

第3章

数据库基础

主要内容



[返回总目录](#)

- └ 基 本 概 念
- └ 关系数据库
- └ 常见的数据库对象
- └ 范 式



数据库技术产生于二十世纪六十年代末，它作为数据管理最有效的手段，在各行各业中得到越来越广泛的应用。可以说：任何一个行业的信息化、现代化都离不开数据库。本章将介绍数据库的一些基础知识。

3.1 基本概念

3.1.1 数据（Data）

所谓数据就是描述事物的符号。在我们的日常生活中，数据无所不在，数字、文字、图表、图象、声音等都是数据。人们通过数据来认识世界，交流信息。

3.1.2 数据库（DB 即 Database）

数据库，顾名思义，就是数据存放的地方。在计算机中，数据库是数据和数据库对象的集合。所谓数据库对象是指表（Table）、视图（View）、存储过程（Stored Procedure）、触发器（Trigger）等。这些数据库对象将在以后的章节中介绍。

3.1.3 数据库管理系统（DBMS）

数据库管理系统是用于管理数据的计算机软件。数据库管理系统使用户能方便地定义和操纵数据，维护数据的安全性和完整性，以及进行多用户下的并发控制和恢复数据库。

3.1.4 数据库系统（DBS 即 Database System）

数据库系统狭义地讲是由数据库、数据库管理系统和用户构成，广义地讲是由计算机硬件、操作系统、数据库管理系统以及在它支持下建立起来的数据库、应用程序、用户和维护人员组成的一个整体。

3.2 关系数据库

数据库这一概念提出后，先后出现了几种数据模型。其中基本的数据模型有三种：层次模型系统、网络模型系统和关系模型系统。60 年代末期提出的关系模型具有数据结构简单灵活、易学易懂且具有雄厚的数学基础等特点，从 70 年代开始流行，发展到现在已成为数据库的标准。目前广泛使用的数据库软件都是基于关系模型的关系数据库管理系统。

3.2.1 关系模型（RM 即 Relational Model）

关系模型把世界看作是由实体（Entity）和联系（Relationship）构成的。

所谓实体就是指现实世界中具有区分与其它事物的特征或属性并与其它实体有联系的对象。在关系模型中实体通常是以表的形式来表现的。表的每一行描述实体的一个实例，表的每一列描述实体的一个特征或属性。

所谓联系就是指实体之间的关系，即实体之间的对应关系。联系可以分为三种：

- 一对一的联系。如：一个人只有一种性别，一个人→性别为一对一的联系；
- 一对多的联系。如：相同性别的人有许多个，性别→人为一对多的联系；
- 多对一的联系。如：很多人有同一个性别，人→性别为多对一的联系。

通过联系就可以用一个实体的信息来查找另一个实体的信息。

关系模型把所有的数据都组织到表中。表是由行和列组成的，行表示数据的记录，列表示记录中的域。表反映了现实世界中的事实和值。

3.2.2 关系数据库（RDB，即 Relational Database）

所谓关系数据库就是基于关系模型的数据库。

1. 关系数据库管理系统（RDBMS）

关系数据库管理系统就是管理关系数据库的计算机软件。

2. 关键字（Key）

关键字是关系模型中的一个重要概念，它是逻辑结构，不是数据库的物理部分。

- 候选关键字（Candidate Key）

如果一个属性集能惟一地标识表的一行而又不含多余的属性，那么这个属性集称为候选关键字。

- 主关键字（Primary Key）

主关键字是被挑选出来，作表的行的惟一标识的候选关键字。一个表只有一个主关键字。主关键字又可以称为主键。

- 公共关键字（Common Key）

在关系数据库中，关系之间的联系是通过相容或相同的属性或属性组来表示的。如果两个关系中具有相容或相同的属性或属性组，那么这个属性或属性组被称为这两个关系的公共关键字。

- 外关键字（Foreign Key）

如果公共关键字在一个关系中是主关键字，那么这个公共关键字被称为另一个关系的外关键字。由此可见，外关键字表示了两个关系之间的联系。以另一个关系的外关键字作主关键字的表被称为主表，具有此外关键字的表被称为主表的从表。外关键字又称作外键。

3.3 常见的数据库对象

数据库对象是数据库的组成部分，常见的有以下几种：

3.3.1 表（Table）

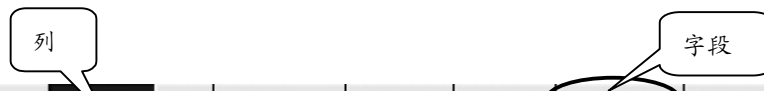
数据库中的表与我们日常生活中使用的表格类似，它也是由行（Row）和列（Column）

组成的。图 3-1 中黑底白字的部分称为行，图 3-2 中黑底白字的部分称为列。列由同类的信息组成，每列又称为一个字段，每列的标题称为字段名。行包括了若干列信息项。一行数据称为一个或一条记录，它表达有一定意义的信息组合。一个数据库表由一条或多条记录组成，没有记录的表称为空表。每个表中通常都有一个主关键字，用于惟一地确定一条记录，如图 3-1 中的 emp_id 列。图 3-1 中的第一条记录，即第一行数据所表达的信息为：编号为 10010001 的员工叫张三，性别男，出生于 1968 年 2 月 14 日，工作级别为 1，工作部门编号为 1001，受雇日期是 1996 年 8 月 2 日，月工资是 ¥8000 元。



emp_id	e_name	sex	birthday	job_level	dept_id	hire_date	e_wage
10010001	张三	1	68-2-14	1	1001	96-8-2	8000
10010002	李四	1	70-3-5	2	1001	98-6-23	5000
10010003	王二	0	66-7-8	3	1001	93-3-2	2000
10020001	张龙	1	53-10-12	2	1002	90-4-23	7000
10020002	赵虎	1	58-6-23	3	1002	91-9-2	4500
10020003	钱一	0	73-6-25	3	1002	96-3-15	3500
10030001	王胡	1	63-11-30	2	1003	94-4-15	4500
10030002	马汉	0	69-1-23	3	1003	96-3-31	3500

图3-1 员工信息表的行



emp_id	e_name	sex	birthday	job_level	dept_id	hire_date	e_wage
10010001	张三	1	68-2-14	1	1001	96-8-2	8000
10010002	李四	1	70-3-5	2	1001	98-6-23	5000
10010003	王二	0	66-7-8	3	1001	93-3-2	2000
10020001	张龙	1	53-10-12	2	1002	90-4-23	7000
10020002	赵虎	1	58-6-23	3	1002	91-9-2	4500
10020003	钱一	0	73-6-25	3	1002	96-3-15	3500
10030001	王胡	1	63-11-30	2	1003	94-4-15	4500
10030002	马汉	0	69-1-23	3	1003	96-3-31	3500

图3-2 表的列

3.3.2 索引 (Index)

索引是根据指定的数据库表列建立起来的顺序。它提供了快速访问数据的途径，并且可监督表的数据，使其索引所指向的列中的数据不重复。如图 3-3 所示。

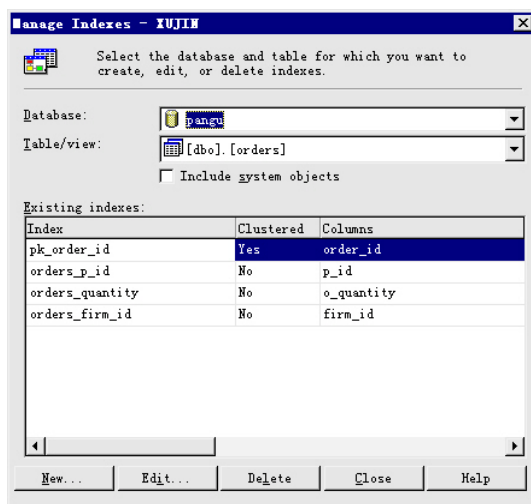


图3-3 管理索引

3.3.3 视图（View）

视图看上去同表似乎一模一样，具有一组命名的字段和数据项，但它其实是一个虚拟的表，在数据库中并不实际存在。视图是由查询数据库表产生的，它限制了用户能看到和修改的数据。由此可见，视图可以用来控制用户对数据的访问，并能简化数据的显示，即通过视图只显示那些需要的数据信息。例如：

从图 3-2 中抽取字段员工编号 `emp_id`、员工姓名 `e_name`、员工部门编号 `dept_id`、员工工资 `e_wage` 组成一个员工工资信息表。如图 3-4 所示。

	emp_id	e_name	dept_id	e_wage
▶	10010001	张三	1001	8000
	10010002	李四	1001	5000
	10010003	王二	1001	2000
	10020001	张龙	1002	7000
	10020002	赵虎	1002	4500

图3-4 员工工资信息视图

从图 3-2 中抽取数据组成工作级别（`job_level`）为 2 的员工信息表。如图 3-5 所示。

	emp_id	e_name	sex	birthday	job_level	dept_id	hire_date	e_wage
	10010002	李四	1	70-3-5	2	1001	98-6-23	5000
	10020001	张龙	1	53-10-12	2	1002	90-4-23	7000
	10030001	王朝	1	63-11-30	2	1003	94-4-15	4500
	10040001	那五	0	72-5-8	2	1004	96-9-1	7000

图3-5 `job_level` 为 2 的员工信息视图

3.3.4 图表 (Diagram)

在 SQL Server 中图表其实就是数据库表之间的关系示意图。利用它可以编辑表与表之间的关系。如图 3-6 所示。

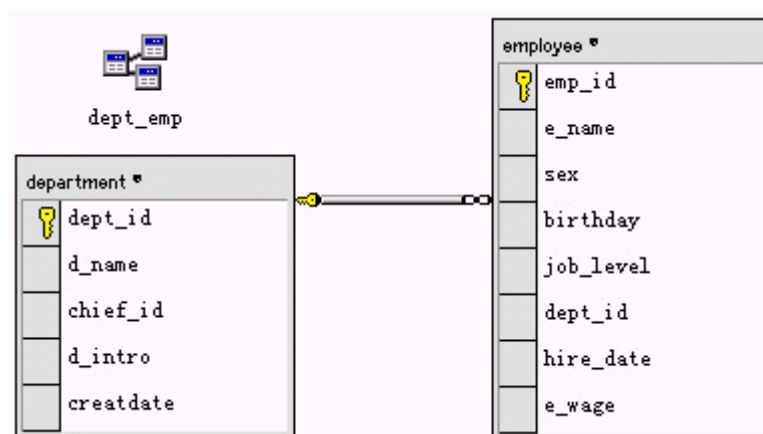


图3-6 图表 dept_emp

3.3.5 缺省值 (Default)

缺省值是当在表中创建列或插入数据时，对没有指定其具体值的列或列数据项赋予事先设定好的值。如图 3-7 所示。



图3-7 定义金额的缺省值

3.3.6 规则 (Rule)

规则是对数据库表中数据信息的限制。它限定的是表的列。如图 3-8 所示。

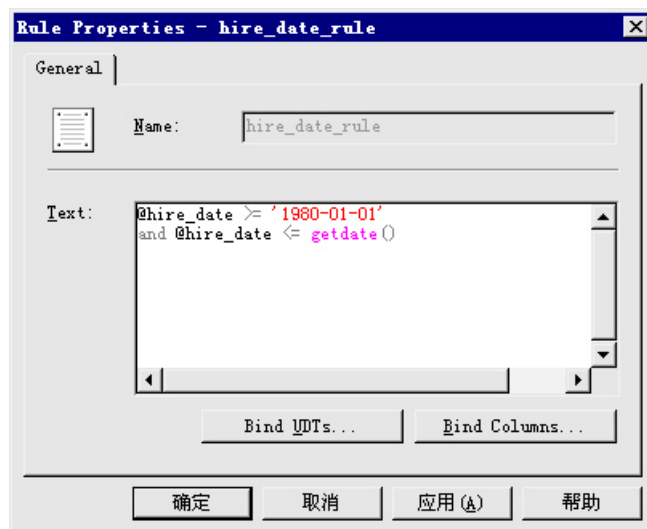


图3-8 规则 hire_date_rule

3.3.7 触发器（Trigger）

触发器是一个用户定义的 SQL 事务命令的集合。当对一个表进行插入、更改、删除时，这组命令就会自动执行。如图 3-9 所示。

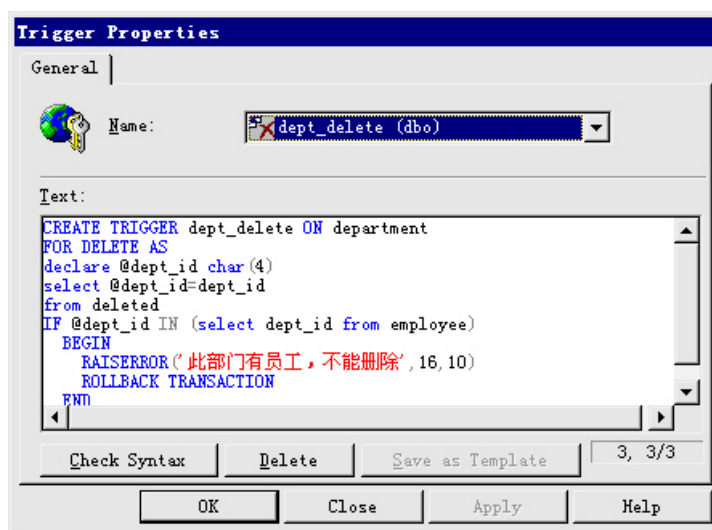


图3-9 触发器 dept_delete

3.3.8 存储过程（Stored Procedure）

存储过程是为完成特定的功能而汇集在一起的一组 SQL 程序语句，经编译后存储在数据库中的 SQL 程序。如图 3-10 所示。

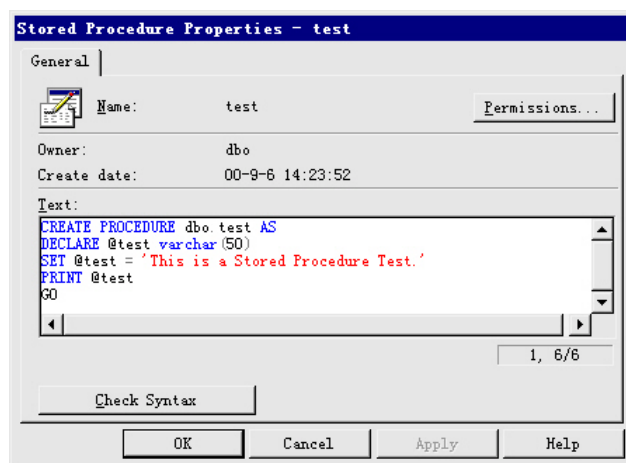


图3-10 存储过程 test

3.3.9 用户（User）

所谓用户就是有权限访问数据库的人。

数据库对象还有很多，我们将在以后的章节中详细介绍。

3.4 范式

构造数据库必须遵循一定的规则。在关系数据库中，这种规则就是范式。范式是符合某一种级别的关系模式的集合。关系数据库中的关系必须满足一定的要求，即满足不同的范式。目前关系数据库有六种范式：第一范式（1NF）、第二范式（2NF）、第三范式（3NF）、第四范式（4NF）、第五范式（5NF）和第六范式（6NF）。满足最低要求的范式是第一范式（1NF）。在第一范式的基础上进一步满足更多要求的称为第二范式（2NF），其余范式以次类推。一般说来，数据库只需满足第三范式（3NF）就行了。下面我们举例介绍第一范式（1NF）、第二范式（2NF）和第三范式（3NF）。

3.4.1 第一范式（1NF）

在任何一个关系数据库中，第一范式（1NF）是对关系模式的基本要求，不满足第一范式（1NF）的数据库就不是关系数据库。

所谓第一范式（1NF）是指数据库表的每一列都是不可分割的基本数据项，同一列中不能有多值，即实体中的某个属性不能有多值或者不能有重复的属性。如果出现重复的属性，就可能需要定义一个新的实体，新的实体由重复的属性构成，新实体与原实体之间为一对多关系。在第一范式（1NF）中，表的每一行只包含一个实例的信息。例如，对于图 3-2 中的员工信息表，不能将员工信息都放在一列中显示，也不能将其中的两列或多列在一列中显示；员工信息表的每一行只表示一个员工的信息，一个员工的信息在表中只

出现一次。简而言之，第一范式就是无重复的列。

3.4.2 第二范式（2NF）

第二范式（2NF）是在第一范式（1NF）的基础上建立起来的，即满足第二范式（2NF）必须先满足第一范式（1NF）。第二范式（2NF）要求数据库表中的每个实例或行必须可以被惟一地区分。为实现区分通常需要为表加上一个列，以存储各个实例的惟一标识。如图 3-2 员工信息表中加上了员工编号（emp_id）列，因为每个员工的员工编号是惟一的，因此每个员工可以被惟一区分。这个惟一属性列被称为主关键字或主键、主码。

第二范式（2NF）要求实体的属性完全依赖于主关键字。所谓完全依赖是指不能存在仅依赖主关键字一部分的属性，如果存在，那么这个属性和主关键字的这一部分应该分离出来形成一个新的实体，新实体与原实体之间是一对多的关系。为实现区分通常需要为表加上一个列，以存储各个实例的惟一标识。简而言之，第二范式就是非主属性非部分依赖于主关键字。

3.4.3 第三范式（3NF）

满足第三范式（3NF）必须先满足第二范式（2NF）。简而言之，第三范式（3NF）要求一个数据库表中不包含已在其它表中已包含的非主关键字信息。例如，存在一个部门信息表，其中每个部门有部门编号（dept_id）、部门名称、部门简介等信息。那么在图 3-2 的员工信息表中列出部门编号后就不能再将部门名称、部门简介等与部门有关的信息再加入员工信息表中。如果不存在部门信息表，则根据第三范式（3NF）也应该构建它，否则就会有大量的数据冗余。简而言之，第三范式就是属性不依赖于其它非主属性。

3.5 本章小结

本章介绍了数据库的基本概念、常见的数据库对象和构造数据库的规则，它们有助于您理解以后章节中的内容。