

情境 3 权限管理系统数据库设计

任务 3.1 权限管理系统数据库用户需求分析



任务描述

权限系统一直以来都是应用系统不可缺少的一个部分。系统在实际应用过程中,一般都会由多种身份的用户来使用,而不同的用户又拥有不同的系统使用权限。那么,如何在数据库设计的过程中,为系统的不同用户来设计各自不同的使用权限?

若每个应用系统在数据库设计过程中都重复对系统的权限进行设计,以满足不同用户的需求,那将会浪费不少宝贵时间,所以花时间来设计一个相对通用的权限系统是很有意义的。

本系统的设计目标是通过用户管理和角色管理对应用系统的所有资源进行权限分配,例如,应用系统的各功能菜单,各个界面的按钮控件等都可以进行权限的分配。



相关知识

1. 权限

应用系统中的权限是指可以对数据执行的各种处理或操作的集合。例如,新生入学管理系统中,可以对学生基本信息进行输入、删除、更新等操作,还可以进行学生信息的浏览、打印和根据输入条件完成记录的查询等操作。能够对系统中的数据进行不同的处理和操作,就意味着拥有了相应的权限。权限可以分为授予权限和访问权限,本系统所研究的是资源的访问权限。

2. 角色

角色是权限的集合。一个角色可以拥有多个不同的权限。例如,高校课务管理系统中有教师角色、学生角色等,不同角色拥有不同权限。

3. 用户

用户是应用系统的具体操作和使用用户。每个用户都可以拥有自己的权限,拥有了某一权限就意味着可以完成该权限规定的操作。本系统用户通过分配角色,得到该角色所对应的用户的权限。

4. 用户分组

有些时候,系统为了更好地管理用户,可以再对用户进行分组归类,简称为用户分组。组可以具有自己的角色信息、权限信息。例如,一个 QQ 群可以看成是一个分组,它拥有多个用户,一个用户也可以加入多个群。每个群具有自己的权限信息,如查看群共享。QQ 群也可以具有自己的角色信息,例如普通群、高级群等。



任务分析与实施

根据系统用户需求分析,得到系统功能结构如图 3-1 所示。

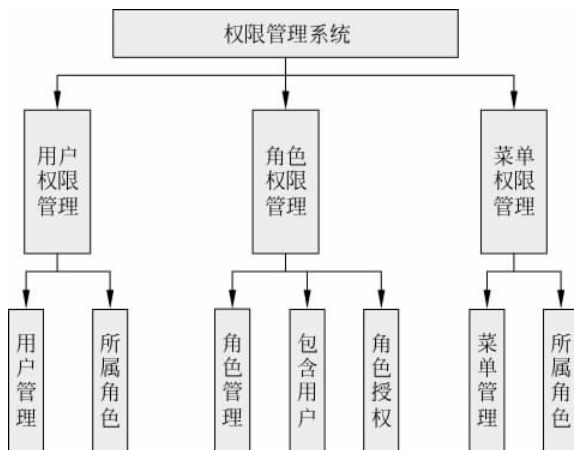


图 3-1 权限管理系统功能结构图

(1) 用户权限管理模块。负责进行系统用户的添加、删除、更新和维护。完成用户角色的指定。

(2) 角色权限管理模块。可以完成角色的创建、删除、更新和维护。为角色授予菜单使用权限,同时为角色指定用户。

(3) 菜单权限管理模块。负责对系统主菜单和子菜单项进行添加、删除、更新和维护。通过分配系统菜单的使用权限给不同角色的方式使用用户拥有不同权限。对于高校课程管理系统来讲,菜单可以由学生信息管理、教师信息管理、课程选修、成绩录入等菜单项组成。对于图书管理系统来讲,菜单可以由图书入库、图书查询、图书借阅、图书归还、欠费收缴等系统菜单项组成。给不同的用户授予不同菜单项的使用权限。

任务 3.2 权限管理系统数据库概念设计



任务描述

根据权限管理系统的用户需求,完成系统数据库的概念设计。绘制系统 E-R 图,明

确各实体属性和关系。



相关知识

权限管理过程中常常会出现 3 种对象,即用户对象、角色对象和权限对象。它们之间呈现出较为复杂的逻辑关系。

1. 用户对象

一个用户可以归属于 $0 \sim n$ 多个不同角色。一般来说,它的权限集是自身具有的权限和所属的各角色具有的权限的合集。它与权限、角色之间的关系都是 $n:m$ 的逻辑关系。

2. 角色对象

它可以包含 $0 \sim n$ 多个用户。角色与权限、用户之间的关系是 $n:m$ 的关系。一个角色可以拥有多个权限,一个权限也可以属于多个角色。

3. 权限对象

一个权限可以属于 $0 \sim n$ 多个用户,也可以属于 $0 \sim n$ 个角色。因此角色、权限和用户之间都是 $n:m$ 的关系。

以高校课务管理系统数据库设计为例,系统中包含系统管理员、教务员、普通教师、班主任和学生 5 种角色。其中,张老师作为一名普通教师可以拥有多种不同的权限(如浏览选课信息);张老师可以属于多种不同角色(如张老师是教务员,同时还是任课教师)。

系统中权限可以包括学生基本信息的录入、浏览、检索、删除、更新等操作。其中,一个权限可以属于不同角色和分组。例如,浏览学生基本信息的权限可以既属于普通教师角色,又属于系统管理员角色。同理,这个权限既可以属于行政人员分组,又可以属于专业教师分组。

综上所述,在权限管理系统中,用户、权限和角色相互之间呈现的是多对多($n:m$)的对象关系。



任务分析与实施

1. 信息收集

创建数据库之前,必须充分理解和分析系统需要实现的功能,以及系统实现相关功能的具体要求。在此基础上,考虑系统需要存储哪些对象,这些对象需要保存哪些基本信息。由图 3-1 所示的权限管理系统功能结构图分析可得系统需要保存的对象如下。

(1) 用户对象。用来记录并保存系统使用者的基本信息。用户可以属于不同角色,

通过角色划分得到不同权限。

(2) 权限对象。包含系统中对某一个角色指定系统资源使用权利分配的信息。在一个实际的应用系统中,权限可以理解为对一个网页浏览的授权,对一个功能菜单的使用授权,甚至是一个页面上某个按钮的使用授权。例如,高校课程管理系统的资源可以如图 3-2 所示的菜单来实现控制。

(3) 角色对象。为具有相同权限的用户集合定义一个角色,系统可以通过角色来给角色中的成员统一分配权限的对象。

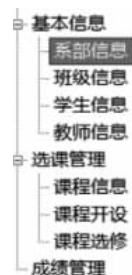


图 3-2 高校课务管理系统主菜单(部分)

2. 确定实体及属性

为达到系统权限管理的功能要求,对系统对象进行分析并完成基本属性的抽取。

用户: 用户编号、用户名称、登录名、登录密码、联系电话、电子邮箱、登录时间、是否为管理员、是否有效。

角色: 角色编码、角色名称、角色描述、是否有效。

权限菜单: 菜单编码、菜单名称、父菜单编码、菜单窗体、菜单项序号、是否有效。

3. 标识实体间关系

如前所述,系统中用户对象、权限对象和角色对象相互之间是一种较为复杂的多对多数据关系。

根据上述对象(实体)分析,权限管理系统的 E-R 图如图 3-3 所示。

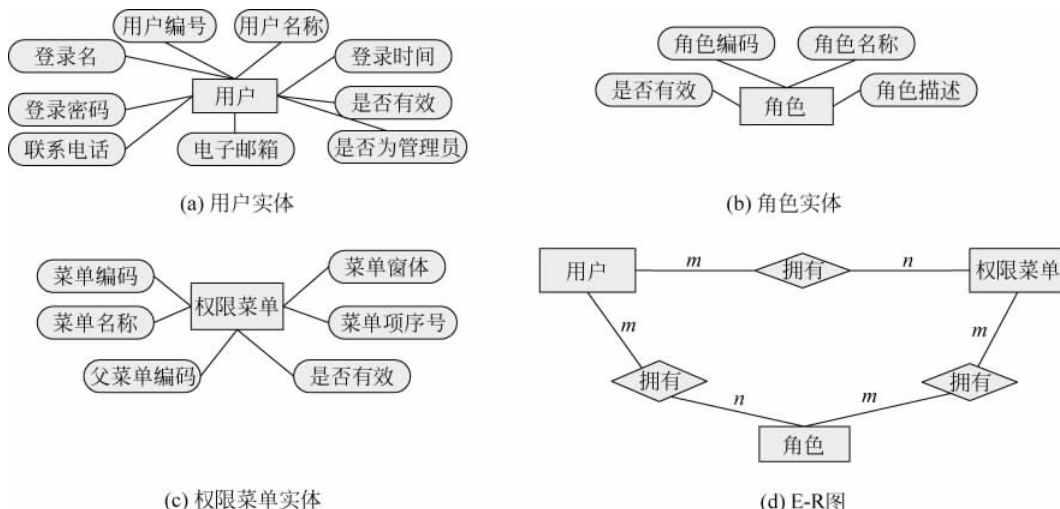


图 3-3 权限管理系统 E-R 图

任务 3.3 权限管理系统数据库逻辑设计



任务描述

根据系统概念设计完成具体逻辑设计。进一步将 E-R 图转化为二维关系表,确定表结构、字段类型和主外键联系等。



相关知识

3.3.1 Identity 标识字段应用

用 Identity 关键字定义的字段又叫标识字段,一个标识字段是唯一标识表中每条记录的特殊字段,标识字段的值是整数类型。当一条新记录添加到这个表中时,系统就给这个字段自动递增赋给一个新值,默认情况下是加 1 递增。每个表只可以有一个标识字段。

当需要在多个数据库间进行数据的复制时(SQL Server 的数据分发、订阅机制允许进行库间的数据复制操作),自动增长型字段可能造成数据合并时的主键冲突。因此,自动增长数据类型的字段不宜做表的外键字段。

3.3.2 UniqueIdentifier 标识字段应用

与 UniqueIdentifier 类似,SQL Server 还提供了另一种标识字段定义方式,即 UniqueIdentifier 数据类型,同时提供了一个生成函数 NEWID()。使用 NEWID()可以生成一个唯一的 UniqueIdentifier。UniqueIdentifier 在数据库中占用 16 个字节。

UniqueIdentifier 字段的使用存在一些缺陷:首先,它的长度是 16 个字节,是整数的 4 倍长,会占用大量存储空间。而且 UniqueIdentifier 的生成没有规律,要想在上面建立索引(绝大多数数据库在主键上都有索引)是一个非常耗时的操作。使用 UniqueIdentifier 型数据做主键要比使用 Integer 型数据慢,所以,出于效率考虑,尽可能避免使用 UniqueIdentifier 型数据作为主键键值。



任务分析与实施

1. 表及主键定义

根据权限管理系统概念设计,可以由实体直接转化得到的表及其主键定义,如表 3-1 所示。

表 3-1 权限管理系统实体表

实 体	数 据 表	主 键
用户	用户表 TB_User	用户编码 UserId
权限菜单	权限表 TB_PowerMenu	权限编码 PowerId
角色	角色表 TB_Role	角色编码 RoleId

由于在权限管理系统概念设计阶段,分析得到实体间存在较为复杂的多对多关系,所以需要进一步将多对多关系转化为一对多关系。转化后的结果如图 3-4 所示。

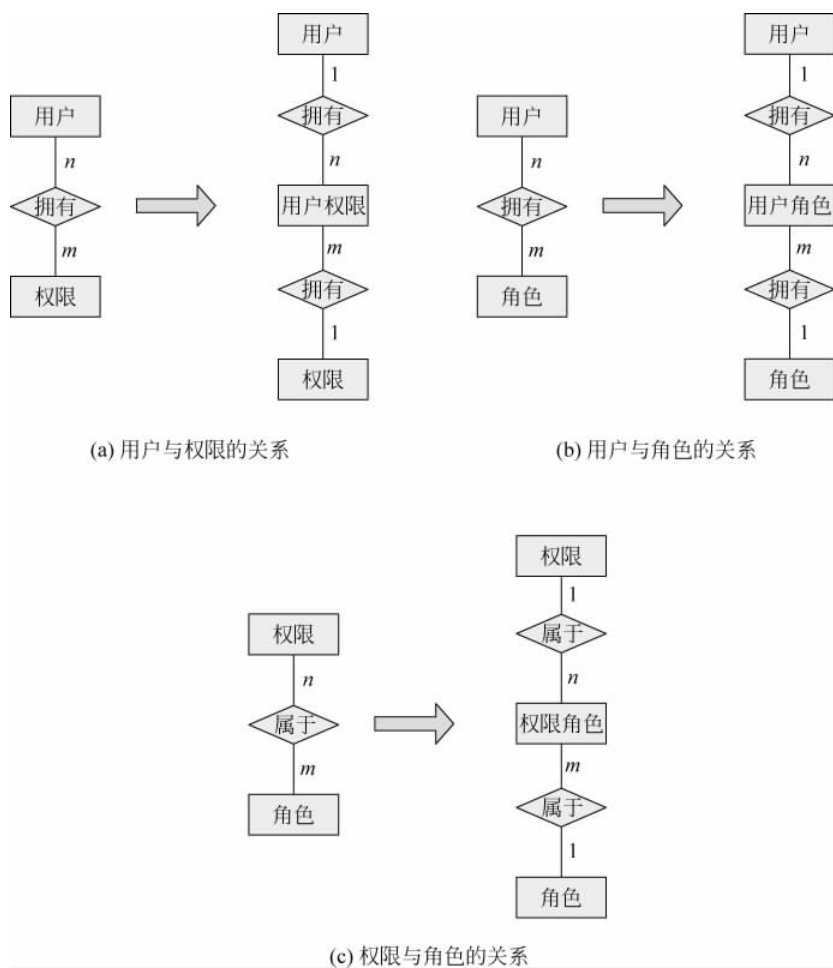


图 3-4 多对多关系转化成 E-R 图

多对多关系转化成一一对多关系后产生的新表结构如图 3-5 所示。在新增加的表中必须包含原来多对多关系表的主键字段。

新增表主键定义如表 3-2 所示。

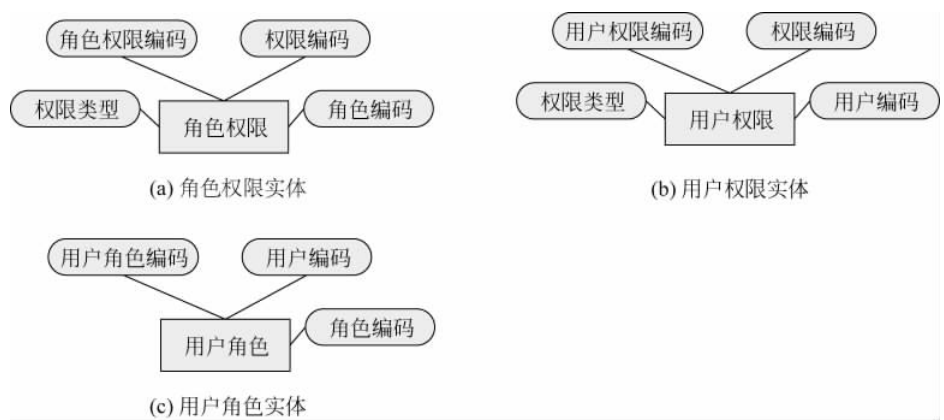


图 3-5 多对多关系转化后新增加的表

表 3-2 转化后新表及其主键

数 据 表	主 键	主键数据类型
用户权限表 TB_UserPower	UserPowerId	int, 标识字段
用户角色表 TB_UserRole	UserRoleId	int, 标识字段
角色权限表 TB_RolePower	RolePowerId	int, 标识字段

2. 外键定义

根据概念设计得到的实体对象间存在的多对多数据联系,设计表外键。以用户权限表的外键设计为例,主键表是用户 TB_User 表和权限 TB_Power 表,主键字段分别为两个表中的 UserId 字段和 PowerId 字段。外键表是用户权限表 TB_UserPower,其中将其所关联的用户表和权限表中的主键 UserId 和 PowerId 字段设置为该表的外键字段。

3. 表字段与类型定义

合理设计表的字段类型才能够准确地存储数据和表与表之间的逻辑联系。根据权限管理系统的概念设计,对各表字段定义如表 3-3~表 3-8 所示。

表 3-3 TB_PowerMenu

PK	字段名称	字段类型	NOT NULL	默认值	约束	字段说明
●	PowerId	int	○		主键	权限编号
	PowerMenuName	varchar(20)	○			权限菜单名称
	WinforLink	varchar(30)	○			菜单窗体链接
	Sort	smallint	○			菜单排列序号
	ParentMenu	int				父菜单编号,为空表示无父菜单项
	IsValid	bit	○	1	CHECK	是否可用,0:不可用,1:可用

表 3-4 TB_Role

PK	字段名称	字段类型	NOT NULL	默认值	约束	字段说明
●	RoleId	int	○		主键	角色编号
	RoleName	varchar(20)	○			角色名称
	IsValid	bit	○	1	默认值	是否有效
	Description	varchar(30)	○			角色描述

表 3-5 TB_User

PK	字段名称	字段类型	NOT NULL	默认值	约束	字段说明
●	UserId	int	○		主键	用户编号, 标识字段, 流水编号
	UserName	char(8)	○			用户名称
	LoginId	varchar(20)	○			登录账号
	LogPwd	varchar(10)	○			
	LogTime	smalldatetime	○	系统时间	DEFAULT	登录时间
	Email	varchar(40)	○		CHECK	电子邮箱, @符号前后必须为有效字符
	TelePhone	char(11)	○		CHECK	联系电话
	IsAdmin	bit	○	0	DEFAULT	是否为管理员
	IsValid	bit	○	1	DEFAULT	是否有效

表 3-6 TB_UserRole

PK	字段名称	字段类型	NOT NULL	默认值	约束	字段说明
●	UserRoleId	int	○		主键	用户角色编号, 标识字段 Identity
	UserId	int	○		外键	用户编号, 主键表 TB_User
	RoleId	int	○		外键	角色编号, 主键表 TB_Role

表 3-7 TB_UserPower

PK	字段名称	字段类型	NOT NULL	默认值	约束	字段说明
●	UserPowerId	int	○		主键	用户权限编号
	UserId	int	○		外键	用户编号
	PowerId	int	○		外键	权限编号
	PowerType	bit	○		CHECK	权限类型, 0: 访问; 1: 授权

表 3-8 TB_RolePower

PK	字段名称	字段类型	NOT NULL	默认值	约束	字段说明
●	RolePowerId	int	○		主键	角色权限编号
	RoleId	int	○		外键	角色编号
	PowerId	int	○		外键	权限编号
	PowerType	bit	○		CHECK	权限类型, 0: 访问; 1: 授权

其中,很多表的主键都设置为 int 类型,如用户表 TB_User 表的主键字段设置为 int 类型,作为 Identity(1,1)标识字段的自增数据类型处理,同时,该字段在表 TB_UserRole 中又作为外键字段出现。那么,Identity 自增类型的字段适合做外键吗? 自增类型字段的值在向表中添加数据时是不需要人为输入的,系统会根据表中当前记录值生成下一条新记录的 Identity 字段值。如果使用自动增长类型的字段做外键,一旦发生数据复制、移动等操作,由于自增字段的值重新排列编号,就会出现数据合并过程中主键冲突的问题。因此,表 3-3~表 3-8 中的主键字段设计进行修改,修改后的表主外键属性设计如表 3-9 所示。

表 3-9 权限管理系统主外键设计表(优化后)

表	主 键	类 型	外 键	类 型
用户表 TB_User	UserId	char(6)		
权限表 TB_Power	PowerId	char(4)		
角色表 TB_Role	RoleId	char(4)		
用户权限表 TB_UserPower	UserPowerId	int	UserId	char(6)
			PowerId	char(4)
用户角色表 TB_UserRole	UserRoleId	int	UserId	char(6)
			RoleId	char(4)
角色权限表 TB_RolePower	RolePowerId	int	RoleId	char(4)
			PowerId	char(4)

其中,用户权限表的主键 UserPowerId 数据类型是 int 型,可设置为标识字段,该字段的取值没有实际含义,只是表示表记录的一个流水号。TB_UserRole 和 TB_RolePower 表主键设计同理。

情境 4 创建新生入学管理系统数据库

任务 4.1 安装和配置 SQL Server 2012 数据库服务器



任务描述

在反复论证并完成系统逻辑设计方案之后,可以进入数据库的物理实施阶段。在物理实施阶段,选用微软的 SQL Server 2012 数据库系统进行新生入学管理系统数据库的创建和表的设计。



相关知识

2012 年 3 月,Microsoft 全球发布了 SQL Server 2012 RTM,越来越多的人看到 SQL Server 2012 拥有的众多新增功能和增强特性,它不仅可以有效地执行大规模联机事务处理,完成数据库仓库和电子商务应用等许多具有挑战的工作,更为当今社会大数据时代的到来带来新的应用与挑战。

4.1.1 SQL Server 2012 特性

SQL Server 2012 完全重新定义了 SQL Server 的数据平台,不仅为小型、中型和大型的机构建立其下一代 IT 基础架构的应用提供了基石,而且包括许多新的和改进的功能来帮助用户更有效率地工作,这也使它成为大规模联机事务处理、数据仓库和大数据处理应用程序的优秀数据库平台。下面列举了 SQL Server 2012 的几个重要功能。

1. 数据库引擎

数据库引擎是 SQL Server 2012 系统的核心服务,它是存储和处理关系格式的数据或 XML 文档数据的服务,负责完成数据的存储、处理和安全管理。SQL Server 2012 在数据库引擎方面引入了多项改进的核心功能用来提高程序员的开发能力和工作效率。