**实验一 项目准备**

## **实验目标**

本实验任务是了解“Q-ITOffer”锐聘网站后台管理系统的需求分析、开发环境搭建以及数据库创建：

* 【实验1-1】 项目需求分析。
* 【实验1-2】 项目开发环境搭建。
* 【实验1-3】 项目数据库创建。

【实验1-1】项目需求分析

### 功能描述

“Q\_ITOffer”锐聘网站后台管理系统是对“Q\_ITOffer”锐聘网站前台数据进行查看、审核、修改、删除、添加等操作的综合管理平台，该系统主要包括对企业和职位的管理、求职者简历的管理以及对整个网站系统的管理。该系统分系统管理员、企业管理员和普通用户三个角色，其中系统管理员拥有该系统的所有操作权限，主要负责分配用户权限以及对系统用户的管理；企业管理员拥有企业和职位的管理权限，主要负责企业和职位信息以及职位申请的查询、添加、修改和删除；普通用户拥有对求职者简历管理的权限，主要负责简历的查询和审核。

### 实验分析

1. “Q\_ITOffer”锐聘网站后台管理系统功能结构如图1- 1所示。



图1- 1 系统功能结构图

1. “Q\_ITOffer”锐聘网站后台管理系统拥有3个角色：系统管理员、企业管理员、普通用户，各角色功能用例图如图1- 2至图1- 4所示。



图1- 2 系统管理员用例图



图1- 3 企业管理用户用例图



图1- 4 普通用户用例图

1. “Q\_ITOffer”锐聘网站后台管理系统数据库设计

本项目数据库要求采用Oracle 11g数据库，在教材贯穿项目所建的数据库基础上，新增加“Q\_ITOffer”锐聘网站后台管理系统用户表tb\_users，改变后的数据库实体关系图如图1- 5所示。



图1- 5 数据库实体关系图

## **【实验1-2】项目开发环境搭建**

### 知识点回顾

* JDK安装与配置，参见教材【附录A】
* Tomcat服务器的下载安装及使用，参见教材【1.2.3】章节
* Eclipse工具下载及使用，参见教材【附录B】
* Eclipse中的Tomcat配置，参见教材【1.2.4】章节
* 项目的创建过程，参见教材【1.2.5】章节

### 功能描述

安装配置Java开发环境；安装配置Tomcat服务器；在Eclipse中配置Tomcat服务器；使用Eclipse工具，创建Dynamic Web Project项目；在项目中创建包目录结构，了解包的命名和功能含义。

### 实验步骤

1. 参照教材【附录A】下载、安装配置JDK 8.0。
2. 参照教材【1.2.3】章节下载、安装配置Tomcat 8.0服务器。
3. 参照教材【附录B】下载、安装开发工具Eclipse Luna。
4. 参照教材【1.2.4】章节在Eclipse Luna中配置Tomcat 8.0服务器。
5. 在Eclipse中创建“Q\_ITOffer”锐聘网站后台管理系统“Q\_ITOffer\_Experiment01”，操作步骤如下。

在Eclipse的菜单中依次选择“File🡪New🡪 Dynamic Web Project”菜单项，如图1- 6所示；或直接在项目资源管理器空白处右键单击，在弹出菜单中选择“New🡪Dynamic Web Project”菜单。

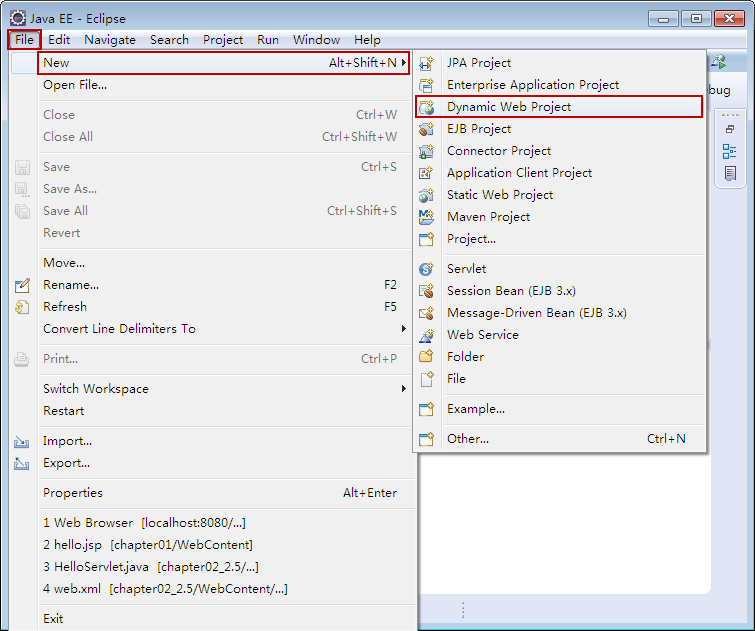


图1- 6 新建项目菜单

在弹出的创建项目对话框中，如图1- 7所示，输入项目名称“Q\_ITOffer\_Experiment01”，在“Dynamic web module version”选项中选择3.0版。

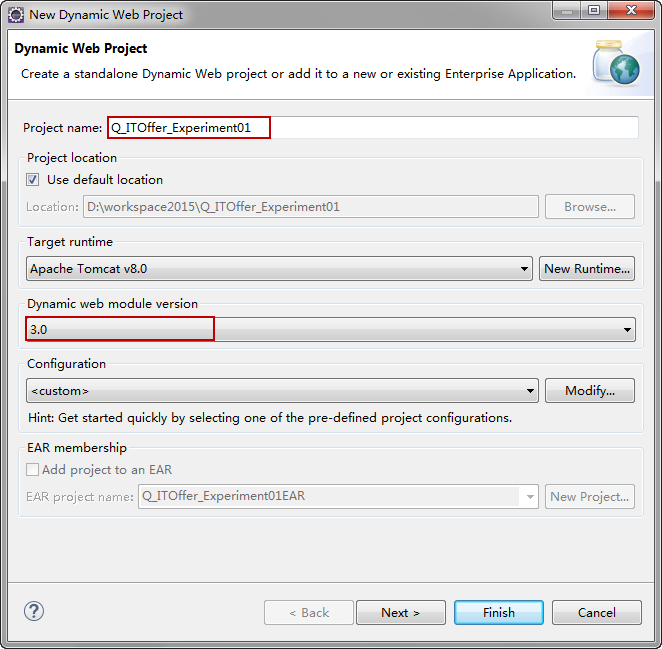


图1- 7 创建动态Web项目

依次点击“Next >”，在出现图1- 8所示窗口时，勾选上“Generate web.xml deployment descriptor”选项。

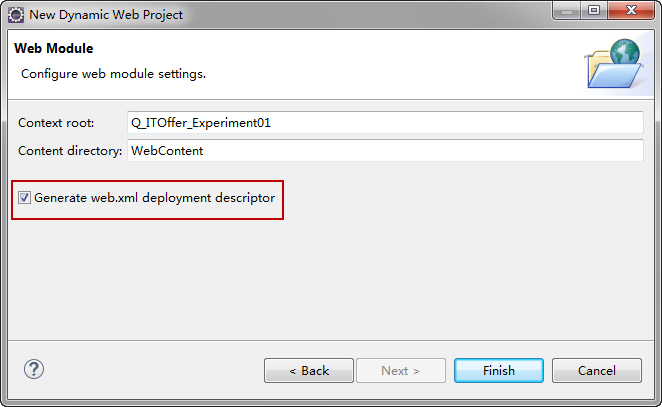


图1- 8 创建动态Web项目

点击“Finish”，完成“Q\_ITOffer\_Experiment01”项目的创建。创建后的项目目录结构如图1- 9所示。

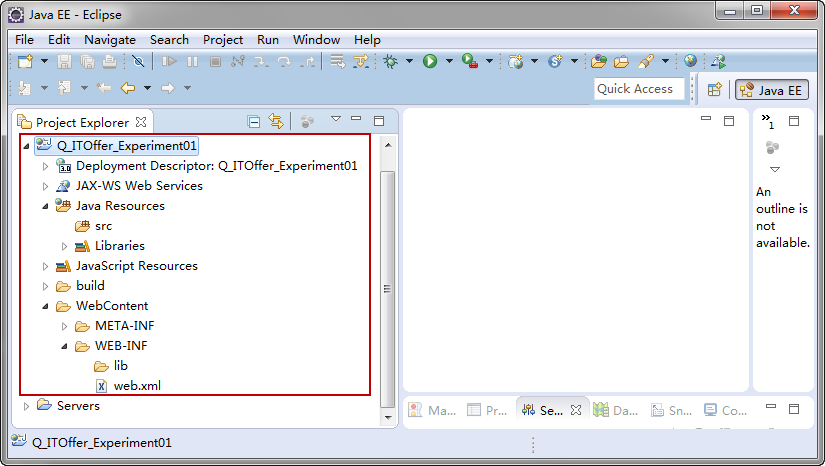


图1- 9 项目目录结构

1. 创建项目所需的资源目录，并了解各目录的含义。创建完成后的目录结构如图1- 10所示。

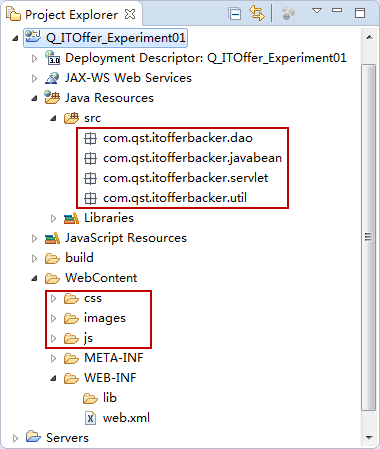


图1- 10 创建项目资源目录

图1- 10中css目录用于存放项目所需的样式表文件，即.css文件；images目录用于存放项目所需的图片；js目录用于存放项目使用的JavaScript文件；lib目录用于存放项目用到的第三方jar文件；com.qst.itofferbacker.dao包用于存放项目中各模块的数据库操作类；com.qst.itofferbacker.javabean包用于存放项目所需的JavaBean类；com.qst.itofferbacker.servlet包用于存放项目中用到的Servlet；com.qst.itofferbacker.util包用于存放项目所使用到的工具类。

## **【实验1-3】项目数据库创建**

### 功能描述

项目所需的数据库的创建及准备工作。

### 实验分析

1. 创建负责管理本项目的数据库用户和角色。
2. 创建表结构和项目所需的基础数据。
3. 创建项目获取数据库连接操作的工具类，并进行数据库连接测试。

### 实验步骤

1. 在Oracle数据库中使用SYSTEM账户创建本项目的数据库用户并授权，例如设置数据库用户名为：qstitoffer；密码为：qstitoffer123，相应操作如图1- 11所示。（注意：如果在教材的贯穿项目中已经完成此功能，则此处无需再重复操作。）

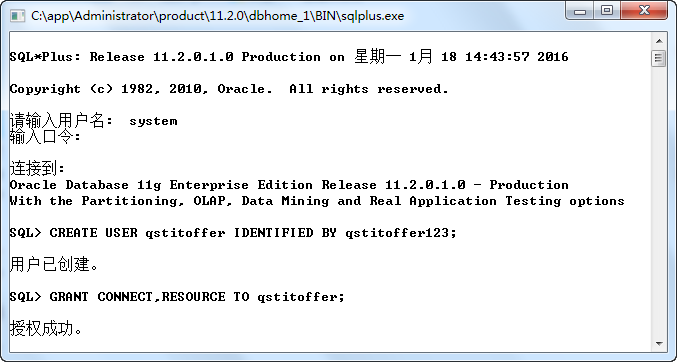


图1- 11 创建数据库用户并授权

1. 数据库用户和权限创建成功后，如果在教材的贯穿项目实现时未进行相应数据库表结构的创建，则需要读者先到QST青软实训在线教育平台www.moocollege.cn/JavaWeb下载项目前期所需的基础表结构和基础数据的创建脚本ITOffer.sql，然后将脚本保存到本地磁盘，按照图1- 12进行脚本执行操作。

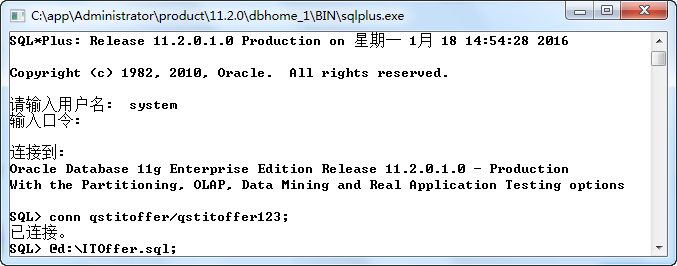


图1- 12 表结构创建SQL语句的执行

1. 本项目在上述SQL脚本所创建的表结构基础上，新增用于存放后台管理用户信息的用户表（tb\_users）及其基础数据，表结构如表1- 1所示，相应的SQL脚本如【代码1- 1】所示，执行方式同图1- 12。

表1- 1 用户信息表（TB\_USERS）

|  |  |  |  |
| --- | --- | --- | --- |
| **列名** | **类型** | **约束** | **描述** |
| USER\_ID | NUMBER | 主键 | 用户标识 |
| USER\_LOGNAME | VARCHAR2(50) |  | 登录名 |
| USER\_PWD | VARCHAR2(50) |  | 登录密码 |
| USER\_EMAIL | VARCHAR2(50) |  | 用户邮箱 |
| USER\_ROLE | NUMBER | 取值为1（系统管理员）、2（企业管理员）或3（普通用户） | 用户角色 |
| USER\_STATE | NUMBER | 取值为0（禁用）、2（启用） | 用户状态 |

**【代码1- 1】ITOfferBacker.sql**

-- @公司 青软实训

-- @作者 fengjj

-- 创建后台管理用户信息表

CREATE TABLE TB\_USERS(

USER\_ID NUMBER PRIMARY KEY,

USER\_LOGNAME VARCHAR2(50) NOT NULL,

USER\_PWD VARCHAR2(50) NOT NULL,

USER\_REALNAME VARCHAR2(50) NOT NULL,

USER\_EMAIL VARCHAR2(50) NOT NULL,

USER\_ROLE NUMBER NOT NULL,

USER\_STATE NUMBER NOT NULL,

CONSTRAINT TB\_USERS\_CHK1 CHECK (USER\_ROLE IN (1,2,3)),

CONSTRAINT TB\_USERS\_CHK2 CHECK (USER\_STATE IN (0,1))

);

-- 创建用户信息表所使用的序列

CREATE SEQUENCE SEQ\_ITOFFER\_USERS MINVALUE 0 START WITH 0 INCREMENT BY 1;

-- 用户信息表基础数据

INSERT INTO tb\_users(user\_id, user\_logname, user\_pwd, user\_realname, user\_email, user\_role, user\_state) VALUES(SEQ\_ITOFFER\_USERS.NEXTVAL,'admin','123456','青软实训','admin@test.com',1,1);

1. 将ojdbc6.jar文件复制到项目“Q\_ITOffer\_Experiment01”的lib目录下，在com.qst.itofferbacker.util包下创建DBUtil.java数据库操作工具类，代码如【代码1- 2】所示。

**【代码1- 2】DBUtil.java**

package com.qst.itofferbacker.util;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

/\*\*

\* 数据库连接获取和释放工具类

\* @公司 青软实训

\* @作者 fengjj

\*/

public class DBUtil {

static String user = "**qstitoffer**";

static String password = "**qstitoffer123**";

static String url = "**jdbc:oracle:thin:@127.0.0.1:1521:orcl**";

static {

try {

Class.forName("oracle.jdbc.driver.OracleDriver");

} catch (ClassNotFoundException e) {

e.printStackTrace();

}

}

**public static void main(String[] args) {**

**Connection conn = DBUtil.getConnection();**

**System.out.println("获取的Connection：" + conn);**

**}**

public static Connection getConnection() {

Connection conn = null;

try {

conn = DriverManager.getConnection(url, user, password);

} catch (SQLException e) {

e.printStackTrace();

}

return conn;

}

public static void closeJDBC(ResultSet rs, Statement stmt, Connection conn) {

if (rs != null) {

try {

rs.close();

} catch (SQLException e) {

e.printStackTrace();

}

}

if (stmt != null) {

try {

stmt.close();

} catch (SQLException e) {

e.printStackTrace();

}

}

if (conn != null) {

try {

conn.close();

} catch (SQLException e) {

e.printStackTrace();

}

}

}

}

1. 在Eclipse工具中打开上述代码，按照图1- 13所示方式运行此Java类，如果出现图1- 14所示类似结果，则表示获取数据库连接成功。

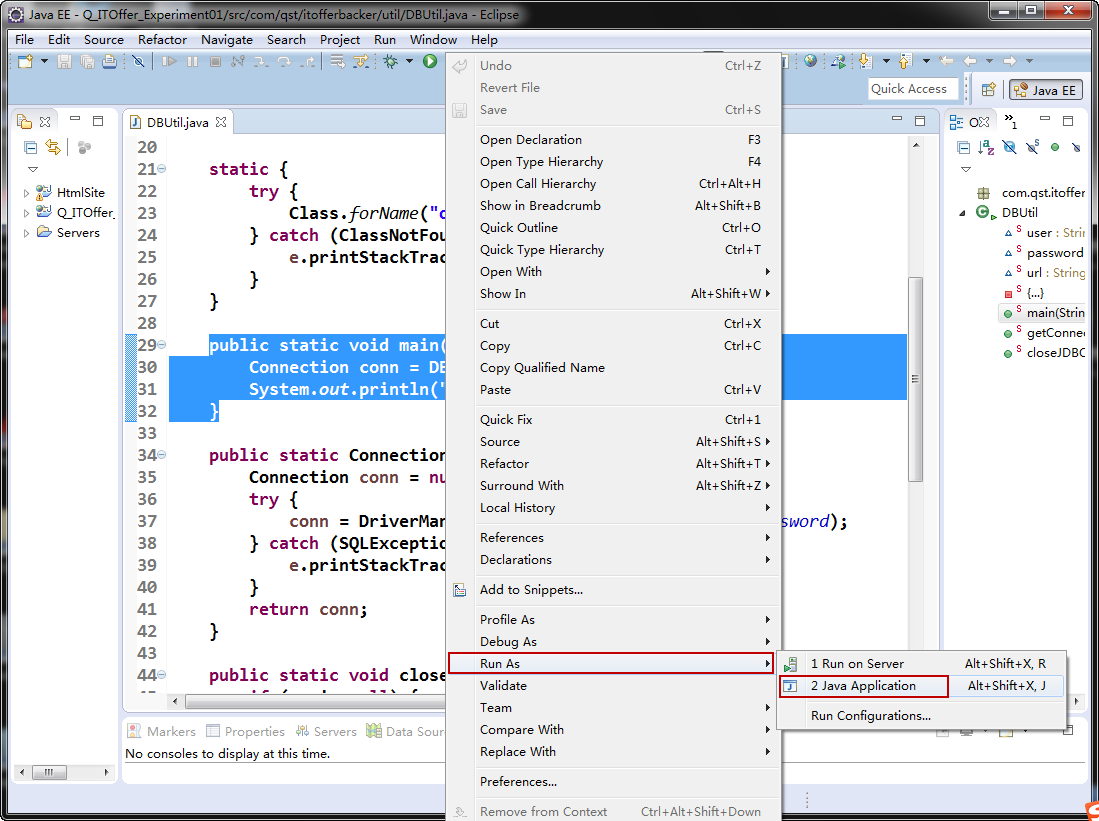


图1- 13 含main函数代码的执行

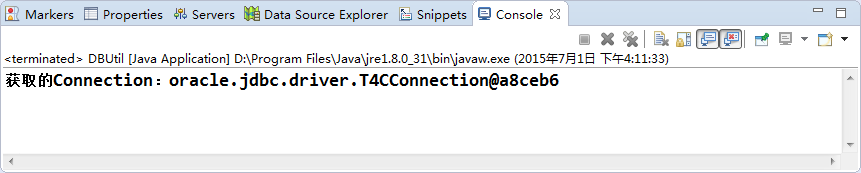


图1- 14 DBUtil.java的执行结果