

2000

# WebSQL: An Interactive Web Tool for Teaching Structured Query Language

Grove N. Allen

*University of Minnesota*, allen088@umn.edu

Follow this and additional works at: <http://aisel.aisnet.org/amcis2000>

---

## Recommended Citation

Allen, Grove N., "WebSQL: An Interactive Web Tool for Teaching Structured Query Language" (2000). *AMCIS 2000 Proceedings*. 384.

<http://aisel.aisnet.org/amcis2000/384>

This material is brought to you by the Americas Conference on Information Systems (AMCIS) at AIS Electronic Library (AISEL). It has been accepted for inclusion in AMCIS 2000 Proceedings by an authorized administrator of AIS Electronic Library (AISEL). For more information, please contact [elibrary@aisnet.org](mailto:elibrary@aisnet.org).

# WebSQL: An Interactive Web Tool for Teaching Structured Query Language

Gove N. Allen, Department of Information and Decision Sciences  
University of Minnesota, allen088@umn.edu

## Abstract

WebSQL is an interactive Internet tool for teaching Structured Query Language (SQL). It can be accessed from a Web Server running at the University of Minnesota or downloaded and installed on a Web Server at any accredited educational institution. It facilitates the learning of SQL by providing an interface to major database management systems for query execution on the web. It allows students to learn with large, rich databases, without the usual problems associated with installing client/server database systems in a lab on a traditional campus. The accompanying interactive SQL tutorial makes use of these advantages in by allowing students to invoke the WebSQL query interface and dynamically modify and execute any example query.

## Introduction

WebSQL is an interactive Internet tool for teaching Structured Query Language (SQL). It can be accessed from a Web Server running at the University of Minnesota or downloaded and installed on a Web Server at any accredited educational institution. The installation at the University of Minnesota provides real time access to a number of different databases implemented in a number of different database management systems (DBMSs), including Oracle 8, MS SQL Server 7, Interbase 5, and MS Access 2000. WebSQL has several advantages over LAN-based or standalone alternatives for teaching SQL. These include the ability to install the DBMS and a number of alternative databases on machines directly accessible to the web server and use "thin clients" to access them, the ability to log query attempts and analyze errors, and the ability to track the amount of time taken to successfully compose various types of queries. This paper presents WebSQL, discusses its advantages in teaching SQL and introduces the specific implementation available at <http://www.WebSQL.org>. WebSQL also logs the queries made by students giving instructors the ability to examine the process by which their students are solving homework problems. This allows instructors to identify common problems students are making as well as to determine how long students are spending completing assignments.

## Interactive SQL

Virtually all relational database management systems (RDBMSs) provide a mechanism to interactively execute SQL statements against a database. In Oracle the tool is

called "SQL\*PLUS", in MS SQL Server it is called the "Query Analyzer," in Interbase it is called "ISQL." These enable a connection to a specified database and present a "blank screen" in which SQL statements can be entered and executed. Query results are typically displayed in a separate window. Often queries and query results can be saved to a log.

## Web-Based vs. LAN-Based SQL

In a LAN-based environment, the DBMS must be installed on one machine and the interactive SQL tools must be installed and configured on each machine that will be used to access it. Currently this is often achieved by having a particular lab configured with the interactive SQL tools. However, in today's climate where many students are part-time, labs are overcrowded, and most students own computers that are comparable to (or superior to) lab equipment, it is inconvenient at best and infeasible at worst to use a LAN-based environment to teach SQL.

One of the biggest benefits of using a Web-based tool for teaching SQL is that the client machine requires no software beyond a browser. This means that students can access a database and execute SQL queries from any machine that can connect to the Internet. This benefit is particularly important for distributed learning programs, where students may never come to a traditional campus. Although most RDBMSs provide an interactive SQL tool that is capable of making a remote connection over the Internet, as mentioned above, these must be licensed and installed on each client, requiring significant time and expertise.

## Web-Based vs. Standalone SQL

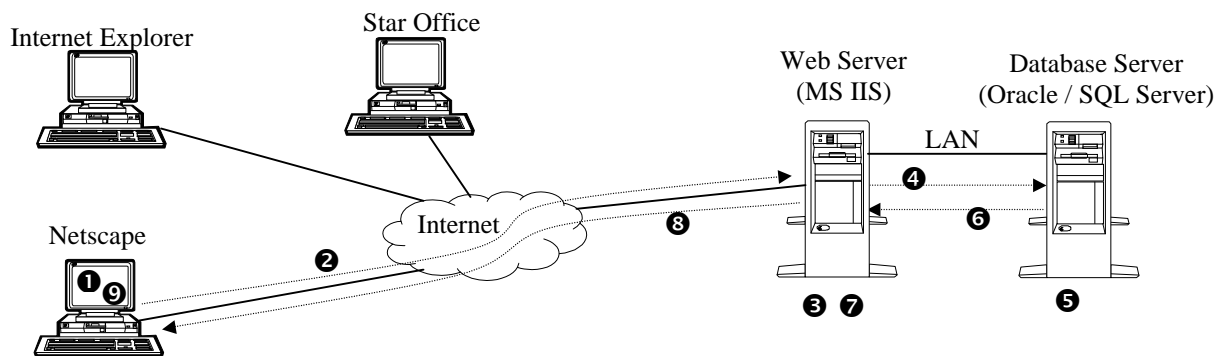
The problem of installing remote connection tools can be solved by eliminating the client/server model altogether and using local installations of the database engine. This is one of the reasons MS Access is a popular tool for teaching SQL in spite of its relatively non-standard and awkward SQL implementation. Products such as Personal Oracle and Local Interbase have also been used for this reason. While this approach solves the problems surrounding remote connection, it brings other problems to the surface. In the standalone model, students must install an entire database package simply to execute SQL statements against a database. Complicating matters, in the case of MS Access, the DBMS is often installed along with other software or is pre-installed (with undesirable settings) by the computer manufacturer.

Further, when a student relies on a local installation of a database engine, the student must also rely on local data and local hardware. This requires the database to be distributed to each student and causes several problems. First, students will frequently have different, incompatible versions, of the DBMS. The "least common denominator" must be used. More importantly, it is infeasible to use databases of more than a few megabytes due to the logistical issues of physical distribution. Few experiences help students understand the power of today's RDBMSs like seeing the results of a query that processes hundreds of thousands to millions of rows in fractions of a second.

Thus a Web-based tool for executing SQL statements allows students to get right to the business of learning SQL with the benefit of large and rich databases, without the need for lengthy and frustrating local installations. Furthermore, they can complete their exercises from virtually anywhere in the world.

### Interactive SQL Tutorials

The value of using interactive tutorials for teaching SQL is clear. Feedback is fundamental to the learning process (Brehmer and Allard, 1991; Senge 1992, Sterman 1987). Making query results available to students studying SQL increases the level of feedback they receive as they are forming their earliest perceptions of Structured Query Language. With a Web-based, interactive SQL tool, creating an interactive tutorial is a simple matter.



- 1 A user with a browser requests a query to be executed.
- 2 The browser sends an HTTP request (including the query) to the web server.
- 3 The web server begins building the HTML response to the HTTP request.
- 4 The web server requests the database server to execute the query.
- 5 The database server executes the query.
- 6 The database server sends the result set of the query to the web server.
- 7 The web server uses the query results to finish building the HTML response.
- 8 The web server sends the HTML response to the browser that requested it.
- 9 The browser renders and displays the HTML document containing the results of the query.

Figure 1. The process used by WebSQL to access databases.

Figure 2: Database Connection

Links can be embedded in the tutorial that directly connect to the tool and enable students to execute example queries with the click of a mouse.

Moreover, confirmation bias, the propensity to affirm prior interpretations and to ignore, or reinterpret evidence that runs counter to an already-formed interpretation, can hinder the learning process (Hutchins, 1995). By providing an environment that facilitates students' efforts to find out what does not work, in addition to what does work, students can easily test assumptions made about syntax—before those assumptions are summarily accepted or expelled by the confirmation bias. One of the main problems with virtually all static tutorials is that they only show what works. One of the most powerful techniques in learning is experiencing that which does not work (Sitkin, 1992). An interactive SQL tutorial allows students to experience what does not work by allowing any of the example queries to be modified and re-executed.

## WebSQL

WebSQL is an interactive environment for executing SQL statements against a DBMS and for the displaying of the results via the Internet. It was developed at the University of Minnesota by the author for instructional use. Built using Microsoft's Active Server Pages (ASP) and Active Data Objects (ADO), WebSQL accepts an SQL statement from a browser as a part of an HTTP request, executes the statement, builds an HTML file with the results of the query, and then sends that file back to the browser for display. Figure 1 shows the steps WebSQL takes to process a query and return its result set.

Because the web server generates a static HTML document in response to the browser's query request, WebSQL will work with virtually any browser and from behind most firewalls. WebSQL returns a specified number of records to the browser (the default is 10) and allows the user to retrieve the next part of the result set if desired. This makes WebSQL very well suited for use by students who access the Internet through dial-up connections. The user controls the amount of data that will be transferred.

There are two steps necessary to access a database using WebSQL, connecting to the desired database and specifying the query. Students connect to the desired database by selecting from a list of available databases and supplying the proper username and password for the selected database. The WebSQL login page is shown in Figure 2. Once a connection is established, a blank screen is displayed in which the user can specify a query as seen in figure 3. Since WebSQL passes the query on to the connected database, the only constraints placed on which queries can be executed are those imposed by the database engine itself. Upon execution of the query, the specified number of record from the response set is returned to the browser and the page is rendered and displayed as illustrated in figure 4.

Once a query has executed successfully, users can download the result set in one of several delimited formats by following the "Save Results" link as seen at the bottom of figure 4. This allows students to import the results of their query into another program for further analysis and presentation (a feature not implemented by most interactive SQL tools).

#	BALANCE	ACCOUNTNO
1.	14323.34	1
2.	5033.78	2
3.	105.21	3
4.	9566.91	4
5.	56.22	5
6.	887.39	6
7.	300	7
8.	4824.29	8
9.	5846.25	9
10.	67455	10

Figure 4: Query Results

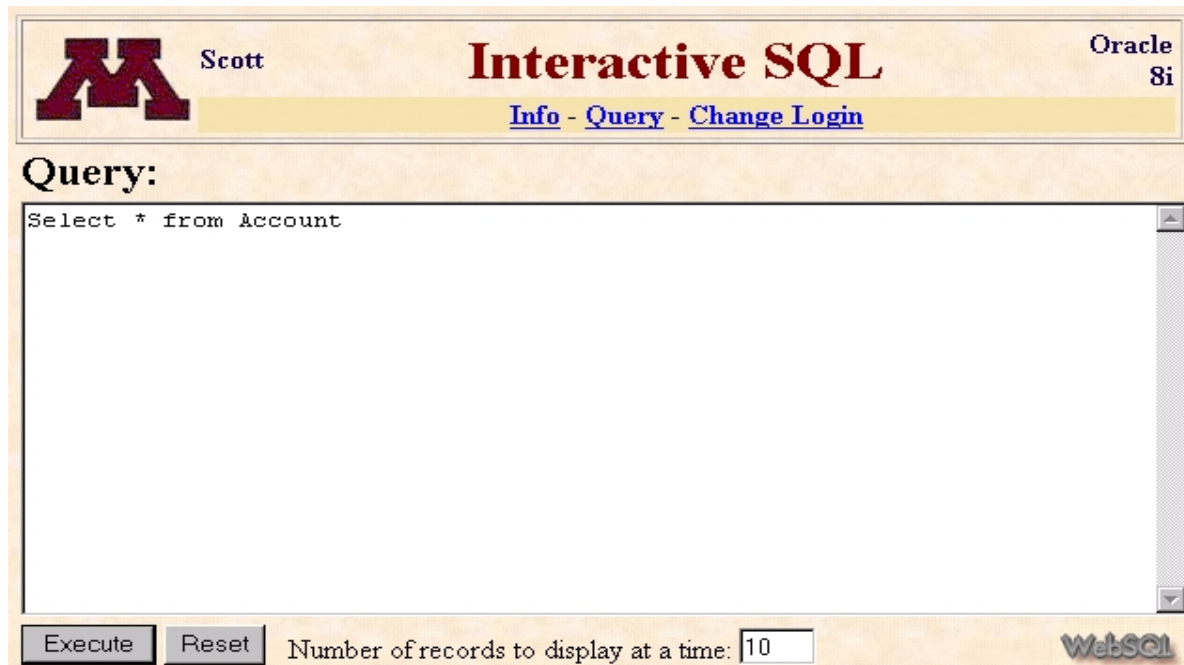


Figure 3: Query Specification

At the University of Minnesota, we provide several databases for students to explore as they familiarize themselves with SQL (and for professors to use as the basis of homework assignments). Three will be highlighted here. The first is a recipe database that has information on over 800 recipes, their ingredients and their nutritional content (including protein, carbohydrates, sodium and fat). The database is implemented in Oracle 8. The second is the history of all automobile crash tests conducted by the National Highway Traffic Safety Administration (NHTSA). Also implemented in Oracle, this database has information on more than 2,500 crash tests conducted over four decades including information on occupant positioning and injury severity. The third is also a database from the NHTSA. Implemented in MS SQL Server 7, this database includes more than 500,000

complaints filed about automobiles and other transportation related products. Also in this database are records of the almost 9,000 manufacturer recalls that have been issued since 1966. This database is significant because of its size—it demonstrates the power of today’s relational database management systems by executing complex queries involving hundreds of thousands of joins and comparisons virtually instantly.

Any of these databases can be selected on the WebSQL Login screen. Information about the database table structure and contents can be obtained by following the “Info” link, which can be seen in figures 3 and 4. Additionally, most RDBMSs allow users to execute meta-data queries to get more specific information about the tables available. For example, Oracle provides the “CAT” view which displays the catalog of tables

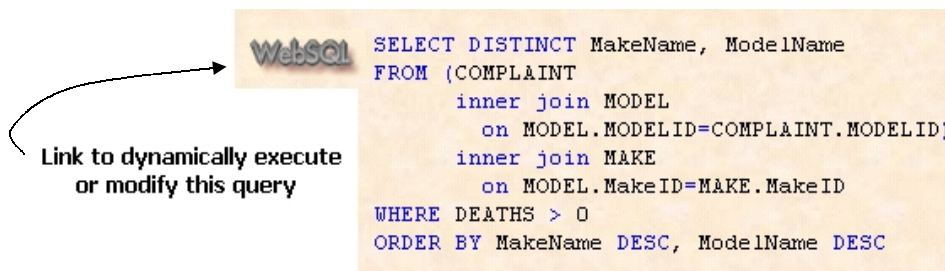
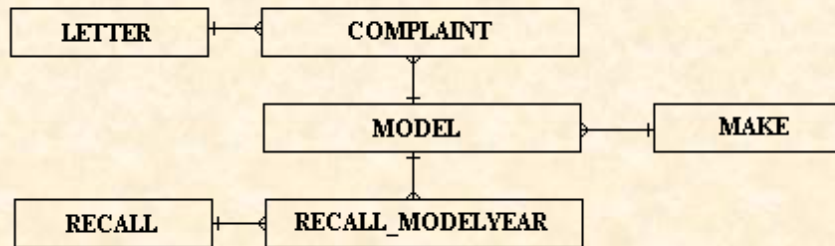


Figure 5: Tutorial with dynamic query execution

Here is an entity-relationship diagram of the data used in this tutorial:



Here are some details about the fields in each table:

LETTER	
Field Name	Description
LETTERID	Unique identifier
LNAME	First name of the person who filed the complaint (fictitious)
FNAME	Last name of the person who filed the complaint (fictitious)
CITY	City where filer resides
STATE	State where filer resides
LETTERDATE	Date on letter (or date complaint filed)
VIN	Vehicle Identification Number
SOURCE	Source of complaint
DATEENTERED	Date entered into system

**Figure 6: Tutorial data model and partial table description**

available to a particular user by issuing the statement “Select \* from CAT.” In the case of MS SQL Server, the meta-data are less accessible, so we have provided views such as “TABLES” and “FIELDS” which allow users access to similar information.

### WebSQL’s Interactive Tutorial

WebSQL contains an interactive SQL tutorial. It allows students to instantly modify and execute any of the queries used in the tutorial. Of course, additional queries can be composed from scratch. Although the tutorial can be printed and used effectively as a static document, the web-version allows students to test their knowledge of SQL as they go. Each example query in the tutorial is accompanied by a link that dynamically executes the query using WebSQL as seen in figure 5. Students then are free to follow the “Query” link to modify and re-execute the query. This feature allows students to execute queries as they study SQL from any computer connected to the Internet, without the need to install any software.

This tutorial uses the NHTSA complaint and recall database, and is described with an entity-relationship

model and descriptions of the fields in the each table. A portion of the supporting information is displayed in Figure 6. This database adds strength to the tutorial in three ways. First it is a large database. Students can see the power of a database management system as queries involving hundreds of thousands of records are executed in fractions of a second. Second, the data are real—not contrived for example purposes. Hence when students execute a query, the results should “make sense.” If not, they will not assume that the results are odd because the data are made-up, but more likely that they did not properly formulate the query. Third, the database contains interesting data. Students who use this tutorial often execute additional queries because they are genuinely interested in the answers. They are curious to see if there have been any recalls issued for cars that they or their parents own. They are also interested to see what complaints have been made against their vehicles. (The author was surprised to learn that his vehicle was under recall for nine different problems—for only one of which did he receive notification.) Students executing queries because they are genuinely interested in the answer to the query may experience a different level of motivation than



```

WebSQL SELECT FName, LName
FROM Letter

```

The first ten records of the result set are as follows:

FName	LName
LYNN	AABY
ROLF	AADLAND
UTE	AADLAND
STUART	AADNES
CLAUDIA	AAKER
DONALD	AAKER
KRISTI	AALBERG
MARTHA	AALID
MARTHA	AALID
MICHAEL	AALID

```

WebSQL SELECT COMPDESC,
SUM(Potential) AS Total,
MAX(Potential) AS Largest,
AVG(Potential) AS Average,
MIN(Potential) AS Smallest,
Count(*) AS NumberOfRecalls
FROM Recall
WHERE COMPDESC LIKE 'BRAKE%'
GROUP BY COMPDESC
HAVING SUM(Potential) > 2000000
ORDER BY COMPDESC

```

Here is the sorted result set:

COMPDESC	Total	Largest	Average	Smallest	NumberOfRecalls
BRAKESHYDRAULIC.ANTI-SKID SYSTEM	4276395	2400000	122182	5	35
BRAKESHYDRAULIC.DISC.ROTOR.DISC HUB	3917271	600000	195863	92	20
BRAKESHYDRAULIC.LINES.HOSE.NON-METALLIC	3657949	830000	26128	5	140
BRAKESHYDRAULIC.LINES.METALLIC	2169360	661585	26136	8	83
BRAKESHYDRAULIC.POWER ASSIST.BOOSTER	2070043	1218387	64688	21	32

```

WebSQL SELECT FName, LName, City, ModelName, MakeName, CompDesc, Csummary
FROM ((LETTER
inner join COMPLAINT
on LETTER.LETTERID=COMPLAINT.LETTERID)
inner join MODEL
on MODEL.MODELID=COMPLAINT.MODELID)
inner join MAKE
on MODEL.MakeID=MAKE.MakeID
WHERE FName = 'MARK'
AND LName = 'LUCAS'
AND City = 'CINCINNATI'

```

Here is the final result set:

Fname	Lname	City	ModelName	MakeName	CompDesc	Csummary
MARK	LUCAS	CINCINNATI	ALLIANCE	CHRYSLER	POWER TRAIN.TRANSMISSION:5 SPEED AND UP	A WHINE DEVELOPED WHILE DRIVING IN 5TH GEAR. AFTER DRIVING A DISTANCE AT 50-55MPH THE TRANSMISSION JUMPED OUT OF 5TH GEAR. THE 5TH GEAR WAS REBUILT. VC

```

WebSQL SELECT MODELID, COMPDESC, CSUMMARY
FROM COMPLAINT
WHERE ModelID NOT IN(
SELECT ModelID
FROM RECALL_MODELYEAR
)

```

Here is the partial result set:

MODELID	COMPDESC	CSUMMARY
10928	EQUIPMENT.CHILD SEAT	RENOLUX TURN-A-TOT CHILD SAFETY SEAT RETAINER CLIP CAN BE RELEASED WITH MINIMUM PRESSURE. AK
8638	EQUIPMENT.JACKS.RAMP	PETERSON INDUSTRIES CAR RAMP COLLAPSED
2715	TIRES	FAILURE OF BF GOODRICH SPACE SAVER TIRE DUE TO BROKEN BEAD AT TIME OF INFLATION

**Figure 7: Example queries with results from WebSQL’s interactive tutorial**

students who execute queries simply to achieve the correct answer for a homework assignment. Using this NHTSA database in the SQL tutorial is an attempt to leverage this dynamic.

The tutorial take users from the very simplest queries to the queries involving multiple aggregate functions with “where”, “group by”, “having”, and “order by” clauses, to queries involving multiple joins to queries with uncorrelated sub-queries. These are all described in the context of extracting meaningful data from the database. Examples of these kinds of queries are shown in Figure 7.

### WebSQL’s Availability

WebSQL is available at no charge to all accredited institutions of higher education for their educational and research purposes. Professors may download the software (including a SQL ’92 complaint, client/server database engine) and run the system at their university, or they may use the software directly from our site at the University of Minnesota. In the latter case, professors are given an administrative account that allows them to create

individual accounts for their students. WebSQL can be accessed via the Internet at <http://www.websql.org>.

### References

Brehmer, B and Allard, R. Dynamic Decision Making: “The Effects of Task Complexitiy and Feedback Delay.” In Jens Rasmussen, Berndt Brehmer and Jacques Leplat (eds.), *Distributed Decison Making; Cognitive Models for Cooperative Work*. Baffins Lane, Chichester, West Sussex, 1991.

Hutchins, E. *Cognition in the Wild*. MIT Press; Cambridge, MA, 1995.

Senge, P. “Systems Thinking and Organizational Learngning: Acting locally and Thinking Globally in the Organization of the Future.,” *European Journal of Operations Research*, (59), 1992, pp.137-150.

Sitkin, S. “Learning Through Failure: The Strategy of Small Losses,” *Research in Organization Behavior*, (14), 1992, pp. 231-266.

Serman, J. D. “Testing Behavioral Simulation Models by Direct Experiment,” *Management Science*, (33:12) December 1987, pp. 1572-1592.