Creating Advance HTML Form Based on Metadata for Web Based Database Applications

Kyawt Kyawt Htay, Khin May Win
Loikaw Computer University
kyawtkyawthtay@gmail.com, winn.km05@gmail.com

Abstract

In this paper, intends to enable the rapid deployment of existing databases on the Web and expanding the potential user audience of the databases. The relational database metadata is applied to generate dynamically from the internal database format to a markup language format such as HTML form. The Meta - information can use to create better HTML forms. That can reduce the load on a DB Server. The HTML form is not a static .html file. It is not dependent on web browser. It is dynamically generate by the system. Therefore, the HTML form is advance. In this approach relational database integrity constraints and referential constraints, can use to validate the data values accepted into the relational database tables. This system is -intended for the implementation and -maintenance of various database backend web applications will be easier, faster and less error prone. The system is a very general method for addressing databases and also includes Metadata features.

Keywords: Relational Database, Database - Metadata, Integrity Constraints, Referential Constraints and HTML Form

1. Introduction

Most modern systems are either web based, within the organization (Intranet) or public. In the - latter case, especially, metadata is utilized to improve communication between hetero geneous information systems for the purposes of obtaining and providing information, for - communication between the user client work - station and the information servers and for electronic business between information - systems. The integration of the Web - and database technology will bring many - opportunities for creating advanced information management applications [1].

The internet is not designed for database application development, and while it has a role in making information stored in databases widely available to millions of people, it is likely (certainly for the next 3-5 years) to be largely restricted to simple database applications. Taking simple data from a database and placing it on the web is a relatively simple task. However, in most cases, corporate data is maintained in a variety of sources, including legacy, relational, and object databases. It is much more complicated when these diverse data-sources must query or updated [1].The increasing popularity and advancement of web technology, many organizations want to modify existing host-based applications [2].This paper is intend to simplify the implementation of integrated Web-DBMS sites. In the Database research that the web changes every thing [3], little effort has devoted to adapt data design methods to the use of the web as the fundamental data interface. This paper is applied relational database theory to develop "safe" and easy to maintain HTML forms based on metadata extracted from system catalogue tables in relational database. In this paper, relational database design method is use to represent the data model.

2. Related Work

One example of building a web interface design utilizing information (Metadata) about a database is the Surveillance, Epidemiology and End-Results (SEER) database. Its purpose was to develop a web-based database interface.

Method: Metadata table design method - describes the database structure by including the four essential elements required to represent a data model: table names, field names, field data types, and linkages among the tables. The link definition requires two fields to specify both the linked table and linked field within the table [4]. The metadata table also stores descriptive information that helps the user understand the meaning and appropriate use of each data field.

3. Relational Database Design Method

In the database development process, the physical database design and logical database design are contained.

In the physical database design process, the system is free to store the data any way it likes-using sequential files, indexing, hashing, pointer chains, compression, etc-provided only that it can map that stored representation to tables at the logical level.

In the logical database design process, this is a process of transformations, the conceptual data model into a logical database model. There are four types of logical database model in use today. There are object-oriented data model, hierarchical data model, network data model and relational data model. The system uses the relational data model.

3.1. Relational Model

The system uses relational data model, which represents data in the form of table or relation. The relational data model is very simple and elegant. The database is a collection of one or more relation, where each relation is a table with rows and columns. The relational model can be regard as having three principle parts, data structure, data integrity and manipulation.

3.2. Data Structure

Data are organizing in the form of tables or relations. Each relation consists of a set of named columns and an arbitrary named of unnamed rows. Each column in a relation corresponds to an attribute of that relation. Each row of a relation corresponds to a record that contains data values for an entity [5].

3.3. Properties of Relations

They are as follows:
- There are no duplicate tuples
- Tuples are unordered, top to bottom
- Attributes are unordered, left to right
- Each tuple contains exactly one value.

3.4. Information Principle or Metadata

The information principle is a very nice principle of the relational database system. The entire information content of the database is representing in one and only in a way, namely as explicit values on column positions in rows in tables. This method of representation is only the method available at the logical level in a -relational system. There is no pointer connecting one table to another [5].

3.5. Classification of Metadata

Metadata is use for several purposes:

- Metadata for the purposes of global -access, from query to optimize recall and relevance.
- Metadata for the purposes of query optimization,
- Metadata, for the purposes of answer integration and explanation,
- Metadata for the purposes of correct analytical processing or interpretation,
- Metadata to overcome multi-linguality and multimedia heterogeneities,

Metadata is essential for -understanding the structure of information, its quality and its relevance. Metadata is essential in explaining answers from ever more complex information systems [6].

3.6. Logical Metadata Table Design

Table1: shows database metadata interface -design for Blood-Requisition table.

Table-Name	Column-Name	Type-Name	Data-Type	Col-Size	Position	Is-Nullable
Blood-requisition	Patient-Id	Int4	4	4	1	No
Blood-requisition	Patient-Name	bpchar	1	12	2	No
Blood-requisition	Ward-Regd-No	bpchar	1	7	4	No
Blood-requisition	Require-Bottle	Int4	4	5	5	No
Blood-requisition	Blood-Gp	bpchar	1	7	6	No

Table2: show the metadata table interface-design for Blood-Requisition table base on primary key and foreign key relationship.

PK-TableName	PKColumn-Name	FKTable-Name	FKColumn-Name	Update-Rule	Delete-Rule
Blood-Gp	Blood-Group	Blood-requisition	Blood-Group	3	3
Ward	Ward-Regd-No	Blood-requisition	Ward-Regd-No	3	3
Blood-requisition	Patient-Id	Transfusion-Patient	Patient-Id	3	3
Donor-Issued	Indent-No	Transfusion-Patient	Indent-No	3	3
Ward	Ward-Regd-No	Transfusion-Patient	Ward-Regd-No	3	3

3.7. Conceptual Schema Definition

Example 1: Data definition for the tables Blood Requisition and Wards in SQL (e.g. Conceptual Schema)

```
CREATE TABLE Wards
(Ward_no CHAR (5) NOT NULL,
Name CHAR (20) NOT NULL,
PRIMARY KEY (Ward_no));

CREATE TABLE Blood-Requisition
(Patient_id CHAR (5) NOT NULL,
Patient-name CHAR (20) NOT NULL,
Date DATE NOT NULL,
Ward_no CHAR (5) NOT NULL,
Requirebtl Int4 NOT NULL,
Bloodgp CHAR (7) NOT NULL,
PRIMARY KEY (Patient_id),
FOREIGN KEY by (Ward_no) REFERENCES
Wards
ON DELETE SET NULL);
```

3.8. Physical Database Design

Example 2: Index definition for the tables Blood Requisition and Wards in SQL

```
CREATE UNIQUE INDEX Wards_ix
ON Wards (Ward_no ASC);
CREATE UNIQUE INDEX Bloodrequisi_ix
ON Blood_requisi (Patient_id ASC);
```

Table3: Shows if the user Elbibas has already created the database Blood_Bank the system catalog table Database contains (among others) the above entry.

Dbname	Creator
Blood_Bank	Elbibas

Table4: Shows an entry is adds to the system catalog table for new create table.

Tbname	Type	DBname	Creator
Wards	Table	Blood_Bank	Elbibas

Table5: Shows an entry is adds to the system catalog table columns for new create table.

Colname	Tbname	Colno	Coltype
Ward_no	Wards	1	CHAR
Name	Wards	2	CHAR

Table6: Shows the primary key declaration and the index creation cause, for the table name.

Ixname	Tbname	Uniquerule
Wards_ix	Wards	P

Table7: Shows the primary key declaration and the index creation cause, for the table's column name.

Ixname	Colname	Colno	Colseq	Ordering
Wards_ix	Ward_no	1	1	A

Table8: Shows the Blood_Requisition table, the-field Ward_no is a foreign key referencing entries in the table Wards.

Tbname	Relname	Reftbname	Colcount	Delete Rule
Blood_Requi	By	Wards	1	C

Table9: Shows to keep track of the columns' -name taking part in the reference table. In this case, there is only one corresponding row.

Tbname	Relname	Colname	Colno	Colseq
Blood_Requisi	By	Ward_no	5	1

4. System Implementation

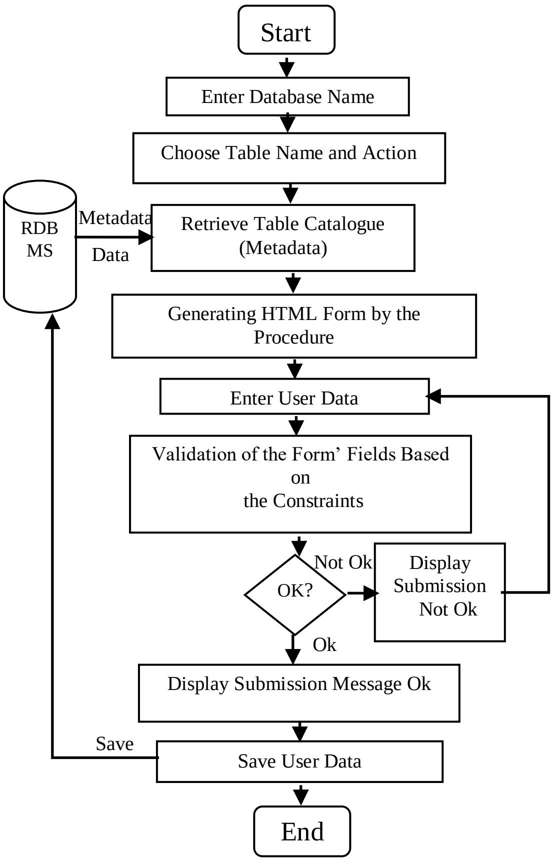

Start
Enter Database Name
Choose Table Name and Action
RDB MS
Metadata
Data
Retrieve Table Catalogue (Metadata)
Generating HTML Form by the Procedure
Enter User Data
Validation of the Form' Fields Based on the Constraints
OK?
Not Ok
Display Submission Not Ok
Ok
Display Submission Message Ok
Save
Save User Data
End

Figure1: System Flow Diagram

4.1. Web to Database Connecting Methods

Web to database connecting methods are generally as follows:

- Middleware technology is becoming - popular to connect database with the web.
- ODBC and JDBC are the most popular type of database access middleware.
- From a network administrator's point - of view, they consists of client and server driver software.
- From a programmer's point of view, they are Application Programming Interfaces APIs.
- ODBC technology now allows web servers directly connect to database.
- Finally, more and more web-based database applications have been - building by using different bridging methods [1].

4.2. Metadata Table Design method

The metadata table describes the database structure by including the seven and six essential column elements required to represent a data model: table names, column names, data types, type name, column size, position and null-able. The link definition requires six fields to specify both the linked table and the linked field within the two tables.

4.3. Step By Step Procedure for Generating and Validation of HTML Form

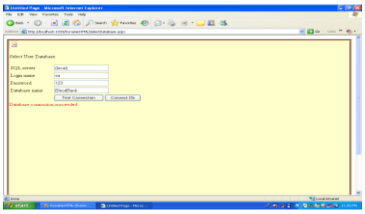
- If a field (based on a database column) is of type character with length (n) then automatically create the correct sized text field using n for the size attribute and display the field name as a text field label.
- If a field(based on a database column)is a foreign key, dynamically representing this field as a drop-down list, with data listed from the primary key table, this is to maintain referential integrity e.g. city field at supplier form.
- If a field (based on a database column) is of a Boolean type, represent it as a radio button with TRUE and FALSE values. These are based on dynamically creating HTML alone; using the - metadata.

This technique to display help messages to the user and validate the input data, for example extending the previous examples:

- If a field (based on a database column) is of - type character or integer, dynamically display a hint, assuming the text field accepts character or integer values.
- If a field (based on a database column) is required (not null), dynamically display a hint saying this field is required .When the form is submitted, the data entered to the form will be validated and a proper feedback - message will appear to the user, if submission was successful or not.

4.4. Screen Shoot for Database User from the Web (Sample Blood-Bank Database)

1. Home Page of the System: The user must enter database name that want to use.

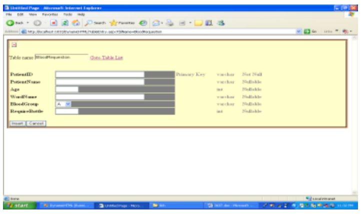

Select User Database

SQL server (local)
Login name sa
Password 123
Database name BloodBank

Database connection succeeded

Figure2: Home Page for System User

2. Entry Form of the System: The user can enter data into the database from the web.



Untitled Page - Microsoft Internet Explorer

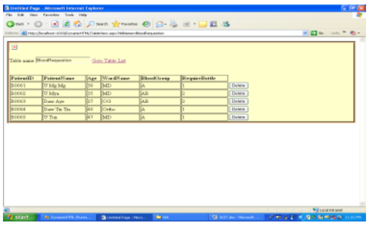
Table name BloodRequestion Goto Table List

PatientID		Primary Key	varchar	Not Null
PatientName			varchar	Nullable
Age			int	Nullable
WardName			varchar	Nullable
BloodGroup	A		varchar	Nullable
RequireBottle			int	Nullable

Insert Cancel

Figure3: Entry Form for Database User

3. View Form of the System: The user can see database from the web.



Untitled Page - Microsoft Internet Explorer

Table name BloodRequestion Goto Table List

PatientID	PatientName	Age	WardName	BloodGroup	RequireBottle	
B0001	U Mg Mg	30	MD	A	1	Delete
B0002	U Mya	35	MD	AB	2	Delete
B0003	Daw Aye	27	OG	AB	2	Delete
B0004	Daw Tin Tin	40	Ortho	A	1	Delete
B0005	U Tin	47	MD	A	1	Delete

Figure4: View Form for Database User

4.5. Advantages of Three Tier Architecture

- Removes a huge processing burden from client machines.
- Can be used to consolidate enterprise - wide business rules as application server's process business rules in a single place for use by multiple - applications. When rules change, only a change to the application server is required.
- Any knowledge of the database server may be hidden from the client and database queries may be presented to client in alternative forms.

4.6. Features of our approach

HTML forms that provide the query interface to the database will generate and validated automatically. HTML is not stored data so database source is need. The HTML form is not requiring maintenance. The HTML form is not a static .html files. It is a dynamic page. World Wide Web access to a database can be providing to existing databases. No changes to the database are required. Rather, our method uses the structure and information contained in the database to generate HTML pages and links when the data is accessed. If new HTML, or other display, standards emerge the database does not need to be changed. For database browsing this can block impossible queries or provide help on possible values. For database updates, this can ensure integrity by automatically requiring all necessary data.

5. Conclusions

The system intends for the various database backend web applications will be easier, faster and less error prone. This system describes the fundamental steps to write-out each HTML control from database fields (database metadata), validate data entry, adhoc querying from the metadata. Another vital function allowed by the metadata is the saving of form fields directly to the database.

It has been established that only C# offers a generalized metadata based approach. Other programming languages would require access to the database management system specific-system catalog. Issues of metadata support across ODBC implementations, and questions about performance implications of accessing metadata. Further extensions, some problems encountered were configuration problems that required carefully reading the associated-documentation. Further work is also required in these areas when the approach is extended from querying metadata automatically to querying the data itself. This is requires to select all possible languages, for exchange from a database table format to dynamically build a form element format.

REFERENCES

[1] An example web-based application is - describe in Section 4. Application architecture analysis and preliminary performance - measurement results are also... ausweb.scu.edu.au/aw99/papers/zhao/paper.html

[2] J. Lu, W. Zhao and B. Glasson, Formal specifications of Web-to-database interfacing - models, In Proceedings of Asia Pacific Web Conference (APWeb98), pp. 133-140, 1998.

[3] P. Bernstein et al. The Asilomar report on database research, ACM Sigmod Record, - 27(4):74-80, Dec.1998.

[4] Mark Weiner, Micah Sherr and Abigail Cohen, Metadata tables to enable dynamic data modeling and web interface design: the SEER example, International Journal of Medical Informatics Volume 65, Issue 1, April 2002, Pages 51-58, ISSN: 1386-505.

[5] Date, C.J., "An Introduction to database systems" Sixth Edition, the system programming series, Addison Wesley publishing Company Inc., 1994

[6] K. G. Jeffery, Metadata the Future of - Information Systems, 12th Conference on advanced information systems engineering, vol 1789, Jun 2000.

[7] Jason Hunter with William Crawford Firest Edition (1998):"ActiveX Data Object (ADO.NET)", O'Reilly & Associates, USA. ISBN: 1-56592-39x.Activex Data Object Web Development Kit, Version 1.0.1, Microsoft system,Inc.http://c#.microsoft.com/products/ado /README.html.