当前电子计算机的采用使模拟技术有了重大的发展。例如，飞行模拟器主要由座舱、运动系统、视景系统和电子计算机 4 个部分组成。飞行员在这样的模拟器中训练，具有逼真的视、听、运动等各种感觉，犹如在空中飞行一样。又如，在人工智能研究领域，用计算机模拟人的思维，对研究人的高级心理活动、揭示人的思维规律能起重大作用。

模拟模型（Simulation Model）：就是根据系统或过程的特性，按一定规律用实验设计或数学方程建模、计算机语言程序编辑等手段建立数学模型、物理模型、系统模型等对系统及运动规律进行贴近模拟和展示的方法。

草业信息模拟模型（Simulation Model for Pratacultural Information）：利用实验设计、系统分析、数学计算和计算机编程等技术，对草业系统中的生物与非生物过程及其与环境和技术的动态关系进行信息收集、定量分析、有效描述和预测。因此，草业信息模拟模型以草业系统中的内在规律和相互关系为基础，综合系统内在特征、环境效应、技术调控之间的因果关系，是一种面向草地生物和非生物环境过程的实验综合设计、系统动力学模型、过程模型及统计学模型的统称。

（二）模拟模型的类型

按不同的划分标准，模拟模型有不同的类型，具体如图 4-1 所示。

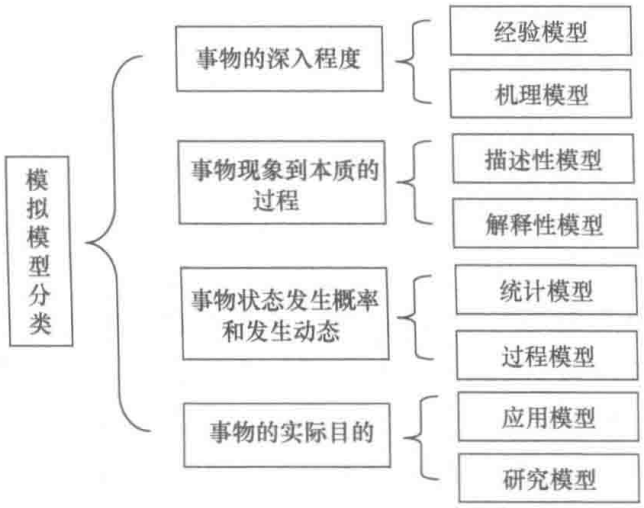


图 4-1　模拟模型类型的划分图

1）按对研究事物的深入程度可分为经验模型（Experiential Model）和机理模型（Mechanism Model）。不分析实际过程的机制，而是根据从实际得到的与过程有关的数据进行数理统计分析，按误差最小原则，归纳出该过程各参数和变量之间的数学关系式，用这种方法所得到的数学表达式称为经验模型。所以，经验模型又被称为黑箱模型。机理模型是在一定的假设条件下，根据主要因素相关性及相互作用的机制，建立模拟条件下现实事物状态的模型，一般具有较好的模拟效果。

常用的 AHP 模型，即层次分析法（Analytic Hierarchy Process，AHP），就是典型的经验模型。这是美国运筹学家、匹兹堡大学 Saaty 教授在 20 世纪 70 年代初期提出的一种经验模型构建方法，主要特点就是通过对各种因素的分析，将复杂的问题变得条理化和层次化，然后利用数理统计的方法赋予每个因素以权重，进而为决策提供参考信息。

例如，李文龙等（2010）利用 AHP 分析方法对甘南藏族自治州（以下简称甘南州）玛曲高寒草地放牧系统的生态风险进行了综合评价。

经验模型的主要特点是不分析实际过程的机制，从这个角度上讲除了耳熟能详的经典模型外，在科研中这种经验模型方法也有广泛应用。例如，冯琦胜等（2011）基于地面实测数据和 MODIS-EVI 及 NDVI 数据，利用统计学方法，确定了青藏高原地区草地地上生物量和植被盖度的最佳遥感反演模型。从模型的构建方法可以看出，这些反演模型也应属于经验模型。

对于机理模型来说，其核心内容是分析主要因素相关性及其相互作用的机制，如常用的 CASA 模型。该模型是 1993 年由 Potter 等提出，它是通过植被吸收光合有效辐射（APAR）和光能转化率（ε）两个变量来估算区域植被净初级生产力（NPP）。例如，张峰等（2008）在利用内蒙古典型草原连续 13 年的地上生物量资料对 CASA 模型验证的基础上，分析了内蒙古典型草原植被的初级生产力。

2）从研究事物现象到本质的过程上可分为描述性模型（Descriptive Model）和解释性模型（Explanatory Model）。描述性模型反映了从特殊到一般的认识过程，它是从分析客观事物的具体特征入手，经过逐步抽象而得到的。把客观事物中的关系概括于一个数学结构之中，是描述性数学模型的主要特征。解释性模型又称为解释结构模型，主要应用图的矩阵表示方法和简单的逻辑运算，对复杂系统的各个组成元素或子系统间的结构关系加以描述的一种模型。通过对表示有向图的相邻矩阵的逻辑运算，得到可达性矩阵，然后分解可达性矩阵，最终使复杂系统分解成层次清晰的多级递阶形式。解释结构模型在制订企业计划、城市规划等领域已广泛使用，尤其对于建立多目标和多种元素之间关系错综复杂的社会系统及其分析，效果更为显著。

例如，Accad 和 Neil（2006）基于地理信息系统和植被群落针对澳大利亚东北部建立起群落变迁的模型是典型的描述性模型。国内张钟军和孙国清（2005）针对植被层内的体散射及植被和地表之间的多次散射、地表辐射分别采用了双矩阵法和积分方程建立起估算地表亮度温度的模型也属此类模型。刘兴元等（2003）分析了影响雪灾的因素，并通过栅格获取法和模糊 Borda 数分析法确定了各因素的权重，进而建立了雪灾对草地畜牧业影响的定量评价和雪灾损失计算模型，这种模型属于解释性模型。

3）按研究事物状态发生的概率及发生的动态可以分为统计模型（Statistical Model）和过程模型（Process Model）。统计模型是 20 世纪 50 年代中期发展起来的一个新的数理研究方向，其主要目的是运用基于统计学原理的数学模型模拟概率事件的发生及分布特点，以及评价建模的适当性和识别数据中可能存在的异常值。在数学模型适当性的评价方面，线性模型中目前主要采用残差分析来判断模型拟合的效果。过程模型在很多情况下用于模拟事物生命周期状态，它将一个事物的运动作为一个连续过程，并进行有效的数学拟合，此类模型不但能表达整个事物在一定时间区间的动态，而且可以解释阶段性的离散状态。

统计模型具有变量少、便于计算等特点。例如，梁天刚等（2009）利用 2001～2008 年甘南牧区草地调查资料和 Terra/MODIS 每日地表反射率产品 MOD09GA，借助数学统计的方法，筛选出相关性最高的草地地上生物量遥感反演模型。这类模型只有一个自变

量——基于 MOD09GA 数据的植被指数。

通常，过程模型具有涉及的变量较多、模型复杂等特点。例如，刘文等（1992）以每天的太阳辐射、气温、降水等因素为驱动变量，建立了棉花生长发育及产量形成的动态模拟模型。白文明和包雪梅（2002）基于紫花苜蓿生理生态学特性建立了水分限制条件下干旱沙区紫花苜蓿的生长发育模拟模型。

4）如果按照研究事物的实际目的则可以分为应用模型（Application Model）和研究模型（Research Model）。前者偏向于建立的模型进行决策、预测、辅助管理等实际应用功能；后者更倾向于设立很多理论假设条件，进行事物状态、过程和极限条件（现实不存在的）的理论探索。

顾名思义，应用模型偏向于在实践中的应用。例如，梁天刚等（2002）设计研发了甘肃省草业开发专家系统的总体结构，在栽培牧草模拟模块中采用的基于温度、降水等多因子的牧草地域适宜性空间模拟模型就属于应用模型。该系统以“环境-植被-家畜”为主线，系统便捷地提供了科学可靠的草业信息和建议。而研究模型主要偏向于科研，对其机理性要求高。例如，黄敬峰等（2001）利用天然草地牧草光谱观测资料、牧草产量资料、气象资料和 NOAA/AVHRR 资料等建立了天然草地牧草产量遥感预测模型及气象预测模型，这就是一个研究模型的典型实例。

二、草业信息模拟模型的特征、意义及功能

（一）草业模型的特征

与任何模拟模型相似，草业信息模拟模型有其自身的特征，具体可以概括如下。

1. 系统性

对研究的生物及非生物过程需要进行系统和全面的定性、定量分析与描述。

2. 动态性

具体指受环境因子和内在特性驱动的各个状态变量具有随时间过程变化的特点，以及不同生育过程的状态变量之间具有相应的一些动态变化关系。

3. 机理性

在经验性或描述性的基础上，通过进行深入的研究，模拟较为全面的系统等级水平，并将其进行有机结合，从而提供对主要生理过程的理解或解释。

4. 预测性

通过正确建立模型的主要驱动变量及其与状态变量的动态关系，对系统行为提供可靠的定量预测。

5. 通用性

原则上适用于任何地点、时间和品种等条件。

6. 便用性

可为非专家操作应用，可利用一般的气候、土壤及品种资料。

7. 灵活性

可容易地进行修改和扩充及与其他系统相耦合。

8. 研究性

除了应用性以外，还可用于不同领域的模拟研究工作，从而弥补实际研究中干扰因素多、周期长、费用高等方面的不足。

（二）草业模拟模型的意义与功能

1. 模拟模型的意义

草业信息模拟模型最重要的意义是对整个草地农业生产系统的知识可以进行一定条件下有效地收集、综合，并量化机理性过程及其相互关系，即综合知识和量化关系。在理解生物与非生物过程及其变量间关系的基础上，进行量化分析和数理模拟，构建模型并开展模拟研究，从而促进对生物与非生物规律由定性描述向定量分析的转化过程，可深化对草地农业系统过程的定量化认识和数字化表达。

2. 模拟模型的功能

合理的草业模拟模型之所以受到肯定和重视，是因为模拟模型具有其他研究手段不可替代的功能，如分析、理解、预测、调控等。模拟模型能够基于定量计算、参数选择等手段对研究目标进行有效的理论分析，可以帮助人们理解和认识草地生物与非生物过程的基本规律及量化关系，并对系统的动态行为和最后表现进行预测，从而辅助进行对草地农业系统的生物生长和生产系统的适时合理调控，实现优质、高产、合理效费比、生态健康、环境安全的可持续发展。

第二节　草业系统模拟的原理与技术

草学、作物学、动植物生理生态知识是建立植物或动物生长系统的概念模型直至量化模型的关键。对于一个面向过程的草地生长模型而言，最重要的技术基础是作物或动物生理生态学原理、系统分析方法和软件工程技术。系统分析方法是草地农业信息模拟研究的基础，软件工程技术是实现模拟模型的工具。

一、系统分析方法

系统是一组具有一定功能的相关成分的集合，可以分解成不同的结构成分，也可合成为一个整体系统。系统研究可分为两个主要方面，即系统分析和系统合成。

系统分析：将一个系统分解成主要成分，研究系统的成分及关系，提供系统的定量

描述，即系统模型来表达系统行为的过程。

系统合成：主要指研究如何运用从系统分析中获得的知识或系统模型来改良已有的系统或进行新的系统设计。

系统组成的基本属性包括系统界面、系统成分、系统环境。系统界面是系统内在成分与系统环境之间的分界线，系统成分是构成系统的内在实体元素，系统环境是影响系统行为的外部因素。

二、模拟研究的尺度

尺度在一切自然科学问题研究中都是很重要的关键因素，一般是空间大小和时间长短的合称。草地农业模拟的尺度具有以下的特性：时间性、空间性、复杂性。其中，复杂性经常趋向于随时间和空间而增加。

模拟的时空尺度决定了适宜模型的选择及模拟方法的采用。大的时空尺度模型一般注重宏观的经验性和描述性，反之，小尺度模型则注重微观的机理性解释。

对于草地植物生长模拟研究来说，时间步长的确定取决于作物生长状态变化的速率，在生长相对快的时期，步长宜短一些；在生长相对慢的时期，步长可选大一些。

三、支持研究

专门服务于草地农业模拟模型的大量研究工作，称为模拟的支持研究。需要从事的农业模拟支持研究主要有已知的因果关系和基本模式两个方面。但是，通常缺乏特定的数量表达或算法程序，如作物器官建成与阶段发育的关系。相对不了解而有待探索的某些过程，称为黑箱，如根系生长与土壤环境及叶片生长的关系。

四、析因法与系数化

草地农业生产过程往往由多个环境因子所控制，包括温度、光照、水分和养分等。在定量化这些因子的相互作用时，是通过单个因子的系数互作进行确定，而非复合因子的多元回归。

析因法的主要特征是以系数的形式来分别建立不同单因子的响应模型或效应因子模型，然后以一定的数学方法来定量这些系数间的互作关系。

系数化将效应因子的特征值一般设定在 0～1。系数互作的计算方法主要有最小法和乘积法。最小法依据最小因子法则，认为系统的表现主要受最小系数的限制。乘积法则依据不同因子的相互作用原则，认为系统的表现同时受多因子的影响，而并非与最小因子的水平呈线性关系。

遗传参数通常是指描述非逆境下种或品种基本遗传性状的一组特征值。一个品种的遗传参数一般以 10～15 个为最适，最多不超过 20 个。

遗传参数既要符合作物或动物生理学的认识和规律，又要为作物或动物育种学家所理解和接受。遗传参数一般依据试验数据通过试错法、最小二乘法决定。有条件，也可

直接通过控制环境下的试验研究获得。

五、草业信息模型模拟的表示方法

对于草业模拟模型构建及分析后，最重要的就是模拟结果的显示和表达，其具体内容一般分为以下四个部分。

（一）系统的界定

系统可用系统的成分、系统的界面和系统的环境来简化描述。

系统输入是影响系统行为而不受系统影响的环境因子，如气象变量等，又称驱动变量。开放系统有一个以上的输入，封闭系统没有输入。

系统输出代表系统的特征和行为，为模拟者所感兴趣。

系统参数是模型成分的特征，通常在模拟中恒定不变，而输入则为变数。

状态变量主要描述系统成分的状况或水平，具动态特征，如生物量。如果状态变量随时间而变，为动态模型。如状态变量受到不同过程的影响而变化，生物量受光合和呼吸作用而变化，称为过程模型或连续模型。

（二）输入输出资料

模型的输入资料以最少为原则，既可容易获得，又可简化模拟运算。例如，作物生长模型的输入资料，总体上可分为气象、土壤、品种、管理四大类。模型的输出要求动态、完整、易于理解，具有先进的可视化及多媒体特点。可利用表格、图形、图像、声音等多种形式来综合实现，结果可同时输出到屏幕、文件、打印机等。输出的步长一般以天为单位。同时，模型输出步长和方式都可由用户根据各自需求而动态设定。

（三）模拟模型的结构成分

任何一个模拟模型均可以根据其自身的特性分解成相互关联的结构成分。例如，整个作物生育及其环境系统一般可分解成 6 个相互关联的亚系统。

第一亚系统为作物的阶段发育与物候期，主要是有关以温光反应为基础的茎顶端发育阶段及以外部形态特征变化为标志的生育时期，如小麦的小穗分化期、小花分化期等，以及分蘖期、拔节期、抽穗期等。

第二亚系统为作物植株的形态发生与器官建成过程，包括根系、叶片、茎秆、小穗、小花、籽粒等器官发生与形成的规律、数量及质量等。

第三亚系统为植株的光能利用与同化物生产，包括叶片和冠层的光合作用、呼吸作用、碳水化合物积累及生物量的计算等。

第四亚系统为不同器官间的物质分配与利用，包括同化物分配系数或分配指数和分配量的实时变化、器官的生长和大小、产量及品质的决定等。

第五亚系统为土壤-植物-大气水分关系，包括土壤水分的移动、吸收、蒸发蒸腾、植物组织的水分平衡等。

第六亚系统为土壤养分（如氮素等）动态与植株利用，包括主要养分元素在土壤中

的转化、根系吸收、体内分配和利用等。

以上各结构成分通过物质和信息的交流联成一个植物生长的动态平衡系统。

（四）概念模型与图示模型表示法

一般的概念模型都具有特定的图示，如图 4-2 所示，小室代表系统中的状态变量，开关代表过程的速率变量，云朵代表系统的输入源，圆圈和箭头代表中间变量，实线箭头表示物质流，虚线箭头表示信息流。

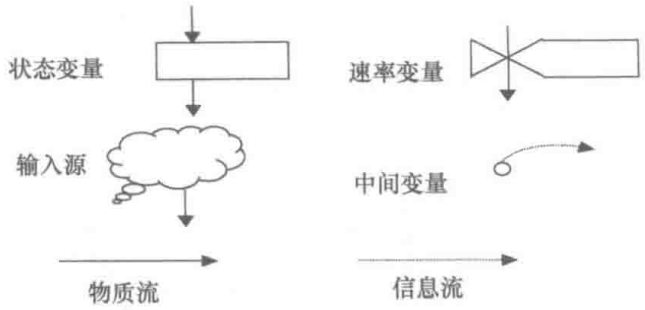


图 4-2　概念模型图

六、模型开发环境与工具

现代模拟模型的建立一般需要以计算机和网络为基础的硬件支持，除此之外还需要以下的开发环境及工具支持（详细内容见第三章）。

1）语言环境：指用于建模模拟编程的计算机语言，经常使用的有 C、Visual Basic、C++、Fortran、Delphi、Java、C#、Python、IDL 等。

2）软件环境：基于计算机等硬件安装的建模所需的专业软件，如 Excel、SPSS、SAS、MATLAB、R、ArcGIS、ENVI、ERDAS 等。

3）系统环境：不同计算机安装有不同的软件操作系统，这在一定程度上决定着专业软件的选择和计算分析的效率。目前，普遍使用的计算机系统有 Windows、Unix、Linux 等。

七、结构化程序设计

在基于计算机技术的模拟模型构建中，结构化程序设计（Structured Programming）是必不可少的部分，它是进行以模块功能和处理过程设计为主的详细设计的基础。该概念最早由 Dijikstra 于 1965 年提出，是软件发展的一个重要的里程碑。其主要观点是采用自顶向下、逐步求精及模块化的程序设计方法；使用三种基本控制结构构造程序，任何程序都有顺序、选择、循环三种基本控制结构构造。结构化程序设计主要强调的是程序的易读性，详细描述处理过程常用三种工具：图形、表格和语言。

（一）主要特点

1）主张使用顺序、选择、循环三种基本结构来嵌套连接成具有复杂层次的“结构化程序”，严格控制 GOTO 语句的使用。用这样的方法编出的程序在结构上具有以

下效果：①以控制结构为单位，只有一个入口，一个出口，所以能独立地理解这一部分；②能够以控制结构为单位，从上到下顺序地阅读程序文本；③由于程序的静态描述与执行时的控制流程容易对应，所以能够方便正确地理解程序的动作。

2）“自顶向下，逐步求精”的设计思想，其出发点是从问题的总体目标开始，抽象低层的细节，先专心构造高层的结构，然后再一层一层地分解和细化。这使设计者能把握主题，高屋建瓴，避免一开始就陷入复杂的细节中，使复杂的设计过程变得简单明了，过程的结果也容易做到正确可靠。

3）“独立功能，单入口单出口”的模块结构，减少模块的相互联系使模块可作为插件或积木使用，降低程序的复杂性，提高可靠性。程序编写时，所有模块的功能通过相应的子程序（函数或过程）的代码来实现。程序的主体是子程序层次库，它与功能模块的抽象层次相对应，编码原则使得程序流程简洁、清晰，增强可读性。

4）主程序员组的组织形式，指程序开发的人员组织方式应采用由一个主程序员（负责全部技术活动）、一个后备程序员（协调、支持主程序员）和一个程序管理员（负责事务性工作，如收集、记录数据和文档资料管理等）为核心，再加上一些专家（如通信专家、数据库专家）和其他技术人员组成小组。这种组织形式突出了主程序员的领导，设计责任集中在少数人身上，有利于提高软件质量，并且能有效地提高软件生产率。

该方法的优点在于：由于模块相互独立，因此在设计其中一个模块时，不会受到其他模块的牵连，因而可将原来较为复杂的问题简化为一系列简单模块的设计。模块的独立性还为扩充已有的系统、建立新系统带来了不少方便，因为我们可以充分利用现有的模块作积木式的扩展。

按照结构化程序设计的观点，任何算法功能都可以通过由程序模块组成的三种基本控制结构的组合（顺序结构、选择结构和循环结构）来实现。

结构化程序设计的基本思想是采用“自顶向下，逐步求精”的程序设计方法和“单入口单出口”的控制结构。自顶向下、逐步求精的程序设计方法从问题本身开始，经过逐步细化，将解决问题的步骤分解为由基本程序结构模块组成的结构化程序框图；“单入口单出口”的思想认为一个复杂的程序，如果它仅是由顺序、选择和循环三种基本程序结构通过组合、嵌套构成，那么这个新构造的程序一定是一个单入口单出口的程序。据此就很容易编写出结构良好、易于调试的程序。

总体而言，结构化程序设计具有以下优点：①整体思路清楚，目标明确；②设计工作中阶段性非常强，有利于系统开发的总体管理和控制；③在系统分析时可以诊断出原系统中存在的问题和结构上的缺陷。

对比它所具有的优点，结构化程序设计的方法也存在不足，主要是：①用户要求难以在系统分析阶段准确定义，致使系统在交付使用时产生许多问题；②用系统开发每个阶段的成果来进行控制，不能适应事物变化的要求；③系统的开发周期长。

（二）基本结构

结构化程序设计包括顺序结构、选择结构和循环结构三种基本结构。

顺序结构：表示程序中的各操作是按照它们出现的先后顺序执行的。

选择结构：表示程序的处理步骤出现分支时，需要根据某一特定的条件选择其中的一个分支执行。选择结构有单选择、双选择和多选择三种形式。

循环结构：表示程序反复执行某个或某些操作，直到某条件为假（或为真）时才可终止循环。在循环结构中最主要的是：什么情况下执行循环，哪些操作需要循环执行。循环结构的基本形式有两种："当型"循环和"直到型"循环。

"当型"循环：表示先判断条件，当满足给定的条件时执行循环体，并且在循环终端处流程自动返回到循环入口；如果条件不满足，则退出循环体直接到达流程出口处。因为是"当条件满足时执行循环"，即先判断后执行，所以称为"当型"循环。

"直到型"循环：表示从结构入口处直接执行循环体，在循环终端处判断条件，如果条件不满足，返回入口处继续执行循环体，直到条件为真时再退出循环到达流程出口处，即先执行后判断。因为是"直到条件为真时为止"，所以称为"直到型"循环。

（三）设计方法

一个复杂问题，肯定是由若干简单的问题构成。模块化是把程序要解决的总目标分解为子目标，再进一步分解为具体的小目标，把每一个小目标称为一个模块。

例如，限制使用 GOTO 语句结构中，结构化程序设计方法的起源来自对 GOTO 语句的认识和争论。肯定的结论是，在块和进程的非正常出口处往往需要用 GOTO 语句，使用 GOTO 语句会使程序执行效率较高；在合成程序目标时，GOTO 语句往往是有用的，如返回语句用 GOTO。否定的结论是，GOTO 语句是有害的，是造成程序混乱的祸根，程序的质量与 GOTO 语句的数量呈反比，应该在所有高级程序设计语言中取消 GOTO 语句。取消 GOTO 语句后，程序易于理解、易于排错、容易维护，容易进行正确性证明。作为争论的结论，1974 年 Knuth 发表了令人信服的总结，并证实了以下观点。

1）GOTO 语句确实有害，应当尽量避免。

2）完全避免使用 GOTO 语句也并非是个明智的方法，有些地方使用 GOTO 语句，会使程序流程更清楚、效率更高。

3）争论的焦点不应该放在是否取消 GOTO 语句上，而应该放在用什么样的程序结构上。其中最关键的是，应在以提高程序清晰性为目标的结构化方法中限制使用 GOTO 语句。

第三节　草业信息模拟模型构建方法

一般情况下，草业模拟模型的建模模拟工作分为以下四步：模型选择与系统定义、资料获取与算法构建（模拟工作的重点和难点）、模块设计与模型实现、模型检验与改进。

一、模型选择与系统定义

首先要弄清模拟研究的目的、水平及对象，以明确模拟研究的范围和成分。通过这

项工作，可以先建立一个描述系统结构与关系的概念模式或概念模型。

如果建模主要是为了研究和机理解释，那么模拟的系统水平和层次就应该低一些，模拟的对象可能包括器官及亚器官。对于一个应用性较强或注重宏观预测的模型而言，研究的系统水平可以高一些，系统的成分简单一些。

二、资料获取与算法构建

资料获取大概有 3 个方面的来源：已有积累、合作、实验支持。已有积累包括自己已有的工作积累或文献资料，其中文献资料主要包括国内外在相关领域所取得的科研成果、出版的专著与教材、科技期刊及学术会议上发表的论文等，以及各地的土壤志、品种志、气象资料等，通过补充实验或支持研究，围绕某个方面获得全新的资料。文献资料主要用于模型的构建。合作途径所获得的资料主要用于模型参数的确定及系统的测试。实验支持研究资料一部分用于模型的构建，另一部分用于模型参数的确定及系统的测试。

三、模块设计与模型实现

首先要选择恰当的编程语言来组织系统，包括模拟算法编程语言和界面编程语言。目前应用比较广泛的模拟算法编程语言主要有 Visual C++、Visual Fortran、R 等宏语言。

四、模型检验与改进

模型的检验包括对模型的敏感性分析、校正、核实、测验等 4 个主要过程。

敏感性分析（Sensitivity Analysis），是对模型灵敏度和动态性的测验，分析模型对主要参数和变量反应的灵敏度，测验模型的结构与过程及系统的成分。结果通常以正负值来表示模型的反应程度。

校正（Calibration），是调整模型的参数和关系，使得模型符合模拟者特定的环境和资料参数，主要检验模型系统的综合表现及对综合变量的反应。

核实（Validation），是指决定模型是否适用于模型研制以外的完全独立的资料。如果是多年、多点、多试验观测值与模拟值的比较，可采用如下方法：①将模拟结果与实际结果进行回归分析；②将实际结果与模拟结果按同一时间坐标绘 1∶1 图进行比较；③检验模拟值与实际值的平均误差。

测验（Test），是比较各种环境下的模拟值与预测值，可看作一个持续的模型核实过程。如果在测验过程中发现明显的偏差，可能还得重复上述模型校正和核实的过程，并对模型算法进行必要的修订和改进。

第四节　草业信息模拟模型的应用

前面我们重点介绍了草业模拟模型的构建方法和流程，下面我们对草业模拟模型在

实际中的主要应用进行分类介绍，并归纳研究中与之相结合的其他主要技术。

一、草业信息模拟模型的应用领域

当前，草业模拟模型的应用呈现多样化趋势，下面我们具体举例说明。

（一）牧草生长发育的模拟

作物或牧草生长发育模拟的基础是植物种的生理生长模型。例如，紫花苜蓿的生长发育可用如下模型进行模拟（白文明和包雪梅，2002）。

$$\mathrm{RDS}=\begin{cases}(\mathrm{DTT}-Tu)/\mathrm{PRET}, & \mathrm{RDS}\leqslant 1\\ (\mathrm{DTT}-\mathrm{PRET}-Tu)/\mathrm{POSTT}+1, & 1<\mathrm{RDS}\leqslant 2\\ (\mathrm{DTT}-\mathrm{PRET}-\mathrm{POSTT}-Tu)/\mathrm{Phou}, & \mathrm{RDS}>2\end{cases}$$

$$\mathrm{DTT}=\begin{cases}0 & T_{\mathrm{mean}}<T_{\mathrm{base}}\text{或}T_{\mathrm{mean}}>T_{\mathrm{max}}\\ \sum(T_{\mathrm{mean}}-T_{\mathrm{base}})T_{\mathrm{base}} & \leqslant T_{\mathrm{mean}}\leqslant T_{\mathrm{max}}\end{cases}$$

式中，RDS 为作物的发育指数，DTT 为播种开始即统计的有效积温（℃），Tu 为播种到出苗所需的有效积温（℃），PRET 为出苗到开花所需的有效积温（℃），POSTT 为开花到种子成熟的有效积温（℃），Phou 为种子成熟到作物停止生长所需的有效积温（℃），T_{base} 和 T_{max} 分别为作物生长所需的最低温度和最高温度（℃）。

（二）草地净初级生产力的模拟

草地净初级生产力（NPP）是草地农业系统的基础，也是草地研究中最重要的组分之一。净初级生产力估算模型总体框架如图 4-3 所示，而模型中估算的 NPP 是植物吸收的光合有效辐射（APAR）与实际光能利用率（ε）共同作用来决定（朱文泉等，2007）。

$$\mathrm{NPP}(x,t)=\mathrm{APAR}(x,t)\times\varepsilon(x,t)$$

式中，APAR(x, t)表示像元 x 在 t 月吸收的光合有效辐射[gC/(m^2·month)]，$\varepsilon(x, t)$表示像元 x 在 t 月的实际光能利用率（gC/MJ）。

与 NPP 相关的模型有很多，有确定净初级生产力因素的模型，还有阐明与净初级生产力关系的模型，如下面介绍的三个模型（Lin et al.，2013）。

1. 分类指数模型

通过链式法则构建的分类指数模型起初是用作带有公用变量（放射性干燥指数）的生理生态特征及一个地区的土壤水分蒸发蒸腾损失总量模型。分类指数模型阐明了 NPP 值取决于生长度日值（GDD）与水分指数 $L(K)$的关系函数。

当 $L(K)=0.58802K^3+0.50698K^2-0.0257081K+0.0005163874$ 时，

$$\mathrm{NPP}=L^2(K)\cdot\frac{0.1\cdot\mathrm{GDD}\cdot[K^6+L(K)K^3+L^2(K)]}{[K^6+L^2(K)][K^5+L(K)K^2]}\times\mathrm{e}^{-\sqrt{13.55+3.17K^{-1}-0.16K^{-2}+0.0032K^{-3}}}$$

上式表示的是根据综合顺序分类法（CSCS）的综合分类指数估计净初级生产力的方法。

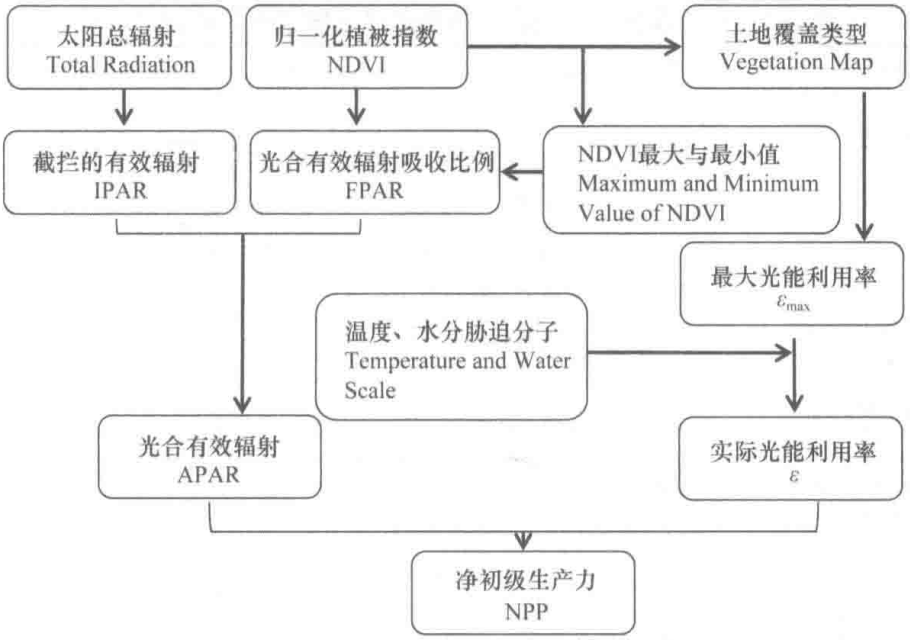


图 4-3　净初级生产力估算模型总体框架

2.MIAMI 模型

MIAMI 模型是第一个也是最广为人知的将 NPP 与降水量及温度联系在一起的模型。MIAMI 模型提供了比较的基线。作为一个以实验为基础的模型，MIAMI 模型将 NPP 与年平均温度（℃）和年降水总量 P（mm）联系在一起，其表达形式如下式

$$\mathrm{NPP} = \min(\mathrm{NPP}_T, \mathrm{NPP}_P)$$

式中，$\mathrm{NPP}_T = \dfrac{3000}{1+\mathrm{e}^{1.315-0.119T}}$，$\mathrm{NPP}_P = 3000(1-\mathrm{e}^{-0.000\,64P})$。

3. Shuur 模型

Shuur 利用关于年平均温度（MAT）和年降水总量（MAP）的指数函数估算 NPP，在这个模型中 NPP 是由两个方程中预测的最小值来定义的。与 MIAMI 模型最主要的区别在于 NPP 值在降水量方程中 MAP 为 2200mm 达到顶峰，之后当 MAP 增加到 8000mm 时，开始逐渐降低。其表达形式如下

$$\mathrm{NPP} = \min(\mathrm{NPP}_{\mathrm{MAP}}, \mathrm{NPP}_{\mathrm{MAT}})$$

式中，$\mathrm{NPP}_{\mathrm{MAT}} = \dfrac{17.6243}{1+\mathrm{e}^{(1.3496-0.0715\mathrm{MAT})}}$，$\mathrm{NPP}_{\mathrm{MAP}} = \dfrac{0.005\,212\mathrm{MAP}^{1.123\,63}}{\mathrm{e}^{0.000\,459\,532\mathrm{MAP}}}$。

（三）草地生物量与可食牧草产量的模拟

草地生物量包括地上与地下两部分，基于遥感技术的统计模型一般可以用回归方法模拟地上部分，地下部分的生物量因其物种多样性导致根部形态变化差异大而不易模拟，但可以通过根冠比的方法大致近似模拟或类比（梁天刚等，2009）。

近年来，Liang 等（2016）通过多因子回归分析建立了青南牧区地上生物量估测模型（表 4-1）。

表 4-1 青南牧区草地地上生物量多因子估测模型的精度比较

主要因子	模型公式	RMSE	r
地理位置与草地盖度	$Y=e^{-42.759}x^{15.317}y^{-7.684}c^{1.535}$	736.27**	0.7851
	$Y=233.221x-565.118y+16.042c-3411.94$	726.91**	0.7667
地理位置与草层高度	$Y=e^{-14.864}x^{12.499}y-10.293h^{0.431}$	733.99**	0.7871
	$Y=178.876x-689.295y+59.223h+6881.767$	694.83**	0.8113
草地生物物理指标	$Y=e^{-4.204}h^{0.658}c^{2.221}$	682.83**	0.8187
	$Y=77.596h+35.917c-2486.03$	700.70**	0.8077
地理位置与草地生物物理指标	$Y=e^{-18.358}x^{6.784}y^{-3.949}h^{0.493}c^{1.615}$	665.01**	0.8292
	$Y=110.615x-406.994y+62.911h+19.6c+2072.057$	661.04**	0.8310

注：Y 是草地地上生物量（kg DW/hm^2），x、y、h 和 c 分别是精度（°）、纬度（°）、草高（cm）、盖度（%）

** 表示 $P<0.001$

（四）草地生态服务价值的模拟

进行草地生态服务价值评估，需要综合考虑草地生态系统现状、草地资源的空间异质性及其在区域社会经济发展中的重要性等因素，将 3S 应用到草地生态服务功能的动态监测中，针对草地生态系统特点，构建草地生态服务功能与价值评价体系和综合评估方法（刘兴元等，2011）。草地生态系统服务价值评估流程见图 4-4。

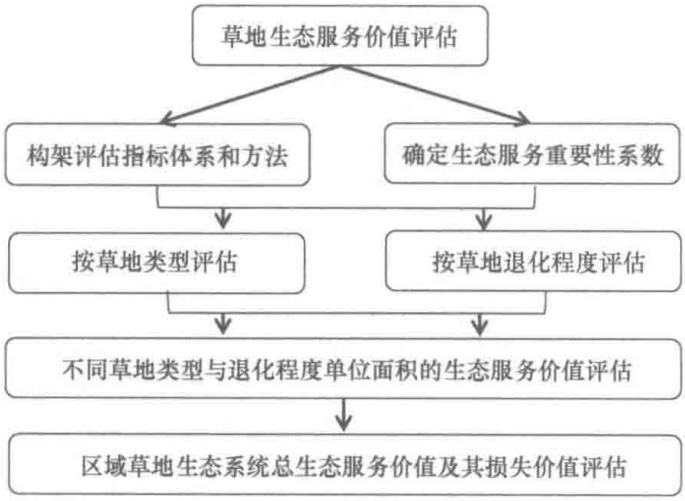


图 4-4 草地生态系统服务价值评估流程

将水分涵养、养分循环、侵蚀控制（土壤保持量）、废物处理、生物多样性保护、释放 O_2、固定 CO_2、消减 SO_2、滞留沙尘、休闲旅游、文化传承、畜牧生产等指标的单位面积生态服务价值累加后，乘以生态服务重要性系数，得出某一草地类型单位面积的生态服务价值，然后把各类型草地生态服务价值合计，得出某一区域的总生态服务价值。

$$P_{ij}=\sum_{x=1}^{12}Vx_{ij}\times E_i$$

$$GV_{ij}=\sum_{j=1}^{n}P_{ij}\times A_{ij}$$

式中，GV_{ij}为某一区域j类草地的总生态服务价值（元），P_{ij}为某一区域j类草地单位面积的生态服务指标的价值（元/hm^2），V_{xij}为草地生态系统服务功能评价指标单价（元），E_i为某一区域的草地生态服务功能重要性系数，A_{ij}为某一区域j类草地的面积（hm^2），i代表区域，j代表草地类型，x代表草地生态系统功能评价指标。

（五）草地养分效应的模拟

草地中N、P等元素可以反映草地的营养程度。土壤中有机碳的含量和组成不仅表明土壤有机质（土壤有机碳）的水平，而且还能够说明营养元素N、P等的有效状态，因为土壤有机碳含量与土壤全氮、全磷和速效氮的含量呈明显的正相关。因此，对草地中有机碳的含量进行模拟，是研究草地养分效应的有效手段。

利用数学模型模拟草地土壤中有机碳储量，可以对离散数据在时间和空间上进行综合分析。根据大量的实测数据及气候变化模拟数据，综合考虑进入土壤的有机碳的数量和决定土壤有机碳分解速率的各种因子，通过数学模型估算草地生态系统的土壤有机碳储量，既可以预测不同情境下土壤有机碳储量的变化趋势，讨论土壤中有机碳的储积及其固定潜力，又可以研究由于气候变化造成的土壤有机碳的综合影响。出于研究目的及研究内容的不同，目前已构建了多种类型的模型，其中有机理过程模型、统计估计模型、相互关系模型，以及基于遥感数据源的遥感模型（任继周和林慧龙，2013）。例如，李东（2011）应用CENTURY模型（一种统计估计模型）模拟了高寒草甸土壤有机碳的动态。

在CENTURY模型中，每个状态遵循遵从以下模型

$$\frac{\mathrm{d}C_i}{\mathrm{d}t} = K_i \cdot M_d \cdot T_d \cdot C_i$$

式中，C_i为不同状态下的碳含量，i=1, 2, 3, 4, 5, 6, 7分别为地表结构库和代谢碳库、土壤结构库和代谢碳库、活性土壤有机碳库、缓性土壤有机碳库和惰性土壤有机碳库，K_i为第i状态下的最大分解速率，M_d为月降水量，T_d为月平均土壤温度对分解的影响。

（六）草地水分效应的模拟

研究草地的水分利用和平衡是保障草地畜牧业稳定发展的重要课题，模拟并预测水分波动对草地群落的影响对于维持草地生态系统的稳定具有重要意义。例如，可利用作物生产力模型（EPIC0509）对作物轮作产量和土壤水分效应进行模拟（王学春等，2011a，2011b）。

依据干燥化指数，将黄绵土的干燥化划分为6个等级

$$SDI = \frac{SM - WM}{SSM - WM}$$

式中，SDI为土壤干燥化指数，WM为土壤凋萎湿度（%），SSM为土壤稳定湿度（%）。

土壤有效含水量的计算公式如下

$$ASW = \sum_{j=1}^{10} ASW_j$$

$$ASW_i = (SW_i - WP_i) \times P_i \times H_i \times 10$$

式中，ASW 为 0～10cm 土层的有效含水量（mm），ASW_i 为第 i 土层的有效含水量（mm），SW_i 为第 i 土层的土壤湿度（%），WP_i 为第 i 土层的土壤萎蔫湿度（%），P_i 为第 i 土层的土壤容重（g/cm^3），H_i 为第 i 土层的土层厚度（cm）。

（七）草畜平衡动态的模拟

草畜平衡是指在一定区域与时间内通过草原和其他途径提供的饲草饲料量与饲养牲畜所需的饲草饲料量达到动态平衡。通常认为植被对放牧压力的反应是线性的、可逆的，因此草地植被动态对放牧强度的反应是可预测的。根据不同放牧压力下草地植被的反应，可构建相关模型，模拟其动态演化（梁天刚等，2011）。

$$Za^* = \frac{1}{t} \times \sum_{k=1}^{t} Za$$

$$Za = \frac{\sum_{i=1}^{n} (A_i \times y_i \times k_{i,1} \times k_{i,2} \times k_{i,3})}{1 \times D}$$

$$Za'' = \frac{1}{t} \times \sum_{k=1}^{t} Za'$$

$$Za' = \frac{\sum_{j=1}^{m} \sum_{i=1}^{n} (A_i \times y_{ij} \times k_{i,1} \times k_{i,2} \times k_{i,3})}{I \times D}$$

式中，Za^*代表各类型草地总的适宜载畜量上限（标准羊单位），Za 为某一年各类型草地总的最大理论载畜量（标准羊单位），k 代表监测年份；Za''为各类型草地总的适宜载畜量下限（标准羊单位），i 代表不同的草地类型，j 代表牧草生长季的月份（5～10 月），Za'为某一年各类型草地在牧草生长季各月最大理论载畜量的平均值，A_i 为第 i 类草地面积（hm^2），y_i 为第 i 类草地在某一年的年最大牧草单产（kg/hm^2），y_{ij} 为第 i 类草地在某一年的牧草生长季各月最大牧草单产的平均值（kg/hm^2），$k_{i,1}$、$k_{i,2}$、$k_{i,3}$ 分别为第 i 类草地的可利用面积系数、可食牧草系数和可食牧草放牧利用率，I 为一只标准成年绵羊日采食量（kg/d），D 为草地放牧利用的天数。

二、模拟模型与其他技术的耦合

通过上面对草业模拟模型主要应用的归纳可以看出，当前草业模拟模型与以下技术的结合形成了草业模拟模型的主要特征，具体如下。

（一）模拟模型与地理信息系统技术的结合

多尺度模拟模型的数据越来越依赖于地理信息系统（GIS）技术的支持，分析方法也与 GIS 更多的相互结合和集成。将 GIS 用于一个环境模型框架中，模型可在一个单独的集成环境下使用数据库、数据可视化和 GIS 的强大分析工具。下面是一个 GIS 与综合顺序分类模型集成的案例（http://cscs.ecograss.com.cn/）（图 4-5）。

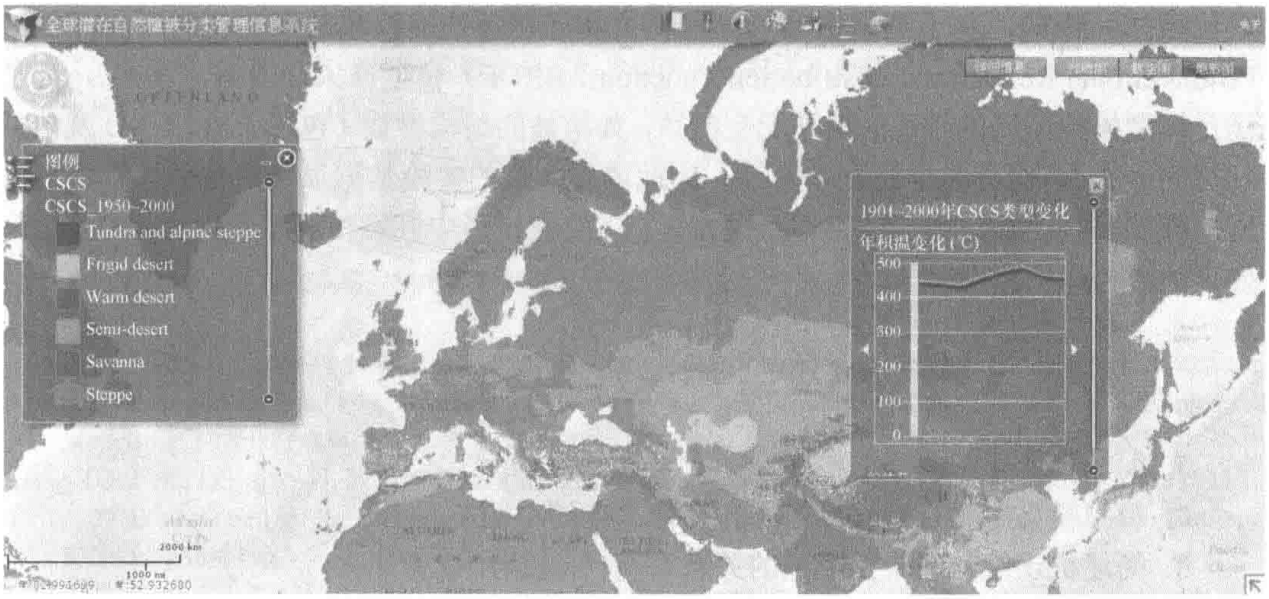


图 4-5　全球潜在自然植被分类管理系统界面（Liang et al.，2012a，2012b）

（二）模拟模型与遥感技术的结合

遥感作为连续时间尺度及不同空间尺度的有效数据源已成为现代草业模型模拟研究的基础之一。将遥感同化到作物生长模型模拟过程中已成为作物生长模型区域化应用的一种有效途径。

例如，利用耦合模型 WOPRO-SAIL 模拟得到的植被冠层反射率计算光谱指数 SAVI′，通过同化由 CCD 影像获取的光谱指数 SAVI，以及最小化 SAVI 和 SAVI′之间的差值（温兆飞等，2012）。在此过程中，引入粒子群优化算法（Particle Swarm Optimization，PSO），运行 WOFOST 软件不断调整待优化参数值（IDTR 和 TSUMST），从而调整 WOPRO-SAIL 模拟得到的 SAVI′，使得其与 SAVI 值不断收敛，直至其差值最小，然后将调整后的待优化参数值作为作物生长模型的初始值，运行 WOFOST，模拟得到较为准确的水稻生长动态过程，从而建立田间尺度遥感-作物模拟同化框架模型 RS-WOPRO-SAIL。

SAVI 计算公式和代价函数表达式分别为

$$\mathrm{SAVI}=\frac{\rho_{\mathrm{nir}}-\rho_{\mathrm{red}}}{\rho_{\mathrm{nir}}-\rho_{\mathrm{red}}+0.5}(1+0.5)$$

$$Q=\frac{1}{P}\sqrt{\sum_{i=1}^{i=P}(x_i-y_i)^2}$$

式中，ρ_{nir} 和 ρ_{red} 分别代表近红外波段（4 波段）和 CCD 影像的红光（3 波段），Q 为代价函数值，P 为外部同化数据个数，x_i 为利用 WOPRO-SAIL 模型模拟的某一时间点植被冠层反射率计算得到的 SAVI′，y_i 为利用某一时间点的 CCD 影像计算得到的 SAVI。

（三）模拟模型与实验验证技术的结合

任何理论模型的模拟都需要与实验验证有效结合，从而得出更符合客观世界规律的

理论模式。例如，基于测量植被的结构特征和光谱特征的植被的双向反射率分布函数（Bidirectional Reflectance Distribution Function，BRDF）模型（宋金玲等，2009），为了满足模型模拟需提供必要的植被配套参数，如植被的结构参数（包括 LAI、LAD 及植被的空间分布）、组分光学特性参数等，因此需要大量的实验数据以确定参数。模拟之后同时需要实测数据与模拟结果进行比较矫正，完善模型中假设相对理想化的部分，达到需要的模拟现实精度。

思 考 题

1. 简述草业模拟模型的概念？
2. 模拟模型主要有哪些分类，各有什么样的特征？
3. 简述机理性与经验性模型的关系？
4. 论述草业模拟模型在草业科学中的应用？
5. 草业模拟模型能与哪些技术有效结合？

参 考 文 献

白文明, 包雪梅. 2002. 乌兰布和沙区紫花苜蓿生长发育模拟研究[J]. 应用生态学报, 13(12): 1605-1609.

冯琦胜, 高新华, 黄晓东, 等. 2011. 2001-2010 年青藏高原草地生长状况遥感动态监测[J]. 兰州大学学报(自然科学版), 47(4): 75-81.

黄敬峰, 王秀珍, 王人潮, 等. 2001. 天然草地牧草产量遥感综合监测预测模型研究[J]. 遥感学报, 5(1): 69-74.

李东. 2011. 基于 CENTURY 模型的高寒草甸土壤有机碳动态模拟研究[D]. 南京: 南京农业大学博士学位论文.

李文龙, 苏敏, 李自珍. 2010. 甘南高寒草地放牧系统生态风险的 AHP 决策分析及管理对策[J]. 草业学报, 19(3): 22-28.

梁天刚, 陈全功, 任继周. 2002. 甘肃省草业开发专家系统的结构与功能[J]. 草业学报, 11(1): 70-75.

梁天刚, 崔霞, 冯琦胜, 等. 2009. 2001-2008 年甘南牧区草地地上生物量与载畜量遥感动态监测[J]. 草业学报, 18(6): 12-22.

梁天刚, 冯琦胜, 夏文韬, 等. 2011. 甘南牧区草畜平衡优化方案与管理决策[J]. 生态学报, 31(4): 1111-1123.

刘文, 王恩利, 韩湘玲. 1992. 棉花生长发育的计算机模拟模型研究初探[J]. 中国农业气象, 13(6): 10-16.

刘兴元, 梁天刚, 郭正刚, 等. 2003. 阿勒泰地区草地畜牧业雪灾的遥感监测与评价[J]. 草业学报, 12(6): 115-120.

刘兴元, 龙瑞军, 尚占环. 2011. 草地生态系统服务功能及其价值评估方法研究[J]. 草业学报, 20(1): 167-174.

任继周, 林慧龙. 2013. 草地土壤有机碳储量模拟技术研究[J]. 草业学报, 22(6): 280-294.

宋金玲, 王锦地, 帅艳民, 等. 2009. 像元尺度林地冠层二向反射特性的模拟研究[J]. 光谱学与光谱分析, 29(8): 2141-2147.

王学春, 李军, 方新宇, 等. 2011a. 半干旱区草粮轮作田土壤水分恢复效应[J]. 农业工程学报, 27(1): 81-88.

王学春, 李军, 方新宇, 等. 2011b. 黄土高原半干旱偏旱区草粮轮作田土壤水分恢复效应模拟[J]. 应用

生态学报, (01): 105-113.

温兆飞, 张树清, 白静, 等. 2012. 农田景观空间异质性分析及遥感监测最优尺度选择——以三江平原为例[J]. 地理学报, 67(3): 346-356.

张峰, Zhou GS, 王玉辉. 2008. 基于 CASA 模型的内蒙古典型草原植被净初级生产力动态模拟[J]. 植物生态学报, 32(4): 786-797.

张钟军, 孙国清. 2005. 用双矩阵法和积分方程模型估算地表的亮度温度[J]. 遥感学报, 9(5): 531-536.

朱文泉, 潘耀忠, 张锦水, 等. 2007. 中国陆地植被净初级生产力遥感估算[J]. 植物生态学报, 31(3): 413-424.

Accad A, Neil DT. 2006. Modelling pre-clearing vegetation distribution using GIS-integrated statistical, ecological and data models: A case study from the wet tropics of northeastern Australia[J]. Ecological Modelling, 198(1-2): 85-100.

Liang TG, Feng QS, Cao JJ, et al. 2012a. Changes in global potential vegetation distributions from 1911 to 2000 as simulated by the comprehensive sequential classification system approach[J]. Science Bulletin, 57(11): 1298-1310.

Liang TG, Feng QS, Yu H, et al. 2012b. Dynamics of natural vegetation on the Tibetan Plateau from past to future using a comprehensive and sequential classification system and remote sensing data[J]. Grassland Science, 58(4): 208-220.

Liang T, Yang S, Feng Q, et al. 2016. Multi-factor modeling of above-ground biomass in alpine grassland: A case study in the Three-River Headwaters Region, China[J]. Remote Sensing of Environment, 186: 164-172.

Lin HL, Feng QS, Liang TG, et al. 2013. Modelling global-scale potential grassland changes in spatio-temporal patterns to global climate change[J]. International Journal of Sustainable Development & World Ecology, 20(1): 83-96.

第五章　草业基础数据库构建

数据库技术是计算机科学的重要分支之一，也是草业信息学的重要组成部分。学习数据库技术，就是学习如何利用数据库技术研究草业科学、草业信息学所涉及的关系型数据的录入、查询、处理、分析及管理等方面的问题。

自 1968 年世界上第一个信息管理系统诞生以来，数据库技术得到了迅猛的发展。伴随着计算机软硬件的不断革新，网络技术的快速发展，使得数据库技术在各行各业都得到了广泛深入的应用。近年来，随着云计算、大数据等概念的提出、发展和应用，使得数据库技术面临新的挑战和发展机遇。本章主要介绍数据库技术的常用术语和基本概念、关系数据库及其范式、结构化查询语言及其使用方法、草业数据库和大数据及其在草业科学研究和实践中的应用。

第一节　草业基础数据库的概念与特征

草业数据库是数据库技术在草业科学领域中的应用。从数据内容看，草业数据库主要包括草地资源信息数据库、草业生产资源信息数据库、草业科学知识和技术信息数据库、草业政策法规数据库等。本节从传统数据库概念入手，主要介绍数据库的基本概念和结构、关系型数据库及结构化查询语言基本概念和应用实例等。

一、数据库的概念与结构

（一）信息和数据

在草业科学领域，我们经常需要定量或定性地描述草业生态系统中某一组分的存在方式或状态，以及相互联系，如土壤、草地、家畜、牧草等。对于土壤，可以用土壤类型、面积、质地、养分含量等信息进行描述；对于草地，可以用草地的类型、面积、优势种、载畜量、分布范围等信息进行描述；对于家畜，可以用家畜种类、品种、数量、生产能力等信息进行描述；对于牧草，可以用种、品种、形态、分布、饲用价值等信息进行描述。在上述例子中，我们可以认为土壤、草地、家畜和牧草是我们关注的对象，而对它们特征的具体描述需要属性数据。同时，当我们需要说明某一事物（实体）时，通常说明的是它的“特征”，而且同一类型的事物往往可以通过一些相同的特征来描述，这样就形成了对该事物的记录。

数据库中的数据就是按照一定的特征（字段）来描述或记录实体的。一条记录（一行）代表一个实体，而一张表记录一种类型的实体。一般情况下，数据库由多张数据表组成。

（二）数据库和数据库管理系统

数据库（Database）是由一张张记录不同实体的表组成的，简单地说，数据库就是表的集合。严格来说，数据库是指按一定的组织结构长期储存在计算机中可共享的大量数据的有机集合。

数据库管理系统（Database Management System，DBMS）是专门用于数据库管理的计算机软件。其核心目标是方便用户对数据的操作，如实现数据库的创建、数据录入、查询和维护等。著名的数据库管理系统有 Oracle、Sybase、MySQL、Microsoft SQL Server 等，Microsoft Office 也提供了小型的数据库管理系统 Access。数据库管理系统一般具有如下功能：①数据定义功能；②数据操作功能；③数据库的运行和管理功能；④数据库的建立和维护功能。

（三）数据库系统

有了数据库，我们就要使用它，而使用数据库的软件程序就是数据库系统（Database System，DBS），是一个实际可运行的软件系统，通常由软件、数据库和数据库管理员组成（曹卫星，2005）。

由此可见，数据库系统是一个按一定结构来存储和管理数据的计算机软件系统，是逻辑上相关的数据的集合。数据库系统由数据库、数据库管理系统、数据库应用程序、用户组成。

1）数据库：依据某种结构组织并存储于计算机中的数据集合。

2）数据库管理系统：位于用户和数据库之间的数据管理软件。用户对数据的任何操作，如数据定义、查询、维护、运行控制等都是通过数据库管理系统进行的，应用程序对数据库的访问也必须通过数据库管理系统。

3）数据库应用程序：为了完成某些特定任务而专门开发的运行在操作系统之上的计算机程序。数据库应用程序是数据库管理系统实现对外提供数据和信息服务的唯一途径。通过数据库应用程序，可以使用户对数据库的字段和记录进行插入、修改、删除等操作。

4）用户：大体可分为开发人员、管理人员和应用人员，是数据库系统的使用者。

二、关系数据库

（一）关系数据库概述

1970 年美国 IBM 公司 San Jose 研究室的研究员 E. F. Codd 在 Communications of ACM 上发表了题为 *A Relational Model of Data for Large Shared Data Banks* 的论文，首次明确而清晰地提出了一个新的数据库系统模型，即关系模型。

关系数据库（Relational Database，RDB）是指基于关系模型的数据库。由于关系模型既简单又有较坚实的数学基础，因此自 1970 年被提出以来，逐步取代了基于层次模型和网状模型的数据库，是目前使用最广泛的数据模型。

关系模型是一种用二维表表示实体集，用主键标示实体、外键标示实体间联系的数据模型。所以，关系数据库的存储结构是多个二维表格。每个二维表称为一个关系（Relation），每一行称为一个元组（Tuple）或记录（Record），用来描述一个对象（实体）；每一列称为一个属性（Attribute），用来描述一个对象的属性，每个属性都有一个取值范围，称为属性的值域（Domain），表格中每个单元格的内容称为属性值（Value）。在一个关系（表格）中，属性名不能重复。有时，也称关系为表（Table），元组为行（Row），属性为列（Column）。在关系模型中，无论是实体还是实体之间的联系，均采用单一的数据结构即关系（二维表）来表示（陈晓云和徐玉生，2009）。

关系模型的优点是数据结构简单、格式唯一、理论基础严格、数据表之间相互独立。

例如，表 5-1 为高寒草地采样信息，其中每条记录（元组）代表一个样地的相关信息。在这个关系中，有样地编号、采样日期、草地类型、草地利用现状、草地盖度和采样地点等属性，反映各样地不同方面的信息。而且这些信息都有特定的值域，如草地类型的值必须是基于某一草地分类系统的类名称；草地利用现状是围栏封育或者自由放牧，草地盖度的取值为介于 0～100 的实数等。

表 5-1　高寒草地采样信息

样地编号	采样日期	草地类型	草地利用现状	草地盖度	采样地点
2014001	20150801	高寒草甸	围栏封育	90	夏河县
2014002	20150801	高寒草甸	自由放牧	85	夏河县
2014003	20150803	高寒灌丛草甸	围栏封育	95	玛曲县

一般可以采用如下格式表示一个关系：R（D_1，D_2，…，D_n），其中 R 表示关系的名称，D_1，D_2，…，D_n 表示属性。例如，上例可以表示成：高寒草地采样信息（样地编号、采样日期、草地类型、草地利用现状、草地盖度、采样地点）。

由上述例子可以看出，表 5-1 表示的关系与传统的二维表格数据文件有类似之处，但是它们又有区别，严格地说，关系是一种规范化的二维表格，它具有如下性质。

1）属性值不可再分解，即每一列不能再被分解成几列。

2）关系中不存在重复的元组（记录）。

3）理论上各元组间没有先后次序，但使用时可以有行序。

在关系数据库中，对每个指定的关系，经常需要根据某些属性的值来唯一地操作一个元组，也就是说要通过某一个或几个属性唯一地标识一个元组，把这样的属性或属性组称为指定关系的键（Key）。在一个关系中，如果通过一个属性集合的取值就能唯一确定每一个元组，则称该属性集合为该关系的超键（Super Key），有时也称超码。例如，表 5-1 的高寒草地采样信息关系中，属性集合（样地编号）和属性集合（样地编号，采样日期）都能够唯一地确定该关系中的元组，因此属性集合（样地编号）和属性集合（样地编号，采样日期）都是该关系的超键。

如果某一属性集合是超键，但去掉其中任意属性后就不再是超键，则称该属性集合为候选键。或者说某一属性集能够唯一标识关系的元组而又不包含多余的属性，那么这个属性集称为候选键。

通常数据库设计时会选择一个候选键来区分不同的元组，选中的候选键称为主键（Primary Key），或称为主关键字，或主码。一个关系中只有一个主键。如表 5-1 的高寒草地采样信息关系中，“样地编号”就是该关系的主键。一般习惯表示成：高寒草地采样信息（样地编号、采样日期、草地类型、草地利用现状、草地盖度、采样地点）。

如果关系 A 中的某一属性 K，是另一关系 B 的主键，则称 K 是关系 A 参照 B 的外键（Foreign Key），或外关键字。例如，表 5-2 所示的采样地点相关信息关系，其中“采样地点”属性是其主键，那么在表 5-1 所示的高寒草地采样信息关系中，“采样地点”属性就是高寒草地采样信息参照采样地点相关信息的外键，通过“采样地点”这个外键可以将高寒草地采样信息和采样地点相关信息这两个关系联系起来。

表 5-2　采样地点相关信息

采样地点	行政分区编号	面积/km^2
夏河县	623027	8 037
玛曲县	623025	10 190

为了维护数据的一致性，关系数据库的数据与更新操作必须遵循以下三类完整性规则：实体完整性、参照完整性和用户自定义完整性（陈晓云和徐玉生，2009）。

a. 实体完整性规则：要求关系中元组的主键的属性不能为空值。即每一个元组（记录）r 是被唯一标识的。每一个元组都对应一个现实的实体，而现实实体是确定的，并且是可以被区分的，具有唯一性。这就要求记录实体的元组也具有唯一性和可区分的特征。要确保元组的唯一，那就要求元组的主键的属性唯一且不为空。实体完整性就是对现实实体的可区分性和唯一性进行约束的规则。

b. 参照完整性规则：如果属性集 K 是关系 A 的主键，K 也是另一个关系 B 的外键，那么，在关系 B 中，K 的取值只能是空值或是关系 A 中的主键 K 的已有取值。例如，在上例中，高寒草地采样信息关系中“采样地点”属性的取值只能是采样地点相关信息关系中“采样地点”主键的某一个值（夏河县或玛曲县）或是空值，不能出现其他数值。关系的参照完整性反映了主键属性和外键属性之间的引用规则。一般，主键属性和外键属性可以使用不同的名字，但属性的定义域必须相同。上例中采样地点相关信息关系中“采样地点”属性也可以称“县名称”。

c. 用户定义完整性规则：当实体完整性规则和参照完整性规则不能满足需求时，用户可根据需要定义一些约束条件，如规定草地盖度的取值必须是整数。

（二）关系模型规范化

设计任何一种数据库，不管是层次模型、网状模型还是关系模型，都会遇到数据规范化的问题，由于关系模型有严格的数学基础，且可以向其他模型转换，因此通常借助关系模型来研究数据规范化的问题。

如果数据库设计过程中不考虑规范化问题，可能会导致信息重复，或某些信息无法录入系统等问题。可以通过如下示例理解这一问题。

假设有一个关系：草地面积信息（编号，省名称，地区名称，市县名称，草地类名

称)。如果我们要给这个关系添加一个元组(记录),如甘肃省甘南州玛曲县高寒灌丛草甸类,我们需要将甘肃省、甘南州、玛曲县、高寒灌丛草甸类分别加入关系,但玛曲县不只有高寒灌丛草甸一个草地类,它还有高寒草甸类和沼泽类草地,如需要添加这2种草地类,那么就需要再次录入甘肃省、甘南州、玛曲县等信息,即需要将省名、地区名、市县名的内容再重复2遍。考察这个关系可以看出,一个县可能有多个草地类。因此,在这个关系中,省名称、地区名称、市县名称属性将被大量重复。

该关系的另一问题是,除非已录入了某一个县的一种草地类,否则在该关系中将不包含该县的信息。因为当我们想要录入某一个县时,必须填写这个县的某一种草地类名称。

再举一个例子,如有这样一个关系:草地鼠害信息(编号,草地类名称,草地面积,害鼠名称,防治方法)希望存储某一地区草地类鼠害信息数据。那么,分析这个关系,可以发现在使用的过程中可能会面临如下问题。

1)存储冗余:在这个关系中,如果一种草地类有多种害鼠,那么草地类名称和面积信息将会重复存储。如果某种鼠害在多类草地上都有分布,那么害鼠名称、鼠害防治方法等属性将大量重复。

2)插入异常:只有当一类草地上有鼠害时,草地类名称的相关信息才能录入到这个关系中;否则,假设某类草地上没有发现鼠害,这样害鼠名称和防治方法都为空,就无法插入。

3)更新异常:这个关系中存在大量的数据冗余,给数据更新带来很大的代价。例如,希望修改由高原鼢鼠导致的鼠害防治方法,那么就需要同时修改所有害鼠名称为高原鼢鼠的记录,否则会导致数据的不一致,即同一种鼠害可能出现多种防治方法。同理,如果我们希望修改某类草地的面积,那么该类草地所对应的所有记录都要修改。

4)删除异常:假设由于防治取得较好的效果,我们希望删除某类草地上高原鼢鼠的鼠害记录,但会发现如果该草地只有这一条记录(即该草地类只有这一种鼠害),那么,这个草地类的名称、面积等信息也会同时被删除。

由此可见,草地鼠害信息这个关系设计是不合理的。我们可以尝试对这个关系做如下分解:

草地信息(草地类名称,草地面积)

害鼠信息(害鼠名称,防治方法)

草地鼠害信息(编码,草地类名称,害鼠名称)

比较分解前后的关系可以看出:

1)存储冗余分析:通过分解,降低了3个关系的数据冗余,仅在草地鼠害信息关系中存在一定的冗余,但这些冗余是合理的,也是不可避免的。

2)插入异常分析:由于将鼠害信息和草地信息分解成不同的关系,因此有效地避免了插入异常。如需要添加草地类信息,只需要操作草地信息关系;而要添加害鼠信息,则只需要操作害鼠信息关系。如果需要添加某种草地类上的鼠害信息,则只需要在确保害鼠信息关系中有该害鼠记录的前提下,在草地鼠害信息关系中添加相应记录即可。

3)更新异常分析:经过分解,草地信息和鼠害信息相对独立,更新不会出现异常。

4）删除异常分析：分解后 3 个关系相对独立，对草地鼠害信息可以单独进行删除记录的操作，而不会影响草地信息关系和鼠害信息关系的记录。

上述分析可以看出，通过关系的分解，有效地避免了数据冗余和操作异常。那为什么未分解的关系会存在这些问题呢？比较来看，在原来的草地鼠害信息关系中，草地类名称和草地面积是相关的，害鼠名称和防治方法是相关的，但草地类名称、草地面积和防治方法是不相关的，同样，草地面积和害鼠名称也是不相关的。将互相不相关的一些属性组合在一起就会导致上述问题。因此，一个关系数据库中的每个关系的属性间要满足一定的内在联系，而这种联系又可按照对关系的不同要求分为不同的等级，这就是关系的规范化（Normalization）。

（三）关系模型的范式

介绍范式以前，需要先说明几个概念。

1. 函数依赖

函数依赖（Functional Dependency）是指在关系中的一个或一组属性值可以决定其他属性的值。这种关系类似于函数的因变量和自变量的关系，如有一个函数 $y=f(x)$，当 x 给定时，根据函数表达式，y 的数值也就是确定的。同理，如果属性集合 X 中的每一个属性值的集合能够唯一地决定属性集合 Y 中每一个属性值的集合，则称属性集合 Y 函数依赖于属性集合 X，或者 X 函数决定 Y，记为 $X\rightarrow Y$。例如，每个人的身份证号可以决定这个人的姓名，也就是说姓名函数依赖于身份证号或者说身份证号函数决定姓名，即身份证号→姓名。简单地说就是身份证号可以唯一确定姓名，当身份证号一定时，相应的姓名也是一定的。但反过来，由于存在同名的情况，如果有两个叫王五的人，他们每人都有一个身份证号，那么“王五”这个名字就有可能对应 2 个或多个身份证号。所以姓名并不能唯一确定身份证号，身份证号函数并不依赖于姓名。由此可见，这种情况下对于身份证号→姓名而言，它是一种一对多的关系，即一个姓名可以对应多个身份证号。如果我们规定，不允许同名的情况发生，那么当知道姓名时，身份证号就被唯一确定了，即姓名→身份证号。也就是说在不允许同名的情况下，身份证号→姓名和姓名→身份证号同时成立。不难发现，这时身份证号和姓名是一对一的关系。

再举一个例子，如对于这样一个关系：学生成绩（学号，姓名，课程编号，课程名称，课程成绩，任课老师姓名）。如果规定每个学生的学号唯一，每个课程编号只对应一门课程，那么就存在这些函数依赖关系：学号→姓名，课程编号→课程名称，（学号，课程编号）→课程成绩，课程编号→任课老师姓名。

2. 完全函数依赖

如果 $X\rightarrow Y$ 成立，对于 X 的任意一个真子集 x 而言，Y 都不函数依赖于 x，即 $x\rightarrow Y$ 不成立，那么称 Y 完全函数依赖于 X。例如，在上述学生成绩这一关系中，学号和课程编号一定的话，课程成绩也是一定的，即存在（学号，课程编号）→课程成绩。但是，学号并不能确定课程成绩，因为学号只能确定学生，但每个学生可能会上几门课，而每

门课都有成绩。同样，课程编号也不能唯一确定课程成绩，因为会有多个学生上同一门课，而每个学生都有这一门课程的成绩。由此可见，对于集合（学号，课程编号）而言，它决定课程成绩，但它的每一个真子集都不能函数决定课程成绩，因此课程成绩完全函数依赖于（学号，课程编号）集合。

3. 部分函数依赖

如果 $X \rightarrow Y$ 成立，X 存在一个真子集 x，使得 $x \rightarrow Y$ 成立，这时称 Y 部分函数依赖于 X。

例如，考察学生成绩关系可以发现存在（学号，姓名，课程编号）→课程成绩，也就是说（学号，姓名，课程编号）可以函数确定课程成绩，但（学号，姓名，课程编号）的真子集（学号，课程编号）也可以函数确定课程成绩，因此，对于（学号，姓名，课程编号）集合而言，课程成绩部分依赖于它。

4. 传递函数依赖

当同时满足 $X \rightarrow Y$，$Y \rightarrow Z$，且 Y 函数不依赖于 X 时，称 Z 对 X 传递函数依赖。

例如，有这样一个关系：班级信息（学号，姓名，班级，班主任）。对于一个学生而言，他必然属于某一个班级，因此有学号→班级，每个班级都有一个班主任，因此有班级→班主任，但每个班级会有很多学生，因此班级不能函数确定学生，同理，一个班主任有可能带几个班级，所以班主任也不能函数确定班级。因此，学号和班主任满足传递函数依赖关系，即班主任传递函数依赖于学号。

前面已谈及候选键、主键，现从函数依赖的角度做进一步阐述。

对于关系 R（A_1，A_2，…，A_n），X 是{A_1，A_2，…，A_n}的一个子集，如果 $X \rightarrow A_1A_2\cdots A_n$，且不存在任何 X 的真子集 x 使得 $x \rightarrow A_1A_2\cdots A_n$ 成立，则 X 是 R 的候选键。$X \rightarrow A_1A_2\cdots A_n$ 表示 X 能够唯一决定关系 R 中的每个属性，即唯一决定一个元组。如果 R 中有多个候选键，则选定其中一个作为主键。包含在任何候选键中的属性称为主属性（Prime Attribute），否则称为非主属性（Nonprime Attribute）。

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（Normal Form，NF）。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 6 种范式：第一范式（1NF）、第二范式（2NF）、第三范式（3NF）、Boyce-Codd 范式（BCNF）、第四范式（4NF）和第五范式（5NF）。一般而言，前 3 个范式已能满足大多草业信息数据库建设的需要。

（1）第一范式

对任何关系数据库而言，第一范式是最基本的要求，不满足第一范式的数据库就不能称为关系数据库。

第一范式要求数据库表的每一列都是不可分割的基本数据项，同一列中不能包含多种含义。即实体的每一个属性都不能有多个值或者不能有重复的属性。关系中每一行代表一个实体。简单来说，就是在数据表中不能有包含多个含义的列，不能有重复的属性。例如，对于一个草地类型数据库，不能将草地类、亚类、组、型放在同一个字段（属性）

描述，更不能在一个表中出现两个列都存储草地类名称的属性。第一范式是关系数据库的最基本要求。

（2）第二范式

第二范式是基于第一范式提出的，即要满足第二范式，首先需要满足第一范式。

第二范式要求数据库表中记录的每一个实体（每个记录）必须可以被唯一区分，即数据库表中不能有重复的记录。每条记录都表示不同的实体。一般，为了使数据表满足第二范式，往往在表上加一列作为标识属性。同时，标识属性往往被作为表的主键（关键字、主码）。

第二范式要求实体的属性完全函数依赖于主键。数据库表中不存在非主属性字段对任一候选关键字段的部分函数依赖，即所有非关键字段都完全依赖于任意一组候选关键字。简而言之，在第一范式基础上消除非主属性对主属性的部分依赖后就满足了第二范式。为便于理解，举例如下。

例如，有这样一个关系：草地分布（县编码，县名称，草地类名称，草地类特征）用来记录某个省各个县的草地类型及类的特征。在这个关系中，我们可以看出，由于某一个县可能会拥有多种草地类型，因此（县编码，草地类名称）是主键。考察依赖关系可以看出，存在（县编码，草地类名称）→草地类特征和草地类名称→草地类特征，因此存在非主属性草地类特征对主键的部分函数依赖。所以，该关系不满足第二范式。该关系中会出现数据冗余、更新异常和删除异常，如某个草地类在 10 个县都有分布，那么草地类特征属性将会被重复 10 次；如需要修改某一类草地特征的描述信息时，需要同时对包含该草地类的所有记录都进行修改；如果某一类草地只出现在某一个县，当删除该县的记录时，这类草地的相关信息（草地类名称、草地类特征）也被删除。

如果将这个关系进行分解，如分解成 R_1（县编码，县名称，草地类名称）和 R_2（草地类名称，草地类特征），这样就不会存在（县编码，草地类编码）→草地类特征的函数依赖关系，这两个关系都满足第二范式。

再比如，有一个学生成绩表的关系：R（学号，课程编号，成绩，任课老师姓名，任课老师职务）。分析可以看出，由于每个学生的每门课程都会有成绩、任课老师等信息，因此在该关系中（学号，课程编号）是主键，并存在（学号，课程编号）→（任课老师姓名，任课老师职务）和课程编号→（任课老师姓名，任课老师职务）等函数依赖关系，因此（学号，课程编号）→（任课老师姓名，任课老师职务）属于部分函数依赖。如果一个班有 50 人，都选择了某一门课程，那么需要 50 条记录（元组）记录他们的成绩，这门课程的任课老师姓名或任课老师职务将被重复 50 次。如果将这个关系分解成如下两个关系，就不会存在这一问题。

R_1（学号，课程编号，成绩）

R_2（课程编号，任课老师姓名，任课老师职务）

由此可见，第二范式就是要尽可能地减少数据表中的数据冗余。

（3）第三范式

第三范式要求一个数据表中任意两个非主关键字段（非主属性）的数据之间不存在函数依赖关系，即满足第二范式的基础上，消除非主属性对主属性之间的传递依赖关系。

简单地说，第三范式要求不要在一个数据表中存储可基于已有属性（字段）通过简单计算得到的数据。

例如，对于这样一个关系：R（学号，姓名，年龄，班级，班主任），因为一个学生必然对应一个班级，所以在关系 R 中存在学号→（姓名，年龄，班级，班主任），学号是它的主键。但是，每个班级的班主任是一定的，即存在学号→班级→班主任的传递函数依赖关系，所以关系 R 不满足第三范式。同样，不满足第三范式的数据表也存在数据冗余、更新异常和删除异常。要想让该关系满足第三范式，那么需要进行如下分解：

R_1（学号，姓名，年龄，班级）

R_2（班级，班主任）

上述分解可以看出，第三范式要求一张表只表达一个关系或一个实体。

通过上述介绍也可以看出，数据规范化的目的是合理设计数据库表的结构，消除存储异常，使数据冗余最小，便于更新、删除、添加等操作。实现数据规范化的原则可以理解成尽可能地建“小表”，尽量避免建“大表”，即尽量将一个数据表分解成几个表存储，确保每个表只表达一项或一方面的内容。

然而，同时应该看到，数据表的规范化程度越高，数据冗余越小，操作错误的可能性越小，但随着规范程度的提高，在数据库查询检索时需要做的关联操作就越多，数据库操作过程中需要访问和操作的数据表及其联系就越多。例如，R（学号，姓名，年龄，班级，班主任）关系虽然不满足第三范式，但如果要查询某个学生的班主任姓名，只要访问这一个表就可以。如果将其分解成 R_1（学号，姓名，年龄，班级）和 R_2（班级，班主任）两个关系，那么对同样的需求，需要先将 R_1 和 R_2 通过班级属性连接起来再查询，这就会占用更多的系统资源。因此，在数据库设计的过程中应该根据具体的需求和实际情况，选择一个合理的规范标准，而不是一味地追求较高的规范程度，而导致实际应用困难。

三、结构化查询语言

（一）SQL 概述

结构化查询语言（Structured Query Language）一般简称为 SQL。SQL 是关系型数据库管理系统的标准语言。SQL 语句可以用来执行各种操作，如数据库中数据的更新、查询、删除等。目前绝大多数流行的数据库管理系统，如 Oracle、Microsoft SQL Server、MySQL 等都采用了 SQL 标准。

结构化查询语言产生于 20 世纪 70 年代，最初被称为 SQUARE，是 IBM 公司 San Jose 实验室为其研制的关系型数据库系统 System R 配置的查询语言。由于 SQL 结构化查询功能强大，而且简单易学，得到数据库产业的广泛接受，之后被不断地修改和完善，最终成为关系型数据库的标准语言。

结构化查询语言包括以下几个方面的内容。

1）数据定义语言（Data Definition Language，DDL）：用于定义数据库对象，包括定义表、视图、索引等。

2）数据库操作语言（Data Manipulation Language，DML）：用于对数据库中的数据进行查询及插入、删除、修改等操作。

3）嵌入式和动态 SQL：嵌入式 SQL 使程序员可以在高级程序设计语言中直接加入 SQL 语句，动态 SQL 使程序员可以在程序运行过程中构建查询。

4）事务管理：用来定义事务的开始、提交、回滚等。

5）安全性管理：用来对用户进行权限管理。

6）触发器和完整性约束：用来保障数据库中数据的正确性。

7）客户服务器执行和远程数据库存取：控制一个用户的应用程序连接到一个关系型数据库服务器，或者通过网络来访问数据库中的数据。

（二）常见数据类型

不同的关系型数据库管理系统对数据类型的定义不同，但大体上都包括如下几种数据类型：数值型、字符型、二进制类型和其他类型。下面以 Microsoft SQL Server 数据库软件为例，简要介绍不同数据类型的特点。

1. 整数

在 Microsoft SQL Server 中，整数类型包括 tinyint、smallint、int、bigint 等类型（表 5-3）。至于为何需要设置这么多的整数数据类型，这主要是为了便于用户根据具体需要，选取合理的数据类型，并不是说只要是整数就用 int 类型，要同时兼顾实用性和数据存储的合理性。如果确定某一属性的值域不会超过 255，如月份或年龄，那么就没必要选择 int 或 smallint 数据类型，tinyint 就足够了。如果有一个用户数据表，在设计的时候将年龄字段设置成了 int 型，那么意味着每个元组（用户）将多占用 3 个字节的存储空间，如果有 1 万个用户，将多花费 3 万字节空间。如果有 10 万个用户呢？

表 5-3　Microsoft SQL Server 中的整数类型

数据类型	数值范围	存储空间
bigint	-2^{63}～（$2^{63}-1$）	8 字节
int	-2^{31}～（$2^{31}-1$）	4 字节
smallint	-2^{15}～（$2^{15}-1$）	2 字节
tinyint	0~255	1 字节

2. 小数

在数据库中，小数的存储可大体分成两类，一类是精确存储的数据类型（有时也称带固定精度和小数位数的数字类型或定点数据），一类是近似存储的数据类型（也称浮点数据）。例如，在 Microsoft SQL Server 中精确存储的数据类型有 decimal 和 numeric 两种，而近似存储的数据类型有 float 和 real 两种。定义精确存储的小数时需要设置精度和小数位数。例如，decimal（18，4），其中 18 代表精度，表示最多可以存储 18 位的十进制数字，包括小数点左边和右边的位数。一般该精度是 1 至 38 之间的值。4 表示小数位数，即小数点右边可以存储的十进制数字的最大位数，小数位数必须小于精度。

numeric 在功能上和 decimal 类似。例如，需要存储 22.47，可以将其存储在 decimal（4，2）或 numeric（4，2）中，如果将类型设置成 decimal（4，1），则会存储为 22.5。在近似存储的数据类型中，float 型可以指定存储数值尾数的位数（即以科学计数法表示的小数的位数），一般介于 1～53。real 类型相当于 float（24）。因此，在数据库建设过程中应根据存储数据的特征和精度要求选择不同的数据类型。

3. 字符串

字符串数据类型可用于存储字符或字符串，根据字符的编码方式可分为 Unicode 和非 Unicode 两类。Microsoft SQL Server 中提供了 3 种存储非 Unicode 字符串的数据类型：char、varchar 和 text。char 类型是固定长度的字符串类型，定义时需设置字符串的长度，如 char（10）表示存储 10 个字节长度的字符串，char 类型最多允许存储 8000 个字节的字符串。相对于 char 而言，varchar 是可变长度的字符串类型，可根据存储的字符串的大小自动调整存储空间。当需要存储的字符串有可能超过 8000 个字符时，可使用 varchar（max），它最大可存储 2^{31}–1 个非 Unicode 字符。text 可存储长度可变的非 Unicode 数据，最大长度为 2^{31}–1 个字符。同 varchar（max）类似。

对于 Unicode 编码的字符串，Microsoft SQL Server 中提供了 3 种数据类型，nchar、nvarchar 和 ntext。同 char、varchar 和 text 之间的差异类似，只是由于 Unicode 编码每个字符占 2 个字节，因此较 char、varchar 和 text 相比，可存储的字符串长度少一半。例如，nchar 类型存储固定长度的 Unicode 字符串，最多允许存储 4000 个字节的 Unicode 字符串；nvarchar 可存储 1～4000 个 Unicode 字符串，而 nvarchar（max），它最大可存储 $2^{30}-1$ 个 Unicode 字符；text 可存储长度可变的 Unicode 数据，最大长度为 $2^{30}-1$ 个字符。

由上述介绍可以看出，在选择字符串数据类型时，首先需要考量需要存储的字符串是基于哪种编码的，然后再考虑字符串的最大可能长度，最后确定哪种类型最适合应用需求。例如，希望以字符串类型存储身份证号码，那么 char（18）是最好的选择。如果要存储姓名，应考虑到虽然一般姓名多为 2 个汉字到 4 个汉字之间，但我国少数民族的名字一般会较长，因此在类型选择时不能简单地选择 nchar（4），而应该考虑应用范围，确定用户的名字都能准确保存，所以这时就可以考虑选择 nvarchar。

4. 二进制数据类型

二进制数据类型用于存储二进制文件，如图片等。Microsoft SQL Server 提供了 3 种二进制数据类型：image、binary 和 varbinary。其中，image 可存储长度在 0～（$2^{31}-1$）个字节的二进制数据；binary 可存储长度固定不变的二进制数据，允许设置的长度范围为 1～8000 个字节；varbinary 为存储长度可变的二进制数据，允许设置的长度范围为 1～8000 个字节，varbinary（max）最多可存储 $2^{31}-1$ 个字节的数据。

5. 时间和日期

由于时间和日期相对特殊，而且使用频繁，因此，绝大多数数据库管理系统都设置了针对时间和日期的专用数据类型。Microsoft SQL Server 提供了 6 种存储时间和日期的

数据类型。

datetime：占 8 字节，该数据类型将日期和时间作为一个单列值存储。可存储从 1753 年 1 月 1 日到 9999 年 12 月 31 日之间的日期和时间，时间精度为 3.33ms。

samlldatetime：占 4 字节，可存储从 1900 年 1 月 1 日至 2079 年 6 月 6 日之间的日期和时间，时间只精确到分钟。

date：占 3 字节，该数据类型只存储日期，而不存储时间。支持的日期范围从 1 年 1 月 1 日至 9999 年 12 月 31 日。

time：占用 3～5 个字节，专用于存储时间的数据类型，不包含日期。用于存储 1 天 24 小时，支持 100 纳秒的精度。

datetime2：占用 6～8 字节，可存储 1 年 1 月 1 日至 9999 年 1 月 1 日。时间部分的精度可以由存储精度来决定。可设置 0～7 的精度，表示秒的小数位数。如 datatime2（0）精确到秒，最多支持 7 位小数表示的秒。

datetimeoffset：数据存储范围同 datetime2 一样，带有时区分量。

6. 其他数据类型

除了上述介绍的常用数据类型之外，不同的数据库管理系统还提供了差异很大的其他一些特殊用途的数据类型，如 money 类型等。

（三）常用 SQL 语句示例

通过 SQL 语句，可以实现数据表的创建、查询、更新、删除等常规操作，以下就常用的 SQL 语句为例，简要介绍一下 SQL 语句的用法（黄河和王贤志，2011）。

1. 数据表的创建

数据表的创建是数据库构建的第一步，在创建数据表时，需要定义数据表的名称、列（属性）的数据类型、约束等。创建数据表可使用 SQL 的 CREATE TABLE 语句。CREATE TABLE 语句的基本语法格式如下

```
CREATE TABLE table_name
(
column_name data_type [column_constraint]
[, … n]
)
```

其中，table_name 用于指定数据表的名称，column_name 是数据表中列的名称，data_type 是该列的数据类型，以上都是必填项。[column_constraint]用于规定该列的约束条件。[, …n]指 column_name data_type [column_constraint]部分可以重复 n 次，即一个表中可以包含 1～n 个列。

例 1：创建一张草地类型数据表。

```
CREATE TABLE grasslandtype
(
id INT PRIMARY KEY,
```

```
class NVARCHAR（20） NOT NULL，
subclass NVARCHAR（50） NOT NULL，
type NVARCHAR（50）
）
```

通过上述语句可以创建一个名为 grasslandtype 的数据表，它包含 id、class、subclass、type 等字段。其中，id 为主键（Primary Key），class 和 subclass 字段不能为空值（Not Null）。

创建表需要注意的是：每个表必须至少包含一个列，每个列必须有一个名称，并必须规定数据类型。现在，数据表的创建及修改一般都集成在数据库管理系统中，用户可以通过数据库管理系统的界面，交互式地创建和修改数据表。

2. 数据表的删除

在 SQL 中，可以使用 DROP TABLE 语句来实现数据表的删除操作，DROP TABLE 的一般格式如下

```
DROP TABLE table_name
```

其中，table_name 为需要删除的表名称。

例 2：删除表 grasslandtype。

```
DROP TABLE grasslandtype
```

3. 数据表的查询

数据查询是数据库的核心操作，在 SQL 中使用 SELECT 语句实现数据表的查询。SELECT 语句的核心结构如下

```
SELECT select_list
FROM table_list
WHERE search_conditions
GROUP BY group_by_list
HAVING search_conditions
ORDER BY order_list [ASC|DESC]
```

其中，select_list 是需要查询的对象，table_list 是需要查询的表，search_conditions 是查询的条件，group_by_list 是分组或聚类表达式，order_list 是排序列，可以规定排序规则是升序（ASC）还是降序（DESC）。在 SELECT 语句中，WHERE、GROUP BY、HAVING、ORDER BY 都是可选项，即可以省略。

为了更好地理解 SELECT 语句的用法，现假设有如下两个示例数据（表 5-4、表 5-5），通过查询示例介绍 SELECT 语句的功能和用法。

表 5-4 草地类信息

草地类代码	草地类名称	草地类面积/hm^2
1	温性草甸草原类	14 519 331
2	温性草原类	27 477 870
3	温性荒漠草原类	18 921 607
4	高寒草甸草原类	6 865 734

续表

草地类代码	草地类名称	草地类面积/hm^2
5	高寒草原类	4 162 317
6	高寒荒漠草原类	9 566 006
7	温性草原化荒漠类	10 673 418
8	温性荒漠类	45 060 811
9	高寒荒漠类	7 527 763
10	暖性草丛类	6 657 148
11	暖性灌草丛类	11 790 493
12	热性草丛类	14 237 195
13	热性灌草丛类	17 376 693
14	干热稀树灌草丛类	863 144
15	低地草甸类	25 219 621
16	山地草甸类	16 718 926
17	高寒草甸类	63 720 549
18	沼泽类	2 873 812

表 5-5　草地亚类信息

草地亚类代码	草地类代码	草地亚类名称	草地亚类面积/hm^2
1	1	平原丘陵草甸草原亚类	6 085 082
2	1	山地草甸草原亚类	8 239 902
3	1	沙地草甸草原亚类	194 347
4	2	平原丘陵草原亚类	22 321 634
5	2	山地草原亚类	10 678 429
6	2	沙地草原亚类	8 096 508
7	7	平原丘陵荒漠草原亚类	966 068
8	7	山地荒漠草原亚类	8 054 712
9	7	沙地荒漠草原亚类	2 206 207
10	8	土砾质荒漠亚类	37 005 456
11	8	沙质荒漠亚类	6 561 697
12	8	盐土质荒漠亚类	1 493 658
13	15	低湿地草甸亚类	6 520 454
14	15	盐化低地草甸亚类	12 266 339
15	15	滩涂盐化低地草甸亚类	377 287
16	15	沼泽化低地草甸亚类	6 055 541
17	16	中、低山山地草甸亚类草地	7 208 367
18	16	亚高山山地草甸亚类	9 510 559
19	17	典型高寒草甸亚类	54 408 729
20	17	盐化高寒草甸亚类	800 757
21	17	沼泽化高寒草甸亚类	8 511 063

例 3：获取表 5-4 草地类信息中的所有记录。

SELECT *

FROM 草地类信息

当不填写任何条件时，即查询全部记录。*为通配符，表示草地类信息表中的所有字段，当使用*作为查询对象时将返回草地类信息的全部记录。

例 4：获取草地类信息表中类编码等于 1 的记录。

SELECT *

FROM 草地类信息

WHERE 草地类代码 = 1

例 5：获取草地类信息中草地类面积大于 100 000hm^2 的类名称。

SELECT 草地类名称

FROM 草地类信息

WHERE 草地类面积 > 100 000

例 6：获取草地类信息中草地类面积介于 100 000～200 000hm^2 的类名称。

SELECT 草地类名称

FROM 草地类信息

WHERE 草地类面积 BETWEEN 100 000 AND 200 000

当需要按某一条件查询时，可使用 WHERE 后跟条件表达式的方式，表达式中可使用等于（=）、大于（>）、小于（<）、不等于（<>）、大于等于（>=）、小于等于（<=）等判断符号及且（AND）、或（OR）、非（NOT）等逻辑判断符。BETWEEN…AND…表示介于某一范围之内。如果要查询不在某一范围之内，只需要在前面添加 NOT 即可。

例 7：获取草地类信息中草地类面积不介于 100 000～200 000hm^2 的类名称。

SELECT 草地类名称

FROM 草地类信息

WHERE 草地类面积 NOT BETWEEN 100 000 AND 200 000

例 8：查询草地类名称中带有“草甸”字样的草地类名称。

SELECT 草地类名称

FROM 草地类信息

WHERE 草地类名称 LIKE ‘%草甸%’

LIKE 表示模糊查询，其中%表示任意字符，‘%草甸%’表示字符串任意位置包含“草甸”字样的记录，如果写成‘草甸%’，则表示以“草甸”开头的字符串；如果写成‘%草甸’，则表示以“草甸”结尾的字符串。另外需要注意的是，在 SQL 语句中字符串类型（包括 Unicode 编码和非 Unicode 编码）都以单引号引起来。

例 9：查询草地类名称为空值的记录。

SELECT *

FROM 草地类信息

WHERE 草地类名称 IS NULL

NULL 表示空，即属性没有赋值

例 10：查询草地类名称为“高寒草甸类”和“温性草原类”的记录。

```
SELECT *
FROM 草地类信息
WHERE 草地类名称 IN （‘高寒草甸类’，‘温性草原类’）
```

IN 表示包含，当然也可以写成 WHERE（（草地类名称 =‘高寒草甸类’） OR （草地类名称 =‘温性草原类’）），它们是等效的。

例 11：在草地类信息表中查询草地类名称，并将查询结果按草地类面积从大到小排列，显示面积最大的 5 个草地类。

```
SELECT TOP 5 草地类名称
FROM 草地类信息
ORDER BY 草地类面积 DESC
```

TOP 用来限制输出结果的记录数，TOP 5 表示输出查询结果的前 5 项。如果想查询输出结果的前 10 条记录，可以用 TOP 10；如果想输出查询结果的前百分之十，可以用 TOP 10 PERCENT。类似的限制输出语句还有 DISTINCT，其作用是将查询结果中重复出现的记录排除在外，可以理解为去除重复记录。

例 12：在草地亚类信息表中查询每种草地类包含的草地亚类的个数。

```
SELECT 草地类代码，count（*）as 草地亚类个数
FROM 草地亚类信息
GROUP BY 草地类代码
```

使用 GROUP BY 可将查询结果按照某一列或几列的值进行聚类（分组），相同的值将被分为同一组，聚合函数将对每一组分别进行汇总和统计。如在本例中，首先得到草地亚类信息表的全部记录（没有 WHERE 条件约束），然后按照草地类代码字段分组，草地类代码字段中有 1、2、7、8、15、16、17 这 7 组值，因此共分成 7 组，最后使用聚合函数 COUNT（）统计每一组中记录的个数，即各个草地类中草地亚类的个数。as 语句实现对列的重命名，即将 COUNT（*）的结果列命名成草地亚类个数。查询结果如表 5-6 所示。

表 5-6　聚类查询结果

草地类代码	草地亚类个数
1	3
2	3
7	3
8	3
15	4
16	2
17	3

常见的聚合函数有计数函数 COUNT（），求和函数 SUM（），均值函数 AVG（），最大值函数 MAX（），最小值函数 MIN（）。除了 COUNT（）以外，其他函数均忽略空值，即空值不参与运算。

SQL 还支持对 GROUP BY 语句分组后的结果进行条件筛选。

例13：在草地亚类信息表中查询每种草地类的面积，并筛选出草地类面积大于100 000hm^{2}的记录。

```
SELECT 草地类代码，SUM（草地亚类面积）as 草地类面积
FROM 草地亚类信息
GROUP BY 草地类代码
HAVING SUM（草地亚类面积）> 100 000
```

通过 HAVING 语句可以对分类结果进行筛选，HAVING 子句指定查询结果中返回的组应满足的条件。在本例中，使用 HAVING 筛选草地类面积大于 100 000hm^2 的组。

当然，上例也可以写成如下的查询形式

```
SELECT *
FROM
(
SELECT 草地类代码，SUM（草地亚类面积） as 草地类面积
FROM 草地亚类信息
GROUP BY 草地类代码
） as T
WHERE 草地类面积 > 100 000
```

也就是说先按草地类代码分组，查询到每个类的草地类面积，然后以查询结果为被查询表（T），使用 SELECT 语句再次查询草地类面积大于 100 000hm^2 的记录。由此可见，SELECT 语句可以进行嵌套。

前面举了单表查询的几个例子，如果希望同时查询多张表应该如何实现呢？比如希望通过草地类编码将草地类名称关联到草地亚类信息表中。

例 14：通过 SQL 语句将草地类信息表中的草地类名称关联到草地亚类信息表中。

```
SELECT 草地亚类信息.*，草地类信息.草地类名称
FROM 草地亚类信息，草地类信息
WHERE 草地亚类信息.草地类代码 = 草地类信息.草地类代码
```

在本例中，我们希望将草地类信息和草地亚类信息表关联，并返回草地亚类信息表的所有字段和草地类信息表中草地类名称字段，因此在查询对象列表中使用“表名称.字段”形式的表示方法，如草地亚类信息.*表示草地亚类信息表的所有字段，草地类信息.草地类名称表示草地类信息表中草地类名称字段。这样写的好处是查询对象指示清晰，查询的是哪个表的哪个字段是明确的。在 FROM 后面列出了草地亚类信息和草地类信息两个表，并以逗号隔开，表示同时对这两个表进行查询。WHERE 语句规定了两个表进行关联的条件，即将两个表中草地类代码相等的字段相关联。查询结果可参见表5-7，从中可以看出，通过 SELECT 语句的多表关联，得到了一张新的表格，该表格包含了原来的两个数据表的信息，表达出了原来两个表格中所蕴含的信息，这也是数据库查询的核心思想。

表 5-7　多表关联查询结果

草地亚类代码	草地类代码	草地亚类名称	草地亚类面积/hm^2	草地类名称
11	1	平原丘陵草甸草原亚类	6 085 082	温性草甸草原类
12	1	山地草甸草原亚类	8 239 902	温性草甸草原类
13	1	沙地草甸草原亚类	194 347	温性草甸草原类
21	2	平原丘陵草原亚类	22 321 634	温性草原类
22	2	山地草原亚类	10 678 429	温性草原类
23	2	沙地草原亚类	8 096 508	温性草原类
31	7	平原丘陵荒漠草原亚类	966 068	温性草原化荒漠类
32	7	山地荒漠草原亚类	8 054 712	温性草原化荒漠类
33	7	沙地荒漠草原亚类	2 206 207	温性草原化荒漠类
81	8	土砾质荒漠亚类	37 005 456	温性荒漠类
82	8	沙质荒漠亚类	6 561 697	温性荒漠类
83	8	盐土质荒漠亚类	1 493 658	温性荒漠类
151	15	低湿地草甸亚类	6 520 454	低地草甸类
152	15	盐化低地草甸亚类	12 266 339	低地草甸类
153	15	滩涂盐化低地草甸亚类	377 287	低地草甸类
154	15	沼泽化低地草甸亚类	6 055 541	低地草甸类
161	16	中、低山山地草甸亚类草地	7 208 367	山地草甸类
162	16	亚高山山地草甸亚类	9 510 559	山地草甸类
171	17	典型高寒草甸亚类	54 408 729	高寒草甸类
172	17	盐化高寒草甸亚类	800 757	高寒草甸类
173	17	沼泽化高寒草甸亚类	8 511 063	高寒草甸类

例 14 的关联查询语句也可以写成如下形式

SELECT 草地亚类信息.*，草地类信息.草地类名称

FROM 草地亚类信息 INNER JOIN 草地类信息 ON 草地亚类信息.草地类代码 = 草地类信息.草地类代码

这两种写法的效果是等价的。INNER JOIN 表示“内连接”，ON 后面的表达式指示连接的条件。所谓内连接即指查询两个表中都满足条件的记录，而两个表中不满足连接条件的记录将不包含在查询结果中。

同内连接相对应的是外连接，外连接常用于查询相连接的两个表中，至少需要获取一个表的全部记录。外连接又可分为左外连接、右外连接和全外连接 3 种。外连接的查询结果中不仅包含满足连接条件的记录，而且还包含相应表格中不满足连接条件的记录。即左外连接返回所有满足连接条件的记录和 JOIN 关键字左边的表格中所有不满足连接条件的记录；右外连接返回所有满足连接条件的记录和 JOIN 关键字右边的表格中所有不满足连接条件的记录；全外连接返回连接表中所有的记录。

例 15：依据草地亚类信息表中草地类代码字段查找草地类信息表中草地类代码字段对应的草地类名称。

SELECT 草地亚类信息.*，草地类信息.草地类名称

FROM 草地亚类信息 LEFT OUTER JOIN 草地类信息

ON 草地亚类信息.草地类代码 = 草地类信息.草地类代码

左外连接采用 LEFT OUTER JOIN 关键字，位于关键字左边的表格“草地亚类信息”的全部记录将被返回到查询结果中，如果在关键字右边的表格“草地类信息”中没有查询到相对应的记录，那么草地类名称字段将返回空值（NULL）。当然，在本例中由于草地亚类信息表中所有的草地类代码值在草地类信息表中都有对应的记录，因此左外连接的查询结果同内连接的查询结果是一致的。

例 16：依据草地亚类信息表中草地类代码字段查找草地类信息表中草地类代码字段对应的草地类名称，并确保草地类名称全部出现。

SELECT 草地亚类信息.*，草地类信息.草地类名称

FROM 草地亚类信息 RIGHT OUTER JOIN 草地类信息

ON 草地亚类信息.草地类代码 = 草地类信息.草地类代码

右外连接采用 RIGHT OUTER JOIN 关键字，位于关键字右边的表格“草地类信息”的全部记录将被返回到查询结果中，如果在关键字左边的表格“草地亚类信息”中没有相对应的记录，那么相应的字段将返回空值（NULL）。右外连接的查询结果如表 5-8 所示，可以看出在草地亚类信息表中没有对应记录的草地类也出现在查询结果中，相应记录的草地亚类代码、草地类代码、草地亚类名称和草地亚类面积字段都为空值。

表 5-8 右外连接查询结果

草地亚类代码	草地类代码	草地亚类名称	草地亚类面积/hm^2	草地类名称
11	1	平原丘陵草甸草原亚类	6 085 082	温性草甸草原类
12	1	山地草甸草原亚类	8 239 902	温性草甸草原类
13	1	沙地草甸草原亚类	194 347	温性草甸草原类
21	2	平原丘陵草原亚类	22 321 634	温性草原类
22	2	山地草原亚类	10 678 429	温性草原类
23	2	沙地草原亚类	8 096 508	温性草原类
NULL	NULL	NULL	NULL	温性荒漠草原类
NULL	NULL	NULL	NULL	高寒草甸草原类
NULL	NULL	NULL	NULL	高寒草原类
NULL	NULL	NULL	NULL	高寒荒漠草原类
31	7	平原丘陵荒漠草原亚类	966 068	温性草原化荒漠类
32	7	山地荒漠草原亚类	8 054 712	温性草原化荒漠类
33	7	沙地荒漠草原亚类	2 206 207	温性草原化荒漠类
81	8	土砾质荒漠亚类	37 005 456	温性荒漠类
82	8	沙质荒漠亚类	6 561 697	温性荒漠类
83	8	盐土质荒漠亚类	1 493 658	温性荒漠类
NULL	NULL	NULL	NULL	高寒荒漠类
NULL	NULL	NULL	NULL	暖性草丛类
NULL	NULL	NULL	NULL	暖性灌草丛类
NULL	NULL	NULL	NULL	热性草丛类

续表

草地亚类代码	草地类代码	草地亚类名称	草地亚类面积/hm^2	草地类名称
NULL	NULL	NULL	NULL	热性灌草丛类
NULL	NULL	NULL	NULL	干热稀树灌草丛类
151	15	低湿地草甸亚类	6 520 454	低地草甸类
152	15	盐化低地草甸亚类	12 266 339	低地草甸类
153	15	滩涂盐化低地草甸亚类	377 287	低地草甸类
154	15	沼泽化低地草甸亚类	6 055 541	低地草甸类
161	16	中、低山山地草甸亚类草地	7 208 367	山地草甸类
162	16	亚高山山地草甸亚类	9 510 559	山地草甸类
171	17	典型高寒草甸亚类	54 408 729	高寒草甸类
172	17	盐化高寒草甸亚类	800 757	高寒草甸类
173	17	沼泽化高寒草甸亚类	8 511 063	高寒草甸类
NULL	NULL	NULL	NULL	沼泽类

例 17：依据草地亚类信息表中草地类代码字段查找草地类信息表中草地类代码字段对应的草地类名称，并确保草地类名称和草地亚类信息全部出现。

```
SELECT 草地亚类信息.*，草地类信息.草地类名称
FROM 草地亚类信息 FULL OUTER JOIN 草地类信息
ON 草地亚类信息.草地类代码 = 草地类信息.草地类代码
```

全外连接采用 FULL OUTER JOIN 关键字，位于关键字两边的表格的全部记录都将被返回到查询结果中，如果在关键字左边或右边的表格中的记录不满足连接条件时，相应的字段将返回空值（NULL）。在本例中，由于草地亚类信息表中所有的草地类代码在草地类信息中都有对应的记录，因此全外连接的查询结果同右外连接是一致的。

查询多个表的数据也不一定要使用表的连接，也可以使用子查询实现。所谓子查询就是先进行一次数据查询，将查询结果提供给另一个查询使用。

例 18：在草地类信息表中查询草地类代码出现在草地亚类信息表中的记录。

```
SELECT *
FROM 草地类信息
WHERE 草地类信息.草地类代码 IN
(
SELECT DISTINCT 草地亚类信息.草地类代码
FROM 草地亚类信息
)
```

本例首先从草地亚类信息表中查询获得草地类代码集合，然后以此为 IN 的条件集合，在草地类信息表中查询草地类代码属于该集合的记录集。

4. 数据表的更新

数据表的更新包括插入数据、修改数据和删除数据等操作。

SQL 中使用 INSERT 语句实现数据表中数据的插入操作，其语法如下

INSERT [INTO] <表名> [<列名称列表>] VALUES（值列表）

使用 INSERT 语句需注意以下几点。

a. <列名称列表>中的列名称必须是表名指定的表中已有的字段，但可以以任意顺序出现。

b.（值列表）中的数值必须和<列名称列表>中的列名称一一对应，待插入的数据必须同它所在列的数据类型一致，可以是空值（NULL），各个值之间以逗号隔开。

c. 如果没有指定<列名称列表>，则（值列表）中所列待插入数据的顺序必须与表中定义列的顺序一致，且每一列都有值（可以是空值）；如果<列名称列表>中只指定了表中的部分字段，那么未指定的字段将插入默认值或空值。

例 19：在草地类信息表中插入一条新记录。

INSERT INTO 草地类信息（草地类代码，草地类名称，草地类面积）VALUES（18，沼泽类，2 873 812）

INSERT INTO 草地类信息 VALUES（18，沼泽类，2 873 812）

以上两种方式是等价的，都实现了数据的插入。

当需要修改数据库中已有数据时，可以使用 UPDATE 语句。UPDATE 语句的语法如下

UPDATE <表名> SET <列名=表达式>[，…，*n*][WHERE <更新条件>]

UPDATE 语句在使用需注意以下几点。

a. 使用 WHERE 子句来指定需要修改的记录，如果没有 WHERE 子句，则 UPDATE 语句会修改所有的记录。

b. 在一条 UPDATE 语句中不能对同一字段作多次修改。

c. 新修改的数据必须和原来的数据类型相一致，否则更新将失败。

例 20：将草地类信息表中沼泽类的面积从原来的 2 873 812hm^2 改为 2 873 885hm^2。

UPDATE 草地类信息 SET 草地类面积=2 873 885 WHERE 草地类名称 = '沼泽类'

当希望删除某些记录时，可以使用 DELETE 语句实现数据表中记录的删除操作。DELETE 语句的语法如下所示

DELETE [FROM] <表名> [WHERE <删除条件>]

DELETE 语句可以一次从表中删除一行、几行或全部记录。当使用 WHERE 指定条件时，DELETE 语句将删除满足指定条件的记录，如果没有指定条件，则删除表中全部记录。

例 21：删除草地类信息表中草地类代码等于 18 的记录。

DELETE FROM 草地类信息表 WHERE 草地类代码 = 18

第二节　草业基础数据库构建与应用

草业基础数据库构建需要依据数据库的基本理论和方法，结合草业科学的具体应用需求，设计并建立专业数据库。本节在介绍数据库设计的一般流程基础上，采用示例的

方式介绍草业基础数据库设计的一般过程，并简要介绍现有主要草业基础数据库及其未来发展趋势，以及大数据及其在草业科学研究与实践中的应用潜力。

一、草业基础数据库设计

（一）数据库设计概述

数据库设计的目的是建立一个性能良好的、能够满足不同用户需求的数据库系统。美国电气和电子工程师协会（Institute of Electrical and Electronics Engineers，IEEE）对数据库设计的定义是：开发一个满足用户需求的数据库的过程，包括概念数据库设计、逻辑数据库设计和物理数据库设计。由此可见，数据库设计一般包括以下 4 个阶段，即需求分析阶段、概念设计阶段、逻辑结构设计阶段和物理结构设计阶段（陈晓云和徐玉生，2009）。

1. 需求分析阶段

需求分析是数据库设计的第一步，也是最重要的一步。明确而具体的需求分析是数据库设计的前提，需求分析的准确度、精确度和充分程度决定了数据库设计的质量与进度。需求分析主要是信息收集、分析和整理的过程，它将用户的需求具体化、细节化。

2. 概念设计阶段

概念设计的目标是以需求分析为依据，构建一个反映整个系统需求的整体数据库概念模型。概念模型与实现细节（DBMS、编程语言、平台等）无关。概念模型的内容有实体、联系、属性与属性域、主键、约束规则等。具体来讲，就是确定整个系统中涉及哪些实体；每个实体之间存在哪些联系；这些实体都具有哪些属性；基于这些实体、属性和联系，每个实体的每个属性可能的取值范围是什么；应该设置哪些约束条件对某些实体或属性加以约束等。

3. 逻辑结构设计阶段

数据库的逻辑设计是将概念设计的结果转换成特定的数据库管理系统（DBMS）所支持的数据模型的过程。一般分为两个部分：一是将概念模型转换为基本的关系模式，如转换概念模型中的多对多关系，建立局部逻辑数据模型；二是结合具体的 DBMS 和范式理论，将关系规范化，如对实体进行分解，定义完整性、约束等。

4. 物理结构设计阶段

物理设计是将逻辑设计模型用特定的 DBMS 加以实现，包括基本关系的表达、文件的组织、索引的定义、视图的定义，以及完整性、安全性实现的措施等。具体来说，物理设计阶段就是在特定的 DBMS 中，依据逻辑设计结果，将数据库系统搭建起来的过程。

总体来看，需求分析是发现或提出问题的过程，概念设计是理解和分析问题的过程，逻辑设计是描述和定义问题的过程，而物理设计是解决问题的过程。