

Applications

Dynamic Generation of Testing Questions in SQL in DeTC

*Olga Rahneva*¹, *Angel Golev*², *Nikolay Pavlov*²

¹ *Department of Informatics and Statistics, University of Food Technologies, 4000 Plovdiv*

E-mail: rahneva@hiffi-plovdiv.acad.bg

² *Faculty of Mathematics and Informatics, Plovdiv University "Paisii Hilendarski", 4003 Plovdiv*

E-mails: angel@kodar.net npavlov@kodar.net

Abstract: *This paper describes how dynamic questions for testing in SQL are being created, generated and applied into practice at a Distributed e-Testing Cluster – DeTC. As a result, the learners receive unique testing questions, while the database with tests does not grow significantly in size, and the size of the information transferred is minimized.*

Keywords: *Electronic Testing, e-Testing, DeTC, Dynamic Question, Free Answer.*

1. Introduction

Testing examination is one of the most popular and well-developed assessment instruments in higher education [1]. The classic test is a sequence of precisely defined questions, each question suggesting a simple answer, which can be easily checked and assessed as correct, incorrect or partially correct (for example, incomplete).

Questions are often split in types, according to the expected answer:

- a classic type of question – with a “yes/no” answer;
- a multiple-choice question – one answer (MC/SA, Multiple-Choice/Single-Answer);
- a multiple-choice question – more than one correct answer (MC/MA, Multiple-Choice/Multiple-Answer);
- free-type questions – with the answer being a number or a text;
- others.

Most of the existing Web-based systems for testing and assessment provide technologies and tools for creation, submission and assessment of questions of the first three types [2-6]. A certain number of systems process all types of questions [7,8].

However, most of the latter do not provide a solution to the problem for dynamic creation of questions, which forces the development of a large set of test versions, in order to prevent the test from being learned by heart.

The IMS Question & Test Interoperability (QTI) specification describes a data model for the representation of questions and test data and their corresponding results reports. Since 2005 starting with version 2.0, QTI supports parameterized questions via assessment item templates [9].

There exist online testing systems in SQL, but they work with a predefined set of questions on SQL, and choose randomly a subset of them when creating a test [10, 11].

It is described below how dynamic free-type questions for testing in SQL are generated and applied in practice at a Distributed e-Testing Cluster – DeTC. Learners are given unique testing questions by generating random values for the variables in the question. This does not increase the number of the preliminary created questions, decreases the size of the database with tests, and minimizes the size of the transferred information.

DeTC [12-14] is being developed as a joint project of the University in Limerik – Ireland, and the Department of Computer Systems and the Department of Computer Technologies at the University of Plovdiv, Bulgaria, and the Department of Informatics and Statistics at the University of Food Technologies, Plovdiv, Bulgaria.

2. Generating dynamic testing questions in SQL

In DeTC, learners are provided with these types of testing questions in SQL:

- A free text question: write an SQL query on a testing question.
- A multiple-choice question, which can be one of the following:
 - Select the correct answer out of multiple pre-generated SQL queries.
 - Select the error in a pre-generated SQL query from a list of errors.
 - Determine which one from a list of pre-generated SQL queries returns a given result.
 - Determine the result of an SQL query.
- Find and correct the errors in a pre-generated SQL query.

The described solution uses a base of texts and templates to generate multiple correct SQL constructions.

For each group of tests there should be a general description of the used tables, their fields, and value ranges for each data type. According to the questions types, it might be necessary to create these tables in a real database, and populate them with data for correct query execution.

An example is given in Fig. 1.

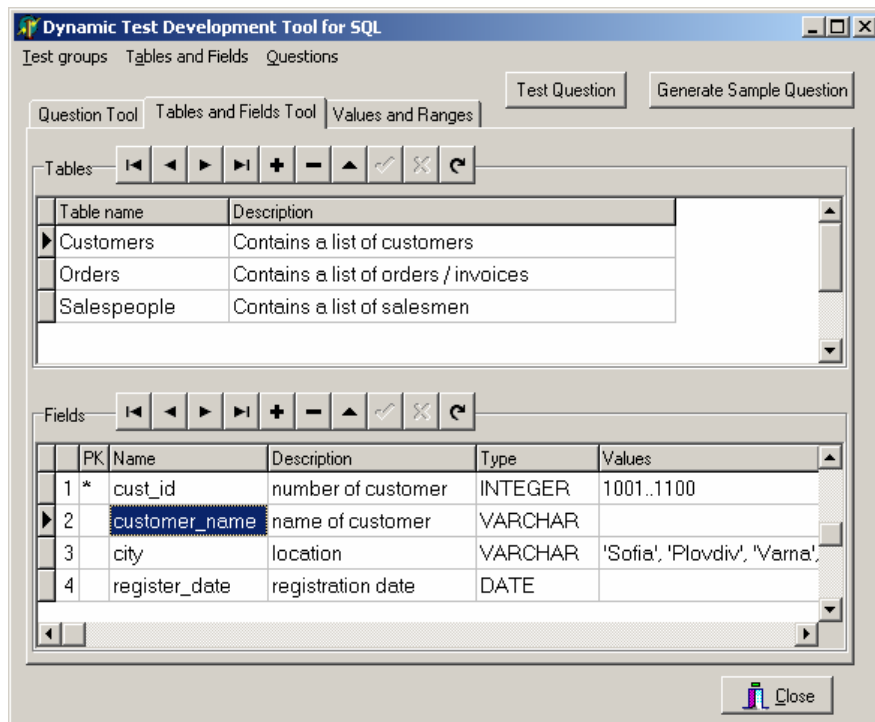


Fig. 1. Description of tables and their fields

Every dynamic question is described by a number of sections. Section `text_pattern` describes the text contents of the questions. Special tags are used to denote the places in the text where names of database tables and fields will be dynamically inserted. Symbol `@` is used as a prefix for tag names, in accordance with the notation in DeTC. These tags are in format `@field1` for the selected field and `@table1` for the selected table, and `@oper1` for comparison operators. The test generation tool (TGT) selects random values for names of fields and tables when generating actual questions.

Section `answer_pattern` describes the contents of a SQL query and uses the tags described above, for example `@field1`, `@table1`, `@oper1`, etc.

Section `system_gen` describes the rules for dynamic selection of tables, fields, operators, data types and value ranges for each type.

Sections `test_tables`, `ranges`, `table_data` contain information about tables, value ranges per data type, and description of data, which are used to create and fill the sample tables. These sample tables are created and populated with data when SQL queries of testing questions have to be executed.

3. Generated elements syntax

The names of all tags, used to generate dynamic SQL queries start with `@`. The templates and system information is described in XML. There a dollar sign “\$” is used instead of `@`.

Table 1

| | |
|---|---|
| <table value='Tables'> \$table1 </table> | Select for @table1 a random name from the list of described table |
| <table name='Orders'> \$table2 </table> | @table2 is linked to physical table Orders |
| <field rand='yes' table='\$table1'> \$field2 </field> | Select for @field2 a random field from @table1 |
| <field name='onum' table='\$table1'> \$field1 </field> | @field1 is statically linked to field "onum" from table @table1 |
| <oper rand='yes' type='\$wfield.type'> \$oper1 </oper> | Select a random comparison operator, applicable to the data type of field @wfield |
| <value rand='yes' type='\$wfield.type' range='free'> \$value1 </value> | Select a value, corresponding to the data type of field @wfield |
| <field name='onum'> <fullname>Order number</fullname> <type> integer </type> ... </field> | Describes a field in a table |

4. Extending DeTC tools

Dynamic Test Development Tool (DTDT) is used to create and describe testing templates. DTDT is an extension of DeTC's test editor Test Development Tool (TDT). It is possible to create testing templates without knowledge of the XML description of the testing tables and their data.

An extension to DeTC's Dynamic Test Assessment Tool (DTAT) is created to enable DTAT assess tests in SQL. The free form syntax of SQL raises two problems before the task of automatic assessment. The first problem is related to the whitespaces and formatting of the answer. The extension of DTAT uses a SQL parser to compare the correct answer of the test question with learner's answer. The second problem is related to the logic of the SQL queries. Additional analysis is necessary when comparison operators are used. Operations in learners' answers and the given correct answers must be checked whether logically equal.

For example "city = 'Plovdiv'" is equal to "not city < > 'Plovdiv'".

The process of generating tests in SQL is outlined in Fig. 2.

DeTC features a conversion tool for exporting DeTC-generated questions into QTI XML format, Version 1.2. Since 2006 this conversion tool is being extended to support QTI Version 2.0, and export both generated questions, and question templates into QTI assessment item templates. This enables interoperability of DeTC with authoring tools, item banks, test constructional tools, and learning systems, which support QTI.

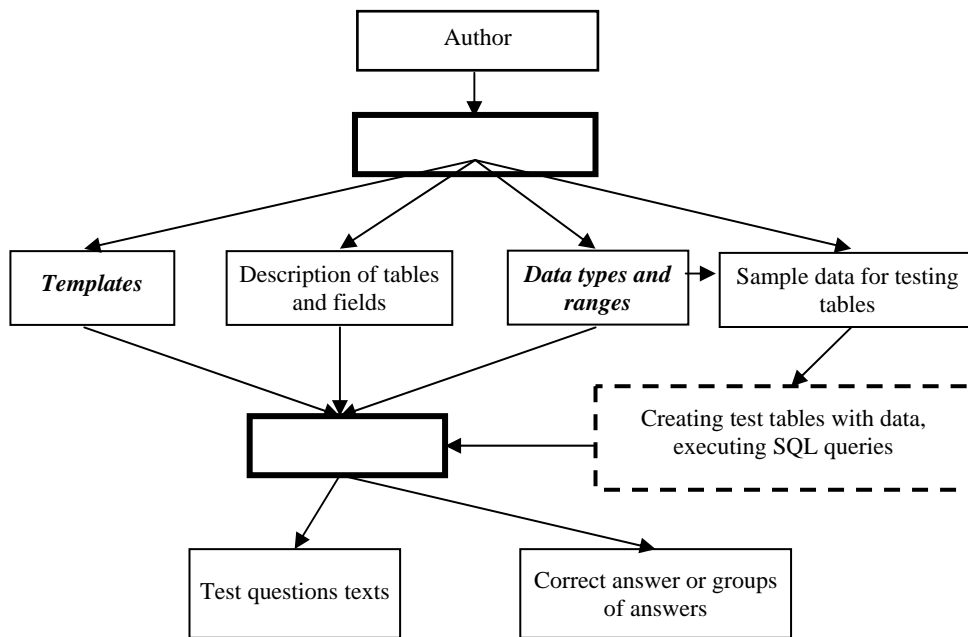


Fig. 2. Generating tests in SQL

5. Real-life implementation

The described approach is used in real-life testing examination in lectures on databases and SQL. Three examples are presented here.

Example 1. Question “Write a SELECT query, which returns the sequential number, sum and date for all records of table ‘Documents’” can be created by the following template:

Write a SELECT query, which returns @field1, @field2 and @field3 for all records of table @table1.

The template of the correct answer will be:

```
SELECT @field, @field2, @field3
FROM @table1
```

A correct answer of the question of Example 1:

```
SELECT cust_id, amts, odate
FROM orders
```

The complete template of Example 1 in XML format is:

```
<test_example1>
  <text_pattern> <![CDATA[
    Write a SELECT query, which returns @field, @field2 and
    @field3 for all records of table @table1. ]]>
  </text_pattern>
  <answer_pattern> <![CDATA[
```

```

        select @field, @field2, @field3
        from @table1 ]]>
</answer_pattern>
<system>
    <table values='Tables'> $table1 </table>
    <field primarykey='yes' table='$table1'> $field </field>
    <field random='yes' table='$table1'> $field2 </field>
    <field random='yes' table='$table1'> $field3 </field>
</system>
</test_example1>
<test_tables>
    <table name='orders' descr='Documents' ... >
        <field name='cust_id'>
            <descr> Sequential number </descr>
            <type> integer </type>
            ...
        </field>
        <field name='amt'>
            <descr> Amount </descr>
            <type> numeric(10,4) </type>
            ...
        </field>
    </table>
    <table name='customers' descr='Customers' ... >
        <field name='...'>
            ...
        </field>
    </table>
    ...
</test_tables>
<ranges/>
<table_data/>

```

Example 2. Question “Write a SELECT query which returns the number of customer, registration date for all records for table “Customers”, meeting the condition “city = ‘Plovdiv’”.

```

<test_example2>
    <text_pattern> <![CDATA[
        Write a SELECT query which returns @fieldlist1 for all records for
        table @table1, meeting the condition @wfield @oper1 @value1.
    ]]>
</text_pattern>
<answer_pattern> <![CDATA[
    select @fieldlist

```

```

        from @table1
        where @wfield @oper1 @value1 ]]>
</answer_pattern>
<system_gen>
  <table values='Tables'> $table1 </table>
  <fieldlist random='yes' table='$table1'>
    $fieldlist1
  </fieldlist>
  <field random='yes' table='$table1'> $wfield </field>
  <oper random='yes' type='$wfield.type'> $oper1 </oper>
  <value random='yes' range='free' type='$wfield.type'>
    $value1
  </value>
</system_gen>
</test_example2>

```

Figs. 3 and 4 demonstrate how DTGT is used to create the dynamic question template, and a sample generated question with its correct answer.

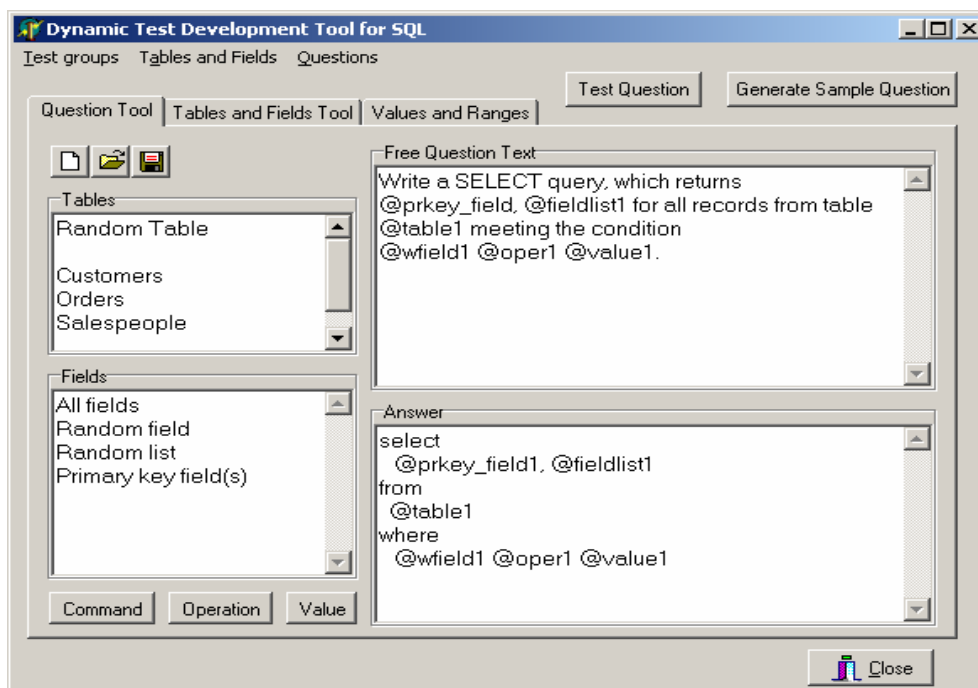


Fig. 3. Creating templates

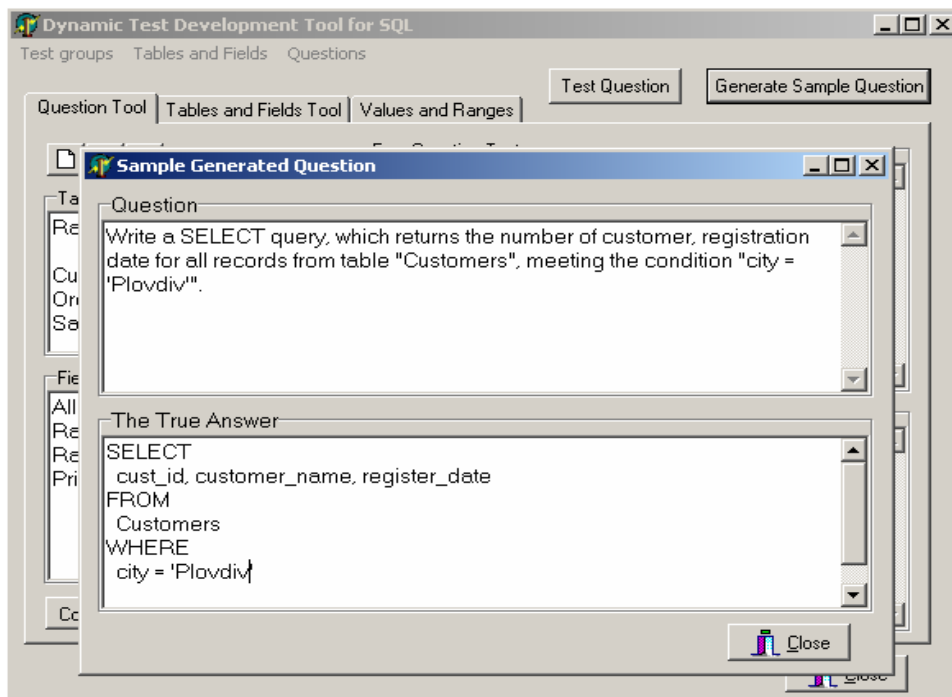


Fig. 4. A sample generated question and its correct answer

Example 3. Question: fill in the missing text in a SQL query.

Fill in the missing text:

```
SELECT *
... orders
... orders.onum > 20
```

Template definition in XML format:

```
<test_example3>
  <text_pattern> <![CDATA[
    Fill in the missing text:
    @blank{SELECT} *
    @blank{from} @table1
    @blank{where} @field1 @oper1 @value1 ]]>
  </text_pattern>
  <system_gen>
    <table values='Tables'> $table1 </table>
    <field random='yes'> $field1 </field>
    <oper random='yes' type='$field1.type'> $oper1 </oper>
    <value type='$field1.type'> $value1 </value>
    <blank places='2'/>
  </system_gen>
</test_example3>
```

The example templates above can be used also to generate multiple-choice

questions like: *what will be the result of the following statement:* The multiple choices are generated by executing the correct queries and some incorrect ones. In this case there can be more than one correct answer.

DeTC is already used to conduct real-life electronic assessment of learners. Results demonstrate that DeTC is recognized by learners as very convenient and easy to use. Vast majority of the learners certify that their knowledge has been assessed impartially [15, 16]. There is no representative sample for testing in databases and SQL yet.

Acknowledgement: This research has been partially supported by the Bulgarian NSF under Contract number VU-MI 107/2005.

References

1. Brusilovsky, P., P. Miller. Web-Based Testing for Distance Education. WebNet 1999, 149-155.
2. <http://wdo.uni-svishtov.bg>
Last vist December 2007.
3. <http://estudy.iccs.bas.bg>
Last vist December 2007.
4. <http://www.nbu.bg>
Last vist December 2007.
5. <http://www.vg-u.de>
Last vist December 2007.
6. Gusev, M., G. Armenski. A New Model of On-line Learning. – In: Proc. of SSGRR Advances in Infrastructure for Electronic Bussiness, Education, Science and Medicine. Roma, 2002.
7. Bontchev, B., T. Iliev. ARCADE – a Web-Based Authoring and Delivery Platform for Distance Education. – 1st Balkan Conference on Informatics (BCI'2003), Thessaloniki, Greece, 21-23 November, 2003.
8. Ueno, M., K. Nagaoaka. Web Based Computerized Testing System for Distance Education. – In: Proc of ICCE 2001, Korea, 2001, 547-554.
9. <http://www.imsglobal.org/question/>
Last vist December 2007.
10. http://www.w3schools.com/sql/sql_quiz.asp
Last vist December 2007.
11. <http://www.sqlquiz.com/>
Last vist December 2007.
12. Rahneva, O. DeTC – Distributed Electronic Testing Cluster. – Scientific and Practical Conference “New Technologies in Education and Professional Learning”, Sofia, 16-17 May 2003, 84-91.
13. Rahneva, O. Testing and Assessment in Distributed Electronic Testing Cluster – DeTC. – In: 12th International Conference Electronics '2003, Sozopol, 24-26 Sept. 2003, Conference Proceedings, Vol. 4, 214-219.
14. Rahneva, O. Generating Dynamic Questions in Distributed e-Testing Cluster. – DeTC, ECEST'04, Bitola, 2004, Vol.1, 305-308.
15. Iliev, A., V. Valchanov, T. Terzieva. Shared Experience of Software Test System Usage for Conducting Exams in the Course of Information Modeling. – In: Jubilee Scientific Conference “Science, Education and Time as Our Concern”, Smolyan, November 30-December 1 2007.
16. Rahnev, A., O. Rahneva, N. Valchanov. Application of DeTC for Examination and Assessment in Qualification “Teacher in Information Technologies”. – In: Thirty Sixth Spring Conference of the Union of Bulgarian Mathematicians, Varna, April 2-6 2007, Conference Proceedings, 397-403.