package com.ss.test;

import java.util.List;

import com.ss.config.ApplicationDAO;

import com.ss.model.QuestionAns;

import com.ss.model.SkillDetails;

import com.ss.model.UserDetails;

public class SSOTEST {

public static void main(String[] args) {

try {

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*Password validation\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

boolean status=ApplicationDAO.validate("sanjeet", "sanjeet");//validation

System.out.println(status);

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*user creation\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

UserDetails udetails=new UserDetails("sanjeet","sanjeet","ss@gmail.com","90999999","java");

int userRecord=ApplicationDAO.saveUser(udetails);

System.out.println(userRecord);

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*Skill Details List\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

List<SkillDetails> skillDetails=ApplicationDAO.viewSkillDetails();

for (int i=0; i<skillDetails.size(); i++)

System.out.println(skillDetails.get(i).getSkill());

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*Save Question Answer Details\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

QuestionAns questionAns=new QuestionAns("what is java ?","Y","N","N","N", "optOne", "optTwo", "optThree", "optFour", "java");

int questionAnsstatus = ApplicationDAO.saveQuestionAns(questionAns);

System.out.println(questionAnsstatus);

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*view all question answer details\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

List<QuestionAns> viewQuestionAns=ApplicationDAO.viewQuestionAns();

if(viewQuestionAns!=null){

for (int i=0; i<viewQuestionAns.size(); i++){

System.out.println(viewQuestionAns.get(i).getId()+" "+viewQuestionAns.get(i).getQuestion()+"\n"+viewQuestionAns.get(i).getOptOne()+" "+viewQuestionAns.get(i).getOptTwo()+" "+viewQuestionAns.get(i).getOptThree()+" "+viewQuestionAns.get(i).getOptFour());

System.out.print(viewQuestionAns.get(i).getAnsOne()+"\t"+viewQuestionAns.get(i).getAnsTwo()+"\t"+viewQuestionAns.get(i).getAnsThree()+"\t\t"+viewQuestionAns.get(i).getAnsFour());

System.out.println("\nSkill:-"+viewQuestionAns.get(i).getSkill());

System.out.println("\n\n\n\n");

}

}

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*delete records from question answer\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

int deltedRecords=ApplicationDAO.deleteQusAns(21);

System.out.println(deltedRecords);

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*show edit question answer on particular details\*\*\*\*\*\*\*\*\*\*\*\*");

QuestionAns viewQuestionAnss=ApplicationDAO.editQuestionAns(22);

if(viewQuestionAnss!=null){

System.out.println(viewQuestionAnss.getId()+" "+viewQuestionAnss.getQuestion()+"\n"+viewQuestionAnss.getOptOne()+" "+viewQuestionAnss.getOptTwo()+" "+viewQuestionAnss.getOptThree()+" "+viewQuestionAnss.getOptFour());

System.out.print(viewQuestionAnss.getAnsOne()+"\t"+viewQuestionAnss.getAnsTwo()+"\t"+viewQuestionAnss.getAnsThree()+"\t\t"+viewQuestionAnss.getAnsFour());

System.out.println("\nSkill:-"+viewQuestionAnss.getSkill());

System.out.println("\n\n\n\n");

}

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*save updated queswer answer\*\*\*\*\*\*\*\*\*\*\*\*");

QuestionAns updatedQuestionAns=new QuestionAns("what is java ?","Y","Y","Y","Y", "optOne", "optTwo", "optThree", "optFour", "skill");//QuestionAns creation

updatedQuestionAns.setId(22);

updatedQuestionAns.setSkill("Python");

int updateStatus=ApplicationDAO.saveUpdatedQuestionAns(updatedQuestionAns);

System.out.println(updateStatus);

} catch (Exception e) {

e.printStackTrace();

}

}

}

package com.ss.config;

import java.sql.CallableStatement;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.ArrayList;

import java.util.List;

import com.ss.model.QuestionAns;

import com.ss.model.SkillDetails;

import com.ss.model.UserDetails;

public class ApplicationDAO {

public static Connection getCon(){

Connection con=null;

try{

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

con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","system","admin123");

}catch(Exception e){System.out.println(e);}

return con;

}

public static boolean validate(String name,String password){//validating user

boolean status=false;

try{

Connection con=getCon();

PreparedStatement ps=con.prepareStatement("select \* from TestLoginUser where name=? and password=?");

ps.setString(1,name);

ps.setString(2,password);

ResultSet rs=ps.executeQuery();

status=rs.next();

con.close();

}catch(Exception e){System.out.println(e);}

return status;

}

public static int saveUser(UserDetails udetails){//saving user by the given method

int status=0;

try{

Connection con=getCon();

PreparedStatement ps=con.prepareStatement("insert into TestLoginUser(name,password,email,contactno,skill) values(?,?,?,?,?)");

ps.setString(1,udetails.getName());

ps.setString(2,udetails.getPassword());

ps.setString(3,udetails.getEmail());

ps.setString(4,udetails.getContactno());

ps.setString(5,udetails.getSkill());

status=ps.executeUpdate();

con.close();

}catch(Exception e){System.out.println(e);}

return status;

}

public static int saveQuestionAns(QuestionAns questionAns) throws SQLException{//saving question answer by procedure calling

int status=0;

CallableStatement callableStatement = null;

try{

Connection con=getCon();

String saveQuestionAns = "{call TestQuestionAnswer(?,?,?,?,?,?,?,?,?,?)}";

callableStatement = con.prepareCall(saveQuestionAns);

callableStatement.setString(1, questionAns.getQuestion());

callableStatement.setString(2, questionAns.getOptOne());

callableStatement.setString(3, questionAns.getOptTwo());

callableStatement.setString(4, questionAns.getOptThree());

callableStatement.setString(5, questionAns.getOptFour());

callableStatement.setString(6, questionAns.getAnsOne());

callableStatement.setString(7, questionAns.getAnsTwo());

callableStatement.setString(8, questionAns.getAnsThree());

callableStatement.setString(9, questionAns.getAnsFour());

callableStatement.setString(10, questionAns.getSkill());

//callableStatement.registerOutParameter(2, java.sql.Types.VARCHAR);

status=callableStatement.executeUpdate();

con.close();

}catch(Exception e){System.out.println(e);

}

finally {

if (callableStatement != null) {

callableStatement.close();

}}

return status;

}

public static List<QuestionAns> viewQuestionAns(){//view all question answer(join both column)

List<QuestionAns> list=new ArrayList<>();

try{

Connection con=getCon();

PreparedStatement ps=con.prepareStatement("select tques.\*,tans.\* from TestQuestion tques inner join TestAnswer tans on tques.id=tans.unid");

ResultSet rs=ps.executeQuery();

while(rs.next()){

QuestionAns questionAns=new QuestionAns();

questionAns.setId(rs.getInt(1));

questionAns.setQuestion(rs.getString(2));

questionAns.setAnsOne(rs.getString("A"));

questionAns.setAnsTwo(rs.getString("B"));

questionAns.setAnsThree(rs.getString("C"));

questionAns.setAnsFour(rs.getString("D"));

questionAns.setOptOne(rs.getString("OPTONE"));

questionAns.setOptTwo(rs.getString("OPTTWO"));

questionAns.setOptThree(rs.getString("OPTTHREE"));

questionAns.setOptFour(rs.getString("OPTFOUR"));

questionAns.setSkill(rs.getString("SKILL"));

list.add(questionAns);

}

con.close();

}catch(Exception e){System.out.println(e);}

return list;

}

public static int deleteQusAns(int unqid){//delete question answer

int status=0;

try{

Connection con=getCon();

String sql = "{ ? = call TestQuestionAnsDel(?) }";//procedure function

CallableStatement cs = con.prepareCall(sql);

cs.setInt(2,unqid);

cs.registerOutParameter(1, java.sql.Types.INTEGER);

cs.execute();

status= cs.getInt(1);

con.close();

}catch(Exception e){System.out.println(e);}

return status;

}

public static QuestionAns editQuestionAns(int unqid){//Edit question answer(join both column) and show values on screen

QuestionAns quesAns=new QuestionAns();

try{

Connection con=getCon();

PreparedStatement ps=con.prepareStatement("select tques.\*,tans.\* from TestQuestion tques inner join TestAnswer tans on tques.id=tans.unid and tques.id="+unqid+"");

ResultSet rs=ps.executeQuery();

while(rs.next()){

//QuestionAns questionAns=new QuestionAns();

quesAns.setId(rs.getInt(1));

quesAns.setQuestion(rs.getString(2));

quesAns.setAnsOne(rs.getString("A"));

quesAns.setAnsTwo(rs.getString("B"));

quesAns.setAnsThree(rs.getString("C"));

quesAns.setAnsFour(rs.getString("D"));

quesAns.setOptOne(rs.getString("OPTONE"));

quesAns.setOptTwo(rs.getString("OPTTWO"));

quesAns.setOptThree(rs.getString("OPTTHREE"));

quesAns.setOptFour(rs.getString("OPTFOUR"));

quesAns.setSkill(rs.getString("SKILL"));

}

con.close();

}catch(Exception e){System.out.println(e);}

return quesAns;

}

//with the help of question id update recent values in question ans table

public static int saveUpdatedQuestionAns(QuestionAns questionAns){

int status=0;

CallableStatement callableStatement = null;

try{

Connection con=getCon();

//procedure calling

String saveQuestionAns = "{call TestQuestionAnswerUpdate(?,?,?,?,?,?,?,?,?,?,?)}";

callableStatement = con.prepareCall(saveQuestionAns);

callableStatement.setInt(1, questionAns.getId());

callableStatement.setString(2, questionAns.getQuestion());

callableStatement.setString(3, questionAns.getOptOne());

callableStatement.setString(4, questionAns.getOptTwo());

callableStatement.setString(5, questionAns.getOptThree());

callableStatement.setString(6, questionAns.getOptFour());

callableStatement.setString(7, questionAns.getAnsOne());

callableStatement.setString(8, questionAns.getAnsTwo());

callableStatement.setString(9, questionAns.getAnsThree());

callableStatement.setString(10, questionAns.getAnsFour());

callableStatement.setString(11, questionAns.getSkill());

//callableStatement.registerOutParameter(2, java.sql.Types.VARCHAR);

status=callableStatement.executeUpdate();

con.close();

}catch(Exception e){System.out.println(e);}

return status;

}

public static List<SkillDetails> viewSkillDetails(){//view all skill in drop down

List<SkillDetails> list=new ArrayList<>();

try{

Connection con=getCon();

PreparedStatement ps=con.prepareStatement("select \* from TestSkill");

ResultSet rs=ps.executeQuery();

while(rs.next()){

SkillDetails skillDetails=new SkillDetails();

skillDetails.setId(rs.getInt(1));

skillDetails.setSkill(rs.getString(2));

skillDetails.setDt(rs.getDate(3));

list.add(skillDetails);

}

con.close();

}catch(Exception e){System.out.println(e);}

return list;

}

}

**package** com.ss.model;

**public** **class** QuestionAns {

**private** **int** id;

**private** String question,

ansOne,ansTwo,ansThree,

ansFour,optOne,optTwo,

optThree,optFour,skill;

**public** QuestionAns() {}

**public** QuestionAns(String question,String ansOne, String ansTwo, String ansThree,

String ansFour, String optOne, String optTwo, String optThree,

String optFour,String skill) {

**super**();

**this**.question=question;

**this**.ansOne=ansOne;

**this**.ansTwo=ansTwo;

**this**.ansThree=ansThree;

**this**.ansFour=ansFour;

**this**.optOne=optOne;

**this**.optTwo=optTwo;

**this**.optThree=optThree;

**this**.optFour=optFour;

**this**.skill=skill;

}

**public** **int** getId() {

**return** id;

}

**public** **void** setId(**int** id) {

**this**.id = id;

}

**public** String getQuestion() {

**return** question;

}

**public** **void** setQuestion(String question) {

**this**.question = question;

}

**public** String getAnsOne() {

**return** ansOne;

}

**public** **void** setAnsOne(String ansOne) {

**this**.ansOne = ansOne;

}

**public** String getAnsTwo() {

**return** ansTwo;

}

**public** **void** setAnsTwo(String ansTwo) {

**this**.ansTwo = ansTwo;

}

**public** String getAnsThree() {

**return** ansThree;

}

**public** **void** setAnsThree(String ansThree) {

**this**.ansThree = ansThree;

}

**public** String getAnsFour() {

**return** ansFour;

}

**public** **void** setAnsFour(String ansFour) {

**this**.ansFour = ansFour;

}

**public** String getOptOne() {

**return** optOne;

}

**public** **void** setOptOne(String optOne) {

**this**.optOne = optOne;

}

**public** String getOptTwo() {

**return** optTwo;

}

**public** **void** setOptTwo(String optTwo) {

**this**.optTwo = optTwo;

}

**public** String getOptThree() {

**return** optThree;

}

**public** **void** setOptThree(String optThree) {

**this**.optThree = optThree;

}

**public** String getOptFour() {

**return** optFour;

}

**public** **void** setOptFour(String optFour) {

**this**.optFour = optFour;

}

**public** String getSkill() {

**return** skill;

}

**public** **void** setSkill(String skill) {

**this**.skill = skill;

}

}

**package** com.ss.model;

**import** java.util.Date;

**public** **class** SkillDetails {

**private** **int** id;

**private** String skill;

**private** Date dt;

**public** SkillDetails() {}

**public** SkillDetails(String skill) {

**super**();

**this**.skill=skill;

}

**public** String getSkill() {

**return** skill;

}

**public** **void** setSkill(String skill) {

**this**.skill = skill;

}

**public** **int** getId() {

**return** id;

}

**public** **void** setId(**int** id) {

**this**.id = id;

}

**public** Date getDt() {

**return** dt;

}

**public** **void** setDt(Date dt) {

**this**.dt = dt;

}

}

**package** com.ss.model;

**public** **class** UserDetails {

**private** **int** id;

**private** String name,password,email,

contactno,skill;

**public** UserDetails() {}

**public** UserDetails(String name, String password, String email, String contactno,String skill) {

**super**();

**this**.name = name;

**this**.password = password;

**this**.email = email;

**this**.contactno = contactno;

**this**.skill=skill;

}

**public** **int** getId() {

**return** id;

}

**public** **void** setId(**int** id) {

**this**.id = id;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** String getPassword() {

**return** password;

}

**public** **void** setPassword(String password) {

**this**.password = password;

}

**public** String getEmail() {

**return** email;

}

**public** **void** setEmail(String email) {

**this**.email = email;

}

**public** String getContactno() {

**return** contactno;

}

**public** **void** setContactno(String contactno) {

**this**.contactno = contactno;

}

**public** String getSkill() {

**return** skill;

}

**public** **void** setSkill(String skill) {

**this**.skill = skill;

}

}

**CREATE** **TABLE** TestLoginUser (

id **int** **NOT** **NULL**,

name **varchar**(100) **NULL**,

password **varchar**(100) **NULL**,

email **varchar**(100) **NULL**,

contactno **varchar**(20) **NULL**,

skill **varchar**(20) **NULL**,

dt **date** **default** sysdate

);

--drop table TestLoginUser

**ALTER** **TABLE** TestLoginUser **ADD** (

**CONSTRAINT** id\_pk **PRIMARY** **KEY** (id));

--drop sequence TestLoginUser\_seq

**CREATE** SEQUENCE TestLoginUser\_seq START **WITH** 1;

--drop trigger TestLoginUser\_on\_insert

**CREATE** **OR** REPLACE TRIGGER TestLoginUserr\_on\_insert

BEFORE **INSERT** **OR** **UPDATE** **ON** TestLoginUser

FOR EACH ROW

**BEGIN**

**SELECT** TestLoginUser\_seq.nextval

**INTO** :new.id

**FROM** dual;

**END**;

**INSERT** **INTO** TestLoginUser (name, password, email, contactno,skill) **VALUES**

('prateek', 'prateek', 'prateek@gmail.com', '9199291212','java');

**INSERT** **INTO** TestLoginUser (name, password, email, contactno,skill) **VALUES**

('sanjeet', 'sanjeet', 'roshan@gmail.com', '9953030303','java');

**select** \* **from** TestLoginUser

**create** **TABLE** TestSkill(

id **int** **NOT** **NULL**,

skill **varchar**(100) **NULL**,

dt **date** **default** sysdate

);

**ALTER** **TABLE** TestSkill **ADD** (

**CONSTRAINT** skillid\_pk **PRIMARY** **KEY** (id));

**CREATE** SEQUENCE TestSkill\_seq START **WITH** 1;

**CREATE** **OR** REPLACE TRIGGER TestSkill\_on\_insert

BEFORE **INSERT** **ON** TestSkill

FOR EACH ROW

**BEGIN**

**SELECT** TestSkill\_seq.nextval

**INTO** :new.id

**FROM** dual;

**END**;

**INSERT** **INTO** TestSkill(skill) **VALUES**

('core java');

**INSERT** **INTO** TestSkill (skill) **VALUES**

('Adv java');

**INSERT** **INTO** TestSkill (skill) **VALUES**

('PlSql');

**INSERT** **INTO** TestSkill (skill) **VALUES**

('c#');

**INSERT** **INTO** TestSkill (skill) **VALUES**

('Dot Net');

**select** \* **from** TestSkill;

--------------------------------------------------------------------------------------------------------

**CREATE** **TABLE** TestQuestion (

id **int** **NOT** **NULL**,

question **varchar**(4000) **NULL**,

skill varchar2(45) **null**,

dt **date** **default** sysdate

);

**ALTER** **TABLE** TestQuestion **ADD** (

**CONSTRAINT** tqueid\_pk **PRIMARY** **KEY** (id));

**CREATE** SEQUENCE Question\_seq START **WITH** 1;

**CREATE** **OR** REPLACE TRIGGER Question\_on\_insert

BEFORE **INSERT** **ON** TestQuestion

FOR EACH ROW

**BEGIN**

**SELECT** Question\_seq.nextval

**INTO** :new.id

**FROM** dual;

**END**;

------------------------------------

**CREATE** **TABLE** TestAnswer (

id **int** **NOT** **NULL**,

unid **int** **unique**,

optOne **varchar**(400),

ansOne **varchar**(1),

optTwo **varchar**(400),

ansTwo **varchar**(1),

optThree **varchar**(400),

ansThree **varchar**(1),

optFour **varchar**(400),

ansFour **varchar**(1),

dt **date** **default** sysdate

);

**ALTER** **TABLE** TestAnswer **ADD** (

**CONSTRAINT** AnsId1\_pk **PRIMARY** **KEY** (id));

**CREATE** SEQUENCE TAnswer\_seq START **WITH** 1;

**CREATE** **OR** REPLACE TRIGGER TAnswer\_on\_insert

BEFORE **INSERT** **ON** TestAnswer

FOR EACH ROW

**BEGIN**

**SELECT** TAnswer\_seq.nextval

**INTO** :new.id

**FROM** dual;

**END**;

------------------------------------------------

**create** **or** replace

**PROCEDURE** TestQuestionAnswer(

valquestion **IN** SYSTEM.TESTQUESTION.QUESTION%**TYPE**,

optOneVal **IN** SYSTEM.TESTANSWER.OPTONE%**TYPE**,

optTwoVal **IN** SYSTEM.TESTANSWER.OPTTWO%**TYPE**,

optThreeVal **IN** SYSTEM.TESTANSWER.OPTTHREE%**TYPE**,

optFourVal **IN** SYSTEM.TESTANSWER.OPTFOUR%**TYPE**,

ansOneVal **IN** SYSTEM.TESTANSWER.A%**TYPE**,

ansTwoVal **IN** SYSTEM.TESTANSWER.B%**TYPE**,

ansThreeVal **IN** SYSTEM.TESTANSWER.C%**TYPE**,

ansFourVal **IN** SYSTEM.TESTANSWER.D%**TYPE**,

skillVal **IN** SYSTEM.TESTQUESTION.skill%**TYPE**

)

**as**

quesId **int**;

**BEGIN**

**insert** **into** TestQuestion(question,skill)**values**(Valquestion,skillVal);

**select** id **into** quesId **from**( **select** id **from** TestQuestion **order** **by** id **desc** )Where ROWNUM <= 1;

**insert** **into** TestAnswer(unid,optOne,a,optTwo,b,

optThree,c,optFour,d

)

**values**

(

quesId,optOneVal,ansOneVal,optTwoVal,ansTwoVal,

optThreeVal,ansThreeVal,optFourVal,ansFourVal

);

-- message := 'Success';

**END** ;

---------------------------------------------------------

**create** **or** replace

FUNCTION TestQuestionAnsDel(unqid **IN** **int**)

RETURN number **IS**

qusAnsId **int** := unqid;

**BEGIN**

**delete** **from** TestQuestion **where** id=unqid;

**delete** **from** TestAnswer **where** unid=unqid;

RETURN qusAnsId;

EXCEPTION

WHEN OTHERS **THEN**

raise\_application\_error(-20001,'An error was encountered - '||SQLCODE||' -ERROR- '||SQLERRM);

**END**;

---------------------------------------------------------

**create** **or** replace

**PROCEDURE** TestQuestionAnswerUpdate(

uniqID **IN** SYSTEM.TESTQUESTION.ID%**TYPE**,

valquestion **IN** SYSTEM.TESTQUESTION.QUESTION%**TYPE**,

optOneVal **IN** SYSTEM.TESTANSWER.OPTONE%**TYPE**,

optTwoVal **IN** SYSTEM.TESTANSWER.OPTTWO%**TYPE**,

optThreeVal **IN** SYSTEM.TESTANSWER.OPTTHREE%**TYPE**,

optFourVal **IN** SYSTEM.TESTANSWER.OPTFOUR%**TYPE**,

ansOneVal **IN** SYSTEM.TESTANSWER.A%**TYPE**,

ansTwoVal **IN** SYSTEM.TESTANSWER.B%**TYPE**,

ansThreeVal **IN** SYSTEM.TESTANSWER.C%**TYPE**,

ansFourVal **IN** SYSTEM.TESTANSWER.D%**TYPE**,

skillVal **IN** SYSTEM.TESTQUESTION.skill%**TYPE**

)

**as**

**BEGIN**

**update** TestQuestion **set**

question=Valquestion,

skill=skillVal **where** id=uniqID;

**update** TestAnswer **set**

unid=uniqID,

optOne=optOneVal,

a=ansOneVal,

optTwo=optTwoVal,

b=ansTwoVal,

optThree=optThreeVal,

c=ansThreeVal,

optFour=optFourVal,

d=ansFourVal **where** unid= uniqID;

**commit**;

**END**;