Cloudscape

Using HTML Tables as Virtual Tables in Cloudscape

A Cloudscape White Paper

Jeffrey Lichtman

November 4, 1998

Contents

Synopsis

This paper describes the benefits of using HTML tables as virtual tables with Cloudscape. It describes how to do this, and includes Java source code for this purpose.

Introduction

A few years ago, the emergence of the internet changed the way that people use computers. As a result, there is now a tremendous amount of data embedded in HTML web pages. Many of these pages are available through the internet, and many are on corporate intranets.

Unfortunately, all one can do with most HTML pages is display them in browsers. Browsers have limited functionality – mostly, they format and display text and graphics. They allow users some limited control over some display options (such as fonts, background colors, etc.), but they do not provide true data processing capabilities. For example, browsers do not allow users to summarize data from HTML pages, or to sort the data in different ways. Browsers do not provide a means for users to write programs to use the data found in HTML pages. And in many cases, the data on an HTML page may not be available to you in any other form.

One construct that lends itself to data processing is the HTML table. The structure of HTML tables is similar to that of tables in relational databases: they both have rows and columns, and usually HTML tables have the same number of columns in each row. Some HTML tables have captions, which are similar to database table names, and some HTML tables have headings (similar to database column names) at the top of each column. Each column usually contains the same type of data for each row (for example, a table that describes products will usually not put prices in the same column with product names).

The regularity of HTML tables, and their similarity to database tables, makes it natural to use them for traditional data processing tasks like reporting, sorting, and summarizing. It would be useful to correlate data from one HTML table with data from another HTML table, or even with data that is stored somewhere else (for example, in a Cloudscape database).

Starting in version 1.5, Cloudscape provides a means to solve these problems, through a feature called the Virtual Table Interface. This interface allows users to make data from external sources directly available to Cloudscape databases. The following sections of this paper describe a Virtual Table Interface (VTI) that gets Cloudscape database tables from tables in HTML pages.

A VTI For HTML Tables

This document is associated with three source files: HTMLTableVTI.java, HTMLTableMetaData.java, and HTML.jj.

HTMLTableVTI.java defines a class that implements the java.sql.ResultSet interface. This is the class that can be used within Cloudscape to get data from an HTML page.

HTMLTableMetaData.java defines a class that implements the java.sql.ResultSetMetaData interface. The ResultSet interface has a getMetaData() method that returns an HTMLTableMetaData object. Cloudscape uses this method to determine the properties of the virtual table.

HTML.jj defines a parser, for use with the JavaCC parser generator. JavaCC is a freeware tool provided by Metamata, Inc. The tool is downloadable from their web site, at http://www.metamata.com. The HTML.jj grammar is derived from a public-domain HTML grammar graciously made available by Kimbo B. Mundy (see http://www.tiac.net/users/kimbo/jack/HTML.jack).

If you download and compile these files, you will have a VTI to get HTML tables and make them available as Cloudscape database tables. You can use any Java compiler with the Java files. For the HTML.jj file, you will need to download the JavaCC parser generator tool from Metamata, install it, and run it. NOTE: The parser generator produces a couple of warning messages, which you can ignore. JavaCC produces a set of Java files, which you can compile using any Java compiler.

To use the HTMLTableVTI class as a virtual table, you will need to use one of its constructors. It has two public constructors:

```
HTMLTableVTI(String url, Integer tableNumber, Boolean headings)

HTMLTableVTI(String url, String tableName, Boolean headings)
```

The first constructor finds a table in a page by ordinal position. Given the URL for an HTML page and a table number, it looks for Nth table in that page. For example, if you pass the number 5 as the tableNumber parameter, it will look for the 5th table in the page.

The second constructor finds a table in a page by name. The name of a table in a page is expected to come after the <table> tag and before the first row of the table, between the <caption> and </caption> tags. Only the textual parts of the caption are included in the table name – tags and other HTML directives are not considered part of the name.

The headings parameter of both constructors tells whether the first row in the HTML table contains column headings or data. If true, the VTI will treat the values in the first row as the names of the columns. If false, it will invent its own names for the columns, and will treat the first row as data. Only the textual parts of the columns will be used for column names.

The datatype of all columns returned by the VTI is LONG VARCHAR. This is because the data in HTML tables is all textual, with no predefined length limit. If you know that the data in one column is all numeric, and you want to treat the values as numbers, you can cast the values to the appropriate number type within the query language. Only the textual parts of the column values will be returned – tags and other HTML directives are not considered part of the column values.

Using the HTMLTableVTI Class

The HTMLTableVTI class can be used from within Cloudscape by putting a call to a constructor in the FROM clause of a SELECT statement (for this to work, the .class files must be within your CLASSPATH). Here are some examples, using a page on Cloudscape's web site that contains three different sample tables. Note that the virtual tables work across the internet. Please look at the example page using a browser before trying these examples. You can try the examples using ij, or Visual JBMS, or any other interface that connects to a Cloudscape database:

```
-- Look at the contents of each table
select *
from new HTMLTableVTI(
            'http://www.cloudscape.com/example.html',
            1,
            true) P;

select *
from new HTMLTableVTI(
            'http://www.cloudscape.com/example.html',
            'Sales',
            true) S;

select *
from new HTMLTableVTI(
            'http://www.cloudscape.com/example.html',
            'Salespeople',
            true) SP;

-- Do a three-way join between the three HTML tables to get the
-- total sales for each salesperson
select SP."Salesperson ID",
       SP."Name",
       sum(cast (S."Quantity" as int) *
           cast (P."Unit Price" as decimal(5,2))) as TotalSales
from new HTMLTableVTI(
            'http://www.cloudscape.com/example.html',
            'Products',
            true) P,
     new HTMLTableVTI(
            'http://www.cloudscape.com/example.html',
            'Sales',
            true) S,
     new HTMLTableVTI(
            'http://www.cloudscape.com/example.html',
            'Salespeople',
            true) SP
where P."Product ID" = S."Product ID"
and   S."Salesperson ID" = SP."Salesperson ID"
group by SP."Salesperson ID", SP."Name";
```

If you run these four queries using ij, you will see results like this:

```
Product ID     |Product Name   |Unit Price
-----------------------------------------------
```

```
WI328xab7      |Widget Type 1  |17.95
TH765lfy5      |Thingy         |5.47
CT834lyf2      |Contraption    |14.98
WH086hgi8      |Whatsis        |2.15
MA754ljf4      |Machine        |29.95
ST575kjg0      |Stuff          |0.15
JU976iuy4      |Junk           |1.00
WI785hkv2      |Widget Type 2  |19.25
OE764jhv1      |Odds and Ends  |12.79
9 rows selected


Sale ID        |Salesperson ID |Product ID     |Quantity
---------------------------------------------------------------
8615387502     |7810762        |ST575kjg0      |79
9717643851     |0876493        |CT834lyf2      |12
0975081641     |0876493        |WI328xab7      |7
1438476812     |8774578        |CT834lyf2      |5
0756967433     |8774578        |WH086hgi8      |9
0956348651     |0876493        |MA754ljf4      |8
6 rows selected

Salesperson ID |Name
-------------------------------
8774578        |Billy Murray
7810762        |Arthur Fields
0876493        |Annette Hanshaw
3 rows selected

Salesperson ID |Name           |TOTALSALES
-----------------------------------------------
0876493        |Annette Hanshaw|545.01
7810762        |Arthur Fields  |11.85
8774578        |Billy Murray   |94.25
3 rows selected
```

You can also use the HTMLTableVTI class to combine data from web pages with data stored in Cloudscape databases. You can sort data from HTML tables. You can use your favorite report writer to format HTML tables in ways that web browsers can't. You can use a JDBC-ODBC bridge to make HTML tables available to non-Java databases. You can write applications that analyze data from HTML tables (including tables from web pages that are on the internet, or that were put on your intranet by organizations other than yours).

Once you have tried the HTMLTableVTI class with Cloudscape's example tables, we invite you to try it with your own examples. The HTML tables can be in any page, including pages that are not owned or controlled by you.

Summary

In summary, the Cloudscape Virtual Table Interface, combined with the HTMLTableVTI class, allows you to treat data in HTML tables as if it were data in a Cloudscape database, with all of Cloudscape's power and flexibility.

With its combination of robust SQL features, support for Java, and embeddable, pure Java architecture, Cloudscape is the data management product of choice for data-driven Java applications.