QuakeTables
Fault Database for Southern California

June 5, 2004

Table of Contents

Team 2
Purpose and contents 3
System Architecture 3
Implementation 8
Functionality 8
Data in QuakeTables Fault Database 15
Performance Evaluation 27
URL of Fault Database System 35
References 35
Appendix A – Fault Database Data Sources 37
Appendix B – Database Schema 41
Appendix C 47

JPL Clearance: CL#04-1973

Team

Andrea Donnellan:
Principal Investigator, Database design and implementation
Jet Propulsion Laboratory
Mail Stop 183-335
4800 Oak Grove Drive
Pasadena, CA 91109-8099
donnellan@jpl.nasa.gov
818-354-4737

Dennis McLeod:
Database semantics and interoperation
University of Southern California
Mail Code 0781
3651 Trousdale Parkway
Los Angeles, CA 90089-0742
mcleod@pollux.usc.edu
213-740-4504

Lisa Grant:
Fault database architect
University of California, Irvine
Environmental Health, Science & Policy
Irvine, CA 92697-7070
lgrant@uci.edu
949-824-5491

Academic Training:

Anne Yun-An Chen:
University of Southern California, Ph.D. graduate student is undertaking development of the fault and federated databases.

Sang-Soo Sung:
University of Southern California, Ph.D. graduate student is undertaking development of the fault and federated databases, and building a system to support semantics-based interoperation

Miryha M. Gould:
University of California, Irvine, Ph.D. graduate student is developing the geological aspects of the fault database.

Gabriela Noriega-Carlos:
University of California, Irvine, Admitted/incoming Ph.D. graduate student is assisting with geological aspects of the fault database.

Purpose and contents

This report documents the design and current contents of the QuakeTables Fault Database developed for the QuakeSim project. An alternate, descriptive title of the QuakeSim project is “Numerical Simulations for Active Tectonic Processes: Increasing Interoperability and Performance.” The objectives of the QuakeSim project are to:

- Develop a solid Earth science framework in order to better understand active tectonic and earthquake processes.
- Construct a fully interoperable system of tools for studying these processes.

The QuakeSim project includes a web accessed, public database of geologic and modeled faults with geometric, kinematic and neotectonic parameters. The fault database is herein named QuakeTables. The contents of QuakeTables can be used to cross-validate different simulation methods, explore competing theories of plate boundary development, perform case studies with widely accepted assumptions, and provide input for visualization software to display simulation results. QuakeSim is sponsored by the NASA Earth Science Enterprise, in partnership with Goddard Space Flight Center, with the full participation of the related science and technology communities. Additional information and documentation is posted on the QuakeSim homepage http://quakesim.jpl.nasa.gov/.

This report is a reference guide that describes the parts and functions of QuakeTables to fulfill a documentation requirement for a QuakeSim project milestone. It is not intended to be a user guide, although it does provide useful information for QuakeTables users.

System Architecture

Background

The QuakeTables database system manages a variety of types of earthquake science data and information. There are pre-existing collections, with web-based access interfaces; there are also some structured collections managed by general-purpose database management systems. This new database enables the characterization of dynamically-defined earthquake faults.

In past work we have developed XML Document Type Definitions to describe various parameters of earthquake faults and input data. We developed the QuakeTables earthquake fault database based on the previous work. We continue to work with communities that have begun to establish data standards, such as the seismic community (effort led by Berkeley), and the International GPS Service.

There has long been a need for establishing a database of faults for seismic hazard analysis. Several databases have been constructed for this purpose by the U.S. Geological Survey (USGS), the California Geological Survey (CGS) and the Southern California Earthquake Center (SCEC), each with a different format. The primary goal of the existing databases [e.g. WGCEP, 1995; Petersen et al., 1996; Cao et al., 2002] and current collaborative efforts by USGS, CGS and SCEC on the "RELM" database, is to provide input for probabilistic assessment of ground motion parameters. Existing databases are not compatible and are not suitably formatted or readily accessible for simulations. For example, much of the focus has been on establishing whether or not certain faults exist or are "active" as defined by the state of California, and how their proposed geometries would affect ground motion estimates.

Most faults in the existing databases have been divided into characteristic segments that are expected to rupture as a unit. Geologic slip rates are assigned to large segments rather than to the specific locations (i.e. geographic coordinates) where they were measured. These simplifications and assumptions are desirable for seismic hazard analysis, but they introduce a level of geologic interpretation and subjective bias that is inappropriate for simulations of fault behavior. The QuakeTables database includes primary geologic and paleoseismic fault parameters (fault location/geometry, slip rate at measured location, measurements of coseismic displacement, dates and locations of previous ruptures, etc.) as well as separate interpreted/subjective fault parameters [Grant, 1999] such as characteristic segments, average recurrence interval, magnitude of characteristic ruptures, etc. Both parameter types will be updated as more data are acquired and interpreted through research and the numerical simulations. Additionally, the database includes model material parameters for rectilinear layers.

To support this earthquake fault database and others, we acquired and employed a state-of-the-art commercially-available general-purpose database management system (MySQL). In particular, we utilized a basic relational database management system running on a PC under LINUX. These systems support the definition, storage, access, and control of collections of structured data. To meet the needs of the simulation community, the database supports the following features:

- extensible type definition capabilities in the database management system (to accommodate application-specific kinds of data)
- the ability to combine information from multiple databases
- mechanisms to efficiently return XML results from requests

The system architecture for QuakeTables is shown in Figure 1. A simplified extended entity relationship (EER) schema for the initial QuakeTables database is shown in Figure 2. The parameters are defined in the data section of this report. A relational implementation of the database shown in simplified form in Figure 2 is described in detail in Appendix B.

Design Requirements

This system allows the user to operate on the data through an Internet connection currently at http://infogroup.usc.edu:8080/index.html. Prospective users should contact one of the above authors for a user name and password. The user interface is browser-based. Java Script must be supported at the user's side. The requirements of this system are the following:

1. Hardware Requirements (HR):

   HR1: The user shall have a computer that has network access.

   HR2: The server shall have Internet connection.

   HR3: The server shall allow Hypertext Transfer Protocol (HTTP).

2. Software Requirement (SR)

   SR1: The user shall have web browsers.

   SR2: The PC on the user's side shall support Java Script program.

   SR3: The server shall have an operating system that supports multi-tasking.

   SR4: The server shall have one database management system for keeping data.

   SR5: The server shall deal with the web base request from users.

3. Security Requirement (ER)

   ER1: The system shall only allow Hypertext Transfer Protocol (HTTP) for the end user.

4. Scalability Requirements (AR)

   AR1: The system administrator(s) can extend the functionality at any time.

   AR2: The system administrator can add more tables and insert additional data anytime.

5. System Feedback Requirements (FR)

   FR1: The system shall pop-up an error message dialog box after invalid data input.

   FR2: The system shall provide enough information about request processing status.

6. Performance Requirement (PR)

   PR1: The system shall provide a response time within 10 seconds.

7. Operational Requirements (OR)

OR1: The system shall provide a method of inserting data.

OR2: The system shall provide a method of deleting data.

OR3: The system shall provide a method of reviewing data.

OR4: The system shall provide the contact information for the end user.

OR5: The system shall provide a log of all events generated during operation.

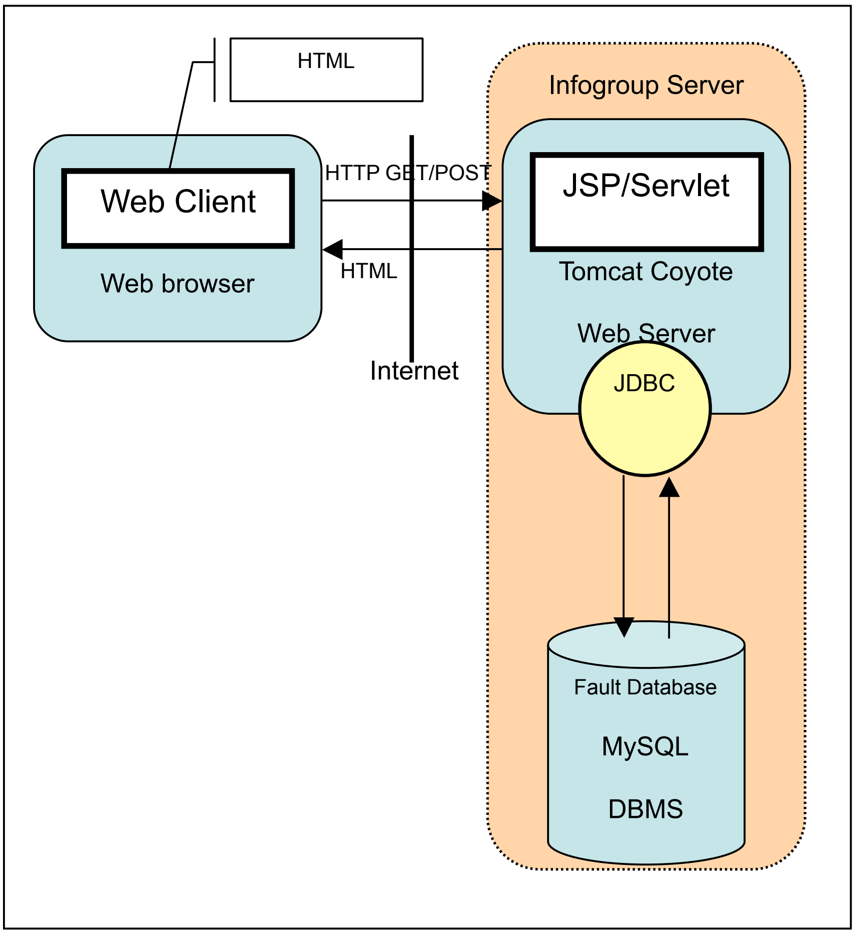
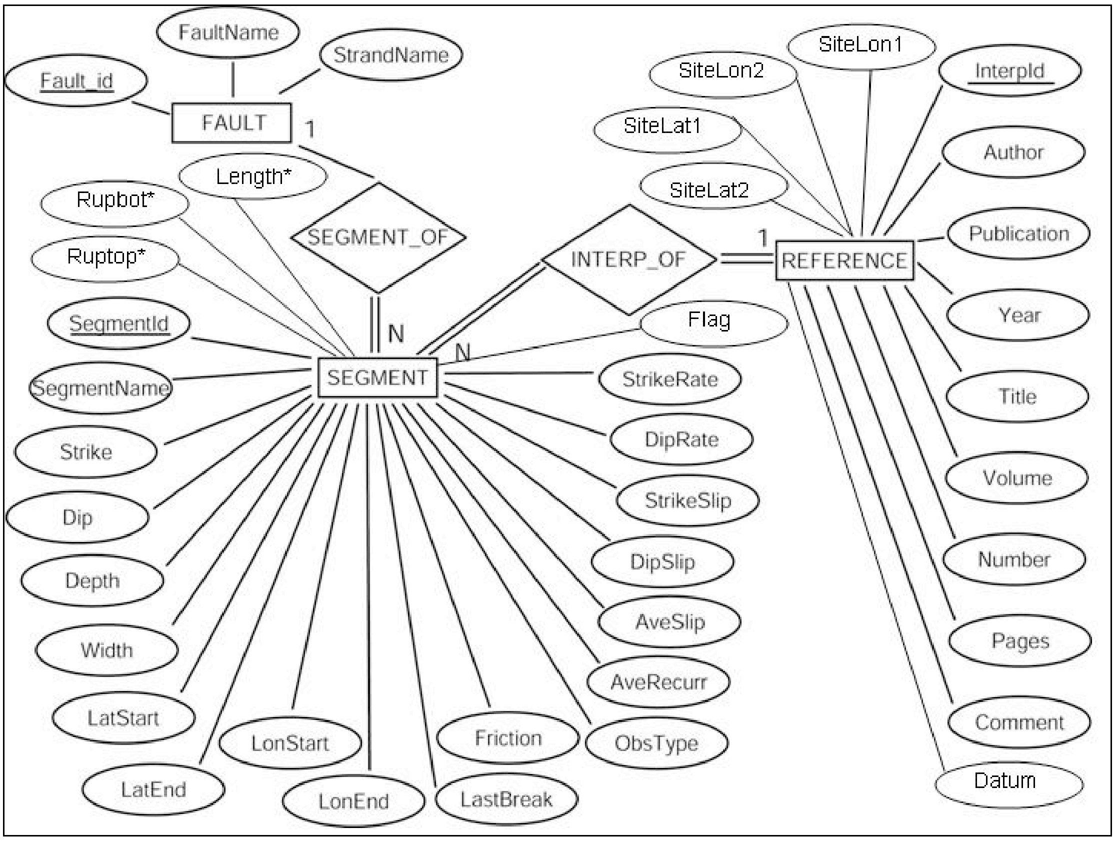


Figure 1. System Architecture.



* This attribute is defined for the California Geological Survey (CGS) fault data set.

Figure 2. A simplified extended entity relationship for the QuakeTables fault database.

Operating System

The operating system installed in our server is Linux, Red Hat 8.0-x86.

Web Server

Apache and Jakarta Tomcat V. 4.1 were installed on our server in order to support web pages (HTML) and JavaServer Pages (JSP).

Database Management System

MySQL 3.23 is used as our backend database management system. MySQL was chosen because it runs on multiple platforms: both web and standalone interfaces are available, and it also has a large testbed for an open source community. Therefore, it provides stable performance in web environments. Consequently, it fits our current needs better than other commercial DBMS, and it is free.

Implementation

User Interface

Hypertext markup language (HTML) and JavaScript are used to program our user interfaces. HTML is used on the World Wide Web and provides the basis for web pages. JavaScript is employed to check input errors and provide alerts to the end users. Our user interfaces are in the formats of forms to provide simple but sufficient functions. Currently, our system does not directly support getting faults in a spatial range. The graphic-based queries will be supported by the future QuakeSim portal or other programmed interfaces that can provide such user-friendly features.

The user interface at http://infogroup.usc.edu:8080/index.html is browser based.

When the user clicks on submit (here means "insert", "select", or "delete"), the backend corresponding JSP programs will be called to further process the request. Separate user interfaces for managing the CGS data set were created to simplify the data entry jobs. The data are sent to the same fault database. Main user interfaces for the fault database are designed to manage all data sets, including the CGS data set. Currently, the interfaces for managing the CGS data set are not available to common users.

Functionality

Insert Data

The insert data function is handled by two programs: Insert.jsp and insert.jsp (one program begins with an upper case "I" and the other with a lower case "i". The name of each program is underscored in this documentation).

Insert.jsp:

There are four purposes of the routine: getting parameters, validating the data, showing a confirmation message, and passing parameters to insert.jsp. The validation of data falls into two categories: existence and format. Existence means that certain parameters must be assigned some values. In Figure 3, one example of our error message regarding the existence check is shown.

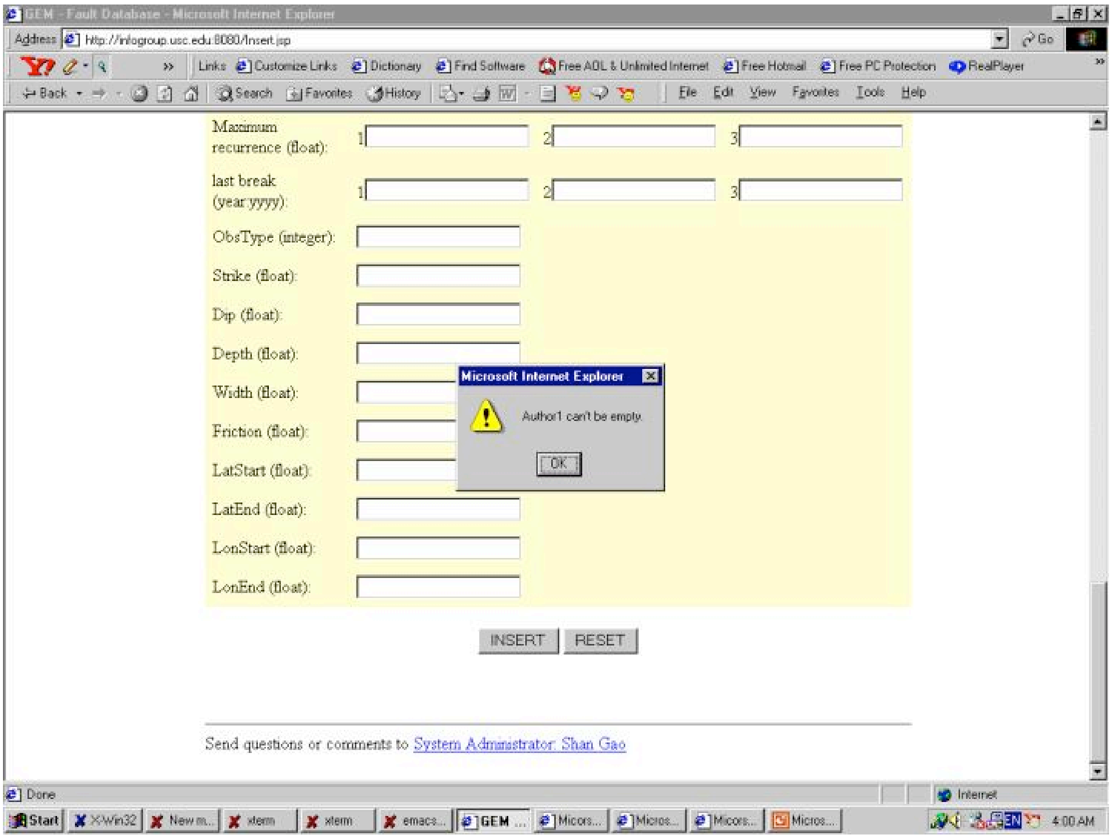


Figure 3. Error message for requiring value for parameter Author1.

Some parameters have to be entered in specified formats. For example, parameter Year has to be a four-digit and an integer number. In Figure 4, we display another example of our corresponding error message.

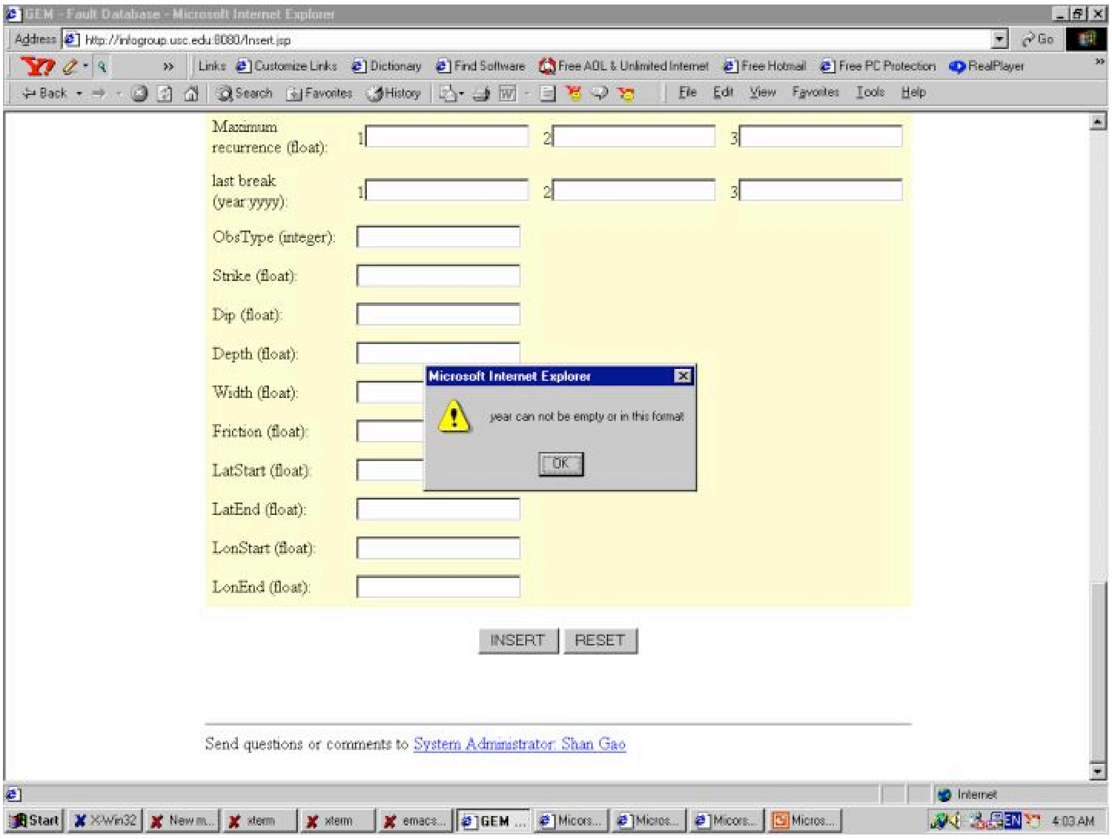


Figure 4. Error message for incorrect format.

A confirmation message will pop-up before the parameters are passed to insert.jsp. Values of four parameters, Fault ID, Fault Name, Segment ID, and Segment Name, will be shown in the pop-up window and the system will ask our user to confirm the inputs. In Figure 5, we show the confirmation message.

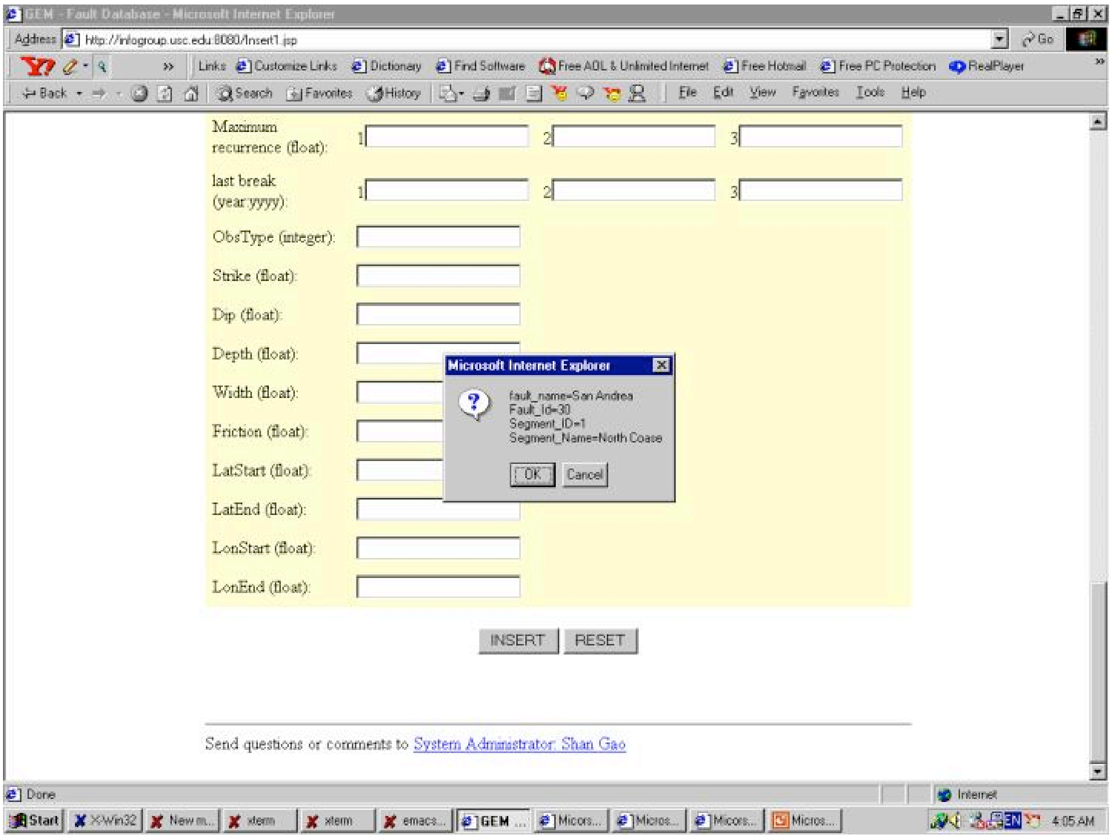


Figure 5. Confirmation message.

insert.jsp

insert.jsp will get the parameters from Insert.jsp, form SQL queries, call JDBC connection to our DBMS, and execute queries. After successfully executing queries, a message will be shown in the browser to inform the user of completion of the insertion.

Select Data

The select data function is handled by two programs: Select.jsp and select.jsp (one program begins with an upper case "S" and the other with a lower case "s". The name of each program is underscored in this documentation).

Select.jsp:

Select.jsp performs three actions: getting parameters, validating the data, and passing parameters to select.jsp. These actions are the same as those of Insert.jsp.

select.jsp

select.jsp will get the parameters from Select.jsp, form SQL queries, call JDBC connection to our DBMS, execute queries, and print out the query results in the browser. If no parameter is specified, all data in QuakeTables would be shown.

Delete Data

The delete data function is handled by two programs: Delete.jsp and delete.jsp (one program begins with an upper case "D" and the other with a lower case "d". The name of each program is underscored in this documentation).

Delete.jsp:

This program will get parameters, validate the data, and pass parameters to delete.jsp. These actions are the same as those of Insert.jsp.

delete.jsp

This routine will get the parameters from Delete.jsp, form SQL queries, call a JDBC connection to our DBMS, execute queries, and print out the query results in the browser. The cascade deletion is not allowed. Therefore, an error message will be shown if the user doesn't specify any parameter. In Figure 6, we display the error message regarding to this.

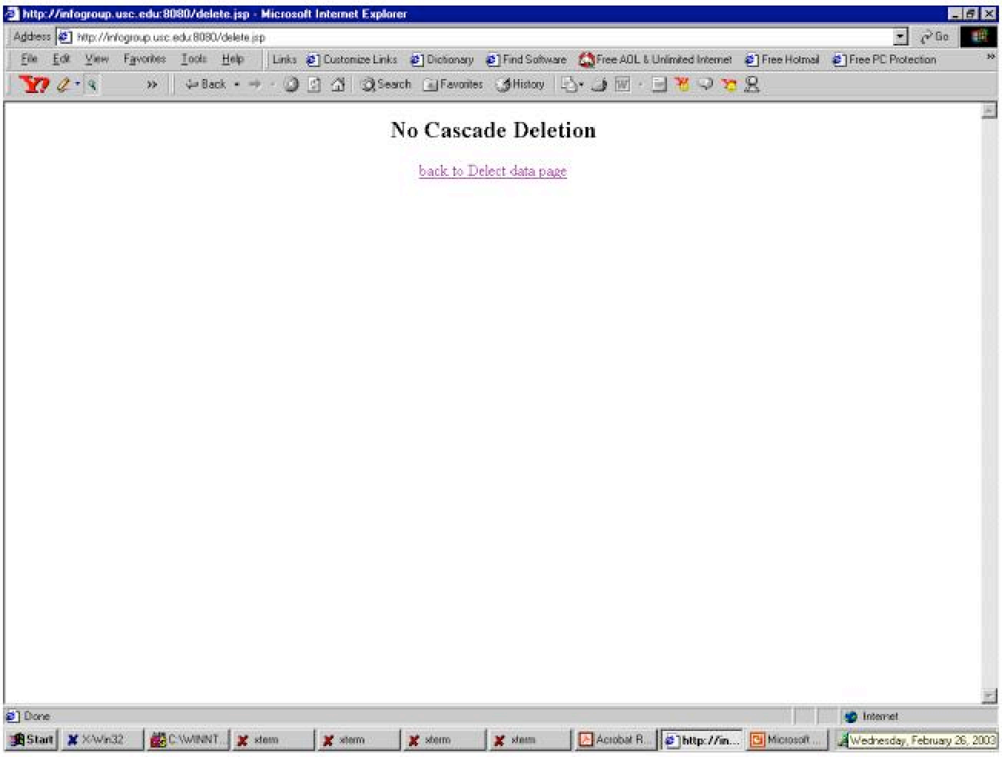


Figure 6. Error message regarding to forbidden cascade deletion.

Insert Layer Data

The insert layer data function is handled by two programs: Insert_Layer.jsp and insert_Layer.jsp (One program name here begins with an upper case "I" and the other with a lower case "i". The name of each program is underscored in this documentation).

***Insert_Layer.jsp*:**

Like Insert.jsp, this routine is designed for four purposes: getting parameters, validating the data, showing confirmation message, and passing parameters to insert_Layer.jsp.

insert_Layer.jsp:

insert_Layer.jsp will get the parameters from Insert_Layer.jsp, form SQL queries, call JDBC connection to our DBMS, and execute queries. After successfully executing queries, a message will be shown in the browser to inform the user of completion of the insertion.

Select Layer Data

The select layer data function is handled by two programs: Select_Layer.jsp and select_Layer.jsp .

Select_Layer.jsp:

Select_Layer.jsp performs three actions: getting parameters, validating the data, and passing parameters to select_Layer.jsp. These actions are the same as those of Insert_Layer.jsp.

select_Layer.jsp

select_Layer.jsp will get the parameters from Select_Layer.jsp, form SQL queries, call JDBC connection to our DBMS, execute queries, and print out the query results in the browser. If no parameter is specified, all data for Layer in QuakeTables would be shown.

Insert CGS Fault Data

The insert CGS Fault data function is handled by two programs: Insert_CGSr.jsp and insert_CGS.jsp (one program begins with an upper case “I” and the other with a lower case “i”. The name of each program is underscored in this documentation).

Insert_CGS.jsp:

Like Insert.jsp, this routine is designed for four purposes: getting parameters, validating the data, showing confirmation message, and passing parameters to insert_CGS.jsp.

insert_CGS.jsp:

insert_CGS.jsp will get the parameters from Insert_CGS.jsp, form SQL queries, call JDBC connection to our DBMS, and execute queries. After successfully executing queries, a message will be shown in the browser to inform the user of completion of the insertion.

Select CGS Fault Data

The select layer data function is handled by two programs: Select_CGS.jsp and select_CGS.jsp (Note that one program begins with an upper case "S" and the other with a lower case "s". To reiterate, the name of each program is underscored in this documentation.).

Select_CGS.jsp:

Select_CGS.jsp performs three actions: getting parameters, validating the data, and passing parameters to select_CGS.jsp. These actions are the same as those of Insert_CGS.jsp.

select_CGS.jsp

select_CGS.jsp will get the parameters from Select_CGS.jsp, form SQL queries, call JDBC connection to our DBMS, execute queries, and print out the query results in the browser. If no parameter is specified, all data in QuakeTables for Layer would be shown.

Delete CGS Fault Data

The delete CGS fault data function is handled by two programs: Delete_CGS.jsp and delete_CGS.jsp (one program begins with an upper case "D" and the other with a lower case "d". The name of each program is underscored in this documentation).

Delete_CGS.jsp:

This program will get parameters, validate the data, and pass parameters to delete_CGS.jsp. These actions are the same as those of Insert_CGS.jsp.

delete_CGS.jsp:

This routine will get the parameters from Delete_CGS.jsp, form SQL queries, call JDBC connection to our DBMS, execute queries, and print out the query results in the browser. The cascade deletion is not allowed. Therefore, an error message will be shown if the user doesn't specify any parameter.

Search Data

Users can use either one author name or one fault name to search for fault data. The 1card search function is handled by the program: wildcard.jsp.

wildcard.jsp

wildcard.jsp accepts either a partial string of one author or one partial fault name as the input. wildcard.jsp forms SQL queries, calls JDBC connection to our DBMS, executes queries, and prints out the query results in the browser. The format of the query results is a list of data entries that include the attribute values of (partial) author name or the fault name requested by the user. In the following table, examples of the user input and the mapped attribute values are shown:

Input Category	User Input	Mapped Attribute Values
Author Name	Rundle	**Rundle**, J. B.
		Rundle, P. B.
	G. L.	**G**urrola, **L**. D.
		Grant, **L**. B.
		Kello**g**g, **L**. H.
Fault Name	valley	Imperial **Valley**
		Panamint **Valley**
		Johnson **Valley**
		Death **Valley**
		Hunter Mountain-Saline **Valley**
		Owens **Valley**
		Concord-Green **Valley**
		Round **Valley**
		Homestead **Valley**
		Morongo **Valley**
	rose	**Rose** Canyon
		Newport-Inglewood-**Rose** Canyon

Data in QuakeTables Fault Database

QuakeTables has been designed to accommodate several types of fault data and data sets, including primary paleoseismic and geologic fault data, non-primary data [Grant and Gould, 2004], and simulated or hypothetical data. It also includes Material Rectilinear Layer parameters. QuakeTables was developed for and initially populated with data from California, but there is no geographic restriction on future data entries.

QuakeTables is currently populated with primary paleoseismic data from major California faults, and with two structured data sets containing summary fault attributes and geographic coordinates of fault segments. The structured data sets include a recent version of Virtual California [Rundle et al., 2002] and two fault databases [Petersen et al., 1996; Cao et al., 2002; Frankel et al., 2002] published by the California Geological Survey (CGS) and U. S. Geological Survey (USGS) for seismic hazard analysis. These structured data sets provide geographic coordinates, geometry, and summary attributes for many active faults and fault segments in California. More paleoseismic data from research publications will be added in the future. Some database population and coordination issues are described by Gould et al. [2003a,b,c] and Chen et al. [2003].

Source

Data in QuakeTables are extracted from refereed journal articles, professional papers, professional reports, and conference abstracts. Sources of data currently in the database are listed in Appendix A. Additional data will be added in the future, as described in Appendix C.

Parameters

The following Tables contain parameters with particular properties that describe the data in QuakeTables. Conventions are shown below in Figure 7.

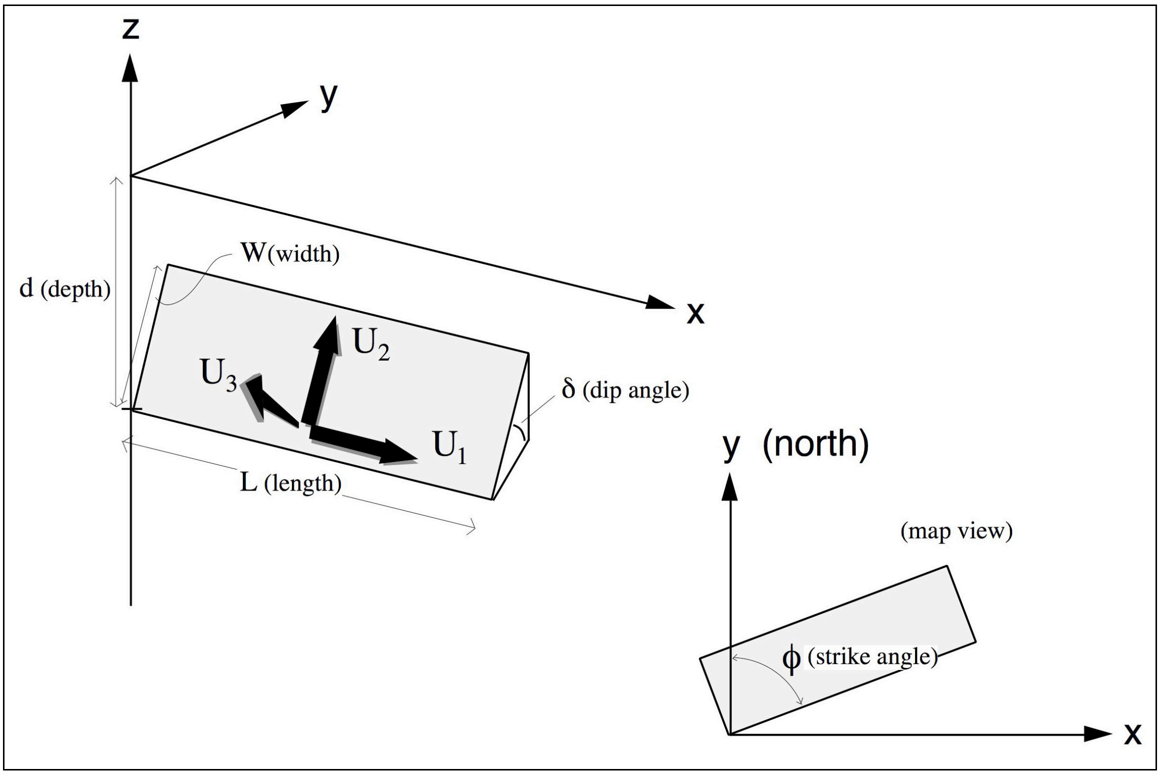


Figure 7. Conventions for geometric parameters in QuakeTables. The parameters follow the conventions for the program DISLOC.

Table: FAULT

Parameter	Full Name	Units	General Definition
			Definitions for CGS Data Set
Fault_id	Fault ID number	Integer	Fault ID numbers for southern California faults (numbers 1 through 51) are extracted from the Southern California Earthquake Center (SCEC) Phase II report (Working Group on California Earthquake Probabilities, 1995). Faults not included in the SCEC report are numbered 52 - 184. Additional faults may be added, beginning with number 185.
FaultName	Fault name	Character	Fault names are taken from study site descriptions provided in each individual data source. Faults are fractures dividing two bodies of rock, along which the rock masses have moved relative to each other.
			Fault names are taken from Petersen et al., 1996 or other CGS publication.
StrandName	Strand name	Character	Strand names are taken from study site descriptions provided in each individual data source. Fault strands are fault branches or splays of a larger fault zone.

Table: REFERENCE

Parameter	Full Name	Units	General Definition Definition for CGS Data Set
InterpId	Interpretation ID number	Integer	The interpretation ID numbers are automatically generated for each individual data source entered into the fault database.
Author	Author names	Character	The authors of each publication are listed in authorship order. The authors (up to 5) of the CGS fault publication are listed in authorship order.
Publication	Publication title	Character	The name of the journal article, professional paper, professional report, or conference abstract. The name of the journal article or CGS report.
Year	Year	Integer year A.D.	The year of publication of the source. The year of publication.
Title	Title	Character	The name of the journal, professional paper, professional report, or conference proceedings which features the publication. The title of the journal article or CGS report.
Volume	Volume	Character	The volume number of the journal, professional paper, professional report, or conference proceedings which features the publication. The volume number of the journal article or CGS report.

Number	Number	Character	The number or issue of the journal, professional paper, professional report, or conference proceedings which features the publication. The number of the journal article or CGS report.
Pages	Pages	Character	The page numbers of the publication. The page numbers of the publication.
Comment	Comment	Character	The comment field allows for additional explanation regarding the data source. The comment field allows for additional explanation regarding the data source.
SiteLon1	Study site longitude1	Decimal degrees	The longitude of paleoseismic study site location. Study site coordinates define the center point of the study area, and are measured on the fault of study.
SiteLat1	Study site latitude1	Decimal degrees	The latitude of paleoseismic study site location. Study site coordinates define the center point of the study area, and are measured on the fault of study.
SiteLon2	Study site longitude2	Decimal degrees	The latitude of paleoseismic study site location. Study site coordinates define the center point of the study area, and are measured on the fault of study.
SiteLat2	Study site latitude2	Decimal degrees	The longitude of paleoseismic study site location. Study site coordinates define the center point of the study area, and are measured on the fault of study.
Datum	Reference datum	Character	Geographic datum for latitude and longitude coordinates.

Table: SEGMENT

Parameter	Full Name	Units	Definition
SegmentId	Segment ID number	Integer	Segment ID numbers are assigned for each fault segment, based on segments recognized by Petersen et al., 1996. The first segment of each fault is number 1. Faults with a single segment have Segment Id=1. Future entries or additional segments would use the next available number in the sequence.
SegmentName	Segment name	Character	Segment names are taken from study site descriptions provided in each individual data source.
Strike	Strike	Degrees azimuth	The geographic orientation of a fault plane. Specifically, strike is the direction of a horizontal line in the plane of a fault, as shown in Figure 6.
Dip	Dip	Degrees	The slope or vertical component of a fault plane.
Depth	Depth	Kilometers	The distance from the surface of the earth to the base of a fault or fault zone.
Width	Width	Kilometers	The surface or subsurface extent of a fault zone.
LatStart	Latitude start	Decimal degrees	The latitude of the location of one end of a fault or fault segment. LatStart corresponds with LonStart.
LatEnd	Latitude end	Decimal degrees	The latitude of the location of one end of a fault or fault segment. LatEnd corresponds with LonEnd.
LonStart	Longitude start	Decimal degrees	The longitude of the location of one end of a fault or fault segment. LonStart

			corresponds with LatStart.
LonEnd	Longitude end	Decimal degrees	The longitude of the location of one end of a fault or fault segment. LonEnd corresponds with LatEnd.
Datum	Datum	Character	The name of the geographic datum corresponding to the latitude and longitude points describing the location of the fault at the surface of the Earth. For example: NAD83. Geographic datums define the size and shape of the earth and the orientation of the coordinate systems used to map the Earth.
LastBreak	Last break	Calendar year	The year of the last earthquake rupture along a section or at a specific location on a fault.
Friction	n/a	n/a	This is a future implementation for Virtual California data users.
ObsType	Observation type	Character	The method of observation used by the source authors for data collection. For example, paleoseismology or geomorphology.
AveRecurr	Average recurrence interval	Years	The mean time between earthquakes of a given magnitude or magnitude range on a fault.
MinRecurr	Minimum recurrence interval	Years	The minimum time between earthquakes of a given magnitude or magnitude range on a fault.
MaxRecurr	Maximum recurrence interval	Years	The maximum time between earthquakes of a given magnitude or magnitude range on a fault.
AveSlip	Average slip per event	Meters	The mean amount of displacement on a fault

	per event		averaged over the number of earthquake events observed.
MinSlip	Minimum slip per event	Meters	The minimum amount of displacement on a fault averaged over the number of earthquake events observed.
MaxSlip	Maximum slip per event	Meters	The maximum amount of displacement on a fault averaged over the number of earthquake events observed.
AveDipSlip	Average dip slip per event	Meters	The mean amount of displacement perpendicular to the strike of a fault averaged over the number of earthquake events observed.
MinDipSlip	Minimum dip slip per event	Meters	The minimum amount of displacement perpendicular to the strike of a fault averaged over the number of earthquake events observed.
MaxDipSlip	Maximum dip slip per event	Meters	The maximum amount of displacement perpendicular to the strike of a fault averaged over the number of earthquake events observed.
AveStrikeSlip	Average strike slip per event	Meters	The mean amount of horizontal displacement parallel to the strike of a fault averaged over the number of earthquake events observed.
MinStrikeSlip	Minimum strike slip per event	Meters	The minimum amount of horizontal displacement parallel to the strike of a fault averaged over the number of earthquake events observed.
MaxStrikeSlip	Maximum strike slip per event	Meters	The maximum amount of horizontal displacement parallel to the strike of a fault averaged over the number of earthquake events observed.
AveDipRate	Average dip slip rate	Millimeters per year	The mean rate of displacement perpendicular to

	slip rate	per year	the strike of a fault averaged over a time period representing one to several large earthquakes.
MinDipRate	Minimum dip slip rate	Millimeters per year	The minimum rate of displacement perpendicular to the strike of a fault averaged over a time period representing one to several large earthquakes.
MaxDipRate	Maximum dip slip rate	Millimeters per year	The maximum rate of displacement perpendicular to the strike of a fault averaged over a time period representing one to several large earthquakes.
AveStrikeRate	Average strike slip rate	Millimeters per year	The mean rate of horizontal displacement parallel to the strike of a fault averaged over a time period representing one to several large earthquakes.
MinStrikeRate	Minimum strike slip rate	Millimeters per year	The minimum rate of horizontal displacement parallel to the strike of a fault averaged over a time period representing one to several large earthquakes.
MaxStrikeRate	Maximum strike slip rate	Millimeters per year	The maximum rate of horizontal displacement parallel to the strike of a fault averaged over a time period representing one to several large earthquakes.
NationalFaultId	National Fault Database ID Number	Integer	The Fault ID number used in the National Quaternary Fault and Fold Database developed by the U.S. Geological Survey (USGS).
FADFaultId	Fault Activity Database Fault ID number	Integer	The Fault ID number used in the Fault Activity Database (FAD) developed by the U.S. Geological Survey, the Southern California Earthquake Center (SCEC)

			and the Regional Earthquake Likelihood Models (RELM) Working Group.
Geometry	Fault geometry	Character	Fault geometry type, such as strike-slip or dip-slip.
Length*	Length	Kilometers	Length of fault or segment.
RateRank*	Slip rate rank	Character	Slip rate rank: W=well constrained, M=moderately constrained, P=poorly constrained, U=unconstrained.
Mmax*	Maximum magnitude	Magnitude units, decimal	Magnitude of maximum magnitude earthquake.
CharRate*	Characteristic rate	Per year	Rate of characteristic earthquakes, as defined by CGS.
DownDipWidth*	Down-dip width	Kilometers	Rupture bottom minus rupture top devided by sine of dip angle.
Ruptop*	Rupture top	Kilometers	Top of rupture plane.
Rupbot*	Rupture bottom	Kilometers	Bottom of rupture plane.
Rake*	Rake	Degrees	Rake of the slip vector in the fault plane.
Dip*	Dip	Degrees	The slope or vertical component of a fault plane.
Daz*	Dip azimuth	Degrees	Azimuth of the fault plane dip direction.
LatN*	Latitude north	Decimal degrees	Latitude of the north end of a fault segment.
LatS*	Latitude south	Decimal degrees	Latitude of the south end of a fault segment.
LonN*	Longitude north	Decimal degrees	Longitude of the north end of a fault segment.
LonS*	Longitude south	Decimal degrees	Longitude of the south end of a fault segment.

Flag	Flag	Character	Identifier for type of data: Flag=U for User-specified fault model Flag=P for Paleoseismic data Flag=C for CGS fault data Flag=V for Virtual California data sets Flag=O for Other data, as described in the comment field.

* The attributes are created for the parameters defined by Petersen et al [1996] and Cao et al [2002] for a structured data set, California Geological Survey (CGS) fault data.

Parameters in the other group, Material Rectilinear Layer, are listed in Table LREFERENCE and Table LAYER:

Table: LREFERENCE

Parameter	Full Name	Units	Definition
Author	Author names	Character	The authors (up to 5) of the publication or data source are listed in authorship order. If the layer is user-specified, then the author is the user.
Publication	Publication title	Character	The name of the journal article or publication.
Year	Year	Integer year A.D.	The year of publication.
Title	Title	Character	The title of the journal article or other publication.
Volume	Volume	Character	The volume number of the journal article or other publication.
Number	Number	Character	The number of the journal article or other publication.
Pages	Pages	Character	The page numbers of the publication.
Comment	Comment	Character	The comment field allows for additional explanation regarding the data source.

Table: LAYER

Parameter	Full Name	Units	Definition
LayerName	Layer name	Character	User specified layer name.
LayerID	Layer ID	Integer	User specified layer ID number.
LatOrigin	Latitude of Origin	Decimal degrees	Analogous to LatEnd, LonEnd for a SEGMENT.
LonOrigin	Longitude of Origin	Decimal degrees	Analogous to LatEnd, LonEnd for a SEGMENT.
Datum	Geographic datum	Character	The name of the geographic datum corresponding to the latitude and longitude points describing the location of the fault at the surface of the Earth. For example: NAD83. Geographic datums define the size and shape of the earth and the orientation of the coordinate systems used to map the Earth.
OriginX	Origin X coordinate	Kilometers	Location of the block southwest corner from the coordinate axis origin.
OriginY	Origin Y coordinate	Kilometers	Location of the block southwest corner from the coordinate axis origin.
OriginZ	Origin Z coordinate	Kilometers	Location of the block southwest corner from the coordinate axis

			origin.
Length	Length	Kilometers	Length of layer.
Width	Width	Kilometers	Width of layer.
Depth	Depth	Kilometers	Depth of layer.
LameLambda	Lame Lambda	Floating point	Lame constant Lambda.
LameLambdaUnits	Lame Lambda Units	Character	Units of Lame constant Lambda.
LameMu	Lame Mu	Floating point	Lame constant Mu.
LameMuUnits	Lame Mu Units	Character	Units of Lame constant Mu.
Viscosity	Viscosity	Floating point	Viscosity.
ViscosityUnits	Viscosity Units	Character	Units of viscosity.
ViscosityExponent	Viscosity Exponent	Floating point	Exponent of viscosity.

Performance Evaluation

Requirement Verification and Evaluation

Legend:

NC – No Change from initial specification

Mo – Modified from initial specification, see justification for description

Add – The new requirement which was added after initial specification

NI – Not Implemented in prototype, see justification for description

I – Implemented in prototype

Hardware Requirements (HR)

Req. #	Status	Justification
HR1	NC, I	This requirement is implemented by designing the browser-based user interfaces.
HR2	NC, I	This requirement is implemented, and the server is accessible on the internet. The domain name of the server is infogroup.usc.edu.
HR3	NC, I	This requirement is implemented. The port for HTTP has been opened since finishing Apache server installation.

Software Requirement (SR)

Req. #	Status	Justification
SR1	NC, I	This requirement is implemented. Web browsers are available in major operation systems.
SR2	NC, NI	Due to the variety of users, a message shall be shown to inform the user about the support of JavaScript functions.
SR3	NC, I	This requirement is implemented. Linux, Red Hat 8.0 has been installed on our server.
SR4	NC, I	This requirement is implemented. MySQL, the DBMS, has been installed on our server.
SR5	NC, I	This requirement is implemented. Jakarta Tomcat V. 4.1 has been installed and started up on our server in order to support web pages and JavaServer Pages (JSP)

Security Requirement (ER)

Req. #	Status	Justification
ER1	NC, I	This requirement is implemented. all the ports except HTTP and SSH have been closed for administrative and security purposes.

Scalability Requirements (AR)

Req. #	Status	Justification
AR1	NC, I	This requirement is implemented. Each function operates independently. Any addition of the function doesn't affect the functionality of the current system.
AR2	NC, I	This requirement is implemented. A commercially-available general-purpose database management system (MySQL) is employed.

System Feedback Requirements (FR)

Req. #	Status	Justification
FR1	NC, I	This requirement is implemented. JavaScript code has been added to the web pages in order to validate the format of the input data.
FR2	NC, I	This requirement is implemented. The query results would be shown to the user.

Performance Requirement (PR)

Req. #	Status	Justification
PR1	NC, I	This requirement is implemented. Web pages were designed to display the results in a short time.

Operational Requirements (OR)

Req. #	Status	Justification
OR1	NC, I	This requirement is implemented. Two programs, Insert.jsp and insert.jsp, were coded to handle the functionalities of data insertions (one program begins with an upper case "I" and the other with a lower case "i". The name of each program is underscored in this documentation).
OR2	NC, I	This requirement is implemented. Two programs, Select.jsp and select.jsp, were coded to handle the functionalities of data selections (one program begins with an upper case "S" and the other with a lower case

		"s").
OR3	NC, I	This requirement is implemented. Two programs, Delete.jsp and delete.jsp, were coded to handle the functionalities of data deletions (one program begins with an upper case "D" and the other with a lower case "d").
OR4	NC, I	This requirement is implemented. The user is able to click the link and generate mails to the system administrator.
OR5	NC, I	This requirement is implemented. The Jakarta Tomcat server records every event in the log file.

Web Pages for Successful Functions

- Insert Data

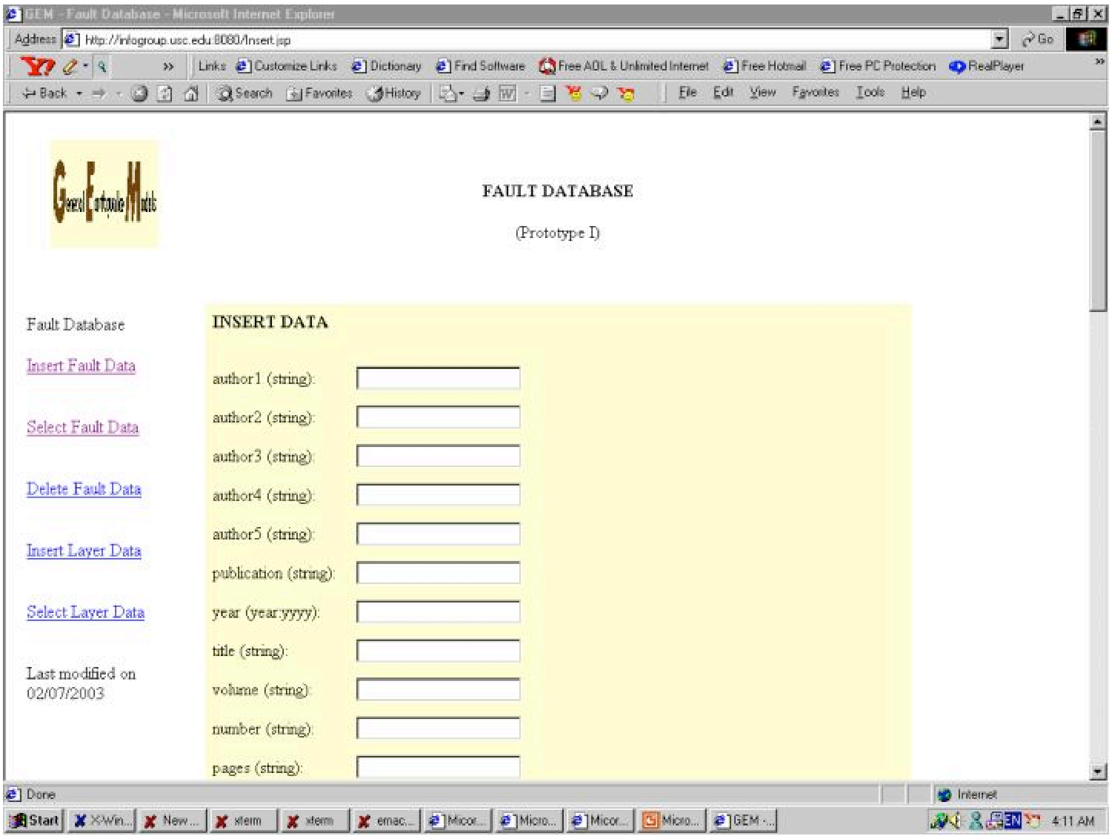


Figure 8. Insert.jsp.

- Select Data

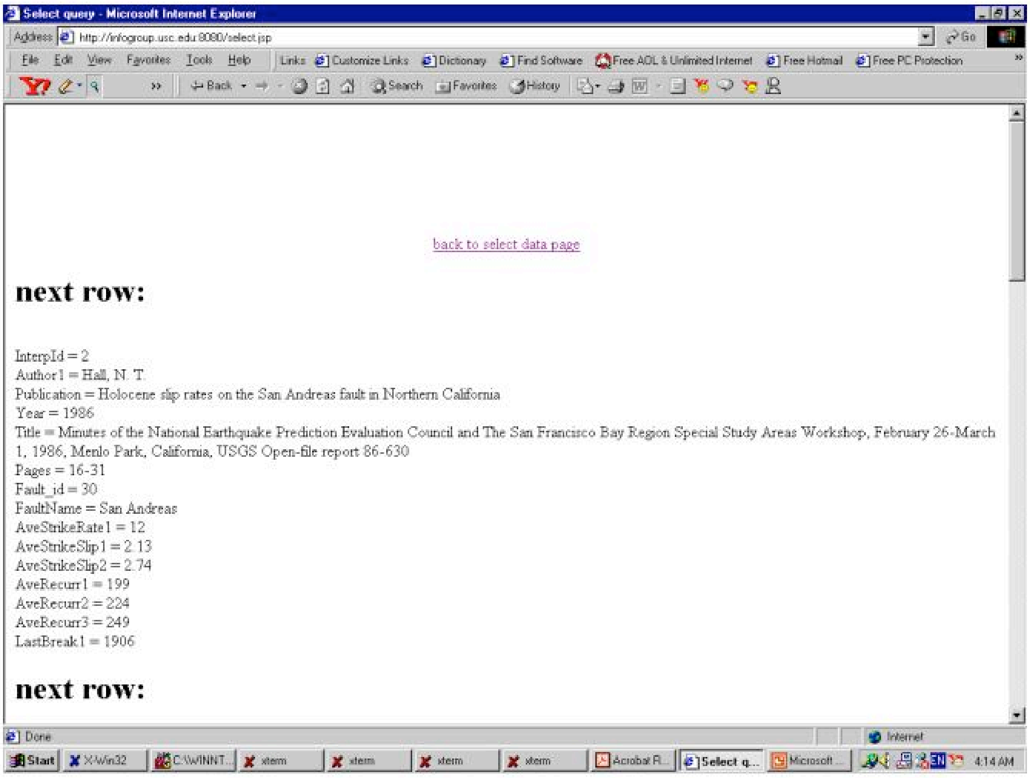


Figure 9. Query result for selecting data.

- Delete Data

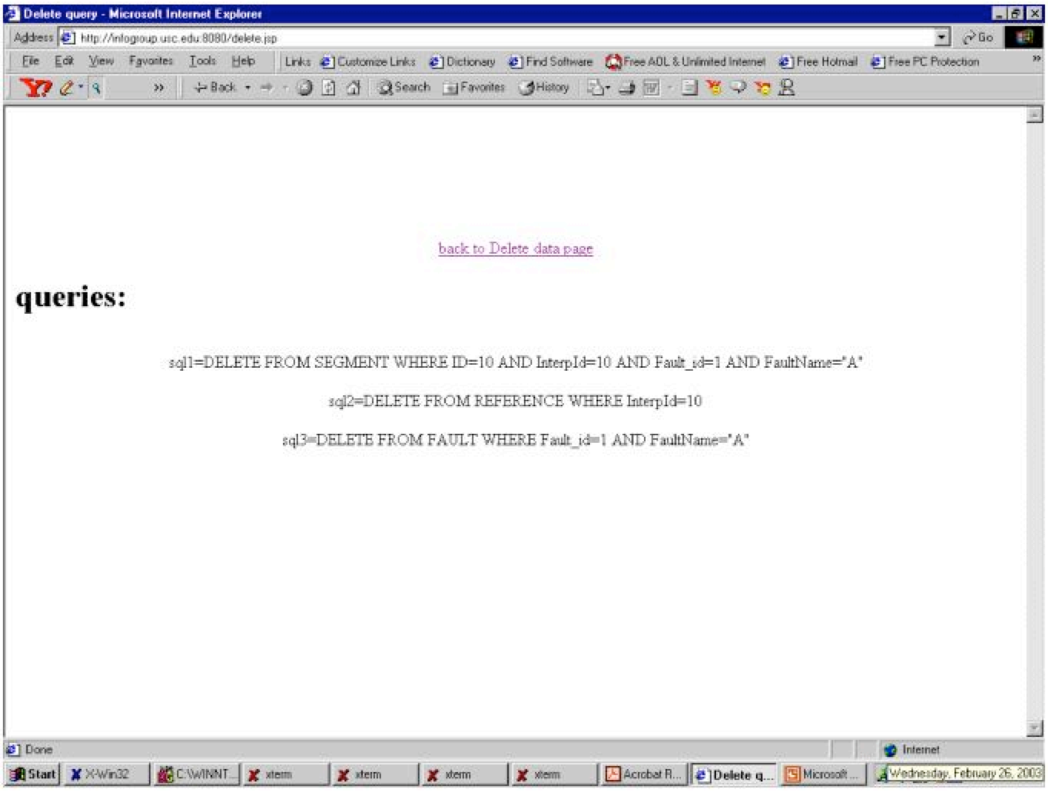


Figure 10. Showing the queries for deleting data.

- wildcard search

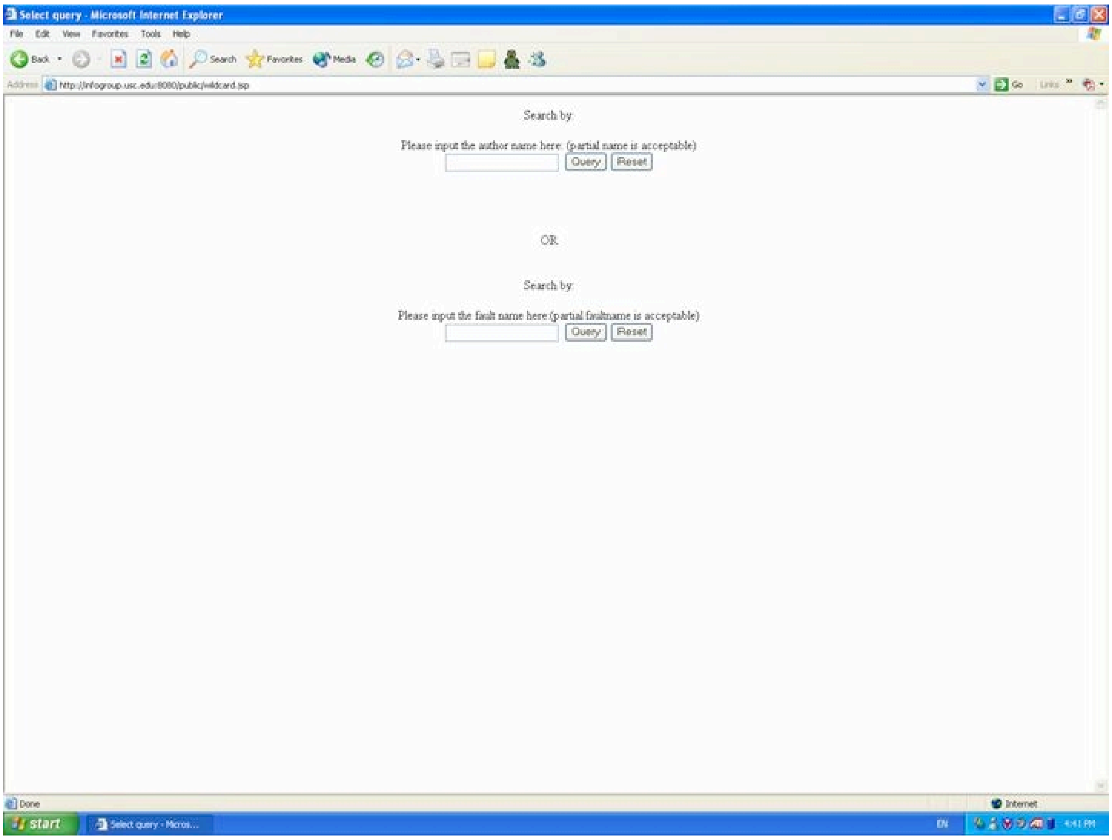


Figure 11. Entry page of the wildcard search.

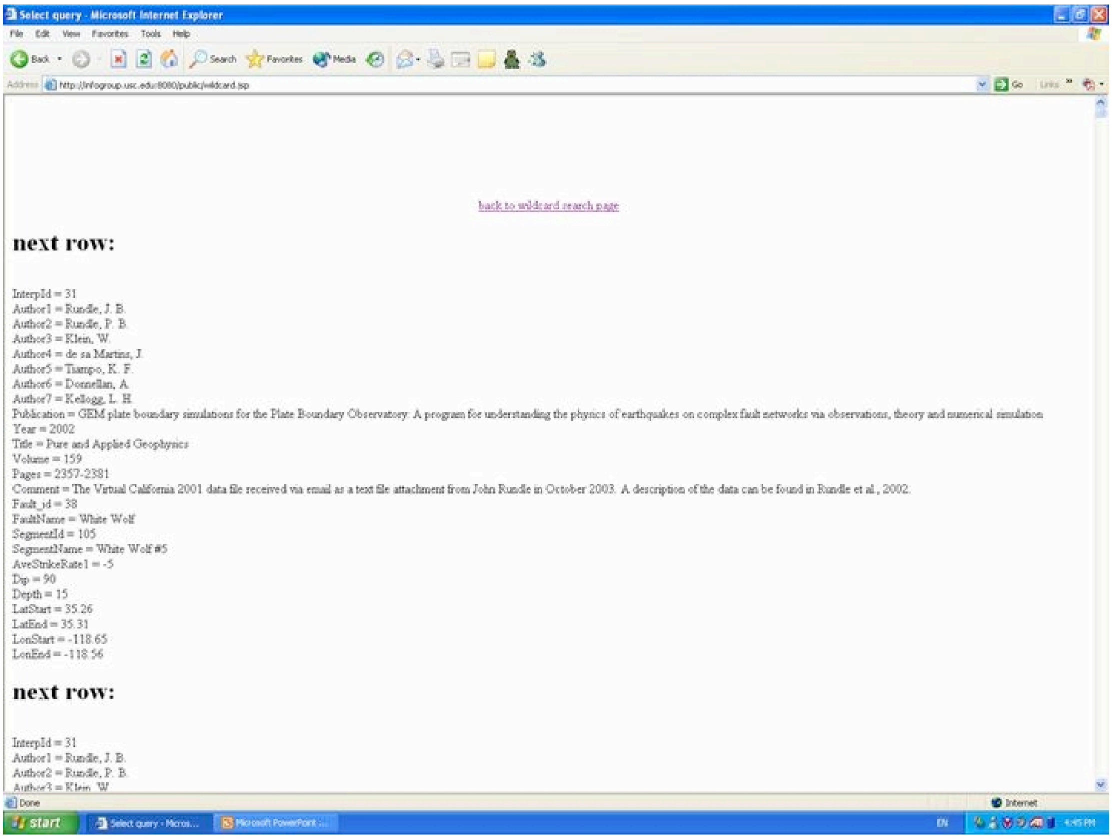


Figure 12. Example of wildcard search with the input of "Rundle" as the author name.

URL of Fault Database System

Authorized user only:

- http://infogroup.usc.edu:8080/index.html

Public Access:

- http://infogroup.usc.edu:8080/public/wildcard.jsp

References

Cao, T., Bryant, W. A., Rowshandel, B., Branum, D., & Wills, C. J. (2002). The revised 2002 California probabilistic seismic hazard maps.

Chen, A. Y., Chung, S., Gao, S., McLeod, D., Donnellan, A., Parker, J., Fox, G., Pierce, M., Gould, M., Grant, L., & Rundle, J. (2003). Interoperability and semantics for heterogeneous earthquake science data. 2003 Semantic Web Technologies for Searching and Retrieving Scientific Data Conference, October 20, 2003, Sanibel Island, Florida.

Frankel, A. D., Petersen, M. D., Mueller, C. S., Haller, K. M., Wheeler, R. L., Leyendecker, E. V., et al. (2002). Documentation for the 2002 update of the national seismic hazard maps (Open-File Report - U. S. Geological Survey No. OF 02-0420). Reston: U. S. Geological Survey.

Gould, M. M., Grant, L. B., Donnellan, A., & McLeod, D. (2003a). The GEM fault database: An update on design and approach. 2003 European Geophysical Society-American Geophysical Union-European Union of Geosciences Joint Assembly Meeting, April 6-11, 2003, Nice, France.

Gould, M.M., Grant, L. B., Donnellan, A., McLeod, D., & Chen, A. Y. (2003b). The QuakeSim fault database for California. 2003 Southern California Earthquake Center (SCEC) Annual Meeting Proceeding and Abstracts, September 7-10, 2003, Oxnard, CA.

Gould, M.M., Grant, L. B., Donnellan, A., McLeod, D., & Chen, A. Y. (2003c). The QuakeSim fault database for California. Geological Society of America Annual Meeting Abstracts with Programs, 35(6).

Grant, L. B., (1999). Integration and Implications of Paleoseismic Data for GEM, EOS Trans. Am. Geophys. Union, v. 80, p. F923.

Grant, L. B., & Gould, M. M. (in press, 2004). Assimilation of paleoseismic data for earthquake simulation. Pure and Applied Geophysics, 106(11/12).

Petersen, M. D., Bryant, W. A., Cramer, C. H., Cao, T., Reichle, M. S., Frankel, A. D., et al. (1996). Probabilistic seismic hazard assessment for the state of California (Open-File Report - U. S. Geological Survey No. OF 96-0706). Reston: U. S. Geological Survey.

Rundle, J. B., Rundle, P. B., Klein, W., de sa Martins, J., Tiampo, K. F., Donnellan, A., et al. (2002). GEM plate boundary simulations for the Plate Boundary Observatory: a program for understanding the physics of earthquakes on complex fault networks via observations, theory and numerical simulation. Pure and Applied Geophysics, 159(10), 2357-2381.

Working Group on California Earthquake Probabilities (WGCEP). (1995). Seismic hazards in southern California: probable earthquakes, 1994-2024. Bulletin of the Seismological Society of America, 85(2), 379-439.

Appendix A – Fault Database Data Sources

Biasi, G. P., Weldon, R. J., II, Fumal, T. E., & Seitz, G. G., (2002). Paleoseismic event dating and the conditional probability of large earthquakes on the Southern San Andreas fault, California. Bulletin of the Seismological Society of America, 92(7), 2761-2781.

Cao, T., Bryant, W. A., Rowshandel, B., Branum, D., & Wills, C. J. (2002). The revised 2002 California probabilistic seismic hazard maps.

Frankel, A. D., Petersen, M. D., Mueller, C. S., Haller, K. M., Wheeler, R. L., Leyendecker, E. V., et al. (2002). Documentation for the 2002 update of the national seismic hazard maps (Open-File Report - U. S. Geological Survey No. OF 02-0420). Reston: U. S. Geological Survey.

Grant, L. B., & Sieh, K., (1993). Stratigraphic evidence for seven meters of dextral slip on the San Andreas Fault during the 1857 earthquake in the Carrizo Plain. Bulletin of the Seismological Society of America, 83(3), 619-635.

Grant, L. B., & Sieh, K., (1994). Paleoseismic evidence of clustered earthquakes on the San Angreas fault in the Carrizo Plain, California. Journal of Geophysical Research, 99(B4), 6819-6841.

Fumal, T. E., Rymer, M. J., & Seitz, G. G., (2002). Timing of large earthquakes since A. D. 800 on the Mission Creek strand of the San Andreas Fault Zone at Thousand Palms Oasis, near Palm Springs, California. Bulletin of the Seismological Society of America, 92(7), 2841-2860.

Fumal, T. E., Pezzopane, S. K., Weldon, R. J., II, & Schwartz, D. P. (1993). A 100-year average recurrence-interval for the San Andreas fault at Wrightwood, California. Science, 259(5092), 199-203.

Fumal, T. E., Weldon, R. J., II, Biasi, G. P., Dawson, T. E., Seitz, G. G., Frost, & W. T., Schwartz, D. P. (2002). Evidence for large earthquakes on the San Andreas fault at the Wrightwood, California, paleoseismic site: A. D. to present. Bulletin of the Seismological Society of America, 92(7), 2726-2760.

Gurrola, L. D., & Rockwell, T. K. (1996). Timing and slip for prehistoric earthquakes on the Superstition Mountain fault, Imperial Valley, southern California. Journal of Geophysical Research, 101(B3), 5977-5985.

Hall, N. T. (1986). Holocene slip rates on the San Andreas fault in northern, California. Paper presented at the The National Earthquake Prediction Evaluation Council and The San Francisco Bay Region Special Study Areas Workshop, February 26-March 1, 1986, Menlo Park, California.

Harden, J. W., & Matti, J. C. (1989). Holocene and late Pleistocene slip rates on the San Andreas fault in Yucaipa, California, using displaced alluvial-fan deposits and soil chronology. Geological Society of America Bulletin, 101(9), 1107-1117.

Jacoby, G. C., Sheppard, P. R., & Sieh, K. E. (1988). Irregular recurrence of large earthquakes along the San Andreas fault: evidence from trees. Science, 241(4862), 196-199.

Keller, E. A., Bonkowski, M. S., Korsch, R. J., & Shlemon, R. J. (1982). Tectonic geomorphology of the San Andreas fault zone in the southern Indio Hills, Coachella Valley, California. Geological Society of America Bulletin, 93(1), 46-56.

Kendrick, K. J., Morton, D. M., Wells, S. G., & Simpson, R. W., (2002). Spatial and temporal deformation along the Northern San Jacinto Fault, Southern California: Implications for slip rates. Bulletin of the Seismological Society of America, 92(7), 2782-2802.

Lienkaemper, J. J., Dawson, T. E., Personius, S. F., Seitz, G. G., Reidy, L. M., & Schwartz, D. P. (2002). A record of large earthquakes on the southern Hayward fault for the Past 500 Years. Bulletin of the Seismological Society of America, 92(7), 2637-2658.

Lienkaemper, J. J., Dawson, T. E., Personius, S. F., Seitz, G. G., Reidy, L. & M., Schwartz, D. P. (2002). A record of large earthquakes on the Southern Hayward fault for the past 500 years. Bulletin of the Seismological Society of America, 92(7), 2637-2658.

Lindvall, S. C., & Rockwell, T. K. (1995). Holocene activity of the Rose Canyon fault zone in San Diego, California. Journal of Geophysical Research, 100(B12), 24,121-124,132.

Lindvall, S. C., Rockwell, T. K., Dawson, T. E., Helms, J. G., & Bowman, K. W. (2002). Evidence for two surface ruptures in the past 500 years on the San Andreas fault at Frazier Mountain, California. Bulletin of the Seismological Society of America, 92(7), 2689-2703.

McGill, S., Dergham, S., Barton, K., Berney-Ficklin, T., Grant, D., Hartling, C., Hobart, K., Minnich, R., Rodriguez, M., Runnerstrom, E., Joanna, R., Schmoker, K., Stumfall, M., Townsend, J., & Williams, J., (2002). Paleoseismology of the San Andreas Fault at the Plunge Creek, near San Bernardino, Southern California. Bulletin of the Seismological Society of America, 92(7), 2803-2840.

Meisling, K. E., & Sieh, K. E. (1980). Disturbance of trees by the 1857 Fort Tejon Earthquake, California. Journal of Geophysical Research, 85(B6), 3225-3238.

Niemi, T. M., & Hall, N. T. (1992). Late Holocene slip rate and recurrence of great earthquakes on the San Andreas fault in Northern California. Geology, 20(3), 195-198.

Noller, J. S., Lettis, W. R., & Simpson, G. D. (1994). Seismic archaeology: using human prehistory to date paleoearthquakes and assess deformation rates of active fault zones. Paper presented at the Workshop on Paleoseismology, 18-22 September 1994, Marshall, California.

Petersen, M. D., Bryant, W. A., Cramer, C. H., Cao, T., Reichle, M. S., Frankel, A. D., et al. (1996). Probabilistic seismic hazard assessment for the state of California (Open-File Report - U. S. Geological Survey No. OF 96-0706). Reston: U. S. Geological Survey.

Rockwell, K. T., Lindvall, S., Herzberg, M., Murbach, D., Dawson, T., & Berger, G., (2000). Paleoseismology of the Johnson Valley, Kickapoo, and Homestead Valley Faults: Clustering of Earthquakes in the Eastern California Shear Zone. Bulleting of the Seismological Society of America, 90(5), 1200-1236.

Rundle, J. B., Rundle, P. B., Klein, W., de sa Martins, J., Tiampo, K. F., Donnellan, A., et al. (2002). GEM plate boundary simulations for the Plate Boundary Observatory: a program for understanding the physics of earthquakes on complex fault networks via observations, theory and numerical simulation. Pure and Applied Geophysics, 159(10), 2357-2381.

Seitz, G. G., Biasi, G. P., & Weldon, R. J., II. (2000). An improved paleoseismic record of the San Andreas fault at Pitman Canyon. AGU Reference Shelf, 4, 563-566.

Shifflett, H., Gray, M. G., Grannell, R., & Ingram, B. L. (2002). New evidence on the slip rate, renewal time, and late Holocene surface displacement, southernmost San Andreas Fault, Mecca Hills, California. Bulletin of the Seismological Society of America, 92(7), 2861-2877.

Sieh, K., Stuvier, M., Brillinger, D., (1989). A more precise chronology of earthquakes produced by the San Andreas Fault in Southern California. Journal of Geophysical Research, 94(B1), 603-623.

Sieh, K. E., & Williams, P. L., (1990). Behavior of the southernmost San Andreas fault during the past 300 years. Journal of Geophysical Research, 95(B5), 6629-6645.

Sims, J. D. (1994). Stream channel offset and abandonment and a 200-year average recurrence interval of earthquakes on the San Andreas Fault at Phelan Creeks, Carrizo Plain, California. Paper presented at the Workshop on Paleoseismology, 18-22 September 1994, Marshall, California.

Stone, E. M., Grant, L. B., & Arrowsmith, J. R. (2002). Recent rupture history of the San Andreas Fault southeast of Cholame in the northern Carrizo Plain, California. Bulletin of the Seismological Society of America, 92(3), 983-997.

Thomas, A. P., & Rockwell, T. K. (1996). A 300- to 550-year history of slip on the Imperial Fault near the U.S.-Mexico border: Missing slip at the Imperial fault bottleneck. Journal of Geophysical Research, 101(B3), 5987-5997.

Young, J. J., Arrowsmith, J., R., Colini, L., Grant, L. B., & Gootee, B., (2002). Three-Dimensional Excavation and Recent Rupture History along the Cholame Segment of the San Andreas Fault. Bulletin of the Seismological Society of America, 92(7), 2670-2688.

Weldon, R. J., II, Fumal, T., E., Powers, T., J., Pezzopane, S., K., Scharer, K., M., & Hamilton, J., C., (2002). Structure and Earthquake Offsets on the San Andreas Fault at the Wrightwood, California, Paleoseismic Site. Bulletin of the Seismological Society of America, 92(7), 2704-2725.

Weldon, R. J., II, & Sieh, K. E. (1985). Holocene rate of slip and tentative recurrence interval for large earthquakes on the San Andreas Fault, Cajon Pass, Southern California. Geological Society of America Bulletin, 96(6), 793-812.

Wesnousky, S. G., Prentice, C. S., Sieh, K. E., (1991). An offset Holocene stream channel and rate of slip along the northern reach of the San Jacinto fault zone, San Bernardino Valley, California. Geological Society of America Bulletin, 103, 700-709.

Appendix B – Database Schema

In the following, we discuss the database schema with SQL statements. These statements have actually been used to create tables in the database.

SQL statements defining the schema for QuakeTables:

```
create table FAULT

    (Fault_id int not null,

    FaultName char(255) not null,

    StrandName char(255),

    primary key(Fault_id, FaultName) );

create table REFERENCE

    (InterpId INT NOT NULL auto_increment,

    Author1 char(255),

    Author2 char(255),

    Author3 char(255),

    Author4 char(255),

    Author5 char(255),

    Publication char(255),

    Year int,

    Title char(255),

    Volume char(255),

    Number char(255),

    Pages char(255),

    Comment char(255),
```

```
    Identifier char(255) ,
    Author6 char(255) ,
    Author7 char(255) ,
    Datum char(255) ,
    SiteLat1 double ,
    SiteLon1 double ,
    SiteLat2 double ,
    SiteLon2 double ,
     primary key(InterpId) );
```

create table SEGMENT

```
     (Fault_id int not null,
     FaultName char(255) not null,
     InterpId INT NOT NULL,
     SegmentId int,
     SegmentName char(255),
    Strike double  ,
    Dip double  ,
    Depth double  ,
    Width double  ,
    LatStart double  ,
    LatEnd double  ,
    LonStart double  ,
    LonEnd double  ,
    LastBreak1 double  ,
    LastBreak2 double ,
    LastBreak3 double ,
    Friction double ,
```

ObsType int(11) ,

AveRecurr1 double ,

AveRecurr2 double ,

AveRecurr3 double ,

MinRecurr1 double ,

MinRecurr2 double ,

MinRecurr3 double ,

MaxRecurr1 double ,

MaxRecurr2 double ,

MaxRecurr3 double ,

AveSlip1 double ,

AveSlip2 double ,

AveSlip3 double ,

MinSlip1 double ,

MinSlip2 double ,

MinSlip3 double ,

MaxSlip1 double ,

MaxSlip2 double ,

MaxSlip3 double ,

AveDipSlip1 double ,

AveDipSlip2 double ,

AveDipSlip3 double ,

MinDipSlip1 double ,

MinDipSlip2 double ,

MinDipSlip3 double ,

MaxDipSlip1 double ,

MaxDipSlip2 double ,

MaxDipSlip3 double ,

AveStrikeSlip1 double ,

AveStrikeSlip2 double ,

AveStrikeSlip3 double ,

MinStrikeSlip1 double ,

MinStrikeSlip2 double ,

MinStrikeSlip3 double ,

MaxStrikeSlip1 double ,

MaxStrikeSlip2 double ,

MaxStrikeSlip3 double ,

AveDipRate1 double ,

AveDipRate2 double ,

AveDipRate3 double ,

MinDipRate1 double ,

MinDipRate2 double ,

MinDipRate3 double ,

MaxDipRate1 double ,

MaxDipRate2 double ,

MaxDipRate3 double ,

AveStrikeRate1 double ,

AveStrikeRate2 double ,

AveStrikeRate3 double ,

MinStrikeRate1 double ,

MinStrikeRate2 double ,

MinStrikeRate3 double ,

MaxStrikeRate1 double ,

MaxStrikeRate2 double ,

```
MaxStrikeRate3 double ,
NationalFaultId int(11)  ,
FADFaultId int(11)  ,
Geometry char(255)  ,
 Length1 double  ,
 Length2 double  ,
 Length3 double  ,
 RateRank char(255)  ,
 Mmax double  ,
 CharRate double  ,
 DownDipWidth1 double  ,
 DownDipWidth2 double  ,
 DownDipWidth3 double  ,
 Ruptop double  ,
 Rupbot double  ,
 Rake double  ,
 Dip double  ,
 Daz double  ,
 LatN double  ,
 LatS double  ,
 LonN double  ,
 LonS double  ,
Flag char(255)  ,
 Foreign key (Fault_id) references FAULT(Fault_id),
 Foreign key (FaultName) references FAULT(FaultName),
 Foreign key (InterpId) references REFERENCE(Interpid) );
```

create table LREFERENCE

```
(InterpId INT NOT NULL auto_increment,
Author1 char(255),
Author2 char(255),
Author3 char(255),
Author4 char(255),
Author5 char(255),
Publication char(255),
Year int,
Title char(255),
Volume char(255),
Number char(255),
Pages char(255),
Comment char(255),
primary key(InterpId) );
```

create table LAYER

```
(InterpId INT NOT NULL,
LayerId int not null,
LayerName char(255),
LatOrigin float,
LonOrigin float,
Datum char(255),
OriginX float,
OriginY float,
OriginZ float,
Length float,
Width float,
```

Depth float,

LameLambda float,

LameLambdaUnits char(255),

LameMu float,

LameMuUnits char(255),

Viscosity float,

ViscosityUnits char(255),

ViscosityExponent float,

primary key(InterpId));

Appendix C

Database as of May 7, 2004

Sources of data are listed in Appendix B. Additional data will be added. Fault segments are from CGS and Virtual California. Only CGS segments recognized by Petersen et al., 1996 and Rundle et al., 2002 are included at this time.

Limitations

In order to benefit geoscience and earthquake science, provisions should be made for additional access to the abundant data in QuakeTables. The hindrance to the provision of semi-public access is the availability of resources. Server(s) with high computing power is/are required. A full database management system implemented with access and concurrency control is another key element required. Finally, in order to effectively support sharing of QuakeTables data with heterogeneously developed applications, the combination and/or comparison with other related data in other “databases” must be supported by descriptions of the semantics of the data (ontologies). We did produce a simple extended entity-relationship database schema for QuakeTables, as a starting point for ontology. Much of our immediate future work focuses on addressing this additional interoperation requirement.