

IBM DB2 Contest Guideline

Prepared by Raymond WONG
Updated on 20 Dec, 2012

Note: The following content is based on the past contests. Contests may vary from years to years. It does not mean that the contest this year will have the same format. The following is just for your reference.

Core Suggestion for Contest

Time is critical.

In the contest, you cannot finish all questions. You should do most of the questions as quickly as possible in order to gain marks.

Besides, attending the training session is very important. You must know all concepts and the environment for the contest.

Contest Details

The total time for the contest is about 2 hours. But, it may be different this year.

Each team consisting of 3 members is given a single PC only. Each member is allowed to bring any hardcopies of resources (e.g., the hardcopy for the source code you prepared by yourself and the book you obtained during the training session). This means that for any time, only one member can use the single PC and type some source codes in the PC. The other 2 members may read other questions during the contest or write (or “draft”) some answers in some blank sheets so that you can quickly type the answers when they use the PC later.

From the beginning of the contest to the time just 30 minutes before the end of the contest, the whole contest venue is equipped with a display board showing the scores of all teams during the contest. However, during the last 30 minutes, the board is closed and does not show any scores for secret.

The following shows the system configuration of the PC according to the past experience.

Browser	IE
Operation System	Windows

Programming Editor	Eclipse-like Editor (developed by IBM)
---------------------------	--

Before the contest starts, you are allowed to test the PC you will use. Besides, you are also asked to test the submission system installed in the PC so that you know that the submission system works well at your PC to submit your code (or the hyperlink of some web applications).

Contest Format

There are two types of questions.

- 1. Concept-Related Questions
- 2. Programming Questions

1. Concept-Related Questions

In the past, there may be about 15~20 concept-related questions. Usually, these questions are multiple-choice (MC) questions. These questions mainly test some basic concepts learnt in the training session. According to the past experience, you will be given a book/some notes in the training session. Some topics in the book are related to some basic operations in IBM DB2. Thus, you need to read the notes and memorize some important parts and concepts of the book.

Suggestion: After you obtain the book in the training session, please read the book carefully. According to the past experience, you are allowed to bring the book with you. Thus, the best suggestion is that you can quickly find the resources whenever you are asked for a particular category/tool/part during the contest. So, I suggest that you should make some pointers/indexes in your book in order to speed up your search on the book. Of course, a “better” suggestion is that if you have time, you can try to memorize some important concepts in the book so that you can avoid searching during the contest.

2. Programming Questions

In the past, there may be about 7~8 programming questions. Each question weighs different scores. Each question is a web-based question. In the past, you were asked to display some HTML files by using Java, Applet and JDBC. Thus, you need to know the “raw” format of the HTML file and how to program in Java and Applet. In the contest, you are given an editor which is like Eclipse. If you don’t know Eclipse, you can read more information about Eclipse in the following link.

<http://www.eclipse.org/downloads/>

This editor allows you to type the code and also compile the code.

In the contest, for each programming question, you are given some codes which provide a skeleton for the whole question. You are required to fill in some missing parts of the codes. Usually, the missing parts correspond to the task to handle the query displayed in the form format in HTML (by using HTML form), retrieve records from the databases (by using SQL and JDBC) and display those records in the HTML file (by using Java, Applet and HTML).

According to the past experience, in most questions, the display format in the HTML file is the table

format. Please make sure that you understand how to write a table in the HTML file from scratch (instead of using some tools like Frontpage and Dreamweaver).

Besides, the HTML file displays an HTML form in order to let users to input some inputs (called input fields) or some choices (called combo boxes). According to the past experience, the number of input fields is at most 5. But, it may vary from years to years.

According to the past contest, after you finish a programming question, you can submit a hyperlink via the submission system in the contest. If your code is totally fine, then you will obtain full scores. However, if your code is partially correct, then you will obtain partial scores. In the past, you are allowed to submit a single programming question for at most 3 times. But, the number of times for submission varies each year. In some years, it was unlimited.

Suggestion: Don't do the questions in sequential order. Try to do less time-consuming questions with high scores first. In the past, time is very important. You could not finish all questions during the contest.

The following shows a sample question given by a past participant.

Example 1 (Timetable): We are given some databases like Course, Professor and Schedule.

Please write a HTML file which takes the professor name as an input and output his/her time table. Please display the time table in the web application in the following format. The following shows the time table for "Prof. Wong".

Time	Mon	Tue	Wed	Thu	Fri
9:00-11:00	COMP123				
11:00-13:00	COMP123		MATH444		
13:00-15:00			MATH444	CHEM222	
15:00-17:00				CHEM222	

Programming Languages You Need to Know

The following shows the programming languages you need to know.

- HTML (e.g., form and table)
- SQL
- Java
- Applet
- Servlet
- JSON
- Javascript

I suggest that you should know how to program in each of the above programming languages **quickly**. Understanding one programming language is necessary. But, the requirement is that you should be able to program **quickly**.

Tools You Need to Know

You are required to know the following tools.

- IBM DB2
- Eclipse-like Editor (developed by IBM)

Before the training, you don't need to know the above tools in detail. You can just browse the web to see what they are first. You will know a lot after you attend the training session. I suggest that before the training session, you should have a good skill for the programming languages I mentioned above (e.g., HTML, Java, Applet and Javascript).

Suggested Strategy for Contest

The following shows some suggested strategies for the contest.

Strategy 1. [Compulsory] Allocate **at least 2** members for the programming questions. If possible, the 3rd member should also know how to do the programming questions (e.g., portal) as much as he can.

I understand that some of your teams do not have **all 3** members who are good at doing the programming questions.

My suggestion is that in this case, at least 2 members should be trained for more programming questions and reading more materials about programming before contest. Besides, I suggest the 3rd member should try his best to do “some” programming questions if he is weak at programming.

Usually, if one of the members is weak at programming or SQL, you can ask him to do the multiple-choice questions and do some programming questions related to “tools” (e.g., IBM portal) (by using some graphical interface). The reason why this member still needs to do the programming questions is that after he finishes all multiple-choice questions, he will still have some remaining time during the contest. According to the past experience, it only takes at most 45 minutes (out of the 2 hours for the whole contest) to complete the multiple-choice questions.

But, if all of the members are good at programming, then each member should be trained for programming.

Strategy 2. [Compulsory] Submit the answers for the multiple-choice questions **near to the end** of the contest. But, please remember to submit it!!!

According to the past experience, you can only submit the answer of the multiple questions once only. Besides, you do not know whether you answer correctly for each question. At the end, you just know the final score of the whole contest.

Strategy 3. [Compulsory] When a member finishes doing all multiple-choice questions, please mark all “unconfident” questions. One of the other members can check the answers of these questions.

Strategy 4. [Optional] When a member finishes doing all multiple-choice questions, one of the other members can double-check the answers of ALL questions. This is OPTIONAL!!! I do not suggest it.

If the member is very smart, good at studying and is very confident for the multiple-choice questions, then I suggest that the double-checking step is NOT necessary. In this case, it is more likely that the scores obtained finally for the multiple-choice question are very high. Doing double-checking may consume a lot of time. Besides, the multiple-choice questions weights 15% according to the past experience (according to 2012 training notes Page 3) (Note: In contest 2010, they weighed 22.5% instead. Although the training material for 2012 shows that they weigh 15%, it may change during the contest). I do not suggest this strategy for this case.

However, if the member is not confident enough, you can consider asking other members to double-check it. But, please be careful for this strategy since double-checking the multiple-choice questions can consume the whole team to spend less time on doing programming questions.

My suggestion is that we should not follow Strategy 4 unless all remaining questions are very difficult to finish.

Strategy 5. [Compulsory] When you are doing the programming questions, you should do the questions which is less time-consuming but has more scores.

Strategy 6. [Compulsory] If all **remaining** questions are very difficult, please write some codes related to display (or the interface output).

According to the past experience, each question asks you to display some outputs. There are two parts for the score. The first part is related to the correctness of the output (i.e., whether the displayed records/tuples are correct or not). The second part is related to the interface output (i.e., the output format). According to the past participants, the second part (i.e., the interface output) weights about 25%. But, it may vary from years to years.

I suggest that you can follow this strategy when there are some “remaining” questions that you do know how to do it (since the interface parts weight not much (only 25%)). It is much more desirable to do the questions you know how to do (provided that you have sufficient time).

Strategy 7. [Compulsory] When you are near to the end of the contest (which means that you have insufficient time to finish some questions), please do some questions which allow you to gain marks (e.g., you can do simple questions and you can follow Strategy 6 above).

Strategy 8. [Compulsory] Do the exercises/labs given during the training session.

According to the past experience, about 2 programming questions (out of 7~8 questions) during the contest are very similar to the questions which appear during the training session. Some of them are exactly the same except some differences in a few lines of code. Thus, please practice the exercises/labs which can be found in the training book.

Strategy 9. [Compulsory] Please prepare a hardcopy of some materials (e.g., codes). Details can be found below.

You can prepare a hardcopy of the following code.

- Sample code of using JDBC in **DB2** illustrating the following concept
 - Connect statement in SQL
 - Select statement in SQL
 - Create statement in SQL
 - Delete statement in SQL
 - Drop statement in SQL
 - Close statement in SQL
- Sample code of using Java Servlet for handling form requests
- Sample code of using JSP for handling form requests
- Sample code (in Java) for displaying table in HTML format
- Sample code for transferring the parameters used in an HTML form to Java Servlet and JSP when a user clicks a “submit” button
- Sample code for Javascript that you may use
- Sample code for Java Applet that you may use
- Sample code for using hadoop
- Schema of each table used in the training session (This is because a past participant told me that the schemas which appear in the training session are very similar to those which appear in the contest).
- All tuples of each table used in the training session.
- Reprinted version of the PPT files (since the printing quality of the training book is not good. You can consider re-printing all PPT materials).

Strategy 10. [Compulsory] Please bring some reference books.

The reference books can be related to any context mentioned above (in Strategy 9).

Strategy 11. [Compulsory] Please write down the schema of each table on blank sheets (if needed) for each question if the schemas which appear in the contest are different from those which appear in the training session.

Some past participants suggest that it should be good if we can write the schema of each table (including all attribute names and their corresponding data types) on blank sheets for the ease of reference. But, I suggest that we should do it when we really need to use those tables.

Strategy 12. [Compulsory] You can submit the answer for each programming question for at most three times only.

Strategy 13. [Compulsory] Prepare a list of exact functions provided by DB2. In this list, we can have a link/reference (or page number) to the corresponding detailed description of the function in the training book.

Strategy 14. [Compulsory] When a member is doing a programming question and is typing programs, another member can read the remaining questions and do these questions on blank sheets (or drafting). You can consider bringing some blank sheets.

Note that only one member can type the PC at each time period. Thus, we need to maximize our output. The second member can type it very quickly according to his written code (or draft) after the first member finishes his typing.

Strategy 15. [Optional] When a member is doing a programming question and is typing programs, another member can consider checking what the first member is typing in order to check whether the code is correct or not. This is optional.

According to this strategy, we may waste one mainpower to do the checking.

In my opinion, I suggest that the second member can check the code at the “middle” time when the first member is typing. Then, the second member can spend his own time on doing some programming questions on “papers”. After this “middle” time, the second member can check the correctness of the first member.

Of course, if the first member is very good and does not need any checking, then I suggest that the double-checking step should be avoided in order to save time for other questions.

Strategy 16. [Compulsory] Please focus on the three new parts for this year (2012) compared with last year. The new parts are “Cloud Computing” (concepts), “Hadoop” (an implementation of cloud computing) and Business Intelligence.

This strategy is the focus only. This does not mean that you ignore the other parts. I guess that it is more likely that the above three parts will appear in the contest since they are new.

Strategy 17. [Compulsory] Please try to think out some possible queries for the databases given in the training session. Then, according to these queries, please write a code for that and bring the hardcopy of the code to the contest.

According to the past experience, the databases used in the training session are very similar to those used in the contest. It is good to “predict” the queries in the contest and “pre-write” the code.

Question Types for Contest

The following shows some programming question types provided by a past participant.

- Simple SQL questions (doing some SQL queries on a single table)
- Complicated SQL questions (doing some SQL queries on multiple tables via “join” operation)
- Questions related to “Portal Service”
- Questions related to how to configure services
- Questions related to how to deploy new services

Some Different Skills You Need to Know (Different from What You Programmed Before)

There are 9 types of skills you need to know.

1. How to write an SQL language
2. How to write an HTML source file
3. How to write some simple Javascript
4. How to handle the request sent from the form in the HTML file
5. How to write a Java language
6. How to write a Java Applet language
7. How to write an event-handling programming (used in Java Applet)
8. How to write a JDBC-related code in Java
9. How to display a programming language (e.g., HTML) by using another programming language (e.g., Java)

1. How to write an SQL language

This is a database contest. Of course, you need to know how to program in the SQL language (the database programming language). If you don't know SQL, it is a must that you need to learn it now. Please go to the library to borrow some books for that.

2. How to write an HTML source file

In the past, the programming questions are related to some web-based applications. You not only need to understand the HTML source file by reading but also need to write the HTML source code by yourself. This is because the questions require you to write (or output) the HTML source file.

In particular, I want to emphasize that you need to know some basic HTML tags (e.g., <p> and
), some form-related tags and some table-related tags. Some resources are provided as follows. You can search more information for the HTML code.

References:

http://www.w3schools.com/html/html_primary.asp

http://www.w3schools.com/html/html_forms.asp

http://www.w3schools.com/html/html_tables.asp

(I suggest that you should borrow books from library or search information from the internet for more materials).

In the past, most of you may use some existing tools (e.g., Dreamweave and Frontpage) to create a webpage. In this contest, you should know how to write the HTML source files. Thus, you need to learn how to program in HTML.

3. How to write some simple Javascript

Usually, when you write an HTML file, you also write with Javascript. Thus, it is better for you to know the Javascript. One of the team members recommend that we can also learn JSON in order to speed up programming (related to some object handling) during the contest.

4. How to handle the request sent from the form in the HTML file

Since the questions are related to some web-based applications, the HTML file usually contains the form. Then, your program needs to handle the request from the form if the user using the form clicks the "submit" button in the form.

If we use Java to handle the form request, there are at least two types of Java variants for handling. The first variant is Java Servlet while the second variant is JSP. I am not sure which type will be used in the contest. But, for the sake of interest, you can read some information as follows.

References:

- Servlet for handling the request

<http://www.apl.jhu.edu/~hall/java/Servlet-Tutorial/Servlet-Tutorial-Form-Data.html>

- JSP for handling the request

http://www.tutorialspoint.com/jsp/jsp_form_processing.htm

(I suggest that you should borrow books from library or search information from the internet for more materials).

5. How to write a Java language

Since the Java language is a host language according to the past experience, you should know how to program in Java. Please borrow books if you don't know Java.

6. How to write a Java Applet language

Writing in a "normal" Java language (or a C++ language) is different from writing a Java Applet language. The first difference is that you need to know the methods related to Java Applet (especially the GUI methods) (e.g., Button and Radio).

References:

<http://www.realapplets.com/tutorial/GuiExample.html>

(I suggest that you should borrow books from library or search information from the internet for more materials).

7. How to write an event-handling programming (used in Java Applet)

As you are required to write in Java Applet, in addition to the methods related to GUI, there are some other new concepts in Java Applet which cannot be found in writing "normal" C++ programs and Java program. The new concepts are related to event handling. In event handling programming, there is no function/method called "main". The major idea is that whenever there is an event (e.g., mouse click and keyboard typing), the program will execute the corresponding parts of the program in order to response the event.

References:

<http://www.javacoffeebreak.com/java107/java107.html>

(I suggest that you should borrow books from library or search information from the internet for more materials).

8. How to write a JDBC-related code in Java

After you understand SQL and Java, of course, you need to integrate these two concepts together. You can link them together by using JDBC. When you write a Java language, you can use some "special"

methods in Java to call the SQL language. Thus, you can retrieve records from the database by using JDBC.

References:

<http://www.users.csbsju.edu/~lziegler/CS317/NetProgramming/JDBCAppletLecture.html>

(I suggest that you should borrow books from library or search information from the internet for more materials).

9. How to display a programming language (e.g., HTML) by using another programming language (e.g., Java)

As you know, you have to use Java (or JDBC) to retrieve records from the database. After that, we need to display the content into an HTML file. Here, you can see that you are writing in Java to obtain the records and store the content of these records into some variable in Java. The next step is to display it. Here, your Java program (one programming language) has to output a file for display. This file is an HTML file. In other words, you need to learn how to write a Java programming language to output another file (which is in the HTML file). In the past, usually, you write an HTML file by yourself. This time, you need to use a Java program to output an HTML file (instead by yourself). You can I

References:

-Servlet

<http://zetcode.com/tutorials/jeetutorials/datasource/>

- JSP

<http://www.easywayserver.com/jdbc/JDBC-query.htm>

(I suggest that you should borrow books from library or search information from the internet for more materials).