

**数据库课程设计**

学生姓名： 张 超

班 学 号： 111141-27

选 题：英语学习助手

指导教师： 杨之江

**中国地质大学信息工程学院**

**2017年1月12日**

目录

[1、需求分析 2](#_Toc472438344)

[2、概念设计 2](#_Toc472438345)

[3、逻辑结构设计 5](#_Toc472438346)

[4、物理设计 6](#_Toc472438347)

[5、系统实施 11](#_Toc472438348)

[6、运行维护 17](#_Toc472438349)

[7、用户手册 19](#_Toc472438350)

[8、附录 24](#_Toc472438351)

[9、总结 27](#_Toc472438352)

**英语学习助手**

### 1、需求分析

（1）录入模块：

实现一些简单的添加功能，如添加单词、添加例句。在数据库层主要执行的是Insert语句，当用户执行添加后，会提示“添加成功”。

（2）试题模块：

这一模块主要是起着复习单词的作用，当进入此模块时，从数据库中随机的查找出4个单词，暂时保存在session中，并显示在页面中（只显示单词的意思）。让用户根据单词的意思填写单词，当用户提交答案后，系统会判断哪一个单词填写错了并将反馈信息显示在页面上；如果用户全部填写正确，则会跳转到一个页面，提示用户“全部回答正确”。

（3）分页浏览模块：

当用户进入此模块时，系统会在数据库中查出第一页的单词（默认一页有10个单词）显示在JSP页面中（以table的形式展示），用户可以在此页面上分页浏览单词、例句；用户可以点击上一页或者下一页，来显示不同的页面，也可以输入页号，跳转到自己想去的页面，当用户输入的页号不合理时（比如页号<0或者页号过大），会给予相应的提示；用户可以对自己浏览的单词进行编辑或删除操作。

（4）查询模块：

用户在此模块可以进行单词或例句的查找。查找的结果不唯一即模糊查询，例如当用户输入a，点击“查找”按钮，会显示所有包含a的单词或例句（最多10个）；用户可以对查找出来的单词进行编辑或删除操作。

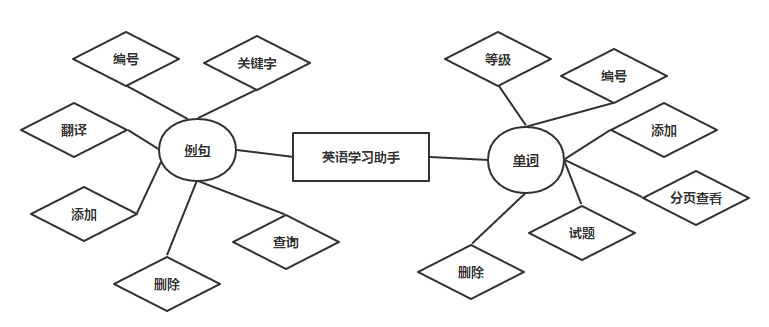
（5）翻译模块：

用户输入一段简单的英文句子，点击“翻译”后，在另外一个编辑框中显示出翻译对应的中文结果。

### 2、概念设计

#### 2.1概念模型（E-R图）

\*\*画图工具采用在线画图工具：http://www.processon.com/diagrams



#### 2.2数据字典

分类表T\_CATEGORY：

|  |  |  |
| --- | --- | --- |
| 字段名 | 类型 | 约束 |
| CID | NUMBER | PRIMARY KEY |
| CNAME | VARCHAR2(255 BYTE) |  |

四级单词表T\_CET4：

|  |  |  |
| --- | --- | --- |
| 字段名 | 类型 | 约束 |
| FID | NUMBER | PRIMARY KEY |
| FWORD | VARCHAR2(255 BYTE) |  |
| FMEAN | VARCHAR2(255 BYTE) |  |
| DIFF | NUMBER |  |
| COMMON | NUMBER |  |
| CID | NUMBER | REFERENCES T\_CATEGORY(CID) |

六级单词表T\_CET6：

|  |  |  |
| --- | --- | --- |
| 字段名 | 类型 | 约束 |
| SID | NUMBER | PRIMARY KEY |
| SWORD | VARCHAR2(255 BYTE) |  |
| SMEAN | VARCHAR2(255 BYTE) |  |
| DIFF | NUMBER |  |
| COMMON | NUMBER |  |
| CID | NUMBER | REFERENCES T\_CATEGORY(CID) |

例句表T\_PHRASE：

|  |  |  |
| --- | --- | --- |
| 字段名 | 类型 | 约束 |
| PID | NUMBER | PRIMARY KEY |
| PSENTENCE | VARCHAR2(255 BYTE) |  |
| PMEAN | VARCHAR2(255 BYTE) |  |
| DIFF | NUMBER |  |
| COMMON | NUMBER |  |
| CID | NUMBER | REFERENCES T\_CATEGORY(CID) |

用户表T\_USER：

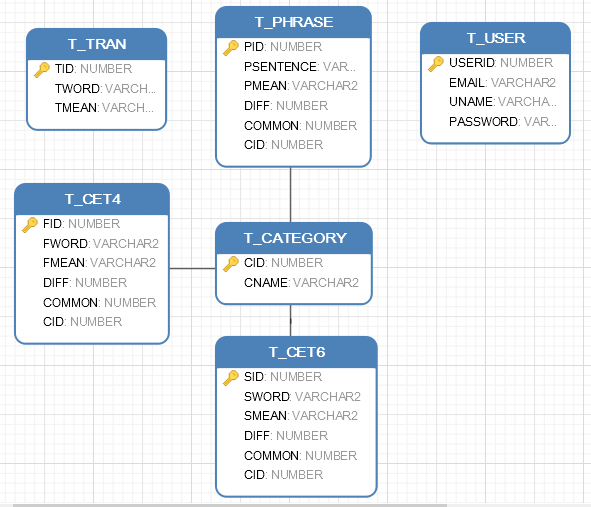
|  |  |  |
| --- | --- | --- |
| 字段名 | 类型 | 约束 |
| USERID | NUMBER | PRIMARY KEY |
| EMAIL | VARCHAR2(255 BYTE) |  |
| UNAME | VARCHAR2(255 BYTE) |  |
| PASSWORD | VARCHAR2(255 BYTE) |  |

翻译表T\_TRAN：

|  |  |  |
| --- | --- | --- |
| 字段名 | 类型 | 约束 |
| TID | NUMBER | PRIMARY KEY |
| TWORD | VARCHAR2(255 BYTE) |  |
| TMEAN | VARCHAR2(255 BYTE) |  |

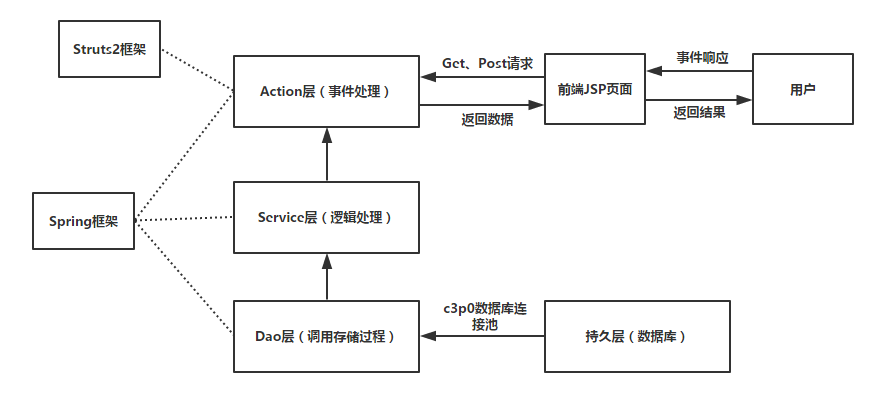
### 3、逻辑结构设计

#### 3.1数据库表之间的关系



#### 3.2系统逻辑设计

系统逻辑结构图：



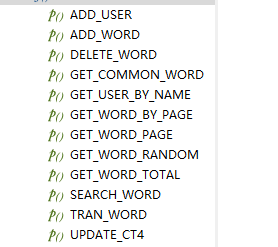
注解：

本英语学习助手系统采用B-S系统设计，前台跟用户的页面采用JSP页面，控件主要有HTML按钮、文本框、表格等；后端采用Struts2和Spring框架设计，前台的一个请求就是一个action，由对应的类—方法处理并返回结果，Spring框架主要分离层与层之间的耦合性和管理对象（各层之间的依赖注入），在上图中已经明确表示出；持久层表示数据库中的信息，有用户、单词、例句等，为了访问数据库快速、便利，用到了c3p0数据库连接池。整个系统的稳定性和设计都还行。

### 4、物理设计

#### 4.1存储过程

示意图：



存储过程封装了一些必要的访问数据库的逻辑语句，简化了客户端的代码，把更多的逻辑处理放在了服务器端。

【添加用户】

--将用户插入表中

CREATE OR REPLACE PROCEDURE add\_user

(u\_email IN VARCHAR2,u\_name IN VARCHAR2,u\_password IN VARCHAR2,o\_result OUT INTEGER)

IS

BEGIN

INSERT INTO T\_USER VALUES(0,u\_email,u\_name,u\_password);

o\_result :=1;

EXCEPTION

WHEN OTHERS THEN

o\_result :=0;

END add\_user;

【添加单词/例句】

--插入单词/例句

CREATE OR REPLACE PROCEDURE add\_word

(word IN VARCHAR2,mean IN VARCHAR2,diff IN NUMBER)

IS

BEGIN

INSERT INTO T\_CET4 VALUES(0,word,mean,diff,0,1);

END add\_word;

【删除单词/例句】

CREATE OR REPLACE PROCEDURE delete\_word

(word IN VARCHAR2)

IS

BEGIN

DELETE FROM T\_CET4 WHERE FWORD=word;

END delete\_word;

【获取常用单词/例句】

CREATE OR REPLACE PROCEDURE get\_common\_word

(common\_word OUT SYS\_REFCURSOR,o\_result OUT INTEGER)

IS

BEGIN

OPEN common\_word FOR

SELECT FWORD,FMEAN,DIFF FROM T\_CET4

WHERE COMMON=1 AND rownum<=10 ORDER BY FID;

o\_result :=1;

EXCEPTION

WHEN OTHERS THEN

o\_result :=0;

END get\_common\_word;

【获取用户】

CREATE OR REPLACE

PROCEDURE get\_user\_by\_name

(user\_name IN VARCHAR2,user\_result OUT SYS\_REFCURSOR,o\_result OUT INTEGER)

IS

BEGIN

OPEN user\_result FOR

SELECT \* FROM T\_USER

WHERE UNAME=user\_name;

o\_result :=1;

EXCEPTION

WHEN OTHERS THEN

o\_result :=0;

END get\_user\_by\_name;

【分页获取单词】

CREATE OR REPLACE

PROCEDURE get\_word\_by\_page

(page\_num IN NUMBER,word\_result OUT SYS\_REFCURSOR,o\_result OUT INTEGER)

IS

BEGIN

OPEN word\_result FOR SELECT \* FROM (

SELECT A.\*, ROWNUM RN

FROM (SELECT \* FROM T\_CET4 ORDER BY FID) A

)

WHERE RN BETWEEN 1+10\*(page\_num-1) AND 10\*page\_num;

o\_result :=1;

EXCEPTION

WHEN OTHERS THEN

o\_result :=0;

END get\_word\_by\_page;

【随机获取单词】

CREATE OR REPLACE

PROCEDURE get\_word\_random

(word\_result OUT SYS\_REFCURSOR)

IS

BEGIN

OPEN word\_result FOR select \* from

(select \* from T\_TRAN order by dbms\_random.value)

where rownum<=4;

END get\_word\_random;

【获取单词总数】

CREATE OR REPLACE

PROCEDURE get\_word\_total

(result OUT INTEGER)

IS

BEGIN

SELECT COUNT(\*) INTO result FROM T\_CET4;

END get\_word\_total;

【模糊查找单词】

CREATE OR REPLACE

PROCEDURE search\_word

(word IN VARCHAR2,word\_result OUT SYS\_REFCURSOR)

IS

BEGIN

OPEN word\_result FOR SELECT \*

FROM T\_CET4 WHERE FWORD LIKE (word||'%') AND ROWNUM<=10;

END search\_word;

【句子翻译】

CREATE OR REPLACE

PROCEDURE tran\_word

(word IN VARCHAR2,mean OUT VARCHAR2)

IS

BEGIN

SELECT TMEAN INTO mean FROM T\_TRAN WHERE TWORD=word;

END tran\_word;

【更新单词/例句】

CREATE OR REPLACE

PROCEDURE update\_Ct4

(word VARCHAR2,mean VARCHAR2)

IS

BEGIN

UPDATE T\_CET4 SET FMEAN=mean WHERE FWORD=word;

END update\_Ct4;

#### 4.2视图

示意图：



视图为数据库中的表提供了一定的安全机制，可以把想用的信息集中在一个视图上，方便数据的查看。

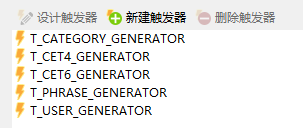
【常用单词/例句视图】

CREATE VIEW COMMON\_WORD(单词,意思)

AS SELECT FWORD,FMEAN FROM T\_CET4 WHERE COMMON=1;

#### 4.3触发器

示意图：



系统中的触发器主要用在表id的自增上，因为Oracle数据库本身没有主键自增的机制，要实现这一功能，必须依赖触发器来完成；为了让系统自动管理主键，定义一个触发器，在插入表数据时，先获取到当前表id的最大值，然后把此值+1，赋给即将插入数据的主键，就间接上实现了表主键id的自动增长。

【用户表触发器】

CREATE OR REPLACE TRIGGER T\_USER\_GENERATOR BEFORE

INSERT ON T\_USER FOR EACH ROW

declare

mid number;

begin

SELECT MAX(USERID) INTO mid FROM T\_USER;

:new.USERID:=mid+1;

end;

【四级表触发器】

CREATE OR REPLACE TRIGGER T\_CET4\_GENERATOR BEFORE

INSERT ON T\_CET4 FOR EACH ROW

declare

mid number;

begin

SELECT MAX(FID) INTO mid FROM T\_CET4;

:new.FID:=mid+1;

end;

【六级表触发器】

CREATE OR REPLACE TRIGGER T\_CET6\_GENERATOR BEFORE

INSERT ON T\_CET6 FOR EACH ROW

declare

mid number;

begin

SELECT MAX(SID) INTO mid FROM T\_CET6;

:new.SID:=mid+1;

end;

【例句表触发器】

CREATE OR REPLACE TRIGGER T\_PHRASE\_GENERATOR BEFORE

INSERT ON T\_PHRASE FOR EACH ROW

declare

mid number;

begin

SELECT MAX(PID) INTO mid FROM T\_PHRASE;

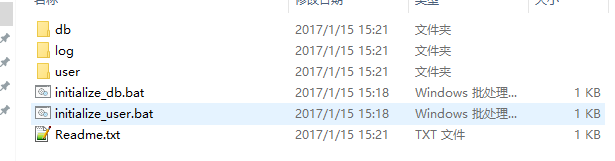
:new.PID:=mid+1;

end;

### 5、系统实施

#### 5.1数据库的初始化

示意图：



运行两个bat文件即可完成数据库的初始化，其中：

1.initialize\_user.bat文件：

（1）用来初始化用户，实质上就是调用sqlplus的命令行，然后登陆到Oracle数据库上，进行一些操作；

（2）因为此操作要新建用户，所以我用system登陆Oracle，密码为：123456；

（3）登陆成功后执行user文件夹下的run\_sql.sql脚本文件，执行这个文件时又会执行createuser.sql文件和grant.sql文件；

（4）此bat文件的运行信息会写在./log/user.log日志文件中。

2.initialize\_db.bat文件：

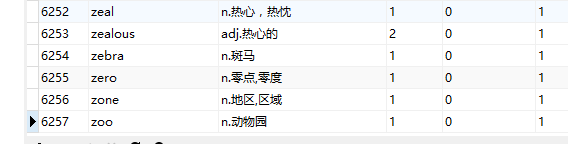
（1）原理同initialize\_user.bat文件，利用新建的用户登录Oracle数据库，执行db文件夹下的run\_sql.sql脚本；

（2）根据脚本内容，会依次执行createTable.sql，createProcedure.sql，createTrigger.sql，createView.sql以达到表，存储过程，触发器，视图的初始化。

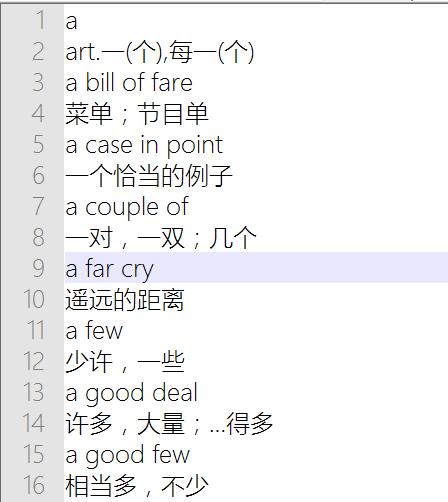
（3）此bat文件的运行信息会写在./log/db.log文件中。

#### 5.2装入数据

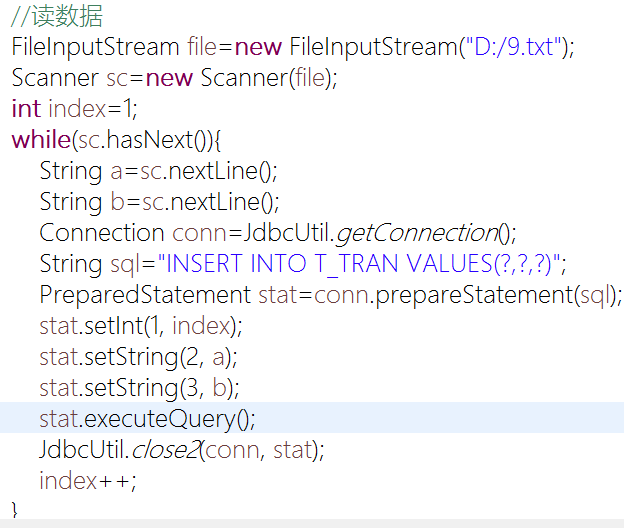
因为本英语学习助手系统的数据量非常大，光单词+例句就有6000+条，如下图所示：



数据量之大，不可能手动导入数据，因此我写了一个java文件用来装入数据，例如单词和例句，有文本文件（网上下载），格式如下：

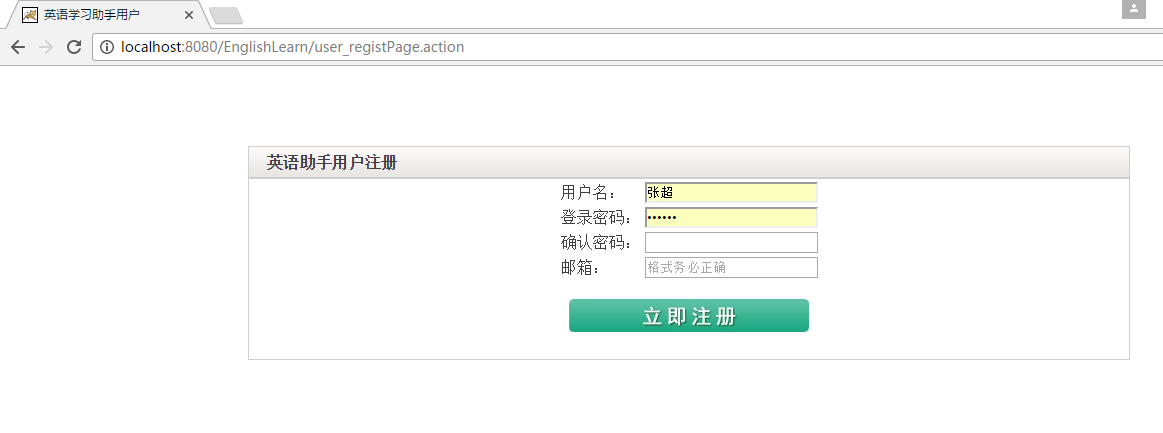


格式为单词/例句--------翻译，一行行的，所以我用Java的文件读入流来完成文件的读入，然后将数据插入到数据库中。如下图所示：



#### 5.3用户注册模块

示意图：



当用户信息填写完成后，点击注册方可注册成功，否则会提示信息未填写完整。

#### 5.4用户登录模块

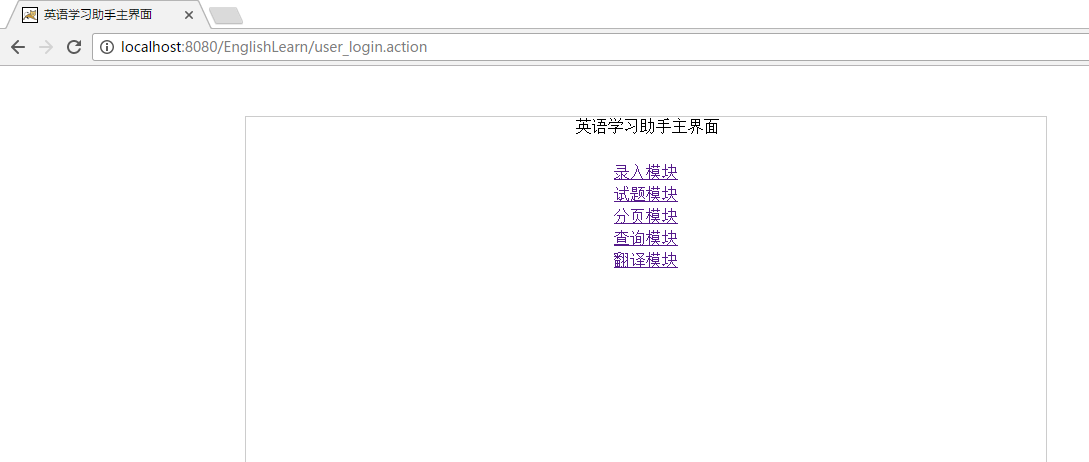
示意图：



在用户已注册的情况下，输入正确的密码即可进入英语助手学习系统。

#### 5.5主菜单

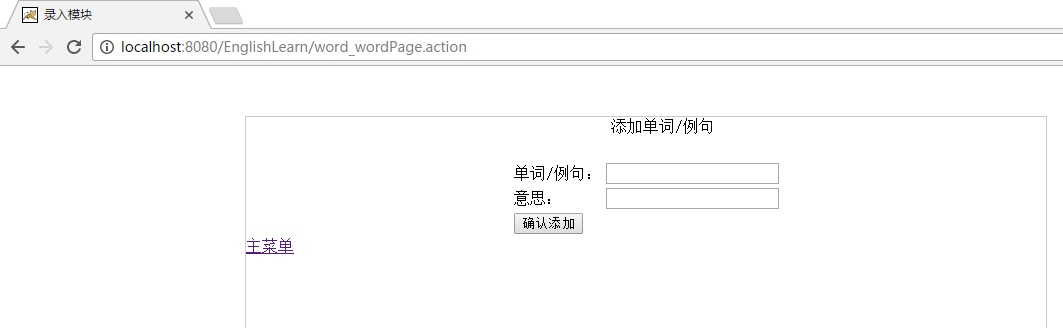
示意图：



用户登录成功后会进入系统主菜单，其中有5个模块可以选择，代表了本系统的功能。

#### 5.6录入模块

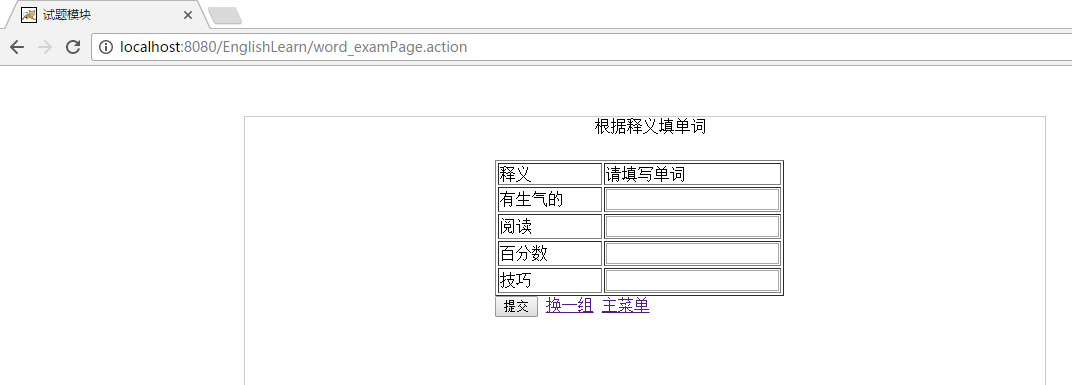
示意图：



用来添加单词和例句（包含基本校验，如用户未填写直接点击添加等）。

#### 5.7试题模块

示意图：



即“根据释义填写单词”，随机从数据库中查询4个单词，只有都填写正确后才能通过；否则会反馈给用户哪一个填写错了。

#### 5.8分页模块

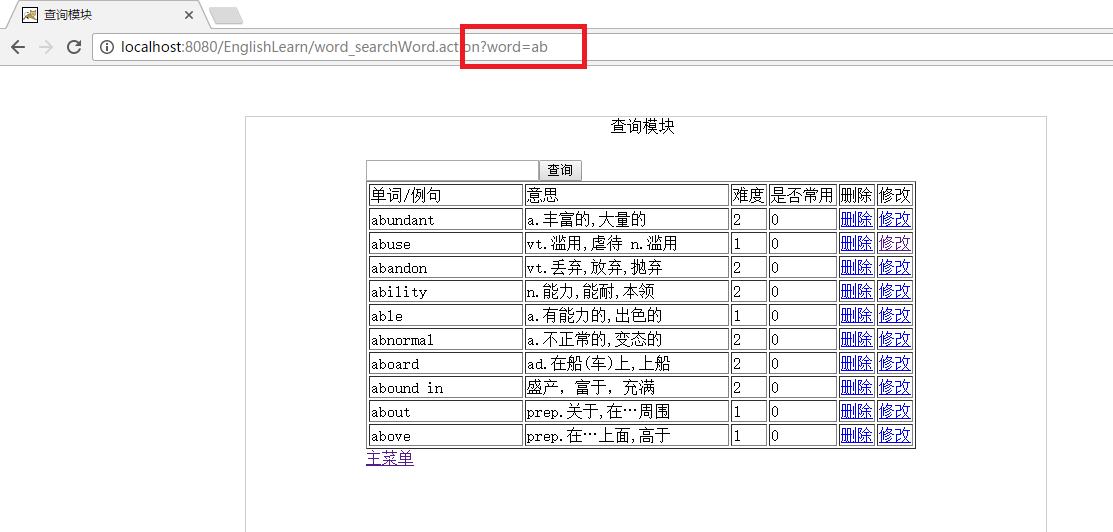
示意图：



用户可以分页浏览单词/例句，也可以对浏览过的单词/例句进行删除或修改操作。

#### 5.9查询模块

示意图：



用户输入单词或例句，点击查询后，模糊查找10条符合的数据；如图查找开头为“ab”的单词和例句。

#### 5.10翻译模块

示意图：

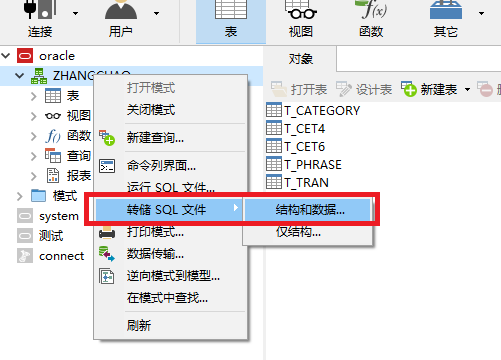


根据输入的句子翻译出对应的结果。

### 6、运行维护

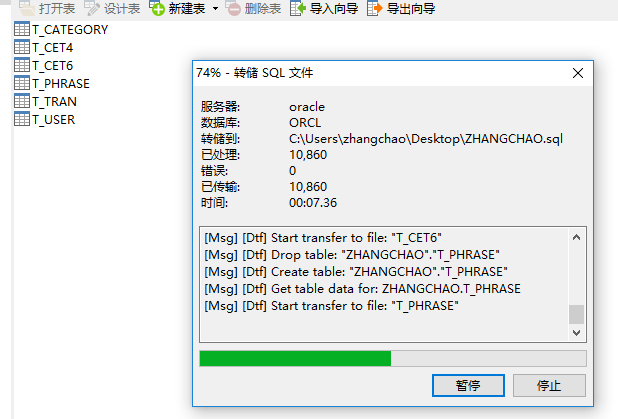
#### 6.1转储数据库

示意图：



使用数据库工具Navicat可将数据库转储成sql文件，这里我将数据库的结构和数据都保存成一个sql文件，这样就能将数据库保存下来。

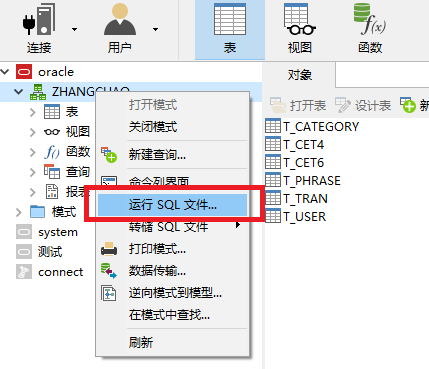
如图正在转储文件：



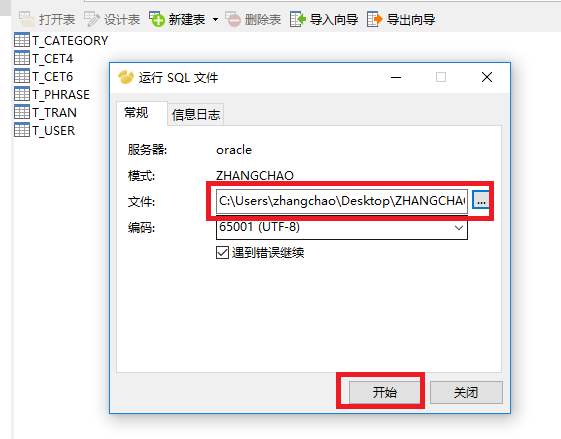
#### 6.2恢复数据库

在5.1数据库初始化的基础上，已经拥有了数据库的结构，恢复数据库可以导入已转储的数据文件，以达到恢复原有数据的目的。

运行sql文件：



执行文件：

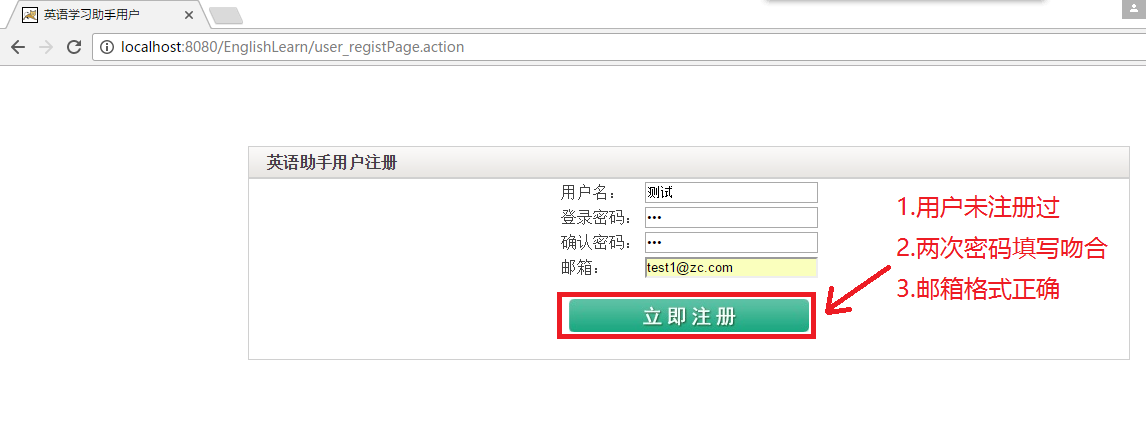


### 7、用户手册

#### 7.1用户注册

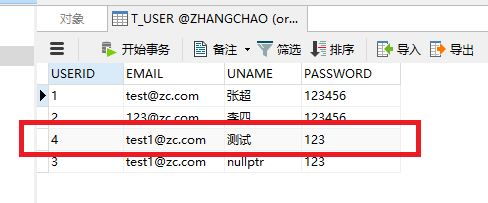
用户必须填写信息完成后，点击注册方可成功将数据录入数据库中：





注册成功后：



数据库中插入一条记录：  


#### 7.2用户登录

根据提示选择用户登录（输入正确的密码）：

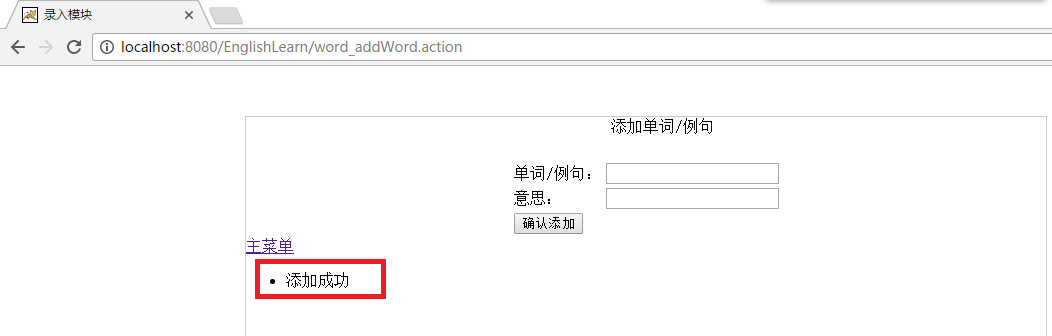


见5.4用户登录模块。

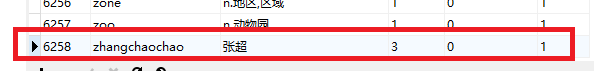
#### 7.3录入模块

添加单词测试：



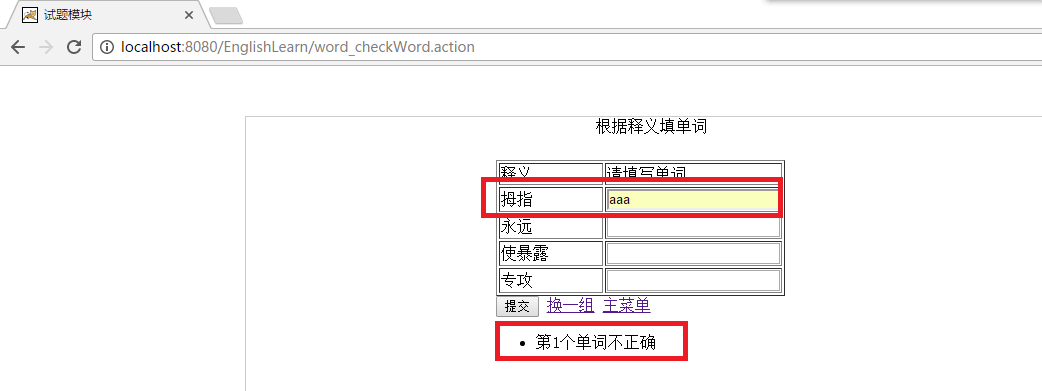


数据库中：

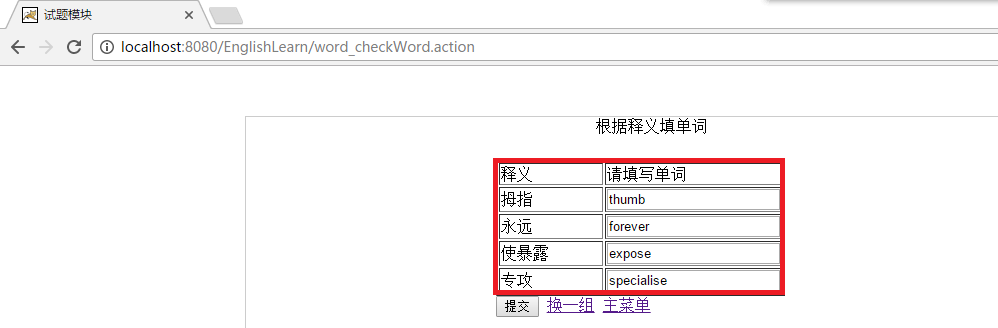


#### 7.4试题模块

按照释义填写单词：



全部填写正确后提交：





#### 7.5分页查询



修改单词/例句：



#### 7.6句子翻译



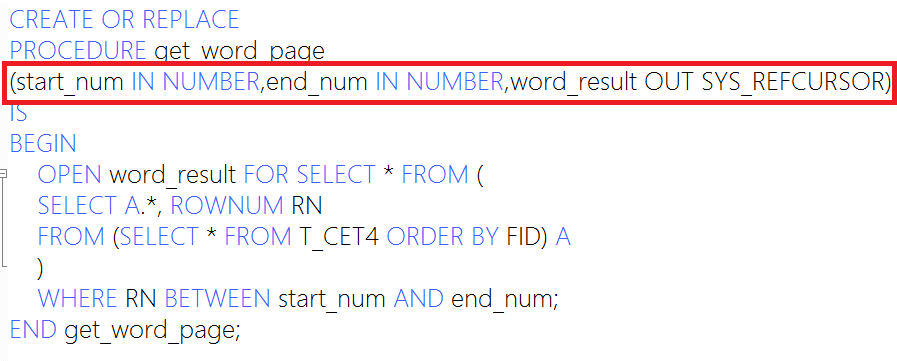


### 8、附录

源程序代码和数据库初始化代码、数据库转储文件会跟电子档一起打包发给老师，这里我主要介绍系统中一些重要的技术。

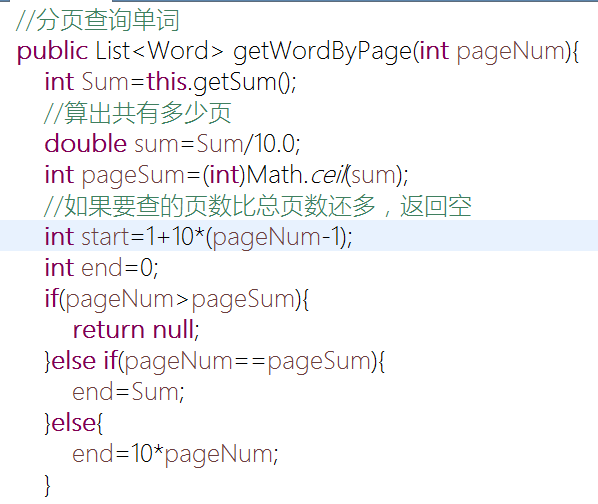
#### 8.1分页技术

最底层，分页查询单词/例句的存储过程：

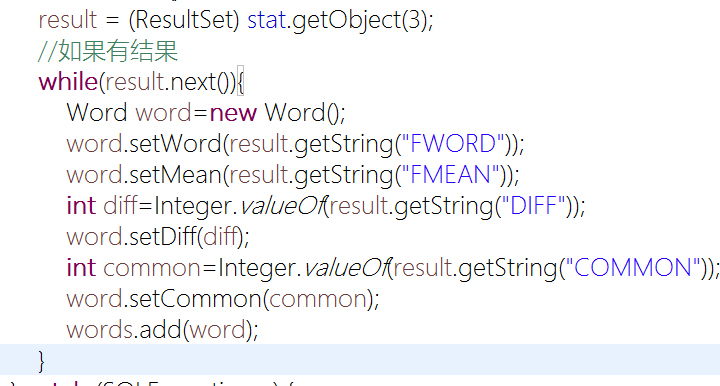


参数：start\_num~end\_num是在数据库中查找的一个范围，返回游标word\_result是返回的结果集。

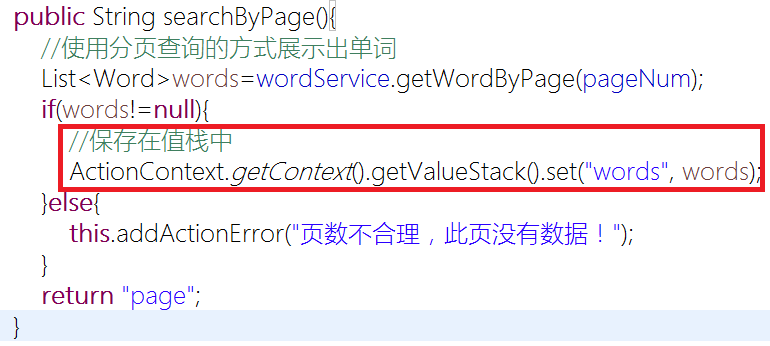
Dao层：



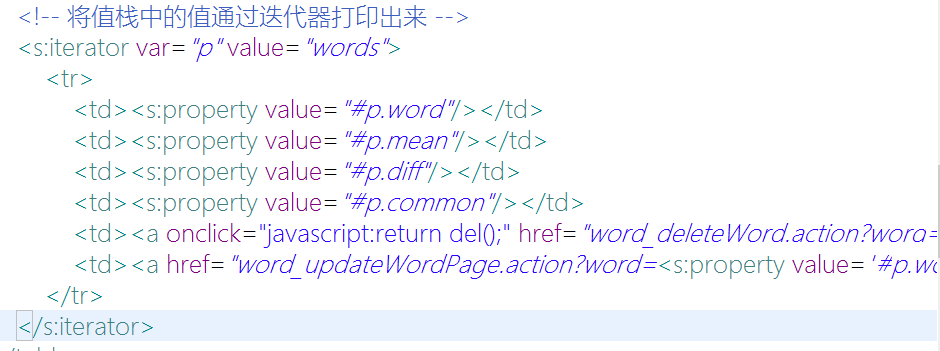
传入一个参数pageNum，即第几页，首先判断这个页数是否合理；然后根据此页数，算出在数据库中查询的范围start~end，调用存储过程时把参数传入，返回游标。



将游标中的数据取出来，转化为结果集；再把结果集的数据转化为List<Word>。



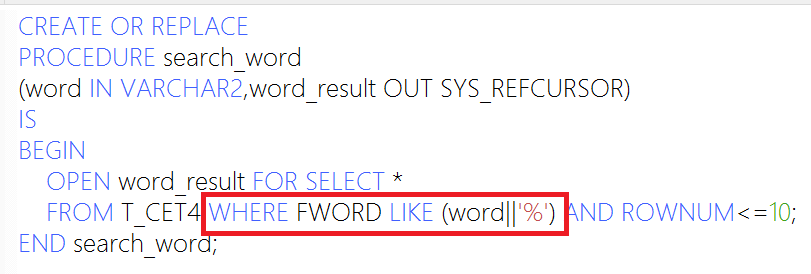
将查出的单词List保存在值栈中。



通过Struts2的标签s:iterator，把单词数据迭代显示在JSP页面中。

#### 8.2模糊查询技术

查询的流程和方法与分页查询一样，在存储过程中用到了通配符“%”，注意在Oracle中用“||”连接字符串：



### 9、总结

本次数据库实习是一次综合性的实习，在web实习的基础上，主要把核心内容放在数据库上，我主要学会了运用存储过程、视图、触发器来解决数据库端的问题；在客户端，学会了能熟练的调用存储过程，并利用游标、结果集来获得自己想要的数据的技能。

在整个系统框架的搭建上，我的能力也有所提高。本次英语学习助手，后端用到了Java开发常用的框架Spring框架和Struts2框架，使开发效率大大提高，主要是使自己的逻辑更加清晰，把各个层的任务分的很开，更利于自己写代码编写，虽然开始时遇到了一些问题，但是还是细心的解决了，后面的完成速度也很快；连接数据库上，用到了c3p0数据库连接池，也是可以提高写代码的效率的，并且在数据库连接上简化了代码。

数据库方面，以前都是基于MySQL开发，没有接触到Oracle这种大型的数据库，有很多代码的语法都不是完全相同的，遇到bug时只能边学边做；印象比较深的就是Oracle数据库在各方面的要求都比较严格，比如语法、权限，还有就是Oracle数据库都是基于大写的，开始时利用Navicat新建用户，登录，发现存在问题，也是由于大小写的问题，后来慢慢的钻研都解决了；数据库方面学的最多的就是存储过程的用法，把一组sql语法封装起来，成一个存储过程，无论是在理解和调用上都是很方便的。

总之本次实习真正的把理论知识运用于实践之中，还是收获颇大的。