第3章 项目数据库的设计与实现

本章首先介绍项目数据库设计的详细过程,然后对MySQL数据库管理 系统的安装和配置步骤进行介绍,接着讲解如何在集成开发环境IDEA中配 置并操作MySQL数据库,最后完成项目数据库表的结构设计、数据库表的 创建以及数据库表中初始数据的插入。

3.1 数据库设计



3.1.1 系统数据流图

第2章通过对系统的业务需求进行分析,明确了本系统应包括系统管理员、考生、教务管理员和招生管理员四类用户角色。将以上四类用户角色作为数据的源头,可以得到系统的顶层数据流,如图3.1所示。





通过对注册登录、信息录入、文件上传、考试报名、信息确认、考场分配、成绩导入、录 取查询等主要业务操作流程进行分析,可以将系统的顶层数据流图进一步细化分解。对主要业 务数据流进行细化分解后,得到如图3.2所示的数据流图。其中,用户信息表中的数据来源于 考生用户注册功能,招考文件表中的数据来源于招生管理员用户的招考文件管理功能,招考信 息相关表格中的数据来源于招生管理员用户的招考信息管理功能,报名信息表中的数据主要来 源于考生用户的在线报名功能和教务管理员的排考功能,成绩及录取信息表中的数据主要来源 于招生管理员用户的成绩信息导入功能。



图3.2 主要业务分解后的数据流



3.1.2 概念模型 E-R 图

概念模型设计阶段的任务是将现实世界中的数据模型抽象为一定的信息结构。这种信息结构常用E-R图(Entity Relationship Diagram,实体联系图)来表示。每个业务功能模块都需要操作一个或多个数据实体(如管理员实体、考生实体、考试科目实体等),最终这些数据实体会被持久化至数据库表中。本项目在概念模型设计阶段抽象出的主要实体包括考试科目、招考专业和考生,这三个实体之间的联系如图3.3所示。



图3.3 系统主要功能E-R图

除了在图3.3中出现的三个实体之外,系统中还包括一些其他实体。下面针对系统中所有 的实体逐一进行介绍。

1. 招考专业实体

招考专业实体如图3.4所示,它用于描述考生报考的专业信息,其属性包括id、专业名称、 计划招生人数。其中,id属性为招考专业实体的主键,其数据类型可设计为int类型,并设置自 增属性。专业名称不能为空且具备唯一性,也可作为该实体的主键。计划招生人数用来表示本 专业的计划招生人数。



图3.4 招考专业实体

2. 考试科目实体

考试科目实体如图3.5所示,它用于描述考试科目的相关信息,其属性包括考试科目id、课程名称、隶属专业、科目考试开始时间和科目考试结束时间。不同专业考试科目的课程名称可能会相同,但对应的课程考试大纲可能不同,考试内容以及考试时间也不一定相同,因此并不能使用课程名称来区分不同的考试科目。考试科目实体的主键只能选用id属性,其数据类型可设计为int类型,并设置自动增长属性。其余属性都是对考试科目信息的详细描述,其中,隶属专业这个属性关联的是招考专业实体的专业名称属性,其取值只能是专业实体中已存在的某个专业名称。



图3.5 考试科目实体

3. 考生实体

考生实体如图3.6所示,它用于描述考生的基本信息,其属性包括id、用户名、身份证号、 密码、姓名、性别和民族。其中,id属性为考生实体的主键,可设计为int类型,并设置自增属 性。用户名属性非空且具备唯一性,也可作为考生实体的主键。其他属性都是对考生个人基本 信息的描述,其中,性别属性可设置为枚举类型,规定可枚举的属性值为"男"和"女"。



4. 站点实体

站点实体如图3.7所示,它用于描述站点基本信息,其属性包括id、学校名称、系统名称、考试名称、联系电话、地址、邮编和版权备案信息。其中,id属性为站点实体的主键,可设计为int类型,并设置自增属性。学校名称、系统名称和考试名称描述了当前考试的基本信息,联系电话、地址和邮编是招考学校的基本信息,版权备案信息用来存储当前站点的版权及备案信息。



5. 管理员实体

系统共有三类管理员用户,管理员实体中除了具有描述管理员基本信息的属性之外,还有 一个用户组属性用于区分不同类别的管理员。管理员实体如图3.8所示,它的主键也是id,可以 设计为int类型,且为自增类型。管理员实体的其他属性还包括管理员用户名、管理员密码、管 理员姓名和所在组。



图3.8 管理员实体

6. 阶段实体

在2.3节中,我们将招生考试报名系统划分为了六个固定的系统阶段,通过系统阶段设置 来区分并限制一些功能操作。阶段实体如图3.9所示,它用于对各个不同系统阶段进行说明描述,其属性包括id、阶段编号、阶段名称、阶段描述以及是否当前阶段。其中,id属性为阶段 实体的主键,可设计为int类型,并设置自增属性。"是否当前阶段"这个属性可以由招生管理 员进行设置,该属性取值为0或者1,当取值为1时表示该阶段属于当前阶段。需要注意的是, 在六个系统阶段中,任意时刻有且仅有一个阶段属于当前阶段。



图3.9 阶段实体

7. 日志实体

日志实体如图3.10所示,它用于描述各类用户登录系统的信息,其属性包括id、用户名、 用户组、登录时间以及登录IP地址。其中,id属性为日志实体的主键,可设计为int类型,并设 置自增属性。用户名来源于登录系统的各类用户的用户名,用户组用于区分不同角色的用户。



图3.10 日志实体

8. 文档实体

文档实体如图3.11所示,它用于描述上传至服务端的招考文件信息,其属性包括id、文档 名称、文档路径和上传时间。其中,id属性为主键,可设计为int类型,并设置自增属性。其他 属性均是对上传文件信息的具体描述。



3.1.3 逻辑结构设计



逻辑结构设计阶段需要将概念模型设计阶段得到的概念模型转换为逻辑模型。 目前最常用的逻辑模型是关系模型,关系模型可以进一步转换为关系数据库中的关系。在设计时,除了要将概念模型E-R图中的实体转换为关系模型之外,实体之间的

联系也需要进行转换。在图3.3所示的E-R图中,存在三组联系,分别是招考专业和 考试科目之间的包含联系、招考专业和考生之间的报名联系以及招考专业和考生之间的录取联 系。其中,招考专业和考试科目之间的包含联系易于转换,通过在考试科目实体中包含隶属专 业这个属性就可以将它们之间的一对多的联系表示出来。招考专业和考生之间的报名联系及录 取联系比较复杂,可以通过单独生成对应的报名关系和录取关系进行转换。

招考专业和考生之间的报名联系是系统中的主要联系,它不仅记录考生与招考专业之间的 对应关系,还会记录确认信息、准考证号、考场号等由联系本身产生的属性。报名联系涉及考 生、报考专业、管理员等多个实体,因此可以将该联系转换为如下所示的关系模式: 报名(id,报考人用户名,报考专业,毕业院校,毕业年份,联系电话,邮政编码,邮寄地址, 收件人,照片位置,确认状态,确认管理员用户名,准考证号,考场编号,考场地址,座位 号,考试签到时间)

其中,报考人用户名来源于考生实体的用户名属性,报考专业来源于招考专业实体的专业 名称属性,确认管理员用户名来源于管理员实体的管理员用户名属性。

考生报考成功后的录取情况对应成绩和录取关系,用于记录学生参加考试后的总成绩和录 取情况,包含考生实体和招考专业实体,因此该联系可以转换为如下所示的关系模式:

成绩(id, 准考证号, 用户名, 姓名, 报考专业, 总成绩, 备注, 录取信息)

概念模型设计阶段的所有实体和联系总共需要转换为十个关系表,分别为站点信息表 (site)、管理员用户表(admin)、阶段信息表(phase)、招考专业表(major)、考试科目 表(course)、考生表(student)、报名信息表(enroll)、成绩及录取信息表(grade)、文 档表(doc)和登录日志表(log)。

站点信息表(site)用于存储站点的基本信息,它的表结构设计如表3.1所示。

字 段 名	数 据 类 型	说明
id	int	ID, 主键, 自增
site_school	varchar(100)	学校名称,非空
site_name	varchar(100)	系统名称,非空
site_test_name	varchar(100)	考试名称,非空
site_location	varchar(100)	地址,非空
site_zip_code	char(6)	邮编,非空
size_contact	varchar(20)	联系电话,非空
site_copy	varchar(100)	版权备案信息,非空

表3.1 站点信息表(site)的表结构

管理员用户表(admin)用于存储管理员的基本信息,它的表结构设计如表3.2所示。

表3.2 管理员用户表(admin)的表结构

字 段 名	数 据 类 型	说 明
id	int	ID,主键,自增
admin_username	varchar(20)	管理员用户名,唯一性,非空
admin_name	varchar(20)	管理员姓名,非空
admin_password	varchar(50)	管理员密码,非空
admin_group	varchar(20)	用户组,非空

阶段信息表(phase)用于存储六个系统阶段信息,并体现当前所处的系统阶段,它的表结构设计如表3.3所示。

表3.3 阶段信息表(phase)的表结构

字段名	数 据 类 型	说明
id	int	ID, 主键, 自增
phase_number	tinyint	阶段编号,非空
phase_name	varchar(20)	阶段名称,非空
phase_description	varchar(255)	阶段描述,非空
phase_is_current	tinyint	是否当前阶段,0表示否,1表示是

招考专业信息表(major)用于存储招考专业的基本信息,它的表结构设计如表3.4所示。

表3.4 招考专业信息表(major)的表结构

字段名	数据类型	说 明
id	int ID, 主键, 自增	
major_name	varchar(50)	专业名称,非空,唯一性
major_plan_number	int	计划招生人数,非空

考试科目表(course)用于存储考试科目的详细信息,它的表结构设计如表3.5所示。

表3.5 考试科目表(course)的表结构

字 段 名	数 据 类 型	说 明
id	int	ID,自增,主键
course_name	varchar(50)	课程名称,非空
major_name	varchar(50)	隶属专业,非空
course_start_timestamp	int	科目考试开始时间,非空
course_end_timestamp	int	科目考试结束时间,非空

考生表(student)用于存储考生的基本信息,它的表结构设计如表3.6所示。

表3.6 考生表(student)的表结构

字 段 名	数 据 类 型	说 明	
id	int ID, 主键, 自增		
student_username	varchar(20)	用户名,非空,唯一性	
student_password	varchar(50)	密码,非空	
student_name	varchar(20)	姓名,非空	
student_sex	enum	性别,枚举值为"男""女"	
student_id_code	char(18)	身份证号,非空,唯一性	
student_nation	varchar(40)	民族,非空	

报名信息表(enroll)用于存储考生的报名信息、报名确认状态、考场、考试签到等信息,它的表结构设计如表3.7所示。

项目数据库的设计与实现

字 段 名	数 据 类 型	说 明
id	int	ID, 主键, 自增
student_username	varchar(20)	报考人用户名,非空,唯一性
major_name	varchar(50)	报考专业,非空
enroll_school	varchar(50)	毕业院校,非空
enroll_graduate_year	char(4)	毕业年份,非空
enroll_contact	varchar(20)	联系电话,非空
entroll_zip_code	char(6)	邮政编码,非空
enroll_address	varchar(100)	邮寄地址,非空
enroll_receiver	varchar(20)	收件人,非空
enroll_photo	varchar(128)	照片位置
enroll_confirm	tinyint	确认状态,非空,默认为0
admin_username	varchar(20)	确认管理员用户名
enroll_exam_number	varchar(20)	准考证号
enroll_room_number	varchar(5)	考场编号
enroll_room_location	varchar(100)	考场地点
enroll_seat_number	char(2)	座位号
enroll_sign_timestamp	int	考试签到时间

表3.7 报名信息表(enroll)的表结构

成绩及录取信息表(grade)用于存储考生最终的总成绩以及考生的录取情况信息,它的表结构设计如表3.8所示。

表3.8 成绩及录取信息表(grade)的表结构

字 段 名	数 据 类 型	说 明
id	int	ID,主键,自增
enroll_exam_number	varchar(20)	准考证号,非空,唯一性
student_name	varchar(20)	考生姓名,非空
major_name	varchar(50)	报考专业,非空
grade_total	int(11)	总成绩,非空
grade_note	varchar(255)	备注
enroll_note	varchar(255)	录取信息,非空

登录日志表(log)用于存储系统用户的登录信息,它的表结构设计如表3.9所示。

表3.9 登录日志表(log)的表结构

字段名	数 据 类 型	说 明
id	int	ID, 主键, 自增

字 段 名	数 据 类 型	说明
log_username	varchar(20)	用户名,非空
log_group	varchar(20)	用户组,非空
log_timestamp	int	登录时间,非空
log_ip	varchar(100)	登录IP地址,非空

续表

文档表(doc)用于描述上传文件的信息,它的表结构设计如表3.10所示。

表3.10 文档表(doc)的表结构

字 段 名	数 据 类 型	说 明
id	int	ID,主键,自增
doc_name	varchar(255)	文档名称
doc_uri	varchar(255)	文档路径
doc_upload_timestamp	int	上传时间





MySQL是一个RDBMS(Relational Database Management System,关系数据库管理系统),由瑞典MySQLAB公司开发,目前属于Oracle公司。MySQL是最好的数据库应用软件之一,尤其是在Web应用开发方面,成为了一些中小企业业务系统数据库的首选。

下面以"MySQL 5.1.68"在Windows 10系统下的安装为例介绍MySQL的安装和基本配置 操作。首先双击下载到本地系统中的MySQL安装文件"mysql-essential-5.1.68-win64.msi", 打开安装向导,弹出如图3.12所示的欢迎界面,单击"Next >"按钮,打开选择安装类型的界 面,如图3.13所示。



图3.12 安装欢迎界面

图3.13 选择安装类型

在图3.13所示的界面中勾选"Typical"进行典型安装,也可以选择"Custom"进行定制安装。单击"Next >"按钮,打开准备安装程序的界面,如图3.14所示。

Java Web 项目实训教程

第2版

图3.14所示的界面中显示了当前的安装路径等信息,单击"Install"按钮,程序开始安装,安装过程需要等待一段时间,如图3.15所示。

当程序安装完成后,显示当前安装版本的MySQL信息,分别如图3.16和图3.17所示。

在图3.17中单击"Next >"按钮,弹出提示配置MySQL服务和注册MySQL服务的界面,如图3.18所示。勾选"Configure the MySQL Server now"配置MySQL服务,然后单击 "Finish"按钮,完成MySQL的安装,同时开启MySQL服务的配置向导,如图3.19所示。



More ...

图3.14 准备安装程序



<section-header><image><image><list-item><list-item><list-item>

图3.15 程序安装中

图3.16 MySQL信息1



图3.18 安装完成并选择配置服务

图3.17 MySQL信息2

< Back Next > Cancel



图3.19 MySQL配置向导

对MySQL实例进行向导式配置,选择详细配置和开发机模式,分别如图3.20和图3.21所示。 按照向导选择多功能数据库,然后配置数据库文件的存放位置,分别如图3.22和图3.23所示。 将数据库连接数配置为100,如图3.24所示。MySQL的默认端口号为3306,无须修改, 如图3.25所示。



图3.20 选择详细配置









MySQL Server Instance Configuration Wizard

Configure the MySOL Server 5.1 server instance.

connections will be assumed.

Please set the approximate number of concurrent connections to the server.

Select this option for database applications that will not require a high number of concurrent connections. A number of 20

Choose this option for highly concurrent applications that may have at any one time up to 500 active connections such as heavily loaded web servers.

rrent

Concurrent connections

MySQL Server Instance Configuration

Decision Support (DSS)/OLAP

Manual Setting

22

Online Transaction Processing (OLTP)



将默认字符集修改为"utf8",如图3.26所示。之后配置服务与环境变量,如图3.27所示。 最后一步是MySQL的安全设置,设置数据库密码,并开放允许远程登录,如图3.28所示。



图3.26 设置字符集

图3.27 配置服务与环境变量

•

< Back Next > Cancel



图3.28 MySQL数据库安全设置

所有配置项设置完成后,将按顺序执行各项配置。如图3.29所示,单击"Execute"按钮 执行,配置成功后的界面如图3.30所示。至此,MySQL数据库配置成功,MySQL服务开启成 功,安全设置应用成功。



图3.29 准备执行配置

图3.30 配置执行成功

MySQL数据库的安装和配置要点总结如下:

(1)根据需要,选择典型安装或者选择定制安装均可。

(2)安装目录和数据库文件存放位置选择默认位置即可。

(3) MySQL数据库的默认端口号为3306, 一般情况下无须修改。

(4) MySQL数据库的默认字符集为拉丁语,配置过程中需要修改为支持中文的"utf8" 字符集。

(5)将MySQL安装为Windows服务,设置为开机自启动,然后在Windows环境变量中包含MySQL的bin目录。

(6)需要为数据库根账户设置一个初始密码,可以不用创建匿名账户。

(7)如果需要重新配置MySQL服务器,可以从Windows系统的"开始"菜单打开"所有程序"→"MySQL"→"MySQL Server 5.1"→"MySQL Server Instance Config Wizard"菜单项,在出现的配置向导中重新配置MySQL服务器。

(8)在MySQL安装路径下有一个名为"my.ini"的文件,这是MySQL的选项配置文件, MySQL启动时会自动加载该文件中的配置选项。可以通过此文件来修改MySQL的一些默认配 置项。



数据库及表的创建可以选择使用命令行的方式,也可以使用一些MySQL管理工具。我们 首先基于命令行的方式创建数据库,然后在集成开发环境IDEA中创建数据库表。如果要对数 据库进行操作,首先需要登录数据库,打开Windows系统的命令提示符,输入以下命令登录数 据库。

mysql -uroot -proot

其中, "-uroot"代表登录时使用的用户名为root, "-proot"代表当前root用户的密码为 root, 该密码是在安装MySQL数据库时设定的。

当登录成功后,可以执行以下命令创建数据库。该行命令表示创建一个名称为"bmxt"的数据库,且使用的默认编码格式为"utf8",这个数据库就是报名系统项目要使用的数据库。

create database bmxt default character set utf8;





1. 创建项目

完成了项目数据库的创建之后,就可以在集成开发环境IDEA中创建数据库表了,需要先完成报名系统项目的创建。打开IDEA开发工具,单击 "+Create New

103

Project",打开"New Project"面板,如果已经打开了IDEA,则可以通过主菜单 "File"→"New"→"Project....",打开"New Project"面板。在"New Project"面板 的左侧选择"Java",单击"Next"按钮,然后在"Project Name"一栏中输入项目名称 "bmxt",并通过"Project location"一栏来选择项目的位置,然后单击"Finish"按钮完成项 目的创建。

由于项目是Java Web项目,因此需要增加对应的框架支持。在IDEA主界面的"1:Project" 视图窗口中,右击项目名称"bmxt",在弹出的菜单中选择"Add Frameworks Support....", 打开"Add Frameworks Support"面板,勾选Java EE下的"Web Application"选项,然后单击 "OK"按钮,完成对应Java EE框架的支持。

2. 配置Tomcat服务器

项目最终会被部署在Web服务器下运行,需要为项目配置一个Web服务器,可以配置一 个Tomcat服务器。首先选择主菜单"Run"→"Edit Configurations...",或者在快捷菜单 栏单击"Add Configurations...",打开"Run/Debug Configurations"面板,单击面板左上 角的"+"按钮,在弹出的"Add New Configuration"列表中展开所有条目,选择"Tomcat Server"→"Local"选项,打开"Tomcat Server"的配置界面。接着在"Tomcat Server"配 置界面的"Server"选项页中找到"Application Server"选项,单击"Configure..."按钮打开 Tomcat配置页面,在其中正确配置Tomcat服务器对应的安装目录,单击"OK"按钮就完成了 Tomcat服务器的配置。返回"Run/Debug Configurations"面板,切换到"Deployment"选项 页,在右侧单击"+"按钮,选择"Artifact...",然后将"Application context"一栏的项目部 署路径修改为"/bmxt",完成项目在Tomcat中的部署。最后单击"OK"按钮保存以上所做的 配置即可。

3. 启动并检查输出

在IDEA主界面中打开"8:Services"视图窗口,右击"Tomcat Server",在弹出的菜单中单击"Run",就可以启动运行Tomcat服务器了。服务器启动后,IDEA会开启服务器输出窗口 "Server Output"以及两个日志查看窗口"Tomcat Localhost Log"和"Tomcat Catalina Log"。 如果在以上窗口中输出的中文呈现为乱码,则可以尝试按照1.5.2节中提供的方案解决。

📕 3.5 在集成开发环境中创建数据库表



3.5.1 配置数据库连接

IDEA开发工具支持可视化的数据库连接配置,配置完成后就可以在集成环境中操作数据 库了,更加便于项目的开发与测试。下面将详细介绍如何在IDEA集成开发环境中配置MySQL 数据库连接。

如图3.31所示,选择主菜单"View"→"Tool Windows"→"Database",打开数据库工具窗口面板。

接着在数据库工具窗口面板中单击右侧的"+"按钮,然后选择"Data Source"→"MySQL", 如图3.32 所示, 打开"Data Sources And Drivers"面板。

😫 <u>F</u> ile <u>E</u> dit	<u>V</u> iew <u>N</u> avigate <u>C</u> ode Analyze <u>R</u> efactor <u>B</u> u	uild R <u>un T</u> ools VC <u>S W</u> in
	<u>I</u> ool Windows	Project Alt+1
hmyt	Appearance •	★ Favorites Alt+2
= Dmxt	Quick Definition Ctrl+Shift+I	Q Find Alt+3
🕎 🔳 Project	Show Siblings	▶ Run Alt+4
e 🗸 📰 purx	Ouick Type Definition	Debug Alt+5
📕 > 🔚 .io	1 Ouick Documentation Ctrl+Shift+空格	Problems Alt+6
> • • •	Open Module Settings F12	Structure Alt+7
> in w	Recent Files Ctrl+E Recently Changed Files	Services Alt+8 * Ant
> IIİli Exter	Recent Locations Ctrl+Shift+E	S Build
> 🗞 Scrat	Recent Changes	Endpoints

图3.31 打开Database工具窗口面板



图3.32 打开"Data Sources and Drivers" 面板

打开的"Data Sources And Drivers"面板如图3.33所示,在面板的右侧"General"标签页中可以对MySQL数据库连接进行详细的配置。其中,"User"一栏需要输入数据库的用户名,项目开发测试时使用的是MySQL的root账号,因此需要输入"root";"Password"一栏需要输入对应数据库账号的密码"root",该密码是在安装MySQL数据库时配置的; "Database"一栏需要填写项目开发时使用的数据库名称"bmxt"。上述配置完成后,在 "URL"一栏中会自动生成"bmxt"数据库的链接地址URL。

除了配置数据库账号的用户名、密码、默认连接的数据库之外,还需要配置加载数据 库驱动程序才能实现对数据库的具体操作,配置的数据库驱动程序版本要与系统中安装的 MySQL版本相匹配。单击图3.33中的"Driver:MySQL",在弹出的下拉选项中选择"Go to Driver...",接着在面板左侧的驱动程序列表中选择"MySQL for 5.1",然后在面板右侧的"Driver files"一栏中添加具体的驱动程序jar包,可在本地系统中选择已下载好的数据库驱动包"mysql-connector-java-5.1.47.jar",或者使用IDEA提供的对应版本的数据库驱动包,如图3.34所示。

Data Sources and Drivers		×
Data Sources Drivers	Name: bmxt@localhost	
+ @ ⊮ ⊻ ← →	Comment	
Project Data Sources		
🔪 bmxt@localhost	General Options SSH/SSL Schemas Advance	ced
	Connection type: default Driver: MySQL	
Problems	Host: localhost	Port: 3306
	User: root	
	Password:	Save: Forever 💌
	Database: bmxt	
	URL: jdbc:mysql://localhost:3306/bm Ovenides settings above	xt e*
	Test Connection MySQL	
(?)		OK Cancel Apply
Data Sources and Drivers Data Sources Drivers Data Sources Drivers	图3.33 配置MySQL连	接 ×
+ - ⓑ ← →	ingree to sit	
Complete Support	Comment:	*
₩2 H2	General Options Advanced	
A HSQLDB (Local)	Class and state Driver	
HSQLDB (Remote)	class. continysqijubcibriver	•
IBM Db2	Driver Files	
IBM DD2 (JTOpen)	+ -	
111 1011 DUE 101 354, 1014	IIII MySQL Connector/J ver. 5.1.47 [stable]	Show License

图3.34 选择对应的MySQL驱动

jdbc:mysql://{host:localhost}?[:{port::3306}][/{database}?][\?<&,user={user},p...

jdbc:mysql://address=\(protocol=tcp\)\(<\)\(,host={host:host_ipv6:localhost},...

jdbc:mysql:///{database}?[\?<&,user={user},password={password},junixsocke...

Cancel

Apply

URL templates

+ - +

default

default

Unix Socket

Create Data Source

MariaDB
 Microsoft SQL Server
 Microsoft SQL Server (jTds)

MongoDB

MySQL

Oracle

?

MySQL for 5.

PostgreSQL器 Snowflake

Oracle for 9.X, 10.X, 11.1

配置完数据库驱动之后,单击"OK"按钮,返回"Data Sources And Drivers"面板界 面。此时,"Driver"属性配置处显示配置好的数据库驱动版本"MySQL for 5.1"。接着就可 以单击"Test Connection"进行连接测试了,如果连接成功,会显示连接成功的提示文本,如 图3.35所示。配置数据库连接成功后,单击"OK"按钮返回即可。

	Name: bmxt@localhost	0
- 6 / 1	$\leftarrow \rightarrow$ Comment:	2
oject Data Sources		
bmxt@localhost	General Options SSH/SSL Schemas Adva	vanced
	Connection type: default Driver: MySQL for 5	5.1
Problems	Host: localhost	Port: 3306
	User: root	
	Password: Chidden>	Save: Forever
	Database: bmxt	
Succeeded DBMS: MySQL (ver. 5.1.40 Case sensitivity: plain=low Driver: MySQL Connector)-community) ver, delimited=lower Java (ver. mysql-connector-java-5.1.47 (Revision: ed3190d80c049b1de29), JDBC4.0)	Сору
fe1903b1ecb4a96a917f7 Ping: 62 ms SSL: no		

图3.35 MySQL数据库连接成功

配置创建了新的数据库连接之后,在Database面板上就会显示出当前已创建成功的数据库 连接"bmxt@localhost",如图3.36所示。



```
山
初频讲解
```

Java Web

106

项目实训教程

第2版

```
图3.36 连接配置成功后的Database面板
```

3.5.2 创建数据库表并初始化数据

首先需要在项目中创建一个SQL脚本文件,用于执行创建数据库表及初始化数据的脚本。在项目bmxt的src目录下,创建包"cn.bmxt.db",然后在该包的右键菜单中选择 "New"→"File",接着输入文件名称"init.sql",就完成了SQL脚本文件的创建。接下来 就可以在文件"cn.bmxt.db/init.sql"中编写SQL脚本逐一创建数据库表并初始化数据了。

对应3.1节中的数据库设计,在报名系统数据库中共需要创建10个表,下面将分别对每个 表的结构定义以及表中数据的初始化进行介绍。

1. 创建并初始化站点信息表

创建站点信息表结构的脚本如代码3.5.2-01所示。

【代码3.5.2-01】创建站点信息表site,代码位置: /src/cn.bmxt.db/init.sql。

```
01 drop table if exists `site`;
02 create table `site`(
03 `id` int auto_increment primary key comment 'ID',
04 `site_school` varchar(100) not null comment '学校名称',
```

```
05 `site_name` varchar(100) not null comment '系统名称',

06 `site_test_name` varchar(100) not null comment '考试名称',

07 `site_location` varchar(100) not null comment '地址',

08 `site_zip_code` char(6) not null commentT '邮编',

09 `site_contact` varchar(100) not null commentT '联系方式',

10 `site_copy` varchar(100) not null commentT '版权备案信息'

11 ) ENGINE = InnoDB AUTO_INCREMENT = 1 DEFAULT CHARSET = utf8;
```

第3章

项目数据库的设计与实现

107

根据逻辑结构设计阶段对站点信息表(site)的表结构设计,书写创建数据库表site的 SQL语句,完成数据表site的创建。代码3.5.2-01中,第01行首先判断数据库中是否已经存在表 site,如果存在则先将表site删除。第02行~第11行是创建数据表site的SQL语句,其中第03行中 的 "auto_increment"用于设置id字段的自增属性,"primary key"用于定义数据库表site的主 键为id字段。

书写完创建站点信息表site的脚本之后,使用鼠标左键选中第01行~第11行代码,然后右击 鼠标,在弹出的菜单中单击"Execute"执行,首次执行时需要选择新的数据库会话连接"New Session",使用3.5.1节中配置好的数据库连接"bmxt@localhost"。执行成功后可在数据库 bmxt中创建表site,在"Database"窗口中将"bxmt@localhost"逐级展开后可看到创建好的数 据库表site,如图3.37所示。



图3.37 Database窗口中呈现的数据库表

接着在创建好的数据表site中初始化一条数据,作为站点的初始信息,在"init.sql"文件中书写插入一条数据的SQL语句,如代码3.5.2-02所示。

【代码3.5.2-02】在数据库表site中初始化一条站点信息,代码位置: /src/cn.bmxt.db/init.sql。

01 insert into `site` values (1,'XXX 大学','2023 年高职升本报名系统','2023 年高职升本专业考试','XXX 市 XXX 路 XXX 号','300000','022-888888888','版权 所有 © 2023-2024 津 ICP 备 XXXX 号');

在执行代码3.5.2-01时已经创建了与数据库连接的会话,后续再去执行"init.sql"文件中

的脚本时只需要选中要执行的代码,单击右键菜单中的"Execute"选项执行即可;也可以先选中要执行的代码,然后单击文件编辑窗口左上角的绿色"▶"按钮执行;还可以使用键盘中的组合快捷键Ctrl+Enter来执行选中的脚本。

2. 创建并初始化管理员信息表

根据逻辑结构设计阶段对管理员信息表(admin)的表结构设计,在文件"init.sql"中书写创建数据库表admin的SQL语句,如代码3.5.2-03所示。

【代码3.5.2-03】创建管理员信息表admin,代码位置: /src/cn.bmxt.db/init.sql。

语句执行后就完成了管理员信息表admin的创建。系统初始时仅需要一个系统管理员,其他管理员信息可以通过管理员维护功能模块中的相关功能由系统管理员添加。在管理员信息表admin中添加一条表示系统管理员信息的数据,同样是在文件"init.sql"中书写一条插入数据的语句,参见代码3.5.2-04,语句执行后就会在admin表中初始化一条系统管理员信息。

【代码3.5.2-04】初始化一条系统管理员信息,代码位置: /src/cn.bmxt.db/init.sql。

```
01 insert into `admin` values (1, 'sys_admin', '超管员', '000000', '系统管理员');
```

3. 创建并初始化阶段信息表

根据逻辑结构设计阶段对阶段信息表(phase)的表结构设计,在文件"init.sql"中书写 创建数据库表phase的SQL语句,如代码3.5.2-05所示。

【代码3.5.2-05】创建阶段信息表phase,代码位置:/src/cn.bmxt.db/init.sql。

```
01 drop table if exists `phase`;
02 create table `phase`(
03 `id` int auto_increment primary key comment 'ID',
04 `phase_number` tinyint not null comment '阶段编号',
05 `phase_name` varchar(20) not null comment '阶段名称',
06 `phase_description` varchar(255) not null comment '阶段描述',
07 `phase_is_current` tinyint comment '是否当前阶段: 0- 否, 1- 是'
08 ) ENGINE = InnoDB AUTO_INCREMENT = 1 DEFAULT CHARSET = utf8;
```

代码3.5.2-05执行之后就完成了阶段信息表phase的创建。接下来需要将六个阶段的信息分别添加至phase表中,其中第一阶段为当前系统所处阶段。在文件"init.sql"中书写6条插入数据的语句,参见代码3.5.2-06,接着同时选中这6条插入语句并执行,完成阶段信息表中阶段信息数据的初始化。

【代码3.5.2-06】初始化系统阶段表phase,代码位置: /src/cn.bmxt.db/init.sql。

- 01 insert into phase values(1, 1, '基础信息维护阶段', '(1)系统管理员:站 点信息管理; (2)招生管理员:招考文件管理。', 1);
- 02 insert into phase values(2, 2, '开放报名阶段', '(1)考生:用户注册、在 线报名、修改报名信息。', 0);
- 03 insert into phase values(3, 3, '现场确认阶段', '(1)招生管理员:现场确认;
 (2)尚未确认考生:用户注册、在线报名、修改报名信息。', 0);
- 04 insert into phase values(4, 4, '排考阶段', '(1)教务管理员: 准考证号分配、 考场安排。', 0);
- 05 insert into phase values(5, 5, '准考证打印与考试阶段', '(1)考生: 准考 证打印、考试签到。', 0);
- 06 insert into phase values(6, 6, '成绩及录取查询阶段', '(1)招生管理员: 成绩及录取信息维护; (2)考生:成绩及录取信息查询。', 0);

4. 创建招考专业信息表

根据逻辑结构设计阶段对招考专业信息表(major)的表结构设计,在文件"init.sql"中 书写创建数据库表major的SQL语句,如代码3.5.2-07所示。选中这段代码并执行,就完成了数 据库表major的创建。

【代码3.5.2-07】创建招考专业信息表major,代码位置: /src/cn.bmxt.db/init.sql。

```
01 drop table if exists `major`;
02 create table `major`(
03 `id` int auto_increment primary key comment 'ID',
04 `major_name` varchar(50) not null unique comment '专业名称',
05 `major_plan_number` int(11) not null comment '计划招生人数 '
06 ) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=utf8;
```

5. 创建考试科目信息表

根据逻辑结构设计阶段对考试科目信息表(course)的表结构设计,在文件"init.sql"中 书写创建数据库表course的SQL语句,如代码3.5.2-08所示。选中这段代码并执行,就完成了数 据库表course的创建。

【代码3.5.2-08】创建考试科目信息表course,代码位置: /src/cn.bmxt.db/init.sql。

```
01 drop table if exists `course`;
02 create table `course`(
03 `id` int auto_increment primary key comment 'ID',
04 `course name` varchar(50) not null comment '课程名称',
```

- 05 `major name` varchar(50) not null comment '隶属专业',
- 06 `course_start_timestamp` int not null comment '科目考试开始时间',
- 07 `course end timestamp` int not null comment '科目考试结束时间 '
- 08) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=utf8;

6. 创建考生信息表

根据逻辑结构设计阶段对学生信息表(student)的表结构设计,在文件"init.sql"中书写 创建数据库表student的SQL语句,如代码3.5.2-09所示。选中这段代码并执行,就完成了数据 库表student的创建。

【代码3.5.2-09】创建考生信息表student,代码位置: /src/cn.bmxt.db/init.sql。

```
01 drop table if exists `student`;
02 create table `student`(
03
    `id` int auto increment primary key comment 'ID',
     `student username` varchar(20) not null unique comment '用户名',
04
05
    `student_password` varchar(50) not null comment '密码',
    `student name` varchar(20) not null comment '姓名',
06
    `student sex` enum('男','女') not null comment '性别',
07
    `student id code` char(18) not null unique comment '身份证号',
80
     `student_nation` varchar(40) not null comment ' 民族 '
09
10 ) ENGINE=InnoDB AUTO INCREMENT=1 DEFAULT CHARSET=utf8;
```

7. 创建报名信息表

根据逻辑结构设计阶段对报名信息表(enroll)的表结构设计,在文件"init.sql"中书写 创建数据库表enroll的SQL语句,如代码3.5.2-10所示。选中这段代码并执行,就完成了数据库 表enroll的创建。

【代码3.5.2-10】创建报名信息表enroll,代码位置: /src/cn.bmxt.db/init.sql。

```
01 drop table if exists `enroll`;
02 create table `enroll`(
    `id` int auto increment primary key comment 'ID',
03
04
    `student_username` varchar(20) not null unique comment '报考人用户
   名',
05
    `major name` varchar(50) not null comment '报考专业',
    `enroll school` varchar(50) not null comment '毕业院校',
06
    `enroll_graduate_year` char(4) not null comment '毕业年份',
07
    `enroll_contact` varchar(20) not null comment ' 联系电话 ',
80
09
    `enroll zip code` char(6) not null comment '邮政编码',
    `enroll address` varchar(100) not null comment '邮寄地址',
10
    `enroll receiver` varchar(20) not null comment '收件人',
11
    `enroll photo` varchar(128) null comment '照片位置',
12
    `enroll_confirm` int not null default '0' comment '确认状态',
13
    `admin_username` varchar(20) null comment '确认管理员用户名',
14
```

```
15 `enroll_exam_number` varchar(20) null comment ' 准考证号 ',
16 `enroll_room_number` varchar(5) null comment ' 考场编号 ',
17 `enroll_room_location` varchar(100) null commentT ' 考场地点 ',
18 `enroll_seat_number` char(2) null comment ' 座位号 ',
19 `enroll_sign_timestamp` int default 0 comment '考试签到时间点 '
20 ) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=utf8;
```

第3章

项目数据库的设计与实现

111

8. 创建成绩及录取信息表

根据逻辑结构设计阶段对成绩及录取信息表(grade)的表结构设计,在文件"init.sql" 中书写创建数据库表grade的SQL语句,如代码3.5.2-11所示。选中这段代码并执行,就完成了 数据库表grade的创建。

【代码3.5.2-11】创建成绩及录取信息表grade,代码位置:/src/cn.bmxt.db/init.sql。

```
01 drop table if exists `grade`;
02 create table `grade` (
    `id` int auto increment primary key comment 'ID',
03
    `enroll_exam_number` varchar(20) not null unique comment ' 准考证
04
   号 ',
05
    `student name` varchar(20) not null comment '考生姓名',
    `major name` varchar(50) not null comment '报考专业',
06
    `grade total` int not null comment ' 总成绩 ',
07
    `grade note` varchar(255) null comment '备注',
80
09
    `enroll note` varchar(255) not null comment '录取信息'
10 ) ENGINE=InnoDB AUTO INCREMENT=1 DEFAULT CHARSET=utf8;
```

9. 创建用户登录日志信息表

根据逻辑结构设计阶段对用户登录日志信息表(log)的表结构设计,在文件"init.sql" 中书写创建数据库表log的SQL语句,如代码3.5.2-12所示。选中这段代码并执行,就完成了数 据库表log的创建。

【代码3.5.2-12】创建用户登录日志信息表log,代码位置: /src/cn.bmxt.db/init.sql。

```
01 drop table if exists `log`;
02 create table `log`(
03 `id` int auto_increment primary key comment 'ID',
04 `log_username` varchar(20) not null comment '用户名',
05 `log_group` varchar(20) not null comment '用户组',
06 `log_timestamp` int not null comment '登录时间',
07 `log_ip` varchar(100) not null comment '登录 IP地址'
08 ) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=utf8;
```

10. 创建文档信息表

根据逻辑结构设计阶段对文档信息表(doc)的表结构设计,在文件"init.sql"中书写创

建数据库表doc的SQL语句,如代码3.5.2-13所示。选中这段代码并执行,就完成了数据库表 doc的创建。

【代码3.5.2-13】创建文档信息表doc,代码位置: /src/cn.bmxt.db/init.sql。

```
01 drop table if exists `doc`;
02 create table `doc` (
03 `id` int auto_increment primary key comment 'ID',
04 `doc_name` varchar(255) not null unique comment '文档名称',
05 `doc_uri` varchar(255) not null comment '文档路径',
06 `doc_upload_timestamp` int not null comment '上传时间'
07 ) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=utf8;
```

11. 完整的SQL脚本

代码3.5.2-01~代码3.5.2-13中所有代码构成了文件"init.sql"的第一个完整版本,我们将 该版本编号为0.01,并记录如代码3.5.2-14所示。

【代码3.5.2-14】文件/src/cn.bmxt.db/init.sql,版本0.01。

```
01 ... 此处为【代码 3.5.2-01】中的全部代码
02 ... 此处为【代码 3.5.2-02】中的全部代码
03 ... 此处为【代码 3.5.2-03】中的全部代码
04 ... 此处为【代码 3.5.2-04】中的全部代码
05 ... 此处为【代码 3.5.2-05】中的全部代码
06 ... 此处为【代码 3.5.2-06】中的全部代码
07 ... 此处为【代码 3.5.2-07】中的全部代码
08 ... 此处为【代码 3.5.2-09】中的全部代码
09 ... 此处为【代码 3.5.2-09】中的全部代码
10 ... 此处为【代码 3.5.2-10】中的全部代码
11 ... 此处为【代码 3.5.2-11】中的全部代码
12 ... 此处为【代码 3.5.2-12】中的全部代码
13 ... 此处为【代码 3.5.2-13】中的全部代码
```