

情境 1

初识数据库和 SQL Server

项目 1 设计 EMIS 数据库

项目 2 安装和使用 SQL Server 2019

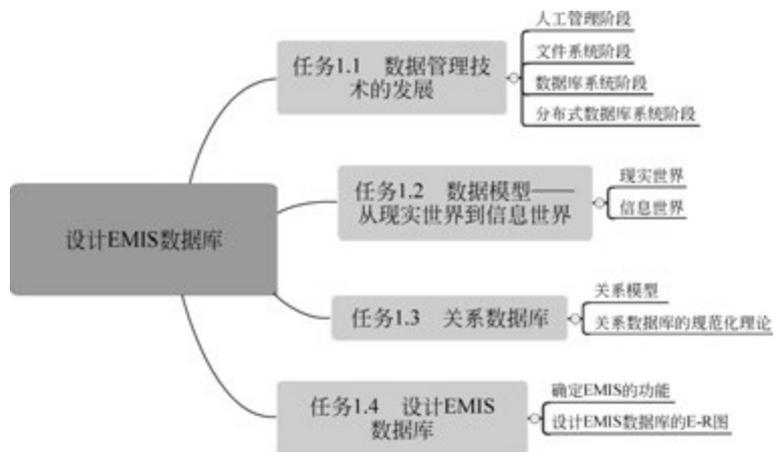
项目 1

设计EMIS数据库

项目背景

EMIS(教学管理信息系统)数据库的设计是实现 EMIS 的前提,通过它,可以实现信息的规范化处理、科学的统计和快速的查询。

内容导航



任务 1.1 数据管理技术的发展

相关知识

随着计算机技术的发展,数据管理技术经历了人工管理、文件系统、数据库系统和分布式数据库系统 4 个阶段。下面介绍这 4 个阶段的历史以及特点。

1. 人工管理阶段

在 20 世纪 50 年代中期以前,计算机最主要的用途是科学计算。当时的硬件外存只有纸带、卡片、磁带这些设备,没有像磁盘这样直接存取的存储设备;在软件方面,没有操作系统,没有专门管理数据的软件,对数据的管理也没有一定的格式。这些技术也决定了当时的数据管理只能靠人工来处理。

人工管理阶段的主要表现如下。

(1) 数据不保存。当时计算机主要用于科学计算,在计算机中装入程序和数据,计算完成后就退出,一般不需要将数据进行长期保存。

(2) 数据无管理。系统没有专用的软件对数据进行管理,每个应用程序都要包括数据的存储结构、存取方法和输入方法等内容。程序员编写程序时,还要负责安排数据的物理存储,程序员负担很重。

(3) 数据不共享。数据是面向程序的,一组数据只能对应一个程序。当多个应用程序涉及相同的数据时必须各自定义,无法相互使用和参照。

(4) 数据不具有独立性。数据的逻辑结构或者物理结构发生变化后,必须对使用它的应用程序进行相应的修改,数据完全依赖于应用程序,数据缺乏独立性。

人工管理阶段的特点如图 1-1 所示。

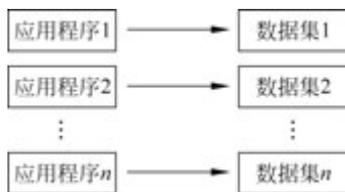


图 1-1 人工管理阶段的特点

2. 文件系统阶段

20 世纪 50 年代到 60 年代中期,在硬件技术的发展下,有了磁盘、磁鼓等可直接存取数据的设备,存取速度也加快了很多。软件方面,在操作系统中有了专门的数据管理软件,又被人们称为文件系统。

(1) 文件管理系统管理数据特点如下。

① 数据可以长期保存,具有多种形式的文件,如顺序文件或者索引文件等。

② 由文件系统管理数据,初步形成了对数据的 4 种操作方式,增加、删除、更新以及查询。

③ 由专用程序(一般由用户定义)负责对应用程序和数据提供存取方法的改变,应用程序和数据之间保持一定的独立性。

(2) 文件系统管理数据的缺点如下。

① 数据冗余大。在文件系统中,文件是面向应用程序的,当不同的应用程序具有部分相同的数据时,也必须建立各自的文件,不能共享这些相同的数据,因此数据冗余大,浪费存储空间。

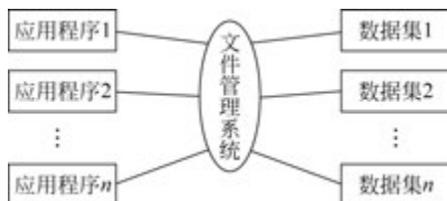


图 1-2 文件系统阶段的特点

② 数据维护困难。由于部分相同的数据重复存储,各自存取,会造成数据的不一致性,给数据的维护带来困难。

文件系统管理阶段的特点如图 1-2 所示。

3. 数据库系统阶段

20 世纪 60 年代末期以来,计算机管理的对象越来越多,范围变得更加广泛,数据量也急速增长,此时,出现了专门管理数据的软件系统——数据库管理系统。运用数据库技术进行数据管理,解决了多用户、多应用共享数据的需求。

数据库系统阶段的特点如下。

(1) 数据库系统实现整体数据的结构化,在数据库中,数据不仅对应一个应用,可以面向完全组织的复杂数据结构。

(2) 数据共享性高,冗余度低,易于扩充。

(3) 数据独立性高,其中物理独立性指的是用户的应用程序与存储在磁盘上数据库中数据的物理存储是相互独立的,逻辑独立性则指的是用户的应用程序与数据库的逻辑结构是相互独立的。

(4) 数据由数据库管理系统统一管理和控制,具有安全性、完整性以及并发控制等特点,且具备数据库备份和恢复的功能。

数据库系统阶段的特点如图 1-3 所示。

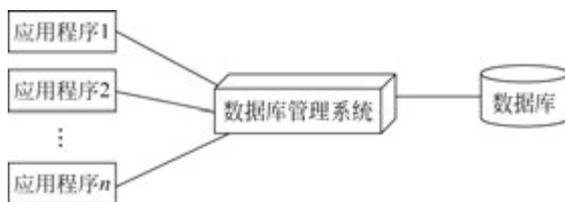


图 1-3 数据库系统阶段的特点

4. 分布式数据库系统阶段

20 世纪 80 年代,分布式数据库系统开始走上历史的舞台。分布式数据库系统是在集中式数据库系统基础上发展而来的,是计算机技术和网络技术结合的产物。一个分布式数据库在逻辑上看上去就只是一个统一的整体,但实际分别存储在不同的物理节点上。一个应用程序可通过网络连接访问分布在不同物理节点的数据库。用户只负责执行全局应用,并不关心访问的是哪个物理节点的数据库。

随着互联网、大数据的飞速发展,单台服务器上的性能已不能满足现实数据变化的需要,将原有的集中式系统改造为分布式系统,同时完成降低成本、增长性能的目的。在这种需求下,未来大规模分布式事务型数据库成为解决分布式系统数据存储、管理问题的主要方向。

分布式数据库系统阶段的特点如下。

(1) 更快的访问速度。分布式数据库系统可以并发地从多个服务器同时读取数据,保证了数据的访问速度。

(2) 更高的可靠性。分布式数据库系统将数据分布于多个物理节点,并利用适当的冗余度提供更高的可靠性。因为一个物理节点出现故障不会引起整个系统崩溃。

(3) 更强的可扩展性。分布式数据库增加了新的节点,因此更易操作。如果系统中增加新的功能或者应用场景,实现起来比集中式系统要更加方便和灵活。

任务 1.2 数据模型——从现实世界到信息世界



相关知识

现实生活中人们见过各种模型,如飞机模型、建筑设计的沙盘、三维立体地图等,这些都是具体的模型。模型可以形象、直观地展示事物的本来特征,使人们对事物的了解更加全面且深入。例如,建造一栋建筑前,用图纸、沙盘等方式展示设计,以便建造时有所参考。数据库系统是使用计算机技术对客观事物进行管理的系统,管理前也需要对客观事物进行抽象和模拟,以创建适合于数据库系统管理的数据模型。本任务介绍数据模型的相关知识。

1. 现实世界

在现实世界中,把客观存在并可以相互区分的事物称为实体。实体既可以是实实在在的事物,也可以是根据事物描述需要抽象而来的。同一个事物,在抽象时,可以根据不同的应用抽象为不同的实体。例如,在学校管理中,对所有来上学的对象抽象为“学生”实体;而在户籍管理中,又将同样的这些学生以及其他的人抽象为“公民”这个实体。例如,教师、学生、课程、项目、案件等均为实体。

每一个实体都具备一些特征,例如,“教师”实体具有教师编号、姓名、职称、生日等特征;对于“学生”实体,则具有学号、姓名、性别、所在班级等特征。

实体集则是具有相同特征的一类实体的集合。例如,所有的教师构成教师实体集;所有的项目构成项目实体集。在一个实体集中,用于区分每个实体的特征称为标识特征。例如,对于“案件”这个实体,案件编号可以作为标识特征。案件类型不能作为标识特征的原因是,通过案件类型不能唯一地识别出具体是哪一个案件。

2. 信息世界

从事物的特征到计算机中的数据表示,涵盖了以下三个领域。

- (1) 现实世界。存在中的客观世界,称为现实世界。
- (2) 信息世界。人们对现实世界的对象抽象后,在人们头脑中的反映。
- (3) 机器世界。信息世界的信息在机器世界中以数据的形式存储。

在信息世界中,描述数据的术语有实体、实体集、属性和联系等。

1) 实体和实体集

可相互区别的对象称为实体(entity),实体可以是具体的人、事或者物,也可以是抽象出来的。实体的集合称为实体集。

2) 属性

实体具有的特性称为属性(attribute)。一个实体由若干特征也就是若干属性来表达。例如,“教师”实体由教师编号、姓名、性别、职称、所在院部等属性组成,(00001,张扬,男,教授,大数据中心)表达了一个具体的教师。

3) 联系

现实世界中无独立存在的事物,事物之间通常都是有联系的,这些联系在信息世界中的反映则是实体内部的联系以及实体与实体之间的联系(relationship)。

实体内部的联系通常是指实体的各属性之间的联系。例如,某学校教务管理系统数据库的“课程”实体中,一门课没有或者有一门先修课程,如图 1-4 所示。



图 1-4 实体内部的联系

实体之间的联系通常指不同实体间的联系,例如“学生”和“课程”之间存在“选修”的联系,如图 1-5 所示。

这些联系可以划分为一对一联系、一对多联系以及多对多联系。

(1) 一对一联系。如果实体集 A 中的每一个实体仅与实体集 B 的一个实体联系;反之则称实体集 A 与实体集 B 具有一对一联系,表示为 1:1。例如,一个人对应一张身份证,一张身份证属于一个人。这里把人和身份证之间的联系称为“属于”关系,如图 1-6 所示。



图 1-5 实体和实体之间的联系



图 1-6 一对一的联系

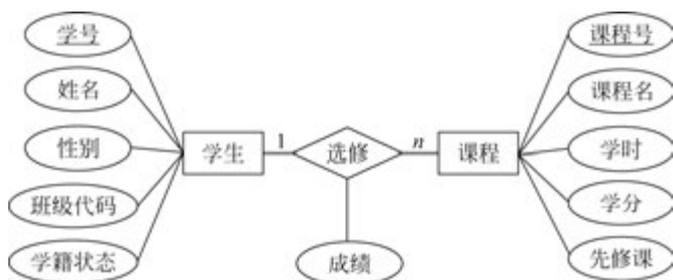
(2) 一对多联系。如果实体集 A 中的任一个实体与实体集 B 的多个实体有对应关系,而实体集 B 中的任一个实体在实体集 A 中仅有一个实体与之对应,则称实体集 A 与实体集 B 的联系是一对多的联系,表示为 1:n。例如“班级”实体集和“学生”实体集,一个班级有多名学生,而一个学生只能属于一个班级,“班级”和“学生”之间存在一对多的联系,如图 1-7 所示。



图 1-7 一对多的联系

(3) 多对多联系。如果实体集 A 中的任一个实体与实体集 B 的多个实体有对应关系,而实体集 B 中的任一个实体在实体集 A 中也有多个实体与之对应,称实体集 A 与实体集 B 的联系是多对多的联系,表示为 $m:n$ 。例如,“学生”实体集和“课程”实体集,一个学生可以选修多门课程,一门课程也可以有多名学生来选修,如图 1-8 所示。

如上的联系图称为“实体—联系”图,这种描述现实世界的方法也称为“实体—联系方法”(entity relationship approach),简称 E-R 方法。用 E-R 图表示的概念模型是独立于数据库管理系统支持的数据模型,是各种数据模型的基础。



任务 1.3 关系数据库



相关知识

现实世界中,实体内部的联系或者实体与实体之间的各种联系均用关系模型来表示。1970年,埃德加·科德提出了关系模型的概念,关系模型的概念简单清晰,便于理解和使用。关系数据模型中无论是实体还是实体间的联系均由关系来表示。在实际的关系数据库中的关系也称为表,一个关系数据库就是由若干表组成。本节介绍关系数据库的相关知识。

1. 关系模型

关系数据库使用关系模型作为数据的组织形式。关系模型允许设计者通过数据库规范化地提炼,去建立一个信息一致性的模型。访问以及其他操作细节由DBMS的引擎来处理,这些细节在逻辑模型中并不需要反映。

20世纪80年代以来,计算机企业新开发的DBMS绝大多数支持关系模型,非关系系统的产品大多数加上了关系模型的接口。关系模型由关系数据结构、关系操作集合以及关系的完整性约束3个部分组成。

1) 关系数据结构

现实世界的实体内部以及实体与实体之间的各种联系均用关系来表示,从用户角度看,关系模型中数据的逻辑结构是一个规范化的二维表。下面以学生表为例,介绍关系模型中的一些专业术语。

(1) 关系(relation)。一个关系对应一个二维表,每个关系都有关系名。例如,表1-1所示的关系名称为“学生”。

表 1-1 “学生”表(t_student)

学号 (student_id)	姓名 (name)	性别 (sex)	出生日期 (birthday)	籍贯 (native)	民族代码 (nation)	班级代码 (class_code)	学籍状态 (strtus)
2020010201	刘丽	女	2002-05-06	海南	1	102	1
2020010202	符飞宇	男	2001-12-07	西藏	4	102	1

续表

学号 (student_id)	姓名 (name)	性别 (sex)	出生日期 (birthday)	籍贯 (native)	民族代码 (nation)	班级代码 (class_code)	学籍状态 (strtus)
2020010203	张江	男	2003-11-20	新疆	5	102	2
2020010204	李想	女	2002-07-19	福建	1	103	1
2020010205	火风	男	2002-05-26	广东	1	103	1

(2) 元组(tuple)。表格中的一行称为一个元组,如学生表中的一条学生记录。

(3) 属性(attribute)。表格中的一列即为一个属性,属性对应存储文件中的一个字段。

(4) 码(key)。码是能唯一标识实体的属性集,如学号是“学生”实体的码。

注:在关系数据库理论中,key 通常译为码;在数据库工程应用中,通常称为键。二者意义相同。

(5) 候选码(candidate key)。如果在一个关系中,存在多个属性(或属性组合)都能用来唯一标识该关系的元组,这些属性(或属性组合)就称为该关系的候选码。

如在“学生”关系中,若姓名无重名,则学号和姓名都是“学生”关系的候选码。

(6) 主码(primary key)。在一个关系的若干候选码中指定作为码的属性(或属性组合),称为该关系的主码,如“学生”关系中应将学号设置为主码。

(7) 主属性(primary attribute)。包含在候选码中的属性都称为主属性,如“学生”关系中学号和无重名的姓名都是主属性。

(8) 非主属性(nonprimary attribute)。不包含在候选码中的属性称为非主属性。

如“学生”关系中性别、出生日期以及籍贯都是主属性。

(9) 域(domain)。属性的域是指属性的取值范围。例如,性别的域为('男','女'),姓名的域则是任意字符串的组合。

(10) 关系模式(relation schema)。关系模式是对关系的描述,一般表示为“关系名(属性 1,属性 2,⋯,属性 n)”,例如“学生(学号,姓名,性别,出生日期,班级代码,学籍状态)”。

在关系模型中,实体用关系模式来表示,实体和实体之间的联系也用关系模式来表示。例如,图 1-7 所示的关系模型中,学生、课程和先修课程三者间的联系可用 3 个关系模式来表示。

学生(学号,姓名,性别,出生日期,班级代码,学籍状态)

课程(课程号,课程名,学分,先修课号)

选修(学号,课程号,成绩)

2) 关系操作集合

关系操作包括增加、删除、更新以及查询数据,这些操作的操作对象是关系,操作结果还是关系,也就是元组的集合。在关系操作中用户只要指出做什么,不需要说明怎么做。

3) 关系的完整性约束

关系的完整性约束主要包括实体完整性、参照完整性以及用户定义完整性三个方面。实体完整性和参照完整性是关系模型必须满足的完整性的约束条件,用户定义完整性则

是指针对具体应用需要自行定义的约束条件。

(1) 实体完整性(entity integrity)。若属性 A 是基本关系 R 的主属性,则属性 A 不能取空值且不能有重复值。

例如,“学生”关系中的“学号”属性不可为空且不可重复。

(2) 参照完整性(referencing relation)。在关系模型中,实体和实体间的联系都用关系来表示,这样,关系与关系之间要通过某些属性来建立起它们间的联系。

例如“学生”关系的主码是“学号”,“课程”关系的主码是“课程号”,而“选修”关系的主码是(学号,课程号)，“选修”关系中的“学号”必须是一个在“学生”关系中存在的学号,“选修”关系中的“课程号”必须是一个在“课程”关系中存在的课程号。例如,在图 1-9 中,“选课”关系的前 7 行记录符合参照完整性的要求,最后一行(5,K005)则不符合参照完整性的要求。因为 5 号学生在“学生”关系中不存在,K005 号课程在“课程”关系中也不存在。

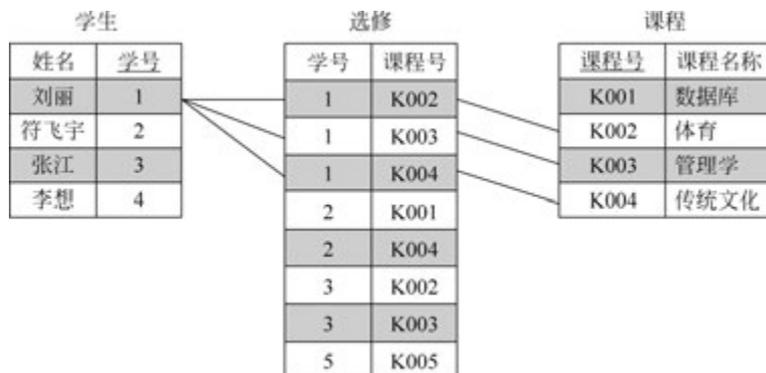


图 1-9 参照完整性

参照完整性的定义是: 设 F 是基本关系 R 的一个或一组属性,但不是关系 R 的非码属性,如果 F 与基本关系 S 的主码 K 相对应,则称 F 是基本关系 R 的外码 (foreign key)。其中,关系 R 称为参照关系 (referencing relation),基本关系 S 称为被参照关系。

例如,“选修”关系的“学号”属性参照“学生”关系的“学号”属性,“选修”关系的“课程号”属性参照“课程”关系的“课程号”属性。因此,“学号”属性和“课程号”属性就是选修关系的外码。主码所在的表称为主表,外码所在的表称为从表。

(3) 用户定义完整性。关系数据库系统都支持实体完整性和参照完整性,但除此之外,用户还需要一些特殊的约束条件。例如,某个属性的取值必须唯一,某个非主属性的取值不得为空,某个属性的取值范围须为 $0 \sim 100$ 等,这些都属于用户自定义完整性。

2. 关系数据库的规范化理论

任务 1.2 中设计的数据库是否正确且合理? 用户该依据什么规则进行检查? 如有不合理的地方又该用什么样的方法进行修改呢? 这些是关系数据库的规范化理论所讨论的问题。

1) 数据库设计中存在的问题

在“学生及选课”表中操作数据时,会发生插入异常、删除异常、数据冗余和更新异常等问题。

(1) 插入异常。新成立一个系部,无法将系部信息插入表中;或者新生刚报到,还未选择课程,那么这个学生的数据也无法插入表中。

(2) 删除异常。某个学生因为个人原因,要将目前选修的课程全部删除,那么这个学生的基本数据也会被删除。

(3) 数据冗余。一个系部有多名学生,每名学生又选修了若干门课程,这个系部的系主任信息存储行数与学生选课数据的行数相同。

(4) 更新异常。若某个系部更换系主任,就要修改这个系部所有学生选修的每门课程对应的元组数据中系主任的信息。若修改不完全或错误,会造成数据的不一致现象。

2) 数据依赖

表 1-2 所示的“学生及选课”关系中,之所以会出现插入异常、删除异常、数据冗余和更新异常等问题,皆因关系中存在完全函数依赖、部分函数依赖以及传递函数依赖的问题。

表 1-2 “学生及选课”关系

学号	姓名	系部名称	系主任	课程名称	成绩
S10201	刘丽	网络	李瑶	数据库	98
S10201	刘丽	网络	李瑶	传统文化	75
S10202	符飞宇	网络	李瑶	体育	65
S10203	张江	经管	符秉义	企业管理	95
S10203	张江	经管	符秉义	传统文化	78
S10204	李想	艺术	王艺	传统文化	99
S10205	火风	艺术	王艺	传统文化	

函数依赖的定义如下。

在关系 $R(X, Y)$ 中, X, Y 都是 R 的属性集,当 X 取值确定时, Y 的取值唯一确定,则称 Y 函数依赖 X , 或称 X 函数决定 Y , 记作 $X \rightarrow Y$ 。

例如,有以下关系:

学生(学号, 姓名, 性别, 籍贯, 系部名称)

在“学生”关系中,“学生”的学号确定时,该学生的姓名可唯一确定(学号 \rightarrow 姓名)。同理,有(学号 \rightarrow 系部名称)。

(1) 完全函数依赖和部分函数依赖

在关系 $R(X, Y)$ 中, X, Y 都是 R 的属性集。若 $X \rightarrow Y$, 且对 X 的任一个真子集 X' , 不存在 $X' \rightarrow Y$, 则 $X \rightarrow Y$ 是完全函数依赖, 否则称为部分函数依赖。

例如,有以下关系:

学生及选课(学号, 系部名称, 系主任姓名, 课程名称, 成绩)。

该关系中存在如下函数依赖。

学号 \rightarrow 系部名称, 系部名称 \rightarrow 系主任
(学号, 课程名称) \rightarrow 成绩

显然,学生及选课的关键字应为(学号, 课程名), 且有以下函数依赖。