Enhancing Database Querying Skills by Choosing a More Appropriate Interface

Jozef Hvorecký
Department of Information Sciences
Vysoká škola manažmentu/City University of Seattle
Bratislava, Slovakia
jhvorecky@vsm.sk

Martin Drlík, Michal Munk
Department of Informatics
Constantine the Philosopher University in Nitra
Nitra, Slovakia
mdrlik@ukf.sk, mmunk@ukf.sk

Abstract – **To benefit from data stored in growing numbers of database systems, the number of users capable of finding relevant information should also grow. They should be capable of posing requests, formulate relevant queries, and interpret the obtained results. The user interface should allow them to concentrate on the core of the problem without disturbances caused by details of the query language's syntax. We show the results of our study of the relationship between the interface offered by query languages and their users' ability to express their requests in a correct and comfortable manner. Our aim was to analyze the state-of-art in introductory database courses from this point of view, to design and to test an educational methodology appropriate for non-professionals. It should help them to achieve an advanced expertise in query design and development.**

All DBMS's contain SQL but many of them also have a version of Query-By-Example (QBE) language. Their developers likely believe that for non-professionals this sort of interface is more appropriate for creating queries than SQL. Our review of literature showed that the topic was hardly studied in depth. To do so, we designed and developed an educational methodology which addresses the frontier field between design of query languages, HCI, users' mental and conceptual models and performed an educational experiment that tested it.

Many of our presumed advantages of QBE have been proved correct. There are statistically significant differences in time, accuracy and users' satisfaction related to query creation. We also found out statistically significant differences in combinations of variables (time, accuracy, and satisfaction) and the level of task difficulty. The result may have a positive influence on non-professionals' exploitation of databases. It may lead to designing and developing a combined methodology that would benefit from positives of both: QBE and SQL.

Keywords- introductory database courses; comparison of QBE and SQL; query formation; mental models of query processing.

I. Problem Statement

The amount of data stored in database applications is constantly growing. To benefit from them, the number of users capable of finding and benefitting from them should also grow. Many of those search activities have been programmed, but new and new questions always arise. For that reason the database users should be capable of posing their requests. The users should be capable of recognizing whether they are solvable within the given database, formulate relevant queries, and interpret their obtained results. These issues are of a distinct importance for non-professionals in ICT as the query creation is for them just a tool for achieving their professional information. The user interface should allow them to concentrate on the core of their expert problem without disturbances caused by details of the query language's syntax. Thus, questions like the next ones can be posed: *Which query language satisfies the non-professionals' expectations the most? How to measure the effects of a query language to its users? What methodology to apply and how far to go in their training?*

In this paper, we show the results of our study that analyzed the differences between SQL and graphical Query-By-Example (QBE) in order to get relevant answers. We concentrate on distinctions that enhance learning processes of ICT non-professionals, and analyze the user's ability to express their requests in a correct and comfortable manner.

II. Purpose of Study

The choice of the background languages for our study was quite natural. SQL is a standard for database management systems (DBMS). Many of them also introduce their own semi-graphical version of the Query-By-Example language, often labeled as a QBE-like language (thereinafter QBE). Developers of DBMS likely believe that this sort of interface is more appropriate for casual and non-professional users. Nevertheless, our examination of literature showed that the topic was hardly studied in depth.

As similar considerations represent a frontier between design of query languages, computer-human interface, and users' mental and conceptual models, we designed an educational methodology which reflects all of these aspects and does not depend on a particular query language. Based on the methodology, we developed two parallel courses of query design and development. The first one follows the classical approach based on SQL. The second one also exploits the advantages of QBE. The methods were used in two parallel classes of the same *Introduction to Databases* course.

Three hypotheses were specified. They addressed speed, quality and comfort of query creation in the two languages. In all of them, we favored QBE i.e. we presumed that the users of

978-1-4244-6571-2/10/$26.00 ©2010 IEEE

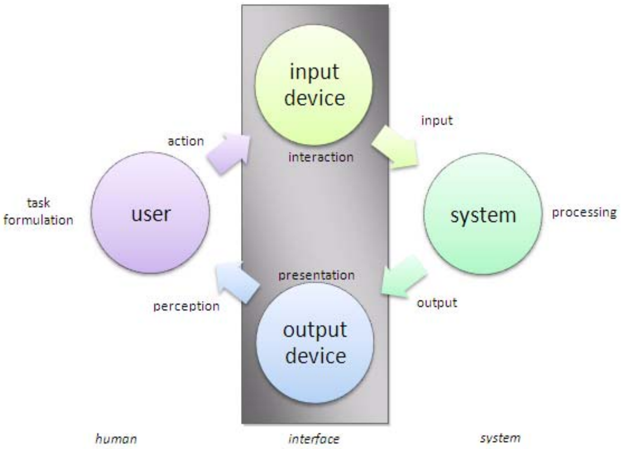

Figure 1. Basic cycle of interaction by Abowd [1].

QBE will be more successful in all three dimensions than users of SQL. Then, a pedagogical experiment was performed – the test group was using our new approach using QBE while the control group used the more traditional SQL.

III. Interaction Models

The human-computer interaction has been the key issue in designing our educational methodology. Here, "humans" mean database end-users, non-professionals in ICT. Our aim is to teach them query creation to the degree in which they can interact with their database applications and gain answers to their questions. Accordingly to Abowd [1], the interaction is executed in the way indicated in Fig. 1. Notice that the differences between QBE and SQL only address its "upper left corner". The rest is virtually identical for any query language because:

- The evaluation of the queries is hidden from the users;
- The presentation of the results does not depend on the used input language. In the same DBMS, it is identical for both of its query languages.

For that reason, the conceptual model of the interface, the user's mental model of this interface, and the query notation are the only factors playing key roles in the communication [2]. The conceptual and mental models are important for the design of human-computer interaction, as well as during users' learning. In a badly-designed conceptual model, confusing interaction can make the solution of the users' task more complicated. The learners with wrong mental models will meet with troubles during forming queries that should answer their requests. Oppositely, well designed interaction simplifies the problem solution on one side, and the creation of learners' mental models on the other one. The users' success rate and, consequently, the popularity of the query language will grow. Some sources [2], [3] indicate that these topics have been neglected by researchers. Nevertheless, we feel inevitability to explain the relevance of their research to our own one.

Prior to that, we introduce our terminology.

A. Conceptual Model of the Interface

For designers, the interface represents a communication channel between front and background activities. It transfers messages between the system and the user. The communication is based on so-called *conceptual model*, i.e., the general framework in which the data and information move between the front-side (visible, operated by the users) and back-side (invisible, executed by the DBMS).

The conceptual model represents the DBMS's functionality. It is preprogrammed. The experienced user's internal (mental) model of this functionality coincides with the conceptual model well. That's why the experienced users are capable of exploiting the features of their DBMS much better than non-experienced ones despite the fact that both categories communicate with the same conceptual model.

B. Learner's Mental Model

The main aim of education is to build proper mental models in the minds of learners. Their future mental models should become close equivalents of the conceptual model of the interface they are trained to use.

Every learner exploits his/her mental model to reason about the system, to anticipate its behavior and to comprehend why it reacts as it does [4]. During learning, each and every learner forms his/her individual model with his/her own internal objects and specific ways of understanding of how the computerized processing works. These models may not fully coincide with the conceptual model [5].

From the designer's point of view, the user and the system are "two partners of the same importance". From instructor's point of view, the conceptual model is "done" (i.e. static), whilst mental models are formed (i.e. dynamic) [5]. The educators are therefore supposed to familiarize their students with the conceptual model in a stepwise manner and verify to what degree their mental models match the reality.

In general, people are not very good at constructing and understanding our own mental models or using them for decision making. So, it is difficult to measure the "distance" between people's mental models and the conceptual model. On the other hand, for day-to-day purposes it is sufficient, if the learner's metaphors and analogies lead to their adequate control of the system. Due to that, the mental models we use are often extremely simple.

The mental models have some powerful advantages. They are flexible – they can be adapted to new situations and be modified as new information becomes available. But mental models have their drawbacks also. They are not easily understood by others; interpretations differ person to person. The assumptions they are based on are usually difficult to examine. As a result, ambiguities and contradictions within them can go undetected, unchallenged, and unresolved.

Also, learning never starts from scratch as each of us already owns many mental models. In truth, "*our decisions and actions are based not on the real world, but on our mental images of that world, of the relationships among its parts, and of the influence our actions have on it*" [5]. During the process of their learning, the computer users have to form their own models and to learn to work with them [6]. This also runs during learning of query creation. The role of teaching methodologies is to split the process into „edible chunks“. Each of them shows a small portion of the conceptual model and creates an additional piece of his/her mental model. These individual mosaics (as well as their previous experience with creating their mental images) are composed in the students' heads into their final mental models.

From the point of view of education, such an approach has one advantage. The educator can formulate "an optimal mental model" and build a methodology aimed to it. This so-called *target mental model* specifies the knowledge the students should get in the end of their learning processes. If there is a difference between the concrete mental model of a student and the instructor's target mental model, the instructor can (a) redesign their communication and/or (b) use other learning materials to enhance the transition from the student's current model to the target one [7]. Actually, this idea did motivate us to replace the traditional SQL-oriented courses to QBE-oriented ones as we felt that the former ones are not appropriate to achieve our target mental model (and consequently, the depth of knowledge) we would expect among non-professional users.

Learning is a back-loop process and its run can be illustrated as follows (Fig. 2):

- *Single-loop learning* is shown in the upper part of the picture. Every person has a stockpile of mental models for standard life situations. When we make a decision and then perform its relevant activity, feedback from the real world gives us information on whether our decision brought the desired effect or not. When the feedback is positive, it reinforces our mental model. In such cases, our aims may change, but our mental model remains the same. It is our dominating method of acting; it is simple and convenient. Once the mental model is fixed (and the responses from the real world coincide with our presumptions), our decisions are coming quickly.
- When the feedback from our environment is negative, *double-loop learning* is necessary. This process covers the entire figure. The negative feedback indicates the necessity to change our mental model. Unlike single-learning loops, this model includes a shift in understanding – from the simple and static one to a broader and more dynamic one. We start taking into account the changes in the surroundings that might cause our mental model's imperfection and try to include them as another piece of mosaic [8]. School education is built on continuing challenges to students' current mental models. Teaching methodologies represent recommended strategies guiding the educators and helping them to facilitate transitions from their current mental models to the targeted ones. Notice that there can be several methodologies describing parallel roads from the given mental model to the identical target.

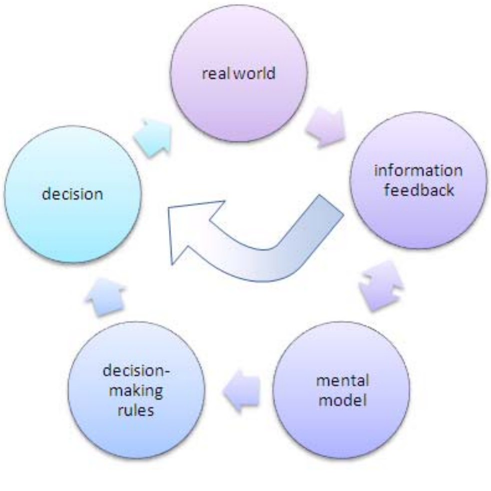

Figure 2. Double-loop learning [7].

C. *Query Creation Target Mental Model*

Our group of learners consists of future non-professionals in ICT. For that reason, the exploitation of databases is not their main aim; it appears as a part of a more general task. Consequently, the target mental model discussed in our paper is composed of the following pieces of knowledge:

1. *Formulation of a question* that can be answered by the means of a given database. As the question mostly appears as a part of a practical problem, it is formulated in a natural language.
2. *Feasibility of the solution* must be discussed. For example, the user must make certain that the addressed database contains all necessary data – otherwise the problem is not solvable and, consequently, there is no reason to open the query creation process.
3. *Transformation of the question* formulated in a natural language to a semantically-equivalent query written in a query language.
4. *Execution of the query* over the database.
5. *Interpretation* of the gained results.

The time allocated to databases in curricula is constant and limited; the proposed methodology must respect these boundaries. For example, the first and the fifth part might be "too specific" and strongly depending on an application field. As our group of learners has not a unified field of study, we overcome this problem by forming a wide range of application databases referring to popular fields like libraries, travel agencies, transportation, personal records, etc. Due to that, we can presume that all students have sufficient prerequisite knowledge and experience to formulate and solve relevant problems of "any" difficulty.

Our methodology is a reflection of Reisner's principal model [3]. Despite its age, the model is still relevant as the structure of databases and their interface has not changed substantially since. In her model, users' performance is determined by the time it takes to think about proper query design, to writ the query as the DBMS input, and to correct various types and numbers of errors. The thinking effort is what Reisner calls the *selection of an appropriate template*. From the point of view of time sequence, the process of query creation consists of two basic templates [9]:

(a) Understanding of the problem and its relation to the database structure;

(b) Formulating a correct syntactic construction that will produce the problem solution.

In detail, the problem consists of the following activities [9]:

- *Understand instructions*: The user must comprehend the instructions that describe the query task to solve the query correctly.
- *Construct deep structure*: The deep structure of a query task is a complete and thorough comprehension of which data must be retrieved from the database and how the result data must be structured. The deep structure of a query task is independent on the used query language.
- *Construct surface structure*: The surface structure of a query task is a query language expression that is a translation of the deep structure. If the user is satisfied with the query expression, he/she submits the query to the processor. When the user is not satisfied with the outcome, he/she makes correction efforts or tests.
- *Submit query*: The user submits the query language expression to the query processor.
- *Evaluate result*: The user reads and tries to interpret the information that he/she received from the query processor. This information may contain an error message or resulting record set (which itself can be correct or wrong).
- *Correction effort*: When the query was not processed or produced a wrong outcome, the user tries to understand the type of error and fix it.
- *Test expression*: The user supposes that the query is correct but makes its additional runs using modified data sets in order to verify its correctness in a more general manner.

The traditional target mental model includes SQL as a carrier for query creation. The reason is quite logical. SQL is a standard component of databases. Thus, the learned knowledge is widely applicable. On the other hand, SQL is a typical programming language with a very formal syntax. As such, it is strict and not well-legible. That's likely why designers of database management systems often introduce a parallel notation named Query-By-Example (QBE).

As SQL is a carrier, it is a substantial but not the decisive element of the target mental model. Moreover, a dependence of a methodology on a particular programming language is not a sign of its high quality. Thus, questions arise: *Can SQL be replaced by QBE? To what degree will the replacement be successful?*

We believed that SQL is too formal for casual and non-professional database users. For that reason we decided to reflect our considerations in the design of a new *Introduction to Databases* course. From one side, it would address the same target mental model as the traditional SQL-based courses do and, from the other side, it would exploit QBE as the query carrier.

To evaluate its effectiveness we designed a pedagogical experiment. The database course was prepared in two versions using the identical educational methodology. Fig. 3 shows its structure. Its third block was delivered in two versions – the traditional one using SQL and the newly developed one based on QBE. The pretest and posttest served for measuring the students' knowledge in the relevant classes (with QBE-class as the tested group and the SQL-class as the control group).

IV. RELATED WORKS

The relation between SQL and QBE was studied many times. Whilst SQL has its standard definition, there are many DBMS-specific versions of QBE. In our paper, we consider its form-based version in Microsoft Access as it was the tool used in our classrooms.

Former experiments comparing the two paradigms did not bring definite evidence on a statistically relevant difference between the effectiveness of their application [10]-[16]. Some of these unambiguous conclusions are presented in the comparative articles [3], [17]-[19]. In accordance to them, both languages require equal capability of abstract thinking and complete specification of the way from the input data table(s) to the output list of results. More recent papers targeting so-called visual query languages analyze different level of abstract thinking that the users of DBMS should have to accomplish their tasks [10], [16]. We have not found a study that would describe research on educational aspects of a graphic-form version of QBE and SQL that would represent an analogy to our study, except in [20].

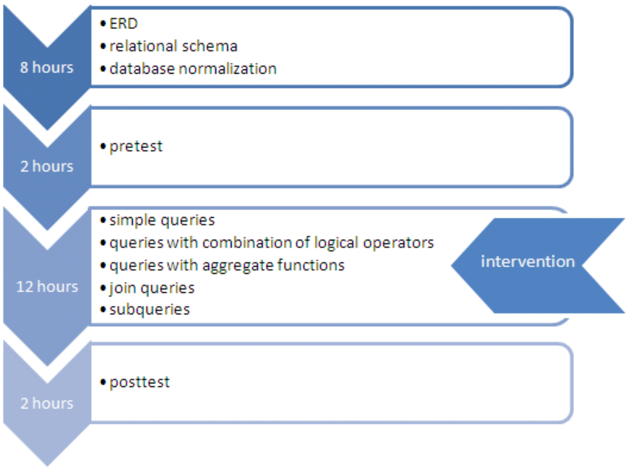

Figure 3. Timeline of the experiment.

V. Research Method

As Fig. 3 shows, the experiment was done during 24 hours of lecture in one semester. The figure also shows its stages. It has the form of a pedagogical experiment with pretest, posttest and two groups – experimental (tested) group and control group.

The design of our experiment followed the standard methodology and its outcomes were statistically evaluated. Its principal steps were as follows:

- Formulation of research hypotheses.
- Creation of experimental and control groups.
- Reliability/Item Analysis:
  - Estimation of reliability of the questionnaire and test.
  - Identification of items that might degrade its relevance and fixation of the problem.
- Execution of the experiment:
  - Simultaneous application and evaluation of the pretest in both groups.
  - Parallel run of courses in the groups.
  - Simultaneous application and evaluation of the posttest in both groups.
- Statistical analysis of the results.
- Interpretation of the results.

A. Target Group

The total number of students participating in the experiment was 116 divided into two groups:

- QBE – 59 students,
- SQL – 57 students.

The groups were formed from the students of the same year-class and were equivalent from the points of view of their age, gender and previous study results. The level of their knowledge was also verified during the pretest and confirmed our presumption on the group equity ($p = 0.330751$).

B. Dependent and Independent Variables

To formulate and verify hypotheses, one needs their variables. The independent variables are query tasks of five difficulty levels to be implemented using two query languages – SQL and the MS Access version of QBE. They represent independent variables in our experiment.

To study which of these tools is more appropriate for our target group the analyzed factors should express their appropriateness for their users. The studied factors were the users' performance and overall satisfaction with this activity.

1) Performance

Performance was measured using two parameters: correctness of the solution and the time for its accomplishing.

The correctness of query is its critical feature. It can be expressed as difference between the presumed (exact) solution and the one produced by the learner. In our evaluations we used the "Correct-Incorrect" scale where the queries semantically equivalent to the right answer (i.e. those always producing the same set of records) were characterized as the "Correct" ones.

Time for completion of the query was the second parameter. Time was measure for each query individually. The student could choose any test task to solve but he/she could not work on two or more tasks simultaneously. Thus, the gained values expressed the time periods each student spent on every of his/her solved tasks.

2) User's Satisfaction

Any software tool will succeed if and only if its users are happy with it. It can be expressed by user's opinions collected using a questionnaire. As personal opinions are rather subjective, 9-point Likert scale was applied in order to quantify their feelings on a numerical scale.

C. Reliability of Test and Questionnaire

In the case of didactical test, the value of reliability coefficient is to be 0.83 (83 %) and expresses the ratio between the sum of variability of individual task's score and the total variability of the test. The used estimations (*Cronbach alpha* = 0.83; *Standardized alpha* = 0.85) are satisfactorily close so one can presume the equal variability of the tasks i.e. no suspicious tasks were identified. All tasks were in correlation with the overall score and their removal did not affect the reliability coefficient. Thus, our selected tasks did not decrease the test reliability.

An identical approach was applied the questionnaire with similar results. In particular, the ratio between the sum of variability of values of individual questionnaire's item and the total variability of the questionnaire was 90%.

As the result, both instruments were claimed to be highly reliable.

D. Hypotheses

Three research hypotheses have been formulated:

- The members of the group using the graphic interface QBE will complete their solutions faster than the members of the group using SQL.
- The members of the group using the graphic interface QBE will produce more exact solutions than the members of the group using SQL.
- The members of the group using the graphic interface QBE will be more satisfied with their solutions and the way they were achieved than the members of the group using SQL.

The first two hypotheses were tested using a posttest consisting of the query creation and identical for the both groups. The third one was verified using the questionnaire.

E. *Groups of Query Tasks in Experiment*

The majority of analytical work comparing a query language towards SQL addressed selection queries i.e. the ones beginning with the keyword SELECT [11], [15], [16], [21]. For that reason we also concentrated on the same problem group.

In order to measure the users' cognitive efforts required for their task solutions, we split the tasks into five levels of difficulty. The criteria for the distribution were type of the query, complexity of the conditions the records have to fulfill, and the number of tables the query must include. Their order expressed their complexity starting with the simplest ones:

- *G*1: one table, a single condition and/or sorting of records,
- *G*2: several tables, a single condition,
- *G*3: combination of records from several tables,
- *G*4: aggregation of values stored in one table,
- *G*5: nested queries and subqueries.

The number of queries in the posttest was partially influenced by its presumed duration (90 minutes). For that reason, it consisted of 20 problems written in a natural language. The number coincides with the amounts of tasks mentioned in similar experiments [11], [13].

VI. Results

The results of the experiment were analyzed using linear models.

A. *Verification of the Hypotheses*

The hypotheses were analyzed using the one-way analysis of variance.

Our first hypothesis stated that the students applying QBE-based approach will need shorter time for query formation. The hypothesis has been proved ($p = 0.000383$).

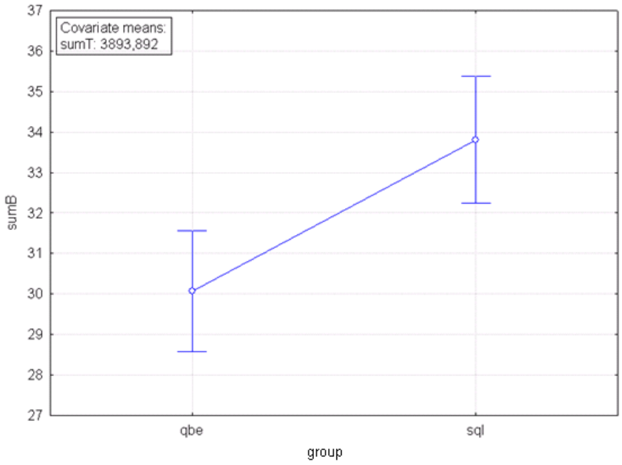

Figure 4. The means with error plot showing the correctness of the solution with respect to the mean time required for its creation.

The second hypothesis stated that the QBE students will create more accurate queries than those using SQL. This hypothesis was not confirmed. The quality of students' queries was almost identical in both groups ($p = 0.516098$).

The third hypothesis presumed that the QBE students will feel better comfort during their query creation and higher satisfaction with their results. This hypothesis has been confirmed ($p = 0.000515$).

B. *More Detailed Results Related to Time Periods Required for Query Creation*

The verification of the first hypothesis has proved that QBE queries have been made faster. To achieve a more detailed picture, we decided to include another factor – the time period necessary for the query completion. The analysis of covariance showed another statistically significant difference. It differed from our expectations. In accordance to it, if the time for the solutions would be identical, the SQL group responses would be more accurate. Fig. 4 shows the estimated means with the respect to the correctness of the query and time.

Based on data in Table 1, the zero hypothesis – the presumption that there is no statistically significant difference between the groups in the precision of the query formation – was rejected at the 99% confidence level. It shows that the relation between the covariate – the variable *sumT* (total time) – and dependent variable *sumB* (total score) – is statistically significant at the 0.01 level.

TABLE I. ANCOVA, UNIVARIATE TESTS OF SIGNIFICANCE FOR THE CORRECTNESS OF THE QUERY CREATION

	SS	***df***	***MS***	***F***	***p***
intercept	22703.16	1	22703.16	718.4995	0.000000
sumT	2217.19	1	2217.19	70.1685	**0.000000**
group	351.54	1	351.54	11.1253	**0.001168**
error	3412.59	108	31.60		

Meaning of the terms: SS = adjusted sum of squares, df = degrees of freedom, MS = adjusted mean square, F = mean square ratio, p = p-value (the probability of the observed data when the null hypothesis is true), intercept = the parameter in an equation, sumT = continuous independent variable (covariate), group = categorical independent variable, error = residual error.

TABLE II. ANCOVA, UNIVARIATE TESTS OF SIGNIFICANCE FOR THE OVERALL SATISFACTION OF THE USER.

	SS	***Df***	***MS***	***F***	***p***
intercept	778.8507	1	778.8507	595.6569	**0.000000**
sumT	24.9029	1	24.9029	19.0455	**0.000028**
group	6.4686	1	6.4686	4.9471	**0.028121**
error	147.7531	113	1.3075		

Meaning of the terms: SS = adjusted sum of squares, df = degrees of freedom, MS = adjusted mean square, F = mean square ratio, p = p-value (the probability of the observed data when the null hypothesis is true), intercept = the parameter in an equation, sumT = continuous independent variable (covariate), group = categorical independent variable, error = residual error

Similarly, the covariance analysis demonstrated a statistically significant difference between the groups even if the time is taken into account (Table 2). The zero hypothesis stating that there is no statistically significant difference between the groups when the time is taken into account could be rejected at the 95% confidence level.

It also shows that the relation between covariate *sumT* a dependent variable *vpS* (overall satisfaction) is statistically significant at the 0.01 level. In this case, the analysis of covariance ($p < 0.05$) confirms the results of the analysis of variance ($p < 0.01$).

C. Comparison of the Groups of Variables

The most interested results have been generated by ANOVA repeated measures. They helped us to show statistically significant differences between combinations of variables (time, accuracy, and satisfaction) and the level of task difficulty. Here, the division of problems into the categories *G*1-*G*5 played its role, too.

Tukey HSD test (Unequal *N*) allows us to analyze the relationship between the categories of tasks separately in the test and control groups. The detail results exceed the presumed size of our contribution. Therefore we prefer visualizing our outcomes in Tables 3-5 and Fig. 5-7.

1) Time Duration

The natural presumption that the more difficult tasks require more time to solve has been proved. Fig. 5 indicates the statistically significant differences between periods necessary for the query creation and selected combinations of time and category of the task. Each line corresponds to one of the groups.

The principle contribution of the next analysis consists in the identification of differences in query creation in the individual groups. In our particular case, the diagonal values are the most interested as they express the comparisons of the time periods. Naturally, we want to know the comparisons of time periods spent on the same tasks.

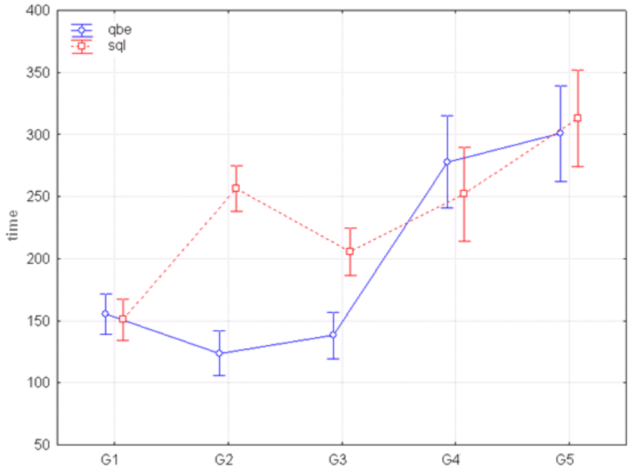

Figure 5. Categorized means with error plot for repeated measures of time periods.

There are larger differences between the groups *G*2 a *G*3 (Table 3): 0.005693 in the former case, 0.611760 in the latter. A statistically significant difference (p<0.01) has been proved in the case *G*2. It implies that the time for creating QBE queries with a single condition in multi-table databases is significantly shorter than that of similar queries in SQL. There was also an observable difference between the *G*3 tasks (aggregated queries) but it was not statistically significant (*G*3).

TABLE III. MULTIPLE COMPARISONS FOR TIME COMPLEXITY

Group (QT)	***SQL(G1)***	***SQL(G2)***	***SQL(G3)***	***SQL(G4)***	***SQL(G5)***
QBE(G1)	1.000000	**0.000032**	0.272150	**0.000073**	**0.000012**
QBE(G2)	0.940429	**0.005693**	**0.001846**	**0.000012**	**0.000012**
QBE(G3)	0.999810	**0.000013**	0.611760	**0.000013**	**0.000012**
QBE(G4)	**0.000012**	0.988066	**0.011454**	0.999009	0.763889
QBE(G5)	**0.000012**	0.445651	**0.000093**	0.303231	0.999998

Meaning of the terms: QT = query type

2) Precision of Solutions

There is a statistically significant difference between the precision of solutions in the student groups. The zero hypothesis stating that there is no significant difference was rejected. We would see the strong relationship between the precision of the solution and the task difficulty – see Fig. 6. The picture indicates a difference between some problem groups – in particular *G*1 and *G*3 – but they are not statistically significant (see Table 4).

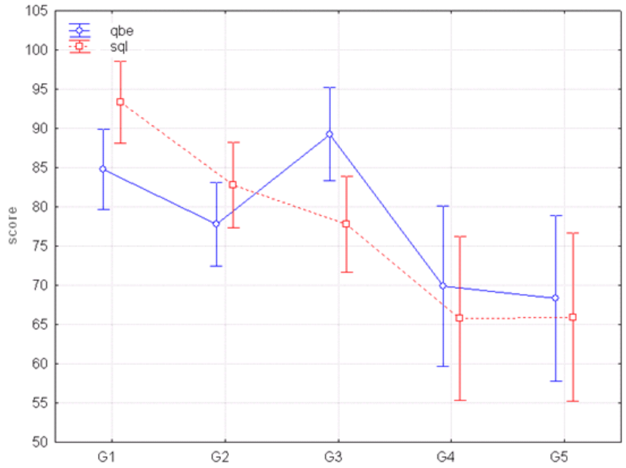

Figure 6. Categorized means with error plot for repeated measures of precision of solutions.

TABLE IV. MULTIPLE COMPARISONS FOR PRECISION OF SOLUTIONS

Group (QT)	***SQL (G1)***	***SQL (G2)***	***SQL (G3)***	***SQL (G4)***	***SQL (G5)***
QBE (G1)	0.992792	0.999999	0.968587	**0.029521**	**0.032071**
QBE (G2)	0.161187	0.999906	1.000000	0.519844	0.538235
QBE (G3)	0.999431	0.981135	0.948587	**0.001530**	**0.001701**
QBE (G4)	**0.001595**	0.411372	0.932978	0.999982	0.999524
QBE (G5)	**0.000509**	0.247468	0.820548	0.999988	1.000000

Meaning of the terms: QT = query type

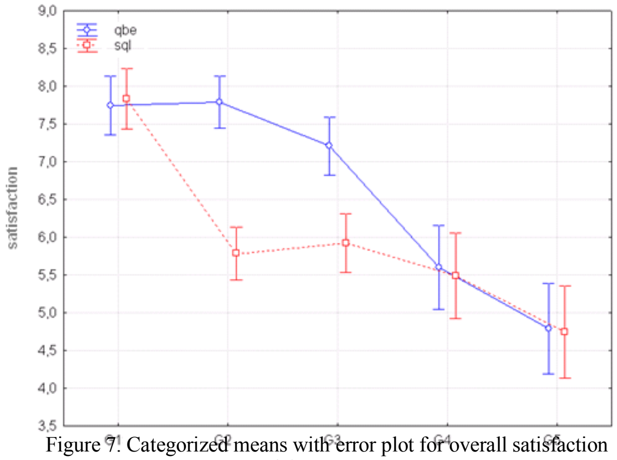

Figure 7. Categorized means with error plot for overall satisfaction of the user.

3) *Overall Satisfaction of Users*

Finally, we focused on the relationship between overall satisfaction of the users and the query type. Overall satisfaction of end-users is decreasing while query complexity is increasing.

One could anticipate this effect – its relevance to the subjective nature of this variable is obvious. If the end-user has to make greater effort to understand the merits of the task and used interface that does not offer sufficiently simple solution, his/her satisfaction would be on decline.

At the same time, the reader of the values on table diagonal comes to the findings that there is a statistically significant difference between overall end-user's satisfaction and different types of queries. There are apparent differences in overall satisfaction of the user in favor of QBE-like graphical user interface in tasks from the groups *G*2 and *G*3 (Fig. 7). A statistically significant difference ($p < 0.05$) has been proved in the case *G*2 (Table 5).

TABLE V. MULTIPLE COMPARISONS FOR OVERALL SATISFACTION OF USER

Group (QT)	***SQL(G1)***	***SQL(G2)***	***SQL(G3)***	***SQL(G4)***	***SQL(G5)***
QBE(G1)	1.000000	**0.000013**	**0.000015**	**0.000012**	**0.000012**
QBE(G2)	1.000000	**0.015005**	**0.000014**	**0.000012**	**0.000012**
QBE(G3)	0.697302	**0.001091**	0.380372	**0.000028**	**0.000012**
QBE(G4)	**0.000012**	0.999945	0.994973	1.000000	0.255619
QBE(G5)	**0.000012**	0.095280	**0.028524**	0.547970	1.000000
			Meaning of the terms: QT = query type		

VII. DISCUSSION

We have prepared our paper as our contribution to the research of interaction between database end-users and relational database systems. In particular, we have been interested in comparing two interfaces designed for query formulation and to the problem of the future end-users training.

In order to identify the differences, we used one-way analysis of variance and analysis of covariance for solving of our research problem. One-way analysis of variance is simpler and it does not require the fulfillment of assumptions about regression in individual groups. On the other hand, if there are some differences in controlled variable between groups, its interpretation is less valid. Therefore it is recommended to realize both analyses and compare their results.

The results of experiment could have been affected by several factors. We must mention the process of participants' selection first of all. Our participants were students of the preliminary database courses. For sake of completeness of our statement, we have to add that the characteristics of students may differ in some details from typical end-user in non-academic environments. On the other hand, our approach is supported by several scientific papers from this research field [13], [15], 19], and [22] that applied a similar approximation of the term "end-user".

The second factor is the number of the participants in experimental and controlled group. Although it may seem sufficient, it is evident that we would become more general results if we could repeat experiment with greater number of participants and/or more than once time. We could not afford such numbers considering (a) the size of our institution (b) regular changes in the syllabus that exclude prolongation of the identical content over several consecutive years.

The different complexity of the tasks is another factor that might have effect on the experimental results. From the very beginning, we understood that tasks leading to subqueries may affect the overall reliability of some results of our test. They are much more complex than other four types of tasks. Their extreme complexity could therefore affect the overall end-user's satisfaction. As they are very frequent in day-to-day situations, we decided to include them to the test despite the above reservation.

We cannot forget in this discussion the limited duration of individual tasks as well as entire testing procedure. Although we took sufficient free float for each task, the clock displayed on the screen could have affected participant's activity and disturbed his/her overall satisfaction.

VIII. CONCLUSIONS AND FUTURE WORK

It has been experimentally confirmed that our methodology of query formulation brings several positive outcomes. These findings could act as an impulse for further considerations about database querying and the ways of their teaching and learning.

The main positive results is proving that the QBE graphic interface allows faster and more comfortable writing of low and medium difficulty tasks compared to the SQL text-based environment does. In these cases, the accuracy of produced queries remains comparable and the comfort of QBE users is higher. One can presume that the students will build their stronger attitude to the tool that allow them faster and more comfortable work i.e. they will tend to use it even in the cases when they would try to avoid SQL.

These results may have a positive influence on non-professionals' exploitation of databases:

a) Due to their quick progress in query creation, they might become higher motivated in further improvement of their knowledge and, consequently, in their more effective and efficient exploitation of their databases.

b) Design of some (very complex) queries via QBE is difficult, if not impossible. The non-professionals should be encouraged to recognize the limits of their expertise and ask ICT professional for their assistance.

Despite the fact that not all of our hypotheses have been confirmed, many of intuitively presumed advantages of QBE have been correct. This should motivate educators to design and develop a combined methodology that would benefit from the positives of both: QBE and SQL.

The research of database query languages and especially visual query languages for end-users has an interdisciplinary character that should teachers take into account in their training as well as pedagogical research. To inspire them, we identified some interesting open questions that might be included into future research:

- What are main trends in end-user database languages? In particular, what perspective do visual languages for query formation have? To what degree can end-users successfully tackle their tasks without being familiar with the internal structure of databases?
- To what degree the interfaces based on natural languages will change the situation? What will be their advantages and drawbacks compared to the form-oriented languages?
- Do contemporary languages oriented to the conceptual design of databases have a potential that would enhance end-user participation?
- Is it possible to integrate advanced features of distinct query languages into one interface that would benefit of each of them?
- Is it possible to create an intuitive interface that would allow mining of information from distributed heterogeneous databases? How to protect the users from getting lost in such environment?

References

[1] N. Murray, C. Goble, C. and N. W. Paton, "A Framework for describing visual interfaces to databases," *J. Visual Languages & Computing,* vol. 9, pp. 429-456, 1998.

[2] P. Sawyer and J. A. Mariani, "Database systems: challenges and opportunities for graphical HCI," *J. Interacting with Computers*, vol. 7, pp. 273-303, 1995.

[3] Reisner, P., "Human factors studies of database query languages: A survey and assessment," *ACM Comput. Surv.,* vol. 13, no. 1, pp. 13-31, 1981.

[4] G. Hagert, "What's in a mental model? On conceptual models in reasoning with spatial descriptions," in *Proc. 9th International Joint Conf. Artificial Intelligence*, San Francisco, CA, pp. 274-277, 1985.

[5] J. D. Sterman, "A Skeptic's Guide to Computer Models," in *Managing a Nation: The Microcomputer Software Catalog* Boulder, CO, pp. 209-229, 1991.

[6] T. P. Moran, "Guest editor's introduction: An applied psychology of the user," *ACM Comput. Surv.*, vol. 13, pp. 1-11, 1981.

[7] S. McDaniel, "What's Your Idea of a Mental Model?," *On-line J. Boxes and Arrows*, http://boxesandarrows.com/, Oct., 2009.

[8] S. Mildeova and V. Vojtko, "System Dynamics," *Systémová dynamika*, Oeconomica, Prague, Czech Republic, 2003.

[9] J. Graaumans, "Usability of XML query languages," Ph.D. dissertation, Instituut voor Informatica en Informatiekunde,Utrecht Univ., 2005.

[10] T. Catarci, and G. Santucci, "Are visual query languages easier to use than traditional ones? An experimental proof," in *Proc. HCI'95 10th Conf. People and Computers*, Huddersfield, United Kingdom, 1995.

[11] T. Catarci, T. and G. Santucci, "Diagrammatic vs. textual query languages: A comparative experiment," in *Proc. 3rd IFIP WG Conf. Visual Database Systems 3 (VDB-3)*, Chapman &; Hall, Ltd., 1995.

[12] J. O. Hauglid, "User interfaces for accessing information in digital repositories," Dept. Comp. and Inform. Science. Norwegian University of Science and Technology, 2005.

[13] H. C. Chan, K. K. Wei, and K. L. Siau, "An empirical study on end-users' update performance for different abstraction levels," *Int. J. Hum.-Comput. Stud.,* vol. 41, no. 3, pp. 309-328, 1994.

[14] S. K. Chang, M. F. Costabile, and S. Levialdi, "Modeling users in an adaptive visual interface for database systems," *Visual Languages and Computing,* 1993.

[15] V. Owei and S. B. Navathe, "Enriching the conceptual basis for query formulation through relationship semantics in databases," *Inf. Syst.*, vol. 26, no. 6, pp. 445-475, 2001.

[16] V. Owei, S. B. Navathe, and H. S. Rhee, "An abbreviated concept-based query language and its exploratory evaluation," *J. Syst. Softw.*, vol. 63, no. 1, pp. 45-67, 2002.

[17] B. Shneiderman, "Improving the human factors aspect of database interactions," *ACM Trans. Database Syst.,* vol. 3, no. 4, pp. 417-439, 1978.

[18] Y. Vassiliou and M. Jarke, "Query languages: a taxonomy," in *Proc. NYU Symp. User Interfaces on Human Factors and Interactive Computer Systems*, New York, United States, 1994.

[19] M. Y. M. Yen and R.W. Scamell, "A human factors experimental comparison of SQL and QBE," *IEEE Trans. Softw. Eng*., vol. 19, no. 4, pp. 390-409, 1993.

[20] H. Topi, J. S. Valacich and J. A. Hoffer, " The effects of task complexity and time availability limitations on human performance in database query tasks,", *Int. J. Hum.-Comput. Stud.*, vol. 62, no. 3, pp. 349-379, 2005.

[21] J. E. Bell and L.A. Rowe, "An exploratory study of ad-hoc query languages to databases," in *Proc. the 8th International Conf. Data Engineering,* IEEE Computer Society, 1992.

[22] H. Chan., K. Siau and K. K. Wei, "The effect of data model, system and task characteristics on user query performance: An empirical study," *SIGMIS Database*, vol. 29, pp. 31-49, 1997.