

Integration of Database Systems

A Design Example and Implementation

Bereket Gebremarian, Angelina C. Rotundo

Abstract— As early as the 1960’s computerized methods to manage commercial data led to the development of database systems [1]. This paper evaluates the design and implementation of a database. It also looks at the performance of SQL statements (database queries) to search and retrieve data from the database. To further demonstrate the SQL statements, the paper discusses the design and implementation of a Graphical User Interface (GUI), which is designed as part of an online forum used to obtain user information and store it in a database.

Index Terms— Database, Database Design, Database-Management System (DBMS), MySQL, Forums, PHP, Apache Server, HTML, Cascading Style Sheets (CSS)



1 INTRODUCTION

A *database* is a collection of data arranged for ease of search and retrieval [1]. As technology and the internet have grown tremendously over the past several years, managing data in an organized and efficient way has become increasingly important. People, businesses and enterprises rely on databases to contain their data and information. A database-management system (*DBMS*), which is the collection of interrelated data and set of programs to access those data, must be designed in such a way as to store and retrieve the database information efficiently and conveniently [2]. The database design process is a crucial step that should be done very carefully.

2 DESIGN (*DATABASE DETAILS*)

Database design involves several steps. The first phase of the database design process is to identify the needs of the users who will be using the database. During this initial phase of the database design process, the database designers must take the time to fully understand the requirements and needs of the database users. At the end of this phase, the database designer should be able to fully understand and identify the user requirements. After the database designer identifies the user requirements, he or she will choose a database model. This phase is considered the conceptual-design phase and results in a conceptual schema of the database. During this phase the database designer takes the user requirements and develops a database schema or high-level overview of the enterprise. The key objective during the conceptual-design phase is to identify the data and their relationships.

“In terms of the relational model, the conceptual-design process involves decisions on *what* attributes we want to capture in the database and *how* to group these attributes to form the various tables” [2]. An entity-relationship (E-R) model is often used to show *how* to group the attributes to form the various tables [2]. Modeling the database helps us to visualize and model the system prior to implementation as well as to identify and manage risks in the system [3].

For this project, the Society of Computational Science contracted our team’s technical services to design and develop a database that contains personal information of members, to assist members to obtain research grants, realize employment opportunities, and record accomplishments.

During the initial phase of the database design, we met with our client to understand their requirements and needs. Based on our client’s requirements and needs specified during this phase, we confirmed that the database shall contain the following information:

- New members **shall** be assigned a unique numeric identifier.
- Members **shall not** provide their social security number.
- Members have a first name, middle initial and last name.
- Members have a birth date; however, that information is provided voluntary not mandatory.
- Members **shall** have a gender.
- Members **shall** have a title. [Mr, Mrs, Dr, Professor, etc]
- Members **shall** have a home address, and possibly a business address and shipping address.
- Members could have a home phone number, a business phone number, and a cell phone.
- Members have a college education. Some members may have multiple degrees.
- The degree and university name **shall** be stored.
- The year of graduation **shall** be stored.
- The GPA **shall not** be stored.
- Members publish papers in journals and write books. This information **shall** be stored to the database.
- Information stored about papers **shall** include the author, co-author, title, and journal.
- Information stored about books **shall** include the author, co-authors, Title, Publisher, and ISBN.

- Members have an area of research interest. Sometimes members have multiple areas of research interest.
- Research interest **shall** be stored as an academic term such as AI, Expert System, Complexity Theory, etc.
- Some members have obtained patents. The patent number and title **shall** be stored to the database.
- The year when the patent was awarded **shall** be stored to database.

After meeting with our client and discussing their requirements, we designed an E-R diagram as shown in Figure 1, to model the database. The E-R diagram shows *how* we decided to group the attributes to form the various tables. The

E-R diagram acted as a template that guided us during this design process [3]. This E-R diagram consists of strong entity sets with primary keys underlined. The relationships are depicted by the diamond shapes.

After our team and client agreed on the E-R diagram, the next step was to implement the database. We created two sets of SQL scripts. The first set of SQL scripts generates the database tables and the second set inserts information/data into the database tables. Both sets of SQL scripts are shown in Table 1 below. Figure 2 shows the database schema diagram, which was generated from MySQL Workbench after the database tables were created and data inserted into the tables.

SQL script to create tables	SQL scripts to populate the tables with the data
<pre>CREATE TABLE title (TITLE_ID varchar(9) NOT NULL, TITLE_NAME varchar(20) DEFAULT NULL, PRIMARY KEY (TITLE_ID));</pre>	<pre>INSERT INTO title (TITLE_ID, TITLE_NAME) VALUES ('100T', 'Dr'); INSERT INTO title (TITLE_ID, TITLE_NAME) VALUES ('101T', 'Miss'); INSERT INTO title (TITLE_ID, TITLE_NAME) VALUES ('102T', 'Mr'); INSERT INTO title (TITLE_ID, TITLE_NAME) VALUES ('103T', 'Mrs'); INSERT INTO title (TITLE_ID, TITLE_NAME) VALUES ('104T', 'Ms'); INSERT INTO title (TITLE_ID, TITLE_NAME) VALUES ('105T', 'Rev'); INSERT INTO title (TITLE_ID, TITLE_NAME) VALUES ('106T', 'Professor');</pre>
<pre>CREATE TABLE degree (DEGREE_ID varchar(9) NOT NULL, DEGREE_NAME varchar(20) DEFAULT NULL, PRIMARY KEY (DEGREE_ID));</pre>	<pre>INSERT INTO degree (DEGREE_ID, DEGREE_NAME) VALUES ('100D', 'BSCS'); INSERT INTO degree (DEGREE_ID, DEGREE_NAME) VALUES ('101D', 'BSEE'); INSERT INTO degree (DEGREE_ID, DEGREE_NAME) VALUES ('102D', 'BSCE'); INSERT INTO degree (DEGREE_ID, DEGREE_NAME) VALUES ('103D', 'MSCS'); INSERT INTO degree (DEGREE_ID, DEGREE_NAME) VALUES ('104D', 'MSEE');</pre>
<pre>CREATE TABLE address_type (ADDRESS_TYPE_ID varchar(9) NOT NULL, NAME varchar(30) DEFAULT NULL, PRIMARY KEY (ADDRESS_TYPE_ID));</pre>	<pre>INSERT INTO address_type (ADDRESS_TYPE_ID, NAME) VALUES ('1000AT', 'Home'); INSERT INTO address_type (ADDRESS_TYPE_ID, NAME) VALUES ('1001AT', 'Business'); INSERT INTO address_type (ADDRESS_TYPE_ID, NAME) VALUES ('1002AT', 'Shipping');</pre>
<pre>CREATE TABLE person (PERSON_ID varchar(9) NOT NULL, LAST_NAME varchar(20) NOT NULL, FIRST_NAME varchar(20) NOT NULL, MIDDLE_INITIAL varchar(20) DEFAULT NULL, TITLE_ID varchar(9) DEFAULT NULL, GENDER varchar(20) NOT NULL, DATE_OF_BIRTH date DEFAULT NULL, PRIMARY KEY (PERSON_ID), KEY PERSON_TITLE_FK1 (TITLE_ID));</pre>	<pre>INSERT INTO person VALUES ('1000P', 'Adams', 'Brooks', 'G', '100T', 'Male', '1966-11-22'); INSERT INTO person VALUES ('1001P', 'Blenn', 'Jones', 'J', '106T', 'Male', '1960-2-14'); INSERT INTO person VALUES ('1002P', 'Turner', 'Willians', 'M', '102T', 'Male', '1975-5-16'); INSERT INTO person VALUES ('1003P', 'Linda', 'Curry', 'B', '104T', 'Female', '1970-10-23'); INSERT INTO person VALUES ('1004P', 'John', 'Smith', '106T', 'Male', '1960-2-06'); INSERT INTO person VALUES ('1005P', 'Abbey', 'Green', 'H', '106T', 'Female', '1962-3-15'); INSERT INTO person VALUES ('1006P', 'Barbara', 'Richard', '100T', 'Female', '1971-4-20'); INSERT INTO person VALUES ('1007P', 'Beverley', 'Mark', 'N', '101T', 'Female', '1978-5-29'); INSERT INTO person VALUES ('1008P', 'George', 'Steven', 'K', '100T', 'Male', '1979-6-11'); INSERT INTO person VALUES ('1009P', 'Nadia', 'Jack', 'P', '100T', 'Female', '1969-1-20'); INSERT INTO person VALUES ('1010P', 'Nancy', 'Ronald', '104T', 'Female', '1982-11-14'); INSERT INTO person VALUES ('1011P', 'Kevin', 'Chuck', 'B', '102T', 'Male', '1985-12-10');</pre>
<pre>CREATE TABLE address (ADDRESS_ID varchar(9) NOT NULL, ADDRESS_TYPE_ID varchar(9) NOT NULL, PERSON_ID varchar(9) NOT NULL, STREET_ADDRESS varchar(100) NOT NULL, CITY varchar(20) DEFAULT NULL, STATE varchar(20) DEFAULT NULL, ZIP CODE varchar(10) DEFAULT</pre>	<pre>INSERT INTO address VALUES ('1000A', '1000AT', '1000P', '2207 Saw ln', 'Orlando', 'FL', '32828', 'USA'); INSERT INTO address VALUES ('1009A', '1000AT', '1003P', '600 Oak St.', 'Kissimme', 'FL', '34741', 'USA'); INSERT INTO address VALUES ('1010A', '1001AT', '1003P', '2699 Lee Rd.', 'Orlando', 'FL', '32789', 'USA'); INSERT INTO address VALUES ('1011A', '1002AT', '1003P', '2670 Lee Rd.', 'Orlando', 'FL', '32789', 'USA'); INSERT INTO address VALUES ('1012A', '1000AT', '1004P', '1600 Budinger Ave.', 'St. Cloud', 'FL', '34769');</pre>

<pre> NULL, COUNTRY varchar(20) DEFAULT NULL, PRIMARY KEY (ADDRESS_ID), KEY ADDRESS_PERSON_FK1 (PERSON_ID), KEY ADDRESS_ADDRESS_TYPE_FK2 (ADDRESS_TYPE_ID)); </pre>	<pre> INSERT INTO address VALUES ('1042A', '1000AT', '1017P', '101 Kent St.', 'Raleigh', 'NC', '27607', 'USA'); INSERT INTO address VALUES ('1043A', '1001AT', '1017P', '9099 Main St.', 'Raleigh', 'NC', '27601', 'USA'); INSERT INTO address VALUES ('1049A', '1001AT', '1019P', '1479 Gene St', 'Winter Park', 'FL', '32789', 'USA'); INSERT INTO address VALUES ('1050A', '1002AT', '1019P', '1480 Gene St', 'Winter Park', 'FL', '32789', 'USA'); </pre>
<pre> CREATE TABLE book (BOOK_ID varchar(9) NOT NULL, AUTHOR varchar(9) NOT NULL, CO_AUTHOR varchar(9) DEFAULT NULL, TITLE varchar(100) NOT NULL, PUBLISHER varchar(100) NOT NULL, ISBN varchar(20) NOT NULL, PUBLISH_DATE date NOT NULL, KEY BOOK_PERSON_FK1 (AUTHOR), KEY BOOK_PERSON_FK2 (CO_AUTHOR)); </pre>	<pre> INSERT INTO book VALUES ('1000BK', '1000P', '1001P', 'Artificial Intelegence', 'Mc Graw Hill', '0072283667', '2000-1-15'); INSERT INTO book VALUES ('1001BK', '1000P', null, 'Expert System', 'Addison Wesley', '0072283000', '2002-2-22'); INSERT INTO book VALUES ('1002BK', '1001P', '1005P', 'Database System', 'Mc Graw Hill', '0045890655', '2005-3-27'); INSERT INTO book VALUES ('1003BK', '1001P', null, 'Database Design Concept', 'Addison Wesley', '0055683689', '2007-9-04'); INSERT INTO book VALUES ('1004BK', '1004P', '1006P', 'Design Pattern', 'South-Western Educational Publishing', '0314234939', '2002-9-04'); INSERT INTO book VALUES ('1005BK', '1005P', null, 'Fundamentals Of C++', 'Jones and Bartlet', '0314234233', '2000-10-08'); INSERT INTO book VALUES ('1006BK', '1008P', '1009P', 'Computer Architecture', 'South-Western Educational Publishing', '0814234111', '2008-11-23'); </pre>
<pre> CREATE TABLE education (EDUCATION_ID varchar(9) NOT NULL, UNIVERSITY_NAME varchar(100) DEFAULT NULL, DEGREE_ID varchar(9) DEFAULT NULL, GRADUATION_DATE date DEFAULT NULL, PERSON_ID varchar(9) DEFAULT NULL, PRIMARY KEY (EDUCATION_ID), KEY EDUCATION_DEGREE_FK1 (DEGREE_ID), KEY EDUCATION_PERSON_FK1 (PERSON_ID)); </pre>	<pre> INSERT INTO education VALUES ('1000E', 'University of Central Florida', '100D', '1988-5-09', '1000P'); INSERT INTO education VALUES ('1001E', 'University of Central Florida', '103D', '1991-5-11', '1000P'); INSERT INTO education VALUES ('1002E', 'University of Central Florida', '106D', '1995-5-09', '1000P'); INSERT INTO education VALUES ('1003E', 'University of Central Florida', '100D', '1982-5-12', '1001P'); INSERT INTO education VALUES ('1004E', 'University of Central Florida', '103D', '1985-5-08', '1001P'); INSERT INTO education VALUES ('1005E', 'University of Central Florida', '106D', '1991-5-11', '1000P'); INSERT INTO education VALUES ('1006E', 'University of Florida', '101D', '1998-5-10', '1002P'); INSERT INTO education VALUES ('1007E', 'University of Texas', '101D', '1992-5-10', '1003P'); INSERT INTO education VALUES ('1008E', 'University of South Florida', '100D', '1982-5-12', '1004P'); INSERT INTO education VALUES ('1009E', 'University of South Florida', '103D', '1984-5-07', '1004P'); INSERT INTO education VALUES ('1010E', 'University of Central Florida', '107D', '1988-5-09', '1004P'); </pre>
<pre> CREATE TABLE research (RESEARCH_ID varchar(9) NOT NULL, TITLE varchar(100) NOT NULL, PERSON_ID varchar(9) DEFAULT NULL, PATENT_NUMBER varchar(20) DEFAULT NULL, AWARD_DATE date DEFAULT NULL, PRIMARY KEY (RESEARCH_ID), KEY RESEARCH_PERSON_FK1 (PERSON_ID)); </pre>	<pre> INSERT INTO research VALUES ('1000R', 'Biomedical Science', '1001P', null, null); INSERT INTO research VALUES ('1001R', 'Biomedical Science', '1000P', '100000PT', '2008-7-04'); INSERT INTO research VALUES ('1002R', 'Expert Systems', '1002P', null, null); INSERT INTO research VALUES ('1003R', 'Expert Systems', '1005P', '100002PT', '2000-5-10'); INSERT INTO research VALUES ('1005R', 'Formal Languages', '1007P', null, null); INSERT INTO research VALUES ('1007R', 'Formal Languages', '1008P', null, null); INSERT INTO research VALUES ('1008R', 'Formal Languages', '1009P', null, null); INSERT INTO research VALUES ('1009R', 'Formal Languages', '1010P', null, null); INSERT INTO research VALUES ('1013R', 'Fuzzy Logic', '1014P', null, null); </pre>
<pre> CREATE TABLE telephone (PERSON_ID varchar(9) NOT NULL, HOME_NUMBER varchar(20) DEFAULT NULL, BUSINESS_NUMBER varchar(20) DEFAULT NULL, CELL_NUMBER varchar(20) DEFAULT NULL, KEY TELEPHONE_PERSON_FK1 (PERSON_ID)); </pre>	<pre> INSERT INTO telephone VALUES ('1000P', '321-542-1234', '321-542-1000', '321-342-1034'); INSERT INTO telephone VALUES ('1001P', '407-123-1234', '407-242-1010', '407-842-1005'); INSERT INTO telephone VALUES ('1002P', '407-042-6758', '407-042-7766', '407-011-2038'); INSERT INTO telephone VALUES ('1003P', '321-888-1188', '321-777-1111', '407-666-0001'); INSERT INTO telephone VALUES ('1004P', '321-987-4321', '321-988-1212', '321-555-3401'); INSERT INTO telephone VALUES ('1005P', '321-365-9877', '321-563-2220', '321-566-3344'); </pre>

	<pre>INSERT INTO telephone VALUES ('1006P', '407-890-5421', '407-891-9999', '407-810-9977'); INSERT INTO telephone VALUES ('1007P', '407-909-7190', '407-990-0971', '407-342-1334');</pre>
<pre>CREATE TABLE journal (JOURNAL_ID varchar(9) NOT NULL, AUTHOR varchar(9) NOT NULL, CO_AUTHOR varchar(20) DEFAULT NULL, TITLE varchar(100) NOT NULL, JOURNALS varchar(1024) NOT NULL, PUBLISH_DATE date NOT NULL, PRIMARY KEY (JOURNAL_ID), KEY JOURNAL_PERSON_FK1 (AUTHOR), KEY JOURNAL_PERSON_FK2 (CO_AUTHOR));</pre>	<pre>INSERT INTO journal VALUES ('1000J', '1000P', '1001P', 'Biomedical Science', 'Involvement of p63 in the herpes simplex virus-1-induced demise of corneal cells', '2003-7-04'); INSERT INTO journal VALUES ('1001J', '1000P', NULL, 'Biomedical Science', ' Enhancement of tolerance development to morphine in rats prenatally exposed to morphine, methadone, and buprenorphine', '2008-7- 04'); INSERT INTO journal VALUES ('1002J', '1001P', NULL, 'Biomedical Science', 'Combinatorial gene therapy renders increased survival in cirrhotic rats', '2001-7-25'); INSERT INTO journal VALUES ('1003J', '1002P', NULL, 'Expert Systems', 'User profiling on the Web based on deep knowledge and sequential questioning', '2002-3-14'); INSERT INTO journal VALUES ('1004J', '1003P', '1002P', 'Expert Systems', 'On fault isolation by neural-networks-based parameter estimation techniques', '2006-6-06'); INSERT INTO journal VALUES ('1015J', '1015P', NULL, 'Fuzzy Logic', 'Intelligent and Fuzzy Systems ', '2008-9-11');</pre>
<pre>CREATE TABLE users (UserId varchar(45) NOT NULL, first_name varchar(45) NOT NULL, last_name varchar(45) NOT NULL, title varchar(45) NOT NULL, message varchar(1024) DEFAULT NULL, PRIMARY KEY (UserId));</pre>	<p>This Table is populated using the forum page of the SCS web site.</p>

Table 1. SQL script that generates the database tables and inserts information/ data

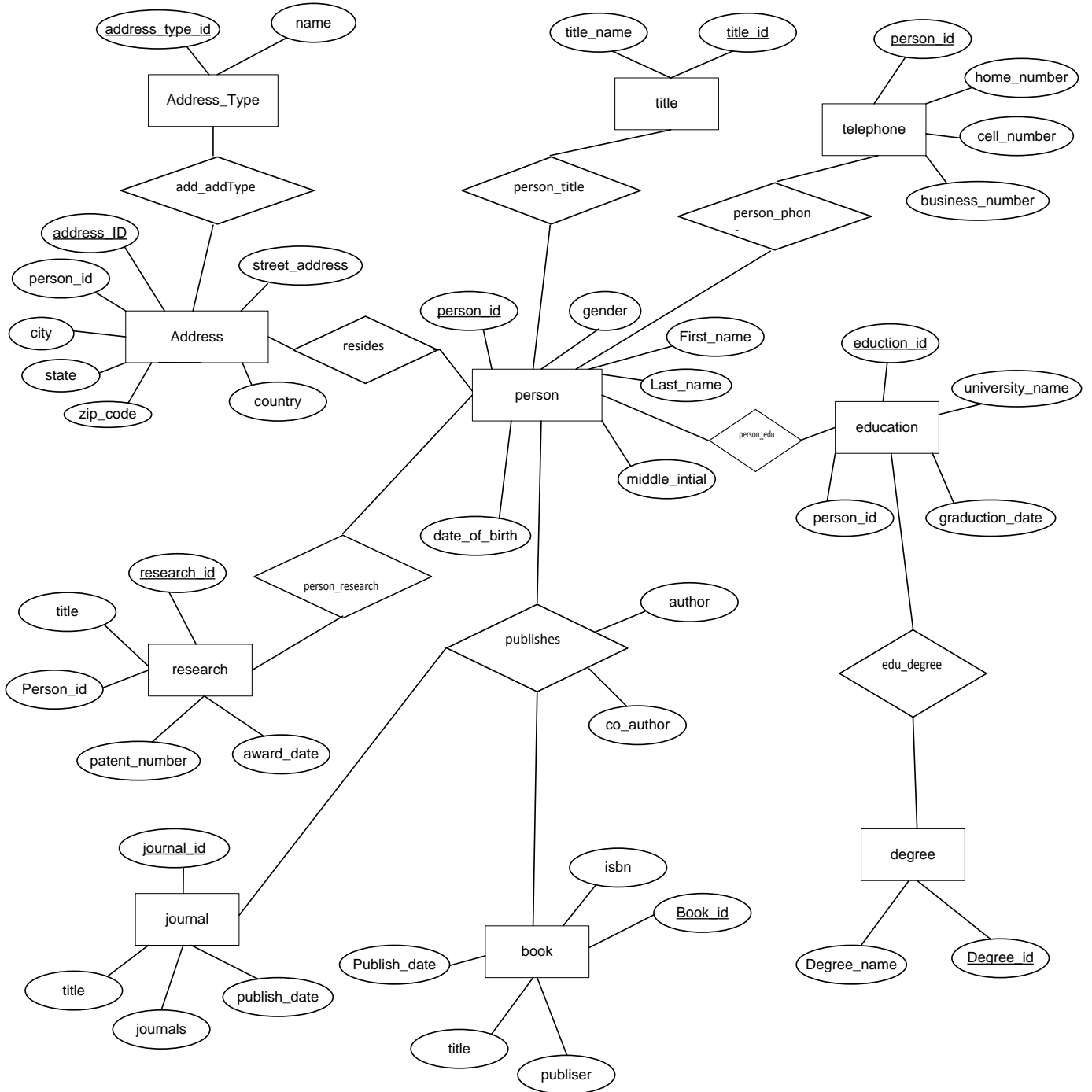


Figure 1. Entity Relationship Diagram

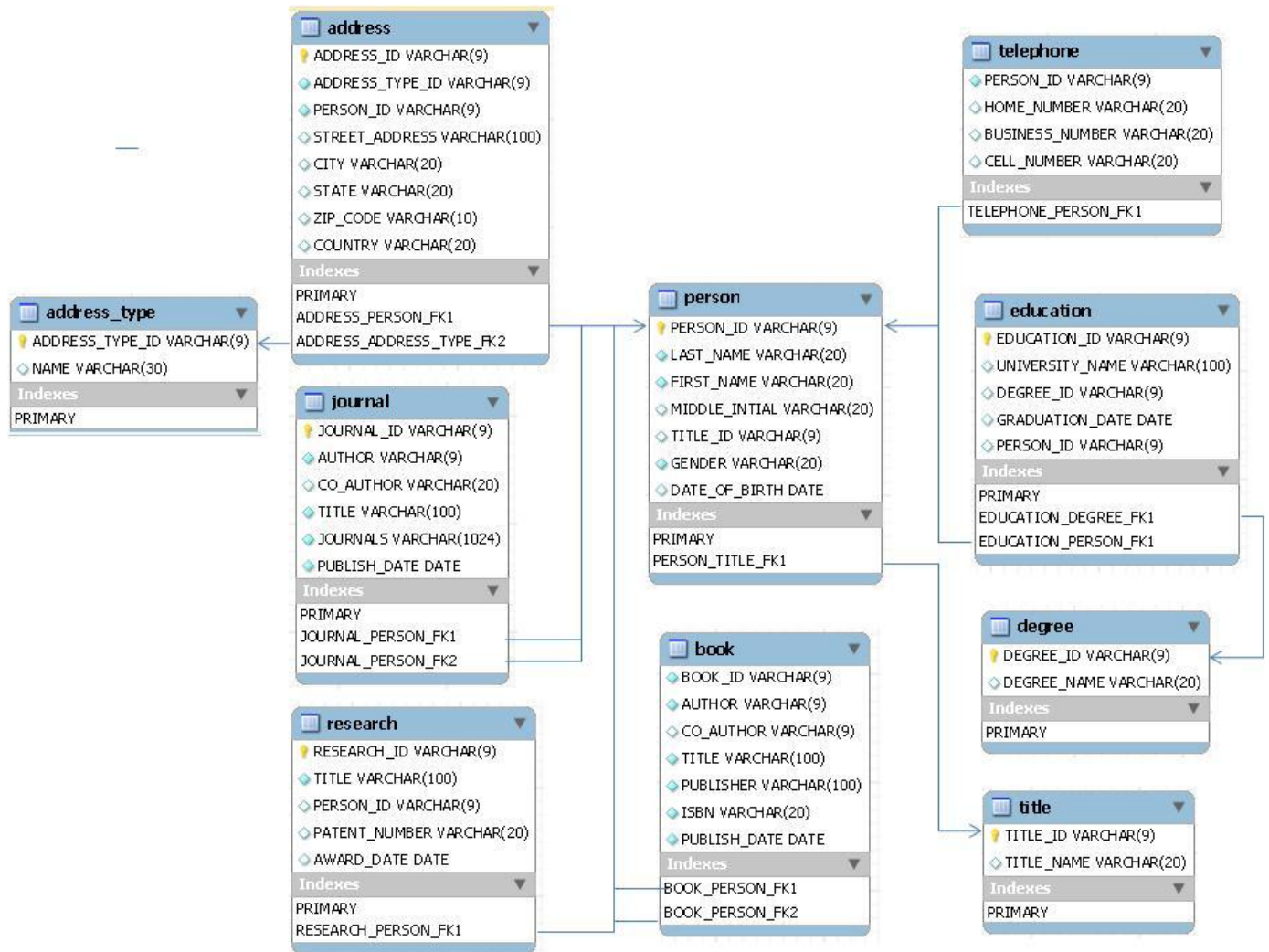


Figure 2. Database Project Schema Diagram

3 SQL QUERIES (*DATABASE QUERIES*)

This section of the paper focuses on the creation of SQL queries (database queries) used to retrieve information from the database created for our client, the Society of Computational Science. The objective was to write SQL statements for the following databases queries specified by our client.

- List all members by last name, first name, and member id. Sort the last names in alphabetical order.
- Provide a count of all members.
- List all members that have college degree in BSCS and MSCS.
- List all papers that have been published by members between the year 1990 and 2000.
- List all papers that have been published by members between the year 2001 and 2010.
- List all members that published more than 5 papers.
- List all members and their cell phone numbers. Sort the last names in alphabetical order.
- List all members that have a college degree in BSCS and live in the state of Florida.

- List all members that have a research interest in Formal Languages and Computational Complexity Theory.
- List all members that graduated with a degree in BSEE and have a patent.
- List all female members that are born before the year 1980.
- List all members that have a shipping address in the state of Alaska.
- List all members that have a research interest in Biomedical Science and have published a book (any book).
- List all members that graduated from the University of Central Florida or the University of South Florida.
- List all members by last name, first name, member id, and the university they graduated from. Include the state they are currently living in. The state shall be listed after the member id.

A listing of the SQL statements and the results obtained for each SQL statement are listed below in Figures 3-1 thru 3-15.

```

/* List all members by last name, first name, and
member id. Sort the last names in alphabetical
order.*/
SELECT Last_Name, First_name, person_id
FROM person
ORDER BY Last_Name;

```

Last_Name	First_name	person_id
Abbey	Green	1005P
Adams	Brooks	1000P
Barbara	Richard	1006P
Barry	White	1012P
Beverley	Mark	1007P
Blenn	Jones	1001P
Frank	Lawand	1015P
George	Steven	1008P
Jason	Mathew	1013P
John	Smith	1004P
Kevin	Chuck	1011P
Linda	Curry	1003P
Nadia	Jack	1009P
Nancy	Ronald	1010P
Naomi	Larry	1014P
Roger	Willie	1019P
Ryan	Justin	1018P
Scott	Dennis	1016P
Talia	Petter	1017P
Turner	Willians	1002P

Figure 3-1. SQL Statement & Results - 1

```

/* Provide a count of all members. */
SELECT COUNT(*) AS 'Number of members'
FROM person;

```

Number of members
20

Figure 3-2. SQL Statement & Results - 2

```
/* List all members that have college degree
in BSCS and MSCS. */
```

```
SELECT
  t.title_name 'Title',
  p.Last_Name 'Last Name',
  p.First_name 'First Name'
FROM Person p
LEFT JOIN education e
  ON e.person_id = p.person_id
LEFT JOIN degree d
  ON d.degree_id = e.degree_id
  AND d.degree_name = 'MSCS'
LEFT JOIN title t
  ON t.title_id = p.title_id
LEFT JOIN
  (SELECT
    p.person_id person_id,
    p.Last_Name Last_Name,
    p.First_name First_Name,
    d.degree_name degree_name
  FROM Person p
  LEFT JOIN education e
    ON e.person_id = p.person_id
  LEFT JOIN degree d
    ON d.degree_id = e.degree_id
    AND d.degree_name = 'BSCS'
  )qry
  ON qry.person_id = p.person_id
WHERE d.degree_name = 'MSCS' AND
  qry.degree_name = 'BSCS'
ORDER By p.Last_name;
```

Title	Last Name	First Name
Dr	Adams	Brooks
Professor	Blenn	Jones
Professor	Frank	Lawand
Professor	John	Smith
Professor	Roger	Willie

Figure 3-3. SQL Statement & Results - 3


```

/* List all papers that have been published by members
between the year 1990 and 2000. */
SELECT
Publish_Date Publish_date,
P.First_name Author,
p2.First_name Co_auther,
j.Title Jouranl_title,
j.Journals Journal
FROM journal j
LEFT JOIN person p
on p.person_id = j.Author
LEFT JOIN person p2
ON p2.person_id = j.CO_AUTHOR
WHERE Publish_Date between '1990-01-01' AND '1999-12-31'
ORDER BY publish_Date;

```

Publish_date	Author	Co_auther	Jouranl_title	Journal
3/23/1990	Dennis		Neural Networks	Network complexity and synchronous behavior ? an experimental approach
11/11/1991	Smith	Green	Formal Languages	Journal of Automata, Languages and Combinato
5/9/1992	Willie	Larry	Neural Networks	Data compression and regression through local principal curves and surfaces
12/12/1992	Green	Richard	Formal Languages	Modularity of Convergence and Strong Convergence in Infinitary Rewriting
1/16/1993	Mark	Chuck	Formal Languages	An upper bound on the number of states for a strongly universal hyperbolic cellular automaton on the pentagrid
4/4/1994	Steven		Computational Complexity Theory	Progress in computational complexity theory
1/8/1995	Smith		Computational Complexity Theory	Equivalence for Composed Communication Problems
6/16/1996	Chuck		Computational Complexity Theory	Physical portrayal of computational complexity
7/7/1998	Jack	Mathew	Fuzzy Logic	IEEE Transactions on Fuzzy Systems
6/12/1999	Lawand		Image Analysis	Respiratory Motion Compensation by Model-Based Catheter Tracking during EP Procedures
9/9/1999	Ronald		Computational Complexity Theory	Computational Complexity and Numerical Stability of Linear Problems
10/10/1999	Dennis		Fuzzy Logic	Bounded linear programs with trapezoidal fuzzy numbers

Figure 3-4. SQL Statement & Results - 4

```

/* List all papers that have been published by members between the year 2001 and 2010. */
SELECT
Publish_Date Publish_date,
P.First_name Author,
p2.First_name Co_auther,
j.Title Jouranl_title,
j.Journals Journal
FROM journal j
LEFT JOIN person p
ON p.person_id = j.Author
LEFT JOIN person p2
ON p2.person_id = j.CO_AUTHOR
WHERE Publish_Date BETWEEN '2001-01-01' AND '2010-12-31'
ORDER BY publish_Date;

```

Publish_date	Author	Co_auther	Jouranl_title	Journal
7/25/2001	Jones		Biomedical Science	Combinatorial gene therapy renders increased survival in cirrhotic rats
2/2/2002	Justin	Larry	Expert Systems	User profiling on the Web based on deep knowledge and sequential questioning
3/14/2002	Willians		Expert Systems	User profiling on the Web based on deep knowledge and sequential questioning
7/4/2003	Brooks	Jones	Biomedical Science	Involvement of p63 in the herpes simplex virus-1-induced demise of corneal cells
5/25/2005	Lawand		Image Analysis	Probabilistic framework for tracking in artifact-prone 3D echocardiograms
4/14/2006	Petter	Willie	Knowledge Representation	Integration of Genetic Fuzzy Systems and Artificial Neural Networks for Stock Price Forecasting
6/6/2006	Curry	Willians	Expert Systems	On fault isolation by neural-networks-based parameter estimation techniques
3/23/2007	Lawand	Larry	Neural Networks	Optical Memory and Neural Networks
6/23/2008	Dennis	Larry	Fuzzy Logic	Fuzzy Sets and Systems
7/4/2008	Brooks		Biomedical Science	morphine in rats prenatally exposed to morphine, methadone, and buprenorphine
7/4/2008	Lawand		Neural Networks	Microelectronics for Neural Networks and Fuzzy Systems
7/10/2008	Petter		Knowledge Representation	A Novel Image Retrieval Model Based on the Most Relevant Features
9/11/2008	Lawand		Fuzzy Logic	Intelligent and Fuzzy Systems
3/23/2009	Dennis	Willie	Fuzzy Logic	Fuzzy Optimization and Decision Making.
2/23/2010	Dennis	Larry	Neural Networks	Principal manifolds and graphs in practice: from molecular biology to dynamical systems

Figure 3-5. SQL Statement & Results - 5

```
/* List all members that published more than 5 papers. */
```

```
SELECT
  qry.Publish_Date 'Publish_date',
  qry.person_id 'Person Id',
  qry.Author Author,
  qry.Co_auther 'Co-author',
  qry.Jouranl_title 'Jouranl title',
  qry.Journal Journal
FROM
  (SELECT
    Publish_Date Publish_date,
    P.person_id Person_Id,
    P.First_name Author,
    p2.First_name Co_auther,
    j.Title Jouranl_title,
    j.Journals Journal,
    count(*) count
  FROM journal j
  LEFT JOIN person p
    ON p.person_id = j.Author
  LEFT JOIN person p2
    ON p2.person_id = j.CO_AUTHOR
  GROUP BY Author
  ORDER BY p.person_id) qry
WHERE qry.count > 5
SELECT person.person_id, Last_Name, First_name,
Cell_Number
FROM person, telephone
WHERE person.person_id = telephone.person_id
ORDER BY Last_Name;
```

Publish_date	Person Id	Author	Co-author	Jouranl title	Journal
8/8/1988	1015P	Lawand	Mathew	Image Analysis	A Dynamic Elastic Model for Segmentation and Tracking of the Heart in MR Image Sequences
6/23/2008	1016P	Dennis	Larry	Fuzzy Logic	Fuzzy Sets and Systems

Figure 3-6. SQL Statement & Results - 6

```
/* List all members and their cell phone numbers. Sort
the last names in alphabetical order.*/
```

```
SELECT person.person_id, Last_Name, First_name,
Cell_Number
FROM person, telephone
WHERE person.person_id = telephone.person_id
ORDER BY Last_Name;
```

person_id	Last_Name	First_name	Cell_Number
1005P	Abbey	Green	321-566-3344
1000P	Adams	Brooks	321-342-1034
1006P	Barbara	Richard	407-810-9977
1012P	Barry	White	325-200-8090
1007P	Beverley	Mark	407-342-1334
1001P	Blenn	Jones	407-842-1005
1015P	Frank	Lawand	407-505-5050
1008P	George	Steven	323-444-5311
1013P	Jason	Mathew	325-201-2010
1004P	John	Smith	321-555-3401
1011P	Kevin	Chuck	324-342-6633
1003P	Linda	Curry	407-666-0001
1009P	Nadia	Jack	323-000-4090
1010P	Nancy	Ronald	324-342-1666
1014P	Naomi	Larry	407-430-2990
1019P	Roger	Willie	320-999-9999
1018P	Ryan	Justin	320-333-3333
1016P	Scott	Dennis	407-500-5000
1017P	Talia	Petter	320-448-4480
1002P	Turner	Willians	407-011-2038

Figure 3-7. SQL Statement & Results - 7

```
/* List all members that have a college degree in BSCS and
live in the state of Florida.*/
```

```
SELECT p.Last_Name      'Last Name',
       p.First_name     'First Name',
       d.degree_Name    'Degree Name',
       aType.name       'Address Type',
       a.state          'State'
FROM person p
LEFT JOIN (education e, degree d, address a, address_type
aType)
ON (p.person_id = e.person_id AND
d.degree_id = e.degree_id AND
a.person_id = p.person_id AND
aType.address_type_id = a.address_type_id)
WHERE d.degree_Name = 'BSCS' AND a.state = 'FL' AND
aType.name = 'Home'
ORDER BY p.Last_name;
```

Last Name	First Name	Degree Name	Address Type	State
Adams	Brooks	BSCS	Home	FL
Beverley	Mark	BSCS	Home	FL
Blenn	Jones	BSCS	Home	FL
John	Smith	BSCS	Home	FL
Roger	Willie	BSCS	Home	FL

Figure 3-8. SQL Statement & Results - 8

```
/* List all members that have a research interest in Formal Languages and Computational Complexity Theory */
```

```
SELECT
  t.title_name 'Title',
  p.Last_Name 'Last Name',
  p.First_name 'First Name'
FROM Person p
LEFT JOIN research r
  ON r.person_id = p.person_id
  AND r.Title = 'Computational Complexity Theory'
LEFT JOIN title t
  ON t.title_id = p.title_id
LEFT JOIN
  (SELECT
    p.person_id person_id,
    p.Last_Name Last_Name,
    p.First_name First_Name,
    r.Title Title
  FROM Person p
  LEFT JOIN research r
    ON r.person_id = p.person_id
  AND r.Title = 'Formal Languages') qry
  ON
  qry.person_id = p.person_id
WHERE r.Title = 'Computational Complexity Theory' AND
  qry.Title = 'Formal Languages'
ORDER By p.Last_name;
```

Title	Last Name	First Name
Dr	George	Steven

Figure 3-9. SQL Statement & Results - 9

```
/* List all members that graduated with a degree in BSEE and have a patent. */
```

```
SELECT p.Last_Name 'Last Name',
  p.First_name 'First Name',
  d.degree_Name 'Degree Name',
  r.patent_number 'Patent Number'
FROM Person p
LEFT JOIN (education e, degree d, research r)
  ON (p.person_id = e.person_id AND
  d.degree_id = e.degree_id AND
  p.person_id = r.person_id)
WHERE d.degree_Name = 'BSEE' AND
  r.patent_number IS NOT NULL
ORDER BY p.Last_name;
```

Last Name	First Name	Degree Name	Patent Number
Abbey	Green	BSEE	100002PT
Ryan	Justin	BSEE	100006PT
Ryan	Justin	BSEE	100010PT
Scott	Dennis	BSEE	100005PT

Figure 3-10. SQL Statement & Results - 10

```
/* List all female members that are born before
the year 1980 */
```

```
SELECT p.Last_Name   'Last Name',
       p.First_name  'First Name',
       p.gender      'Gender',
       p.date_of_birth 'Date of Birth'
FROM person p
WHERE p.date_of_birth < '1980' AND
      p.gender = 'Female'
ORDER BY p.Last_name;
```

Last Name	First Name	Gender	Date of Birth
Abbey	Green	Female	3/15/1962
Barbara	Richard	Female	4/20/1971
Beverley	Mark	Female	5/29/1978
Linda	Curry	Female	10/23/1970
Nadia	Jack	Female	1/20/1969
Talia	Petter	Female	2/10/1967

Figure 3-11. SQL Statement & Results - 11

```
/* List all members that have a shipping address in the
state of Alaska */
```

```
SELECT p.Last_Name   'Last Name',
       p.First_name  'First Name',
       aType.name     'Address Type',
       a.Street_address 'Street address',
       a.city         'City',
       a.state        'State',
       a.zip_code     'Zip Code'
FROM person p
LEFT JOIN (address a, address_type aType)
ON (a.person_id = p.person_id AND
    aType.address_type_id = a.address_type_id)
WHERE a.state = 'AK' AND aType.name = 'Shipping'
ORDER BY p.Last_name;
```

Last Name	First Name	Address Type	Street address	City	State	Zip Code
Frank	Lawand	Shipping	2030 Fram St.	Petersburg	AK	99833
George	Steven	Shipping	2150 Mission Rd.	Anchorage	AK	32806
Nadia	Jack	Shipping	1211 W. Seventh Ave.	Anchorage	AK	99501
Scott	Dennis	Shipping	666 Whittier	Anchorage	AK	99501

Figure 3-12. SQL Statement & Results - 12

```
/* List all members that have a research interest in Biomedical Science and have
published a book (any book). */
```

```
SELECT p.Last_Name 'Last Name',
       p.First_name 'First Name',
       r.Title 'Research Title',
       b.Title 'Book Title',
       b.ISBN 'ISBN',
       b.publish_date 'Publish Date'
FROM Person p
LEFT JOIN (research r, book b)
ON (p.person_id = r.person_id AND
    b.author = p.person_id)
WHERE r.Title = 'Biomedical Science' AND
      b.ISBN IS NOT NULL
ORDER BY p.Last_name;
```

Last Name	First Name	Research T	Book Title	ISBN	Publish Date
Adams	Brooks	Biomedical	Expert System	72283000	2/22/2002
Adams	Brooks	Biomedical	Artificial Intelgence	72283667	1/15/2000
Blenn	Jones	Biomedical	Database System	45890655	3/27/2005
Blenn	Jones	Biomedical	Database Design Concept	55683689	9/4/2007
John	Smith	Biomedical	Design Pattern	314234939	9/4/2002

Figure 3-13. SQL Statement & Results - 13

<pre> /* List all members that graduated from the University of Central Florida or the University of South Florida. */ SELECT p.Last_Name 'Last Name', p.First_name 'First Name', d.degree_Name 'Degree Name', e.university_name 'University Name', e.graduation_date 'Graduation Date' FROM person p LEFT JOIN (education e, degree d) ON (p.person_id = e.person_id AND d.degree_id = e.degree_id) WHERE (e.university_name = 'University of Central Florida' OR e.university_name = 'University of South Florida') AND e.graduation_date IS NOT NULL ORDER BY p.Last_name, d.degree_name; </pre>	Last Name	First Name	Degree Name	University Name	Graduation Date
	Abbey	Green	BSEE	University of Central Florida	5/8/1986
	Abbey	Green	MSEE	University of South Florida	5/9/1988
	Abbey	Green	PHD_EE	University of South Florida	5/10/1992
	Adams	Brooks	BSCS	University of Central Florida	5/9/1988
	Adams	Brooks	MSCS	University of Central Florida	5/11/1991
	Adams	Brooks	PHD_CS	University of Central Florida	5/11/1991
	Adams	Brooks	PHD_CS	University of Central Florida	5/9/1995
	Barbara	Richard	BSCE	University of Central Florida	5/10/1993
	Barbara	Richard	MSCE	University of Central Florida	5/12/1996
	Barbara	Richard	PHD_EE	University of Central Florida	5/10/1999
	Barry	White	BSCS	University of Central Florida	5/9/1999
	Blenn	Jones	BSCS	University of Central Florida	5/12/1982
	Blenn	Jones	MSCS	University of Central Florida	5/8/1985
	Frank	Lawand	BSCS	University of Central Florida	5/8/1985
	Frank	Lawand	MSCS	University of Central Florida	5/9/1988
	George	Steven	BSCE	University of Central Florida	5/11/2001
	George	Steven	MSCE	University of Central Florida	5/12/2003
	George	Steven	PHD_EE	University of South Florida	5/10/2006
	Jason	Mathew	BSEE	University of Central Florida	5/8/1985
	John	Smith	BSCS	University of South Florida	5/12/1982
	John	Smith	MSCS	University of South Florida	5/7/1984
	John	Smith	PHD_EE	University of South Florida	5/7/1984
	John	Smith	PHD_EE	University of Central Florida	5/9/1988
	Nadia	Jack	BSCS	University of South Florida	5/11/1991
	Nadia	Jack	MSEE	University of South Florida	5/10/1994
	Nadia	Jack	PHD_CS	University of Central Florida	5/10/1998
	Roger	Willie	BSCS	University of South Florida	5/13/1981
	Roger	Willie	MSCS	University of South Florida	5/12/1983
	Ryan	Justin	BSEE	University of Central Florida	5/15/1977
	Ryan	Justin	MSEE	University of Central Florida	5/14/1980
	Scott	Dennis	BSEE	University of South Florida	5/7/1984
	Scott	Dennis	MSEE	University of South Florida	5/8/1987

Figure 3-14. SQL Statement & Results - 14


```

/* List all members by last name, first name, member id, and
the university they graduated from. Include the state they are
currently living in. The state shall be listed after the member id. */

```

```

SELECT DISTINCT p.Last_Name 'Last Name',
    p.First_name    'First Name',
    p.person_id     'Member Id',
    a.state         'State',
    e.university_name 'University Name'
FROM person p
LEFT JOIN (address a, education e)
ON (p.person_id = a.person_id AND
    p.person_id = e.person_id)
WHERE a.address_type_id = '1000AT'
ORDER BY p.Last_Name;

```

Last Name	First Name	Member Id	State	University Name
Abbey	Green	1005P	FL	University of South Florida
Abbey	Green	1005P	FL	University of Central Florida
Adams	Brooks	1000P	FL	University of Central Florida
Barbara	Richard	1006P	FL	University of Central Florida
Barry	White	1012P	TX	University of Central Florida
Beverley	Mark	1007P	FL	University of Texas
Blenn	Jones	1001P	FL	University of Central Florida
Frank	Lawand	1015P	AK	University of Central Florida
George	Steven	1008P	AK	University of South Florida
George	Steven	1008P	AK	University of Central Florida
Jason	Mathew	1013P	TX	University of Central Florida
John	Smith	1004P	FL	University of Central Florida
John	Smith	1004P	FL	University of South Florida
Linda	Curry	1003P	FL	University of Texas
Nadia	Jack	1009P	AK	University of Central Florida
Nadia	Jack	1009P	AK	University of South Florida
Naomi	Larry	1014P	FL	University of Louisiana
Roger	Willie	1019P	FL	University of South Florida
Ryan	Justin	1018P	FL	University of Central Florida
Scott	Dennis	1016P	AK	University of South Florida
Talia	Petter	1017P	NC	University of Louisiana
Turner	Willians	1002P	FL	University of Florida

Figure 3-15. SQL Statement & Results - 15

4 GRAPHICAL USER INTERFACE (FORUMS)

Our client gave us an extra task to create an online forum for their website. The purpose of the forum is to let users submit forum messages, which will be posted on the webpage. Our team decided to use a Graphical User Interface (GUI) to obtain information from the user including their name, member id, and forum message. The GUI (Forums) design was a multi-phase process consisting of the following:

- Install Apache Server, PHP and MySQL
- Design of the GUI
- Create the Html code
- Create and link Cascading Style Sheet (CSS) which defines the styles of the GUI
- Create a database to store user information
- Write and incorporate PHP code into the Html page
- Testing of the Forums page

Our team used a two-part approach to the design layout of the Graphical User Interface (Forums). The first part of the layout involves the user entering and submitting data, which can be seen in Figure 5 where the left side of the Graphical User Interface layout is for user input. The second part of the layout approach is for the output and is depicted on the right hand side of the screen as shown in Figure 6. This two-part approach design layout results in a simple, user friendly GUI. Forum messages are generated on the same page which keeps the process simple.

The GUI works as follows: If the user wants to post a forum message, they simply go the Forums page and fill in the required information as follows:

- Members **shall** enter information (general text messages) in a text box and store the information to database.
- Members **shall** provide their member id, last name, and first name in order to store the information.

Member ID	First Name	Last Name	Title	Message
-----------	------------	-----------	-------	---------

© All rights reserved. Society of Computational Science. 2010.
Web Hosting by HostGator

Figure 5. Graphical User Interface – Screen capture before user input is submitted

After they fill in this required information, they click the “Submit” button, which will store the information to the database via a PHP “POST” methods. After the user input is submitted, it is stored in a SQL database called “users.” Figure 4 shows the SQL statement used to create the database table “users.” The messages stored to the database are displayed in a separate text window on the right side of the screen. The first name and the member id are displayed with each message. Figures 5 and 6 show the screen captures for the GUI.

```
CREATE TABLE users (
  UserId varchar(45) NOT NULL,
  first_name varchar(45) NOT NULL,
  last_name varchar(45) NOT NULL,
  title varchar(45) NOT NULL,
  message varchar(1024) DEFAULT NULL,
  PRIMARY KEY (UserId)
);
```

Figure 4. SQL Statement to Create Table “users”

4.1 Forum Design Implementation

The Forum Design was implemented using html (localhost) web page, Apache Server, PHP, and MySQL. The Html and PHP code are included in the accompanying project file CD. Figure 5 shows the Forum design with Html and Figure 6 shows the final page after the user submits information and the PHP is processed. The Html webpage is linked to a Cascading Styles Sheet (CSS), which contains the design styles such as colors, fonts, and size specifications. The PHP code uses a “POST” message to receive the information when it is submitted. The PHP code in conjunction with the Apache Server stores the information in the MySQL table called “users.” The Forums page was tested thoroughly to ensure that the PHP code was working by getting user input, storing it in the database, and displaying it on the screen.

Member ID	First Name	Last Name	Title	Message
1111	Barry	Gobremaran	Test	Forum Message Testing
2222	Angela	Rubundo	Forum Test	This is just a test

© All rights reserved. Society of Computational Science. 2010.
Web Hosting by HostGator

Figure 6. Graphical User Interface – Screen capture after user input is submitted

5 RESULTS

The results obtained from the SQL statements as shown in Table 1 confirm that the E-R diagram created during the initial database design phase was in fact a good model of the database. The results show that the database is meeting our client's expectations.

The results of the Forums process suggest that the output of the Forums page is also meeting our client's expectations. Additional testing was performed during the Forums implementation process to ensure that the PHP was working. There were no major errors encountered.

6 CONCLUSION

The focus of this paper is on the importance of database design. Our team demonstrated this importance by designing and implementing a database for one of our clients, Society of Computational Science. Designing a database involves several steps, most of which are done prior to implementation. The first step in designing a database is to fully understand the customer requirements and needs. The Entity-Relationship (E-R) diagram is used to model the database to show *how* to group the attributes to form the various tables. Once the database is implemented, SQL statements can be used to store, retrieve, and modify data in the database. Our team created several SQL statements for database queries specified by our client. Finally, we took a look at using SQL in conjunction with PHP and Html to retrieve and store data to a database through a web page forum. There were many things to consider during the Forums design and implementation, with testing being the most crucial to ensure that data is stored and posted accurately via the PHP code. Whether designing a database or a webpage, planning is a key step that must be done correctly to ensure that all customer requirements are met and risks minimized.

REFERENCES

- [1] *The American Heritage Dictionary*. 4th ed. New York: New York: Dell Publishing, p 221, 2001.
- [2] A. Silberschatz, H. Korth, S. Sudarshan, *Database System Concepts*. 6th ed. New York: New York: McGraw-Hill Companies, Inc., pp. 3, 16. 2011.
- [3] "Database Design, Implementation, and Normalization," Society of Computational Science, 2010. [Online] Available: Society of Computational Science, <http://www.societycomputationalscience.org>. [Accessed: July 22, 2010]