# 软件设计

# 数据库分析与设计

## 概述

一个数据库应用系统的设计包含俩方面的内容:

1. 结构特性的设计,即数据库模式或数据结构的设计
2. 行为特性的设计,即应用程序,事务处理的设计

采用UML进行系统的分析与设计,可以把系统的应用程序(行为)设计和数据库设计统一起来

数据模型(Data Model)是对现实世界的抽象,是建立数据库的基础,通常在一个数据模型中需要描述数据的组织结构,对数据的操作和数据的完整性约束

当前存在的数据模型大体可以分为俩类:

1. 基于记录的逻辑模型

传统的三种数据模型:层次模型,网状模型和关系模型

他们是以数据记录和数据项作为基本的数据结构

1. 基于对象的逻辑模型:

实体联系模型(ER模型),面向对象数据模型等

他们是更高抽象层次意义上的概念模型

### 传统的数据库设计

传统的关系数据库设计的过程分为四个阶段:需求分析,概念结构设计,逻辑结构设计,物理结构设计

需求分析阶段: 对数据的存储要求和处理要求进行描述,编制出数据库需求分析说明书

数据库的概念结构设计:

或成为企业的组织模式,它是独立于任何一种数据模型和任何具体的DBMS的信息结构模型,是现实世界的纯粹表示

需要借助于某种工具或方法,当前应用最广泛的是ER(实体-联系方法),使用ER图定义企业的信息组织模式

### 使用UML设计数据库

做法:

1. 以ER图为基础进行数据库设计,设计工作按照er模型的概念和方法进行,只是用uml表达
2. 用uml直接进行数据库的分析设计和表达

第二种做法可以把数据库设计与应用系统软件的设计统一进行

#### 综述：

UML的对象类图能够更好地用于对数据库建模,UML的对象类图不但对数据,而且能对行为建模.这些行为在物理数据库中被设计成触发器和存储过程,或者专用的Utility

#### 设计步骤

使用UML进行数据库设计的四个阶段:

##### 业务use case 模型设计

使用use case图,类图，顺序图，活动图等建立use case 模型

##### 逻辑数据模型设计

确定应用所需要的持久数据,对于RDBMS而言,需要设计出表达持久数据的实体类(Entity Class)及其联系,并把它们映射成为关系数据库表,视图等

##### 物理数据模型设计

使用类图 组件图 配置图等设计数据库的物理模型

##### 物理实现设计

把物理数据模型转换到实际的数据库环境

### UML用于数据库设计的规范

#### 表

表是RDBMS的基本建模结构

表又称为二维表,关系，它是具有相同结构的行的集合

表中的每一列称为关系的属性

任一行与列交叉点的元素称作分量,每一个数据必须作为一个分量值存储

每一个表有一个关系模式(Relational Scheme),它由一个表名以及它所有的列明构成,它是一个二维表的表头,规定该表的组成,列的特性,以及完整性约束等

表结构及相关的术语如下图:



##### 要点

在uml中表用类表示,带有构造型<<Table>>,类名即表名,类的属性描述表的列特性,包括列名,数据类型,以及有关的约束,在类的行为部分,根据需要可以给出对表的一些操作,也可以缺省.

如学生表和课程表的UML类图如下:



#### 关键字与索引

##### 概述:

###### 关键字(key)

关键字(key)用于存取表中的数据

主关键字(primary key)是一个属性或属性组,他的值唯一地标识表中的行

外来关键字(foreign key)是一个属性或属性组,她在本表中不是关键字,不能唯一地表示表中的行,但它在另一个表中是主关键字,能够唯一地表示该表中的行,外来关键字体现了俩个表的联系,实现表之间的参照的完整性.

主关键字和外来关键字的表示也可以分别使用构造型<<pk>>和<<fk>>.

###### 索引

索引(Index)支持快速数据存取的物理结构

在表中可以指定某个属性作为索引项,或指定某几个属性组成多级索引项

在UML中索引的构造型为<<Index>>

如图:

