Quiz System Report

Independent Study

Qi Chen - August 11, 2016

Introduction 2

The design of this system 2

- Back-end 2
- Servlet 8
- Front-end 9
- Relationship among back-end, servlet and front-end 9
- data flow 10
- Database 11

Introduction

The Quiz System is used for students to test their java/python ability which combine the QuizJet and QuizPet system.
The improvement of this kind of system compared with old QuizJet and QuizPet system are as followed.

1. combined this two old system into one system, making it easy to maintain.
2. restructured the code in both system and remove the java code in the JSP page as much as possible, making the code more readable and to revise
3. better the interface of the system. for example, the keywords in the test code which is provided for the users is highlighted, making it more readable and comfortable.

The new system is designed and implemented with MVC structure, lowering the coupling, and the following context will discuss how this system works.

The design of this system

Back-end

There are three packages: bean, dao, Util, which are used to get the data from database and transfer them into the context which is needed in the front-end.

1. For bean package
This package is designed according to the context in the database and used to encapsulate the data module.

1.1 Jclasses.java

```
package bean;
public class Jclasses {
	private int classId;
	private String ClassName;
	private String ClassTester;
	private int QuizID;
	public int getClassId()
	public void setClassId(int classId)
	public String getClassName()
	public void setClassName(String className)
	public String getClassTester()
	public void setClassTester(String classTester)
	public int getQuizID()
	public void setQuizID(int quizID)
}
```

In this class, the information of this class given by quizID and be passed to the font-end completely.

1.2. Jquestion.java

```
package bean;
public class Jquestion {
	private int questionID;
	private String title;
	public int getQuestionID()
	public void setQuestionID(int questionID)
	public String getTitle()
	public void setTitle(String title)
}
```

In this class, the information of all quiz's id and title can be passed to the font-end completely, which will be used to the navigation menu.

1.3. Jquiz.java

```
package bean;
import java.sql.Blob;
public class Jquiz {
      private int quizeID;
      private String Title;
      private Blob code;
      private String rdfID;
      private int AnsType;
      private int minVar;
      private int maxVar;
      private int privacy;
      private int quesType;
      private int start;
      private int end;
      public int getQuizeID()
      public void setQuizeID(int quizeID)
      public String getTitle()
      public void setTitle(String title)
      public Blob getCode()
      public void setCode(Blob code)
      public int getMinVar()
      public void setMinVar(int minVar)
      public int getMaxVar()
      public void setMaxVar(int maxVar)
      public int getPrivacy()
      public void setPrivacy(int privacy)
      public int getQuesType()
      public void setQuesType(int quesType)
      public String getRdfID()
      public void setRdfID(String rdfID)
      public int getAnsType()
      public int getStart()
      public void setStart(int start)
      public int getEnd()
      public void setEnd(int end)
      public void setAnsType(int ansType)
}
```

This class is mainly used to pass the detailed information of Quiz, including its id, code and so on from the database to the front-end.

2 For dao package

This package is used to get the information we need from the database and pass them to the servlet with the format which is defined in the bean package

2.1 JclassesDAO.java

```
package dao;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;
import Util.DBConnect;
import bean.Jclasses;
public class JclassesDAO {
      private DBConnect dbc;
      private ResultSet rs;
      public List<Jclasses> getAllClasses(int quizId)
}
```

There is only one function in this class, which is used to get all classes in the quiz.
For example, in the java test system(old QuizJet system), some quiz need more than one classes, like BankAccount,class or CashRegister.class, whose id should be found with this JclassesDAO.class.

2.2 JquestionDAO.java

```
package dao;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;
import Util.DBConnect;
import bean.Jquestion;
public class JquestionDAO {
      private DBConnect dbc;
      private ResultSet rs;
      public List<Jquestion> getListQuestion(String language) throws
SQLException
}
```

This class is to list all the quiz questions given the the language the system tests. For example, if this system is used to test java, the getListQuestion function will return all the java questions.if this system is used to test python, the getListQuestion function will return all the python questions.

2.3 JquizDAO.java

```
package dao;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;
import Util.DBConnect;
import bean.Jquiz;
public class JquizDAO {
      private DBConnect dbc;
      private ResultSet rs;
      public List<Jquiz> getListQuizebyQuestionID(int id ) throws
SQLException{}
      public Jquiz getJquiz(String rdfID) throws SQLException{}
      public Jquiz getJquizByQuizId(int quizID) throws SQLException{}
}
```

This class is used to get the complete information of quiz.

- `getListQuizebyQuestionID()` function can get all quiz information given a question id
- `getJquiz()` function can get the quiz information given a rdfID
- `getJquizByQuizId()` function can get the quiz information given a quizID

3. For Util package

This package is used to encapsulate the functions which are frequently used in the dao package

3.1 DBConnect.java

This class is to connect database and get data from it.

3.2 Tool.java

```
package Util;
import java.util.ArrayList;
import java.util.List;
public class Tool {

      public static List<Integer> getAllSemicolon(String code)
      public static int calcualteNumberofSpace(String string)
}
```

getAllSemicolon() function is used to return all the indexes of new line give a code, which is used to blanket a line for testers to fill in.

calcualteNumberofSpace() function is used to calculate how many spaces before a given string, which is good to make the code needing filling in centered.

3.3 Coding.java

```
package Util;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStreamReader;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;
import java.util.Random;
import org.apache.commons.io.IOUtils;
import bean.Jquiz;
public class Coding {
      public static String getCode(Jquiz jquiz)
      public static String code2Exec(String code)
      public static String code2fontend(String codeStr)
      public static String[] spilitCode(String code)
      public static String getCmd2compile(boolean window, String
compilerHomeDir, String curActionDir,String testerSurfix, String
language, boolean multiClassFlag)
      public static String getCmd2execute(boolean window, String
compilerHomeDir, String curActionDir,String testerSurfix, String
language, boolean multiClassFlag)
      public static String runCode(String compile, String exec, boolean
multiClassFlag, String curActionDir,String language, String testerSurfix,
Integer queType, int size, ArrayList<String> fileNames,String
extraClassDir) throws IOException, InterruptedException
}
```

This class is mainly used to deal with something relevant to coding.

- getCode(Jquiz jquiz) function is used to get the code part give a quiz
- getCode(Jquiz jquiz) function is used to change the format of code to make it suitable to run, for example, changing the “&lt” to “<”.
- getCode(Jquiz jquiz) function is beds to change the format of the code to make it suitable to display on front-end interface. For example, changing the “<” to “&lt”
- getCmd2compile function is used to return the command string how to compile the code
- getCmd2execute function is used to return the command string how to execute the code
- runCode function is used to run the code and return the answer.

3.4 FileOperation.java

```
package Util;
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
import java.sql.SQLException;
import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.Random;
import javax.servlet.ServletOutputStream;
import org.apache.tomcat.util.http.fileupload.FileUtils;
public class FileOperation {
      public static String replace = "___";
      public static char replaceChar = replace.charAt(0);
      public static String createActionFilePath(String appDir, String
compilerHomeDir, String usr, boolean window)
      public static String getExtraClassDir4Multiclasses(String appDir,
boolean window)
      public static String createTempFile(String curActionDir, boolean
multiClassFlag, int queType, String codePart,String language,
ServletOutputStream out)
      public static void deleteTempFiles(String tempFileName, String
curActionDir, String surfix)
      public static String stringTransfer(String string)
      public static void log(String str)
}
```

This class is used to do something relevant to operate file.

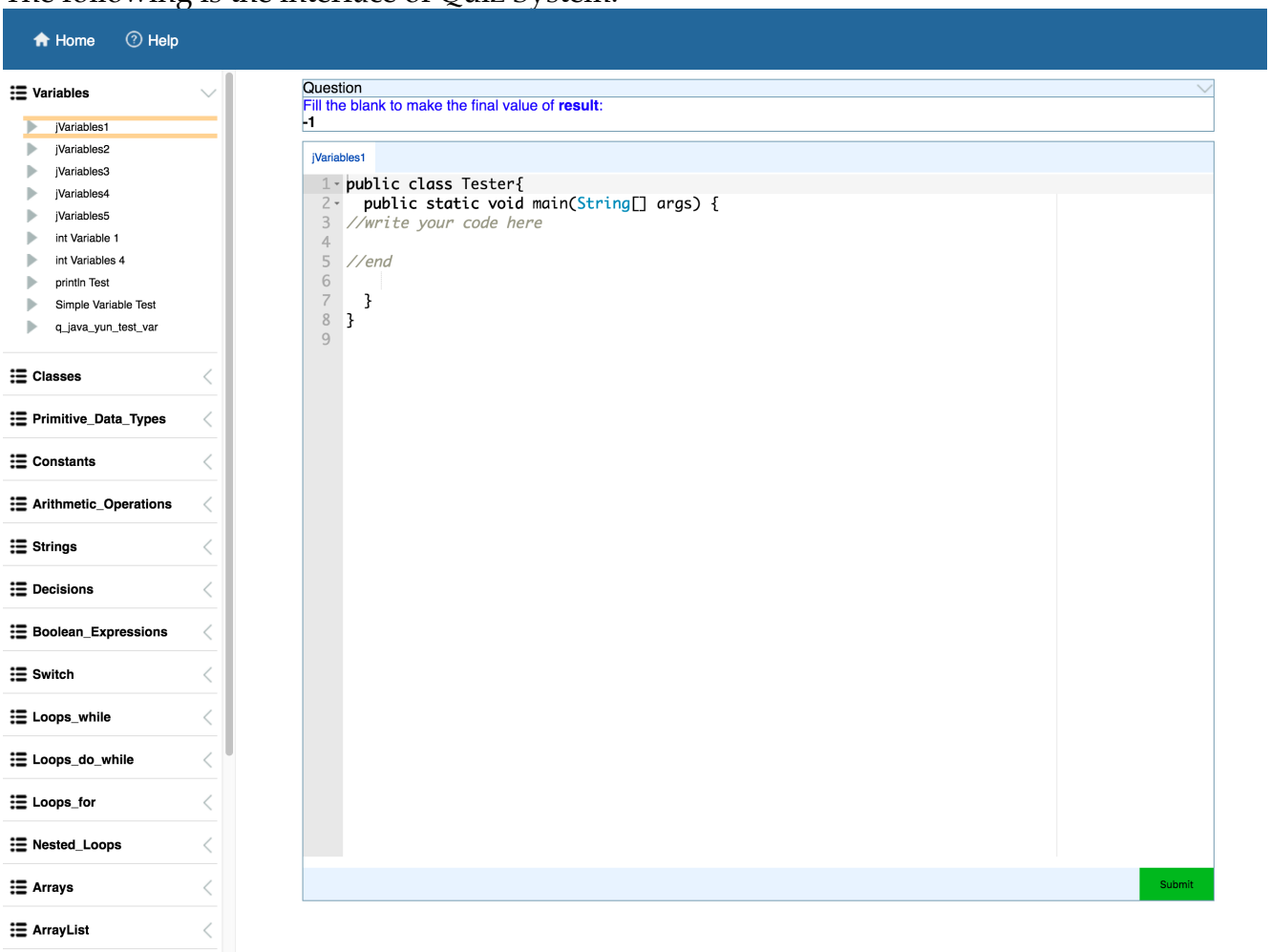
- createActionFilePath function is used to return a file path string to create a temporary java file
- getExtraClassDir4Multiclasses is used to return the file paths strings to create other java files for extra classes
- createTempFile is used to create a temporary java file
- deleteTempFiles is used to delete the temporary java file which is created by previous function
- stringTransfer function is used to make the answer which the code runs more nice looking
- log function is used to report some errors or records

Servlet

This middle layer(servlets) is used to get the data from the back-end and then pass them to the front-end.
it includes three files: MenuNavigation.java, displayQuiz.java and EvaluationAnswer.java

Front-end

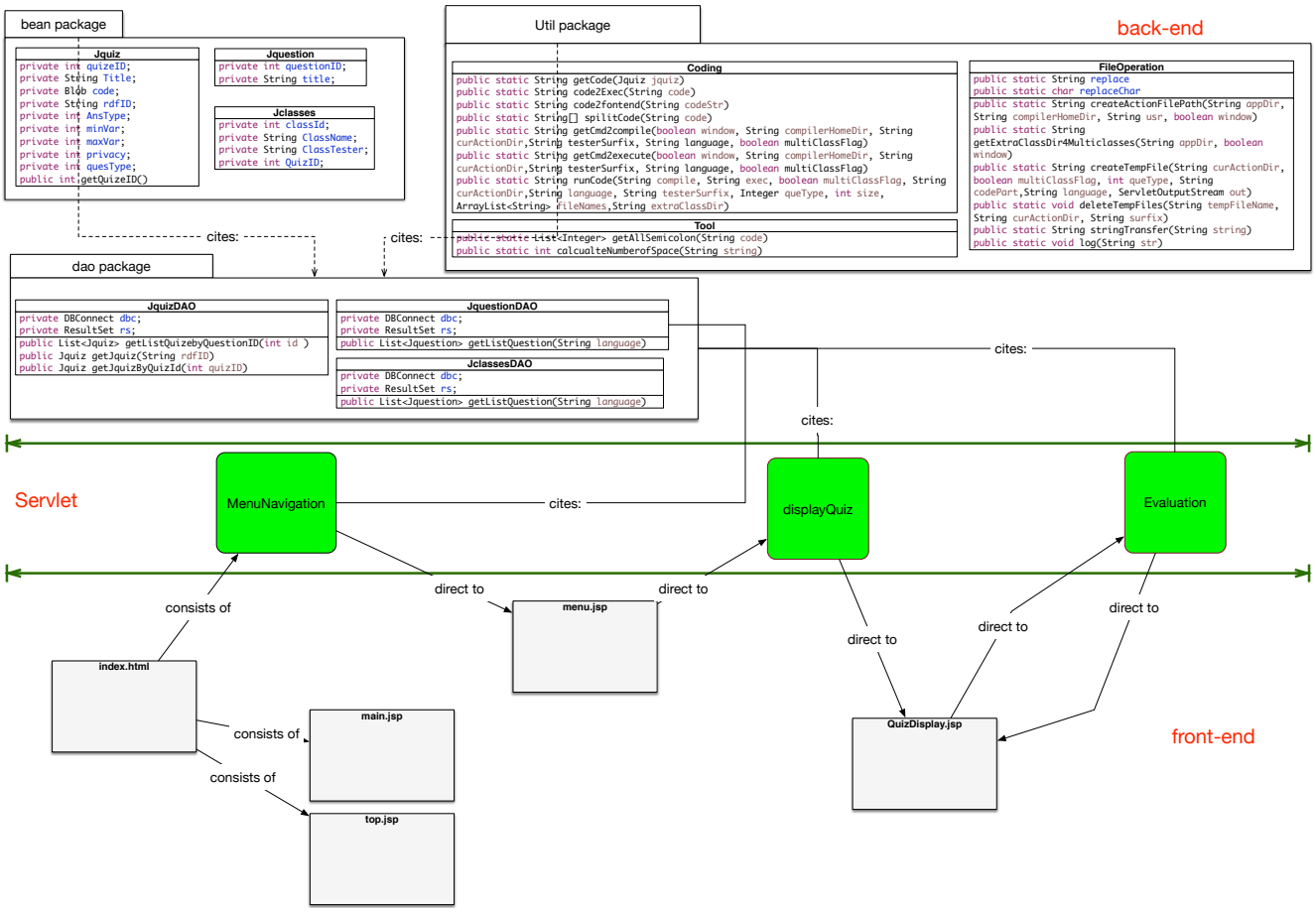
The following is the interface of Quiz System.



The font-end consists of 5 files: index.html, main.html , menu.jsp and QuizDisplay2.jsp

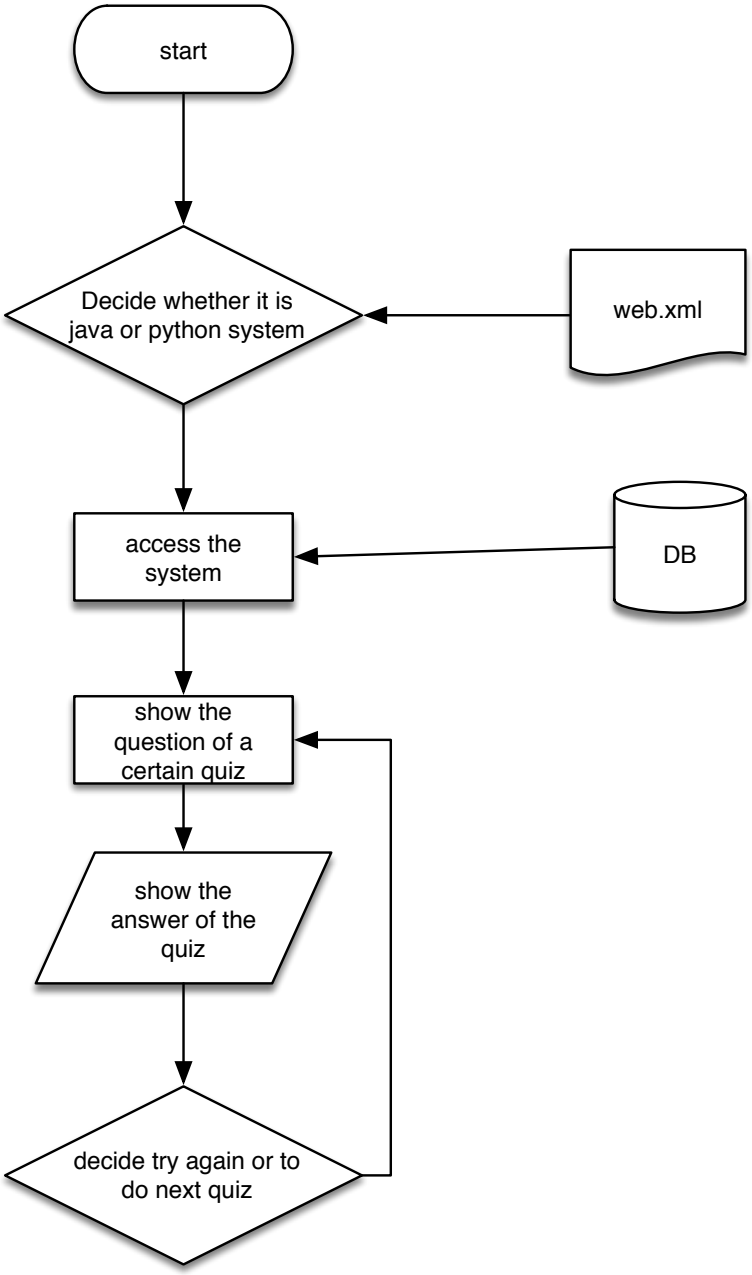
Relationship among back-end, servlet and front-end

The relationship can be seen from the following graph.



data flow

The data flow of this system is as followed.


start
Decide whether it is java or python system
web.xml
access the system
DB
show the question of a certain quiz
show the answer of the quiz
decide try again or to do next quiz

Database

The table which is used in database are as followed:

1. ent_class

Name: ent_class

Column	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	G	Default / Expression
ClassID	INT(10)	☑	☑	☐	☐	☑	☐	☐	☐	'0'
ClassName	VARCHAR(45)	☐	☑	☐	☐	☐	☐	☐	☐	
ClassTester	VARCHAR(45)	☐	☐	☐	☐	☐	☐	☐	☐	NULL

2. rel_quiz_class

Name: rel_quiz_class

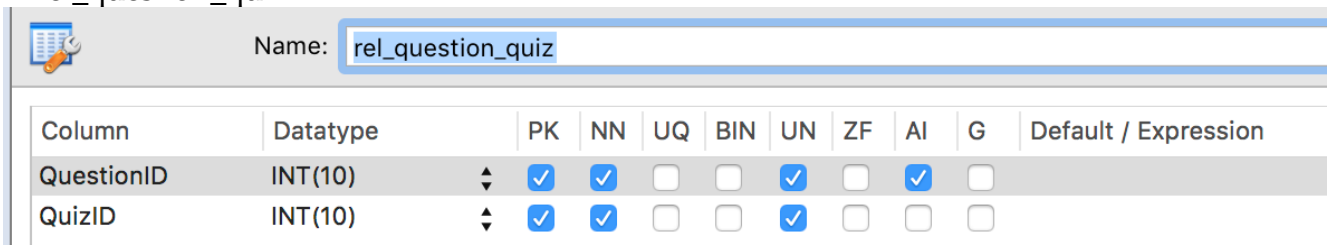
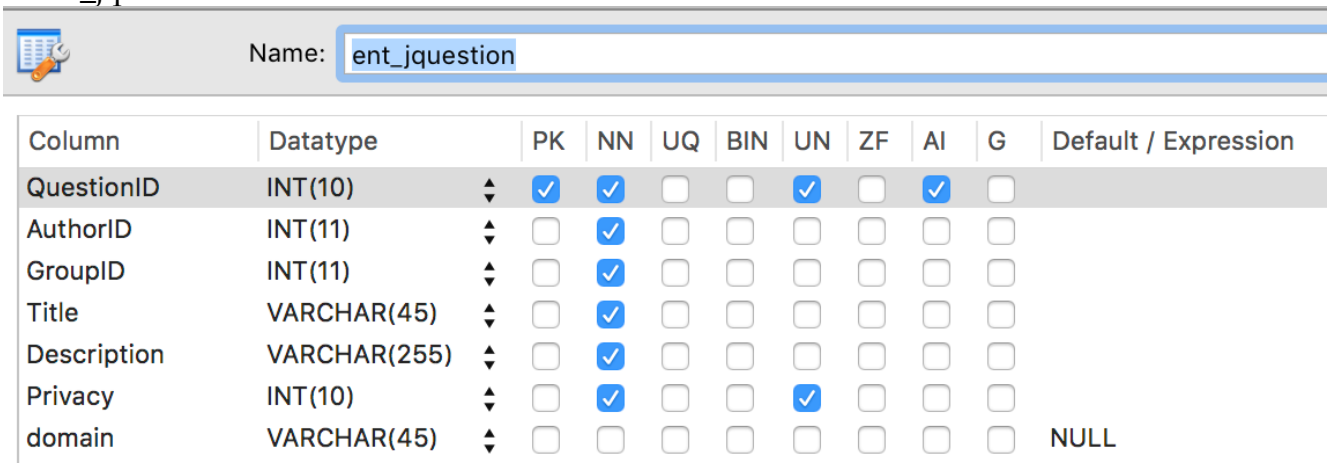
Column	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	G	Default / Expression
QuizID	INT(10)	☑	☑	☐	☐	☑	☐	☐	☐	
ClassID	INT(10)	☑	☑	☐	☐	☑	☐	☐	☐	'0'

3. ent_jquiz

Name: ent_jquiz

Column	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	G	Defau
QuizID	INT(10)	☑	☑	☐	☐	☑	☐	☑	☐	
AuthorID	INT(11)	☐	☑	☐	☐	☐	☐	☐	☐	
GroupID	INT(11)	☐	☑	☐	☐	☐	☐	☐	☐	
Title	VARCHAR(45)	☐	☑	☐	☐	☐	☐	☐	☐	
Description	VARCHAR(255)	☐	☑	☐	☐	☐	☐	☐	☐	
Code	MEDIUMBLOB	☐	☑	☐	☐	☐	☐	☐	☐	
MinVar	INT(11)	☐	☑	☐	☐	☐	☐	☐	☐	
MaxVar	INT(11)	☐	☑	☐	☐	☐	☐	☐	☐	
AnsType	INT(10)	☐	☑	☐	☐	☑	☐	☐	☐	
Privacy	INT(10)	☐	☑	☐	☐	☑	☐	☐	☐	
rdfID	VARCHAR(45)	☐	☑	☐	☐	☐	☐	☐	☐	
QuesType	INT(10)	☐	☑	☐	☐	☑	☐	☐	☐	
start	INT(10)	☐	☐	☐	☐	☐	☐	☐	☐	'2'
end	INT(10)	☐	☐	☐	☐	☐	☐	☐	☐	'4'

4. rel_question_quiz



5. ent_jquestion