ONLINE INFORMATION SYSTEM FOR *SALMONELLA* AND *CAMPYLOBACTER*

Dr. Paul Juell and Chitra Vijayakumar
Computer Science Department
North Dakota State University
IACC 258
Fargo, ND 58105
paul.juell@ndsu.edu
chitra.vijayakumar@ndsu.edu

Dr. Catherine M. Logue
Department of Veterinary &
Microbiological Sciences
North Dakota State University
130A Van Es Hall
Fargo, ND 58105
catherine.logue@ndsu.edu

Abstract

We built a tool to develop database designs from spreadsheets. Data collected from microbiological samples from two turkey-processing plants were stored in the computer as spreadsheets. We found we needed this tool because the current use of spreadsheet was creating problems. We needed web access to the data in the spreadsheet and computer performance suffered because of treating a spreadsheet as a database. This tool will be helpful in converting spreadsheets to develop a database design, especially in the biological field. Using this tool we have developed a web-based database system for storing and retrieving information about two foodborne pathogens, *Salmonella* and *Campylobacter*. We named the database system ISSAC (Information System for *Salmonella* And *Campylobacter*). This system will reduce the time from days and hours to minutes for making tedious strain comparisons as information retrieval has been made easier than spreadsheets.

1 Introduction

We have built a tool that helps in constructing Entity-relationship diagrams from spreadsheets. The motivation for building this tool stems from one of the research projects in the Food Safety Microbiology Laboratory at North Dakota State University, Fargo, North Dakota. This project involves the detection and isolation of two food-borne pathogens, *Campylobacter* and *Salmonella*, on processed poultry. The aim of the project was to determine the influence of farm practices on the incidence of drug resistant *Salmonella* and *Campylobacter* and their potential for entering the food chain in processed product, in this case turkeys. Different strains of the above mentioned two organisms were isolated and their resistance against different antibiotics has been studied. The Minimum Inhibitory Concentration for different antibiotics has been measured.

The data collected so far are stored in the computer as spreadsheets. These types of databases are useful, when the data size is small. On the other hand as the data size increases numerous spreadsheets are required. When the situation requires data from different spreadsheets, this solution becomes inadequate. It is hard to retrieve specific information from a large pool of data. Retrieving and comparing data from multiple spreadsheets lead to errors and is also time-consuming.

When an investigator starts collecting data for a research project, a basic decision to be made involves the method of storing those collected data. The options range from spreadsheets to desktop database applications to server-based systems. Each option has its own advantages and limitations. Some of the factors that determine the right choice are: the amount of data to be stored, and the number of users concurrently accessing the data. Web-access of data is also desirable, especially in an academic setting. With the advent of internet and sophisticated database management systems, these problems can be solved, by developing software for the required purposes.

Data stored in a spreadsheet basically is a single-user file. In addition handling multiple spreadsheets is difficult when one wants to compare the variables located in different spreadsheets. To store large amounts of data, it is best to use a database. This is especially true in circumstances where two or more users share the information. Consequently, there is a need to build a tool to develop database designs from the already existing spreadsheets. Often biological data are stored in spreadsheets. Our tool is useful to create database designs (ER diagrams) from spreadsheet storage.

Using this tool, we have created the database design and developed a web-based database system for storing and retrieving information about two food borne pathogens, *Salmonella* and *Campylobacter*. We named the system ISSAC (Information System for *Salmonella* And *Campylobacter*).

1.1 Organization of Paper

The remainder of the paper is organized as follows: Section 2 introduces the importance of the food safety project and the need for an automated System. Section 3 analyzes the advantages of databases over spreadsheets. Section 4 describes the protocol used for constructing ER diagram from spreadsheets. Section 5 describes the requirements and architecture of the online information system. Section 6 walks through the application. Section 7 describes results. This is followed by a summary and the references used.

2 Importance of the Food Safety Project and the Need for an Automated System

Salmonella and *Campylobacter* are common food-borne pathogens associated with human illness. Different strains of the two pathogens were isolated from two different turkey-processing plants. The plants get birds from different farms. The problem arises when one bird from one of the farms is infected. This can contaminate other birds that are brought into the plant. There are chances that this contamination can be carried over to the finished product. Cross contamination between carcasses can occur during slaughtering, defeathering, carcass washing and during further processing. So there is potential for these pathogens to enter the food chain. *Salmonella* and *Campylobacter* pose major public health problems. Microbiological samples were collected from different points during processing. Some drug resistant strains were found in the finished product.

The use of antibiotics in animal feed is associated with the carriage of resistant bacteria in animals. The resistant bacterial isolates often exhibit cross-resistance to antimicrobial agents that are important for the management of infections in humans. There is found to be strong link between antibiotic-supplemented animal feeds and increases in human morbidity and mortality owing to drug-resistant pathogens (5).

If we can compare across many different parameters such as drug resistance patterns, sources, serotypes etc. we can look for these patterns and relate them to farms, or plants. These relationships would be helpful in determining how the farm practices influence the ultimate risk to the human host should they get exposed to a drug resistant strain.

In order to do the comparison studies, different variables have to be retrieved from the pool of data based on a single parameter. Manual retrieval is time-consuming and can also lead to errors. An automated on-line system would be helpful for this purpose. Better organization of research data and an easier to use system can help in a safer food supply.

Having decided the need for a database management system, let us examine the problems of using spreadsheets as a database, computer performances associated with it, and the effectiveness of using databases.

3 Advantages of Databases over Spreadsheets

Many biologists are still using spreadsheets to store their data even though the data could be better managed in a database. Of course, the basic functions of both spreadsheets and databases are to store and manipulate data. However, there are some distinct disadvantages in using spreadsheets, especially when the data volume increases and web-access of data is essential.

To store large amounts of raw data, it is best to use a database. This is especially true in circumstances where two or more users share the information. Databases require little or no duplication of data between information tables, and changes made to the data do not corrupt the programming (like at the cell level of a spreadsheet where calculations are running). Databases offer better security to restrict users from accessing privileged information, and from changing coded information in the programming. Furthermore, the two most important benefits gained by using a database are the capacity to increase data integrity, and the ease of reporting and sharing data (11).

With spreadsheets, data has to be copied over and over again to maintain it in separate data files. Moreover, spreadsheets are stand-alone applications that are located entirely on the user's hard drive. This limits the number of concurrent users. The disadvantages of spreadsheets include: the inability to efficiently identify data errors, the lack of detailed sorting and querying abilities, and sharing violations among users. Additionally, spreadsheets are restricted to a finite number of records, and can require a large amount of hard-drive space for data storage. Spreadsheets generally do not offer the ability to structure and label data items as fully as a database and usually do not offer the ability to query the database. In general, a spreadsheet is a much simpler program than a database program.

To enable web-access of data, a client/server paradigm is essential, which can require relational databases (2). However, not all relational databases are built on the client/server paradigm and spreadsheets are not a form of relational databases. Relational databases depend on a sound theoretical basis in relational algebra. They maintain, enforce and use relationships between data. Relational databases manipulate only tables and the result of all operations are also tables. The tables are sets, which are themselves sets of rows and columns. Some of the advantages of relational databases include, faster and random access to data, extraction of sets of data that fit certain criteria, and built-in mechanisms for dealing with concurrent access (7).

To access the data from anywhere and at any time, we might want a relational database server. A database server is a specialized process that manages the database itself. The applications are clients to the database server and they never manipulate the database directly, but only make requests for the server to perform these operations (9). This allows the server to add many sophisticated features, such as transaction processing, recovery, backup, access control and etc without increasing the complexity of every application. The server also reduces the risk of data file corruption; because only the server writes to the database (a crash on any client machine will not leave unflushed buffers).

Since relational databases are required for better management and web-access of data, we need to build the tool for converting the already existing spreadsheet form of data to a relational database.

4 Protocol used for constructing ER diagrams from spreadsheets

When creating a database, we are actually modeling real-world items and relationships and storing information about those objects and relationships (10). Relational databases are made of relations, more commonly called tables. Each column in the table has unique name and contains different data. Each column has an associated data type (11). Entity-Relationship (ER) diagrams are simple and clear method of expressing the design of a database (3). We first designed an ER diagram. Based on the ER diagram, we created tables, which led to the development of a relational database. Relations resulting from the translation of an ER diagram will be more accurate than relations created directly by a designer (1).

In order to create a database for ISSAC, we need to create an ER diagram. The different steps we used for the process of constructing ER diagrams are given below:

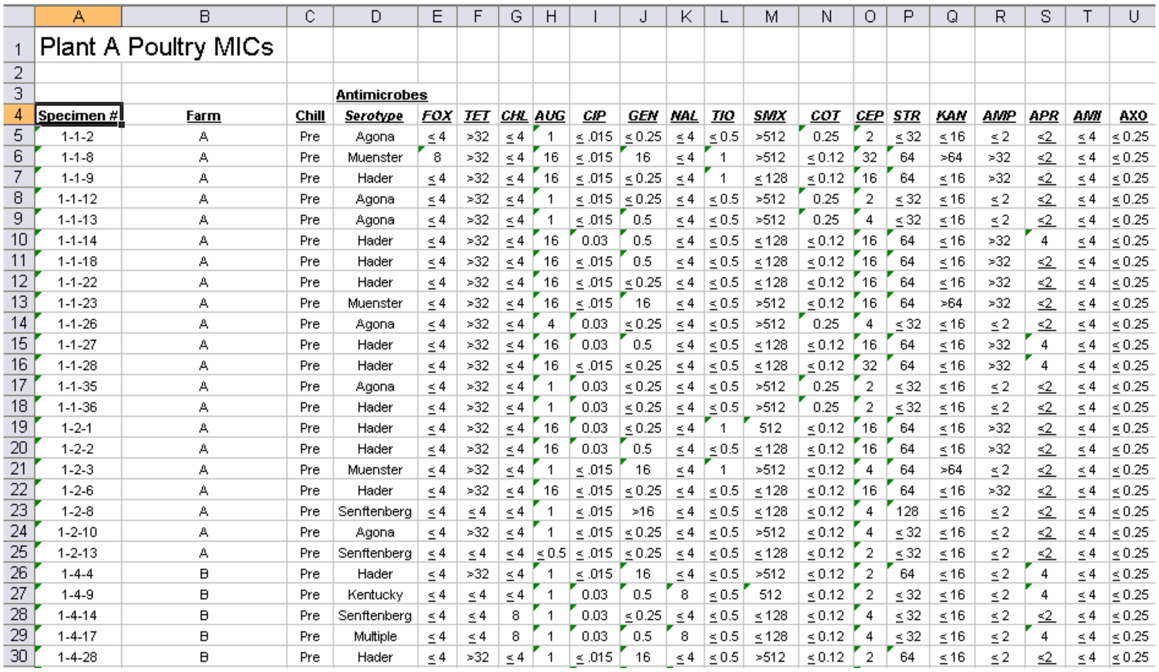
a. Analyze the spreadsheet data file.
b. Identify entities and the corresponding attributes. Usually the first column consists of unique values and forms an entity. If not, create an ID that uniquely identifies each record in the file. The other columns could form the attributes for that entity. Make sure the values in the other columns depend entirely on the first column.
c. If any of the other columns has a composite field, break that field down into its constituent parts. This composite field can be another entity and the constituent parts would be their attributes.
d. After identifying the entities, link the entities with the relationship among them.
e. Discuss with the investigator to improve the database design.
f. Convert entities into tables, identify key attributes for each table, and create links between tables.
g. Complete the ER diagram with the tables and links developed.

The following sections would give a general outline of this process of creating an ER diagram for a relational database from a spreadsheet:

4.1 Identify Entities and Attributes

The fist step in designing a database would be to identify entities in the spreadsheet data files. An entity is an object that is distinguishable from other objects (6). It is often useful to identify a collection of similar entities. An entity is described using a set of attributes. All entities in a given entity set have the same attributes; this is essentially what is meant by similar (3).

Usually, the first field name in the spreadsheet consists of unique values (Fig. 1). If not, identify the field that has unique values. This could be an entity. The other fields in the spreadsheets would form the attributes for this entity. For example, the spreadsheet under study (named as Information System for *Salmonella* And *Campylobacter* or ISSAC) has Specimen Number as the first field and it has unique values. If we consider the entire system as a hierarchical data structure, Specimen Number could form the root node. The other fields, Farm (the place where it was collected from), Chill (the chill water that was used for carcass washing of birds), Serotype, and Antimicrobes form the attributes for the root node (Specimen Number), and these can be considered as the children nodes.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	Plant A Poultry MICs																				
2																					
3				Antimicrobes																	
4	Specimen #	Farm	Chill	*Serotype*	*FOX*	*TET*	*CHL*	*AUG*	*CIP*	*GEN*	*NAL*	*TIO*	*SMX*	*COT*	*CEP*	*STR*	*KAN*	*AMP*	*APR*	*AMI*	AXO
5	1-1-2	A	Pre	Agona	≤4	>32	≤4	1	≤.015	≤0.25	≤4	≤0.5	>512	0.25	2	≤32	≤16	≤2	≤2	≤4	≤0.25
6	1-1-8	A	Pre	Muenster	8	>32	≤4	16	≤.015	16	≤4	1	>512	≤0.12	32	64	>64	>32	≤2	≤4	≤0.25
7	1-1-9	A	Pre	Hader	≤4	>32	≤4	16	≤.015	≤0.25	≤4	1	≤128	≤0.12	16	64	≤16	>32	≤2	≤4	≤0.25
8	1-1-12	A	Pre	Agona	≤4	>32	≤4	1	≤.015	≤0.25	≤4	≤0.5	>512	0.25	2	≤32	≤16	≤2	≤2	≤4	≤0.25
9	1-1-13	A	Pre	Agona	≤4	>32	≤4	1	≤.015	0.5	≤4	≤0.5	>512	0.25	4	≤32	≤16	≤2	≤2	≤4	≤0.25
10	1-1-14	A	Pre	Hader	≤4	>32	≤4	16	0.03	0.5	≤4	≤0.5	≤128	≤0.12	16	64	≤16	>32	4	≤4	≤0.25
11	1-1-18	A	Pre	Hader	≤4	>32	≤4	16	≤.015	0.5	≤4	≤0.5	≤128	≤0.12	16	64	≤16	>32	≤2	≤4	≤0.25
12	1-1-22	A	Pre	Hader	≤4	>32	≤4	16	≤.015	≤0.25	≤4	≤0.5	≤128	≤0.12	16	64	≤16	>32	≤2	≤4	≤0.25
13	1-1-23	A	Pre	Muenster	≤4	>32	≤4	16	≤.015	16	≤4	≤0.5	>512	≤0.12	16	64	>64	>32	≤2	≤4	≤0.25
14	1-1-26	A	Pre	Agona	≤4	>32	≤4	4	0.03	≤0.25	≤4	≤0.5	>512	0.25	4	≤32	≤16	≤2	≤2	≤4	≤0.25
15	1-1-27	A	Pre	Hader	≤4	>32	≤4	16	0.03	0.5	≤4	≤0.5	≤128	≤0.12	16	64	≤16	>32	4	≤4	≤0.25
16	1-1-28	A	Pre	Hader	≤4	>32	≤4	16	≤.015	≤0.25	≤4	≤0.5	≤128	≤0.12	32	64	≤16	>32	4	≤4	≤0.25
17	1-1-35	A	Pre	Agona	≤4	>32	≤4	1	0.03	≤0.25	≤4	≤0.5	>512	0.25	2	≤32	≤16	≤2	≤2	≤4	≤0.25
18	1-1-36	A	Pre	Hader	≤4	>32	≤4	1	0.03	≤0.25	≤4	≤0.5	>512	0.25	2	≤32	≤16	≤2	≤2	≤4	≤0.25
19	1-2-1	A	Pre	Hader	≤4	>32	≤4	16	0.03	≤0.25	≤4	1	512	≤0.12	16	64	≤16	>32	≤2	≤4	≤0.25
20	1-2-2	A	Pre	Hader	≤4	>32	≤4	16	0.03	0.5	≤4	≤0.5	≤128	≤0.12	16	64	≤16	>32	≤2	≤4	≤0.25
21	1-2-3	A	Pre	Muenster	≤4	>32	≤4	1	≤.015	16	≤4	1	>512	≤0.12	4	64	>64	≤2	≤2	≤4	≤0.25
22	1-2-6	A	Pre	Hader	≤4	>32	≤4	16	≤.015	≤0.25	≤4	≤0.5	≤128	≤0.12	16	64	≤16	>32	≤2	≤4	≤0.25
23	1-2-8	A	Pre	Senftenberg	≤4	≤4	≤4	1	≤.015	>16	≤4	≤0.5	≤128	≤0.12	4	128	≤16	≤2	≤2	≤4	≤0.25
24	1-2-10	A	Pre	Agona	≤4	>32	≤4	1	≤.015	≤0.25	≤4	≤0.5	>512	≤0.12	4	≤32	≤16	≤2	≤2	≤4	≤0.25
25	1-2-13	A	Pre	Senftenberg	≤4	≤4	≤4	≤0.5	≤.015	≤0.25	≤4	≤0.5	≤128	≤0.12	2	≤32	≤16	≤2	≤2	≤4	≤0.25
26	1-4-4	B	Pre	Hader	≤4	>32	≤4	1	≤.015	16	≤4	≤0.5	>512	≤0.12	2	64	≤16	≤2	4	≤4	≤0.25
27	1-4-9	B	Pre	Kentucky	≤4	≤4	≤4	1	0.03	0.5	8	≤0.5	512	≤0.12	2	≤32	≤16	≤2	4	≤4	≤0.25
28	1-4-14	B	Pre	Senftenberg	≤4	≤4	8	1	0.03	≤0.25	≤4	≤0.5	≤128	≤0.12	4	≤32	≤16	≤2	≤2	≤4	≤0.25
29	1-4-17	B	Pre	Multiple	≤4	≤4	8	1	0.03	0.5	8	≤0.5	≤128	≤0.12	4	≤32	≤16	≤2	4	≤4	≤0.25
30	1-4-28	B	Pre	Hader	≤4	>32	≤4	1	≤.015	16	≤4	≤0.5	>512	≤0.12	2	64	≤16	≤2	≤2	≤4	≤0.25

Figure 1: A screen print view of the spreadsheet under study

The column header Specimen Number has unique values and can be considered as an entity. However, a closer look reveals that Specimen Number is actually identification for the members of a larger set. The larger set would be the two bacteria, *Salmonella* and *Campylobactor* of the Food Safety Project. So for purposes of convenience, the larger set can be named as Bacteria and can be treated as an entity for the system.

The attribute Antimicrobes is a composite field and has many constituent parts such as FOX, TET, etc. (Fig. 3). Since the attribute Antimicrobes has many children nodes, this could be another entity for the ISSAC System. The constituent parts are names of different antimicrobes. This could be given a collective name as antimicrobeName and forms an attribute for the entity Antimicrobes.

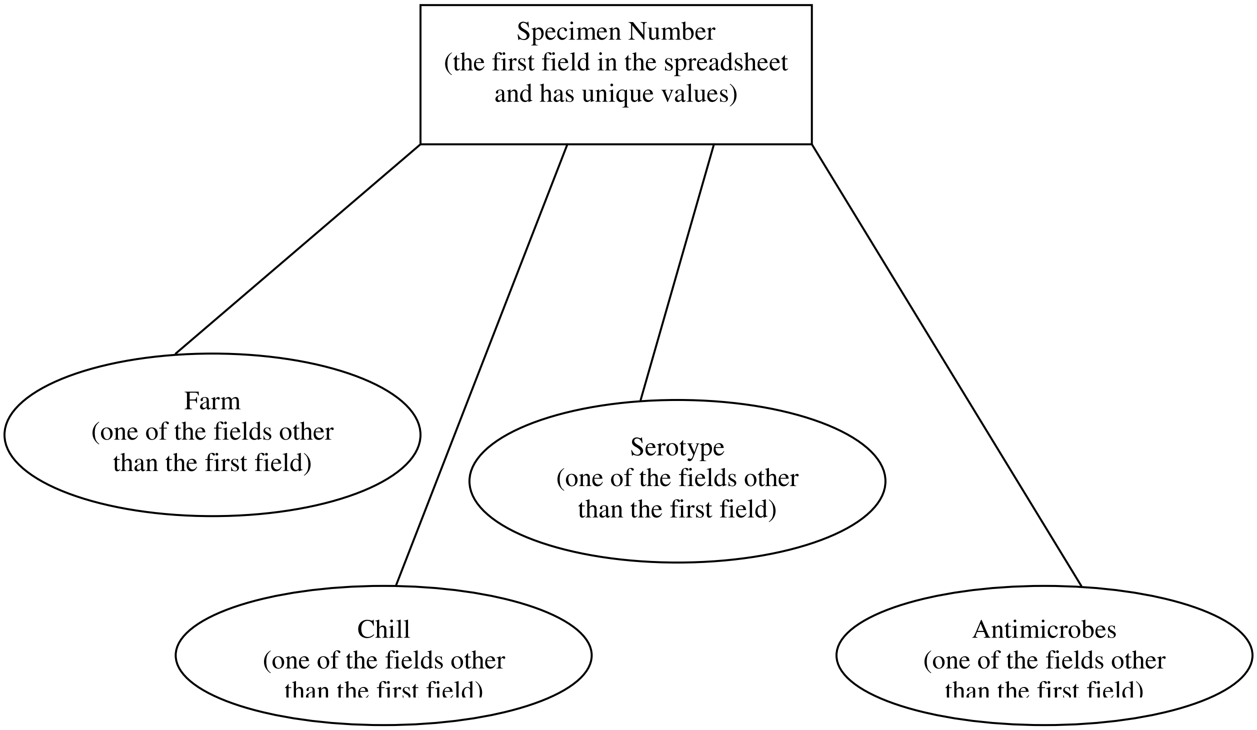


Figure 2: The spreadsheet expressed as a hierarchical data structure

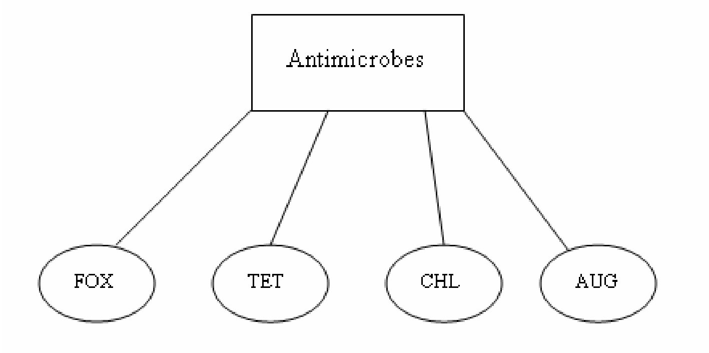


Figure: 3 One of the field names broken down into its constituent parts.

The next step would be to set relationships among the entities found.

4.2 Set Relationships among Entities

A relationship is an association among two or more entities. As with entities, a set of similar relationships is called a relationship set. A relationship can also have descriptive attributes. Descriptive attributes are used to record information about the relationship, rather than about the participating entities (3). Thus, for a given entity pair (entity1, entity2), we cannot have more than one descriptive attribute. In our example, a relationship is set between the two entities, Bacteria and Antimicrobes. Every Bacteria is resistant to an Antimicrobe. This can be included into the system by adding an attribute, MIC (Minimum Inhibitory Concentration), to ResistantTo.

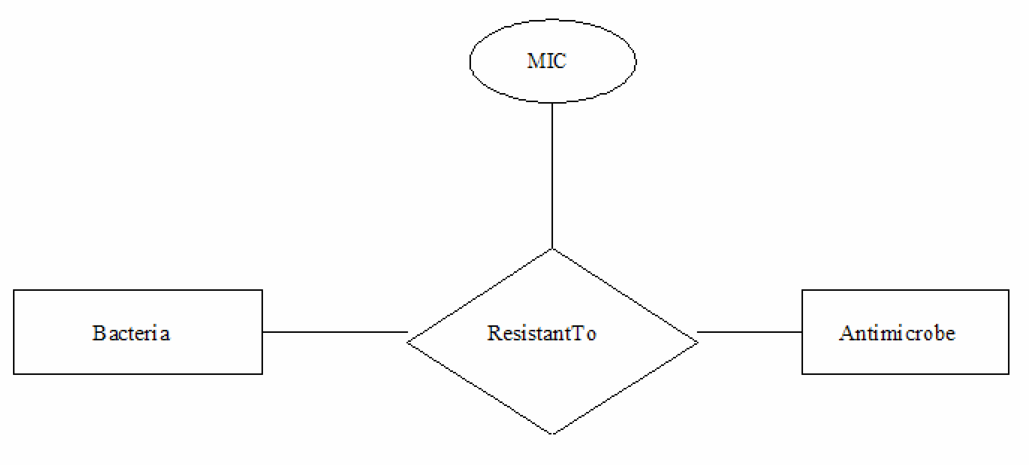


Figure: 4 Relationship among entities

Thus every Bacteria, Antimicrobe pair has one MIC value. This forms a simple Entity-Relationship model showing the relationship between the entities found.

4.3 Discuss with the Investigator

A spreadsheet data set contains the results of a project done by an investigator. Most of the time a spreadsheet has footnotes that correspond to some fields or column headers. A discussion with the investigator would be helpful in adding or deleting any attributes to the entities found. In our example, the spreadsheet has some footnotes about the abbreviations used and date of visit.

4.4 Modify the ER Model

Modify the ER model by adding more attributes or deleting some attributes or changing names for attributes for each entity. In our example (ISSAC), discussion and footnotes in the spreadsheet resulted in adding more attributes to the Bacteria entity. The attributes added include dateVisited and plant; the attribute 'chill' was modified to 'controlPoint'; and the attribute Antimicrobes was modified to 'Antimicrobials'.

4.5 Convert Entities to Tables

Create a table for each entity. The attributes of each entity will form the fields for the corresponding table. Thus ISSAC will have two tables: 1. Bacteria, with specimenNumber, farm, controlPoint, serotype, dateVisited and plant as its attributes; 2. Antimicrobials, with antimicrobialName as its attribute. To save storage space with long names of antimicrobials, an ID was created to uniquely identify each antimicrobial.

4.6 Create Links between Tables

The tables created have to be linked with the descriptive attribute, which describes the relationship between the tables. In our example, it is the MIC which describes the ResistantTo relationship. The tables (Bacteria, Antimicrobials, MIC) in the database, ISSAC, can be given in a text form as shown below. The terms within braces represent the attributes for the corresponding table.
Bacteria (specimenNumber, farm, controlPoint, serotype, dateVisited, plant)
Antimicrobials (antimicrobialId, antimicrobialName)
MIC (specimenNumber, antimicrobialID, mic)

4.7 Identify Key Attributes

The next step is to identify the attribute or sets of attributes that uniquely identify an entity. In our example, specimenNumber uniquely identifies each record. So this forms the primary key for the table Bacteria. For the table Antimicrobials, antimicrobialId would be the primary key. For the MIC table, it is a set attributes that uniquely identifies a record. Hence, specimenNumber and antimicrobialId together form the primary key.

4.8 Complete the ER diagram

Complete the ER diagram with all the modifications. In our ISSAC example, the final ER diagram is shown below. The primary keys are underlined.

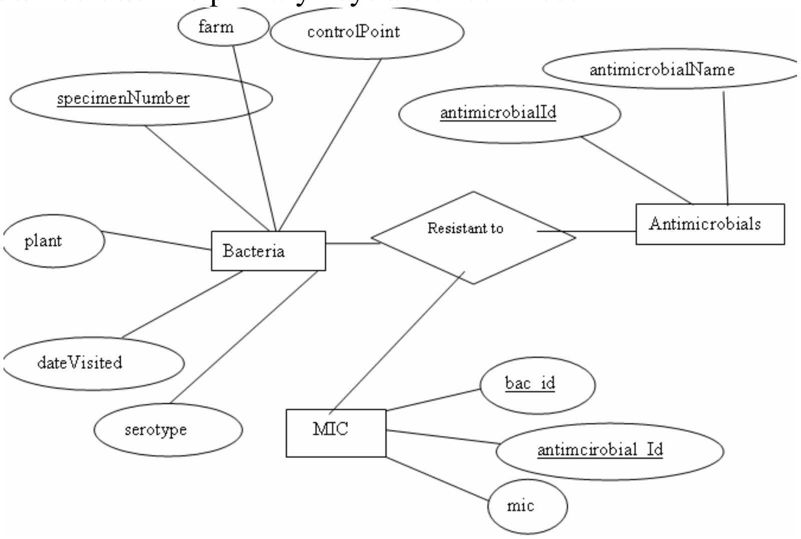


Figure 5: Complete ER diagram for ISSAC

5 Requirements and Architecture of the Online Information System

The following requirements were gathered from discussion with the principal investigator of the Food Safety research project. Here is a summary of the functional and systems requirements:

1. The Online Information System is primarily aimed for comparison studies in research projects.
2. The administrator would be the principal investigator of this research project.
3. Administrator can authorize, delete instructors.
4. Instructors include research staff working on this project.
5. Instructors can add, delete, modify, and search data in ISSAC.
6. Instructors can authorize, and delete students.
7. Students can search the database for information about two food borne pathogens*,* *Salmonella* and *Campylobacter* based on keywords. However, they cannot add, delete or modify any data.

We used a three-tier architecture (Fig.6) model for this online information system. At the base of this application is the database tier, consisting of the database management system that manages the database containing the data users create, delete, modify and query. Built on top of the database tier is the complex middle tier, which contains most of the application logic and communicates data between other tiers. On top is the client tier, usually web browser software that interacts with the application

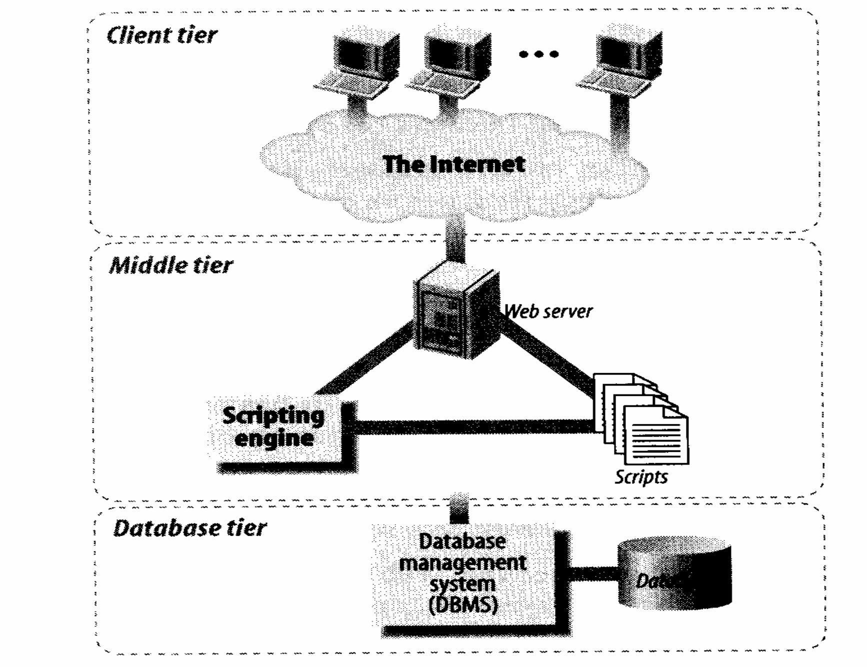


Figure 6: Three tier architecture

5.1 The Client Tier

Web browser software in the client tier processes and displays HTML resources, issues HTTP requests for resources, and processes HTTP responses. There are significant advantages in using a web browser as the client tier, including easy deployment and support on a wide range of platforms (8).

5.2 The Middle Tier

The majority of the application logic is in the middle tier. The client tier presents data to and collects data from the user; the database tier stores and retrieves the data. The middle tier serves most of the remaining roles that bring together the other tiers: it drives the structure and content of the data displayed to the user, and it processes input from the user as it is formed into queries on the database to read and write data. It also adds state management to the HTTP protocol.

The middle tier application logic integrates the Web with the database management system. In the application framework used in this tool, the components of the middle tier are a web server, a web scripting language, and the scripting language engine. A web server processes HTTP requests and formulates responses. In the case of web database applications, these requests are often for programs that interact with and underlying database management system (4). The web server that we used for our web database application is the Apache Software Foundation's Apache HTTP server.

We used the PHP scripting language as our middle tier scripting language. PHP is an open source project of the Apache Software Foundation and it is the most popular Apache HTTP server add-on module. PHP is particularly suited to web database applications because of its integration tools for the web and database environments. In particular, the flexibility of embedding scripts in HTML pages permits easy integration with the client tier. The database tier integration support is also excellent with more than 15 libraries available to interact with almost all popular database management systems. Moreover, PHP greatly reduces the complexity of using cgi-bin variables and connecting to databases, and allows programming in an easy-to-learn language within web pages (12).

5.3 The Database Tier

In a three-tier application, the database tier manages the data. The data management typically includes storage and retrieval of data, as well as managing updates, allowing simultaneous, or concurrent, access by more than one middle-tier processes, providing security, ensuring the integrity of data, and providing support services such as data backup. In many web database applications, these services are provided by a RDBMS system, and the data stored in a relational database. In the web database application developed, we used MySQL RDBMS to manage data. MySQL has a well-deserved reputation for speed, and it is particularly well designed for applications where retrieval of data is more common.

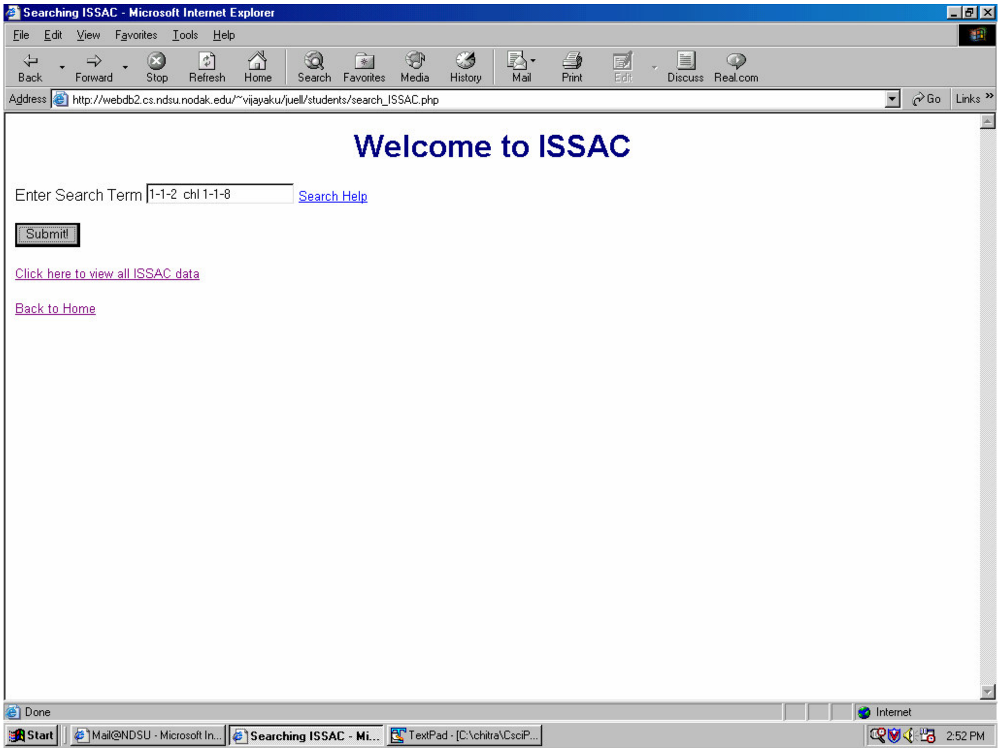
6 Walk through the Application

This web-based system (ISSAC) plays an important role in improving data storage, accessibility and data retrieval. The system has a user-friendly interface, which facilitates easy interaction for users with the system. The first page of this database is a login page where users can access through a user name and password. ISSAC is password-protected, as security is important for a web-based system. Users include administrators, instructors and students. Administrator can authorize or delete an instructor.

Instructors can log in to ISSAC through a user name and password. The next page allows them to access ISSAC, which in turn allows them to add, delete, modify or view any or all ISSAC data. They can also do a search based on a single or combination of keywords. Instructors can authorize or delete students.

Students can access the database for viewing all or data that match a single or combination of keywords. However, they can't make any changes to the database.

The purpose of the database is to provide an improved organization of data and to permit a variety of comparative studies based on key word searches (Fig.7 & 8) on the vast amount of data stored in the database.


Searching ISSAC - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites Media History Mail Print Edit Discuss Real.com
Address
Go Links
Search Help
Submit!
Done
Internet
Start
Mail@NDSU - Microsoft In...
Searching ISSAC - Mi...
TextPad - [C:\chitra\CsciP...
2:52 PM

http://webdb2.cs.ndsu.nodak.edu/~vijayaku/juell/students/search_ISSAC.php

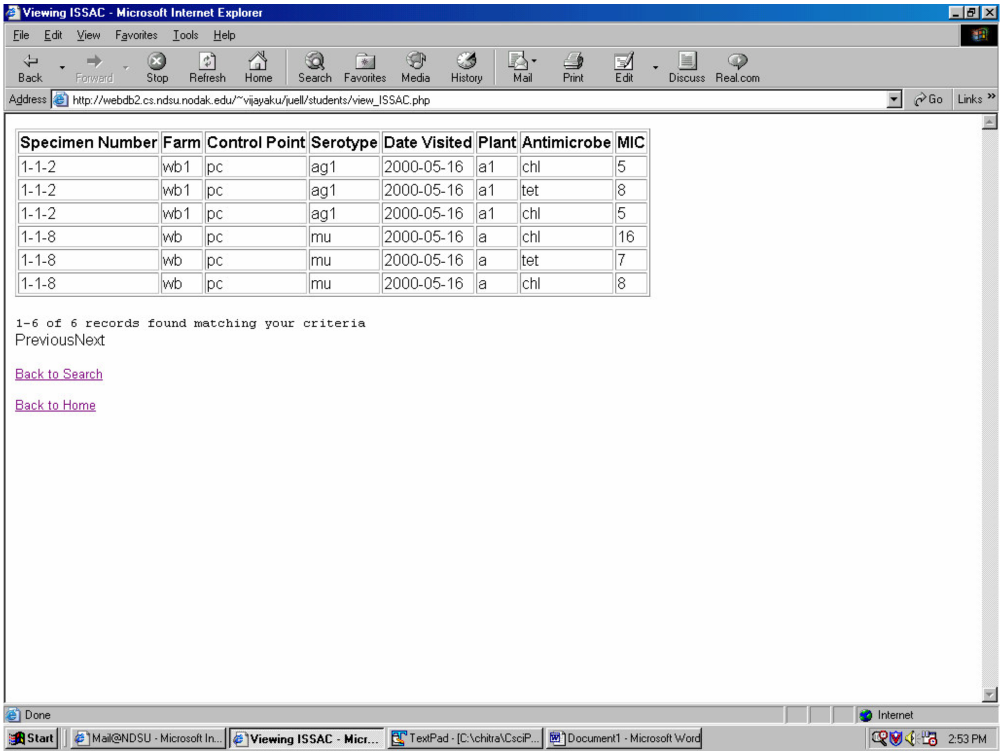
Welcome to ISSAC

Enter Search Term 1-1-2 chl 1-1-8

Click here to view all ISSAC data

Back to Home

Figure 7: Keyword search in ISSAC


Viewing ISSAC - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites Media History Mail Print Edit Discuss Real.com
Go Links
Done
Internet
Start
Mail@NDSU - Microsoft In...
Viewing ISSAC - Micr...
TextPad - [C:\chitra\CsciP...
Document1 - Microsoft Word
2:53 PM

Address http://webdb2.cs.ndsu.nodak.edu/~vijayaku/juell/students/view_ISSAC.php

Specimen Number	Farm	Control Point	Serotype	Date Visited	Plant	Antimicrobe	MIC
1-1-2	wb1	pc	ag1	2000-05-16	a1	chl	5
1-1-2	wb1	pc	ag1	2000-05-16	a1	tet	8
1-1-2	wb1	pc	ag1	2000-05-16	a1	chl	5
1-1-8	wb	pc	mu	2000-05-16	a	chl	16
1-1-8	wb	pc	mu	2000-05-16	a	tet	7
1-1-8	wb	pc	mu	2000-05-16	a	chl	8

```
1-6 of 6 records found matching your criteria
```

PreviousNext

Back to Search

Back to Home

Figure 8: Keyword search result in ISSAC

ISSAC's user interface connects to a web-server and accesses a mySQL database. The database uses normalized tables and integrates other programs to simplify and speed up data analysis. The results of a query, list of bacteria or farms or plants or serotypes or antimicrobials or MICs or a combination of one or more attributes are displayed in a table. The easy-to-use user interface makes it simple for the researchers to do a variety of otherwise complex SQL database queries in an easy and rapid manner.

This database system has replaced the spreadsheet files that were used before in the department. Many spreadsheets were used to store all the data. For comparison studies, they had to open all the spreadsheets and do the search manually. This database application has reduced the researchers' time significantly in retrieving information from the pool of data. Moreover, many users can access ISSAC concurrently to obtain information as opposed to spreadsheets, which are single-user files.

7 Results

We built a tool for constructing ER diagrams, which is the blueprint for developing a database, from spreadsheets. Using this tool we have developed a web-based database application. The fundamental job of a database application is to warehouse information—especially diverse information—and then enable users to access and reconfigure it in nearly any customized order. Although spreadsheets can do that job, too, they can't do it quite as well or as easily as databases. Also a spreadsheet is not really a database. We can store some data and do some calculations. We made an effort to convert spreadsheet forms of data into an on-line database. The protocol that we used would be helpful for anyone, especially in the biological field in the spreadsheet conversion process.

The web-based database application we developed using our tool is being used by the Department of Veterinary & Microbiological Sciences, North Dakota State University, Fargo, ND. Earlier the data was stored in spreadsheets. Instructors and students used to go to the laboratory contact the person in-charge, and obtain the information they needed. This is time-consuming for both especially for the laboratory personnel when a number of students are requesting information. Now, things are made easy. This new system will give widespread accessibility and time independence for its users.

8 Summary

We built a conversion tool for constructing ER diagrams from spreadsheets. ER diagrams are considered as the blueprint for developing database programs. We got the idea when we started developing a database program for the Veterinary & Microbiological Sciences Department at North Dakota State University, Fargo, North Dakota. We developed a protocol for the conversion process. This would be helpful for developing database programs from already existing spreadsheet form of storage, especially in biological field. Most research projects in biological field, involve applying different tests to different samples. Usually, the results of these tests are stored in spreadsheets. Using the protocol we have developed an online database program. The online database system we developed is now being used by the department. The system is very useful for all the members of the research project, including students. In an academic setting, this type of database system is useful as it provides time-independence; widespread accessibility; easy maintenance of large volume of data; and comparison of different data values.

9 References

1. Antony, S.R. and Batra, D., "CODASYS: A Consulting Tool for Novice Database Designers," ACM Sigmis Database, Vol. 33, No. 3, Aug. 2002

2. Bi, C., S.V. Vrbsky, and N. Jukic, "A Security Paradigm for Web Databases," Proceedings of the 37th annual Southeast regional conference, Apr. 1999

3. Dey, D., Storey, V.C., and Barron, T.M., "Improving Database Design through the Analysis of Relationships," ACM Transactions on Database Systems, Vol. 24, No. 4, Dec. 1999, pp. 453-486

4. Fraternali, P., "Tools and Approaches for Developing Data-Intensive Web Applications: A survey," ACM Computing Surveys, Vo. 31, No. 3, Sep. 1999, pp. 227-263

5. Hamer, D.H. and C.J. Gill, "From the farm to the kitchen table: the negative impact of antimicrobial use in animals on humans," Nutrition Reviews, Vol. 60,

no. 8, Aug. 2002, pp. 261-264

6. Rubenstem, W.B., "A Database Design for Musical Information," ACM SIGMOD Record, Proceedings of the 1987 ACM SIGMOD international conference on Management of Data, Vol. 16, No. 3, Dec. 1987, pp. 479-490

7. Russell, J., ""Yes, But Does It Scale?" Practical Considerations for Datbase-Driven Information Systems," Proceedings of the 19th annual international conference on computer documentation, Oct. 2001

8. Shadgar, B. and Holyer, I., "Adapting Databases and WebDAV Protocol," Proceedings of the 13th International Conference on World Wide Web, May 2004, pp. 612-620

9. Spiegler, I., "Automating Database Construction," ACM Sigmis Database, Vol. 14, No. 3, Apr. 1983, pp. 21-29

10. Storey, V.C., and Goldstein, R.C., "A Methodology for Creating User Views in Database Design," ACM Transactions on Database Systems, Vol. 13, No. 3, Sep. 1988, pp. 305-338

11. Turker, C. and M. Gertz, "Semantic integrity support in SQL: 1999 and commercial (object-) relational database management systems," The VLDB Journal, Vol. 10, 2001, pp. 241-269

12. Williams, H.E. and Lane, D. "Web Database Applications with PHP & MySQL," O'Reilly & Associates, Inc., 2002