、多对多联系。



3.概念模型的表示方法

概念模型的表示方法很多，最常用的是实体 - 联系方法。该方法用E-R图来描述现实世界的概念模型，提供了实体型、属性和联系的方法。

实体型：用矩形表示，矩形框内写明实体名。

属性：用椭圆形表示，并用无向边将其与相应实体连接起来。

联系：用菱形表示，并用无向边将其与相应实体连接起来，同时，在无向边旁标上联系的类型（1：1，1：n，m：n）。



例如图1—1就是一个班级的概念模型的E—R图。

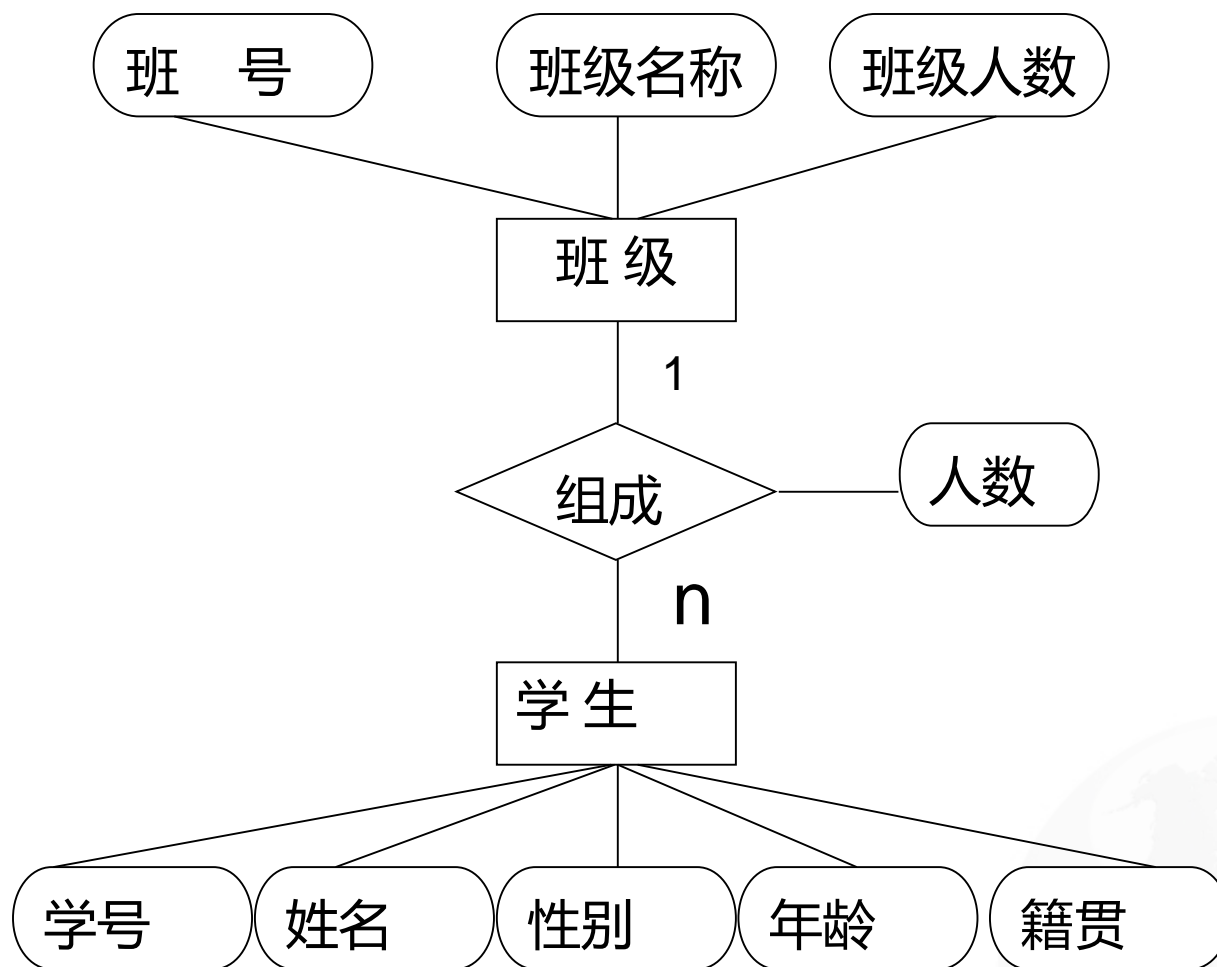


图1—1

联系的类型有三种：

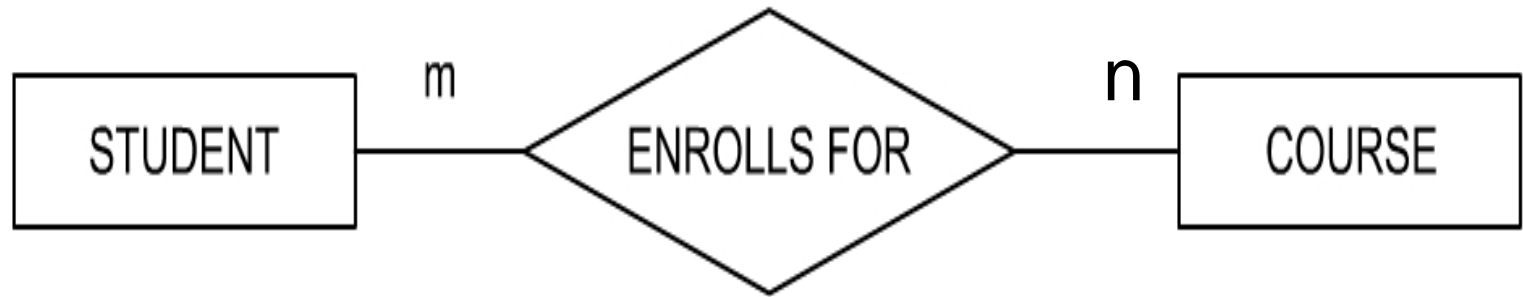
- ✎ 一对一关系：对于特定的系DEPARTMENT (象社会科学系)仅有一个系主任 (DEPARTMENT HEAD)。



✎ **一对多关系：一个学生 STUDENT 仅可以主修（ MAJOR ）一门课程，但是有许多学生 STUDENT 将注册给定的MAJOR课程。**



✎ **多对多关系：一个学生 STUDENT 可以学习许多课程 COURSE 和许多学生 STUDENT 可以注册给定的课程 COURSE。**



1.2.2 数据模型

数据模型有三个基本组成要素：数据结构、数据操作和完整性约束。

1. 数据结构：用于描述系统的静态特性，是所研究的对象类型的集合，数据模型按数据结构分为网状模型、层次模型、关系模型。

2. 数据操作：用于描述系统的动态特性，是指对数据库中各种对象的实例允许执行的操作的集合，包括操作及有关的操作规则。例如检索和更新（包括插入、删除、修改）

3. 数据的约束条件：是一组完整性规则的集合，完整性规则是给定的数据及其联系所具有的制约和储存规则，用以限定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。

1.2.3 数据模型

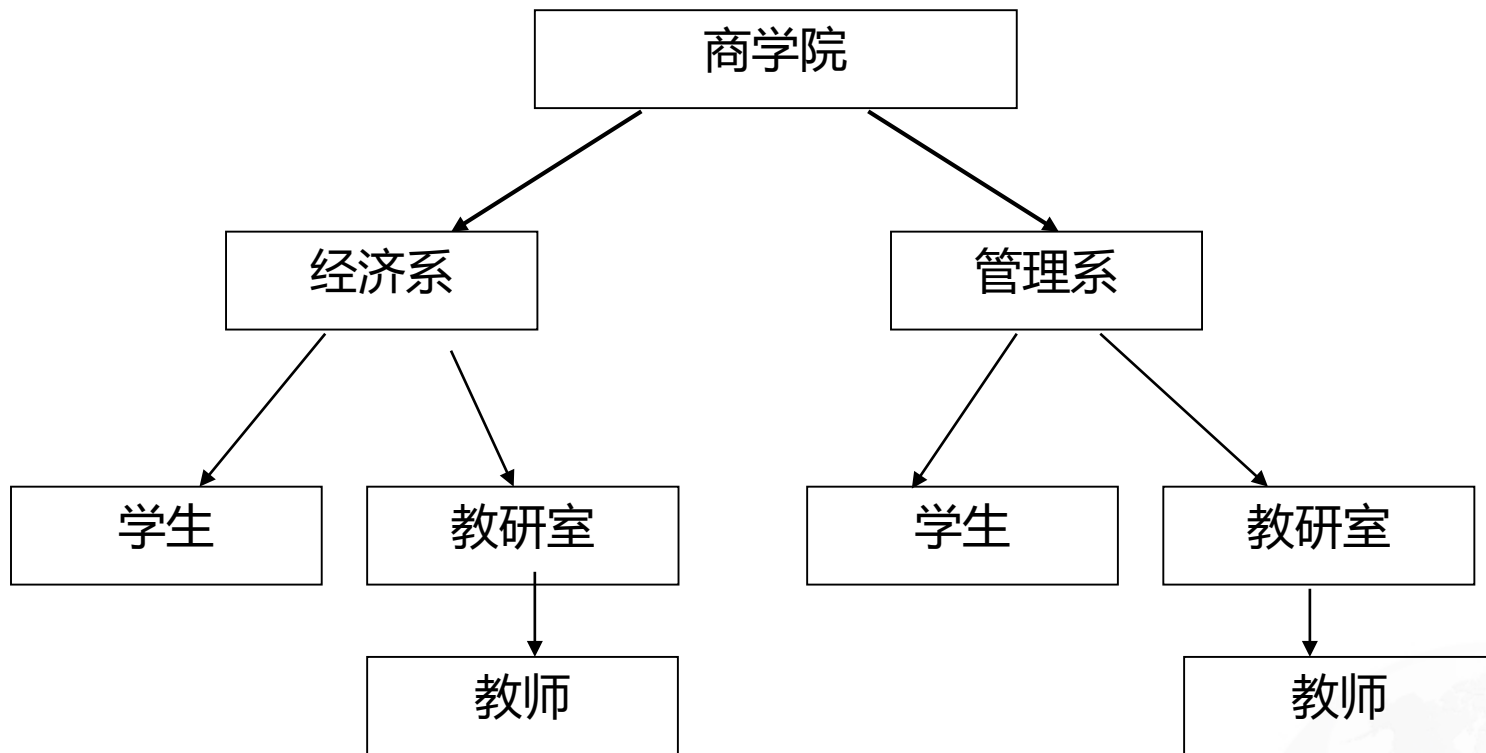
✎ 层次模型

✎ 网状模型

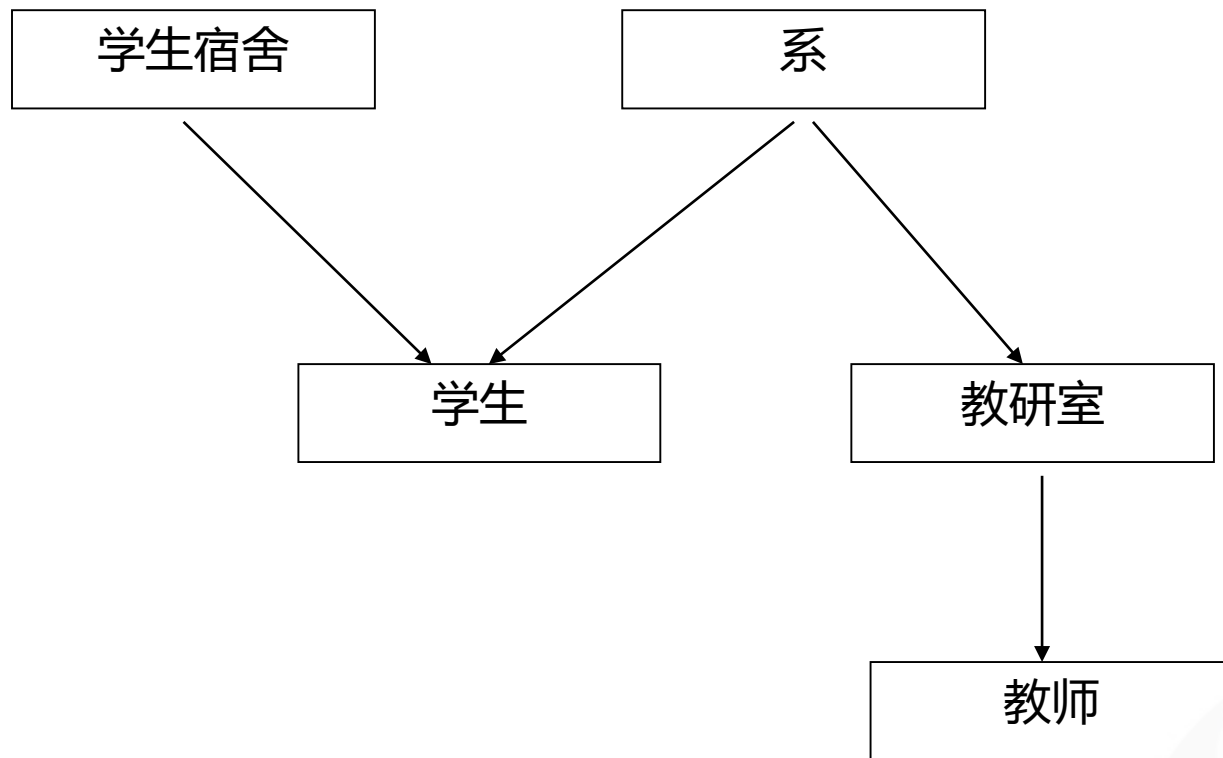
✎ 关系模型



1、层次模型



2、网状模型



3. 关系模型

(1) 关系数据结构

**关系模型（ relational model ）的特点是
用一组二维表来表示数据和数据之间的联系。**



表1-2

学号	姓名	性别	年龄	所在系
0000101	王萧	男	17	经济系
000207	李云虎	男	18	机械系
010302	郭敏	女	18	信息系
010408	高红	女	20	土木系
· · ·	· · ·	· · ·	· · ·	· · ·
020309	王睿	男	19	信息系
020506	路旭青	女	21	管理系

关系模型相关术语

✎ **关系**：一个关系对应一张二维表，表1-2中的这张学生记录表就是一个关系

✎ **元组**：表中的一行即为一个元组，对应表中一个记录，若表1-2有20行，就有20个元组；

✎ **属性**：表中的一列即为一个属性，表1-2有5列，对应5个属性(学号，姓名，性别，年龄和所在系)；

✎ **域(domain)**：属性的取值范围，如表1-2中学生年龄的域应是(16—28)，性别的域是(男，女)，系别的域是一个学校所有系名的集合。

✎ **码 (key)：**表中的某个属性（组），它可以唯一确定一个元组，则称该属性组为“候选码”。若一个关系有多个候选码，则选定其中一个为主码。如表1-2中的学号，是该学生关系的码。

✎ **外键：**设 F 是关系 R 的一个或一组属性，但不是 R 的候选码，如果 F 与关系 S 的主码 K_S 相对应，则称 F 是关系 R 的外键。

✎ **关系模式：**对关系的描述，一般表示为：

关系名(属性1，属性2，．．．，属性 n)

例如，上面的关系可描述为：

学生(学号，姓名，性别，年龄，所在系)



(2) 关系数据模型的存储结构

关系数据模型中，实体及实体间的联系都用表来表示。在数据库的物理组织中，表以文件形式存储，每一个表通常对应一种文件结构。



(3) 关系数据模型的优缺点

- ✎ 结构简单、清晰，用户易懂易用
- ✎ 关系模型的存取路径对用户透明，从而具有更高的数据独立性，更好的安全保密性，也简化了程序员的工作和数据库开发建立的工作
- ✎ 由于存取路径对用户透明，查询效率往往不如非关系数据模型。



1.3 关系数据库

关系模型是目前最重要的一种模型。美国IBM公司的研究员E . F . Codd于1970年发表题为“大型共享系统的关系数据库的关系模型”的论文，文中首次提出了数据库系统的关系模型。20世纪80年代以来，计算机厂商新推出的数据库管理系统(DBMS)几乎都支持关系模型，非关系系统的产品也大都加上了关系接口。数据库领域当前的研究工作都是以关系方法为基础。目前，最普遍的关系数据库有SQL Server 、 Sybase、 Oracle、 Access等。



1.3.1 关系数据的规范化概念

- ✎ 一组相互关联的数据称为一个“关系”
- ✎ “关系”下的每个数据项则称为“数据元素”
- ✎ “关系”落实到具体数据库上就是基本表，而“数据元素”就是基本表中的一个字段。表中的第一行用来存放字段名，称为“关系模式”，其余各行用来存放字段的值，称为“记录”。
- ✎ 同一张表中不应存放两个完全相同的记录



表1-1 专业表

专业编号	专业名称	办公室	电话
01	计算机科学与技术	1201	80688868
02	电子商务	1301	80688867
03	土木工程	1401	80688866
04	经济管理	1501	80688865
05	机械设计	2601	80688864

表1-2 学生表

学号	姓名	性别	年龄	专业编号
9901001	王凯	女	23	01
9901002	李云陆	男	21	01
9903003	刘敏	男	22	01
9903004	高红波	男	23	02
9904002	李睿	男.....	21.....	03.....
.....				

1. **关键字**：能够用来唯一标识记录的属性或属性的组合，称为“关键字”。如表1-4 课程选修表（包括字段学号、课程编号）
2. **主键**：被指定作为关键字的属性或属性组合，称为该表的“主关键字”或“主键”，如表1-1中指定专业编号作为该表的主关键字
3. **外键**：当数据表中的某属性或属性组合不是该表的关键字（或只是关键字的一部分），但却是另一数据表的关键字时，称该属性或属性组合为这个表的“外部关键字”或“外键”



4.关系的种类：

- a. 一对一关系
- b. 一对多关系
- c. 多对多关系



1.3.2 关系数据的规范化处理

- ✎ 关系数据库中的关系是要满足一定要求的，满足不同程度要求的为不同范式。目前遵循的主要范式包括第一范式（1 NF）、第二范式（2 NF）、第三范式（3 NF）、第四范式（4NF）等。
- ✎ 规范化设计的过程就是按不同的范式，将一个二维表不断地分解成多个二维表并建立表之间的关联，最终达到一个表只描述一个实体或者实体间的一种联系的目标。



1、第一范式

关系中每一个数据项必须是不可再分的，满足这个条件的关系模式就属于第一范式。关系数据库中的所有数据表都必然满足第一范式。

2、第二范式

在一个满足第一范式的关系中，如果所有非主属性都完全依赖于主码，则称这个关系满足第二范式。

3、第三范式

对于满足第二范式的关系，如果每一个非主属性都不传递依赖于主码，则称这个关系满足第三范式。

(详图见课本)



1.3.3 关系的完整性

关系的完整性规则是对关系的某种约束条件。在关系模型中有三类完整性约束：**实体完整性、参照完整性和用户定义的完整性**。其中，实体完整性和参照完整性是关系模型必须满足的完整性约束条件，被称为关系的两个不变性条件，应该由关系系统自动支持。



1、实体完整性

实体完整性规则：若属性A是基本关系R的主属性，则属性A不能取空值。

例如，在学生关系S(S#, SN, SS, SA, SD)中，S#属性为主码，则S#不能取空值。

**实体完整性规则规定基本关系的所有主属性都不能取空值，而不仅是主码整体不能取空值。
例如，学生选课关系SC(S#, C#,G)中，(S#,C#)为主码，则S#和C#两属性都不能取空值。**



2、参照完整性

现实世界中的实体之间往往存在某种联系，在关系模型中，实体及实体间的联系都是用关系来描述的。这样就自然存在着关系与关系间的引用。先来看1个例子。


【例1】在学生—课程关系数据库中，包括学生关系S、课程关系C和选修关系SC，这三个关系分别为：

学生(学号,姓名,性别,年龄,所在系)

课程(课程号,课程名,学分)

选修(学号,课程号,成绩)



 这三个关系之间也存在着属性的引用，即选修关系引用了学生关系的主码“学号”和课程关系的主码“课程号”。显然，选修关系中的学号值必须是确实存在的学生的学号，即学生关系中有该学生的记录；选修关系中的课程号值也必须是确实存在的课程的课程号，即课程关系中有该课程的记录。换句话说，**选修关系中某些属性的取值需要参照其他关系的属性取值。**

 不仅两个或两个以上的关系间可以存在引用关系，同一关系内部属性间也可能存在引用关系。



✎ 定义：设F是关系R的一个或一组属性，但不是关系R的码，如果F与关系S的主码Ks相对应，则称F是基本关系R的外码（foreign key），并称关系R为参照关系，关系S为被参照关系。

显然，被参照关系S的主码Ks和参照关系的外码F必须定义在同一个（或一组）域上。

✎ 在例子中，选修关系的“学号”属性与学生关系的主码“学号”相对应，因此“学号”属性是选修关系的外码；学生关系为被参照关系，选修关系为参照关系。选修关系的“课程号”属性与课程关系的主码“课程号”相对应，因此“课程号”属性也是选修关系的外码；课程关系为被参照关系，选修关系为参照关系。

✎ **参照完整性规则：若属性（或属性组）F是关系R的外码，它与关系S的主码Ks相对应（基本关系R和S不一定是不同的关系），则对于R中每个元组在F上的值为取空值（F的每个属性值均为空值），或者等于S中某个元组的主码值。**

✎ **例子中选修关系中每个元组的学号属性只能取下面两类值：**

（1）空值，表示尚未有学生选课。

（2）非空值，这时该值必须是学生关系中某个学生的学号，表示某个未知的学生不能选课。

✎ **同样，选修关系中每个元组的课程号只能取下面两类值：**

（1）空值，表示尚未开课。

（2）非空值，这时该值必须是课程关系中的某个课程号，表示不能选未开设的课。



3、用户定义的完整性

实体完整性和参照完整性适用于任何关系数据库系统。除此之外，不同的关系数据库系统根据其应用环境的不同，往往还需要一些特殊的约束条件。用户定义的完整性就是针对某一具体关系数据库的约束条件，它反映某一具体应用所涉及的数据必须满足的语义要求。例如，学生关系的年龄在15~30之间，选修关系的成绩必须在0~100之间等。




1.4 SQL的功能

表1-12 SQL语言的动词

SQL功能	动词
数据查询 数据定义 数据操纵 数据控制	SELECT CREATE, DROP, ALTER INSERT, UPDATE, DELETE GRANT, REVOKE

1 . 数据定义

(1) 定义表

 **CREATE TABLE 表名(**
列名 数据类型 [列级完整性约束条件]
[, 列名 数据类型 [列级完整性约束条件] . . .])
[, 表级完整性约束条件] ;



(2) 修改表

✎ ALTER TABLE 表名

[ADD新列名 数据类型 [完整性约束条件]

[DROP 完整性约束名]

[MODIFY 列名 数据类型] ;



(3) 删除表

✎ DROP TABLE 表名 ;

(4) 建立索引

✎ CREATE UNIQUE CLUSTER INDEX 索引名 ON 表名(列名 [次序] [, 列名[次序]]...);



2 . 数据查询

✎ **SELECT [ALL|DISTINCT 目标列表达式[, 目标列表达式]**

FROM 表名或视图名[, < 表名或视图名 > ; . . .

[WHERE 条件表达式]

[GROUP BY列名1 [HAVING 条件表达式]

[ORDER BY列名2 [ASC|DESC]



3 . 数据更新

(1) 插入数据

SQL用INSERT来插入数据。SQL通常有两种形式：

①插入单个元组，将新元组插入指定表中。语句格式为：

INSERT

INTO 表名 [(列名1[, 列名2]...)]

VALUES(常量1[, 常量2] ...)



②插入查询结果

语句格式为：

INSERT

INTO 表名 [(列名1[, 列名2]...)]

查询；

其功能是以批量插入，一次将查询的结果全部插入指定表中。



(2) 修改数据

修改操作用UPDATE实现，其语句的一般格式为：

UPDATE 表名

SET 列名 = 表达式 [, 列名 = 表达式] ...

[WHERE 条件] ;

其功能是修改指定表中满足WHERE条件的元组。其中SET子句用于指定修改值，即用表达式的值取代相应的属性列值。如果省略WHERE子句，则表示要修改表中的所有元组。



(3) 删除数据

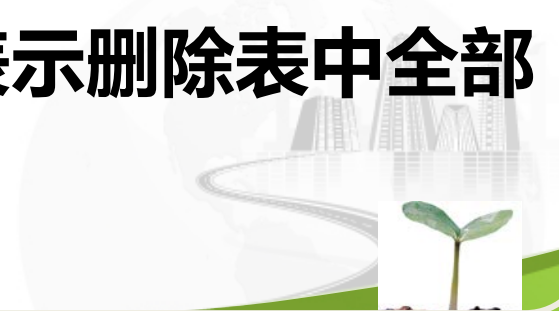
删除数据用DELETE语句，语句格式为

DELETE

FROM 表名

[WHERE 条件] ;

功能是从指定表中删除满足WHERE条件的所有元组。如果省略WHERE子句，表示删除表中全部元组。



1.5 数据库系统结构

1.5.1 数据库系统的模式结构

1.5.2 数据库系统的体系结构

1.5.3 数据库管理系统



1.5.1 数据库系统的模式结构

1. 数据库系统的三级模式结构

数据库系统是由外模式，模式和内模式三级模式组成，与之对应的是数据库的三级结构：局部逻辑结构、全局逻辑结构和物理存储结构。

（1）外模式：也称子模式或用户模式，它是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。一个数据库可以有多个外模式。



- (2) 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，是以数据模型为基础，综合考虑所有用户需求的公用数据视图，与具体数据值、应用程序、开发语言、开发工具无关。一个数据库只有一个模式。**
- (3) 内模式：也称存储模式，它是数据物理和存储结构的描述，是数据在数据库内部的表示方式，如，数据保存在磁盘、磁带，是什么形式、是否被压缩和加密等。一个数据库只有一个内模式。**



2 . 数据库的二级映象功能与数据独立性

数据库系统在这三级模式之间提供了两层映象：外模式/模式映象和模式/内模式映象。正是这两层映象保证了数据库系统的数据能够具有较高的**逻辑独立性**和**物理独立性**。



① 数据的逻辑独立性

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 / 模式映象，它定义了该外模式与模式之间的对应关系。当模式改变时(例如，增加新的数据类型、新的数据项、新的关系等)，由数据库管理员对各个外模式 / 模式的映象作相应改变，可以使外模式保持不变，从而应用程序不必修改，保证了数据的逻辑独立性。



② 数据的物理独立性

数据库中只有一个模式，也只有一个内模式，所以模式 / 内模式映象是唯一的，它定义了数据全局逻辑结构与存储结构之间的对应关系。当数据库的存储结构改变了(例如，采用了更先进的存储结构)，由数据库管理员对模式 / 内模式映象作相应改变，可以使模式保持不变，从而保证了数据的物理独立性。



1.5.2 数据库系统的体系结构

从最终用户角度来看，数据库系统分为单用户结构、主从式结构、分布式结构和客户 / 服务器结构。

1 . 单用户数据库系统

是一种早期的最简单的数据库系统。在这种系统中，整个数据库系统（包括应用程序、DBMS、数据）都装在一台计算机上，由一个用户独占，不同机器之间不能共享数据。



2 . 主从式结构

是指一个主机带多个终端的多用户结构。在这种结构中，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。

3 . 分布式结构

是指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。



4 . 客户 / 服务器结构

随着工作站功能的增强和广泛使用，人们开始把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，支持用户的应用，称为客户机，这就是客户 / 服务器结构的数据库系统。



在客户 / 服务器结构中，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（而不是整个数据），从而显著减少了网络上的数据传输量，提高了系统的性能、吞吐量和负载能力。另一方面，客户 / 服务器结构的数据库往往更加开放。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，应用程序具有更强的可移植性，同时也可以减少软件维护开销。



1.5.3 数据库管理系统

1. DBMS的功能

由于不同DBMs要求的硬件资源、软件环境是不同的，因此其功能与性能也存在差异，但一般说来，DBMS的功能主要包括以下6个方面。

(1) 数据定义

数据定义包括定义构成数据库结构外模式、模式和内模式，定义各个外模式与模式之间的映射，定义模式与内模式之间的映射，定义有关的约束条件，例如，为保证数据库中数据具有正确语义而定义的完整性规则、为保证数据库安全而定义的用户口令和存取权限等。

(2) 数据操纵

数据操纵包括对数据库数据的检索、插入、修改和删除等基本操作。

(3) 数据库运行管理

对数据库的运行进行管理是DBMS运行时的核心部分，包括对数据库进行并发控制、安全性检查、完整性约束条件的检查和执行、数据库的内部维护(如索引、数据字典的自动维护)等。所有访问数据库的操作都要在这些控制程序的统一管理下进行，以保证数据的安全性、完整性、一致性以及多用户对数据库的并发使用。



(4) 数据组织、存储和管理

数据库中需要存放多种数据，如数据字典、用户数据、存取路径等，DBMS负责分门别类地组织、存储和管理这些数据，确定以何种文件结构和存取方式物理地组织这些数据，如何实现数据之间的联系，以便提高存储空间利用率以及提高随机查找、顺序查找、增、删、改等操作的时间效率。

(5) 数据库的建立和维护

建立数据库包括数据库初始数据的输入与数据转换等。维护数据库包括数据库的转储与恢复、数据库的重组与重构造、性能的监视与分析等。



(6) 数据通信接口

DBMS需要提供与其他软件系统进行通信的功能。例如，提供与其他DBMS或文件系统的接口，从而能够将数据转换为另一个DBMS或文件系统能够接受的格式，或者接收其他DBMS或文件系统的数据库。



1.6 数据库新技术

- ❖ 数据库技术与分布处理技术相结合，出现了**分布式数据库**；
- ❖ 数据库技术与并行处理技术相结合，出现了**并行数据库**；
- ❖ 数据库技术与人工智能相结合，出现了**演绎数据库、知识库和主动数据库**；
- ❖ 数据库技术与多媒体处理技术相结合，出现了**多媒体数据库**；
- ❖ 数据库技术与模糊技术相结合，出现了**模糊数据库**。
- ❖ **数据仓库**



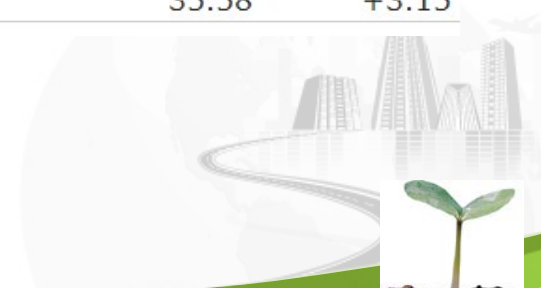
Oracle基础



常见数据库排行榜

April 2013

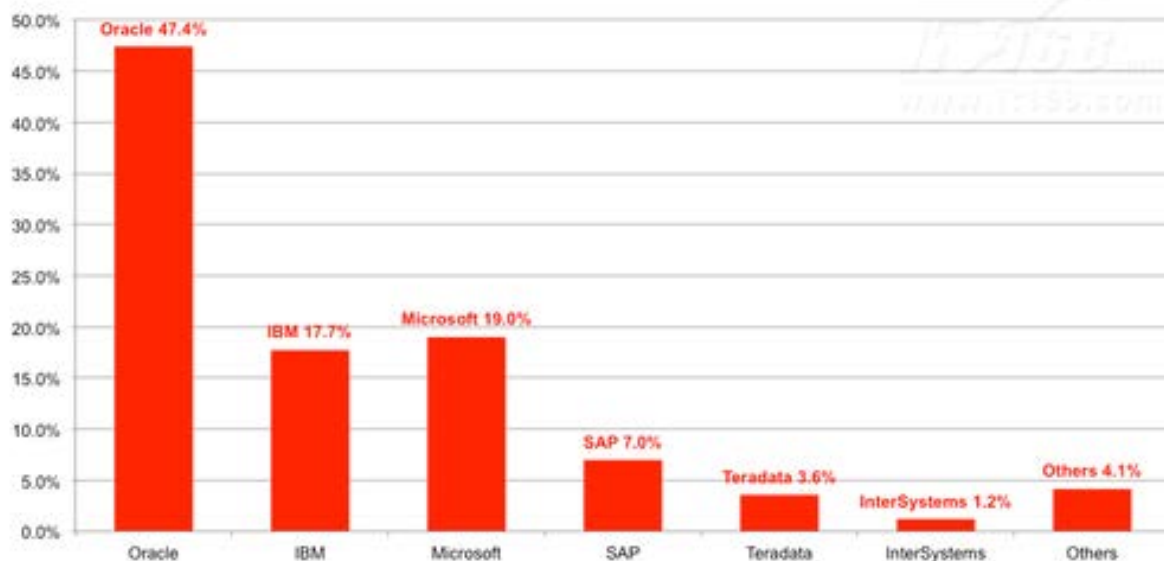
Rank	Last Month	DBMS	Database Model	Score	Changes
1.		1. Oracle	Relational DBMS	1560.59	+27.20
2.	↑	3. MySQL	Relational DBMS	1342.45	+47.24
3.	↓	2. Microsoft SQL Server	Relational DBMS	1278.15	-40.21
4.		4. PostgreSQL	Relational DBMS	174.09	-3.07
5.		5. Microsoft Access	Relational DBMS	161.40	-8.77
6.		6. DB2	Relational DBMS	155.02	-4.31
7.		7. MongoDB	Document store	129.75	+5.52
8.	↑	9. SQLite	Relational DBMS	88.94	+5.68
9.	↓	8. Sybase	Relational DBMS	80.16	-5.25
10.		10. Solr	Search engine	46.15	+2.99
11.		Teradata	Relational DBMS	44.93	
12.		11. Cassandra	Wide column store	38.57	+2.21
13.		12. Redis	Key-value store	35.58	+3.15





- ✎ Oracle数据库是Oracle（中文名称叫甲骨文）公司的核心产品，在所有的数据库管理系统中（比如：微软的SQL Server，IBM的DB2等）占有重要地位，用户涉及面非常广，包括：银行、电信、移动通信、航空、保险、金融、电子商务和跨国公司等。

Gartner: 2013 RDBMS 市场份额



Oracle11g简介

- ✎ Oracle 11g是甲骨文公司在2007年7月12日推出的最新数据库软件，Oracle 11g有400多项功能，经过了1500万个小时的测试，开发工作量达到了3.6万人/月。相对过往版本而言，Oracle 11g具有了与众不同的特性。
- ✎ Oracle是第一个跨整个产品线（数据库、业务应用软件和应用程序开发与决策支持工具）开发和部署100%基于互联网的企业软件的公司。Oracle是世界领先的信息管理软件供应商和世界第二大独立软件公司。

• 认证

Oracle认证专员(Oracle Certified Associate, 简称OCA)

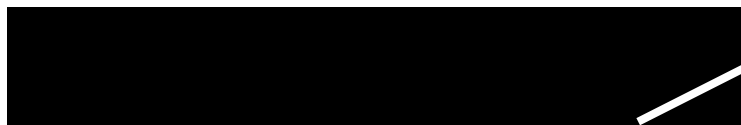
✎ **Oracle认证专家(Oracle Certified Professional, 简称OCP)**

✎ **Oracle认证大师(Oracle Certified Master, 简称OCM)**

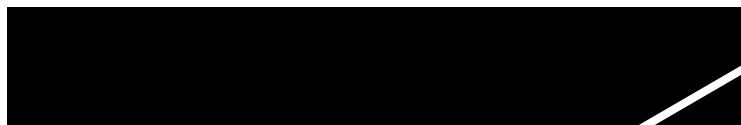


主要内容

1



2



3



Oracle概述

PART 1



安装Oracle11g

系统要求

系统要求	最小值
物理内存	1GB
虚拟内存	2 倍物理内存
可用磁盘空间	基本安装需要 4.55GB 高级安装需要 4.92GB
显示适配器	256 色
处理器	550MHz 主频的 Intel(x86)、AMD64 和 Intel EM64T 系列
系统结构	32 位或 64 位系统处理器
操作系统	Windows 2000 SP1 以上 Windows Server 2003 所有版本 Windows Server 2003 R2 所有版本 Windows XP Professional 版 Windows Vista Business、Enterprise 和 Ultimate 版
网络协议	TCP/IP (SSL)、命名管道
Web 浏览器	Netscape Navigator 7.2 以上 Mozilla 1.7 Microsoft Internet Explorer 6.0 SP2 以上 Firefox 1.0.4 以上



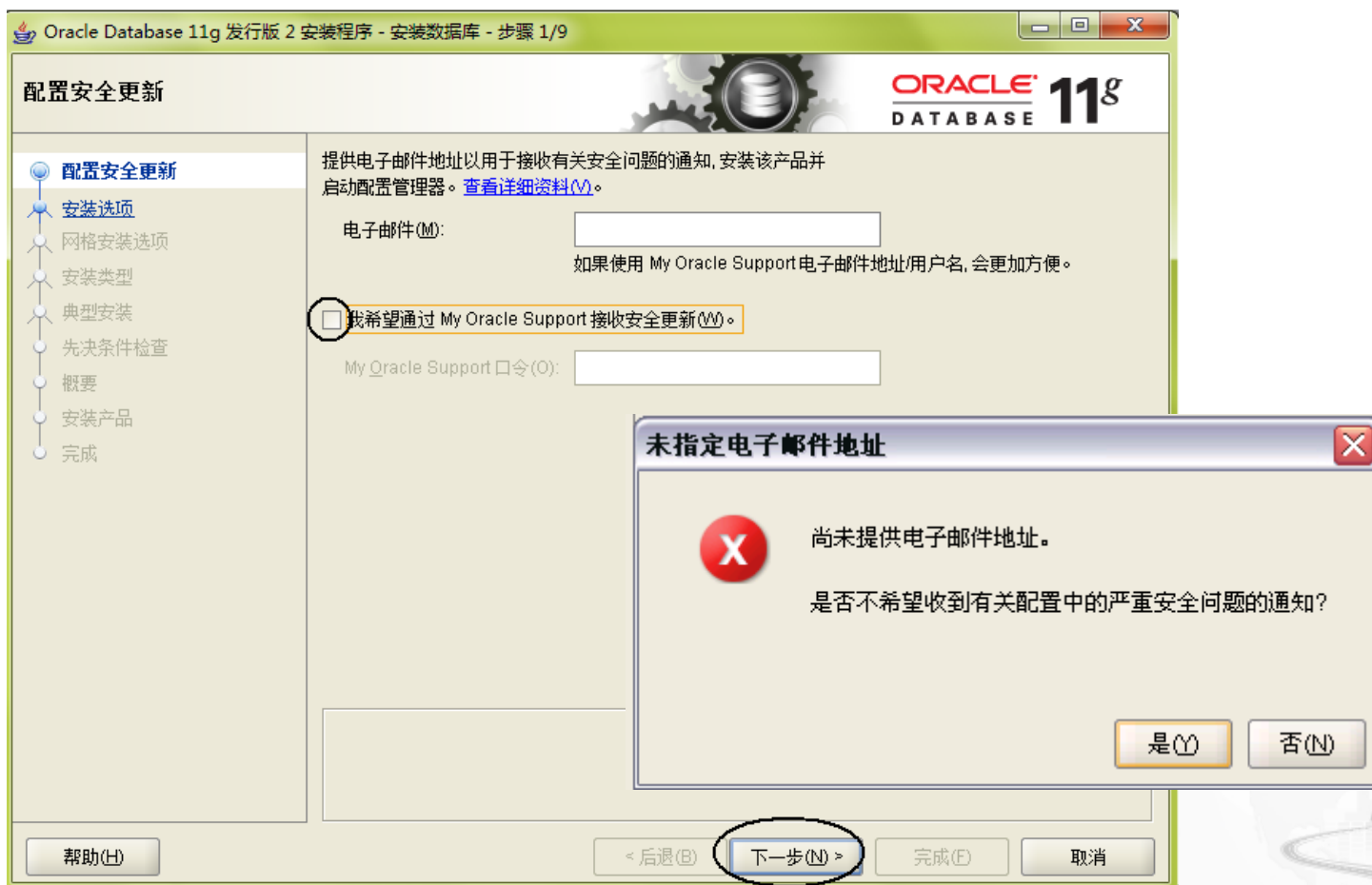
1. 下载racle11g后，解压到一个文件夹下，单击“setup.exe”文件即可启动安装界面。如下图：

```
正在启动 Oracle Universal Installer...  
检查监视器: 监视器配置至少必须显示 256 种颜色 高于 256 。 实际为 4294967296  
通过  
准备从以下地址启动 Oracle Universal Installer C:\Users\GONGLO~1\AppData\Local\Temp\OraInstall12010-05-12_06-28-43PM. 请稍候...
```



检测计算机软、硬件环境是否符合满足系统安装的最小需求。

2. 稍微等待一会，就会出现如下图所示的安装画面，取消下图所示的选中，然后单击"下一步"继续，同时在出现的信息提示框单击"是"继续。



3. 之后就会出现安装选项对话框，默认点击"下一步"继续



4. 然后就是安装配置，在这要注意的是：管理口令的格式要至少包含一个大写字母，一个小写字母和一个数字，否则会提示警告，正确输入后点击"下一步"继续。

Oracle Database 11g 发行版 2 安装程序 - 安装数据库 - 步骤 4/8

典型安装配置

配置安全更新
安装选项
系统类
典型安装
先决条件检查
概要
安装产品
完成

使用基本配置执行完全数据库安装。

Oracle 基目录(S): D:\app\GongLongchao 浏览(B)

软件位置(L): D:\app\GongLongchao\product11.2.0\dbhome_1 浏览(O)

数据库文件位置(D): D:\app\GongLongchao\oradata 浏览(W)

数据库版本(E): 企业版 (3.27GB)

字符集(C): 默认值 (ZHS16GBK)

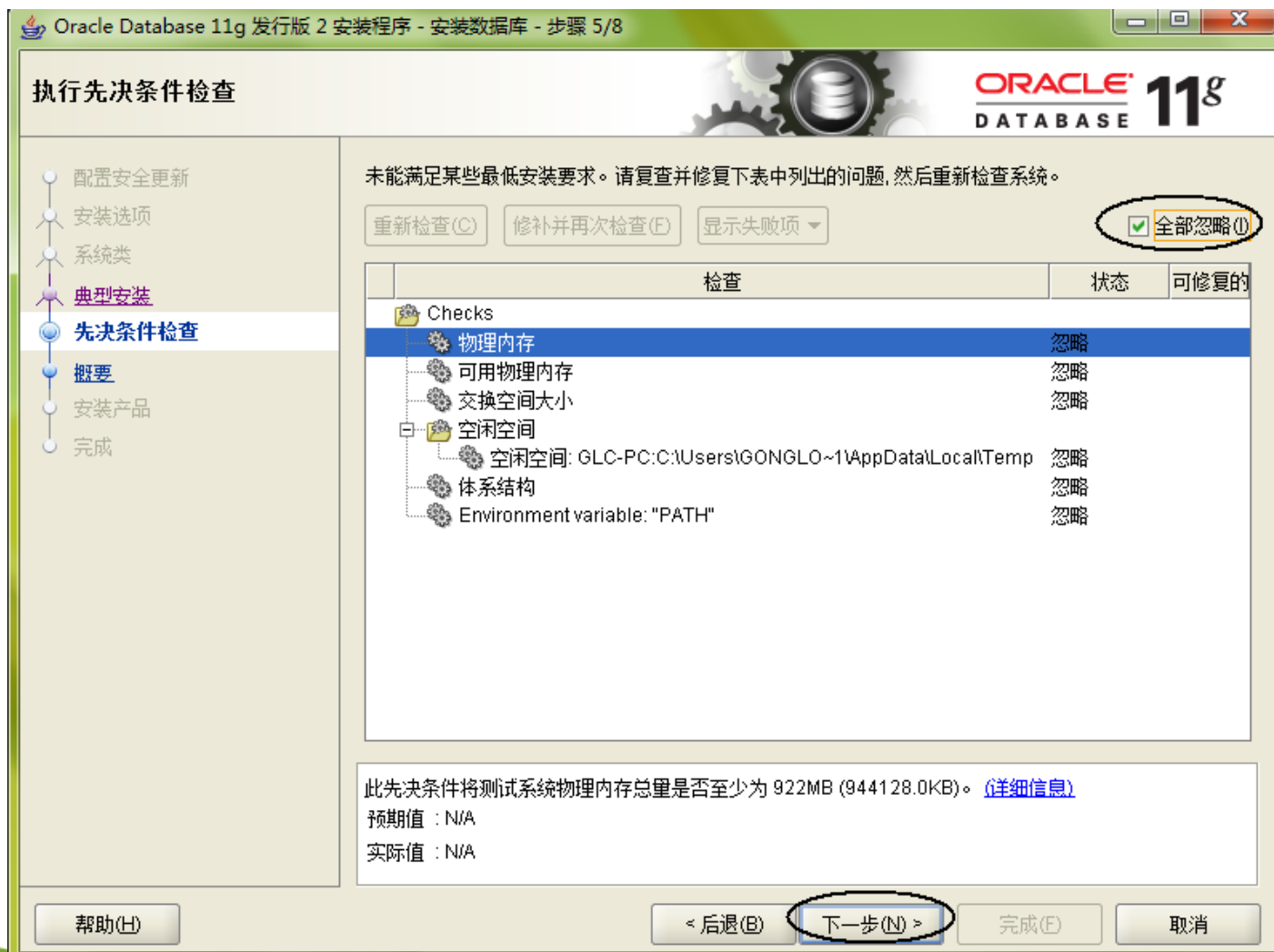
全局数据库名(S): orcl

管理口令(P): *****

确认口令(C): ? *****

帮助(H) < 后退(B) 下一步(N) > 完成(F) 取消

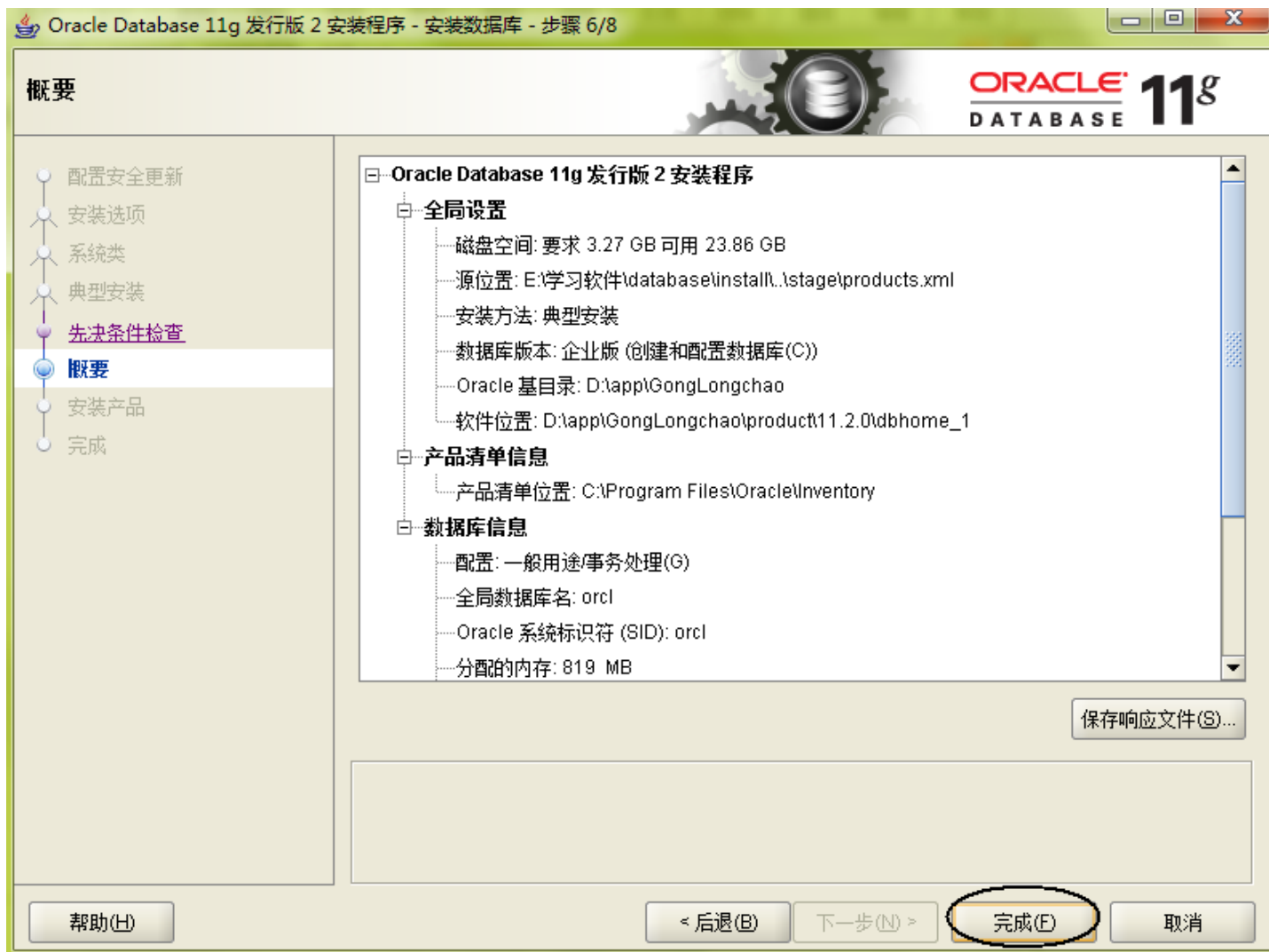
5. 之后会出现"先决条件检查"对话框，选中"全部忽略"并单击"下一步"以继续。



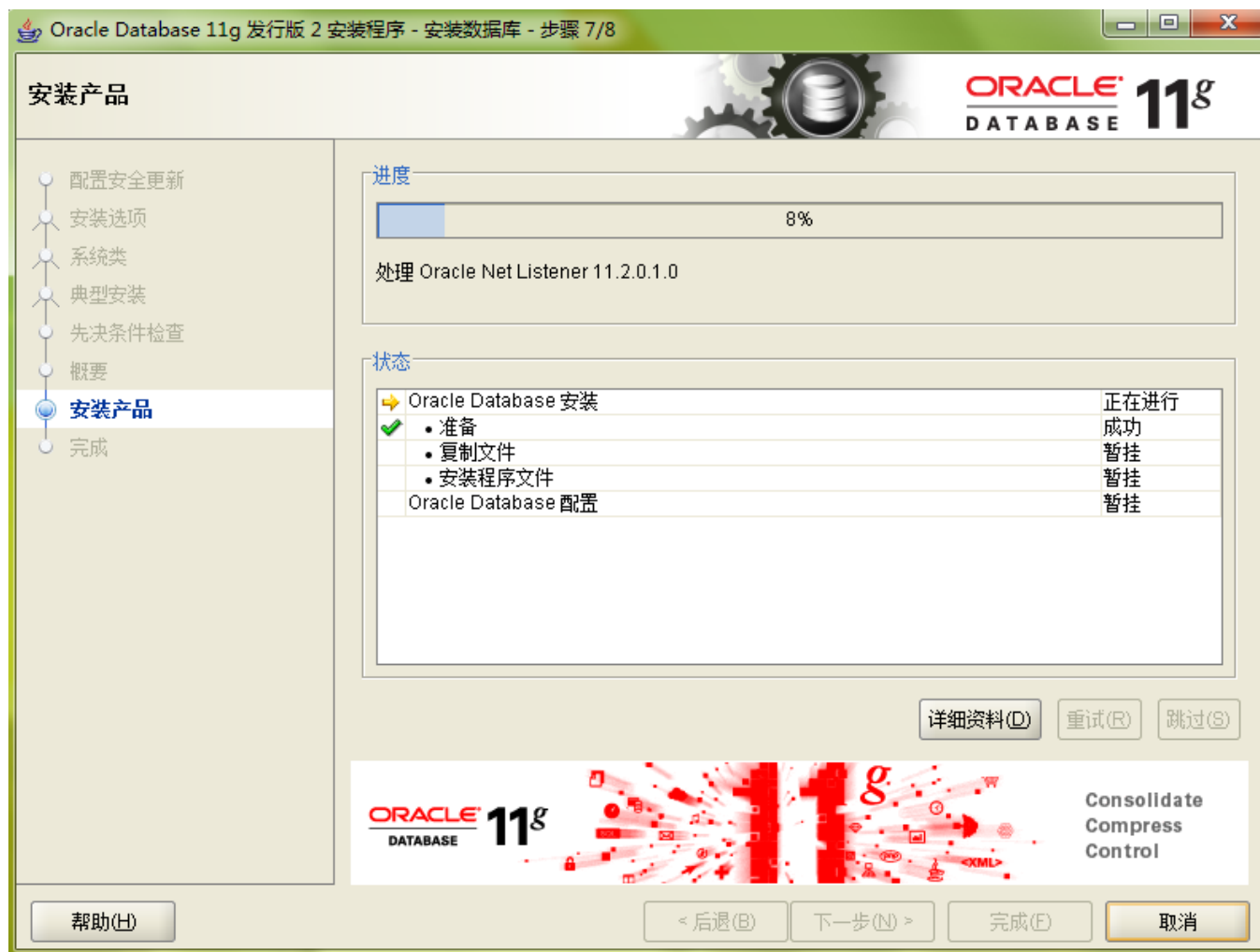
6. 之后会出现安装类型对话框，点击"下一步"继续，如果奔出你是安装在Windows Server上的话就选择服务器类。



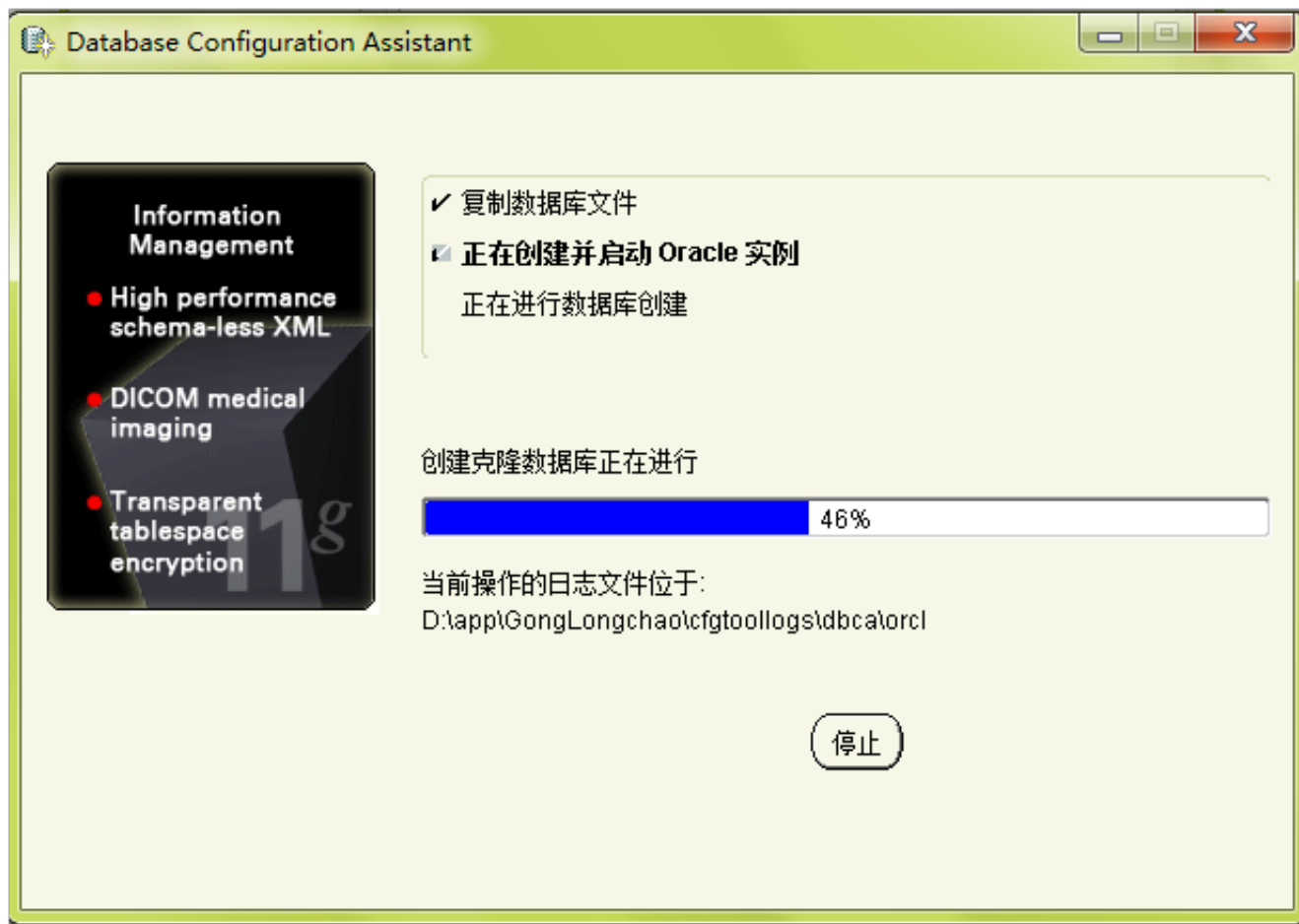
7. 之后点击"完成"就开始安装了



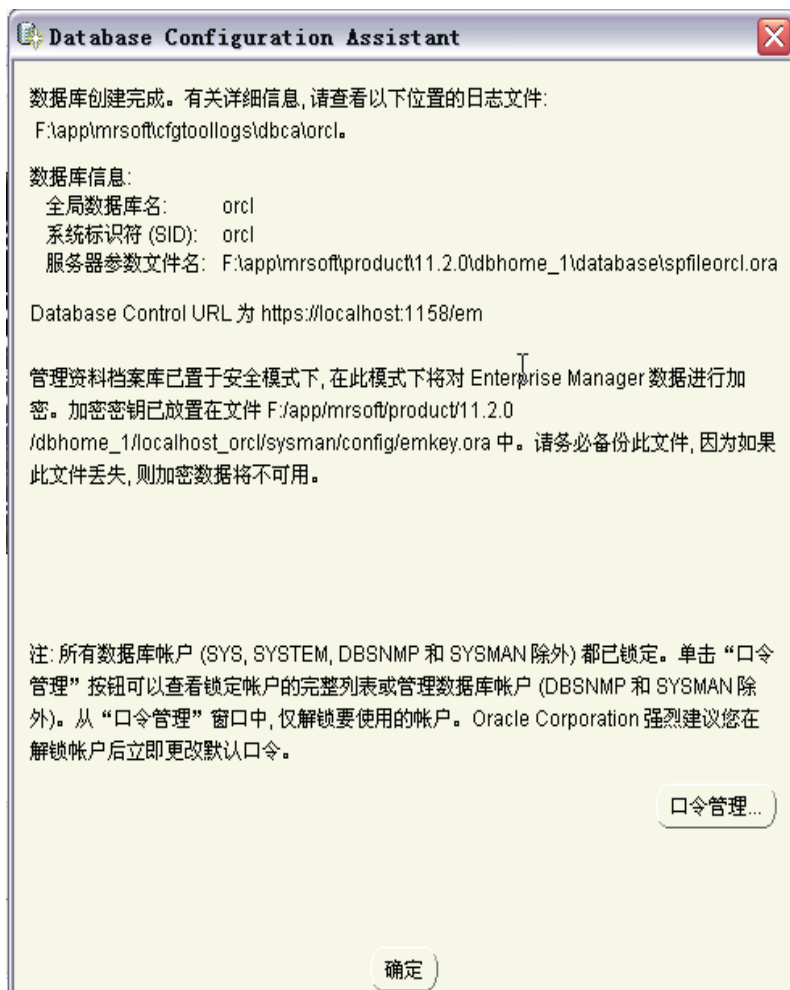
8. 安装画面如下图所示：



9. 当上述进度到达100%时会出现线面的对话框，请耐心等待它完成。



10. 出现该窗口时可以点击口令管理可以对数据库用户进行锁定/解锁以及更改默认口令。



11. 然后再弹出的确认对话框点击"确定"这时会回到主界面然后再单击"关闭"完成安装



- ✎ Oracle 服务器由Oracle 数据库和 Oracle 实例组成
- ✎ Oracle 实例由系统全局区内存结构和用于管理数据库的后台进程组成
- ✎ Oracle 中用于访问数据库的主要查询工具有 SQL*Plus、iSQL*Plus 和 PL/SQL
- ✎ Oracle 企业管理器是用于管理、诊断和调整多个数据库的工具
- ✎ Oracle 中的 SYSTEM 用户和 SYS 用户具有管理权限，而 SCOTT 用户只有基本的权限

Oracle客户端工具

SQL*Plus工具

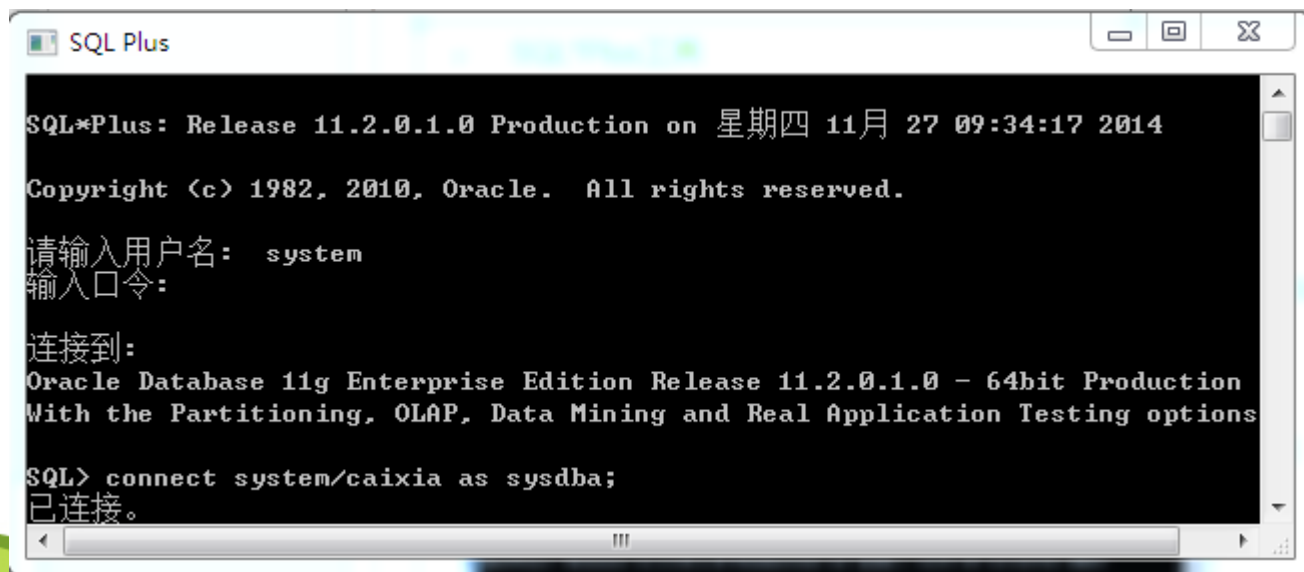
该工具是Oracle系统默认安装下，自带的一个客户端工具。

(1) 启动 sqlplus客户端

“开始” / “所有程序” / Oracle_OraDb11g_home1 / “应用程序开发/SQL*plus命令，启动了SQL*Plus工具。

(2) 连接到数据库

SQL>CONNECT 用户名/ 口令 AS [SYSOPER|SYSDBA|SYSASM]



```
SQL Plus

SQL*Plus: Release 11.2.0.1.0 Production on 星期四 11月 27 09:34:17 2014

Copyright (c) 1982, 2010, Oracle. All rights reserved.

请输入用户名: system
输入口令:

连接到:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> connect system/caixia as sysdba;
已连接。
```



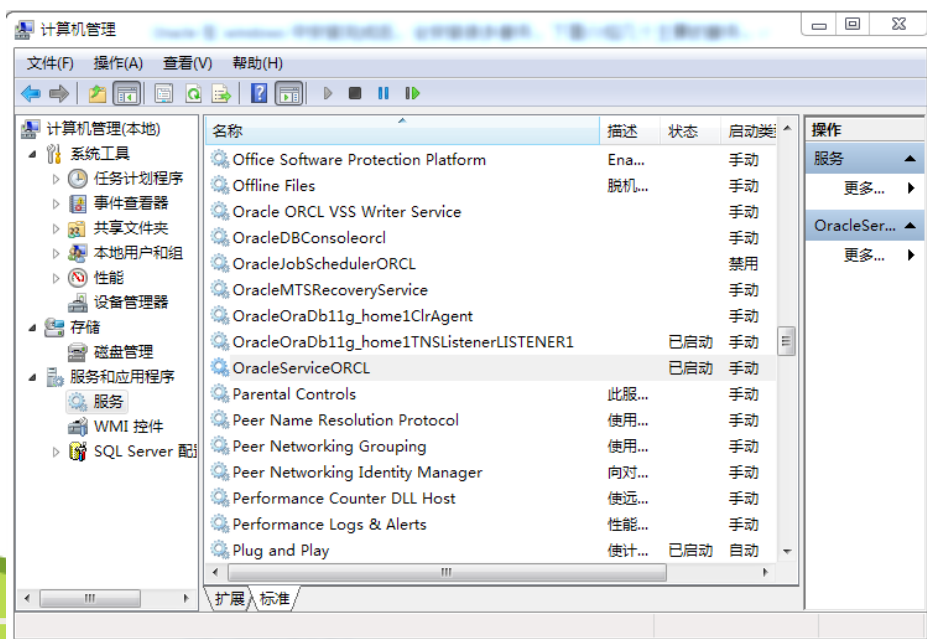
PL/SQL Developer

在实际Oracle开发中，经常使用一个功能强大的第三方工具：“[PL/SQL Developer](#)”工具。[PL/SQL Developer](#)基本上可以实现Oracle开发中的任何操作。需通过网络配置向导[配置网络服务](#)后才能正常与服务器连接。



Oracle服务

- ✎ OracleServiceORCL，该服务是数据库启动的基础，只有该服务启动了，Oracle数据库才能正常启动。这是必须启动的服务。
- ✎ OracleOraDb11g_home1TNSListener，该服务是服务器端为客户端提供的监听服务，只有该服务在服务器上正常启动，客户端才能连接到服务器。该监听服务接收客户端发出的请求，然后将请求传递给数据库服务器。一旦建立了连接，客户端和数据库服务器就能直接通信了。



SQL语言

PART 2



用户模式

模式是一个数据库对象的集合。模式为一个数据库用户所有,在一个模式内部不可以直接访问其他模式的数据库对象

SCOTT模式

Oracle提供的SCOTT模式的目的是,就是为了给用户提供一些示例表和数据来展示Oracle数据库的一些特性。SCOTT模式拥有的模式对象(都是数据表)如图所示。

EMP	
EMPNO	NUMBER
ENAME	VARCHAR2
JOB	VARCHAR2
MGR	NUMBER
HIREDATE	DATE
SAL	NUMBER
COMM	NUMBER
DEPTNO	NUMBER

DEPT	
EMPNO	NUMBER
DNAME	VARCHAR2
LOC	VARCHAR2

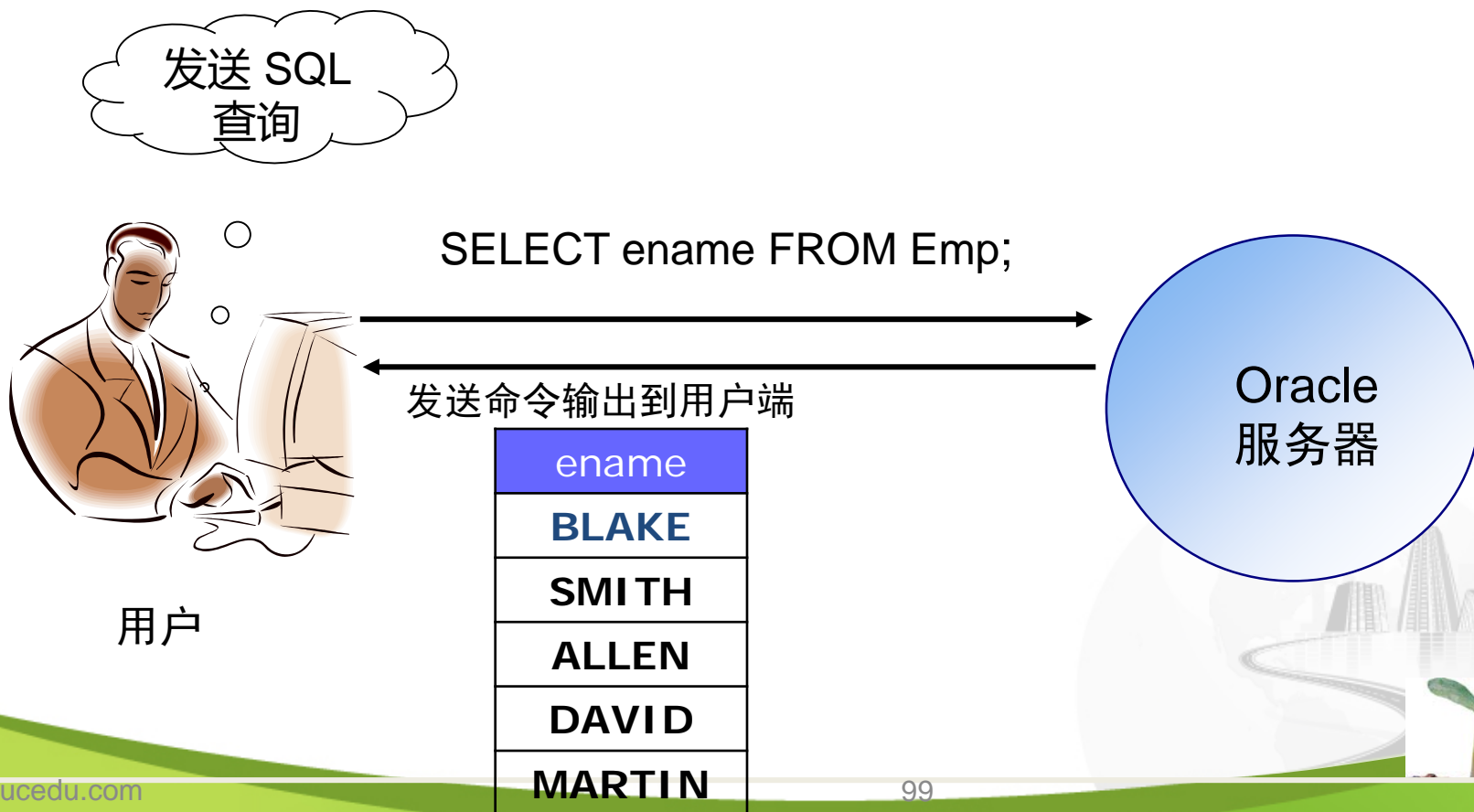
SALGRADE	
GRADE	NUMBER
LOSAL	NUMBER
HISAL	NUMBER

BONUS	
ENAME	VARCHAR2
JOB	VARCHAR2
SAL	NUMBER
COMM	NUMBER



❖ 通过 SQL 可以实现与 Oracle 服务器的通信

- ❖ SQL 是 Structured Query Language（结构化查询语言）的首字母缩写词
- ❖ SQL 是数据库语言，Oracle 使用该语言存储和检索信息
- ❖ 表是主要的数据库对象，用于存储数据



数据定义语言

数据定义语言用于改变数据库结构，包括创建、更改和删除数据库对象

数据表

表是**Oracle**数据库中主要的数据存储容器，表中的数据被组织成行和列。表中的每个列均有一个名称，并且每个列都具有一个指定的数据类型和大小，

◆ 创建表(CREAT)

格式

```
CREAT TABLE 表名 (  
    列名 数据类型 [default 缺省值] [not null]  
    [, 列名 数据类型 [default 缺省值] [not null]  
    .....  
    [, primary key ( 列名 [, 列名] ... ) ]  
    [, foreign key ( 列名 [, 列名] ... )  
        references 表名 ( 列名 [, 列名] ... ) ]  
    [, check ( 条件 ) ] )
```



基本数据类型

1. 数值类型

数值类型主要包括**NUMBER**、**PLS_INTEGER**和**BINARY_INTEGER**三种基本类型

2. 字符类型

字符类型主要包括**VARCHAR2**、**CHAR**、**LONG**、**NCHAR**和**NVARCHAR2**等。这些类型的变量用来存储字符串或字符数据。

3. 日期类型

日期类型只有一个种——即**DATE**类型，用来存储日期和时间信息，**DATE**类型的存储空间是7个字节，分别使用一个字节存储世纪、年、月、天、小时、分钟和秒。

4. 布尔类型

布尔类型也只有一种——即**BOOLEAN**，主要用于程序的流程控制和业务逻辑判断，其变量值可以是**TRUE**、**FALSE**或**NULL**中的一种。



【示例1】创建一个学生基本信息表**Student**，该表包括了学号、姓名、系别编号、班级编号、年龄、性别。

字段名	字段类型	Not Null	说明
Sno	Char	Primary key	学号
Sname	char	Not Null	学生姓名
Sdept	char		学院
Sclass	char		班级
Sage	number		年龄
Ssex	char		性别

```
CREATE TABLE student (  
    SNO CHAR(5) ,  
    SNAME CHAR(10) NOT NULL,  
    SDEPT CHAR(2) NOT NULL,  
    SCLASS CHAR(2) NOT NULL,  
    SAGE NUMBER(2),  
    SSEX CHAR(2),  
    CONSTRAINT SNO_PK PRIMARY KEY(SNO))
```

主键约束



图书管理信息表

字段	数据类型	数据长度	能否为空	中文名	备注
ISBN	Char	20	否	图书编码	主码
Title	Char	30	是	书名	
Pubname	Char	30	是	出版者名	
Author	Char	30	是	作者名	
Authorno	Number	30	是	作者号	
ZNCode	Number	30	是	中南分校码	

创建Books表

```
create table Books  
(isbn char(20) not null primary key,  
title char(30),  
pubname char(30),  
author char(30),  
authorno number(30),  
zncode number(30));
```



◆ 查看表(DESCRIBE)

格式: *DESCRIBE* 表名

示例 *describe student;*

◆ 修改表(ALTER)

格式: **ALTER TABLE** 表名

[**add** 子句]

增加新列

[**drop** 子句]

删除列

[**modify** 子句]

修改列定义示例

示例 *alter table student add(province varchar2(10));*

alter table student drop(province);

◆ 删除表 (DROP)

格式: *DROP TABLE* 表名

示例 *drop table student cascade constraints;*



数据操作语言

查询数据

PL/SQL的数据查询是一种从数据库中检索符合搜索条件的记录生成数据记录集合，并把它们存入数据记录集对象中的操作。

语法格式

```
SELECT          <选择表达式>
FROM            [用户方案.]<表或视图>
[ WHERE         <查询条件> ]
[ GROUP BY     <分组表达式>
[ HAVING       <分组统计条件> ] ]
[ ORDER BY     <排序表达式> ]
```



简单查询

只包含SELECT子句和FROM子句的查询就是简单查询，SELECT子句和FROM子句是SELECT语句的必选项，也就是说每个SELECT语句都必须包含这两个子句。

【示例】需要了解学生的所有信息，代码如下

```
SELECT * FROM Student;
```

【示例】学生表中，查找出所有的列并且使用 AS关键字为这些列指定中文的别名，SNO（学号）、姓名(SNAME)、系别编号（SDEPT）、班级编号(SCLASS)、年龄(SAGE)、性别（SSEX）代码如下。

```
SELECT SNO AS “学号”、SNAME AS “姓名”、SDEPT AS “系别编号”、  
SCLASS AS “班级编号”、SAGE AS “年龄”，SSEX AS “性别” FROM student;
```



筛选查询

在SELECT语句中使用WHERE子句可以实现对数据行的筛选操作，只有满足WHERE判断条件的行才会显示在结果集中。在SELECT 语句中,WHERE子句位于FROM子句之后，

格式

```
SELECT columns_list  
FROM table_name  
WHERE conditional_expression  
columns_list:字段列表 table_name : 表名
```

【示例】学生表中，查询出年龄<20岁的学生信息，并且按年龄排序。

```
select*from student where sage<20 order by sage;
```



插入数据

插入数据就是将数据记录添加到已经存在的数据表中

格式

```
INSERT INTO <table_name> [column_list1(,column_name2)]  
VALUES(express1[,express2]....)
```

table_name:要插入的表名

column_name:指定表的完全或部分列名称

express: 要插入的值列表



【示例】 向student表中插入一下学生信息：

```
INSERT INTO Student VALUES('96001', '马小燕', 'CS', '01', 21, '女');  
INSERT INTO Student VALUES('96002', '黎明', 'CS', '01', 18, '男');  
INSERT INTO Student VALUES('96003', '刘东明', 'MA', '01', 18, '男');  
INSERT INTO Student VALUES('96004', '赵志勇', 'IS', '02', 20, '男');  
INSERT INTO Student VALUES('97001', '马蓉', 'MA', '02', 19, '女');  
INSERT INTO Student VALUES('97002', '李成功', 'CS', '01', 20, '男');  
INSERT INTO Student VALUES('97003', '黎明', 'IS', '03', 19, '女');  
INSERT INTO Student VALUES('97004', '李丽', 'CS', '02', 19, '女');  
INSERT INTO Student VALUES('96005', '司马志明', 'CS', '02', 18, '男');
```



更新数据

如果表中的数据不正确或不符合需求时，就可以对其进行修改。更新数据时，更改的列数可以由用户自己指定，更新的条数可以通过WHERE子句加以限制。

UPDATE table_name

SET{column_name1=express[,column_name2=express2...]}

[WHERE condition]

【示例】将student表中的男生年龄增加1岁

update student

set sage=sage+1

where ssex= '男'



删除数据

Oracle系统提供了向数据库添加记录的功能，也提供了从数据库删除数据的功能，可以使用DELETE语句进行删除。

格式

```
DELETE FROM table_name
```

```
[WHERE condition]
```

【示例】从student表中删除年龄是18岁的记录

```
delete from student where sage= '18'
```



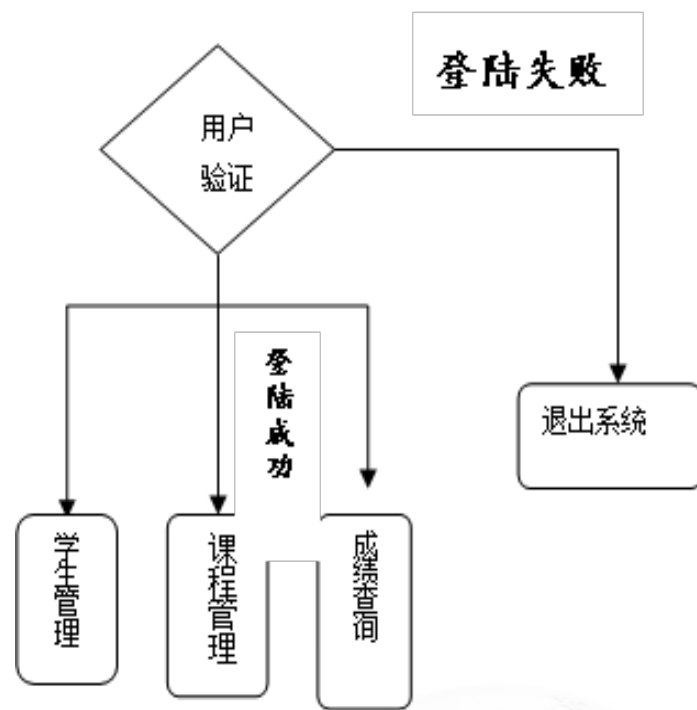
学生选课成绩管理系统

PART 3



系统功能

- ✎ 使得学生的成绩管理工作更加清晰、条理化、自动化。
- ✎ 通过用户名和密码登录系统，查询课程基本资料，学生所选课程成绩，修改用户密码等功能，容易地完成学生信息的查询操作。
- ✎ 设计人机友好界面，功能安排合理，操作使用方便，并且进一步考虑系统在安全性，完整性，并发控制，备份和恢复等方面的功能要求。



系统界面设计

系统登录

该部分主要实现登陆限制功能及退出系统。

学生信息

该部分主要实现四个功能：查询学生信息，添加学生信息，修改学生信息，删除学生信息。

教师信息

该部分主要实现四个功能：查询教师信息，添加教师信息，修改教师信息，删除教师信息。

课程信息

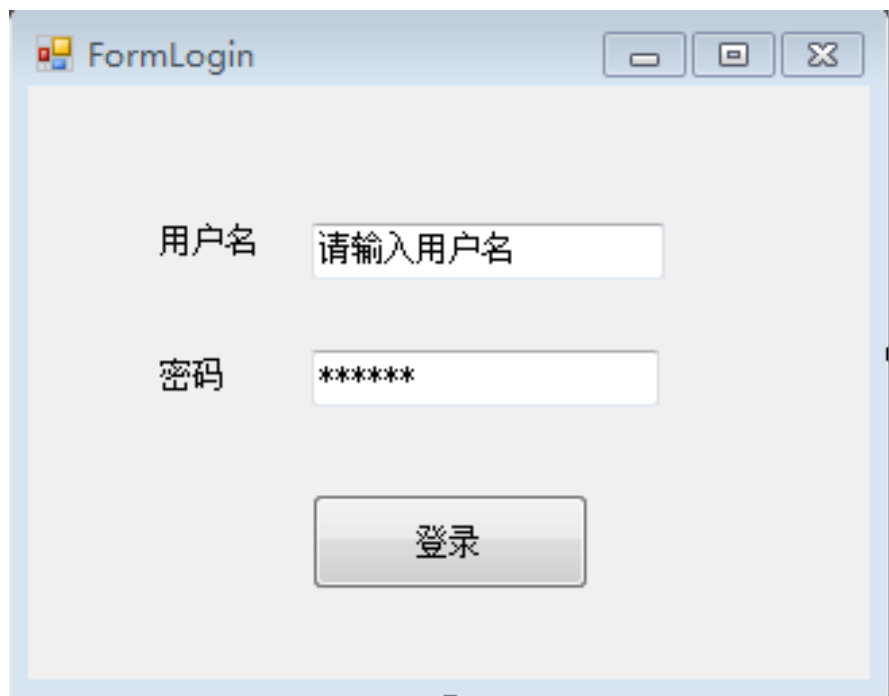
该部分主要实现四个功能：查询课程信息，添加课程信息，修改课程信息，删除课程信息。

学生成绩

该部分主要实现两大部分功能：表信息检索，按课程检索，按个人检索，按班级检索



登入界面——

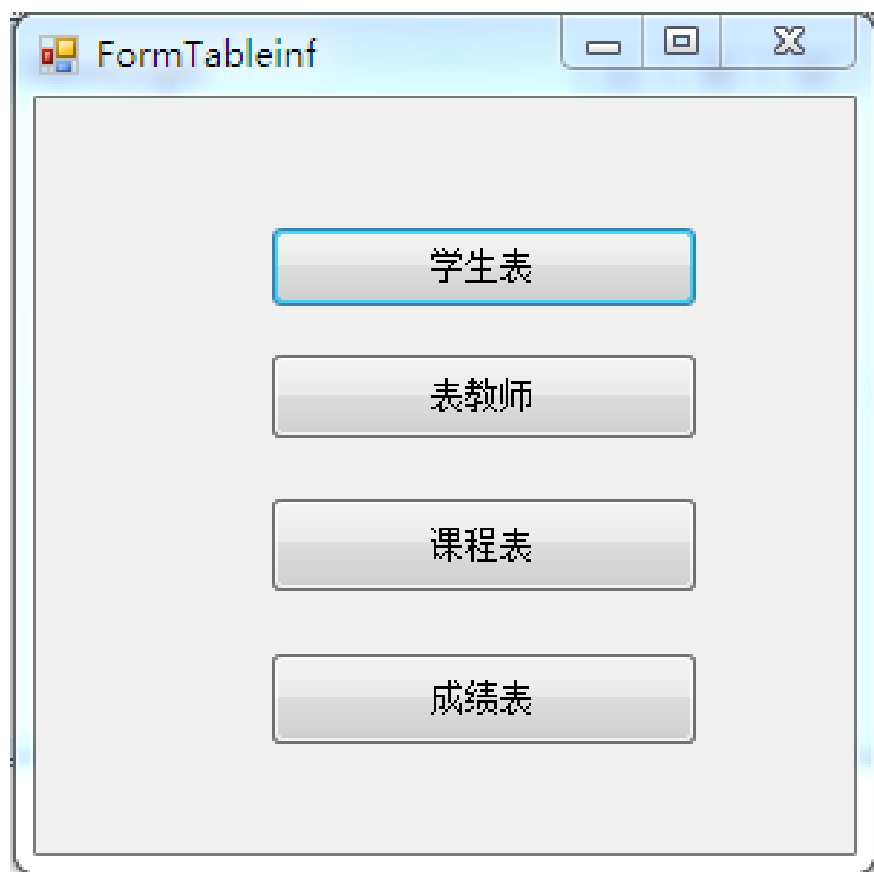


A screenshot of a Windows-style application window titled "FormLogin". The window has a light gray background and a blue title bar with standard minimize, maximize, and close buttons. Inside the window, there are two input fields: the first is labeled "用户名" (Username) and contains the placeholder text "请输入用户名"; the second is labeled "密码" (Password) and contains seven asterisks "*****". Below these fields is a single button labeled "登录" (Login).

通过左图所示界面可以登录系统，只有输入了正确的用户名和密码才能进入系统，否则提示“请输入正确的用户名和密码！”



学生选课成绩管理系统——



学生信息管理——

	SNO	SNAME	SDEPT	SCLASS	SAGE	SSEX
▶	96001	马小燕	CS	01	21	女
	96002	黎明	CS	01	18	男
	96003	刘东明	MA	01	18	男
	96004	赵志勇	IS	02	20	男
	97001	马蓉	MA	02	19	女
	97002	李成功	CS	01	20	男
	97003	黎明	IS	03	19	男
	97004	李丽	CS	02	19	女
	96005	司马志明	CS	02	18	男
*						

此界面可以实现学生信息的查询、修改、添加、删除等操作

教师信息管理——

	SNO	SNAME	SDEPT	SCLASS	SAGE	SSEX
▶	96001	马小燕	CS	01	21	女
	96002	黎明	CS	01	18	男
	96003	刘东明	MA	01	18	男
	96004	赵志勇	IS	02	20	男
	97001	马蓉	MA	02	19	女
	97002	李成功	CS	01	20	男
	97003	黎明	IS	03	19	男
	97004	李丽	CS	02	19	女
	96005	司马志明	CS	02	18	男
*						

此界面可以实现学生信息的查询、修改、添加、删除等操作

学生成绩检索 ——

学生成绩检索

表信息检索 请选择字段名 ▼ 请选择比较符 ▼ 检索

按课程检索 要检索的目标 ▼ 001 ▼ 检索 选课总数

按个人检索 要检索的目标 ▼ 学生号 ▼ 检索 选课总数

按班级检索: 要检索的目标 ▼ 学院 ▼ 班级 ▼ 检索 选课总数

	SCORE	MAX (SCORE)	MIN (SCORE)	AVG (SCORE)	COUNT (*)
▶					5
*					

此界面可以按表信息、课程信息、个人信息、班级信息等对学生成绩进行查询

谢谢观赏

Thank You

