

第1部分 实用教程

第1章 数据库基础

Oracle 是甲骨文软件系统有限公司（以下简称甲骨文公司）开发的数据库管理系统，Oracle 11g 是目前最流行的 Oracle 数据库管理系统版本之一。在介绍 Oracle 数据库之前，首先介绍数据库基本概念。

1.1 数据库基本概念

1.1.1 数据库系统



1. 数据库

数据库（DB）是存放数据的仓库，这些数据是相关联并按一定的格式存放在计算机内的。例如，把一个学校的学生、课程、成绩等数据有序地组织并存放在计算机内，就可以构成一个数据库。

2. 数据库管理系统

数据库管理系统（DBMS）是按一定的数据模型组织数据形成数据库，并对数据库进行管理的系统。简单地说，DBMS 就是管理数据库的系统（软件）。数据库系统管理员（DataBase Administrator, DBA）通过 DBMS 对数据库进行管理。

目前比较流行的 DBMS 有：

Oracle、SQL Server、MySQL、Access 等。其中，Oracle 是大型关系数据库管理系统，本书介绍的是当前应用最广泛的 Oracle 11g 版。

3. 数据库系统

数据、数据库、数据库管理系统与操作数据库的应用程序，加上支撑它们的硬件平台、软件平台和与数据库有关的人员一起构成了一个完整的数据库系统。如图 1.1 所示为数据库系统的构成。

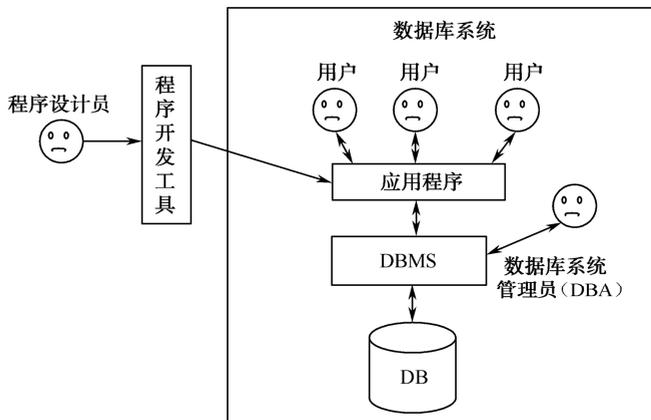


图 1.1 数据库系统的构成

1.1.2 数据模型

数据库管理系统根据数据模型对数据进行存储和管理，数据模型主要有层次模型、网状模型和关系模型。随着信息管理内容的不断扩展和新技术的层出不穷，数据库技术面临着前所未有的挑战，面对新的数据形式，人们提出了丰富的数据模型，如面向对象模型、半结构化模型等。

1. 层次模型

层次模型将数据组织成一对多关系的结构，可采用关键字来访问其中每个层次的每个部分。它的优点：存取方便且速度快；结构清晰，容易理解；数据修改和数据库扩展容易实现；检索关键属性十分方便。它的缺点：结构不够灵活；同一属性数据要存储多次，数据冗余大；不适合于拓扑空间数据的组织。

如图 1.2 所示为按层次模型组织的数据示例。

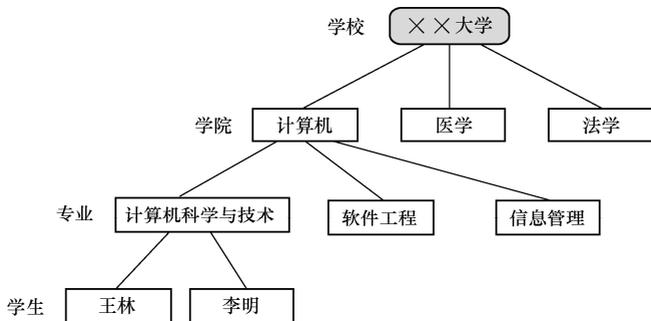


图 1.2 按层次模型组织的数据示例

2. 网状模型

网状模型具有多对多类型的数据组织方式。它的优点：能明确而方便地表示数据间的复杂关系；数据冗余小。它的缺点：网状结构的复杂性增加了用户查询和定位的困难；需要存储数据间联系的指针，使得数据量增大；数据的修改不方便。

如图 1.3 所示为按网状模型组织的数据示例。

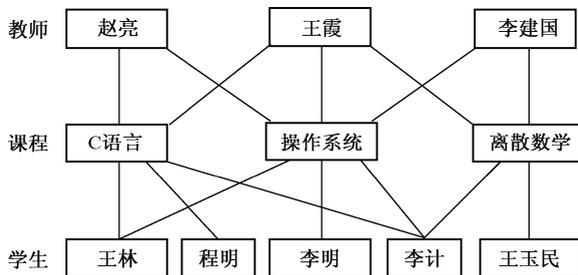


图 1.3 按网状模型组织的数据示例

3. 关系模型

关系模型以记录组或二维数据表的形式组织数据，以便于利用各种实体与属性之间的关系进行存储和变换，不分层也无指针，是建立空间数据和属性数据之间关系的一种非常有效的数据组织方法。它的优点：结构特别灵活，概念单一，能满足所有布尔逻辑运算和数学运算规则形成的查询要求；能

搜索、组合和比较不同类型的数据；增加和删除数据的操作非常方便；具有更高的数据独立性、更好的安全保密性。它的缺点：当数据库大时，查找满足特定关系的数据会很费时，且无法表达空间关系。

例如，在学生成绩管理系统所涉及的“学生”表、“课程”表和“成绩”表中，“学生”表的主要信息有学号、姓名、性别、出生时间、专业、总学分、备注；“课程”表的主要信息有课程号、课程名、开课学期、学时和学分；“成绩”表的主要信息有学号、课程号和成绩。如表 1.1~表 1.3 所示分别描述了学生成绩管理系统中“学生”表、“课程”表和“成绩”表的部分数据。

表 1.1 “学生”表

学号	姓名	性别	出生时间	专业	总学分	备注
151101	王林	男	1997-02-10	计算机	50	
151103	王燕	女	1996-10-06	计算机	50	
151108	林一帆	男	1996-08-05	计算机	52	已提前修完一门课
151202	王林	男	1996-01-29	通信工程	40	有一门课不及格，待补考
151204	马琳琳	女	1996-02-10	通信工程	42	

表 1.2 “课程”表

课程号	课程名	开课学期	学时	学分
0101	计算机基础	1	80	5
0102	程序设计与语言	2	68	4
0206	离散数学	4	68	4

表 1.3 “成绩”表

学号	课程号	成绩	学号	课程号	成绩
151101	101	80	151108	101	85
151101	102	78	151108	102	64
151101	206	76	151108	206	87
151103	101	62	151202	101	65
151103	102	70	151204	101	91

表格中的一行称为一个记录，一列称为一个字段，每列的标题称为字段名。如果给每个关系表取一个名字，则有 n 个字段的关系表结构可表示为：关系表名（字段名 1，…，字段名 n ），通常把关系表的结构称为关系模式。

在关系表中，如果一个字段或几个字段组合的值可唯一标识其对应记录，则称该字段或字段组合为码。

例如，表 1.1 中的“学号”可唯一标识每一个学生，表 1.2 中的“课程号”可唯一标识每一门课程。表 1.3 中的“学号”和“课程号”可唯一标识每一个学生一门课程的成绩。

有时，一个表可能有多个码，如表 1.1 中，姓名不允许重名，则“学号”、“姓名”均是学生信息表码。对于每一个关系表，通常可指定一个码为“主码”，在关系模式中，一般用下画线标出主码。

设表 1.1 的名字为 xsb，关系模式可表示为：xsb（学号，姓名，性别，出生时间，专业，总学分，备注）。

设表 1.2 的名字为 kcb，关系模式可表示为：kcb（课程号，课程名，开课学期，学时，学分）。

设表 1.3 的名字为 cjb，关系模式可表示为：cjb (学号, 课程号, 成绩, 学分)。

通过上面的分析可以看出，关系模型更适合组织数据，所以使用最广泛。Oracle 是流行的大型关系数据库管理系统。

关系数据库分为两类：桌面数据库和客户-服务器数据库。

一般而言，桌面数据库用于小型的、单机的应用程序，它不需要网络和服务器，实现起来比较方便，但它只提供数据的存取功能。如 Access、FoxPro 和 Excel 等。

客户-服务器数据库主要适用于大型的、多用户的数据库管理系统，包括两部分：一部分驻留在客户机上，用于向用户显示信息及实现与用户的交互；另一部分驻留在服务器中，主要用来实现对数据库的操作和对数据的计算处理。在开发数据库应用程序时，也可以将其放在一台计算机上进行调试，调试完成后再把数据库放到服务器上。

大型关系数据库管理系统一般为 Oracle、SQL Server、DB2、Ingers、Informix 和 Sybase 等。小型关系数据库管理系统一般为 MySQL 和 SQLite。其中 SQLite 是一个强大的嵌入式关系数据库管理系统；MySQL 是流行的 RDBMS，目前 Oracle 11g 仍然是使用最多的数据库版本。

1.1.3 关系数据库语言

结构化查询语言 (Structured Query Language, SQL) 是用于关系数据库查询的结构化语言。它的功能包括数据定义语言、数据操纵语言、数据控制语言和数据查询语言。

(1) 数据定义语言 (DDL)。DDL 用于执行数据库的任务，对数据库及数据库中的各种对象进行创建、删除、修改等操作。如前所述，数据库对象主要包括表、默认约束、规则、视图、触发器、存储过程。DDL 包括的主要语句及功能如表 1.4 所示。

表 1.4 DDL 包括的主要语句及功能

语 句	功 能
CREATE	创建数据库或数据库对象
ALTER	对数据库或数据库对象进行修改
DROP	删除数据库或数据库对象

(2) 数据操纵语言 (DML)。DML 用于操纵数据库中的各种对象，以及检索和修改数据。DML 包括的主要语句及功能如表 1.5 所示。

表 1.5 DML 包括的主要语句及功能

语 句	功 能
SELECT	从表或视图中检索数据
INSERT	将数据插入到表或视图中
UPDATE	修改表或视图中的数据
DELETE	从表或视图中删除数据

(3) 数据控制语言 (DCL)。DCL 用于安全管理，可确定哪些用户可以查看或修改数据库中的数据。DCL 包括的主要语句及功能如表 1.6 所示。

表 1.6 DCL 包括的主要语句及功能

语 句	功 能
GRANT	授予权限
REVOKE	收回权限
DENY	收回权限, 并禁止从其他角色继承许可权限

(4) 数据查询语言 (DQL)。用户通过 SELECT 语言实现各种查询功能。

许多关系数据库管理系统均支持 SQL 语言, 如 Oracle、SQL Server、MySQL 等。但不同数据库管理系统之间的 SQL 语言则不能完全通用。例如, 甲骨文公司的 Oracle 数据库所使用的 SQL 语言是 Procedural Language/SQL (PL/SQL), 而微软公司的 SQL Server 数据库系统支持的则是 Transact-SQL (T-SQL)。PL/SQL 是 ANSI SQL 的扩展加强版 SQL 语言, 除了提供标准的 SQL 命令, 还对 SQL 进行了许多补充。



1.2 数据库设计

数据模型按不同的应用层次分成三种类型: 概念数据模型、逻辑数据模型、物理数据模型。

1.2.1 概念数据模型

概念数据模型 (Conceptual Data Model) 主要用来描述信息世界的概念化结构, 它能使数据库的设计人员在设计的初始阶段, 摆脱计算机系统及 DBMS 的具体技术问题, 集中精力分析数据及数据之间的联系, 与具体的数据管理系统无关。概念数据模型必须换成逻辑数据模型, 才能在 DBMS 中实现。

概念数据模型的建模: 一方面, 具有较强的语义表达能力, 能够方便直接表达应用中的各种语义知识; 另一方面, 简单、清晰、易于用户理解。在概念数据模型中最常用的是 E-R 模型、扩充的 E-R 模型、面向对象模型及谓词模型。

通常, E-R 模型把每类数据对象的个体称为“实体”, 而每类对象个体的集合称为“实体集”, 例如, 在学生成绩管理系统中主要有“学生”和“课程”两个实体集。而其他非主要的实体可以有很多, 如班级、班长、任课教师、辅导员等。

把每个实体集涉及的信息项称为属性, 就“学生”实体集而言, 它的属性有: 学号、姓名、性别、出生时间、专业、总学分和备注。“课程”实体集属性有: 课程号、课程名、开课学期、学时和学分。

实体集中的实体彼此是可区别的。如果实体集中的属性或最小属性组合的值能唯一标识其对应实体, 则将该属性或属性组合称为码。码可能有多个, 对于每一个实体集, 可指定一个码为主码。

如果用矩形框表示实体集, 用带半圆的矩形框表示属性, 用线段连接实体集与属性, 当一个属性或属性组合指定为主码时, 在实体集与属性的连接线上标记一斜线, 则可以用如图 1.4 所示的形式描述学生成绩管理系统中的实体集及每个实体集涉及的属性。

实体集 A 和实体集 B 之间存在各种关系, 通常把这些关系称为“联系”。将实体集及实体集联系的图表示称为实体 (Entity)-联系 (Relationship) 模型, 即 E-R 模型。

E-R 图就是 E-R 模型的描述方法, 即实体-联系图。通常, 关系数据库的设计者使用 E-R 图来对信息世界建模。在 E-R 图中, 用矩形表示实体型, 用带半圆的矩形框表示属性, 用菱形表示联系。从分析用户项目涉及的数据对象及数据对象之间的联系出发, 到获取 E-R 图的这个过程称为概念结构设计。

两个实体集 A 和 B 之间的联系可能是以下三种情况之一。

1. 一对一的联系 (1:1)

实体 A 中的一个实体最多与实体 B 中的一个实体相联系，实体 B 中的一个实体也最多与实体 A 中的一个实体相联系。例如，“班级”与“班长”这两个实体集之间是一对一的联系，因为一个班级只有一个班长，反过来，一个班长只属于一个班级。“班级”与“班长”两个实体集的 E-R 图如图 1.5 所示。

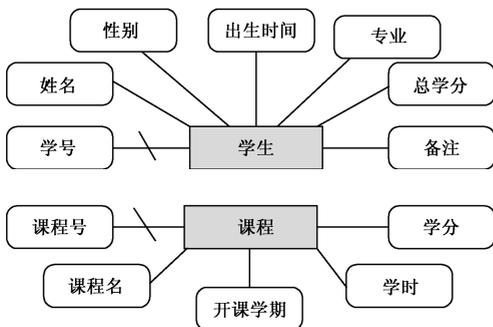


图 1.4 “学生”和“课程”实体集属性的描述

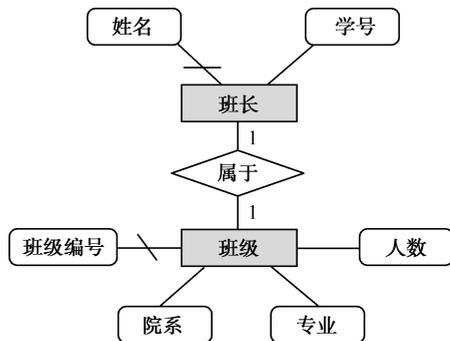


图 1.5 “班级”与“班长”两个实体集的 E-R 图

2. 一对多的联系 (1:n)

实体 A 中的一个实体可以与实体 B 中的多个实体相联系，而实体 B 中的一个实体最多与实体 A 中的一个实体相联系。例如，“班级”与“学生”这两个实体集之间是一对多的联系，因为一个班级可有若干个学生，反过来，一个学生只能属于一个班级。“班级”与“学生”两个实体集的 E-R 图如图 1.6 所示。

3. 多对多的联系 (m:n)

实体 A 中的一个实体可以与实体 B 中的多个实体相联系，而实体 B 中的一个实体也可与实体 A 中的多个实体相联系。例如，“学生”与“课程”这两个实体集之间是多对多的联系，因为一个学生可选多门课程，反过来，一门课程可被多个学生选修。“学生”与“课程”两个实体集的 E-R 图如图 1.7 所示。

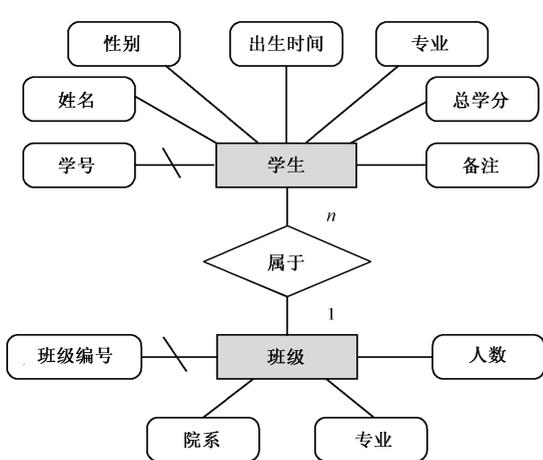


图 1.6 “班级”与“学生”两个实体集的 E-R 图

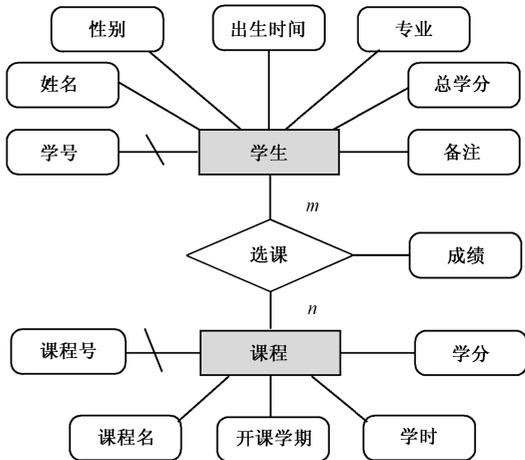


图 1.7 “学生”与“课程”两个实体集的 E-R 图

1.2.2 逻辑数据模型

逻辑数据模型 (Logical Data Model) 是用户从数据库所看到的模型, 是具体的数据库管理系统 (DBMS) 所支持的数据模型。此模型既要面向用户, 又要面向系统, 主要用于数据库管理系统的实现。

前面用 E-R 图描述学生成绩管理系统中实体集与实体集之间的联系, 为了设计关系型的学生成绩管理数据库, 需要确定包含哪些表呢? 每个表的结构又是怎样的呢?

前面已介绍了实体集之间的联系, 下面将根据三种联系从 E-R 图获得关系模式的方法。

1. (1:1) 联系的 E-R 图到关系模式的转换

对于 (1:1) 的联系, 既可以单独对应一个关系模式, 也可以不单独对应一个关系模式。

(1) 联系单独对应一个关系模式, 则由联系属性、参与联系的各实体集的主码属性构成关系模式, 其主码可选参与联系实体集的任一方主码。

例如, 考虑图 1.5 描述的“班级”(bjb) 与“班长”(bzb) 实体集通过属于 (syb) 联系 E-R 模型, 可设计如下关系模式 (下画线表示该字段为主码):

bjb (班级编号, 院系, 专业, 人数)

bzb (学号, 姓名)

syb (学号, 班级编号)

(2) 联系不单独对应一个关系模式, 将联系的属性及一方的主码加入另一方实体集对应的关系模式中。

例如, 考虑图 1.5 描述的“班级”(bjb) 与“班长”(bzb) 实体集通过属于 (syb) 联系 E-R 模型, 可设计如下关系模式:

bjb (班级编号, 院系, 专业, 人数)

bzb (学号, 姓名, 班级编号)

或者:

bjb (班级编号, 院系, 专业, 人数, 学号)

bzb (学号, 姓名)

2. (1:n) 联系的 E-R 图到关系模式的转换

对于 (1:n) 的联系, 既可以单独对应一个关系模式, 也可以不单独对应一个关系模式。

(1) 联系单独对应一个关系模式, 则由联系的属性、参与联系的各实体集的主码属性构成关系模式, n 端的主码作为该关系模式的主码。

例如, 图 1.6 描述的“班级”(bjb) 与“学生”(xsb) 实体集 E-R 模型可设计如下关系模式:

bjb (班级编号, 院系, 专业, 人数)

xsb (学号, 姓名, 性别, 出生时间, 专业, 总学分, 备注)

syb (学号, 班级编号)

(2) 联系不单独对应一个关系模式, 则将联系的属性及 1 端的主码加入 n 端实体集对应的关系模式中, 主码仍为 n 端的主码。

例如, 图 1.6 描述的“班级”(bjb) 与“学生”(xsb) 实体集 E-R 模型可设计如下关系模式:

bjb (班级编号, 院系, 专业, 人数)

xsb (学号, 姓名, 性别, 出生时间, 专业, 总学分, 备注, 班级编号)

3. ($m:n$) 联系的 E-R 图到关系模式的转换

对于 ($m:n$) 的联系，单独对应一个关系模式，该关系模式包括联系的属性、参与联系的各实体集的主码属性，该关系模式的主码由各实体集的主码属性共同组成。

例如，图 1.7 描述的“学生”(xsb)与“课程”(kcb)实体集之间的联系可设计如下关系模式：

xsb (学号, 姓名, 性别, 出生时间, 专业, 总学分, 备注)

kcb (课程号, 课程名称, 开课学期, 学时, 学分)

cjb (学号, 课程号, 成绩)

关系模式 cjb 的主码是由“学号”和“课程号”两个属性组合起来构成的，一个关系模式只能有一个主码。

至此，已介绍了根据 E-R 图设计关系模式的方法。通常，将这个设计过程称为逻辑结构设计。

在设计完成一个项目的关系模式后，就可以在数据库管理系统环境下创建数据库、关系表及其他数据库对象，输入相应数据，并根据需要对数据库中的数据进行各种操作。

1.2.3 物理数据模型

物理数据模型 (Physical Data Model) 是面向计算机物理表示的模型，描述了数据在储存介质上的组织结构，它不但与具体的 DBMS 有关，而且还与操作系统和硬件有关。每一种逻辑数据模型在实现时都有其对应的物理数据模型。DBMS 为了保证其独立性与可移植性，大部分物理数据模型的实现工作由系统自动完成，而设计者只要设计索引、聚集等特殊结构即可。