Using Database Backed Web Applications for the Implementation of Interactive Tutorials on WWW

Karl M. Göschka, Eveline Riedling, Wolfgang Radinger, Jürgen Falb
Institute of Computer Technology
Vienna University of Technology
A-1040 Vienna, Austria

Abstract *– Distance education, in most cases now ODL (Open and Distance Learning) has been proven in modern research to be the most efficient way to go in order to enable growing numbers of students to get especially engineering education as soon and as excellent as possible. Creating interactive simulations and laboratories further enhances the efficiency of the learning process significantly. As one possible implementation of these concepts a Web accessible tutorial for interactive database training is presented. The implementation of this laboratory is essentially a database backed Web application with the necessary technical and organizational framework. We also present a qualitative and quantitative comparison between a pure HTML solution and a Java alternative. As first results we observe organizational and educational advantages, e.g. better student lecturer interaction and personalized access to information combined with interactive simulation systems. With this approach to engineering education, which is new for our department at the Vienna University of Technology, we hope to create most efficient learning and teaching environments, fostering creativity and excellence in order to be ready for the already enhanced international competition.*

1 Introduction - Open and Distance Learning

The Vienna University of Technolgy is confronted with many problems, which are quite common nowadays: An increasing ratio between students and teachers, severe budget cuts, increasing demand for new, additional topics and the still remaining obligation of top level research and high quality education. Therefore it is essential for us to investigate new ways in knowledge representation, learning and educational interaction to enhance the efficieny of the whole learning process. According to research results [14], modern learning theories ask for:

- personalized access to information
- interactive simulation systems
- providing material for learning at own speed
- learning alone or in groups (different types of learners)
- more personalized guidance through tutors (according to strength or deficiencies of the learner)

Distance education, in most cases now ODL (Open and Distance Learning) has been proven in modern research to be the most efficient way to go in order to enable growing numbers of students to get especially engineering education as soon and as excellent as possible. Currently, ODL is described as learning environment, student centered, offering course material or modules using modern IT (Information Technology), too. Some scientists [7] even think about the role of IT & T (Information Technology and Telematics) as best tool to deliver educational innovations, provided that a well defined balance between functions as implemented by IT & T and other media is maintained. Creating interactive simulations and laboratories further enhances the efficiency of the learning process significantly, as proven in international research [6]. As Renwick [21] describes, to guarantee these aspects, new paradigms of teaching and learning have to be taken into account. We have to go from fixed structures of teaching to new ways of learning, to flexible responses and personal empowerment.

Considering research on ODL, as described by Gastkemper [7], the personal responsibility of learners is an important factor to ensure a successfull learning process: "...In terms of didactics or educational support this has also led to different levels of freedom as to the way of studying. However, in view of the relatively fixed offering (academic) courses and curricula as mentioned above, such didactics are often based primarily on the structure of content, pre-specified objectives and a (hypothetical) average student." (from [7] p.20).

Keeping these aspects of modern education in mind, we at the Institute of Computer Technology, try to implement open and distance learning and teaching in several lectures and laboratories. Experiments in remote laboratory work are currently done in a more learner controlled way, as will be shown later. Reason for the stricter guidance of learners was the obligation to keep students motivated, to provide immediate feedback to student work, to avoid a high drop out rate and to keep the tutor involvement to a reasonable, manageable level. According to research results of the Open University, UK, user guidance is a significant factor for success. Though mainly contacts by tutors (email, video conferencing) and student group contact were the most important success factors to minimize drop out rates [4], immediate feedback in the learning environment can be seen as critical success factor, too.

The following description of the remote laboratory case at our institute shows one of the successfull ways of implemeting open and distance education in our institution. But first our previous approaches with lecture notes on WWW are shortly described and why they have not been as successfull as the remote laboratory work.

2 Previous steps: Lecture Notes and User Survey

At the beginning of 1997 the traditionally held face-to-face lectures left almost no possibility for lecturer-student interaction and discussion. The same time first pilots started to create hypertext lecture notes. They were seen as a supplement to the traditional paperware with some advantages. The idea was to give the students the possibility to learn some parts of the lecture on their own to save some time during the lecture for discussion. This was the very first approach, to combine traditionally held lectures (sometimes more than 300 students per lecture) with efforts to implement features of educational technology.

Most of our students are studying Computer Technology, a branch of Electrical and Electronical Engineering. They are very interested in experiencing the Internet, every student gets free Internet access at the university, about 40% have also Internet access from their home. Therefore we expected a positive attitude towards Web material.

The Web lecture notes have been tested by a pilot group and afterwards we have made a user survey with the results in figure 1. We can see that the students appreciate the idea of WWW material in general. But they would not learn directly from the Web but rather print out at least parts of it and use it as a reference. What they really wanted were interactive tutorials and on-line testing.

If we compare the hypertext version to the paper material, the main disadvantage of hypertext material is, that reading a computerized version is not as comfortable as reading a paper version. As long as there is no device as easy to handle as paper and capable of displaying on-line information, many people will prefer course material on paper. This can be seen in the questioning of our students, too.

On the other hand the advantages of hypertext are striking. As already known, hypertext material can be established in a much more structured way, leading to a better overview. The reader can always choose to go deeper into a subject or to move on (student centered learning, learning at own speed). Additionally, he can configure the pages according to his personal preferences of fonts or even colors. A further advantage is the possible integration of multi-media into hypertext. In many cases a good animation gives a better explanation than single pictures and words. Other features like a full-text search are impossible in paper versions, making hypertext usefull as reference guide.

These results have lead us to new approaches: If hypertext material should be successfull, it is not enough just to translate a paper version to the Web, but it is necessary to take advantage of the hypertext features as described above. But the main attraction of the Web is interactivity, leading us to the design of an interactive laboratory following the rules of open and distance learning concepts.

3 The Interactive Distance Laboratory

This laboratory accompanies the lecture "Databases on the World Wide Web", which covers three topics: First, design, implementation and maintenance of relational databases: Entity-Relationship-Design (ER), Structured Query Language (SQL) and database programming. Second, the World Wide Web (WWW): Design and structuring of document collections, Hyper Text Markup Language (HTML), Uniform Resource Locators (URLs), Hypertext Transfer Protocol (HTTP), Common Gateway Interface (CGI) and Web server administration. Third connecting databases to the WWW: Troubles with the stateless HTTP, restrictions of the user interface (UI) through HTML, techniques for connections (CGI, API) and integration (Oracle Web Server, Hyper Wave).

Within the laboratory the students have the following tasks: Design of a small database with ER and implementation with SQL. Some queries with SQL. Design of a small document collection including a nice homepage. Connecting the database to the Web, creating some dynamic HTML documents from the database on the fly and doing some update operations in the database using the HTML FORM-interface.

The software used includes an Oracle 7.3.3 Server on WindowsNT 4.0 on an Intel PC or Solaris 2.5 on a Sun SPARC workstation, the Oracle Web Server 2.1 and PL/SQL as server-side programming technique. The client software necessary to pass the laboratory is just a Web-browser.

3.1 The Interactive SQL Tutorial

The first part of the laboratory is an interactive SQL-tutorial: Each group of four students gets an example where 15 queries have to be solved on a given data structure. Figure 2 shows the welcome screen for one of these examples: The data structure is given, including all base relations and a short description.

After reading and understanding the data structure, the student can proceed to the first query task, where also the expected results are shown. Now it is time to enter the first SQL query (figure 3). Because the first try

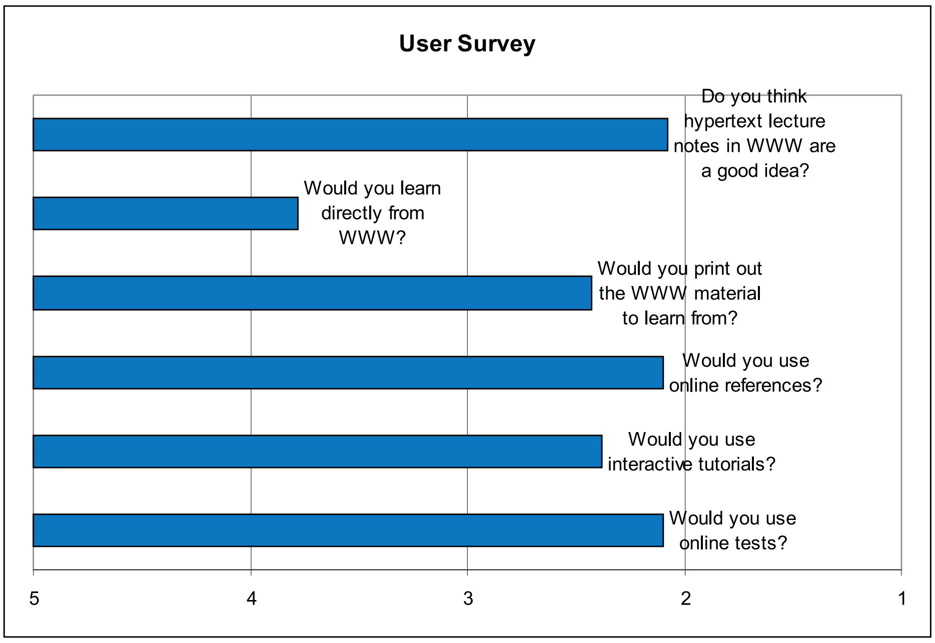
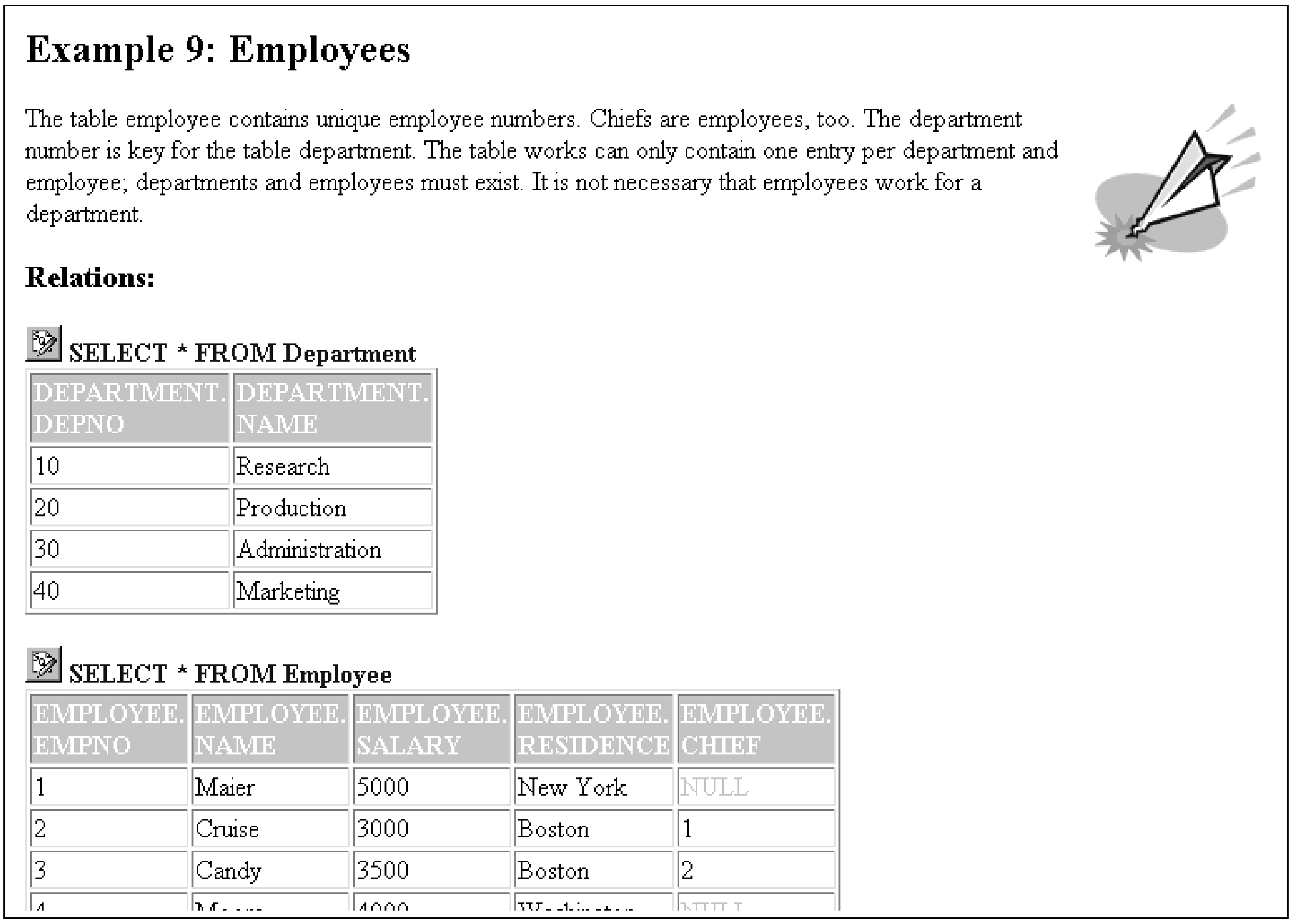


Figure 1: User survey. The evaluation range was from 1 to 5 where 1 is best.



Example 9: Employees

The table employee contains unique employee numbers. Chiefs are employees, too. The department number is key for the table department. The table works can only contain one entry per department and employee; departments and employees must exist. It is not necessary that employees work for a department.

Relations:

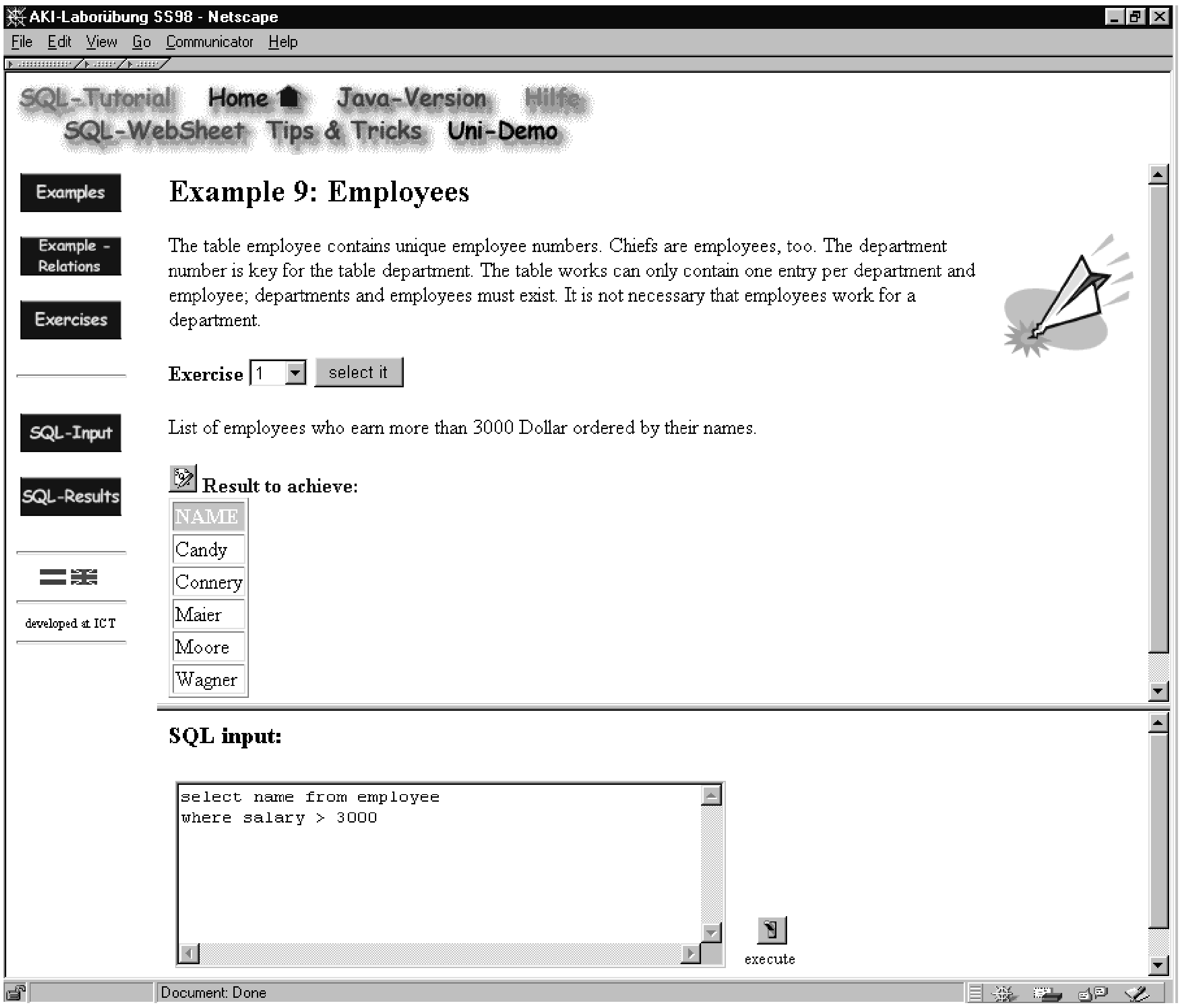
SELECT * FROM Department

DEPARTMENT. DEPNO	DEPARTMENT. NAME
10	Research
20	Production
30	Administration
40	Marketing

SELECT * FROM Employee

EMPLOYEE. EMPNO	EMPLOYEE. NAME	EMPLOYEE. SALARY	EMPLOYEE. RESIDENCE	EMPLOYEE. CHIEF
1	Maier	5000	New York	NULL
2	Cruise	3000	Boston	1
3	Candy	3500	Boston	2
4	[illegible]	[illegible]	[illegible]	[illegible]

Figure 2: At the beginning the data structure is presented and explained.


AKI-Laborübung SS98 - Netscape
File Edit View Go Communicator Help
SQL-Tutorial Home Java-Version Hilfe
SQL-WebSheet Tips & Tricks Uni-Demo
Examples
Example - Relations
Exercises
SQL-Input
SQL-Results
developed at ICT
Exercise 1 select it
execute
Document: Done

Example 9: Employees

The table employee contains unique employee numbers. Chiefs are employees, too. The department number is key for the table department. The table works can only contain one entry per department and employee; departments and employees must exist. It is not necessary that employees work for a department.

List of employees who earn more than 3000 Dollar ordered by their names.

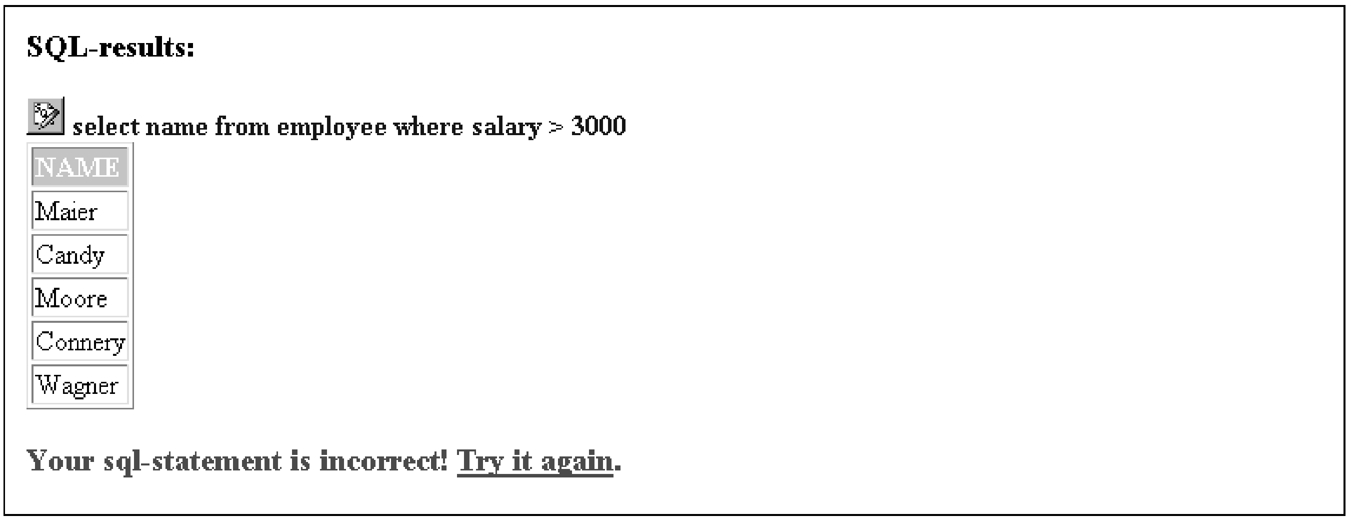
Result to achieve:

NAME
Candy
Connery
Maier
Moore
Wagner

SQL input:

```
select name from employee
where salary > 3000
```

Figure 3: First task and first SQL query.



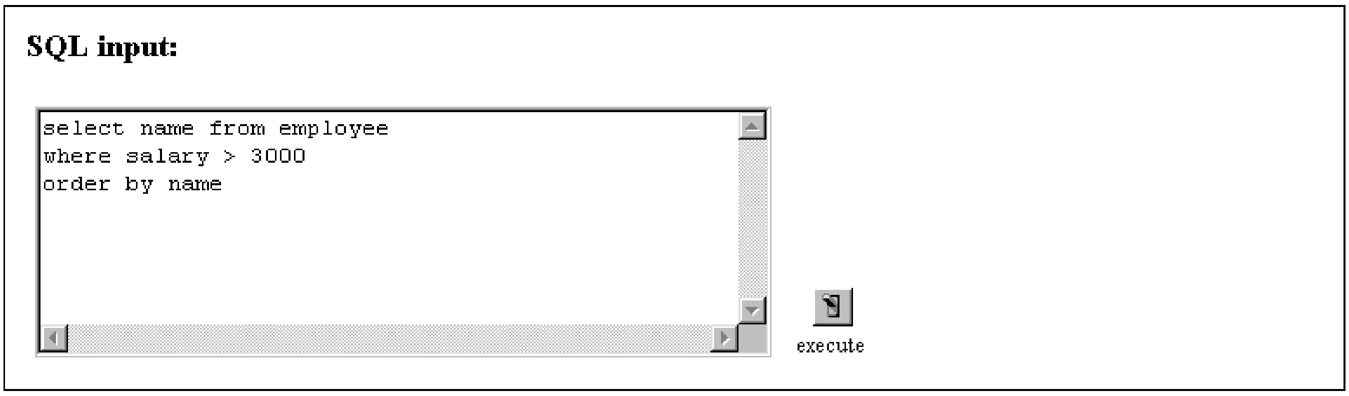
SQL-results:

select name from employee where salary > 3000

NAME
Maier
Candy
Moore
Connery
Wagner

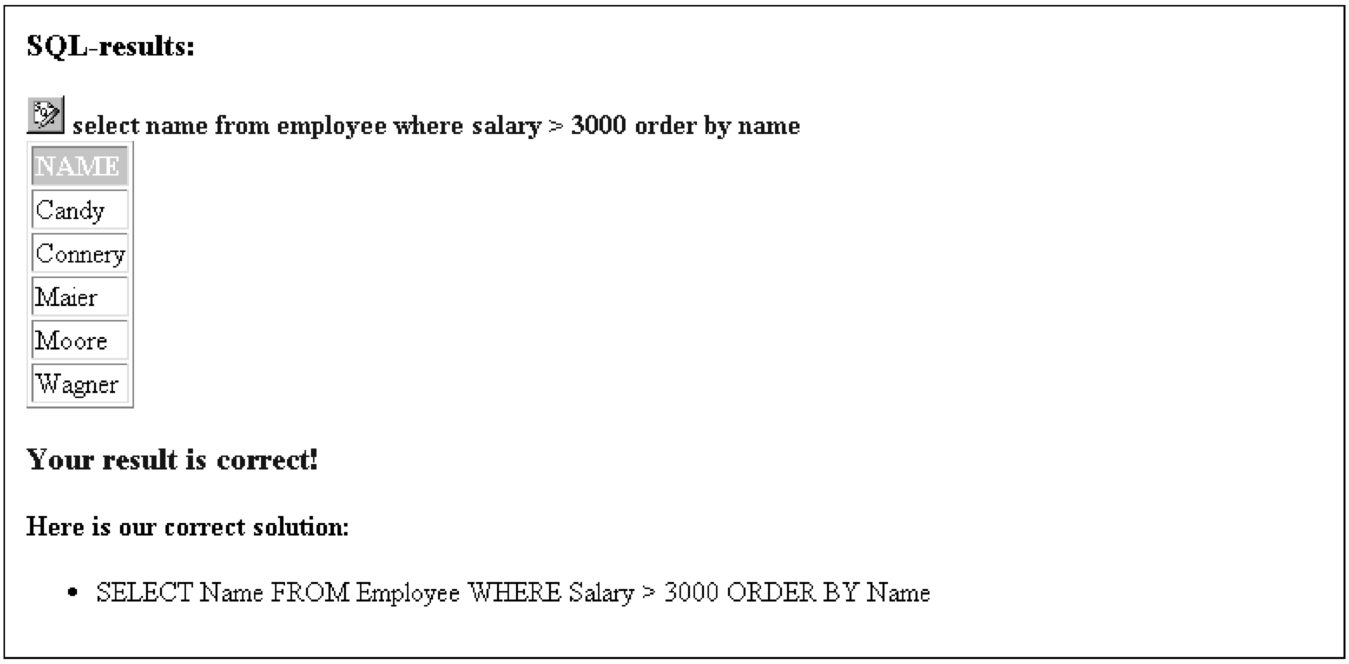
Your sql-statement is incorrect! Try it again.

Figure 4: Negative response to the first SQL query.


SQL input:

```
select name from employee
where salary > 3000
order by name
```

Figure 5: Second SQL query.


SQL-results:

select name from employee where salary > 3000 order by name

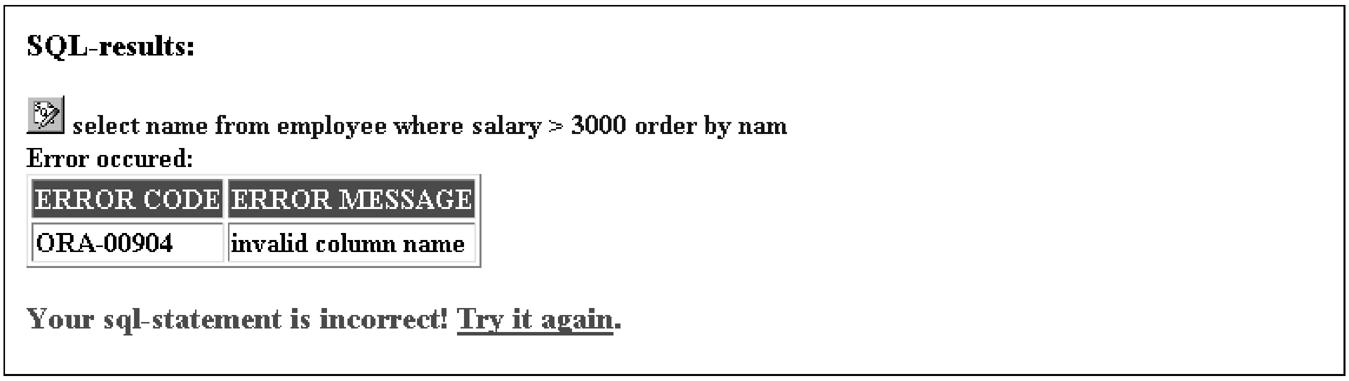
NAME
Candy
Connery
Maier
Moore
Wagner

Your result is correct!

Here is our correct solution:

- SELECT Name FROM Employee WHERE Salary > 3000 ORDER BY Name

Figure 6: Positive response to the second SQL query.


SQL-results:

select name from employee where salary > 3000 order by nam
Error occured:

ERROR CODE	ERROR MESSAGE
ORA-00904	invalid column name

Your sql-statement is incorrect! Try it again.

Figure 7: Error message.

was syntactically correct but not a solution to the given problem, we obtain a response as in figure 4. After entering the correct SQL statement as in figure 5, we receive a confirmation of the correctness (figure 6). If one enters a syntactically incorrect statement, we get an error message from the database as in figure 7. To determine, if the student has entered a correct solution, we do not parse the SQL input but rather compare the result of the student's SQL request with the result of our proven SQL solution statement executed on the same actual data. Figure 6 shows, how we provide alternative solutions afterwards, if the student has already solved the task.

This provides the possibility of misuse, when a student enters a SQL statement which is tailored towards the expected results but only works on the actual data. Our system does determine this statement as correct and provides alternative results as in figure 8. But we save every SQL statement which leads to a correct response into a log file and can later evaluate, if someone tried to cheat.

3.2 Database Backed Web Applications

The second part of the laboratory is to design and implement a small database and to implement database interaction from the HTML user interface, e.g. to insert a form input into the database or to generate a Web page dynamically from the database contents. The implementation is done through PL/SQL programs located in the database. They are accessed through the Web Request Broker, an Oracle specific interface between the Web server and the database. This interface is CGI compliant but much faster than the original CGI interface implementation.

To complete the second laboratory part another Web interface is provided, which allows the execution of any SQL statement to create and modify the data structure, the editing of the student's database contents, the upload and compilation of the PL/SQL code and the upload of static HTML pages and pictures. Again, the only thing needed at the client side is a Web browser.

The complete laboratory is available at `http://aki.ict.tuwien.ac.at/` in both English and German. Only the second part of the laboratory is restricted to authorized access, the first part and the demo application are available without restriction and everybody is invited to have a look at them.

3.3 Organizational Framework

The real outstanding feature of this laboratory is, that there is no need to be present at the institute, which is absolutely unusual at the Vienna University of Technology in these days. For traditional reasons there are laboratory hours, when the PC room of the institute is reserved for the laboratory, because the other user rooms at the university are sometimes overcrowded and not every student has Internet access from home. Advanced students (tutors) give assistance mainly through a mailing list, once a week there are office hours for personal contact. Thus the lab can be passed from everywhere on the Internet, with just a Web browser at the client side. For the students this means the possibility to learn at their own speed, to learn when and where they want (student centered learning).

For the lecturer this approach saves time and gives the possibility to provide a much better examination at the end of the laboratory. When formerly examination was done by a written test or an anonymous protocol delivery, where the students only got their ratings and no real feedback on their work, it is now possible to give each group of four students 90 minutes of a personal discussion on their work, where it is mandatory to be present. This in turn has significantly increased the average rating of the final examination of the whole lecture.

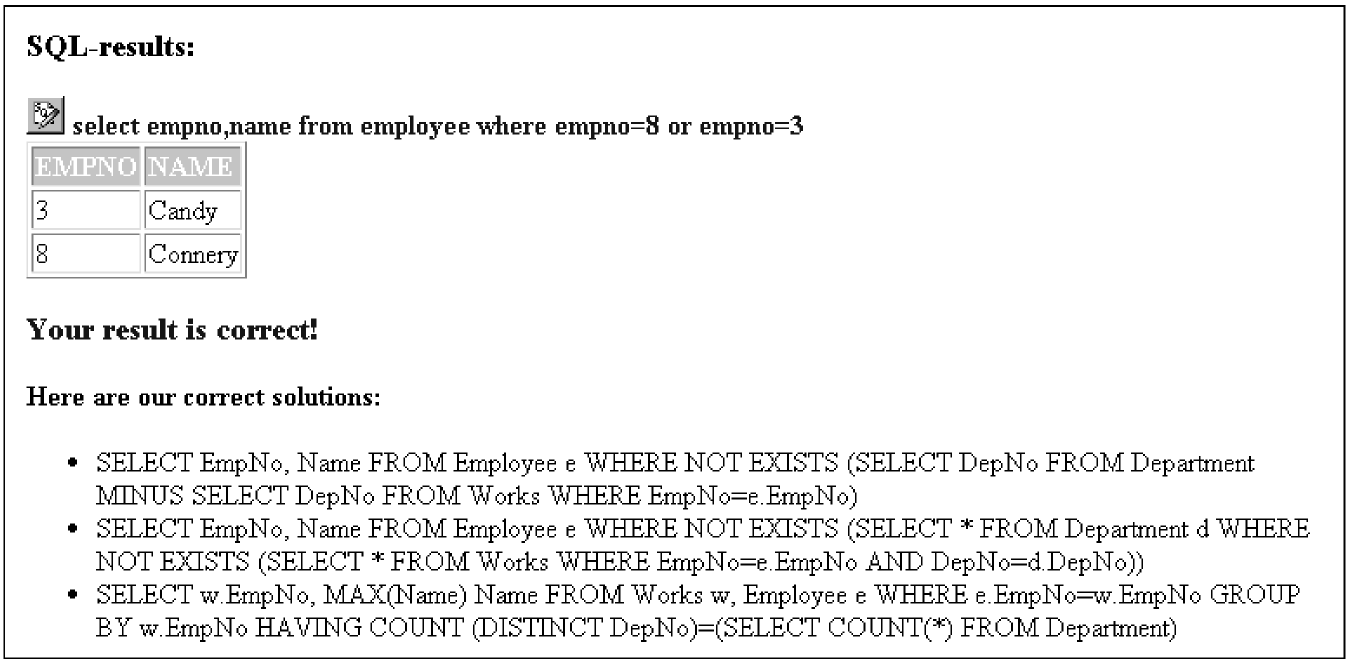
This large amount of time each student gets is just another very unusual feature at TU Vienna, where too few lecturers are in charge of too many students, normally leaving very few time of communication with each student.

3.4 Technical Implementation: PL/SQL and Java

The laboratory itself is implemented as database backed Web application, using PL/SQL programs to generate the Web interface pages automatically from the database contents and to compute the user input data. The only difference from the students' implementation is, that we have to use dynamic embedded SQL rather than static embedded SQL, because we have to pass the students' SQL statements to the database and then include the results in dynamically generated HTML pages. Thus the meta-level of tutorial implementation uses essentially the same technologies as the students have to use when completing the tutorial.

The standard user interface of our tutorial has to work with nearly any browser and platform and should not depend on proprietary nonstandard extensions like Plug–Ins or Active–X. Thus our user interface is restricted to the possibilities of pure HTML. This concerns the representation, but even more the user interaction: the only functional elements are following a link, pressing a submit button of a form or clicking on an image or image-map. They always result in an HTTP (Hypertext Transfer Protocol) connection. Ergo, any functionality resides at the server side, implemented with PL/SQL.

Another problem arises due to the different nature of databases and the Web: HTTP is stateless [11] but connection orientation is inevitable for real–life database transactions. In our approach, we use short URL (Uniform Resource Locator) encoding for state maintenance: the complete session state information is kept in the database. With every HTTP connection, a short session handle has to be passed back and forth between client and server. This technique allows more than one concurrent session. To avoid misuse of the session handle, we use encoded handles. As Netscape's Cookies [12] become standardized [RFC 2109 [22]] we can also use short Cookies, but we do not depend on them.



SQL-results:

select empno,name from employee where empno=8 or empno=3

EMPNO	NAME
3	Candy
8	Connery

Your result is correct!

Here are our correct solutions:

- SELECT EmpNo, Name FROM Employee e WHERE NOT EXISTS (SELECT DepNo FROM Department MINUS SELECT DepNo FROM Works WHERE EmpNo=e.EmpNo)
- SELECT EmpNo, Name FROM Employee e WHERE NOT EXISTS (SELECT * FROM Department d WHERE NOT EXISTS (SELECT * FROM Works WHERE EmpNo=e.EmpNo AND DepNo=d.DepNo))
- SELECT w.EmpNo, MAX(Name) Name FROM Works w, Employee e WHERE e.EmpNo=w.EmpNo GROUP BY w.EmpNo HAVING COUNT (DISTINCT DepNo)=(SELECT COUNT(*) FROM Department)

Figure 8: The possibility to cheat is overcome by a logging mechanism.

In our implementation, a pure CGI compliant interface between Web server and database server would be sufficient. But for a sequence of HTTP connections, CGI is rather ineffective. This is due to the overhead of spawning a new process each time which, in turn, opens and closes the connection to the database. This performance loss is normally overcome by two approaches [2]. A proprietary approach we do *not* use is to link server programs directly with the Web server [NSAPI [19], ISAPI [17]]. Another approach is to prefork multiple processes, which the Web server communicates with and which stay connected to the database. This is done with FastCGI [1, 16] by Open Market and the Web Request Broker of the Web Application Server by Oracle [20]. Both approaches improve server performance, but only FastCGI is a non–proprietary standard with easy migration from CGI. The Oracle WRB is at least CGI compatible.

Using these techniques we have implemented our SQL tutorial with PL/SQL procedures at the server side: Every statement except `SELECT` and `DESCRIBE` is directly passed to the database and the results are passed back to the client. If a `DESCRIBE` statement is recognized, the data dictionary is consulted and structural information about the table or view (attributes and constraints) is delivered to the client. In the case of a `SELECT` statement, we analyze the statement to build an appropriate result structure. Then we pass the statement to the database, catch the result into our result structure and build a HTML table to show the resulting table to the client in a convenient way. If the input statement was syntactically wrong, the database error is shown instead of a result. We also execute our correct SQL statement to determine, if the student's SQL statement is correct.

The Web interface for the second part of the laboratory is implemented almost the same way, but there are some enhancements for a comfortable upload of static HTML pages and images into the static Web structure in the server's file system including the possibility to create and delete subfolders. Some of these parts are implemented with Perl scripts. Other parts of the user interface allow the upload and compilation of PL/SQL procedures, in this case the result is either "ok" or a list of errors arising during complilation. Also the error log file of the server can easily be displayed in the Web interface. Thus anything needed to implement and debug a database backed Web application and static Web pages including images on a Web server is provided through a single easy-to-use Web interface.

As an alternative implementation technique we have provided the same functionality for the two parts of the laboratory with two Java applets (also available on the Web). We can summarize the following differences between the Java and the pure HTML implementation:

User Interface The user interface design with Java is more powerful but also more complicated than with pure HTML. The usage of Java frames and dialogues gives us the possibility to design a user interface with menues, buttons, etc. like any other standard user interface. With HTML the possibilities of the user interface are more restricted – the user interface lacks the possibility of constraints and event handling: An HTTP session has to be performed before functional dependencies or restrictions can affect other inputs. To achieve this HTTP session, a submit button has to be pressed explicitly. It is usually not enough to fill an input field or make a selection. Thus the user interface cannot react directly on user input.

Performance and Load The Java applet has to be downloaded first. Then the applet loads all the data that remains unchanged during the whole excercise from the database. Thus the usage of Java results in a longer latency at the beginning but afterwards takes some load off the server and the network compared to the pure HTML implementation, where a lot of database transactions have to be performed on every single user interaction. We will compare network load, server load, overall performance and user satisfaction of both approaches throughout the laboratory.

File Upload The upload of files from the local file system to the server is not possible with Java due to the restrictive security model. Therefore we had to build a combination of a server-side Java listener-application and CGI perl-scripts to integrate the usage of a small HTML frame with a form upload into our Java applet in an almost homogenous way.

Browser Compatibility: Can easily be achieved with pure HTML. With Java browser compatibility has to be tested carefully but is still possible.

Multi Media Integration: Images and other Multimedia content can be integrated in both approaches: The advantage of Java is: There is no need for a proprietary installation of PlugIns. Moreover Java objects are available for playing almost any multimedia format available on the Internet.

Database Transactions: With pure HTML (even with cookies) the connection to the database is stateless, resulting in complex state handling and timeout mechanisms, which also compromises the security of such a system. With JDBC or RMI connection orientation can easily be achieved thus guaranteeing safe transactions.

Server events and collaboration: For collaboration it is necessary that the server can trigger an event on the client side. With HTML Netscape has developed some extensions (server push, channels) to implement active servers, but these are proprietary and not widely used. On the other hand with Java servlets an active server can be implemented (remote method invocation over sockets without HTTP).

4 Results and Future Work

The feedback of the students was overwhelmingly positive: The vast majority of the students (97%) liked the possibility to work at the laboratory anytime and anywhere, only 3% felt insecure and would have prefered a more restrictive course. Almost the same percentage (95%) appreciated the final discussion instead of having a written examination or an anonymous protocol delivery. Although getting a worse rate for making many mistakes, they felt it was a good chance to recognize their deficiencies. This resulted in better ratings at the final examination of the whole lecture.

The possibility to work on the laboratory remote from the Internet has some advantages: It saves time, resources and organizational effort, leaving more time for a very intensive and personal feedback discussion at the end of the work. On top of that, on-line assistance is done by email during the whole course to give answers to problems arising during work.

By providing interactive tutorials and references on the Web, only some very difficult parts of the lecture have to be held in the traditional way, face-to-face, whereas many parts of easier but necessary theory can be given by the Web leaving more time to use the lecture more intensively for questions and discussion with the students. So the university could migrate back from an unpersonal mass-university to a place, where students can discuss problems and solutions with their lecturers in a more personal way.

Now we plan to change the educational system for our institute significantly. Whereas currently most lecturers use the traditional way and only few are even thinking about using the possibilities of these new technologies, the changes will be inevitable. From our experiences we are sure, that the mentioned problems will be overcome by time, when other lecturers recognize, that there are possibilities to make even the best traditional lecture better by sensible use of the new technologies.

In addition we plan to start national and international cooperation on modern approaches to engineering education especially with Finland, Norway and Spain. These environments will be created using Web pages and Java applets for interactive and non-interactive simulations and multimedia applications, but we will also provide text versions on theoretical parts for download. For remote student groups we establish mailing lists and video conferencing, mainly desktop conferencing with remote computer centers, and more remote laboratories. With this approach to engineering education, which is new for our department at the Vienna University of Technology, we hope to create most efficient learning and teaching environments, fostering creativity and excellence in order to be ready for the already enhanced international competition.

References

[1] Adida, B.: *It all starts at the server.* IEEE Internet Computing, Vol. 1, Nr. 1, pp. 75-77, 1997.

[2] Adida, B.: *Taking Web clients to the next level.* IEEE Internet Computing, Vol. 1, Nr. 2, pp. 65-67, 1997.

[3] Catarci, T.: *Interacting with Databases in the Global Information Infrastructure.* IEEE Communications Magazine, Vol. 35, Nr. 5, pp. 72-76, 1997.

[4] Davis, N.: *Telematics in Education: The UK Case.* In: Veen, W.; Collis, B.; et al.: Telematics in Edu-

cation: The European Case, Academic Book Centre, ABC, de Lier, the Netherlands, 1995.

[5] Engels, G.; Gogolla, M.; Hohenstein, U.; Hülsmann, K.; Löhr-Richter, P.; Saake, G.; Ehrich, H-D.: *Conceptual modeling of database applications using an extended ER model.* Data&Knowledge Engineering, North Holland, Part 9, Nr. 2, pp. 157-204, 1992.

[6] Field, M.; Weedon, R.: *Professional training in computing: The UK Open University·s Computing for Commerce and Industry Programme.* In: Open and Distance Learning - Critical Success Factors, International Conference, Geneva, 10-12 Oct. 1994, Proceedings, Berne 1995.

[7] Gastkemper, F.: *Pedagogy.* In: Open and Distance Learning - Critical Success Factors, International Conference, Geneva, 10-12 Oct.1994, Proceedings, Berne 1995.

[8] Göschka, K.: *Design and implementation of database powered web systems – experiences from the DEMETER project.* Proceedings of the IFIP WG2.4 'Systems Implementation 2000 Conference', Berlin, Germany, Feb 23-26, 1998, to be published by Chapman & Hall, 1998.

[9] Graham, I.: *The HTML Sourcebook.* Wiley & Sons, Inc., New York, 3rd edition 1997.

[10] Hirohido, H.: *Experimental Analysis of User's Behaviour in Hypermedia CAI Systems.* In: Liberating the Learner, WCCE 95. Proceedings, Birmingham 1995.

[11] Iyengar, A.: *Dynamic Argument Embedding: Preserving State on the World Wide Web.* IEEE Internet Computing, Vol. 1, Nr.2, pp. 50-56 1997.

[12] Kristol, D.; Montulli, L.: *HTTP State Management Mechanism.* draft-ietf-http-state-man-mec-02.ps on [23], 1997.

[13] Riedling, E.; Göschka, K.; Manninger, M.: *Education at the Vienna University of Technology: Traditional Lecture Based Education vs. Telematics Based Education.* Proceedings (ISBN 1-885189-03-6) of the 'International Conference on Engineering Education: Progress Through Partnerships', pp.717-725, Chicago, Illinois, USA, August 13-15, 1997.

[14] Riedling, E.: *Computerunters"utztes Lernen - Analyse der enthusiastischen Bef"urwortung von Unterrichtsprogrammen.* Dissertation. TU Wien, 1989.

[15] Stürner, G.: *Oracle 7. Die verteilte semantische Datenbank. Release 7.3.* dbms publishing, Weissach, 4.Auflage, 1996.

[16] `http://www.fastcgi.com` Open Market, FastCGI.

[17] `http://www.microsoft.com/msdn/sdk/platforms doc/sdk/internet/src/ isapimrg.html` Microsoft Internet Server API.

[18] `http://java.sun.com` The Java Language.

[19] `http://www.netscape.com/newsref/std/ server_api.html` Netscape Server API.

[20] `http://www.oracle.com` Oracle Web Application Server and Web request Broker.

[21] Renwick, W. L.: *Organisational Strategies.* Open and Distance Learning - Critical Success Factors, International Conference, Geneva, 10-12 Oct.1994, Proceedings, Berne 1995.

[22] `ftp://ftp.univie.ac.at/netinfo/rfc` Requests for Comment (RFCs): 822, 977, 1036, 1319, 1345, 1521, 1522, 1630, 1738, 1766, 1808, 1866 und 1867.

[23] `http://www.w3.org`, World Wide Web Consortium, Internet Engineering Task Force.