# RECOGNIZING RECORDS FROM THE EXTRACTED CELLS OF GENEALOGICAL MICROFILM TABLES

by

Kenneth M. Tubbs

A thesis submitted to the faculty of

Brigham Young University

in partial fulfillment of the requirements for the degree of

Master of Science

Department of Computer Science

Brigham Young University

December 2001

BRIGHAM YOUNG UNIVERSITY

GRADUATE COMMITTEE APPROVAL

of a thesis submitted by

Kenneth Tubbs

This thesis has been read by each member of the following graduate committee and by majority vote has been found to be satisfactory.

| | |
|---|---|
| _____ | _____ |
| Date | David W. Embley, Chair |
| _____ | _____ |
| Date | William A. Barrett |
| _____ | _____ |
| Date | Kent E. Seamons |

BRIGHAM YOUNG UNIVERSITY

As chair of the candidate's graduate committee, I have read the thesis of Kenneth M. Tubbs in its final form and have found that (1) its format, citations, and bibliographical style are consistent and acceptable and fulfill university and department style requirements; (2) its illustrative materials including figures, tables, and charts are in place; and (3) the final manuscript is satisfactory to the graduate committee and is ready for submission to the university library.

_____ _____

Date David W. Embley
Chair, Graduate Committee

Accepted for the Department _____

David W. Embley
Graduate Coordinator

Accepted for the College _____

G. Rex Bryce
Associate Dean, College of Physical and Mathematical Sciences

# ABSTRACT

RECOGNIZING RECORDS FROM THE EXTRACTED CELLS OF GENEALOGICAL MICROFILM TABLES

Kenneth M. Tubbs

Department of Computer Science

Master of Science

This thesis describes an algorithmic process to automatically identify and extract records found in genealogical, microfilm tables. Our table-processing algorithm accepts input as an XML file describing the individual cells of a genealogical table taken from microfilm, and it produces SQL statements to insert the coordinates of a table's value cells into a database. Two key features drive the algorithm: (1) geometric layout and (2) label matching with respect to a given genealogical ontology. Our algorithm succeeds when it organizes the tables cells described in the input XML file into appropriate records and creates SQL statements to insert them into a database. Our algorithm operates in three steps. (1) It extracts features from the cells described in an XML input file. (2) The algorithm applies correlation rules that relate and update the collected evidence. (3) The algorithm produces records for human users to verify and SQL insert statements that

enter the coordinates of table cells into a relational database. The algorithm achieved a precision of 93.20 percent, a recall of 92.41 percent, and an accuracy of 92.15 percent on the database fields it populated on our test corpus of microfilm tables.

**CONTENTS**

CHAPTER 1 – INTRODUCTION AND RELATED WORK . . . . . 1

CHAPTER 2 – INPUT RESOURCES AND OUTPUT . . . . . 8
2.1 XML Input File . . . . . 8
2.2 Ontology Input File . . . . . 9
2.3 Output File . . . . . 15

CHAPTER 3 – METHODOLOGY . . . . . 18
3.1 Feature Extraction . . . . . 18
3.2 Update Rules . . . . . 27
3.3 Verification and Record Storage . . . . . 33

CHAPTER 4 – EXPERIMENTAL RESULTS AND ANALYSIS . . . . . 36
4.1 Measurements . . . . . 36
4.2 Training Set . . . . . 38
4.3 Test Set . . . . . 41
4.4 Discussion . . . . . 42

CHAPTER 5 – CONCLUSION AND FUTURE WORK . . . . . 53
5.1 Conclusion . . . . . 53
5.2 Future Work . . . . . 54

REFERENCES . . . . . 55

APPENDIX 1 – INPUT XML FILE . . . . . 59

APPENDIX 2 – SQL STATEMENTS . . . . . 70

APPENDIX 3 – SCREEN SHOTS . . . . . 79

APPENDIX 4 – INDEX OF MICROFILM ROLLS . . . . . 81

# CHAPTER 1 – INTRODUCTION AND RELATED WORK

Millions of people want genealogical information [Fam01]. Yet this wealth of information is nearly inaccessible because extracting and organizing it by hand is slow, error-prone and tedious. Presently, it is available only by monotonous, individual explorations. In addition, individuals who do not live near microfilm libraries must order microfilm in large rolls. This costs them a significant amount of both time and money. Imagine the ability to search thousands of microfilm documents quickly and from any computer. Envision further the ability of algorithms to help extract automatically the structure and content of these microfilm documents. These capabilities would greatly accelerate the progress of genealogical research. This thesis addresses one of the underlying obstacles to achieving this vision.

One type of genealogical microfilm document is a table with machine-printed labels and hand-written values. See Figure 1.0 for an example. This type of table appears as a rectangular grid of cells. A *cell* is a rectangle within the grid. A series of horizontally connected cells is a *row*, and a series of vertically connected cells is a *column*. Tables with machine-printed labels and hand-written values contain three different types of cells. The first type of cell contains machine-printed text. These cells are *label cells*. They are often at the head of a row or column and declare the type of information contained in the cells of the row or column. The second type of cell contains hand-written text. These cells are *value cells*. They contain hand-written data described by a label cell. The third type of cell contains no machine-printed or hand-written text.

These cells are *empty cells*. They provide layout information and complete the rectangular shape of a table.

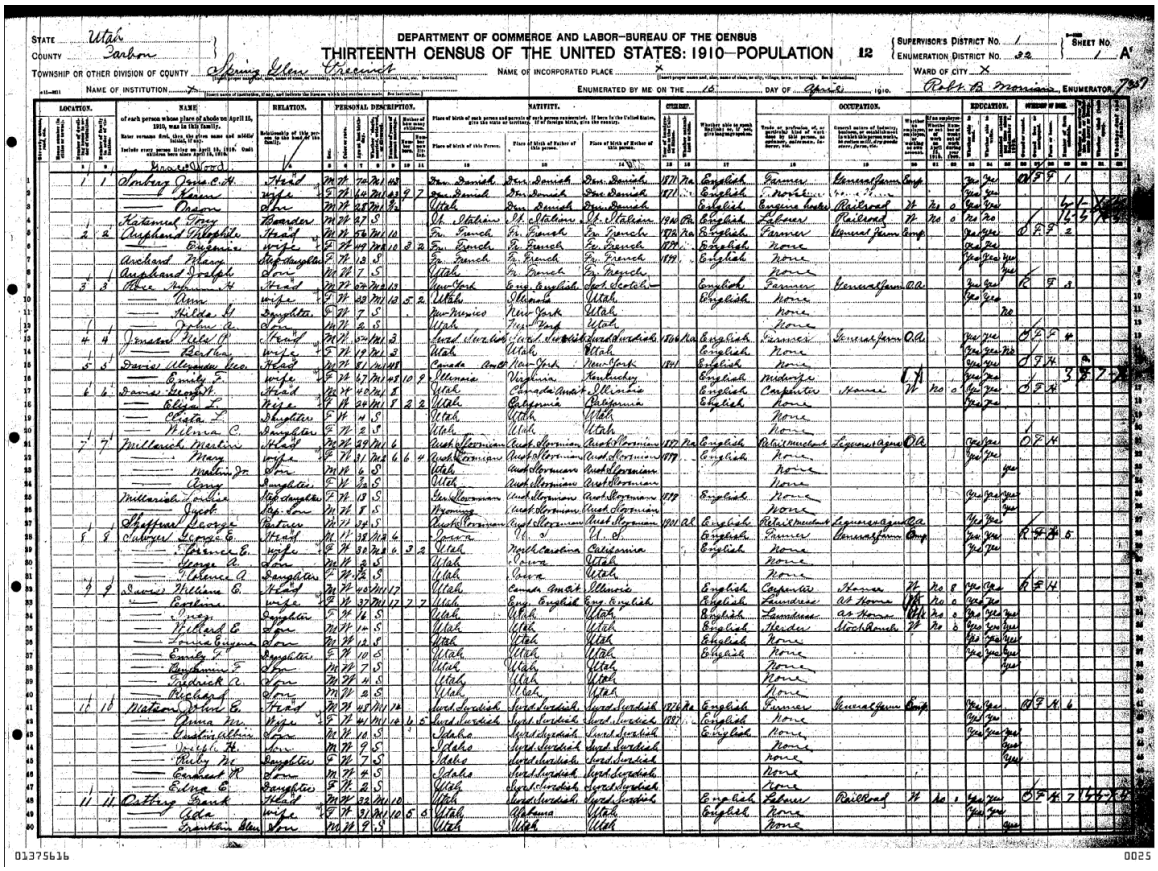



**Figure 1.0**

This thesis takes advantage of research currently underway to identify and zone the cells that make up tables found in microfilm [CK97] [NB01]. For cells that contain old, typeset text, other research efforts are underway to translate the machine-printed text in the image to a character string using optical character recognition [LLN+96] [Rim01] [WMR97]. These research efforts generate three attributes about each of the cells in a table: (1) the coordinates of each cell, (2) whether each cell is empty, and (3) the

character string for each cell that contains machine-printed text. The hand-written text within table cells is not available because it cannot yet be automatically obtained.

Other research explores the ability to transmit pieces of microfilm images across the Internet at various resolutions [BGS+98] [KB01]. This work will allow information retrieval systems to answer queries about a microfilm table by transmitting only the relevant parts of the original microfilm image.

Current research efforts in table understanding are advancing on two fronts. (1) Research efforts concentrate on developing pattern recognition algorithms to identify table structure from lines and white space. (2) Other research efforts match table text values to the fields of a database [LCC99]. While the first research efforts can be used for both tables found in images [BEB01] [KD01] and machine-printed text tables [LCC99], the second research efforts can only be applied to tables with machine-printed values [LCC99]. Neither of these research efforts directly applies to microfilm tables with machine-printed labels and hand-written values for three reasons. (1) This problem represents table structure as the coordinates of extracted cells (not lines or white space). (2) This problem does not provide the text of the hand-written values. (3) Matching table values to database fields is not sufficient to identify all the records contained in a genealogical table.

Present table recognition research focuses on modeling the geometry of tabular documents. The current techniques for identifying tabular structure include template

matching [HD97] [KS99], regular expression creation [LCC99], statistical modeling [BBI98] [BEB01] and grammar matching [GK95] [AAM+01]. These techniques have been applied to both printed text tables [LCC99] and images containing tables [Has98] [KD01]. Other research efforts match the printed-text of cell values to database fields using regular expressions and user created templates [LCC99]. Yet, they do not consider the relationships or constraints between the data values. In addition, these approaches cannot be applied when the hand-written values are unknown. While all these research efforts exploit the observable properties of tables, they ignore the constraints and relationships between the data presented in the tables. The field of table processing needs research to explore the combination of geometric and ontological constraints to identify table record structure

In this thesis, we exploit the relationships and constraints between data objects to identify records from genealogical tables. Our *ontology* is a model that describes the relationships and constraints between different types (sets) of data objects [Emb98]. Thus, a genealogical ontology describes the relationships between objects such as "Person" and "Full Name". We can map the label cells (since the handwritten values are unknown) in a table to the object sets in an ontology to determine the relationships between values cells described by the label cells. A *record* is an appropriate collection of label-value pairs with respect to the object sets in an ontology. Figure 1.1 shows the top left corner of the table in Figure 1.0. Suppose that the label cell "NAME" maps to an object set "Full Name" that relates to the object set "Person" in a genealogical ontology.

Then, the label “NAME” describes the “Full Name” value cells that are in the “Person” record.

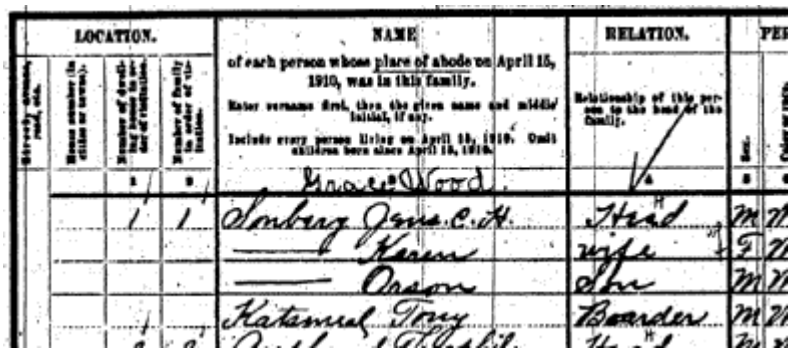



**Figure 1.1**

The automatic recognition of table records is difficult for five reasons. (1) Tables have many different layouts and styles. Lopresti and Nagy explain that although many tables share common geometric characteristics, they partition information differently [LN00]. Even tables representing the same information can arrange the cells in various layouts of columns and rows. (2) Tables can divide a single label into a multi-level hierarchy of label cells [HKL+01]. This makes it difficult to discover the correct associations between label cells and the fields of a database. (3) Tables factor data items in different ways. For example, in Figure 1.1**,** the column headed by “LOCATION” has only one value cell, and the column headed by “NAME” has four value cells. The “LOCATION” label cell *factors* the “NAME” label cell because the value cell under “LOCATION” should be included in each record containing one of value cells under “NAME”. (4) Microfilm tables often lack information or are ambiguous. These situations force automated data extraction algorithms to make decisions based on what is

most likely. (5) A record extraction algorithm must match the label cells in a table to the standardized fields in a database.

In this thesis, we describe an algorithmic process to automatically identify records from the extracted cells of genealogical, microfilm tables using both geometric and ontological constraints. Our table-processing algorithm accepts input as an XML file describing a genealogical table taken from microfilm, and it produces SQL statements to insert the coordinates of a table's value cells into a database. Our algorithm succeeds when it organizes the tables cells described in the input XML file into appropriate records and creates SQL statements to insert them into a database.

Our algorithm operates in three steps:

1. It collects features to answer seven questions about the table cells described in an XML file. These seven questions describe seven relationships that exist between a table's cells. The extracted features impose geometric and ontological constraints on these relationships. The seven questions are as follows:
    a. Which label cells describe which value cells?
    b. Which value cells are in the same row?
    c. Which value cells are in the same column?
    d. Which value cells are geometrically similar?
    e. Which label cells form multi-level, hierarchical labels?
    f. Which label cells map to which fields in a database?
    g. Which label cells describe value cells that factor multiple records in the table?

2. The algorithm applies correlation rules that support the evidence collected to answer one question with the evidence collected to support another question. The correlation rules are heuristics that generalize the structure and layout of table cells and enforce the constraints of the genealogical ontology.
3. The algorithm produces records for human users to verify and SQL insert statements that enter the coordinates of value cells into a relational database.

The main contribution of this research is to develop an algorithmic process that uses both geometric and ontological constraints to identify the records contained within genealogical tables with hand-written values. We explore this contribution in the following chapters. In Chapter 2, we describe the format of the XML input file, the genealogical ontology and the output SQL statements. In Chapter 3, we present the details of our algorithm for identifying table records. We list and analyze the features extracted to support each of the seven relationships between table cells, and we discuss the correlation rules used to corroborate the evidence collected about the seven relationships. In Chapter 4, we describe our algorithm's results on training and test sets. In Chapter 5, we conclude and offer several areas for future research.

# CHAPTER 2 – INPUT RESOURCES AND OUTPUT

## 2.1 XML Input File

Our table-processing algorithm accepts input as an XML file that describes a genealogical table taken from a microfilm roll. Research is currently underway to identify and zone the cells that make up printed tables found in microfilm [NB01] and to translate machine-printed text into a character string for each cell with printed-text using optical character recognition [LLN+96] [Rim01]. These raw cells are stored in an XML document that represents the table on the microfilm document [Xml]. The XML file describes attributes about each of the cells in a table. The attributes include the coordinates of each cell, whether or not each cell is empty, and the character string for each cell that contains printed text. Figure 2.1.1 shows part of an input XML file for the image in Figure 1.0. A complete input file for the image in Figure 1.0 is in Appendix 1.

```
<table source=" 0823344_1.gif" ontology="genealogy.xml">

        <cell rect="25,114,132,136" printed_text="LOCATION" empty="0"/>
        <cell rect="134,115,326,192" printed_text="NAME of each person whose place
            of abode on April 15, 1910 was in this family. Enter surname first, then the given
            name and middle initial, if any.  Include every person living on April 15, 1910. Omit
            children born since April 15, 1910." empty="0"/>
        <cell rect="415,133,437,194" printed_text="Sex." empty="0"/>

          • • •


          <cell rect="117,68,135,80" printed_text="" empty="0" />
          <cell rect="117,80,135,92" printed_text="" empty="1" />
          <cell rect="152,80,255,92" printed_text="" empty="1" />
          <cell rect="152,92,255,104" printed_text="" empty="1" />
          <cell rect="'335,80,350,92'" printed_text="" empty="1" />
          < cell rect="350,92,369,104" printed_text="" empty="1" />

          • • •


</table>
```

**Figure 2.1.1**

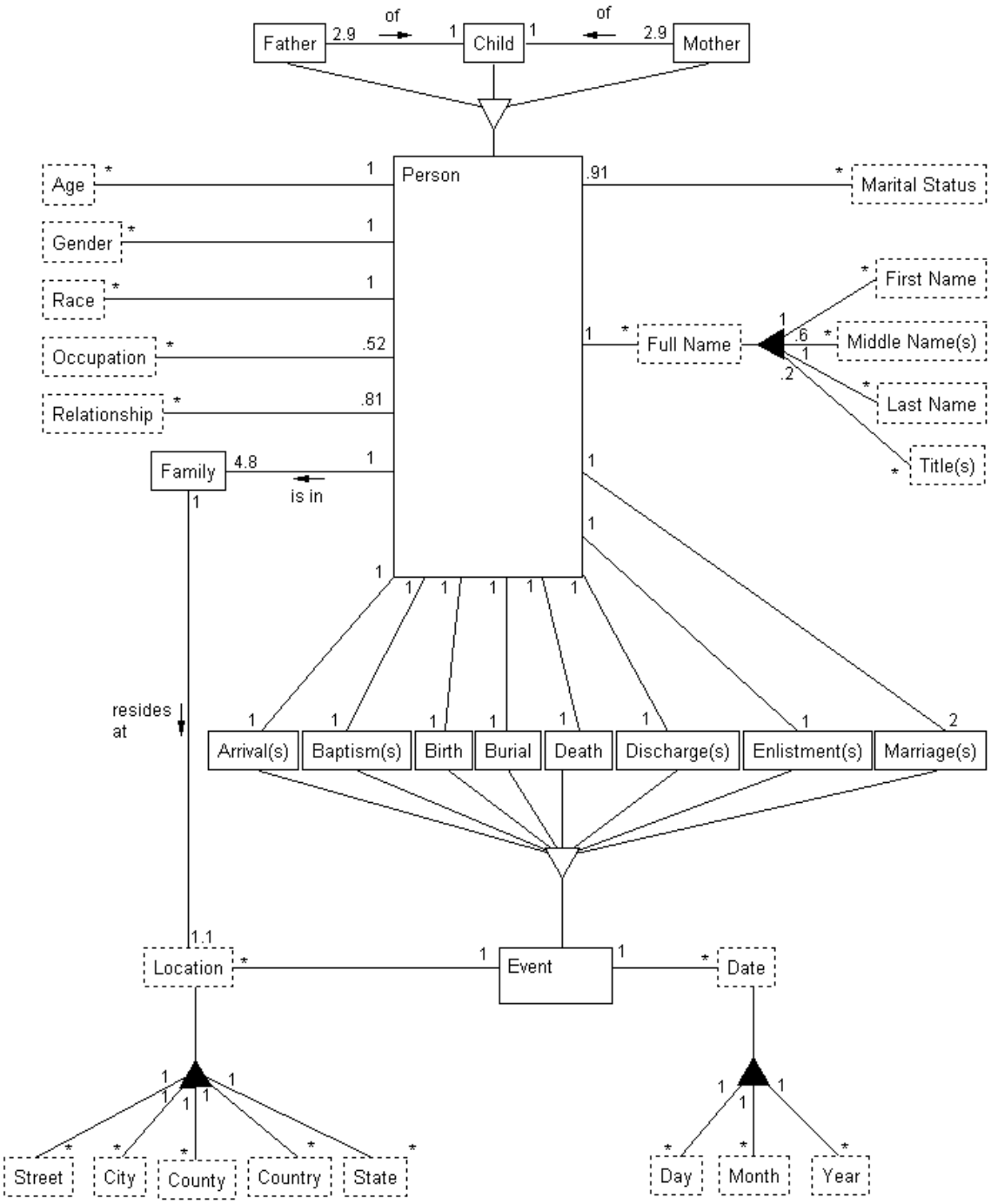
The “table” element in the XML file bounds a table’s cells. The “source” attribute gives the name of the original microfilm image. The “ontology” attribute gives the name of the ontology used to identify records. Each of the “cell” elements in Figure 2.1.1 is a cell in the table. The attribute “rect” provides the coordinates for every cell’s top left corner and lower right corner in an x-y coordinate system with the origin at the top left. The first number is the left x coordinate, the second number is the top y coordinate, the third number is the right x coordinate, and the last number is the bottom y coordinate. These coordinates define the rectangular boundary of the cell. The midpoint of a cell is the coordinate exactly in the horizontal and vertical middle of the cell. The attribute “printed_text” gives the printed text in each table cell, if there exists any. The attribute “empty” is set to zero if the table cell is empty and set to one if the table cell has machine-printed text or hand-written text.

## 2.2 Ontology Input File

A genealogical ontology describes the relationships between genealogical objects. For the purpose of record extraction from tables, we created a genealogical ontology that represents the expected relationships for genealogical objects in tables. See Figure 2.2.1. The boxes in the figure represent genealogical objects, such as “Age” or “Person”. The boxes outlined in dotted line are lexical object sets (objects in these object sets are character strings), while the boxes are non-lexical object sets (objects in these sets are object identifiers). Thus, in Figure 2.2.1, “Age” is a lexical object set, while “Person” is a non-lexical object set. Relationships exist between and connect object sets. Aggregations (shown with black triangles) and specializations (shown with white

triangles) are two special relationships. The participation constraints (the numbers for each relationship) denote the expected number of times a relationship exists between objects. They represent only the number of expected relationships for a particular snapshot from a person's life (that would be recorded in a table) and not the expected number of relationships over a person's lifetime. Thus, if "Occupation" appears in a microfilm table, we expect it to appear on average .52 times in each "Person" record. We calculated the participation constraints by counting the number of objects for each object set in the training set documents. The symbol "*" indicates that the relationship is unbounded.

**Figure 2.2.1**


Father
2.9
of
1
Child
1
of
2.9
Mother
Person
Age
Gender
Race
Occupation
Relationship
Family
4.8
is in
.52
.81
.91
Marital Status
First Name
Middle Name(s)
Last Name
Title(s)
Full Name
.6
.2
resides at
Arrival(s)
Baptism(s)
Birth
Burial
Death
Discharge(s)
Enlistment(s)
Marriage(s)
Location
Event
Date
Street
City
County
Country
State
Day
Month
Year

We converted the genealogical ontology in Figure 2.2.1 into the XML representation in Figure 2.2.2. The extraction algorithm accepts the genealogical ontology in this XML form. The element “Ontology” bounds the objects and relationships in the genealogical ontology. Each of the “ObjectSet” elements represents an object set in the ontology. The attribute “id” is a unique id for each object set. The attribute “Name”, for the element “ObjectSet”, is the name of the object set. The attribute “syn” contains a list of synonyms for the name of the object set. The element “Relationship” represents each relationship (line) in the ontology. The attribute “set1” is the name of the first object set in the relationship; the attribute “set 2” is the second object set in the relationship. The attribute “id1” is the unique identifier for the first object set; the attribute “id2” is the unique identifier for the second object set. The attribute “participation1” is the participation constraint nearest the first object set, and the attribute “participation2” is the participation constraint nearest the second object set in the relationship. The element “Specialization” represents a specialization relationship in the ontology. The element “Aggregation” represents an aggregation relationship in the ontology.

**Figure 2.2.2 – Part 1 of 2**

```
<Ontology>

    <ObjectSet id="0" name="Person" syn="" lex="0"/>
    <ObjectSet id="1" name="Family" syn="families" lex="0"/>
    <ObjectSet id="2" name="Event" syn="" lex="0"/>
    <ObjectSet id="3" name="Age" syn="age birthday" lex="1"/>
    <ObjectSet id="4" name="Gender" syn="gender sex male female males females" lex="1"/>
    <ObjectSet id="5" name="Relationship" syn="relationship relation" lex="1"/>
    <ObjectSet id="6" name="Occupation" syn="occupation job profession work career livelihood employment
        rank" lex="1"/>
    <ObjectSet id="7" name="Race" syn="race ethnic group tribe line linage nationality color black white" lex="1"/>
    <ObjectSet id="8" name="Full Name" syn="full name whom who" lex="1"/>
    <ObjectSet id="9" name="First Name" syn="first given christian" lex="1"/>
    <ObjectSet id="10" name="Middle Name(s)" syn="middle initial" lex="1"/>
    <ObjectSet id="11" name="Last Name" syn="last surname" lex="1"/>
    <ObjectSet id="12" name="Title(s)" syn="title" lex="1"/>
    <ObjectSet id="13" name="Arrival(s)" syn="arrival immigrated immigration" lex="0"/>
    <ObjectSet id="14" name="Baptism(s)" syn="baptism christening" lex="0"/>
    <ObjectSet id="15" name="Birth" syn="birth born nativity" lex="0"/>
    <ObjectSet id="16" name="Burial" syn="burial buried funeral interment" lex="0"/>
    <ObjectSet id="17" name="Marriage(s)" syn="marriage wedding matrimony" lex="0"/>
    <ObjectSet id="18" name="Discharge(s)" syn="discharge release veteran" lex="0"/>
    <ObjectSet id="19" name="Enlistment(s)" syn="enlistment recruitment draft" lex="0"/>
    <ObjectSet id="20" name="Death" syn="death decease" dnf="(~last)" lex="0"/>
    <ObjectSet id="21" name="Location" syn="location place site where abode" lex="1"/>
    <ObjectSet id="22" name="Street" syn="street road lane avenue ave boulevard neighborhood house borough
        ward" lex="1"/>
    <ObjectSet id="23" name="City" syn="city town village community municipality suburb hamlet parish" lex="1"/>
    <ObjectSet id="24" name="County" syn="county district region" lex="1"/>
    <ObjectSet id="25" name="State" syn="state territory province" lex="1"/>
    <ObjectSet id="26" name="Country" syn="country nation land" lex="1"/>
    <ObjectSet id="27" name="Date" syn="date when" lex="1"/>
    <ObjectSet id="28" name="Year" syn="year" lex="1"/>
    <ObjectSet id="29" name="Month" syn="month" lex="1"/>
    <ObjectSet id="30" name="Day" syn="day" lex="1"/>
    <ObjectSet id="31" name="Marital Status" syn="marital status condition married widowed single" lex="1"/>
    <ObjectSet id="32" name="Mother" syn="mother" lex="1"/>
    <ObjectSet id="33" name="Father" syn="father" lex="1"/>
    <ObjectSet id="34" name="Child" syn="" lex="1"/>

    <Relationship set1="Person" id1="0" participation1="1" set2="Age" id2="3" participation2="*"/>
    <Relationship set1="Person" id1="0" participation1="1" set2="Gender" id2="4" participation2="*"/>
    <Relationship set1="Person" id1="0" participation1=".81" set2="Relationship" id2="5" participation2="*"/>
    <Relationship set1="Person" id1="0" participation1=".52" set2="Occupation" id2="6" participation2="*"/>
    <Relationship set1="Person" id1="0" participation1="1" set2="Race" id2="7" participation2="*"/>
    <Relationship set1="Person" id1="0" participation1="1" set2="Full Name" id2="8" participation2="*"/>
    <Relationship set1="Person" id1="0" participation1="1" set2="Family" id2="1" participation2="4.8"/>
    <Relationship set1="Person" id1="0" participation1="*" set2="Event" id2="2" participation2="*"/>
    <Relationship set1="Person" id1="0" participation1=".91" set2="Marital Status" id2="31" participation2="*"/>
    <Relationship set1="Person" id1="0" participation1="1" set2="Arrival(s)" id2="13" participation2="1"/>
    <Relationship set1="Person" id1="0" participation1="1" set2="Baptism(s)" id2="14" participation2="1"/>
    <Relationship set1="Person" id1="0" participation1="1" set2="Birth" id2="15" participation2="1"/>
    <Relationship set1="Person" id1="0" participation1="1" set2="Burial" id2="16" participation2="1"/>
    <Relationship set1="Person" id1="0" participation1="1" set2="Marriage(s)" id2="17" participation2="2"/>
    <Relationship set1="Person" id1="0" participation1="1" set2="Discharge(s)" id2="18" participation2="1"/>
    <Relationship set1="Person" id1="0" participation1="1" set2="Enlistment(s)" id2="19" participation2="1"/>
    <Relationship set1="Event" id1="2" participation1="1" set2="Date" id2="27" participation2="*"/>
    <Relationship set1="Event" id1="2" participation1="1" set2="Location" id2="21" participation2="*"/>
    <Relationship set1="Family" id1="1" participation1="1" set2="Location" id2="21" participation2="1.1"/>
    <Relationship set1="Father" id1="33" participation1="2.9" set2="Child" id2="34" participation2="1"/>
    <Relationship set1="Mother" id1="32" participation1="2.9" set2="Child" id2="34" participation2="1"/>
```

**Figure 2.2.2 – Part 2 of 2**

```
    <Specialization set1="Event" id1="2" set2="Arrival" id2="13"/>
    <Specialization set1="Event" id1="2" set2="Baptism" id2="14"/>
    <Specialization set1="Event" id1="2" set2="Birth" id2="15"/>
    <Specialization set1="Event" id1="2" set2="Burial" id2="16"/>
    <Specialization set1="Event" id1="2" set2="Marriage" id2="17"/>
    <Specialization set1="Event" id1="2" set2="Discharge" id2="18"/>
    <Specialization set1="Event" id1="2" set2="Enlistment" id2="19"/>
    <Specialization set1="Event" id1="2" set2="Death" id2="20"/>
    <Specialization set1="Person" id1="0" set2="Mother" id2="32"/>
    <Specialization set1="Person" id1="0" set2="Father" id2="33"/>
    <Specialization set1="Person" id1="0" set2="Child" id2="34"/>
    <Aggregation set1="Location" id1="21" participation1="1" set2="Street" id2="22" participation2="*"/>
    <Aggregation set1="Location" id1="21" participation1="1" set2="City" id2="23" participation2="*"/>
    <Aggregation set1="Location" id1="21" participation1="1" set2="County" id2="24" participation2="*"/>
    <Aggregation set1="Location" id1="21" participation1="1" set2="State" id2="25" participation2="*"/>
    <Aggregation set1="Location" id1="21" participation1="1" set2="Country" id2="26" participation2="*"/>
    <Aggregation set1="Full Name" id1="8" participation1="1" set2="First Name" id2="9" participation2="*"/>
    <Aggregation set1="Full Name" id1="8" participation1=".6" set2="Middle Name(s)" id2="10"
        participation2="*"/>
    <Aggregation set1="Full Name" id1="8" participation1="1" set2="Last Name" id2="11" participation2="*"/>
    <Aggregation set1="Full Name" id1="8" participation1=".2" set2="Title(s)" id2="12" participation2="*"/>
    <Aggregation set1="Date" id1="27" participation1="1" set2="Year" id2="28" participation2="*"/>
    <Aggregation set1="Date" id1="27" participation1="1" set2="Month" id2="29" participation2="*"/>
    <Aggregation set1="Date" id1="27" participation1="1" set2="Day" id2="30" participation2="*"/>

</Ontology>
```

## 2.3 Output

We created the XML database description, Figure 2.3.1, from the ontology in Figure 2.2.1. The “Database” element bounds the tables in the database. Each of the “Table” elements represents a Table in the database. The attribute “id” is the unique identifier for the table. The attribute “name” is the name of the table. The attribute “objectset_id” is the identifier of the non-lexical object set represented by the table. The element “Field: represents a field in the table. The attribute “name” represents the name of the field. Notice that the names of the table fields in the database correlate with the object sets in the ontology. The attribute “objectset_id” is the unique identifier for the object set represented by the field in the table. The attribute “foreign_key” is set to zero if the field is not a foreign key and is set to one if the field is a foreign key. The attribute “foreign_table” specifies the table where the field is a key.

## Figure 2.3.1

```
<Database>

    <Table id="0" name="PERSON" objectset_id="0">
        <Field name="Person_Identifier" foreign_key="0"/>
        <Field name="Full_Name" objectset_id="8" foreign_key="0"/>
        <Field name="First_Name" objectset_id="9" foreign_key="0"/>
        <Field name="Middle_Name(s)" objectset_id="10" foreign_key="0"/>
        <Field name="Last_Name" objectset_id="11" foreign_key="0"/>
        <Field name="Title(s)" objectset_id="12" foreign_key="0"/>
        <Field name="Age" objectset_id="3" foreign_key="0"/>
        <Field name="Gender" objectset_id="4" foreign_key="0"/>
        <Field name="Relationship" objectset_id="5" foreign_key="0"/>
        <Field name="Occupation" objectset_id="6" foreign_key="0"/>
        <Field name="Race" objectset_id="7" foreign_key="0"/>
        <Field name="Marital_Status" objectset_id="31" foreign_key="0"/>
        <Field name="Family_Identifier" objectset_id="2" foreign_key="1" foreign_table="1"/>
        <Field name="Arrival(s)_Identifier" objectset_id="13" foreign_key="1" foreign_table="2"/>
        <Field name="Baptism(s)_Identifier" objectset_id="14" foreign_key="1" foreign_table="2"/>
        <Field name="Birth_Identifier" objectset_id="15" foreign_key="1" foreign_table="2"/>
        <Field name="Burial_Identifier" objectset_id="16" foreign_key="1" foreign_table="2"/>
        <Field name="Marriage(s)_Identifier" objectset_id="17" foreign_key="1" foreign_table="2"/>
        <Field name="Discharge(s)_Identifier" objectset_id="18" foreign_key="1" foreign_table="2"/>
        <Field name="Enlistment(s)_Identifier" objectset_id="19" foreign_key="1" foreign_table="2"/>
        <Field name="Death_Identifier" objectset_id="20" foreign_key="1" foreign_table="2"/>
    </Table>

    <Table id="1" name="FAMILY" objectset_id="1">
        <Field name="Family_Identifier" foreign_key="0"/>
        <Field name="Location" objectset_id="21" foreign_key="0"/>
        <Field name="Street" objectset_id="22" foreign_key="0"/>
        <Field name="City" objectset_id="23" foreign_key="0"/>
        <Field name="County" objectset_id="24" foreign_key="0"/>
        <Field name="State" objectset_id="25" foreign_key="0"/>
        <Field name="Country" objectset_id="26" foreign_key="0"/>
    </Table>

    <Table id="2" name="EVENT" objectset_id="2">
        <Field name="Event_Identifier" foreign_key="0"/>
        <Field name="Location" objectset_id="21" foreign_key="0"/>
        <Field name="Street" objectset_id="22" foreign_key="0"/>
        <Field name="City" objectset_id="23" foreign_key="0"/>
        <Field name="County" objectset_id="24" foreign_key="0"/>
        <Field name="State" objectset_id="25" foreign_key="0"/>
        <Field name="Country" objectset_id="26" foreign_key="0"/>
        <Field name="Date" objectset_id="27" foreign_key="0"/>
        <Field name="Day" objectset_id="30" foreign_key="0"/>
        <Field name="Month" objectset_id="29" foreign_key="0"/>
        <Field name="Year" objectset_id="28" foreign_key="0"/>
    </Table>

    <Table id="3" name="MOTHER_CHILD" objectset_id="32">
        <Field name="Mother_Identifier" objectset_id="32" foreign_key="1" foreign_table="0"/>
        <Field name="Child_Identifier" objectset_id="34" foreign_key="1" foreign_table="0"/>
    </Table>

    <Table id="4" name="FATHER_CHILD" objectset_id="33">
        <Field name="Father_Identifier" objectset_id="33" foreign_key="1" foreign_table="0"/>
        <Field name="Child_Identifier" objectset_id="34" foreign_key="1" foreign_table="0"/>
    </Table>

</Database>
```

The output file contains a sequence of SQL insert statements. Since the hand-written text in the values cells are not machine readable, the algorithm places the coordinates of each value cell into their proper places in a database. Figure 2.3.2 shows part of the resulting SQL statements for the table in Figure 1.0. A complete set of SQL statements for the image in Figure 1.0 is in Appendix 2.

```
INSERT INTO PERSON (person_identifier, age, relationship, occupation, name_identifier,
        family_identifier, birth_event_identifier) (1, '335,80,350,92', '255,80,301,92', '369,80,507,92', 1, 1, 1)

INSERT INTO NAME (name_identifier, full_name) (1, '152,80,255,92')

INSERT INTO FAMILY (family_identifier, location_identifier) (1, 1)

INSERT INTO BIRTH_EVENT (birth_event_identifier, location_identifier) (1, 2)

INSERT INTO LOCATION (location_identifier, street, city, county) (1, '117,80,135,92', '7,34,93,50',
        '600,33,681,50')

INSERT INTO LOCATION (location_identifier, location) (2, '507,80,625,92')

INSERT INTO PERSON (person_identifier, age, relationship, name_identifier, family_identifier,
        birth_event_identifier) (2, '350,92,369,104', '255,92,301,104', 2,1, 2)

INSERT INTO NAME (name_identifier, full_name) (2, '152,92,255,104')

INSERT INTO BIRTH_EVENT (birth_event_identifier, location_identifier) (2, 3)

INSERT INTO LOCATION (location_identifier, location) (3, '507,92,625,104')

...
```

**Figure 2.3.2**

# CHAPTER 3 – METHODOLOGY

## 3.1 Feature Extraction

Let C be the union of the set of all label cells L, the set of all value cells V, and the set of all empty cells E. Let O be the set of all object sets in the genealogical ontology. Let $D = C \cup O$. An *evidence matrix* **M** is a two-dimensional matrix that relates $F \subseteq D$ in the first dimension to $S \subseteq D$ in the second dimension. The value at position $\mathbf{M}_{ij}$ is a real number between zero and one that represents the confidence of a relationship between an element $f_i$ in F and an element $s_j$ in S. A confidence value of zero indicates that the relationship is inappropriate, while a confidence value of one indicates that the relationship is certain. We generate confidence values by measuring features about the table cells and object sets in D that support or refute the existence of the relationship expressed by **M**.

By examining a training set, we discovered seven relationships that should be identified in order to recognize the record structure of cells in a genealogical table. (1) A label cell describes the type of information contained in the row or column of the value cells it heads. (2) A value cell is in a row with other value cells. (3) A value cell is in a column with other value cells. (4) A pair of value cells that contains the same type of information usually has the same dimensions. (5) Label cells can form a multi-level label. (6) The text of a label cell can match an object set in the genealogical ontology. (7) The value cells associated with a label cell can factor the value cells of another label cell.

We created seven evidence matrices to represent each of the seven relationships. The first matrix **LV** relates a table's label cells in L to its values cells in V. We generate **LV** to determine whether a value cell falls within the row or column headed by a label cell. The second matrix **VR** horizontally relates value cells in V to other value cells in V. We create **VR** to discover how likely value cells are in the same row with other values cells. The third matrix **VC** vertically relates value cells in V to other value cells in V. We create **VC** to discover how likely value cells are in the same column with other value cells. The fourth matrix **VV** relates the dimensions of a value cell in V to another value cell in V. Based on dimensional similarity, we create **VV** to discover how likely value cells contain the same type of information. The fifth matrix **LL** relates the label cells of L in a table to other label cells of L that are not associated with values but that form a composite label with other label cells. The sixth matrix **LO** matches a table's label cells in L with the object sets in O in the genealogical ontology. The matrix **LO** allows the algorithm to identify the types records contained in the table. The seventh matrix **FM** describes how confidently a label cell $l_i$ in L has values cells that factor the value cells of other label cells in a table. This matrix is the factoring matrix.

### 3.1.1 **LV:** Label Cells Describe the Content of which Value Cells

A label cell describes the type of information contained in the row or column of the value cells it heads. Let $l_i$ be a label cell in L and $v_j$ be a value cell in V. The algorithm determines if label $l_i$ heads the row or column of $v_j$ by extracting two features from $l_i$ and $v_j$, multiplying the two produced confidence values, and storing the result in

**$LV_{ij}$.** The closer $\mathbf{LV}_{ij}$ is to 1, the more confident the algorithm is that $l_i$ heads the row or column of $v_j$.

The first feature measures whether a continuous path of cells exists between a label cell $l_i$ and a value cell $v_j$ through a horizontal or vertical sequence of adjacent value cells or empty cells. If a path does not exist between $l_i$ and $v_j$, then the label cell does not head the row or column of the value cell. The algorithm computes the path between cells by first determining the cell directly to the left, right, above and beneath each table cell. To compute the cell directly to the left of any given cell c in C the algorithm looks at all cells with a midpoint to the left of c's midpoint and between c's top and bottom coordinates. It then selects the cell with the shortest Euclidean distance between its midpoint and c's midpoint. The algorithm computes the cell directly right, above, and beneath c in a similar manner. We use this technique because genealogical tables on microfilm are often warped or slightly rotated. This means that a horizontal line may not pass though every cell in a row nor likewise a column. Since each cell chooses locally its neighbor to the left, right, above, and beneath, the algorithm can relate cells at the opposite ends of a row or column that is slightly rotated or warped.

By observing the training set, we discovered that genealogical tables most often contain columns of value cells headed above by a label cell. In contrast, the least common orientation occurs when a label cell describes the contents of a value cell that is above it. To enforce this preference, we varied the confidence values assigned to label-cell, value-cell pairs in **LV** by the likelihood of their orientation. If a path of cells exists

between label cell $l_i$ and value cell $v_j$, and $v_j$ is below $l_i$, then the algorithm sets $\mathbf{LV}_{ij}$ to 1. If a path exists between $l_i$ and $v_j$, and $v_j$ is to the right of $l_i$, then the algorithm sets $\mathbf{LV}_{ij}$ to .75. If a path exists between $l_i$ and $v_j$, and $v_j$ is to the left of $l_i$, then the algorithm sets $\mathbf{LV}_{ij}$ to .5. If a path exists between $l_i$ and $v_j$, and $v_j$ is above $l_i$, then the algorithm sets $\mathbf{LV}_{ij}$ to .25. If a path does not exist between $l_i$ and $v_j$, then $\mathbf{LV}_{ij}$ is set to 0.

The second feature compares the height and width of each value cell with the height and width of each label cell. We extract this feature because a value cell in a row generally has the same height as the label cell that heads the row, and a value cell in a column generally has the same width as the label cell that heads the column. To accomplish this, the algorithm calculates the difference in height and the difference in width between each label-cell value-cell pair using the first equation in Figure 3.1.0. It then generates the confidence value by dividing the smaller of the two differences by the largest minimum difference for all label-cell value-cell pairs. The algorithm then subtracts the result from one to ensure that small differences receive confidence values near one and large differences receive confidence values near zero.

$$\mathbf{Difference}_{ij} = \text{Minimum}\{\ |\text{Height}(l_i) - \text{Height}(l_j)|, |\text{Width}(l_i) - \text{Width}(l_j)|\ \}$$

$$\mathbf{LV}_{ij} = 1 - \frac{\text{Difference}_{ij}}{\text{Maximum}\{\text{Difference}_{ab},\ 1 \le a \le |L|,\ 1 \le b \le |V|\ \}}$$

**Figure 3.1.0**

### 3.1.2 **VR**: Value Cells in Same Row

A value cell can be in the same row with other value cells. To determine whether value cell $v_i$ is in the same row as value cell $v_j$, the algorithm measures whether a continuous path of cells exists between value cell, $v_i$, and a value cell, $v_j$, through a horizontal sequence of adjacent value cells or empty cells. We extract this feature because value cells in the same row of a table in the training set are connected by a sequence of value cells or empty cells. If a path does not exist between $v_i$ and $v_j$, then $v_i$ is not in the same row as $v_j$. The algorithm computes the path between a pair of value cells using the method described to identify a path of cells between a label cell and value cell in Section 3.1.1. If a path exists between $v_i$, and $v_{j,}$ then the algorithm sets $\mathbf{VR}_{ij}$ to 1, else the algorithm sets $\mathbf{VR}_{ij}$ to 0. We set $\mathbf{VR}_{ij}$ to 0 because when a path of cells does not exist between two value cells, they cannot be in the same row.

### 3.1.3 **VC**: Value Cells in Same Column

The algorithm populates the **VC** matrix by attempting to identify a continuous sequence of value cells and empty cells between each value-cell pair. The algorithm identifies the path with the same method used to populate **VR**, except it looks for a vertical sequence of cells instead of a horizontal sequence. If a path exists between $v_i$, and $v_{j,}$ then $v_i$ and $v_j$ are in the same column, and the algorithm sets $\mathbf{VC}_{ij}$ to 1, else the algorithm sets $\mathbf{VC}_{ij}$ to 0.

### 3.1.4 **VV**: Value Cells Similarity

We observed from the training set that value cells in a table containing the same type of data generally had the same height and width. To support this heuristic, the algorithm extracts two features that compare both the height and width of the each value-cell pair respectively. For each value cell pair $v_i$, and $v_j$, the algorithm multiplies the confidence value produced by each feature and stores the result at $\mathbf{VV}_{ij}$. The first feature measures the difference in the height between $v_i$ and $v_j$. To generate a confidence value, the algorithm normalizes this difference by dividing it by the largest height difference for all value-cell pairs and stores it at position $\mathbf{VV}_{ij}$. The second feature measures the difference in the width between value cell $v_i$ and value cell $v_j$. This algorithm normalizes this difference by dividing it by the largest width difference for all value-cell pairs and stores it at position $\mathbf{VV}_{ij}$. The algorithm then subtracts the produce of both confidence values (for each feature) from one to ensure that small differences receive confidence values near one and large differences receive confidence values near zero.

### 3.1.5 **LL**: Composite Labels

The algorithm extracts three features to determine if a pair of label cells forms a composite label. The goal of this matrix is to create a graph of connected the label cells forming the same composite label. The algorithm multiplies the confidence value generated by each feature, for each label cell $l_i$ and label cell $l_j$ and stores the result at $\mathbf{LL}_{ij}$.

The first feature used to populate the **LL** matrix measures the distance between the midpoints of the each label cell with every other label cell. We do this because in the training set, label cells in the same composite label are close in Euclidean space. To compute the confidence value for a label cell $l_i$ and a label cell $l_j$ for this first feature, the algorithm first finds the distance between midpoints of $l_i$ and $l_j$. It divides this distance by the maximum distance between all label-cell pairs and subtracts this result from one. If the resulting value of $\mathbf{LL}_{ij}$ is less than zero, then the algorithm sets $\mathbf{LL}_{ij}$ to zero.

The second feature measures whether a line exists that passes through the midpoints of a pair of a label cells and through the rectangle of any value cell. We do this because a composite label must eventually connect with the value cells it heads. The algorithm first calculates the slope and intercept of the line that runs between the midpoints of a pair of label cells. It then solves for the intersection of the line with each value cell. If such an intersection exists for the line between the midpoints of a label cell $l_i$ and a label cell $l_j$ then the algorithm lets the confidence value be one for this feature, otherwise it lets the confidence value be zero at $\mathbf{LL}_{ij.}$

The third features measures whether labels cells share a common border. We measure this feature because the label cells within a composite label in the training set share a border. For each label cell $l_i$ the algorithm finds all label cells that share a border to the left, right, above and below with $l_i$. To compute the cell directly to the left of $l_i$, the algorithm looks at all label cells with a midpoint to the left of $l_i$'s midpoint and between $l_i$'s top and bottom coordinates. A label cell $l_i$ shares a border with label cell $l_j$ on the left,

if the distance between the left border of $l_i$ and the right border of $l_j$ is less than or equal to 3 pixels. We use a threshold of 3 because the padding between label cells in the training set is no more than 3. The algorithm calculates label cells bordering to the right, above and blow in the same manner. If a border exists between $l_i$ and $l_j$, the algorithm lets the confidence value be one for this feature otherwise the algorithm lets the confidence value be zero at $\mathbf{LL}_{ij}$**.** Thus, since we multiply the confidence values of the three extracted features, the final $\mathbf{LL}_{ij}$ value is either zero or the value of the first feature.

### 3.1.6 **LO**: Label Cell maps to Object Sets

Figure 3.1.6 shows an example of a match (shown by the arrow) between the label cell with printed text “The name of every person whose place of abode on the first day of June, 1870. was in this family.” and the object set “Full Name” in the genealogical ontology. The algorithm extracts two features to map each label cell to zero or more object sets. It then multiplies the confidence value generated by each feature for each label cell $l_i$ and object set pair $o_j$ and stores the result at $\mathbf{LO}_{ij}$.

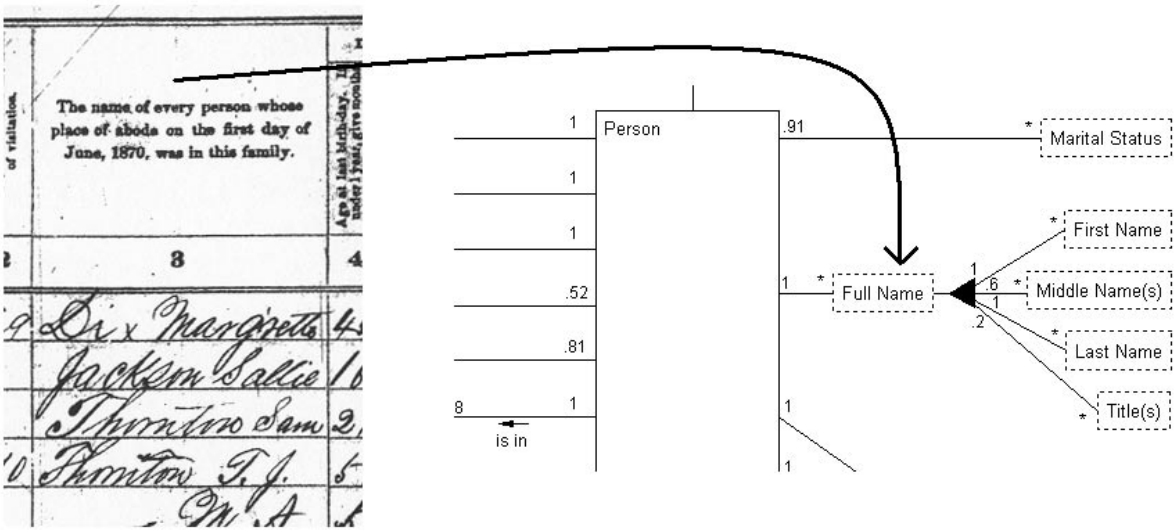



**Figure 3.1.6**

The first feature matches the synonyms of the object sets in Figure 2.2.2, with the text of the label cells in a table while giving preference to words matched at the beginning of the character string for a label cell. First, the algorithm removes the *stop words* (the articles, prepositions, and other common words) from the printed text associated with each label cell. We generate the list of stop words by taking the common words from the labels of the table cells in our training set. If a synonym for an object set, $o_j$, matches a word in the label cell, $l_i$, then the algorithm sets $\mathbf{LO}_{ij}$ for the first feature to 1 divided by the position of the matched word in the label cell's character string without the stop words. We divide by this position of the word because in our training set the label cells generally have representative words at the beginning of the label's character string.

The second feature looks at the order in which the object sets match words in the character string of a label cell. An object set matches a word w if w is in the synonym list of the object set. The algorithm looks at each word in order from the beginning to the end of the character string. If a word of the label cell $l_i$ matches an object set $o_j$ then the algorithm multiplies the confidence value at $\mathbf{LO}_{ij}$ (as set for the first feature) by one divided by the count of previously matched words in the character string of the label cell. We divide by the count of previously matched words because in the training set the order that object sets matched the words of a label cell generally indicated their relevance to the label cell.

### 3.1.7 **FM:** Label Cell Factoring

Since the labels are not yet associated with value cells, the algorithm can only use the position of a label cell in relation to the position of another label cell to predict if its value cells factor the other label cell's value cells. In the training set, the value cells of a label cell only factor the value cells of another label cell if the label cell is above or to the left of the other label cell. To support this preference, the algorithm determines if a label cells is above or to the left of every other label cell. If the midpoint of a label cell $l_i$ is to the left or if the midpoint of $l_i$ is above the midpoint of label cell $l_j$ then the algorithm sets $FM_{ij}$ to 1; otherwise it sets it to zero.

## 3.2 Correlation Rules

A *correlation rule* is an expression for updating the values of one evidence matrix using the values of another evidence matrix, information from the genealogical ontology,

or other value elsewhere in the same evidence matrix. These correlation rules attempt to find corroborating evidence to increase confidence values or to find conflicting evidence to decrease confidence values.

The algorithm iteratively applies a set of six correlation rules to refine the seven previously populated evidence matrices. We iterate through the rules because each rule makes only a small change to the confidence values of a matrix at each step or only performs an action when the values of in an evidence matrix reach a predefined threshold. Thus one rule cannot dominate the effects of other rules and each rule is dependant upon the changes made by other rules. In addition, the rules work together to allow the confidence values to gradually settle. The algorithm iterates until no rules can update confidence values or until 1000 iteration, which ever is first. We set the maximum number of iterations to 1000 because applying more than 1000 iterations did not affect the results of the algorithm on the training set.

The first correlation rule uses the confidence values in **VV** that relate geometrically similar value cells to support the concept that value cells associated to the same label cell in **LV** should be geometrically similar. The evidence matrix **LV** stores the confidence value of a label cell heading the column or row of a particular value cell. The rule disassociates a value cell that is not similar to other value cells associated with its label cell. Let A be the set of all value cells that are headed by a label cell $l_i$ with a confidence value greater than .5 in **LV**. For each value cell $v_j$ in A, we consider every other value cell $v_k$ in A (i.e. with $v_j$ fixed, we consider all k where $1 \leq k \leq |A|$ and $j \neq k$).

If the confidence value of every $v_j$-$v_k$ pair is less than .5 in **VV**, then the algorithm disassociates $v_j$ from A by placing a zero at position $\mathbf{LV}_{ij}$. Note that i references the label cell $l_i$, and j references the value cell $v_j$ to be removed.

The second correlation rule maps a label cell that relates to two or more objects sets of an aggregation (in the ontology) to only the aggregate head. For example, in Figure 3.2.0 a label cell has a high confidence of association with both "First name" and "Last name" in the matrix **LO**. The matrix **LO** describes how confidently each label cell maps to each object set in the ontology. In this case, the label cell "NAME and Surname of each Person" should map to the head of the aggregation, "Full name". To support this, the algorithm checks if each label is associated with two or more objects in an aggregation with a confidence value greater than .5 in **LO** for each object set. If such an association exists between a label and two or more object sets, the algorithm maps the label cell to the head of the aggregation. It accomplishes this by placing a one in the **LO** matrix at $\mathbf{LO}_{i,head}$ to relate the label $l_i$ to the head object set of the aggregation $o_{head}$. The other objects in the aggregation are disassociated from the label cell by placing a zero in the **LO** matrix.

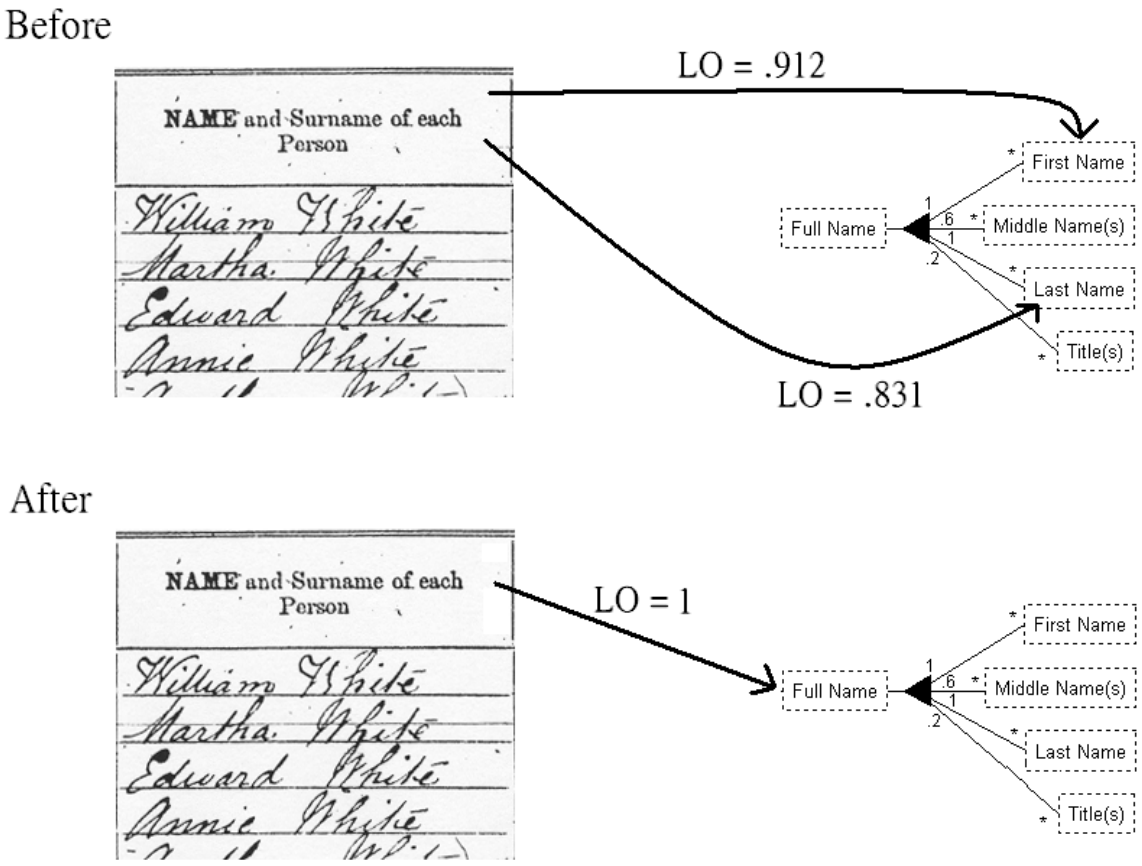
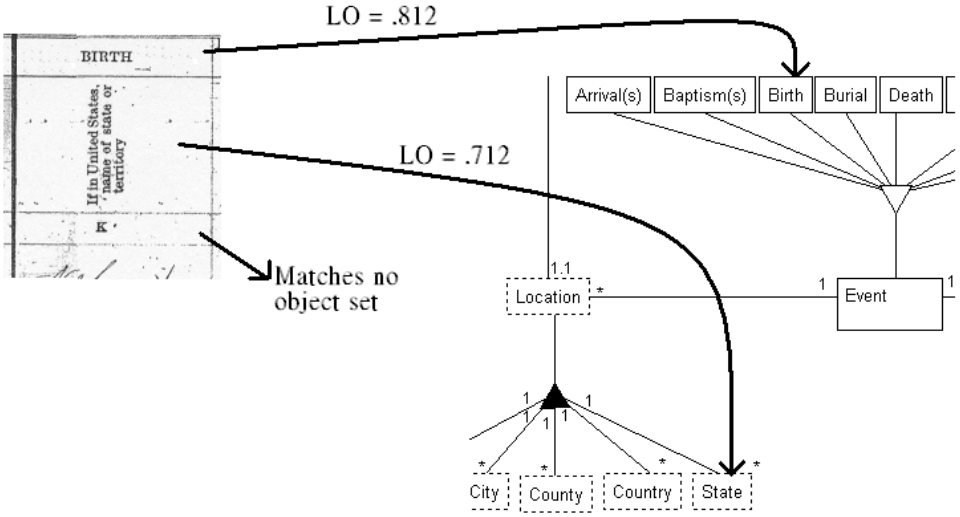



**Figure 3.2.0**

The third correlation rule raises the confidence values for the label cells that associate with value cells in unlikely geometric orientations. Remember that when the **LV** matrix was populated, labels associated with values to the right of them or above them were given confidence value cells of .5 and .25 respectively. Now, the algorithm raises those confidence values if they are the largest confidence values for a value cell in **LV**. First, the algorithm associates each label cell with the value cells that have the highest confidence of association with it in the **LV** matrix. Next, it increments the confidence value in matrix **LV** by .1 at each step of the iteration for each of these label-cell, value-cell pairs.

The fourth correlation rule distributes label-to-object set mappings across multi-level groups of label cells. We do this to combine groups of label cells into one composite label. For example, Figure 3.2.1 is a three-level group of label cells where the label cell “Birth” maps to the object set “Birth”, the label cell “If in United States, name of state or territory” maps to “State” and the label cell “K” does not map to any object set in the genealogical ontology. To create the composite label, the algorithm first gathers all the multi-level groups of labels using the confidence values in the **LL** matrix. If the confidence value for two labels is greater than .5, then the algorithm groups the labels. We do this recursively to create groups of multi-level label cells. Next, the algorithm copies the object sets associated with each label in the group to the other object sets in the label cell in the group. By distributing the object set mappings to each label cell in a composite label, we aggregate the label cells into one composite label with one set of object set mappings.

Before




After

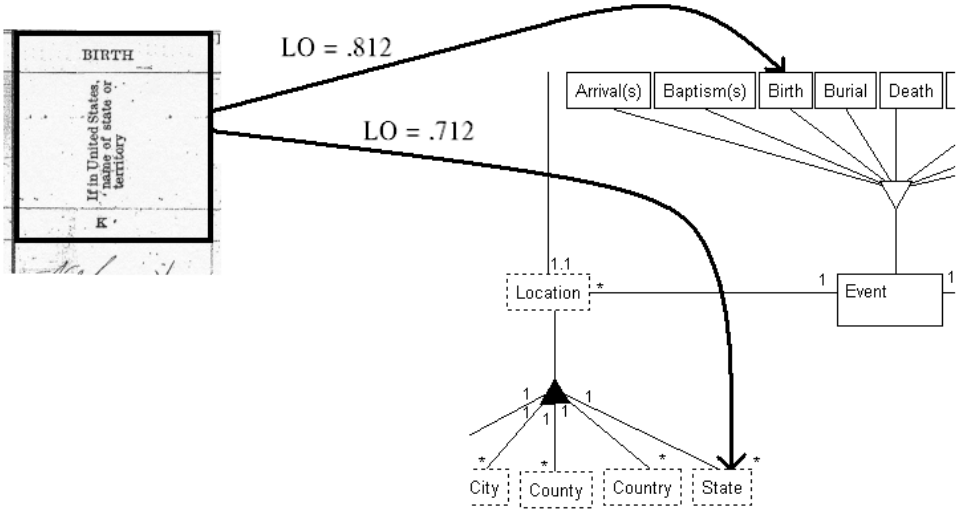



**Figure 3.2.1**

The fifth correlation rule refines the factoring matrix, **FM,** using a genealogical ontology, O. An expected cardinality ratio exists for each pair of object sets $o_i$ and $o_j$ in O. For example, the cardinality ratio between the object set "Person" and "Family" is

equal to 4.8 in the genealogical ontology; meaning that if both "Person" and "Family" exist in a table, there should be approximately 4.8 people in each family. For each pair of label cells, the algorithm considers each label-cell object-set pair that has a confidence value greater than .5 in **LO**. The algorithm then updates the factoring matrix, **MF**, using the following rule: **FM**ij = **FM**ij * [1 - | $O_{ij} - N_i/N_j$| + .6]. We determined the threshold .6 empirically using the training set data. If the resulting value of $\mathbf{FM}_{ij}$ is less than zero, the algorithm sets $\mathbf{FM}_{ij}$ to zero; likewise if the resulting value of $\mathbf{FM}_{ij}$ is greater than one, the algorithm sets $\mathbf{FM}_{ij}$ to one. The term $N_i$ is the number of non-empty value cells associated with the label cell $l_i$ in L. The algorithm calculates the value $N_i$ for a label cell $l_i$ in L by counting the non-empty value cells associated with $l_i$ with a confidence value greater than .5 in matrix **LV.** Similarly the algorithm counts the number of non-empty value cells $N_j$ associated with for a label cell $l_j$ in L. This update rule compares the expected cardinality ratio $O_{ij}$ for a pair of label cells with the observed cardinality ratio $N_i/N_j$ and updates the respective value in **MF**.

The sixth correlation rule increases the confidence value for a label cell that maps to two object sets in **LO**, if the object sets are separated by only one edge in the genealogical ontology. The matrix **LO** maps label cells in L to the object sets in the genealogical ontology. When the algorithm populates **LO**, it assigns lower confidence values to object sets that match words at the end of an object set's character string. The goal of this rule is to increase the confidence values for these object sets if they relate to the other object sets that map to the label cell. If a label cell matches the child of an aggregation or a specialization object set in the ontology, we only consider the head of

the aggregation or specialization. For example, consider Figure 3.2.2, which shows part of the genealogical ontology. Suppose a label cell maps to the object set "State" with a confidence value of .781 in **LO** and maps to the object set "Birth" with a confidence value of .321 in **LO**. Since "State" is part of the aggregation "Location" and "Birth" is a specialization of "Event", the algorithm compares the distance between "Location" and "Event". If two object sets are separated by only one edge in the ontology, the algorithm changes the lesser of the label cell's two object-set confidence values to the higher of the two confidence values. Figure 3.2.2 shows "Location" and "Event" separated by only one edge (in bold). Thus in the example, the algorithm increases the confidence value of the label cell matching "State" to .781.

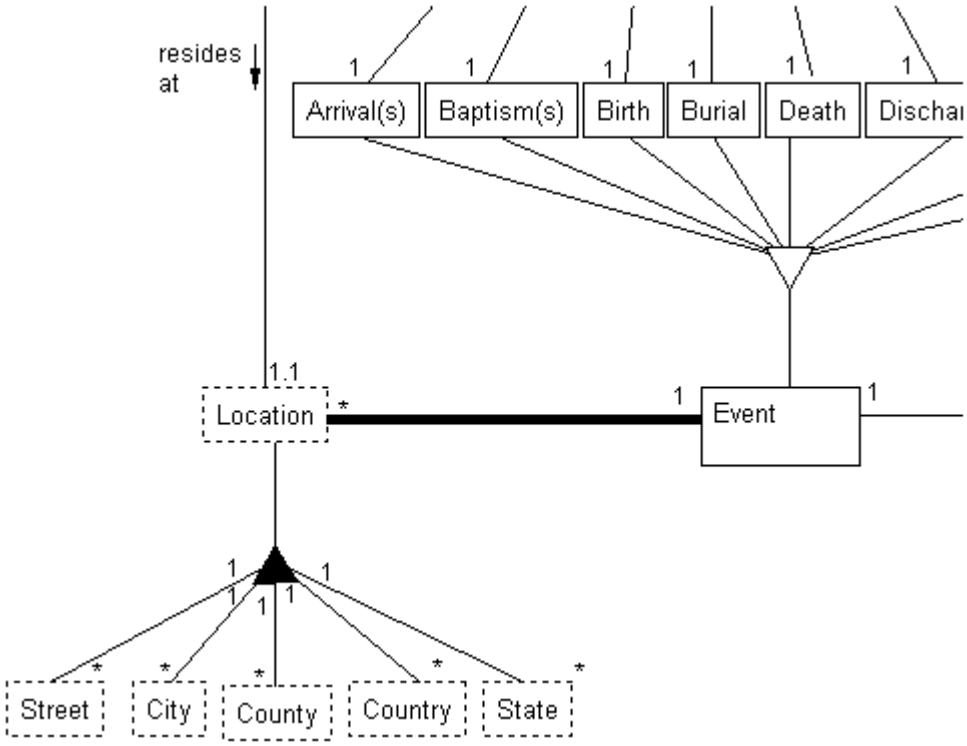



**Figure 3.2.2**

## 3.3 Record Verification and Storage

The algorithm generates records by observing the confidence values within the evidence matrices. It gathers labeled value cells into records one by one, and begins a new record when the participation constraints of the ontology prevent a value cell from being included in the current record.

We visually present the value cells contained within each record for a user to verify (shown in Appendix 3). The user can verify that the algorithm correctly associated value cells with label cells, that it correctly matched the label cells to object sets in the genealogical ontology, and that it correctly identified the boundaries of the record.

Once verified, the algorithm stores value cells of the extracted records in a database. For each value cell that is part of a record, the algorithm stores its coordinates in a table of a database by creating an SQL insert statement. The algorithm only stores the coordinates of the value cells because the hand-written data they contain cannot yet be automatically translated from the original microfilm table. Later, individuals or algorithms can extract the handwritten values and associate them with their coordinates in the database records [Jag97].

# CHAPTER 4 – EXPERIMENTAL RESULTS AND ANALYSIS

## 4.1 Measurements

We ran the algorithm on a training set of 25 tables and a test set of 75 tables. We gathered the tables from 25 different microfilm rolls (listed in Appendix 4). We measured the overall success of the algorithm by observing the precision, recall, and accuracy of the fields in the SQL statements. We measured the precision of the SQL statement fields by dividing the number of correctly populated fields by the total number of populated fields. We measured the recall by dividing the number of correctly populated fields by the total number of value cells the algorithm should have correctly placed in the table. We calculated the accuracy for each of the results using the harmonic mean: 2 / ((1 / precision) + (1 / recall)) [BR99, page 82].

In addition, we measured the intermediate success of the algorithm by observing the precision, recall, and accuracy of the results produced by each of the seven evidence matrices. We took the measurements for each of the evidence matrices after the algorithm populated the matrices and applied the correlation rules.

The first evidence matrix, **LV**, associates a table's label cells with value cells. We measured the precision of **LV** by dividing the number of correctly identified label-value pairs in **LV** by the total number of label-value pairs in **LV**. We measured the recall of **LV** by dividing the number of correctly identified label-value pairs in **LV** by the total number of label-value pairs in the table.

The second evidence matrix, **VR**, identifies value-cell pairs that are in the same row. We measured the precision of **VR** by dividing the number of value cells placed in the correct row by the total number of value cells placed in rows by **VR**. We measured the recall of **VR** by dividing the number of value cells placed in the correct row by the total number of value cells in rows of the table.

The third evidence matrix, **VC**, identifies value-cell pairs that are in the same column. We measured the precision of **VC** by dividing the number of value cells placed in the correct column by the total number of value cells placed in columns by **VC**. We measured the recall of **VC** by dividing the number of value cells placed in the correct column by the total number of value cells in columns in the table.

The fourth evidence matrix, **VV**, measures the geometric similarity of value-cell pairs. We counted value-cell pairs whose geometric similarity confidence exceeded .5. We measured the precision of **VV** by dividing the number of value cells that the algorithm correctly associated with similar value cells (it has a **VV** confidence greater than or equal to .5 with each similar value cell) by the total of value cells associated by **VV**. We measured the recall of **VV** by dividing the number of value cells the algorithm correctly associated with similar value cells by the total number similar value cell associations in the table.

The fifth evidence matrix, **LL**, measures how likely a label cell is the parent of another label cell in a multi-level label. We measured the precision of **LL** by dividing

the number of correctly identified multi-level labels by the number of multi-level labels identified in **LL**. We measured the recall of **LL** by dividing the number of correctly identified multi-level labels by the number of multi-level labels in the table.

The sixth evidence matrix, **LO**, maps the machine-printed text of each label cell to object sets in the genealogical ontology. We measured the precision of **LO** by dividing the number of correctly mapped label cells by the number of mapped label cells identified in **LO**. We measured the recall of **LO** by dividing the number of correctly mapped label cells by the number of label cells the algorithm should have mapped to the ontology in the table.

The seventh evidence matrix, **FM**, measures how confidently a label cell's value cells factor the record containing another label cell's value cells. We measured the precision of **FM** by dividing the number of correctly factored label cells by the number of label cells factored in **FM**. We measured the recall of FM by dividing the number of correctly factored label cells by the number of label cells the algorithm should have factored in the table.

## 4.2 Training Set

We identified and adjusted the relationships, extracted features, correlation rules, and threshold values for the algorithm using the 25 tables in the training set. We gathered the 25 tables from five different microfilm rolls. Each microfilm roll contained tables with a different layout. For each of the tables on the same roll of microfilm, we

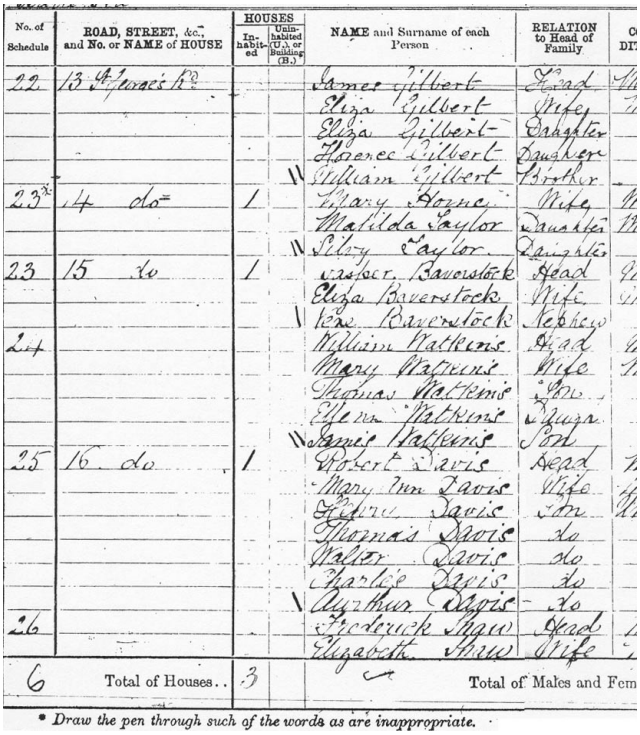
ran the algorithm and averaged the resulting values of the respective evidence matrices. The algorithm then produced SQL statements for each of the tables. Figure 4.2.0 shows the precision, recall and accuracy we were able to achieve with the algorithm on the training set.

| **Training Set** | | | | |
|---|---|---|---|---|
| Relationships | Matrix | Precision | Recall | Accuracy |
| Label Cell, Value Cell Pairs | **LV** | 100 | 100 | 100 |
| Value Cell Pairs (row) | **VR** | 100 | 100 | 100 |
| Value Cell Pairs (column) | **VC** | 100 | 100 | 100 |
| Value Cell Pairs (similar) | **VV** | 100 | 100 | 100 |
| Multi-level Labels | **LL** | 100 | 100 | 100 |
| Label Cell to Object Set Mappings | **LO** | 74.45 | 100 | 84.65 |
| Factored Label Cells | **FM** | 100 | 100 | 100 |
| Fields in Database Tables | **Overall** | 99.42 | 100 | 99.71 |

**Figure 4.2.0**

The algorithm correctly identified 100 percent of the geometric relationships detected by the matrices **LV**, **VR**, **VC**, **VV**, and **LL**. In addition, the algorithm correctly mapped all the label cells that should have mapped to object sets in the ontology. Yet its precision for matrix **LO** was only 74.45 percent because the algorithm also mapped some irrelevant labels cells to object sets. This occurred because it performed only naive natural-language processing on the labels cells. The overall precision for the populated fields in the database tables is 99.4 percent (and not 100 percent) because of ambiguous factorings within some of the tables. For example, Figure 4.2.1 contains an ambiguous

factoring. In this figure, both the label cell "No. of Schedule" and the label cell "ROAD, STREET, &c., and No. or NAME or HOUSE" factor each person record on each row of the table. Yet, since the algorithm maps only the label cell "ROAD, STREET, &c., and No. or NAME or HOUSE" to the genealogical ontology, it combines the third and fourth families into the same family. This occurs because the cell under the label cell "ROAD, STREET, &c., and No. or NAME or HOUSE" is empty for the forth family. We can overcome ambiguous factoring by reading the content (hand-writing) of each value cell to infer the boundary of a family record in the table. For example in Figure 4.2.1, the algorithm could use the text of the value cells, such as "Head", "Wife", "Daughter" and "Son", under the label cell "RELATION to Head of Family" to determine the members of each family.



| No. of Schedule | ROAD, STREET, &c., and No. or NAME of HOUSE | HOUSES Inhabited | HOUSES Uninhabited (U.) or Building (B.) | NAME and Surname of each Person | RELATION to Head of Family |
|---|---|---|---|---|---|
| 22 | 13 St James's Rd | | | James Gilbert | Head |
| | | | | Eliza Gilbert | Wife |
| | | | | Eliza Gilbert | Daughter |
| | | | | Florence Gilbert | Daughter |
| | | | \| | William Gilbert | Brother |
| 23 | 14 do | 1 | | Mary Hayne | Wife |
| | | | | Matilda Taylor | Daughter |
| | | | \| | Lilly Taylor | Daughter |
| 23 | 15 do | 1 | | Casper Baverstock | Head |
| | | | | Eliza Baverstock | Wife |
| | | | \| | Ezra Baverstock | Nephew |
| 24 | | | | William Watkins | Head |
| | | | | Mary Watkins | Wife |
| | | | | Thomas Watkins | Son |
| | | | | Ellen Watkins | Daughter |
| | | | \| | James Watkins | Son |
| 25 | 16 do | 1 | | Robert Davis | Head |
| | | | | Mary Ann Davis | Wife |
| | | | | Edwin Davis | Son |
| | | | | Thomas Davis | do |
| | | | | Walter Davis | do |
| | | | | Charlie Davis | do |
| | | | \| | Arthur Davis | do |
| 26 | | | | Frederick Shaw | Head |
| | | | | Elizabeth Shaw | Wife |
| 6 | Total of Houses.. | 3 | | | Total of Males and Fem |

* *Draw the pen through such of the words as are inappropriate.*

**Figure 4.2.1**

4.3 Test Set

To evaluate the algorithm, we applied it to a set of 75 different tables in the test set. Unlike with the training set, we did not use the test set to change the relationships, extracted features, correlation rules or threshold values of the algorithm. We gathered the 75 tables from 15 different microfilm rolls. Each microfilm roll contained a different layout. For each of the tables on the same roll of microfilm, we ran the algorithm and averaged the values for the respective evidence matrices. The algorithm then produced SQL statements for each of the tables. Figure 4.3.0 shows the precision, recall and accuracy for the algorithm on the training set.

| **Test Set** | | | | |
|---|---|---|---|---|
| Relationships | Matrix | Precision | Recall | Accuracy |
| Label Cell, Value Cell Pairs | **LV** | 100 | 98.12 | 98.97 |
| Value Cell Pairs (row) | **VR** | 100 | 100 | 100 |
| Value Cell Pairs (column) | **VC** | 100 | 100 | 100 |
| Value Cell Pairs (similar) | **VV** | 100 | 100 | 100 |
| Multi-level Labels | **LL** | 100 | 99.67 | 99.82 |
| Label Cell to Object Set Mappings | **LO** | 84.98 | 92.76 | 88.18 |
| Factored Label Cells | **FM** | 100 | 93.4 | 93.47 |
| Fields in Database Tables | **Overall** | 93.20 | 92.41 | 92.15 |

**Figure 4.3.0**

The algorithm achieved lower overall precision and recall measurements on the test set than on the training set and thus also lower accuracy. Notice that while the intermediate precision increased for **LO** (and remained constant for the other matrices),

the overall precision decreased. The overall precision decreased mainly because the incorrectly matched label cells in the test set associated with a larger percentage of a table's value cells than did the incorrectly matched label cells in the training set. Thus, when the algorithm incorrectly matched a label cell to an object set in the test set, it inserted more value cells incorrectly in SQL fields. The overall recall decreased because the intermediate recall of the matrices **LV**, **LL**, **FM** and **LO** decreased. We further discuss both precision and recall failures in Section 4.4.

## 4.4 Discussion

The algorithm successfully identified seven different types of geometric and ontological structure. We present and discuss each of the seven types by giving an example of each to illustrate the effect of the extracted features. Later in the discussion, we also describe several failures.

(1) In most cases, the algorithm correctly mapped labels to object sets. The algorithm produced the SQL in Figure 4.3.0 B from the first row in the Figure 4.3.0 A. In this example, the algorithm discovered three "People" records. This occurred because the algorithm matched the object set "Father" to the label cell "Place of Birth of the FATHER of this person, naming the State or Territory of the United States, or the Country, if of foreign Birth" and the object set "Mother" to the label cell "Place of Birth of the MOTHER of this person, naming the State or Territory of the United States, or the Country, if of foreign Birth". In addition, it associated the correct birthplace with each of the three "People" records. Thus, the algorithm created the relationship between child and mother

and child and father using the MOTHER_CHILD table and FATHER_CHILD database tables respectively.

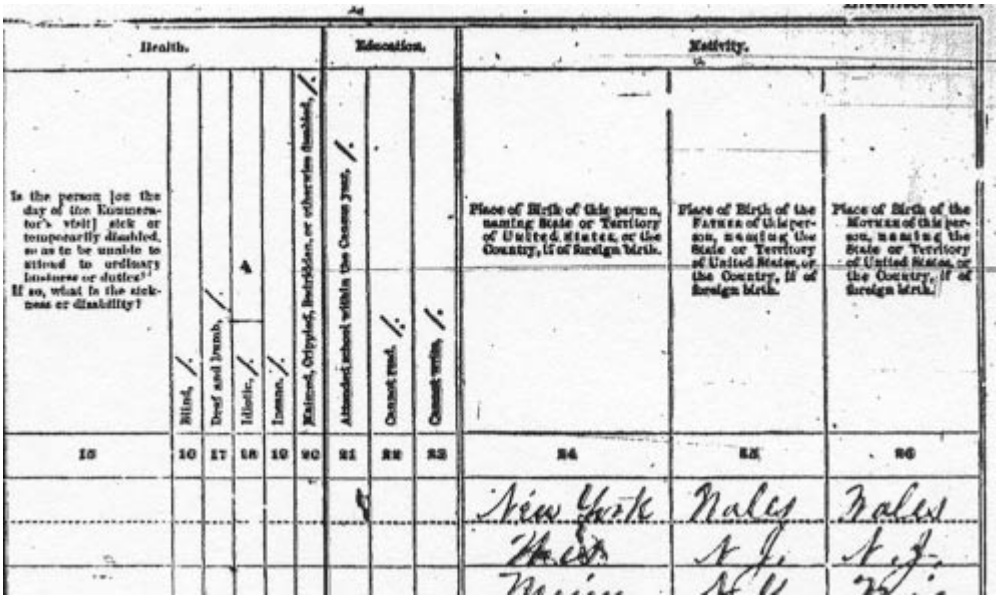


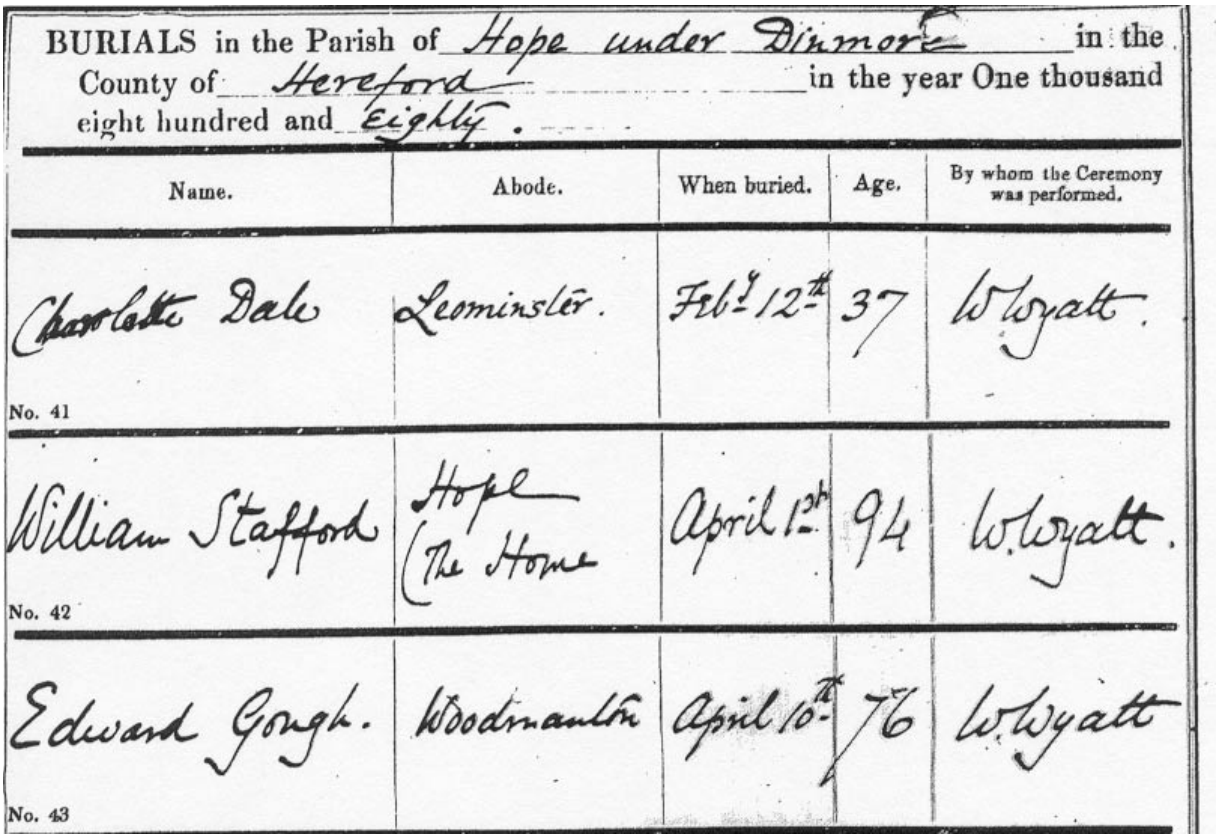

**Figure 4.3.0 A**

```
INSERT INTO PERSON (Person_Identifier, Full_Name, Age, Gender, Occupation,
        Race, Family_Identifier, Birth_Identifier) (1, '109,455,267,478',  '314,456
        ,336,479', '291,456,314,478', '505,457,637,480', '267,456,291,478', 1, 1)
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (2, 2)
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (3, 3)
INSERT INTO MOTHER_CHILD (Mother_Identifier, Child_Identifier) (3, 1)
INSERT INTO FATHER_CHILD (Father_Identifier, Child_Identifier) (2, 1)
INSERT INTO EVENT (Event_Identifier, Location) (1, '894,460,997,483')
INSERT INTO EVENT (Event_Identifier, Location) (2, '997,460,1076,483')
INSERT INTO EVENT (Event_Identifier, Location) (3, '1076,461,1153,484')
```

**Figure 4.3.0 B**

(2) The algorithm correctly discovered record boundaries. The first row in the table in Figure 4.3.1 A produced 2 tuples to the database table PERSON using the SQL statements in Figure 4.3.1 B. This occurred because the genealogical ontology states a

person can have only one “Full Name” at a time. Thus the label cell “By whom the Ceremony was performed” indicates a new person record.



BURIALS in the Parish of Hope under Dinmore in the County of Hereford in the year One thousand eight hundred and Eighty.

| Name. | Abode. | When buried. | Age. | By whom the Ceremony was performed. |
|---|---|---|---|---|
| Charlotte Dale<br>No. 41 | Leominster. | Feb.y 12th | 37 | W Wyatt. |
| William Stafford<br>No. 42 | Hope (The Home | April 1st | 94 | W. Wyatt. |
| Edward Gough.<br>No. 43 | Woodmanton | April 10th | 76 | W. Wyatt |

**Figure 4.3.1 A**

INSERT INTO PERSON (Person_Identifier, Full_Name, Age, Family_Identifier, Burial_Identifier) (1, '70,243,331,373', '620,243,687,370', 1, 1)
INSERT INTO FAMILY (Family_Identifier, Location) (1, '331,243,508,372')
INSERT INTO EVENT (Event_Identifier, Date) (1, '508,243,620,371')
INSERT INTO PERSON (Person_Identifier, Full_Name) (9, '687,241,861,372')

**Figure 4.3.1 B**

(3) The algorithm correctly identified label cells with factored values by averaging the confidences over multiple table instances of a microfilm roll. Figure 4.3.2

shows two different table instances from the same microfilm row. In both instances, the label cell “Families number in the order of visitation” maps to the object set “Family” and the label set “Name of every person whose usual place of abode on the first day of June was in this family” maps to the object set “Full Name”. In the table on the left, the number of value cells associated with “Full Name” divided by the number of value cells associated with “Family” is 3. In the second table, the number of value cells associated with “Full Name” divided by the number of value cells associated with “Family” is 6.5. Since the ontology states that the expected number of “Full Name” objects per “Family” is 4.8, “Family” will not factor “Full Name” in either of these tables (since the difference is greater than .6). Yet the average number of “Full Name” value cells per “Family” values cells observing both tables is 4.75. The algorithm correctly factored “Family” for each “Full Name” after taking the average.

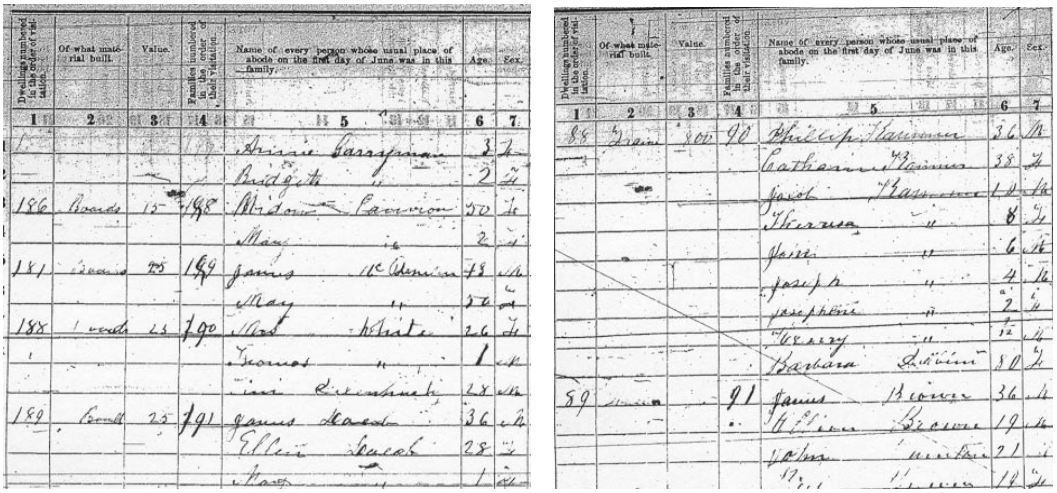


**Figure 4.3.2**

(4) The algorithm correctly identified related object sets when matching label cells to object sets. The algorithm assigns the following object-set confidence to the label cell shown in Figure 4.3.3 after initially populating the evidence matrix **LO** (before applying the correlation rules): “Birth” (0.05), “County” (1.0), “State” (0.5) and “Country” (0.09375). The object set “County” received a confidence of 1 because “county” appears near the front of the sentence, while the object set “Birth“ received a confidence of 0.05 because it appeared near the end of the sentence and match last. At this point, using a threshold of .5, the algorithm would associate the label cell with “County” and “State” only. After the algorithm ran the set of correlation rules, the object-set confidences for the label cell were as follows: “Birth” (1.0), “Location” (1.0). The algorithm mapped the object sets “County”, “State” and “Country” to “Location” because they are part of an aggregation where “Location” is the head. The algorithm raised the value of “Birth” to 1, because “Birth” is topologically close to “Location” in the ontology. Thus, the label cell in Figure 4.3.3 maps to both the object set “Birth” and the object set “Location”.

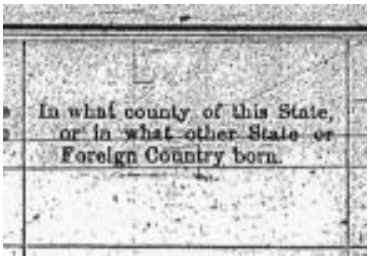



**Figure 4.3.3**

(5) The algorithm correctly identified multi-level groups of label cells. The algorithm assigns object set confidences in Figure 4.3.3 A to the label cells shown in Figure 4.3.3 B after initially populating the evidence matrix **LO** (before applying the correlation rules).

| Mapping Confidences | | | | |
|---|---|---|---|---|
| | Object Sets | | | |
| Label Cells | | "Age" | "Gender" | "Color" |
| | "Age" | 1 | 0 | 0 |
| | "Sex" | 0 | 1 | 0 |
| | "Color" | 0 | 0 | 1 |
| | "4" | 0 | 0 | 0 |
| | "5" | 0 | 0 | 0 |
| | "6" | 0 | 0 | 0 |

**Figure 4.3.3 A**

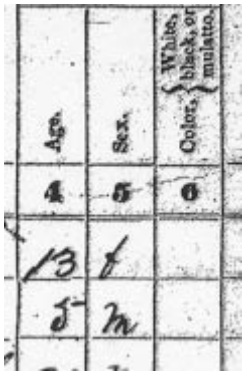



**Figure 4.3.4 B**

The cell labels "4", "5" and "6" have a confidence of zero because they did not map to any of the object sets in the ontology. Still, label cells "4", "5" and "6" are part of the composite label and associate with the value cell columns below them. The fourth correlation rule connects the multi-level label cells by distributing the object set

confidences. Figure 4.3.4 C shows the confidence values for the label cells after the algorithm ran the set of correlation rules. The label cell "4" is now associated with the object set "Age" with a value of one. The algorithm distributed the object set confidence from the label "Age" to the label "4" because they had a high confidence of being in a multi-level label in the evidence matrix **LL**.

| Mapping Confidences | | | | |
|---|---|---|---|---|
| | Object Sets | | | |
| Label Cells | | "Age" | "Gender" | "Color" |
| | "Age" | 1 | 0 | 0 |
| | "Sex" | 0 | 1 | 0 |
| | "Color" | 0 | 0 | 1 |
| | "4" | 1 | 0 | 0 |
| | "5" | 0 | 1 | 0 |
| | "6" | 0 | 0 | 1 |

**Figure 4.3.4 C**

(6) The algorithm correctly eliminated geometrically dissimilar value cells from groups of values cells associated with a label cell. In Figure 4.3.5 the label cell "Families number in the order of visitation" heads a column of value cells. All the value cells in the column have approximately the same width and height except for the last value cell, $v_{last}$, which is nearly three times as tall. After the algorithm initially populates the **LV** matrix, $v_{last}$ associates with high confidence with the label cell "Families number in the order of visitation" because it is the label cell's column. Yet after the algorithm initially populates the **VV** matrix**,** $v_{last}$ has low confidences of similarity with the other value cells in its column. Using the first correlation rule, the

algorithm lowers the confidence of $v_l$ with the label cell "Families number in the order of visitation" in **LV** until it is eliminated from the column.

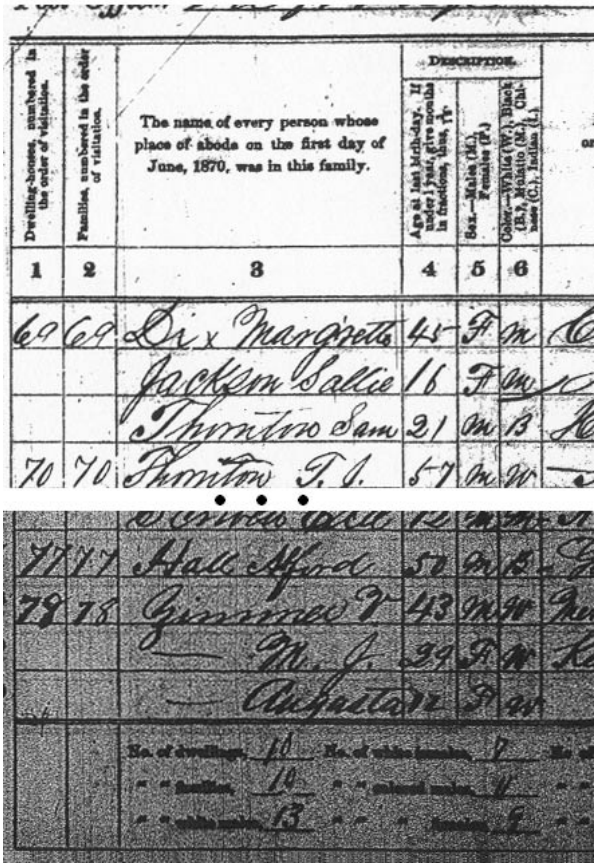



**Figure 4.3.5**

(7). The algorithm correctly identified value cells that are in the same rows and columns. In Figure 4.3.6, a horizontal line crosses the table entering at the bottom edge of the row of numbered label cells and leaving the table at the top edge of the row of number label cells. Our algorithm correctly identifies this row because it checks locally. Instead of attempting to determine directly if a cell on the left side of the table is in the

same row as a cell on the right side of the table, the algorithm determines only whether the closest cell is in the same row.

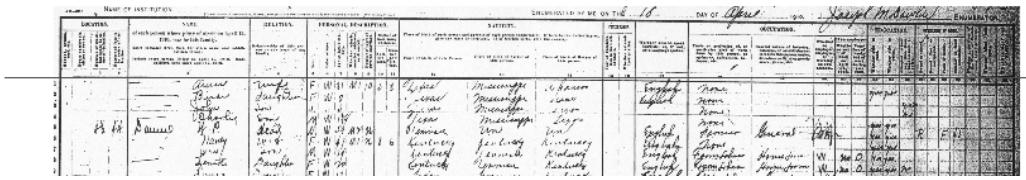
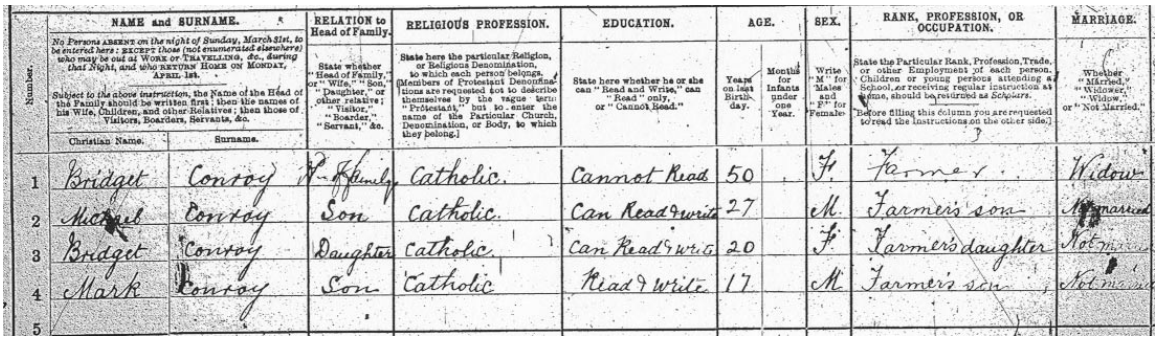


**Figure 4.3.6**

The algorithm incorrectly identified three different types of structure and content information. We calculated the number of tables that exhibited each of the error types. We present and discuss each of the three types of error and provide a possible solution for each.

(1) The algorithm incorrectly identified ambiguous factorings in 3 out of the 75 tables in the test set. We identified and discussed this error with respect to the training set. Ambiguous factoring occurs when a value cell that factors a record is missing. If we could use optical character recognition to read the hand-written values, we could overcome this obstacle using the content of the value cells to identify record boundaries; see Section 4.2. In addition, we can infer a record boundary from the geometric pattern of value cells whose label cell does not map to the genealogical ontology. For example, in Figure 4.2.1 the label cell "No. of Schedule" does not match an object set in the genealogical ontology. Yet, by observing the geometric pattern of its values cells, we can determine that its values also identify a family record boundary.

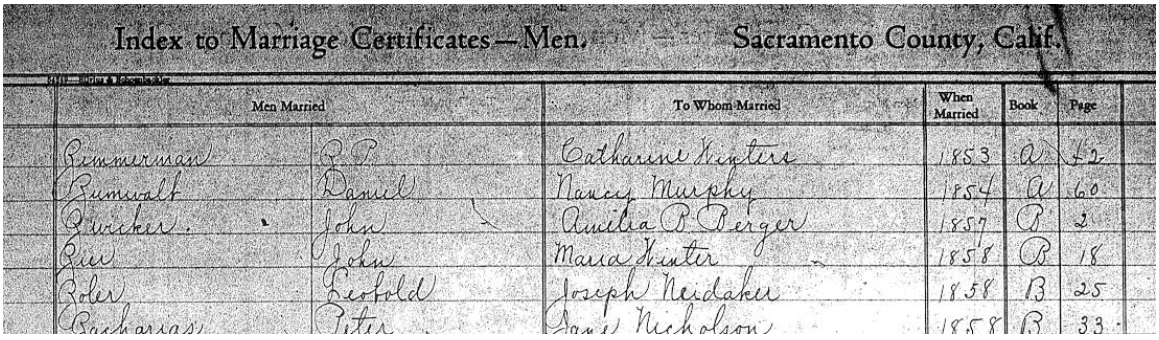
(2) The algorithm incorrectly matched object sets to label cells with long character strings of machine-printed text. Of the 75 test tables, 40 exhibited this problem to some degree. Since the label cell text is constant over a roll of microfilm, if the algorithm generated this type of error on one table, it generated the same error for all tables on the same roll. Figure 4.3.7 shows a microfilm table with long label cell names. For example the algorithm incorrectly mapped the label cell with the machine-printed text “State here the particular Religion, or Religious Denomination, which each person belongs. [Members of Protestant Denominations are requested not to describe themselves by the vague term 'Protestant,' but to enter the name of the Particular Church, Denomination, or Body, to which they belong.]”. Since the label cell’s text begins with the word “State”, the algorithm mapped the label cell to the object set “State”, meaning the location “State” shown by Figure 2.21. In addition, this label cell is part of a composite label cell. Since the other label cell “RELIGIOUS PROFESSION” in the composite label incorrectly maps to the object set “Occupation”, the composite label incorrectly maps to both the objects “State” and “Occupation”. Natural language processing can be used to overcome this problem by detecting the part-of-speech and meaning of these words. Since the word “State” in the label cell is a verb, it is possible to determine that the algorithm should not match it to the noun “State” that is a location. Also, the word “PROFESSION” in the label cell “RELIGIOUS PROFESSION” is a noun meaning both an occupation and a professed belief. It’s adjective, “RELIGIOUS”, is needed to disambiguate the cases.



| Number. | NAME and SURNAME. Christian Name. | Surname. | RELATION to Head of Family. | RELIGIOUS PROFESSION. | EDUCATION. | AGE. Years on last Birth-day. | Months for Infants under one Year. | SEX. | RANK, PROFESSION, OR OCCUPATION. | MARRIAGE. |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Bridget | Conroy | Head of Family | Catholic. | Cannot Read | 50 | | F. | Farmer | Widow |
| 2 | Michael | Conroy | Son | Catholic. | Can Read & write | 27 | | M. | Farmer's son | Not married |
| 3 | Bridget | Conroy | Daughter | Catholic. | Can Read & write | 20 | | F. | Farmer's daughter | Not married |
| 4 | Mark | Conroy | Son | Catholic | Read & Write | 17 | | M | Farmer's son | Not married |
| 5 | | | | | | | | | | |

**Figure 4.3.7**

(3) The algorithm incorrectly associated label cells to ambiguous columns of value cells. Of the 75 tables in the test set, 10 exhibited this problem. Figure 4.3.8 shows an example of a label ambiguously describing two columns of value cells. In this table, the algorithm maps the label cell "Men Married" to the object set "Full Name". Although the algorithm knows that both columns are associated with the object set "Full Name", it cannot determine that the first column of value cells actually maps to the object set "Last Name" and the second column of value cells actually maps to the object set "First Name" because it does not have access to the hand-written text. If we can have access to the hand-written values, we can overcome this problem by using the content of each value cell to determine its data type. Otherwise, the algorithm could only guess which way to concatenate the two values to form a "Full Name" value.



Index to Marriage Certificates—Men. Sacramento County, Calif.

| Men Married | | To Whom Married | When Married | Book | Page |
|---|---|---|---|---|---|
| Zimmerman | P. P. | Catharine Kyters | 1853 | A | 42 |
| Zumwalt | Daniel | Nancy Murphy | 1854 | A | 60 |
| Zwicker | John | Amelia P. Berger | 1857 | B | 2 |
| Zur | John | Maria Hinter | 1858 | B | 18 |
| Zoler | Leobold | Joseph Neidaker | 1858 | B | 25 |
| Zacharias | Peter | Daniel Nicholson | 1858 | B | 33 |

**Figure 4.3.8**

# CHAPTER 5 – CONCLUSION AND FUTURE WORK

## 5.1 Conclusion

Our algorithm achieved a precision of 93.20 percent, a recall of 92.41 percent, and an accuracy of 92.15 percent on the database fields populated from the microfilm tables in our test set. It performed with high accuracy because it exploited both the geometric and ontological properties of tables. In addition, we demonstrated that a set of evidence matrices and correlation rules could be developed to identify the geometric and ontological relationships within a table.

Our research made two contributions to the field of computer science. (1) It successfully added to the current techniques of table recognition by exploiting both constraints of an ontology and tabular geometry. (2) It presented an original technique for combining extracted features using correlation rules.

In addition, our research produced four artifacts. (1) We created a tool that implements the algorithm described in this thesis. The tool allows users to add and remove features to the set of extracted features for each evidence matrix and rules to the set of correlation rules. The tool enables users to correct the cells described in the XML input file for a microfilm table. The tool enables users to verify visually the records extracted from a table. (2) We created a genealogical ontology that models the relationships and constraints between genealogical objects. (3) We discovered and analyzed a set of features and correlation rules. (4) We gathered and zoned a corpus of microfilm tables from original microfilm images and made these available for use by

others. The web site www.rdhd.byu.edu contains a downloadable version of the tool and the corpus of microfilm images.

## 5.2 Future Work

We recommend the following three areas for future work. (1) Researchers can extend this work by investigating the use of natural language processing to map the label cells in the table to the object sets in the genealogical ontology. (2) Researches can conduct investigations using the extracted character strings from the hand-written value cells. Partial recognition of static hand-writing is already possible [Jag97]. This will allow the algorithm to confirm the type of data within value cells and identify the type of data contained in columns without labels or with ambiguous label-cell text. (3) With enough input data, researcher can investigate weighting the value of the feature extracted to initially populate the evidence matrices. In addition, researchers can investigate the use of machine learning techniques to learn the proper sets of correlation rules.

REFERENCES


[AAM+01] A. Amano, N. Asada, T. Montoymam, T. Sumiyoshi, and K. Suzuki. Table Form Document Synthesis by Grammar-Based Structure Analysis. In *Proceedings of the Sixth International Conference on Document Analysis and Recognition*, pages 533-537, Seattle, Washington, 10–13 September 2001.

[BBI98] R. Brugger, F. Bapst, and R. Ingold. A DTD Extension for Document Structure Recognition. Technical Report, University of Fribourg, 1998.

[BEB01] F. Bourgeois, H. Emptoz, and S. Bensafi. Document Understanding Using Probabilistic Relaxation: Application on Tables of Contents of Periodicals. In *Proceedings of the Sixth International Conference on Document Analysis and Recognitio*n, pages 508-512, Seattle, Washington, 10–13 September 2001.

[BGS+98] M. Boliek, M. Gormish, E. L. Schwartz, and A. F. Keith. Decoding Compression with Reversible Embedded Wavelets (CREW) Codestreams. *Journal of Electronic Imaging*, 7(3):402-209, 1998.

[BR99] R. Baeza-Yates and Ribeiro-Neto B. Modern Information Retrieval. *Addison-Wesley*, 1999.

[BZI97] R. Brugger, A. Zramdini, and R. Ingold. Modeling Documents for Structure Recognition using Generalized N-Grams. In *Proceedings of the International Conference on Document Analysis and Recognition*, 1:56–60, Ulm, Germany, 18-20 August 1997.

[CK97] N. Chiba and T. Kanade. A Tracker for Broken and Closely-Spaced Lines. Technical Report, Carnegie Mellon University, 1997.

[Emb98] D.W. Embley, Object Database Development: Concepts and Principles, *Addison-Wesley*, Reading, Massachusetts, 1998.

[Fam01] FamilySearch.org. Facts and Statistics. http://www.familysearch.org, November 2001.

[GK95] E. Green and M. Krishnamoorthy. Model-Based Analysis of Printed Tables. In *Proceedings of the Third International Conference on Document Analysis and Recognition*, 2:80-91, Montreal, Canada, 14-16 August 1995.

[Has98] Terrence B. Hass. The Development of a Prototype Knowledge-Based Table-Processing System. Masters Thesis, Department of Computer Science, Brigham Young University, 1998.

[HD97] M. Hurst and S. Douglas. *Layout* and *Language:* Preliminary Experiments in Assigning Logical Structure to Table Cells. In *Proceedings of Fifth Applied Natural Language Processing Conference*, pages 217-220, Washington, DC, 31 March – 3 April 1997.

[HKL+01] Jianying Hu; R. Kashi, D. Lopresti, G. Nagy and G. Wilfong. Why Table Ground-Truthing is Hard. In *Proceedings of the Sixth International Conference on Document Analysis and Recognition*, pages 129–133, Seattle, Washington, 10–13 September 2001.

[Hur01] M. Hurst. Layout and Language: Exploring Text Block Discovery in Tables using Linguistic Resources. In *Proceedings of the Sixth International Conference on Document Analysis and Recognition*, pages 523–527, Seattle, Washington, 10–13 September 2001.

[Jag97] S. Jager. Recovering Dynamic Information from Static, Handwritten Word Images. Doctoral Dissertation, University of Freiburg, 1997.

[JAVA] The JAVA home page: http://java.sun.com/

[JE99] A. Jobbins and L. Evett. Segmenting Documents using Multiple Lexical Features. In *Proceedings of the Fifth International Conference on Document Analysis and Recognition*, pages 721–724, Bangalore, India, 20-22 September 1999.

[JHD99] M. Junker, R. Hoch, and A. Dengel. On the Evaluation of Document Analysis Components by Recall, Precision, and Accuracy. In *Proceedings of the Fifth International Conference on Document Analysis and Recognition*, pages 713–716, Bangalore, India, 20-22 September 1999.

[KB01] D. Kennard and W. Barrett. Just-In-Time Browsing for Digital Microfilm. Presented at the *Workshop on Technology for Family History and Genealogical Research*, Provo, Utah, 29 April 2001.

[KD01] T . Kieninger and A. Dengel. Applying the T-Recs Table Recognition System to the Business Letter Domain. In *Proceedings of the Sixth International Conference on Document Analysis and Recognition*, pages 518–522, Seattle, Washington, 10–13 September 2001.

[KS99] T. Kochi and T. Saitoh. User-defined Template for Identifying Document Type and Extracting Information from Documents. In *Proceedings of the Fifth International Conference on Document Analysis and Recognition*, pages 127-130, Bangalore, India, 20-22 September 1999.

[LCC99] S.W. Liddle, D.M. Campbell, and C. Crawford. Automatically Extracting Structure and Data from Business Reports. In *Proceedings of the Eighth International Conference on Information Knowledge Management*, Kansas City, Missouri, 2-6 November 1999.

[LLN+96] Y. Li, D. Lopresti, G. Nagy, and A. Tomkins. Validation of Image Defect Models for Optical Character Recognition. In *Proceedings of IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(2):99-108, February 1996.

[LN00] D. Lopresti and G. Nagy. A Tabular Survey of Automated Table Processing. Technical Report, Bell Labs, Lucent Technologies Inc., 2000.

[Nag00] G. Nagy. Twenty Years of Document Image Analysis in PAMI. In *Proceedings of IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):38-62, January 2000.

[NB01] H. Nielson and W. Barrett. Automatic Zoning of Digitized Documents. Presented at the *Workshop on Technology for Family History and Genealogical Research*, Provo, Utah, 29 April 2001.

[Rim01] M. Rimer. Brigham Young University Neural Networks and Machine Learning Laboratory, Provo, Utah, November 2001.

[SBW97] J. Shamilian, H. Baird and T. Wood. A Retargetable Table Reader. In *Proceedings of the Forth International Conference on Document Analysis and Recognition*, 1:158-163, Ulm, Germany, 18-20 August 1997.

[WMR97] V. Wu, R. Manmatha and E. Riseman. Finding Text in Images. In *Proceedings of the International Conference on Digital Libraries*, pages 3-12, Philadelphia, Pennsylvania, 23-26 July 1997.

[WPH01] Y. Wang, I. Phillips, and R. Haralick. Automatic Table Ground Truth Generation and a Background-Analysis-Based Table Structure Extraction Method. In *Proceedings of the Sixth International Conference on Document Analysis and Recognition*, pages 528–532, Seattle, Washington, 10–13 September 2001.

[Xml] The W3C Architecture Domain. http://www.w3.org/XML/

# APPENDIX 1 – INPUT XML FILE

```
<table source="tables/1405269/1405269_1.gif" ontology="ontology.xml">
    <zone rect="5,84,22,94" printed_text="In Cities" empty="0"/>
    <zone rect="5,94,13,162" printed_text="Name of Street" empty="0"/>
    <zone rect="13,94,22,162" printed_text="House Number" empty="0"/>
    <zone rect="23,84,33,162" printed_text="Dwelling houses numbered in order of visitation" empty="0"/>
    <zone rect="32,84,43,162" printed_text="Famalies numbered in order of visitation" empty="0"/>
    <zone rect="43,84,102,162" printed_text="The name of each Person whose place of abode, on 1st day of
        June, 1885, was in this family." empty="0"/>
    <zone rect="102,84,127,94" printed_text="Personal Description." empty="0"/>
    <zone rect="102,93,111,162" printed_text="Color - White, W; Black, B; Malato, M; Chinese, C; Indian, I."
        empty="0"/>
    <zone rect="111,94,119,162" printed_text="Sex - Male, M; Female, F" empty="0"/>
    <zone rect="118,93,127,162" printed_text="Age at last birthday prior to June 1 1884. If under 1 year, give
        months in fractions" empty="0"/>
    <zone rect="127,84,138,162" printed_text="If born within the Census year, give the month" empty="0"/>
    <zone rect="139,84,168,162" printed_text="Relationship of each person to the head of the family - whether
        wife, son, daughter, servant, boarder, or other." empty="0"/>
    <zone rect="167,85,184,94" printed_text="Civil Condition" empty="0"/>
    <zone rect="168,94,173,162" printed_text="Single" empty="0"/>
    <zone rect="173,94,179,162" printed_text="Married" empty="0"/>
    <zone rect="179,94,184,162" printed_text="Widowed" empty="0"/>
    <zone rect="184,84,191,162" printed_text="Married during Census year." empty="0"/>
    <zone rect="191,84,251,94" printed_text="Occupation." empty="0"/>
    <zone rect="191,94,240,162" printed_text="Profession, Occupation, trade of each person male or female"
        empty="0"/>
    <zone rect="240,94,251,162" printed_text="Number of months this person has been unemployed during the
        census year." empty="0"/>
    <zone rect="252,84,312,94" printed_text="Health" empty="0"/>
    <zone rect="251,95,282,162" printed_text="Is the person [on the day of the Enumerator's visit] sick or
        temporarily disabled. so as to be usable to attend to ordinary business or duties. If so, what is the sikness
        or disability." empty="0"/>
    <zone rect="281,94,287,162" printed_text="Blind" empty="0"/>
    <zone rect="287,95,293,162" printed_text="Deaf and Dumb" empty="0"/>
    <zone rect="294,95,299,162" printed_text="Idiotic" empty="0"/>
    <zone rect="299,94,305,162" printed_text="Insane" empty="0"/>
    <zone rect="305,94,312,162" printed_text="Kained, Crippled, Bedridden, or otherwise disabled" empty="0"/>
    <zone rect="312,84,336,94" printed_text="Education" empty="0"/>
    <zone rect="312,93,319,162" printed_text="Attended school within the Census year" empty="0"/>
    <zone rect="319,94,328,162" printed_text="Cannot read" empty="0"/>
    <zone rect="329,93,336,162" printed_text="Cannot write." empty="0"/>
    <zone rect="336,84,434,95" printed_text="Nativity" empty="0"/>
    <zone rect="336,94,376,162" printed_text="Place of Birth of the person naming State or Territory of United
        States or the Country, if of foreign birth." empty="0"/>
    <zone rect="375,94,405,162" printed_text="Place of Birth of the Father of this person naming the State or
        Territory of United States or the Country, if of foreign birth." empty="0"/>
    <zone rect="405,95,433,162" printed_text="Place of birth of the Mother of the person, naming the State or
        Territory of United States, or the Country, if of foreign birth." empty="0"/>

    <zone rect="5,161,13,169" printed_text="" empty="1"/>
    <zone rect="5,169,13,178" printed_text="" empty="1"/>
    <zone rect="5,178,13,186" printed_text="" empty="1"/>
    <zone rect="5,186,13,194" printed_text="" empty="1"/>
    <zone rect="5,194,13,203" printed_text="" empty="1"/>
    <zone rect="5,203,13,212" printed_text="" empty="1"/>
    <zone rect="5,212,13,221" printed_text="" empty="1"/>
    <zone rect="5,221,13,229" printed_text="" empty="1"/>
    <zone rect="5,229,13,237" printed_text="" empty="1"/>
    <zone rect="5,237,13,245" printed_text="" empty="1"/>
    <zone rect="5,245,13,254" printed_text="" empty="1"/>
    <zone rect="5,254,13,261" printed_text="" empty="1"/>
    <zone rect="5,261,13,269" printed_text="" empty="1"/>
    <zone rect="5,269,13,278" printed_text="" empty="1"/>
    <zone rect="5,278,13,286" printed_text="" empty="1"/>
    <zone rect="5,286,13,293" printed_text="" empty="1"/>
    <zone rect="5,293,13,302" printed_text="" empty="1"/>
    <zone rect="5,302,13,310" printed_text="" empty="1"/>
    <zone rect="5,310,13,317" printed_text="" empty="1"/>
    <zone rect="5,317,13,326" printed_text="" empty="1"/>
    <zone rect="5,326,13,334" printed_text="" empty="1"/>
    <zone rect="5,334,13,344" printed_text="" empty="1"/>
    <zone rect="5,344,13,351" printed_text="" empty="1"/>
    <zone rect="5,351,13,361" printed_text="" empty="1"/>
    <zone rect="5,361,13,368" printed_text="" empty="1"/>
    <zone rect="5,368,13,376" printed_text="" empty="1"/>
    <zone rect="5,376,13,384" printed_text="" empty="1"/>
    <zone rect="5,384,13,392" printed_text="" empty="1"/>
    <zone rect="5,392,13,400" printed_text="" empty="1"/>
    <zone rect="5,400,13,409" printed_text="" empty="1"/>
    <zone rect="5,409,13,416" printed_text="" empty="1"/>
    <zone rect="5,416,13,425" printed_text="" empty="1"/>
```

<zone rect="5,425,13,433" printed_text="" empty="1"/>
<zone rect="5,433,13,441" printed_text="" empty="1"/>
<zone rect="5,441,13,449" printed_text="" empty="1"/>
<zone rect="5,449,13,456" printed_text="" empty="1"/>
<zone rect="5,456,13,464" printed_text="" empty="1"/>
<zone rect="5,464,13,474" printed_text="" empty="1"/>
<zone rect="5,474,13,481" printed_text="" empty="1"/>
<zone rect="5,481,13,490" printed_text="" empty="1"/>
<zone rect="5,490,13,497" printed_text="" empty="1"/>
<zone rect="5,497,13,506" printed_text="" empty="1"/>
<zone rect="5,506,13,514" printed_text="" empty="1"/>
<zone rect="5,514,13,522" printed_text="" empty="1"/>
<zone rect="5,522,13,530" printed_text="" empty="1"/>
<zone rect="5,530,13,538" printed_text="" empty="1"/>
<zone rect="5,538,13,545" printed_text="" empty="1"/>
<zone rect="5,545,13,554" printed_text="" empty="1"/>
<zone rect="5,554,13,561" printed_text="" empty="1"/>
<zone rect="5,561,13,569" printed_text="" empty="1"/>
<zone rect="5,569,13,579" printed_text="" empty="1"/>
<zone rect="13,161,23,169" printed_text="" empty="1"/>
<zone rect="13,169,23,178" printed_text="" empty="1"/>
<zone rect="13,178,23,186" printed_text="" empty="1"/>
<zone rect="13,186,23,194" printed_text="" empty="1"/>
<zone rect="13,194,23,203" printed_text="" empty="1"/>
<zone rect="13,203,23,212" printed_text="" empty="1"/>
<zone rect="13,212,23,221" printed_text="" empty="1"/>
<zone rect="13,221,23,229" printed_text="" empty="1"/>
<zone rect="13,229,23,237" printed_text="" empty="1"/>
<zone rect="13,237,23,245" printed_text="" empty="1"/>
<zone rect="13,245,23,254" printed_text="" empty="1"/>
<zone rect="13,254,23,261" printed_text="" empty="1"/>
<zone rect="13,261,23,269" printed_text="" empty="1"/>
<zone rect="13,269,23,278" printed_text="" empty="1"/>
<zone rect="13,278,23,286" printed_text="" empty="1"/>
<zone rect="13,286,23,293" printed_text="" empty="1"/>
<zone rect="13,293,23,302" printed_text="" empty="1"/>
<zone rect="13,302,23,310" printed_text="" empty="1"/>
<zone rect="13,310,23,317" printed_text="" empty="1"/>
<zone rect="13,317,23,326" printed_text="" empty="1"/>
<zone rect="13,326,23,334" printed_text="" empty="1"/>
<zone rect="13,334,23,344" printed_text="" empty="1"/>
<zone rect="13,344,23,351" printed_text="" empty="1"/>
<zone rect="13,351,23,361" printed_text="" empty="1"/>
<zone rect="13,361,23,368" printed_text="" empty="1"/>
<zone rect="13,368,23,376" printed_text="" empty="1"/>
<zone rect="13,376,23,384" printed_text="" empty="1"/>
<zone rect="13,384,23,392" printed_text="" empty="1"/>
<zone rect="13,392,23,400" printed_text="" empty="1"/>
<zone rect="13,400,23,409" printed_text="" empty="1"/>
<zone rect="13,409,23,416" printed_text="" empty="1"/>
<zone rect="13,416,23,425" printed_text="" empty="1"/>
<zone rect="13,425,23,433" printed_text="" empty="1"/>
<zone rect="13,433,23,441" printed_text="" empty="1"/>
<zone rect="13,441,23,449" printed_text="" empty="1"/>
<zone rect="13,449,23,456" printed_text="" empty="1"/>
<zone rect="13,456,23,464" printed_text="" empty="1"/>
<zone rect="13,464,23,474" printed_text="" empty="1"/>
<zone rect="13,474,23,481" printed_text="" empty="1"/>
<zone rect="13,481,23,490" printed_text="" empty="1"/>
<zone rect="13,490,23,497" printed_text="" empty="1"/>
<zone rect="13,497,23,506" printed_text="" empty="1"/>
<zone rect="13,506,23,514" printed_text="" empty="1"/>
<zone rect="13,514,23,522" printed_text="" empty="1"/>
<zone rect="13,522,23,530" printed_text="" empty="1"/>
<zone rect="13,530,23,538" printed_text="" empty="1"/>
<zone rect="13,538,23,545" printed_text="" empty="1"/>
<zone rect="13,545,23,554" printed_text="" empty="1"/>
<zone rect="13,554,23,561" printed_text="" empty="1"/>
<zone rect="13,561,23,569" printed_text="" empty="1"/>
<zone rect="13,569,23,579" printed_text="" empty="1"/>
<zone rect="23,161,32,169" printed_text="1"empty="0"/>
<zone rect="23,169,32,178" printed_text="" empty="1"/>
<zone rect="23,178,32,186" printed_text="" empty="1"/>
<zone rect="23,186,32,194" printed_text="" empty="1"/>
<zone rect="23,194,32,203" printed_text="" empty="1"/>
<zone rect="23,203,32,212" printed_text="" empty="1"/>
<zone rect="23,212,32,221" printed_text="" empty="1"/>
<zone rect="23,221,32,229" printed_text="" empty="1"/>
<zone rect="23,229,32,237" printed_text="" empty="1"/>
<zone rect="23,237,32,245" printed_text="" empty="1"/>
<zone rect="23,245,32,254" printed_text="" empty="1"/>
<zone rect="23,254,32,261" printed_text="" empty="1"/>
<zone rect="23,261,32,269" printed_text="" empty="1"/>
<zone rect="23,269,32,278" printed_text="" empty="1"/>
<zone rect="23,278,32,286" printed_text="" empty="1"/>
<zone rect="23,286,32,293" printed_text="" empty="1"/>
<zone rect="23,293,32,302" printed_text="" empty="1"/>
<zone rect="23,302,32,310" printed_text="" empty="1"/>
<zone rect="23,310,32,317" printed_text="" empty="1"/>
<zone rect="23,317,32,326" printed_text="" empty="1"/>
<zone rect="23,326,32,334" printed_text="" empty="1"/>
<zone rect="23,334,32,344" printed_text="" empty="1"/>
<zone rect="23,344,32,351" printed_text="" empty="1"/>
<zone rect="23,351,32,361" printed_text="" empty="1"/>
<zone rect="23,361,32,368" printed_text="" empty="1"/>
<zone rect="23,368,32,376" printed_text="" empty="1"/>
<zone rect="23,376,32,384" printed_text="" empty="1"/>
<zone rect="23,384,32,392" printed_text="" empty="1"/>
<zone rect="23,392,32,400" printed_text="" empty="1"/>
<zone rect="23,400,32,409" printed_text="" empty="1"/>
<zone rect="23,409,32,416" printed_text="" empty="1"/>
<zone rect="23,416,32,425" printed_text="" empty="1"/>
<zone rect="23,425,32,433" printed_text="" empty="1"/>
<zone rect="23,433,32,441" printed_text="" empty="1"/>
<zone rect="23,441,32,449" printed_text="" empty="1"/>
<zone rect="23,449,32,456" printed_text="" empty="1"/>
<zone rect="23,456,32,464" printed_text="" empty="1"/>
<zone rect="23,464,32,474" printed_text="" empty="1"/>
<zone rect="23,474,32,481" printed_text="" empty="1"/>
<zone rect="23,481,32,490" printed_text="" empty="1"/>
<zone rect="23,490,32,497" printed_text="" empty="1"/>
<zone rect="23,497,32,506" printed_text="" empty="1"/>
<zone rect="23,506,32,514" printed_text="" empty="1"/>
<zone rect="23,514,32,522" printed_text="" empty="1"/>
<zone rect="23,522,32,530" printed_text="" empty="1"/>
<zone rect="23,530,32,538" printed_text="" empty="1"/>
<zone rect="23,538,32,545" printed_text="" empty="1"/>
<zone rect="23,545,32,554" printed_text="" empty="1"/>
<zone rect="23,554,32,561" printed_text="" empty="1"/>
<zone rect="23,561,32,569" printed_text="" empty="1"/>
<zone rect="23,569,32,579" printed_text="" empty="1"/>
<zone rect="32,161,43,169" printed_text="2"empty="0"/>
<zone rect="32,169,43,178" printed_text="" empty="1"/>
<zone rect="32,178,43,186" printed_text="" empty="1"/>
<zone rect="32,186,43,194" printed_text="" empty="1"/>
<zone rect="32,194,43,203" printed_text="" empty="1"/>
<zone rect="32,203,43,212" printed_text="" empty="1"/>
<zone rect="32,212,43,221" printed_text="" empty="1"/>
<zone rect="32,221,43,229" printed_text="" empty="1"/>
<zone rect="32,229,43,237" printed_text="" empty="1"/>
<zone rect="32,237,43,245" printed_text="" empty="1"/>
<zone rect="32,245,43,254" printed_text="" empty="1"/>
<zone rect="32,254,43,261" printed_text="" empty="1"/>
<zone rect="32,261,43,269" printed_text="" empty="1"/>
<zone rect="32,269,43,278" printed_text="" empty="1"/>
<zone rect="32,278,43,286" printed_text="" empty="1"/>
<zone rect="32,286,43,293" printed_text="" empty="1"/>
<zone rect="32,293,43,302" printed_text="" empty="1"/>
<zone rect="32,302,43,310" printed_text="" empty="1"/>
<zone rect="32,310,43,317" printed_text="" empty="1"/>

<zone rect="32,317,43,326" printed_text="" empty="1"/>
<zone rect="32,326,43,334" printed_text="" empty="1"/>
<zone rect="32,334,43,344" printed_text="" empty="1"/>
<zone rect="32,344,43,351" printed_text="" empty="1"/>
<zone rect="32,351,43,361" printed_text="" empty="1"/>
<zone rect="32,361,43,368" printed_text="" empty="1"/>
<zone rect="32,368,43,376" printed_text="" empty="1"/>
<zone rect="32,376,43,384" printed_text="" empty="1"/>
<zone rect="32,384,43,392" printed_text="" empty="1"/>
<zone rect="32,392,43,400" printed_text="" empty="1"/>
<zone rect="32,400,43,409" printed_text="" empty="1"/>
<zone rect="32,409,43,416" printed_text="" empty="1"/>
<zone rect="32,416,43,425" printed_text="" empty="1"/>
<zone rect="32,425,43,433" printed_text="" empty="1"/>
<zone rect="32,433,43,441" printed_text="" empty="1"/>
<zone rect="32,441,43,449" printed_text="" empty="1"/>
<zone rect="32,449,43,456" printed_text="" empty="1"/>
<zone rect="32,456,43,464" printed_text="" empty="1"/>
<zone rect="32,464,43,474" printed_text="" empty="1"/>
<zone rect="32,474,43,481" printed_text="" empty="1"/>
<zone rect="32,481,43,490" printed_text="" empty="1"/>
<zone rect="32,490,43,497" printed_text="" empty="1"/>
<zone rect="32,497,43,506" printed_text="" empty="1"/>
<zone rect="32,506,43,514" printed_text="" empty="1"/>
<zone rect="32,514,43,522" printed_text="" empty="1"/>
<zone rect="32,522,43,530" printed_text="" empty="1"/>
<zone rect="32,530,43,538" printed_text="" empty="1"/>
<zone rect="32,538,43,545" printed_text="" empty="1"/>
<zone rect="32,545,43,554" printed_text="" empty="1"/>
<zone rect="32,554,43,561" printed_text="" empty="1"/>
<zone rect="32,561,43,569" printed_text="" empty="1"/>
<zone rect="32,569,43,579" printed_text="" empty="1"/>
<zone rect="43,161,102,169" printed_text="3" empty="0"/>
<zone rect="43,169,102,178" printed_text="" empty="1"/>
<zone rect="43,178,102,186" printed_text="" empty="1"/>
<zone rect="43,186,102,194" printed_text="" empty="1"/>
<zone rect="43,194,102,203" printed_text="" empty="1"/>
<zone rect="43,203,102,212" printed_text="" empty="1"/>
<zone rect="43,212,102,221" printed_text="" empty="1"/>
<zone rect="43,221,102,229" printed_text="" empty="1"/>
<zone rect="43,229,102,237" printed_text="" empty="1"/>
<zone rect="43,237,102,245" printed_text="" empty="1"/>
<zone rect="43,245,102,254" printed_text="" empty="1"/>
<zone rect="43,254,102,261" printed_text="" empty="1"/>
<zone rect="43,261,102,269" printed_text="" empty="1"/>
<zone rect="43,269,102,278" printed_text="" empty="1"/>
<zone rect="43,278,102,286" printed_text="" empty="1"/>
<zone rect="43,286,102,293" printed_text="" empty="1"/>
<zone rect="43,293,102,302" printed_text="" empty="1"/>
<zone rect="43,302,102,310" printed_text="" empty="1"/>
<zone rect="43,310,102,317" printed_text="" empty="1"/>
<zone rect="43,317,102,326" printed_text="" empty="1"/>
<zone rect="43,326,102,334" printed_text="" empty="1"/>
<zone rect="43,334,102,344" printed_text="" empty="1"/>
<zone rect="43,344,102,351" printed_text="" empty="1"/>
<zone rect="43,351,102,361" printed_text="" empty="1"/>
<zone rect="43,361,102,368" printed_text="" empty="1"/>
<zone rect="43,368,102,376" printed_text="" empty="1"/>
<zone rect="43,376,102,384" printed_text="" empty="1"/>
<zone rect="43,384,102,392" printed_text="" empty="1"/>
<zone rect="43,392,102,400" printed_text="" empty="1"/>
<zone rect="43,400,102,409" printed_text="" empty="1"/>
<zone rect="43,409,102,416" printed_text="" empty="1"/>
<zone rect="43,416,102,425" printed_text="" empty="1"/>
<zone rect="43,425,102,433" printed_text="" empty="1"/>
<zone rect="43,433,102,441" printed_text="" empty="1"/>
<zone rect="43,441,102,449" printed_text="" empty="1"/>
<zone rect="43,449,102,456" printed_text="" empty="1"/>
<zone rect="43,456,102,464" printed_text="" empty="1"/>
<zone rect="43,464,102,474" printed_text="" empty="1"/>
<zone rect="43,474,102,481" printed_text="" empty="1"/>
<zone rect="43,481,102,490" printed_text="" empty="1"/>
<zone rect="43,490,102,497" printed_text="" empty="1"/>
<zone rect="43,497,102,506" printed_text="" empty="1"/>
<zone rect="43,506,102,514" printed_text="" empty="1"/>
<zone rect="43,514,102,522" printed_text="" empty="1"/>
<zone rect="43,522,102,530" printed_text="" empty="1"/>
<zone rect="43,530,102,538" printed_text="" empty="1"/>
<zone rect="43,538,102,545" printed_text="" empty="1"/>
<zone rect="43,545,102,554" printed_text="" empty="1"/>
<zone rect="43,554,102,561" printed_text="" empty="1"/>
<zone rect="43,561,102,569" printed_text="" empty="1"/>
<zone rect="43,569,102,579" printed_text="" empty="1"/>
<zone rect="102,161,111,169" printed_text="4" empty="0"/>
<zone rect="102,169,111,178" printed_text="" empty="1"/>
<zone rect="102,178,111,186" printed_text="" empty="1"/>
<zone rect="102,186,111,194" printed_text="" empty="1"/>
<zone rect="102,194,111,203" printed_text="" empty="1"/>
<zone rect="102,203,111,212" printed_text="" empty="1"/>
<zone rect="102,212,111,221" printed_text="" empty="1"/>
<zone rect="102,221,111,229" printed_text="" empty="1"/>
<zone rect="102,229,111,237" printed_text="" empty="1"/>
<zone rect="102,237,111,245" printed_text="" empty="1"/>
<zone rect="102,245,111,254" printed_text="" empty="1"/>
<zone rect="102,254,111,261" printed_text="" empty="1"/>
<zone rect="102,261,111,269" printed_text="" empty="1"/>
<zone rect="102,269,111,278" printed_text="" empty="1"/>
<zone rect="102,278,111,286" printed_text="" empty="1"/>
<zone rect="102,286,111,293" printed_text="" empty="1"/>
<zone rect="102,293,111,302" printed_text="" empty="1"/>
<zone rect="102,302,111,310" printed_text="" empty="1"/>
<zone rect="102,310,111,317" printed_text="" empty="1"/>
<zone rect="102,317,111,326" printed_text="" empty="1"/>
<zone rect="102,326,111,334" printed_text="" empty="1"/>
<zone rect="102,334,111,344" printed_text="" empty="1"/>
<zone rect="102,344,111,351" printed_text="" empty="1"/>
<zone rect="102,351,111,361" printed_text="" empty="1"/>
<zone rect="102,361,111,368" printed_text="" empty="1"/>
<zone rect="102,368,111,376" printed_text="" empty="1"/>
<zone rect="102,376,111,384" printed_text="" empty="1"/>
<zone rect="102,384,111,392" printed_text="" empty="1"/>
<zone rect="102,392,111,400" printed_text="" empty="1"/>
<zone rect="102,400,111,409" printed_text="" empty="1"/>
<zone rect="102,409,111,416" printed_text="" empty="1"/>
<zone rect="102,416,111,425" printed_text="" empty="1"/>
<zone rect="102,425,111,433" printed_text="" empty="1"/>
<zone rect="102,433,111,441" printed_text="" empty="1"/>
<zone rect="102,441,111,449" printed_text="" empty="1"/>
<zone rect="102,449,111,456" printed_text="" empty="1"/>
<zone rect="102,456,111,464" printed_text="" empty="1"/>
<zone rect="102,464,111,474" printed_text="" empty="1"/>
<zone rect="102,474,111,481" printed_text="" empty="1"/>
<zone rect="102,481,111,490" printed_text="" empty="1"/>
<zone rect="102,490,111,497" printed_text="" empty="1"/>
<zone rect="102,497,111,506" printed_text="" empty="1"/>
<zone rect="102,506,111,514" printed_text="" empty="1"/>
<zone rect="102,514,111,522" printed_text="" empty="1"/>
<zone rect="102,522,111,530" printed_text="" empty="1"/>
<zone rect="102,530,111,538" printed_text="" empty="1"/>
<zone rect="102,538,111,545" printed_text="" empty="1"/>
<zone rect="102,545,111,554" printed_text="" empty="1"/>
<zone rect="102,554,111,561" printed_text="" empty="1"/>
<zone rect="102,561,111,569" printed_text="" empty="1"/>
<zone rect="102,569,111,579" printed_text="" empty="1"/>
<zone rect="111,161,118,169" printed_text="5" empty="0"/>
<zone rect="111,169,118,178" printed_text="" empty="1"/>
<zone rect="111,178,118,186" printed_text="" empty="1"/>
<zone rect="111,186,118,194" printed_text="" empty="1"/>
<zone rect="111,194,118,203" printed_text="" empty="1"/>
<zone rect="111,203,118,212" printed_text="" empty="1"/>

<zone rect="111,212,118,221" printed_text="" empty="1"/>
<zone rect="111,221,118,229" printed_text="" empty="1"/>
<zone rect="111,229,118,237" printed_text="" empty="1"/>
<zone rect="111,237,118,245" printed_text="" empty="1"/>
<zone rect="111,245,118,254" printed_text="" empty="1"/>
<zone rect="111,254,118,261" printed_text="" empty="1"/>
<zone rect="111,261,118,269" printed_text="" empty="1"/>
<zone rect="111,269,118,278" printed_text="" empty="1"/>
<zone rect="111,278,118,286" printed_text="" empty="1"/>
<zone rect="111,286,118,293" printed_text="" empty="1"/>
<zone rect="111,293,118,302" printed_text="" empty="1"/>
<zone rect="111,302,118,310" printed_text="" empty="1"/>
<zone rect="111,310,118,317" printed_text="" empty="1"/>
<zone rect="111,317,118,326" printed_text="" empty="1"/>
<zone rect="111,326,118,334" printed_text="" empty="1"/>
<zone rect="111,334,118,344" printed_text="" empty="1"/>
<zone rect="111,344,118,351" printed_text="" empty="1"/>
<zone rect="111,351,118,361" printed_text="" empty="1"/>
<zone rect="111,361,118,368" printed_text="" empty="1"/>
<zone rect="111,368,118,376" printed_text="" empty="1"/>
<zone rect="111,376,118,384" printed_text="" empty="1"/>
<zone rect="111,384,118,392" printed_text="" empty="1"/>
<zone rect="111,392,118,400" printed_text="" empty="1"/>
<zone rect="111,400,118,409" printed_text="" empty="1"/>
<zone rect="111,409,118,416" printed_text="" empty="1"/>
<zone rect="111,416,118,425" printed_text="" empty="1"/>
<zone rect="111,425,118,433" printed_text="" empty="1"/>
<zone rect="111,433,118,441" printed_text="" empty="1"/>
<zone rect="111,441,118,449" printed_text="" empty="1"/>
<zone rect="111,449,118,456" printed_text="" empty="1"/>
<zone rect="111,456,118,464" printed_text="" empty="1"/>
<zone rect="111,464,118,474" printed_text="" empty="1"/>
<zone rect="111,474,118,481" printed_text="" empty="1"/>
<zone rect="111,481,118,490" printed_text="" empty="1"/>
<zone rect="111,490,118,497" printed_text="" empty="1"/>
<zone rect="111,497,118,506" printed_text="" empty="1"/>
<zone rect="111,506,118,514" printed_text="" empty="1"/>
<zone rect="111,514,118,522" printed_text="" empty="1"/>
<zone rect="111,522,118,530" printed_text="" empty="1"/>
<zone rect="111,530,118,538" printed_text="" empty="1"/>
<zone rect="111,538,118,545" printed_text="" empty="1"/>
<zone rect="111,545,118,554" printed_text="" empty="1"/>
<zone rect="111,554,118,561" printed_text="" empty="1"/>
<zone rect="111,561,118,569" printed_text="" empty="1"/>
<zone rect="111,569,118,579" printed_text="" empty="1"/>
<zone rect="118,161,127,169" printed_text="6" empty="0"/>
<zone rect="118,169,127,178" printed_text="" empty="1"/>
<zone rect="118,178,127,186" printed_text="" empty="1"/>
<zone rect="118,186,127,194" printed_text="" empty="1"/>
<zone rect="118,194,127,203" printed_text="" empty="1"/>
<zone rect="118,203,127,212" printed_text="" empty="1"/>
<zone rect="118,212,127,221" printed_text="" empty="1"/>
<zone rect="118,221,127,229" printed_text="" empty="1"/>
<zone rect="118,229,127,237" printed_text="" empty="1"/>
<zone rect="118,237,127,245" printed_text="" empty="1"/>
<zone rect="118,245,127,254" printed_text="" empty="1"/>
<zone rect="118,254,127,261" printed_text="" empty="1"/>
<zone rect="118,261,127,269" printed_text="" empty="1"/>
<zone rect="118,269,127,278" printed_text="" empty="1"/>
<zone rect="118,278,127,286" printed_text="" empty="1"/>
<zone rect="118,286,127,293" printed_text="" empty="1"/>
<zone rect="118,293,127,302" printed_text="" empty="1"/>
<zone rect="118,302,127,310" printed_text="" empty="1"/>
<zone rect="118,310,127,317" printed_text="" empty="1"/>
<zone rect="118,317,127,326" printed_text="" empty="1"/>
<zone rect="118,326,127,334" printed_text="" empty="1"/>
<zone rect="118,334,127,344" printed_text="" empty="1"/>
<zone rect="118,344,127,351" printed_text="" empty="1"/>
<zone rect="118,351,127,361" printed_text="" empty="1"/>
<zone rect="118,361,127,368" printed_text="" empty="1"/>
<zone rect="118,368,127,376" printed_text="" empty="1"/>
<zone rect="118,376,127,384" printed_text="" empty="1"/>
<zone rect="118,384,127,392" printed_text="" empty="1"/>
<zone rect="118,392,127,400" printed_text="" empty="1"/>
<zone rect="118,400,127,409" printed_text="" empty="1"/>
<zone rect="118,409,127,416" printed_text="" empty="1"/>
<zone rect="118,416,127,425" printed_text="" empty="1"/>
<zone rect="118,425,127,433" printed_text="" empty="1"/>
<zone rect="118,433,127,441" printed_text="" empty="1"/>
<zone rect="118,441,127,449" printed_text="" empty="1"/>
<zone rect="118,449,127,456" printed_text="" empty="1"/>
<zone rect="118,456,127,464" printed_text="" empty="1"/>
<zone rect="118,464,127,474" printed_text="" empty="1"/>
<zone rect="118,474,127,481" printed_text="" empty="1"/>
<zone rect="118,481,127,490" printed_text="" empty="1"/>
<zone rect="118,490,127,497" printed_text="" empty="1"/>
<zone rect="118,497,127,506" printed_text="" empty="1"/>
<zone rect="118,506,127,514" printed_text="" empty="1"/>
<zone rect="118,514,127,522" printed_text="" empty="1"/>
<zone rect="118,522,127,530" printed_text="" empty="1"/>
<zone rect="118,530,127,538" printed_text="" empty="1"/>
<zone rect="118,538,127,545" printed_text="" empty="1"/>
<zone rect="118,545,127,554" printed_text="" empty="1"/>
<zone rect="118,554,127,561" printed_text="" empty="1"/>
<zone rect="118,561,127,569" printed_text="" empty="1"/>
<zone rect="118,569,127,579" printed_text="" empty="1"/>
<zone rect="127,161,139,169" printed_text="7" empty="0"/>
<zone rect="127,169,139,178" printed_text="" empty="1"/>
<zone rect="127,178,139,186" printed_text="" empty="1"/>
<zone rect="127,186,139,194" printed_text="" empty="1"/>
<zone rect="127,194,139,203" printed_text="" empty="1"/>
<zone rect="127,203,139,212" printed_text="" empty="1"/>
<zone rect="127,212,139,221" printed_text="" empty="1"/>
<zone rect="127,221,139,229" printed_text="" empty="1"/>
<zone rect="127,229,139,237" printed_text="" empty="1"/>
<zone rect="127,237,139,245" printed_text="" empty="1"/>
<zone rect="127,245,139,254" printed_text="" empty="1"/>
<zone rect="127,254,139,261" printed_text="" empty="1"/>
<zone rect="127,261,139,269" printed_text="" empty="1"/>
<zone rect="127,269,139,278" printed_text="" empty="1"/>
<zone rect="127,278,139,286" printed_text="" empty="1"/>
<zone rect="127,286,139,293" printed_text="" empty="1"/>
<zone rect="127,293,139,302" printed_text="" empty="1"/>
<zone rect="127,302,139,310" printed_text="" empty="1"/>
<zone rect="127,310,139,317" printed_text="" empty="1"/>
<zone rect="127,317,139,326" printed_text="" empty="1"/>
<zone rect="127,326,139,334" printed_text="" empty="1"/>
<zone rect="127,334,139,344" printed_text="" empty="1"/>
<zone rect="127,344,139,351" printed_text="" empty="1"/>
<zone rect="127,351,139,361" printed_text="" empty="1"/>
<zone rect="127,361,139,368" printed_text="" empty="1"/>
<zone rect="127,368,139,376" printed_text="" empty="1"/>
<zone rect="127,376,139,384" printed_text="" empty="1"/>
<zone rect="127,384,139,392" printed_text="" empty="1"/>
<zone rect="127,392,139,400" printed_text="" empty="1"/>
<zone rect="127,400,139,409" printed_text="" empty="1"/>
<zone rect="127,409,139,416" printed_text="" empty="1"/>
<zone rect="127,416,139,425" printed_text="" empty="1"/>
<zone rect="127,425,139,433" printed_text="" empty="1"/>
<zone rect="127,433,139,441" printed_text="" empty="1"/>
<zone rect="127,441,139,449" printed_text="" empty="1"/>
<zone rect="127,449,139,456" printed_text="" empty="1"/>
<zone rect="127,456,139,464" printed_text="" empty="1"/>
<zone rect="127,464,139,474" printed_text="" empty="1"/>
<zone rect="127,474,139,481" printed_text="" empty="1"/>
<zone rect="127,481,139,490" printed_text="" empty="1"/>
<zone rect="127,490,139,497" printed_text="" empty="1"/>
<zone rect="127,497,139,506" printed_text="" empty="1"/>
<zone rect="127,506,139,514" printed_text="" empty="1"/>
<zone rect="127,514,139,522" printed_text="" empty="1"/>

<zone rect="127,522,139,530" printed_text="" empty="1"/>
<zone rect="127,530,139,538" printed_text="" empty="1"/>
<zone rect="127,538,139,545" printed_text="" empty="1"/>
<zone rect="127,545,139,554" printed_text="" empty="1"/>
<zone rect="127,554,139,561" printed_text="" empty="1"/>
<zone rect="127,561,139,569" printed_text="" empty="1"/>
<zone rect="127,569,139,579" printed_text="" empty="1"/>
<zone rect="139,161,168,169" printed_text="8" empty="0"/>
<zone rect="139,169,168,178" printed_text="" empty="1"/>
<zone rect="139,178,168,186" printed_text="" empty="1"/>
<zone rect="139,186,168,194" printed_text="" empty="1"/>
<zone rect="139,194,168,203" printed_text="" empty="1"/>
<zone rect="139,203,168,212" printed_text="" empty="1"/>
<zone rect="139,212,168,221" printed_text="" empty="1"/>
<zone rect="139,221,168,229" printed_text="" empty="1"/>
<zone rect="139,229,168,237" printed_text="" empty="1"/>
<zone rect="139,237,168,245" printed_text="" empty="1"/>
<zone rect="139,245,168,254" printed_text="" empty="1"/>
<zone rect="139,254,168,261" printed_text="" empty="1"/>
<zone rect="139,261,168,269" printed_text="" empty="1"/>
<zone rect="139,269,168,278" printed_text="" empty="1"/>
<zone rect="139,278,168,286" printed_text="" empty="1"/>
<zone rect="139,286,168,293" printed_text="" empty="1"/>
<zone rect="139,293,168,302" printed_text="" empty="1"/>
<zone rect="139,302,168,310" printed_text="" empty="1"/>
<zone rect="139,310,168,317" printed_text="" empty="1"/>
<zone rect="139,317,168,326" printed_text="" empty="1"/>
<zone rect="139,326,168,334" printed_text="" empty="1"/>
<zone rect="139,334,168,344" printed_text="" empty="1"/>
<zone rect="139,344,168,351" printed_text="" empty="1"/>
<zone rect="139,351,168,361" printed_text="" empty="1"/>
<zone rect="139,361,168,368" printed_text="" empty="1"/>
<zone rect="139,368,168,376" printed_text="" empty="1"/>
<zone rect="139,376,168,384" printed_text="" empty="1"/>
<zone rect="139,384,168,392" printed_text="" empty="1"/>
<zone rect="139,392,168,400" printed_text="" empty="1"/>
<zone rect="139,400,168,409" printed_text="" empty="1"/>
<zone rect="139,409,168,416" printed_text="" empty="1"/>
<zone rect="139,416,168,425" printed_text="" empty="1"/>
<zone rect="139,425,168,433" printed_text="" empty="1"/>
<zone rect="139,433,168,441" printed_text="" empty="1"/>
<zone rect="139,441,168,449" printed_text="" empty="1"/>
<zone rect="139,449,168,456" printed_text="" empty="1"/>
<zone rect="139,456,168,464" printed_text="" empty="1"/>
<zone rect="139,464,168,474" printed_text="" empty="1"/>
<zone rect="139,474,168,481" printed_text="" empty="1"/>
<zone rect="139,481,168,490" printed_text="" empty="1"/>
<zone rect="139,490,168,497" printed_text="" empty="1"/>
<zone rect="139,497,168,506" printed_text="" empty="1"/>
<zone rect="139,506,168,514" printed_text="" empty="1"/>
<zone rect="139,514,168,522" printed_text="" empty="1"/>
<zone rect="139,522,168,530" printed_text="" empty="1"/>
<zone rect="139,530,168,538" printed_text="" empty="1"/>
<zone rect="139,538,168,545" printed_text="" empty="1"/>
<zone rect="139,545,168,554" printed_text="" empty="1"/>
<zone rect="139,554,168,561" printed_text="" empty="1"/>
<zone rect="139,561,168,569" printed_text="" empty="1"/>
<zone rect="139,569,168,579" printed_text="" empty="1"/>
<zone rect="168,161,173,169" printed_text="9" empty="0"/>
<zone rect="168,169,173,178" printed_text="" empty="1"/>
<zone rect="168,178,173,186" printed_text="" empty="1"/>
<zone rect="168,186,173,194" printed_text="" empty="1"/>
<zone rect="168,194,173,203" printed_text="" empty="1"/>
<zone rect="168,203,173,212" printed_text="" empty="1"/>
<zone rect="168,212,173,221" printed_text="" empty="1"/>
<zone rect="168,221,173,229" printed_text="" empty="1"/>
<zone rect="168,229,173,237" printed_text="" empty="1"/>
<zone rect="168,237,173,245" printed_text="" empty="1"/>
<zone rect="168,245,173,254" printed_text="" empty="1"/>
<zone rect="168,254,173,261" printed_text="" empty="1"/>
<zone rect="168,261,173,269" printed_text="" empty="1"/>
<zone rect="168,269,173,278" printed_text="" empty="1"/>
<zone rect="168,278,173,286" printed_text="" empty="1"/>
<zone rect="168,286,173,293" printed_text="" empty="1"/>
<zone rect="168,293,173,302" printed_text="" empty="1"/>
<zone rect="168,302,173,310" printed_text="" empty="1"/>
<zone rect="168,310,173,317" printed_text="" empty="1"/>
<zone rect="168,317,173,326" printed_text="" empty="1"/>
<zone rect="168,326,173,334" printed_text="" empty="1"/>
<zone rect="168,334,173,344" printed_text="" empty="1"/>
<zone rect="168,344,173,351" printed_text="" empty="1"/>
<zone rect="168,351,173,361" printed_text="" empty="1"/>
<zone rect="168,361,173,368" printed_text="" empty="1"/>
<zone rect="168,368,173,376" printed_text="" empty="1"/>
<zone rect="168,376,173,384" printed_text="" empty="1"/>
<zone rect="168,384,173,392" printed_text="" empty="1"/>
<zone rect="168,392,173,400" printed_text="" empty="1"/>
<zone rect="168,400,173,409" printed_text="" empty="1"/>
<zone rect="168,409,173,416" printed_text="" empty="1"/>
<zone rect="168,416,173,425" printed_text="" empty="1"/>
<zone rect="168,425,173,433" printed_text="" empty="1"/>
<zone rect="168,433,173,441" printed_text="" empty="1"/>
<zone rect="168,441,173,449" printed_text="" empty="1"/>
<zone rect="168,449,173,456" printed_text="" empty="1"/>
<zone rect="168,456,173,464" printed_text="" empty="1"/>
<zone rect="168,464,173,474" printed_text="" empty="1"/>
<zone rect="168,474,173,481" printed_text="" empty="1"/>
<zone rect="168,481,173,490" printed_text="" empty="1"/>
<zone rect="168,490,173,497" printed_text="" empty="1"/>
<zone rect="168,497,173,506" printed_text="" empty="1"/>
<zone rect="168,506,173,514" printed_text="" empty="1"/>
<zone rect="168,514,173,522" printed_text="" empty="1"/>
<zone rect="168,522,173,530" printed_text="" empty="1"/>
<zone rect="168,530,173,538" printed_text="" empty="1"/>
<zone rect="168,538,173,545" printed_text="" empty="1"/>
<zone rect="168,545,173,554" printed_text="" empty="1"/>
<zone rect="168,554,173,561" printed_text="" empty="1"/>
<zone rect="168,561,173,569" printed_text="" empty="1"/>
<zone rect="168,569,173,579" printed_text="" empty="1"/>
<zone rect="173,161,179,169" printed_text="10"empty="0"/>
<zone rect="173,169,179,178" printed_text="" empty="1"/>
<zone rect="173,178,179,186" printed_text="" empty="1"/>
<zone rect="173,186,179,194" printed_text="" empty="1"/>
<zone rect="173,194,179,203" printed_text="" empty="1"/>
<zone rect="173,203,179,212" printed_text="" empty="1"/>
<zone rect="173,212,179,221" printed_text="" empty="1"/>
<zone rect="173,221,179,229" printed_text="" empty="1"/>
<zone rect="173,229,179,237" printed_text="" empty="1"/>
<zone rect="173,237,179,245" printed_text="" empty="1"/>
<zone rect="173,245,179,254" printed_text="" empty="1"/>
<zone rect="173,254,179,261" printed_text="" empty="1"/>
<zone rect="173,261,179,269" printed_text="" empty="1"/>
<zone rect="173,269,179,278" printed_text="" empty="1"/>
<zone rect="173,278,179,286" printed_text="" empty="1"/>
<zone rect="173,286,179,293" printed_text="" empty="1"/>
<zone rect="173,293,179,302" printed_text="" empty="1"/>
<zone rect="173,302,179,310" printed_text="" empty="1"/>
<zone rect="173,310,179,317" printed_text="" empty="1"/>
<zone rect="173,317,179,326" printed_text="" empty="1"/>
<zone rect="173,326,179,334" printed_text="" empty="1"/>
<zone rect="173,334,179,344" printed_text="" empty="1"/>
<zone rect="173,344,179,351" printed_text="" empty="1"/>
<zone rect="173,351,179,361" printed_text="" empty="1"/>
<zone rect="173,361,179,368" printed_text="" empty="1"/>
<zone rect="173,368,179,376" printed_text="" empty="1"/>
<zone rect="173,376,179,384" printed_text="" empty="1"/>
<zone rect="173,384,179,392" printed_text="" empty="1"/>
<zone rect="173,392,179,400" printed_text="" empty="1"/>
<zone rect="173,400,179,409" printed_text="" empty="1"/>
<zone rect="173,409,179,416" printed_text="" empty="1"/>

<zone rect="173,416,179,425" printed_text="" empty="1"/>
<zone rect="173,425,179,433" printed_text="" empty="1"/>
<zone rect="173,433,179,441" printed_text="" empty="1"/>
<zone rect="173,441,179,449" printed_text="" empty="1"/>
<zone rect="173,449,179,456" printed_text="" empty="1"/>
<zone rect="173,456,179,464" printed_text="" empty="1"/>
<zone rect="173,464,179,474" printed_text="" empty="1"/>
<zone rect="173,474,179,481" printed_text="" empty="1"/>
<zone rect="173,481,179,490" printed_text="" empty="1"/>
<zone rect="173,490,179,497" printed_text="" empty="1"/>
<zone rect="173,497,179,506" printed_text="" empty="1"/>
<zone rect="173,506,179,514" printed_text="" empty="1"/>
<zone rect="173,514,179,522" printed_text="" empty="1"/>
<zone rect="173,522,179,530" printed_text="" empty="1"/>
<zone rect="173,530,179,538" printed_text="" empty="1"/>
<zone rect="173,538,179,545" printed_text="" empty="1"/>
<zone rect="173,545,179,554" printed_text="" empty="1"/>
<zone rect="173,554,179,561" printed_text="" empty="1"/>
<zone rect="173,561,179,569" printed_text="" empty="1"/>
<zone rect="173,569,179,579" printed_text="" empty="1"/>
<zone rect="179,161,184,169" printed_text="11"empty="0"/>
<zone rect="179,169,184,178" printed_text="" empty="1"/>
<zone rect="179,178,184,186" printed_text="" empty="1"/>
<zone rect="179,186,184,194" printed_text="" empty="1"/>
<zone rect="179,194,184,203" printed_text="" empty="1"/>
<zone rect="179,203,184,212" printed_text="" empty="1"/>
<zone rect="179,212,184,221" printed_text="" empty="1"/>
<zone rect="179,221,184,229" printed_text="" empty="1"/>
<zone rect="179,229,184,237" printed_text="" empty="1"/>
<zone rect="179,237,184,245" printed_text="" empty="1"/>
<zone rect="179,245,184,254" printed_text="" empty="1"/>
<zone rect="179,254,184,261" printed_text="" empty="1"/>
<zone rect="179,261,184,269" printed_text="" empty="1"/>
<zone rect="179,269,184,278" printed_text="" empty="1"/>
<zone rect="179,278,184,286" printed_text="" empty="1"/>
<zone rect="179,286,184,293" printed_text="" empty="1"/>
<zone rect="179,293,184,302" printed_text="" empty="1"/>
<zone rect="179,302,184,310" printed_text="" empty="1"/>
<zone rect="179,310,184,317" printed_text="" empty="1"/>
<zone rect="179,317,184,326" printed_text="" empty="1"/>
<zone rect="179,326,184,334" printed_text="" empty="1"/>
<zone rect="179,334,184,344" printed_text="" empty="1"/>
<zone rect="179,344,184,351" printed_text="" empty="1"/>
<zone rect="179,351,184,361" printed_text="" empty="1"/>
<zone rect="179,361,184,368" printed_text="" empty="1"/>
<zone rect="179,368,184,376" printed_text="" empty="1"/>
<zone rect="179,376,184,384" printed_text="" empty="1"/>
<zone rect="179,384,184,392" printed_text="" empty="1"/>
<zone rect="179,392,184,400" printed_text="" empty="1"/>
<zone rect="179,400,184,409" printed_text="" empty="1"/>
<zone rect="179,409,184,416" printed_text="" empty="1"/>
<zone rect="179,416,184,425" printed_text="" empty="1"/>
<zone rect="179,425,184,433" printed_text="" empty="1"/>
<zone rect="179,433,184,441" printed_text="" empty="1"/>
<zone rect="179,441,184,449" printed_text="" empty="1"/>
<zone rect="179,449,184,456" printed_text="" empty="1"/>
<zone rect="179,456,184,464" printed_text="" empty="1"/>
<zone rect="179,464,184,474" printed_text="" empty="1"/>
<zone rect="179,474,184,481" printed_text="" empty="1"/>
<zone rect="179,481,184,490" printed_text="" empty="1"/>
<zone rect="179,490,184,497" printed_text="" empty="1"/>
<zone rect="179,497,184,506" printed_text="" empty="1"/>
<zone rect="179,506,184,514" printed_text="" empty="1"/>
<zone rect="179,514,184,522" printed_text="" empty="1"/>
<zone rect="179,522,184,530" printed_text="" empty="1"/>
<zone rect="179,530,184,538" printed_text="" empty="1"/>
<zone rect="179,538,184,545" printed_text="" empty="1"/>
<zone rect="179,545,184,554" printed_text="" empty="1"/>
<zone rect="179,554,184,561" printed_text="" empty="1"/>
<zone rect="179,561,184,569" printed_text="" empty="1"/>
<zone rect="179,569,184,579" printed_text="" empty="1"/>
<zone rect="184,161,191,169" printed_text="12"empty="0"/>
<zone rect="184,169,191,178" printed_text="" empty="1"/>
<zone rect="184,178,191,186" printed_text="" empty="1"/>
<zone rect="184,186,191,194" printed_text="" empty="1"/>
<zone rect="184,194,191,203" printed_text="" empty="1"/>
<zone rect="184,203,191,212" printed_text="" empty="1"/>
<zone rect="184,212,191,221" printed_text="" empty="1"/>
<zone rect="184,221,191,229" printed_text="" empty="1"/>
<zone rect="184,229,191,237" printed_text="" empty="1"/>
<zone rect="184,237,191,245" printed_text="" empty="1"/>
<zone rect="184,245,191,254" printed_text="" empty="1"/>
<zone rect="184,254,191,261" printed_text="" empty="1"/>
<zone rect="184,261,191,269" printed_text="" empty="1"/>
<zone rect="184,269,191,278" printed_text="" empty="1"/>
<zone rect="184,278,191,286" printed_text="" empty="1"/>
<zone rect="184,286,191,293" printed_text="" empty="1"/>
<zone rect="184,293,191,302" printed_text="" empty="1"/>
<zone rect="184,302,191,310" printed_text="" empty="1"/>
<zone rect="184,310,191,317" printed_text="" empty="1"/>
<zone rect="184,317,191,326" printed_text="" empty="1"/>
<zone rect="184,326,191,334" printed_text="" empty="1"/>
<zone rect="184,334,191,344" printed_text="" empty="1"/>
<zone rect="184,344,191,351" printed_text="" empty="1"/>
<zone rect="184,351,191,361" printed_text="" empty="1"/>
<zone rect="184,361,191,368" printed_text="" empty="1"/>
<zone rect="184,368,191,376" printed_text="" empty="1"/>
<zone rect="184,376,191,384" printed_text="" empty="1"/>
<zone rect="184,384,191,392" printed_text="" empty="1"/>
<zone rect="184,392,191,400" printed_text="" empty="1"/>
<zone rect="184,400,191,409" printed_text="" empty="1"/>
<zone rect="184,409,191,416" printed_text="" empty="1"/>
<zone rect="184,416,191,425" printed_text="" empty="1"/>
<zone rect="184,425,191,433" printed_text="" empty="1"/>
<zone rect="184,433,191,441" printed_text="" empty="1"/>
<zone rect="184,441,191,449" printed_text="" empty="1"/>
<zone rect="184,449,191,456" printed_text="" empty="1"/>
<zone rect="184,456,191,464" printed_text="" empty="1"/>
<zone rect="184,464,191,474" printed_text="" empty="1"/>
<zone rect="184,474,191,481" printed_text="" empty="1"/>
<zone rect="184,481,191,490" printed_text="" empty="1"/>
<zone rect="184,490,191,497" printed_text="" empty="1"/>
<zone rect="184,497,191,506" printed_text="" empty="1"/>
<zone rect="184,506,191,514" printed_text="" empty="1"/>
<zone rect="184,514,191,522" printed_text="" empty="1"/>
<zone rect="184,522,191,530" printed_text="" empty="1"/>
<zone rect="184,530,191,538" printed_text="" empty="1"/>
<zone rect="184,538,191,545" printed_text="" empty="1"/>
<zone rect="184,545,191,554" printed_text="" empty="1"/>
<zone rect="184,554,191,561" printed_text="" empty="1"/>
<zone rect="184,561,191,569" printed_text="" empty="1"/>
<zone rect="184,569,191,579" printed_text="" empty="1"/>
<zone rect="191,161,240,169" printed_text="13"empty="0"/>
<zone rect="191,169,240,178" printed_text="" empty="1"/>
<zone rect="191,178,240,186" printed_text="" empty="1"/>
<zone rect="191,186,240,194" printed_text="" empty="1"/>
<zone rect="191,194,240,203" printed_text="" empty="1"/>
<zone rect="191,203,240,212" printed_text="" empty="1"/>
<zone rect="191,212,240,221" printed_text="" empty="1"/>
<zone rect="191,221,240,229" printed_text="" empty="1"/>
<zone rect="191,229,240,237" printed_text="" empty="1"/>
<zone rect="191,237,240,245" printed_text="" empty="1"/>
<zone rect="191,245,240,254" printed_text="" empty="1"/>
<zone rect="191,254,240,261" printed_text="" empty="1"/>
<zone rect="191,261,240,269" printed_text="" empty="1"/>
<zone rect="191,269,240,278" printed_text="" empty="1"/>
<zone rect="191,278,240,286" printed_text="" empty="1"/>
<zone rect="191,286,240,293" printed_text="" empty="1"/>
<zone rect="191,293,240,302" printed_text="" empty="1"/>
<zone rect="191,302,240,310" printed_text="" empty="1"/>

<zone rect="191,310,240,317" printed_text="" empty="1"/>
<zone rect="191,317,240,326" printed_text="" empty="1"/>
<zone rect="191,326,240,334" printed_text="" empty="1"/>
<zone rect="191,334,240,344" printed_text="" empty="1"/>
<zone rect="191,344,240,351" printed_text="" empty="1"/>
<zone rect="191,351,240,361" printed_text="" empty="1"/>
<zone rect="191,361,240,368" printed_text="" empty="1"/>
<zone rect="191,368,240,376" printed_text="" empty="1"/>
<zone rect="191,376,240,384" printed_text="" empty="1"/>
<zone rect="191,384,240,392" printed_text="" empty="1"/>
<zone rect="191,392,240,400" printed_text="" empty="1"/>
<zone rect="191,400,240,409" printed_text="" empty="1"/>
<zone rect="191,409,240,416" printed_text="" empty="1"/>
<zone rect="191,416,240,425" printed_text="" empty="1"/>
<zone rect="191,425,240,433" printed_text="" empty="1"/>
<zone rect="191,433,240,441" printed_text="" empty="1"/>
<zone rect="191,441,240,449" printed_text="" empty="1"/>
<zone rect="191,449,240,456" printed_text="" empty="1"/>
<zone rect="191,456,240,464" printed_text="" empty="1"/>
<zone rect="191,464,240,474" printed_text="" empty="1"/>
<zone rect="191,474,240,481" printed_text="" empty="1"/>
<zone rect="191,481,240,490" printed_text="" empty="1"/>
<zone rect="191,490,240,497" printed_text="" empty="1"/>
<zone rect="191,497,240,506" printed_text="" empty="1"/>
<zone rect="191,506,240,514" printed_text="" empty="1"/>
<zone rect="191,514,240,522" printed_text="" empty="1"/>
<zone rect="191,522,240,530" printed_text="" empty="1"/>
<zone rect="191,530,240,538" printed_text="" empty="1"/>
<zone rect="191,538,240,545" printed_text="" empty="1"/>
<zone rect="191,545,240,554" printed_text="" empty="1"/>
<zone rect="191,554,240,561" printed_text="" empty="1"/>
<zone rect="191,561,240,569" printed_text="" empty="1"/>
<zone rect="191,569,240,579" printed_text="" empty="1"/>
<zone rect="240,161,251,169" printed_text="14"empty="0"/>
<zone rect="240,169,251,178" printed_text="" empty="1"/>
<zone rect="240,178,251,186" printed_text="" empty="1"/>
<zone rect="240,186,251,194" printed_text="" empty="1"/>
<zone rect="240,194,251,203" printed_text="" empty="1"/>
<zone rect="240,203,251,212" printed_text="" empty="1"/>
<zone rect="240,212,251,221" printed_text="" empty="1"/>
<zone rect="240,221,251,229" printed_text="" empty="1"/>
<zone rect="240,229,251,237" printed_text="" empty="1"/>
<zone rect="240,237,251,245" printed_text="" empty="1"/>
<zone rect="240,245,251,254" printed_text="" empty="1"/>
<zone rect="240,254,251,261" printed_text="" empty="1"/>
<zone rect="240,261,251,269" printed_text="" empty="1"/>
<zone rect="240,269,251,278" printed_text="" empty="1"/>
<zone rect="240,278,251,286" printed_text="" empty="1"/>
<zone rect="240,286,251,293" printed_text="" empty="1"/>
<zone rect="240,293,251,302" printed_text="" empty="1"/>
<zone rect="240,302,251,310" printed_text="" empty="1"/>
<zone rect="240,310,251,317" printed_text="" empty="1"/>
<zone rect="240,317,251,326" printed_text="" empty="1"/>
<zone rect="240,326,251,334" printed_text="" empty="1"/>
<zone rect="240,334,251,344" printed_text="" empty="1"/>
<zone rect="240,344,251,351" printed_text="" empty="1"/>
<zone rect="240,351,251,361" printed_text="" empty="1"/>
<zone rect="240,361,251,368" printed_text="" empty="1"/>
<zone rect="240,368,251,376" printed_text="" empty="1"/>
<zone rect="240,376,251,384" printed_text="" empty="1"/>
<zone rect="240,384,251,392" printed_text="" empty="1"/>
<zone rect="240,392,251,400" printed_text="" empty="1"/>
<zone rect="240,400,251,409" printed_text="" empty="1"/>
<zone rect="240,409,251,416" printed_text="" empty="1"/>
<zone rect="240,416,251,425" printed_text="" empty="1"/>
<zone rect="240,425,251,433" printed_text="" empty="1"/>
<zone rect="240,433,251,441" printed_text="" empty="1"/>
<zone rect="240,441,251,449" printed_text="" empty="1"/>
<zone rect="240,449,251,456" printed_text="" empty="1"/>
<zone rect="240,456,251,464" printed_text="" empty="1"/>
<zone rect="240,464,251,474" printed_text="" empty="1"/>
<zone rect="240,474,251,481" printed_text="" empty="1"/>
<zone rect="240,481,251,490" printed_text="" empty="1"/>
<zone rect="240,490,251,497" printed_text="" empty="1"/>
<zone rect="240,497,251,506" printed_text="" empty="1"/>
<zone rect="240,506,251,514" printed_text="" empty="1"/>
<zone rect="240,514,251,522" printed_text="" empty="1"/>
<zone rect="240,522,251,530" printed_text="" empty="1"/>
<zone rect="240,530,251,538" printed_text="" empty="1"/>
<zone rect="240,538,251,545" printed_text="" empty="1"/>
<zone rect="240,545,251,554" printed_text="" empty="1"/>
<zone rect="240,554,251,561" printed_text="" empty="1"/>
<zone rect="240,561,251,569" printed_text="" empty="1"/>
<zone rect="240,569,251,579" printed_text="" empty="1"/>
<zone rect="251,161,281,169" printed_text="15"empty="0"/>
<zone rect="251,169,281,178" printed_text="" empty="1"/>
<zone rect="251,178,281,186" printed_text="" empty="1"/>
<zone rect="251,186,281,194" printed_text="" empty="1"/>
<zone rect="251,194,281,203" printed_text="" empty="1"/>
<zone rect="251,203,281,212" printed_text="" empty="1"/>
<zone rect="251,212,281,221" printed_text="" empty="1"/>
<zone rect="251,221,281,229" printed_text="" empty="1"/>
<zone rect="251,229,281,237" printed_text="" empty="1"/>
<zone rect="251,237,281,245" printed_text="" empty="1"/>
<zone rect="251,245,281,254" printed_text="" empty="1"/>
<zone rect="251,254,281,261" printed_text="" empty="1"/>
<zone rect="251,261,281,269" printed_text="" empty="1"/>
<zone rect="251,269,281,278" printed_text="" empty="1"/>
<zone rect="251,278,281,286" printed_text="" empty="1"/>
<zone rect="251,286,281,293" printed_text="" empty="1"/>
<zone rect="251,293,281,302" printed_text="" empty="1"/>
<zone rect="251,302,281,310" printed_text="" empty="1"/>
<zone rect="251,310,281,317" printed_text="" empty="1"/>
<zone rect="251,317,281,326" printed_text="" empty="1"/>
<zone rect="251,326,281,334" printed_text="" empty="1"/>
<zone rect="251,334,281,344" printed_text="" empty="1"/>
<zone rect="251,344,281,351" printed_text="" empty="1"/>
<zone rect="251,351,281,361" printed_text="" empty="1"/>
<zone rect="251,361,281,368" printed_text="" empty="1"/>
<zone rect="251,368,281,376" printed_text="" empty="1"/>
<zone rect="251,376,281,384" printed_text="" empty="1"/>
<zone rect="251,384,281,392" printed_text="" empty="1"/>
<zone rect="251,392,281,400" printed_text="" empty="1"/>
<zone rect="251,400,281,409" printed_text="" empty="1"/>
<zone rect="251,409,281,416" printed_text="" empty="1"/>
<zone rect="251,416,281,425" printed_text="" empty="1"/>
<zone rect="251,425,281,433" printed_text="" empty="1"/>
<zone rect="251,433,281,441" printed_text="" empty="1"/>
<zone rect="251,441,281,449" printed_text="" empty="1"/>
<zone rect="251,449,281,456" printed_text="" empty="1"/>
<zone rect="251,456,281,464" printed_text="" empty="1"/>
<zone rect="251,464,281,474" printed_text="" empty="1"/>
<zone rect="251,474,281,481" printed_text="" empty="1"/>
<zone rect="251,481,281,490" printed_text="" empty="1"/>
<zone rect="251,490,281,497" printed_text="" empty="1"/>
<zone rect="251,497,281,506" printed_text="" empty="1"/>
<zone rect="251,506,281,514" printed_text="" empty="1"/>
<zone rect="251,514,281,522" printed_text="" empty="1"/>
<zone rect="251,522,281,530" printed_text="" empty="1"/>
<zone rect="251,530,281,538" printed_text="" empty="1"/>
<zone rect="251,538,281,545" printed_text="" empty="1"/>
<zone rect="251,545,281,554" printed_text="" empty="1"/>
<zone rect="251,554,281,561" printed_text="" empty="1"/>
<zone rect="251,561,281,569" printed_text="" empty="1"/>
<zone rect="251,569,281,579" printed_text="" empty="1"/>
<zone rect="281,161,287,169" printed_text="16"empty="0"/>
<zone rect="281,169,287,178" printed_text="" empty="1"/>
<zone rect="281,178,287,186" printed_text="" empty="1"/>
<zone rect="281,186,287,194" printed_text="" empty="1"/>
<zone rect="281,194,287,203" printed_text="" empty="1"/>

<zone rect="281,203,287,212" printed_text="" empty="1"/>
<zone rect="281,212,287,221" printed_text="" empty="1"/>
<zone rect="281,221,287,229" printed_text="" empty="1"/>
<zone rect="281,229,287,237" printed_text="" empty="1"/>
<zone rect="281,237,287,245" printed_text="" empty="1"/>
<zone rect="281,245,287,254" printed_text="" empty="1"/>
<zone rect="281,254,287,261" printed_text="" empty="1"/>
<zone rect="281,261,287,269" printed_text="" empty="1"/>
<zone rect="281,269,287,278" printed_text="" empty="1"/>
<zone rect="281,278,287,286" printed_text="" empty="1"/>
<zone rect="281,286,287,293" printed_text="" empty="1"/>
<zone rect="281,293,287,302" printed_text="" empty="1"/>
<zone rect="281,302,287,310" printed_text="" empty="1"/>
<zone rect="281,310,287,317" printed_text="" empty="1"/>
<zone rect="281,317,287,326" printed_text="" empty="1"/>
<zone rect="281,326,287,334" printed_text="" empty="1"/>
<zone rect="281,334,287,344" printed_text="" empty="1"/>
<zone rect="281,344,287,351" printed_text="" empty="1"/>
<zone rect="281,351,287,361" printed_text="" empty="1"/>
<zone rect="281,361,287,368" printed_text="" empty="1"/>
<zone rect="281,368,287,376" printed_text="" empty="1"/>
<zone rect="281,376,287,384" printed_text="" empty="1"/>
<zone rect="281,384,287,392" printed_text="" empty="1"/>
<zone rect="281,392,287,400" printed_text="" empty="1"/>
<zone rect="281,400,287,409" printed_text="" empty="1"/>
<zone rect="281,409,287,416" printed_text="" empty="1"/>
<zone rect="281,416,287,425" printed_text="" empty="1"/>
<zone rect="281,425,287,433" printed_text="" empty="1"/>
<zone rect="281,433,287,441" printed_text="" empty="1"/>
<zone rect="281,441,287,449" printed_text="" empty="1"/>
<zone rect="281,449,287,456" printed_text="" empty="1"/>
<zone rect="281,456,287,464" printed_text="" empty="1"/>
<zone rect="281,464,287,474" printed_text="" empty="1"/>
<zone rect="281,474,287,481" printed_text="" empty="1"/>
<zone rect="281,481,287,490" printed_text="" empty="1"/>
<zone rect="281,490,287,497" printed_text="" empty="1"/>
<zone rect="281,497,287,506" printed_text="" empty="1"/>
<zone rect="281,506,287,514" printed_text="" empty="1"/>
<zone rect="281,514,287,522" printed_text="" empty="1"/>
<zone rect="281,522,287,530" printed_text="" empty="1"/>
<zone rect="281,530,287,538" printed_text="" empty="1"/>
<zone rect="281,538,287,545" printed_text="" empty="1"/>
<zone rect="281,545,287,554" printed_text="" empty="1"/>
<zone rect="281,554,287,561" printed_text="" empty="1"/>
<zone rect="281,561,287,569" printed_text="" empty="1"/>
<zone rect="281,569,287,579" printed_text="" empty="1"/>
<zone rect="287,161,294,169" printed_text="17"empty="0"/>
<zone rect="287,169,294,178" printed_text="" empty="1"/>
<zone rect="287,178,294,186" printed_text="" empty="1"/>
<zone rect="287,186,294,194" printed_text="" empty="1"/>
<zone rect="287,194,294,203" printed_text="" empty="1"/>
<zone rect="287,203,294,212" printed_text="" empty="1"/>
<zone rect="287,212,294,221" printed_text="" empty="1"/>
<zone rect="287,221,294,229" printed_text="" empty="1"/>
<zone rect="287,229,294,237" printed_text="" empty="1"/>
<zone rect="287,237,294,245" printed_text="" empty="1"/>
<zone rect="287,245,294,254" printed_text="" empty="1"/>
<zone rect="287,254,294,261" printed_text="" empty="1"/>
<zone rect="287,261,294,269" printed_text="" empty="1"/>
<zone rect="287,269,294,278" printed_text="" empty="1"/>
<zone rect="287,278,294,286" printed_text="" empty="1"/>
<zone rect="287,286,294,293" printed_text="" empty="1"/>
<zone rect="287,293,294,302" printed_text="" empty="1"/>
<zone rect="287,302,294,310" printed_text="" empty="1"/>
<zone rect="287,310,294,317" printed_text="" empty="1"/>
<zone rect="287,317,294,326" printed_text="" empty="1"/>
<zone rect="287,326,294,334" printed_text="" empty="1"/>
<zone rect="287,334,294,344" printed_text="" empty="1"/>
<zone rect="287,344,294,351" printed_text="" empty="1"/>
<zone rect="287,351,294,361" printed_text="" empty="1"/>
<zone rect="287,361,294,368" printed_text="" empty="1"/>
<zone rect="287,368,294,376" printed_text="" empty="1"/>
<zone rect="287,376,294,384" printed_text="" empty="1"/>
<zone rect="287,384,294,392" printed_text="" empty="1"/>
<zone rect="287,392,294,400" printed_text="" empty="1"/>
<zone rect="287,400,294,409" printed_text="" empty="1"/>
<zone rect="287,409,294,416" printed_text="" empty="1"/>
<zone rect="287,416,294,425" printed_text="" empty="1"/>
<zone rect="287,425,294,433" printed_text="" empty="1"/>
<zone rect="287,433,294,441" printed_text="" empty="1"/>
<zone rect="287,441,294,449" printed_text="" empty="1"/>
<zone rect="287,449,294,456" printed_text="" empty="1"/>
<zone rect="287,456,294,464" printed_text="" empty="1"/>
<zone rect="287,464,294,474" printed_text="" empty="1"/>
<zone rect="287,474,294,481" printed_text="" empty="1"/>
<zone rect="287,481,294,490" printed_text="" empty="1"/>
<zone rect="287,490,294,497" printed_text="" empty="1"/>
<zone rect="287,497,294,506" printed_text="" empty="1"/>
<zone rect="287,506,294,514" printed_text="" empty="1"/>
<zone rect="287,514,294,522" printed_text="" empty="1"/>
<zone rect="287,522,294,530" printed_text="" empty="1"/>
<zone rect="287,530,294,538" printed_text="" empty="1"/>
<zone rect="287,538,294,545" printed_text="" empty="1"/>
<zone rect="287,545,294,554" printed_text="" empty="1"/>
<zone rect="287,554,294,561" printed_text="" empty="1"/>
<zone rect="287,561,294,569" printed_text="" empty="1"/>
<zone rect="287,569,294,579" printed_text="" empty="1"/>
<zone rect="294,161,299,169" printed_text="18"empty="0"/>
<zone rect="294,169,299,178" printed_text="" empty="1"/>
<zone rect="294,178,299,186" printed_text="" empty="1"/>
<zone rect="294,186,299,194" printed_text="" empty="1"/>
<zone rect="294,194,299,203" printed_text="" empty="1"/>
<zone rect="294,203,299,212" printed_text="" empty="1"/>
<zone rect="294,212,299,221" printed_text="" empty="1"/>
<zone rect="294,221,299,229" printed_text="" empty="1"/>
<zone rect="294,229,299,237" printed_text="" empty="1"/>
<zone rect="294,237,299,245" printed_text="" empty="1"/>
<zone rect="294,245,299,254" printed_text="" empty="1"/>
<zone rect="294,254,299,261" printed_text="" empty="1"/>
<zone rect="294,261,299,269" printed_text="" empty="1"/>
<zone rect="294,269,299,278" printed_text="" empty="1"/>
<zone rect="294,278,299,286" printed_text="" empty="1"/>
<zone rect="294,286,299,293" printed_text="" empty="1"/>
<zone rect="294,293,299,302" printed_text="" empty="1"/>
<zone rect="294,302,299,310" printed_text="" empty="1"/>
<zone rect="294,310,299,317" printed_text="" empty="1"/>
<zone rect="294,317,299,326" printed_text="" empty="1"/>
<zone rect="294,326,299,334" printed_text="" empty="1"/>
<zone rect="294,334,299,344" printed_text="" empty="1"/>
<zone rect="294,344,299,351" printed_text="" empty="1"/>
<zone rect="294,351,299,361" printed_text="" empty="1"/>
<zone rect="294,361,299,368" printed_text="" empty="1"/>
<zone rect="294,368,299,376" printed_text="" empty="1"/>
<zone rect="294,376,299,384" printed_text="" empty="1"/>
<zone rect="294,384,299,392" printed_text="" empty="1"/>
<zone rect="294,392,299,400" printed_text="" empty="1"/>
<zone rect="294,400,299,409" printed_text="" empty="1"/>
<zone rect="294,409,299,416" printed_text="" empty="1"/>
<zone rect="294,416,299,425" printed_text="" empty="1"/>
<zone rect="294,425,299,433" printed_text="" empty="1"/>
<zone rect="294,433,299,441" printed_text="" empty="1"/>
<zone rect="294,441,299,449" printed_text="" empty="1"/>
<zone rect="294,449,299,456" printed_text="" empty="1"/>
<zone rect="294,456,299,464" printed_text="" empty="1"/>
<zone rect="294,464,299,474" printed_text="" empty="1"/>
<zone rect="294,474,299,481" printed_text="" empty="1"/>
<zone rect="294,481,299,490" printed_text="" empty="1"/>
<zone rect="294,490,299,497" printed_text="" empty="1"/>
<zone rect="294,497,299,506" printed_text="" empty="1"/>
<zone rect="294,506,299,514" printed_text="" empty="1"/>

<zone rect="294,514,299,522" printed_text="" empty="1"/>
<zone rect="294,522,299,530" printed_text="" empty="1"/>
<zone rect="294,530,299,538" printed_text="" empty="1"/>
<zone rect="294,538,299,545" printed_text="" empty="1"/>
<zone rect="294,545,299,554" printed_text="" empty="1"/>
<zone rect="294,554,299,561" printed_text="" empty="1"/>
<zone rect="294,561,299,569" printed_text="" empty="1"/>
<zone rect="294,569,299,579" printed_text="" empty="1"/>
<zone rect="299,161,305,169" printed_text="19"empty="0"/>
<zone rect="299,169,305,178" printed_text="" empty="1"/>
<zone rect="299,178,305,186" printed_text="" empty="1"/>
<zone rect="299,186,305,194" printed_text="" empty="1"/>
<zone rect="299,194,305,203" printed_text="" empty="1"/>
<zone rect="299,203,305,212" printed_text="" empty="1"/>
<zone rect="299,212,305,221" printed_text="" empty="1"/>
<zone rect="299,221,305,229" printed_text="" empty="1"/>
<zone rect="299,229,305,237" printed_text="" empty="1"/>
<zone rect="299,237,305,245" printed_text="" empty="1"/>
<zone rect="299,245,305,254" printed_text="" empty="1"/>
<zone rect="299,254,305,261" printed_text="" empty="1"/>
<zone rect="299,261,305,269" printed_text="" empty="1"/>
<zone rect="299,269,305,278" printed_text="" empty="1"/>
<zone rect="299,278,305,286" printed_text="" empty="1"/>
<zone rect="299,286,305,293" printed_text="" empty="1"/>
<zone rect="299,293,305,302" printed_text="" empty="1"/>
<zone rect="299,302,305,310" printed_text="" empty="1"/>
<zone rect="299,310,305,317" printed_text="" empty="1"/>
<zone rect="299,317,305,326" printed_text="" empty="1"/>
<zone rect="299,326,305,334" printed_text="" empty="1"/>
<zone rect="299,334,305,344" printed_text="" empty="1"/>
<zone rect="299,344,305,351" printed_text="" empty="1"/>
<zone rect="299,351,305,361" printed_text="" empty="1"/>
<zone rect="299,361,305,368" printed_text="" empty="1"/>
<zone rect="299,368,305,376" printed_text="" empty="1"/>
<zone rect="299,376,305,384" printed_text="" empty="1"/>
<zone rect="299,384,305,392" printed_text="" empty="1"/>
<zone rect="299,392,305,400" printed_text="" empty="1"/>
<zone rect="299,400,305,409" printed_text="" empty="1"/>
<zone rect="299,409,305,416" printed_text="" empty="1"/>
<zone rect="299,416,305,425" printed_text="" empty="1"/>
<zone rect="299,425,305,433" printed_text="" empty="1"/>
<zone rect="299,433,305,441" printed_text="" empty="1"/>
<zone rect="299,441,305,449" printed_text="" empty="1"/>
<zone rect="299,449,305,456" printed_text="" empty="1"/>
<zone rect="299,456,305,464" printed_text="" empty="1"/>
<zone rect="299,464,305,474" printed_text="" empty="1"/>
<zone rect="299,474,305,481" printed_text="" empty="1"/>
<zone rect="299,481,305,490" printed_text="" empty="1"/>
<zone rect="299,490,305,497" printed_text="" empty="1"/>
<zone rect="299,497,305,506" printed_text="" empty="1"/>
<zone rect="299,506,305,514" printed_text="" empty="1"/>
<zone rect="299,514,305,522" printed_text="" empty="1"/>
<zone rect="299,522,305,530" printed_text="" empty="1"/>
<zone rect="299,530,305,538" printed_text="" empty="1"/>
<zone rect="299,538,305,545" printed_text="" empty="1"/>
<zone rect="299,545,305,554" printed_text="" empty="1"/>
<zone rect="299,554,305,561" printed_text="" empty="1"/>
<zone rect="299,561,305,569" printed_text="" empty="1"/>
<zone rect="299,569,305,579" printed_text="" empty="1"/>
<zone rect="305,161,312,169" printed_text="20"empty="0"/>
<zone rect="305,169,312,178" printed_text="" empty="1"/>
<zone rect="305,178,312,186" printed_text="" empty="1"/>
<zone rect="305,186,312,194" printed_text="" empty="1"/>
<zone rect="305,194,312,203" printed_text="" empty="1"/>
<zone rect="305,203,312,212" printed_text="" empty="1"/>
<zone rect="305,212,312,221" printed_text="" empty="1"/>
<zone rect="305,221,312,229" printed_text="" empty="1"/>
<zone rect="305,229,312,237" printed_text="" empty="1"/>
<zone rect="305,237,312,245" printed_text="" empty="1"/>
<zone rect="305,245,312,254" printed_text="" empty="1"/>
<zone rect="305,254,312,261" printed_text="" empty="1"/>
<zone rect="305,261,312,269" printed_text="" empty="1"/>
<zone rect="305,269,312,278" printed_text="" empty="1"/>
<zone rect="305,278,312,286" printed_text="" empty="1"/>
<zone rect="305,286,312,293" printed_text="" empty="1"/>
<zone rect="305,293,312,302" printed_text="" empty="1"/>
<zone rect="305,302,312,310" printed_text="" empty="1"/>
<zone rect="305,310,312,317" printed_text="" empty="1"/>
<zone rect="305,317,312,326" printed_text="" empty="1"/>
<zone rect="305,326,312,334" printed_text="" empty="1"/>
<zone rect="305,334,312,344" printed_text="" empty="1"/>
<zone rect="305,344,312,351" printed_text="" empty="1"/>
<zone rect="305,351,312,361" printed_text="" empty="1"/>
<zone rect="305,361,312,368" printed_text="" empty="1"/>
<zone rect="305,368,312,376" printed_text="" empty="1"/>
<zone rect="305,376,312,384" printed_text="" empty="1"/>
<zone rect="305,384,312,392" printed_text="" empty="1"/>
<zone rect="305,392,312,400" printed_text="" empty="1"/>
<zone rect="305,400,312,409" printed_text="" empty="1"/>
<zone rect="305,409,312,416" printed_text="" empty="1"/>
<zone rect="305,416,312,425" printed_text="" empty="1"/>
<zone rect="305,425,312,433" printed_text="" empty="1"/>
<zone rect="305,433,312,441" printed_text="" empty="1"/>
<zone rect="305,441,312,449" printed_text="" empty="1"/>
<zone rect="305,449,312,456" printed_text="" empty="1"/>
<zone rect="305,456,312,464" printed_text="" empty="1"/>
<zone rect="305,464,312,474" printed_text="" empty="1"/>
<zone rect="305,474,312,481" printed_text="" empty="1"/>
<zone rect="305,481,312,490" printed_text="" empty="1"/>
<zone rect="305,490,312,497" printed_text="" empty="1"/>
<zone rect="305,497,312,506" printed_text="" empty="1"/>
<zone rect="305,506,312,514" printed_text="" empty="1"/>
<zone rect="305,514,312,522" printed_text="" empty="1"/>
<zone rect="305,522,312,530" printed_text="" empty="1"/>
<zone rect="305,530,312,538" printed_text="" empty="1"/>
<zone rect="305,538,312,545" printed_text="" empty="1"/>
<zone rect="305,545,312,554" printed_text="" empty="1"/>
<zone rect="305,554,312,561" printed_text="" empty="1"/>
<zone rect="305,561,312,569" printed_text="" empty="1"/>
<zone rect="305,569,312,579" printed_text="" empty="1"/>
<zone rect="312,161,319,169" printed_text="21"empty="0"/>
<zone rect="312,169,319,178" printed_text="" empty="1"/>
<zone rect="312,178,319,186" printed_text="" empty="1"/>
<zone rect="312,186,319,194" printed_text="" empty="1"/>
<zone rect="312,194,319,203" printed_text="" empty="1"/>
<zone rect="312,203,319,212" printed_text="" empty="1"/>
<zone rect="312,212,319,221" printed_text="" empty="1"/>
<zone rect="312,221,319,229" printed_text="" empty="1"/>
<zone rect="312,229,319,237" printed_text="" empty="1"/>
<zone rect="312,237,319,245" printed_text="" empty="1"/>
<zone rect="312,245,319,254" printed_text="" empty="1"/>
<zone rect="312,254,319,261" printed_text="" empty="1"/>
<zone rect="312,261,319,269" printed_text="" empty="1"/>
<zone rect="312,269,319,278" printed_text="" empty="1"/>
<zone rect="312,278,319,286" printed_text="" empty="1"/>
<zone rect="312,286,319,293" printed_text="" empty="1"/>
<zone rect="312,293,319,302" printed_text="" empty="1"/>
<zone rect="312,302,319,310" printed_text="" empty="1"/>
<zone rect="312,310,319,317" printed_text="" empty="1"/>
<zone rect="312,317,319,326" printed_text="" empty="1"/>
<zone rect="312,326,319,334" printed_text="" empty="1"/>
<zone rect="312,334,319,344" printed_text="" empty="1"/>
<zone rect="312,344,319,351" printed_text="" empty="1"/>
<zone rect="312,351,319,361" printed_text="" empty="1"/>
<zone rect="312,361,319,368" printed_text="" empty="1"/>
<zone rect="312,368,319,376" printed_text="" empty="1"/>
<zone rect="312,376,319,384" printed_text="" empty="1"/>
<zone rect="312,384,319,392" printed_text="" empty="1"/>
<zone rect="312,392,319,400" printed_text="" empty="1"/>
<zone rect="312,400,319,409" printed_text="" empty="1"/>

<zone rect="312,409,319,416" printed_text="" empty="1"/>
<zone rect="312,416,319,425" printed_text="" empty="1"/>
<zone rect="312,425,319,433" printed_text="" empty="1"/>
<zone rect="312,433,319,441" printed_text="" empty="1"/>
<zone rect="312,441,319,449" printed_text="" empty="1"/>
<zone rect="312,449,319,456" printed_text="" empty="1"/>
<zone rect="312,456,319,464" printed_text="" empty="1"/>
<zone rect="312,464,319,474" printed_text="" empty="1"/>
<zone rect="312,474,319,481" printed_text="" empty="1"/>
<zone rect="312,481,319,490" printed_text="" empty="1"/>
<zone rect="312,490,319,497" printed_text="" empty="1"/>
<zone rect="312,497,319,506" printed_text="" empty="1"/>
<zone rect="312,506,319,514" printed_text="" empty="1"/>
<zone rect="312,514,319,522" printed_text="" empty="1"/>
<zone rect="312,522,319,530" printed_text="" empty="1"/>
<zone rect="312,530,319,538" printed_text="" empty="1"/>
<zone rect="312,538,319,545" printed_text="" empty="1"/>
<zone rect="312,545,319,554" printed_text="" empty="1"/>
<zone rect="312,554,319,561" printed_text="" empty="1"/>
<zone rect="312,561,319,569" printed_text="" empty="1"/>
<zone rect="312,569,319,579" printed_text="" empty="1"/>
<zone rect="319,161,329,169" printed_text="22"empty="0"/>
<zone rect="319,169,329,178" printed_text="" empty="1"/>
<zone rect="319,178,329,186" printed_text="" empty="1"/>
<zone rect="319,186,329,194" printed_text="" empty="1"/>
<zone rect="319,194,329,203" printed_text="" empty="1"/>
<zone rect="319,203,329,212" printed_text="" empty="1"/>
<zone rect="319,212,329,221" printed_text="" empty="1"/>
<zone rect="319,221,329,229" printed_text="" empty="1"/>
<zone rect="319,229,329,237" printed_text="" empty="1"/>
<zone rect="319,237,329,245" printed_text="" empty="1"/>
<zone rect="319,245,329,254" printed_text="" empty="1"/>
<zone rect="319,254,329,261" printed_text="" empty="1"/>
<zone rect="319,261,329,269" printed_text="" empty="1"/>
<zone rect="319,269,329,278" printed_text="" empty="1"/>
<zone rect="319,278,329,286" printed_text="" empty="1"/>
<zone rect="319,286,329,293" printed_text="" empty="1"/>
<zone rect="319,293,329,302" printed_text="" empty="1"/>
<zone rect="319,302,329,310" printed_text="" empty="1"/>
<zone rect="319,310,329,317" printed_text="" empty="1"/>
<zone rect="319,317,329,326" printed_text="" empty="1"/>
<zone rect="319,326,329,334" printed_text="" empty="1"/>
<zone rect="319,334,329,344" printed_text="" empty="1"/>
<zone rect="319,344,329,351" printed_text="" empty="1"/>
<zone rect="319,351,329,361" printed_text="" empty="1"/>
<zone rect="319,361,329,368" printed_text="" empty="1"/>
<zone rect="319,368,329,376" printed_text="" empty="1"/>
<zone rect="319,376,329,384" printed_text="" empty="1"/>
<zone rect="319,384,329,392" printed_text="" empty="1"/>
<zone rect="319,392,329,400" printed_text="" empty="1"/>
<zone rect="319,400,329,409" printed_text="" empty="1"/>
<zone rect="319,409,329,416" printed_text="" empty="1"/>
<zone rect="319,416,329,425" printed_text="" empty="1"/>
<zone rect="319,425,329,433" printed_text="" empty="1"/>
<zone rect="319,433,329,441" printed_text="" empty="1"/>
<zone rect="319,441,329,449" printed_text="" empty="1"/>
<zone rect="319,449,329,456" printed_text="" empty="1"/>
<zone rect="319,456,329,464" printed_text="" empty="1"/>
<zone rect="319,464,329,474" printed_text="" empty="1"/>
<zone rect="319,474,329,481" printed_text="" empty="1"/>
<zone rect="319,481,329,490" printed_text="" empty="1"/>
<zone rect="319,490,329,497" printed_text="" empty="1"/>
<zone rect="319,497,329,506" printed_text="" empty="1"/>
<zone rect="319,506,329,514" printed_text="" empty="1"/>
<zone rect="319,514,329,522" printed_text="" empty="1"/>
<zone rect="319,522,329,530" printed_text="" empty="1"/>
<zone rect="319,530,329,538" printed_text="" empty="1"/>
<zone rect="319,538,329,545" printed_text="" empty="1"/>
<zone rect="319,545,329,554" printed_text="" empty="1"/>
<zone rect="319,554,329,561" printed_text="" empty="1"/>
<zone rect="319,561,329,569" printed_text="" empty="1"/>
<zone rect="319,569,329,579" printed_text="" empty="1"/>
<zone rect="329,161,336,169" printed_text="23"empty="0"/>
<zone rect="329,169,336,178" printed_text="" empty="1"/>
<zone rect="329,178,336,186" printed_text="" empty="1"/>
<zone rect="329,186,336,194" printed_text="" empty="1"/>
<zone rect="329,194,336,203" printed_text="" empty="1"/>
<zone rect="329,203,336,212" printed_text="" empty="1"/>
<zone rect="329,212,336,221" printed_text="" empty="1"/>
<zone rect="329,221,336,229" printed_text="" empty="1"/>
<zone rect="329,229,336,237" printed_text="" empty="1"/>
<zone rect="329,237,336,245" printed_text="" empty="1"/>
<zone rect="329,245,336,254" printed_text="" empty="1"/>
<zone rect="329,254,336,261" printed_text="" empty="1"/>
<zone rect="329,261,336,269" printed_text="" empty="1"/>
<zone rect="329,269,336,278" printed_text="" empty="1"/>
<zone rect="329,278,336,286" printed_text="" empty="1"/>
<zone rect="329,286,336,293" printed_text="" empty="1"/>
<zone rect="329,293,336,302" printed_text="" empty="1"/>
<zone rect="329,302,336,310" printed_text="" empty="1"/>
<zone rect="329,310,336,317" printed_text="" empty="1"/>
<zone rect="329,317,336,326" printed_text="" empty="1"/>
<zone rect="329,326,336,334" printed_text="" empty="1"/>
<zone rect="329,334,336,344" printed_text="" empty="1"/>
<zone rect="329,344,336,351" printed_text="" empty="1"/>
<zone rect="329,351,336,361" printed_text="" empty="1"/>
<zone rect="329,361,336,368" printed_text="" empty="1"/>
<zone rect="329,368,336,376" printed_text="" empty="1"/>
<zone rect="329,376,336,384" printed_text="" empty="1"/>
<zone rect="329,384,336,392" printed_text="" empty="1"/>
<zone rect="329,392,336,400" printed_text="" empty="1"/>
<zone rect="329,400,336,409" printed_text="" empty="1"/>
<zone rect="329,409,336,416" printed_text="" empty="1"/>
<zone rect="329,416,336,425" printed_text="" empty="1"/>
<zone rect="329,425,336,433" printed_text="" empty="1"/>
<zone rect="329,433,336,441" printed_text="" empty="1"/>
<zone rect="329,441,336,449" printed_text="" empty="1"/>
<zone rect="329,449,336,456" printed_text="" empty="1"/>
<zone rect="329,456,336,464" printed_text="" empty="1"/>
<zone rect="329,464,336,474" printed_text="" empty="1"/>
<zone rect="329,474,336,481" printed_text="" empty="1"/>
<zone rect="329,481,336,490" printed_text="" empty="1"/>
<zone rect="329,490,336,497" printed_text="" empty="1"/>
<zone rect="329,497,336,506" printed_text="" empty="1"/>
<zone rect="329,506,336,514" printed_text="" empty="1"/>
<zone rect="329,514,336,522" printed_text="" empty="1"/>
<zone rect="329,522,336,530" printed_text="" empty="1"/>
<zone rect="329,530,336,538" printed_text="" empty="1"/>
<zone rect="329,538,336,545" printed_text="" empty="1"/>
<zone rect="329,545,336,554" printed_text="" empty="1"/>
<zone rect="329,554,336,561" printed_text="" empty="1"/>
<zone rect="329,561,336,569" printed_text="" empty="1"/>
<zone rect="329,569,336,579" printed_text="" empty="1"/>
<zone rect="336,161,375,169" printed_text="24"empty="0"/>
<zone rect="336,169,375,178" printed_text="" empty="1"/>
<zone rect="336,178,375,186" printed_text="" empty="1"/>
<zone rect="336,186,375,194" printed_text="" empty="1"/>
<zone rect="336,194,375,203" printed_text="" empty="1"/>
<zone rect="336,203,375,212" printed_text="" empty="1"/>
<zone rect="336,212,375,221" printed_text="" empty="1"/>
<zone rect="336,221,375,229" printed_text="" empty="1"/>
<zone rect="336,229,375,237" printed_text="" empty="1"/>
<zone rect="336,237,375,245" printed_text="" empty="1"/>
<zone rect="336,245,375,254" printed_text="" empty="1"/>
<zone rect="336,254,375,261" printed_text="" empty="1"/>
<zone rect="336,261,375,269" printed_text="" empty="1"/>
<zone rect="336,269,375,278" printed_text="" empty="1"/>
<zone rect="336,278,375,286" printed_text="" empty="1"/>
<zone rect="336,286,375,293" printed_text="" empty="1"/>
<zone rect="336,293,375,302" printed_text="" empty="1"/>

<zone rect="336,302,375,310" printed_text="" empty="1"/>
<zone rect="336,310,375,317" printed_text="" empty="1"/>
<zone rect="336,317,375,326" printed_text="" empty="1"/>
<zone rect="336,326,375,334" printed_text="" empty="1"/>
<zone rect="336,334,375,344" printed_text="" empty="1"/>
<zone rect="336,344,375,351" printed_text="" empty="1"/>
<zone rect="336,351,375,361" printed_text="" empty="1"/>
<zone rect="336,361,375,368" printed_text="" empty="1"/>
<zone rect="336,368,375,376" printed_text="" empty="1"/>
<zone rect="336,376,375,384" printed_text="" empty="1"/>
<zone rect="336,384,375,392" printed_text="" empty="1"/>
<zone rect="336,392,375,400" printed_text="" empty="1"/>
<zone rect="336,400,375,409" printed_text="" empty="1"/>
<zone rect="336,409,375,416" printed_text="" empty="1"/>
<zone rect="336,416,375,425" printed_text="" empty="1"/>
<zone rect="336,425,375,433" printed_text="" empty="1"/>
<zone rect="336,433,375,441" printed_text="" empty="1"/>
<zone rect="336,441,375,449" printed_text="" empty="1"/>
<zone rect="336,449,375,456" printed_text="" empty="1"/>
<zone rect="336,456,375,464" printed_text="" empty="1"/>
<zone rect="336,464,375,474" printed_text="" empty="1"/>
<zone rect="336,474,375,481" printed_text="" empty="1"/>
<zone rect="336,481,375,490" printed_text="" empty="1"/>
<zone rect="336,490,375,497" printed_text="" empty="1"/>
<zone rect="336,497,375,506" printed_text="" empty="1"/>
<zone rect="336,506,375,514" printed_text="" empty="1"/>
<zone rect="336,514,375,522" printed_text="" empty="1"/>
<zone rect="336,522,375,530" printed_text="" empty="1"/>
<zone rect="336,530,375,538" printed_text="" empty="1"/>
<zone rect="336,538,375,545" printed_text="" empty="1"/>
<zone rect="336,545,375,554" printed_text="" empty="1"/>
<zone rect="336,554,375,561" printed_text="" empty="1"/>
<zone rect="336,561,375,569" printed_text="" empty="1"/>
<zone rect="336,569,375,579" printed_text="" empty="1"/>
<zone rect="375,161,405,169" printed_text="25"empty="0"/>
<zone rect="375,169,405,178" printed_text="" empty="1"/>
<zone rect="375,178,405,186" printed_text="" empty="1"/>
<zone rect="375,186,405,194" printed_text="" empty="1"/>
<zone rect="375,194,405,203" printed_text="" empty="1"/>
<zone rect="375,203,405,212" printed_text="" empty="1"/>
<zone rect="375,212,405,221" printed_text="" empty="1"/>
<zone rect="375,221,405,229" printed_text="" empty="1"/>
<zone rect="375,229,405,237" printed_text="" empty="1"/>
<zone rect="375,237,405,245" printed_text="" empty="1"/>
<zone rect="375,245,405,254" printed_text="" empty="1"/>
<zone rect="375,254,405,261" printed_text="" empty="1"/>
<zone rect="375,261,405,269" printed_text="" empty="1"/>
<zone rect="375,269,405,278" printed_text="" empty="1"/>
<zone rect="375,278,405,286" printed_text="" empty="1"/>
<zone rect="375,286,405,293" printed_text="" empty="1"/>
<zone rect="375,293,405,302" printed_text="" empty="1"/>
<zone rect="375,302,405,310" printed_text="" empty="1"/>
<zone rect="375,310,405,317" printed_text="" empty="1"/>
<zone rect="375,317,405,326" printed_text="" empty="1"/>
<zone rect="375,326,405,334" printed_text="" empty="1"/>
<zone rect="375,334,405,344" printed_text="" empty="1"/>
<zone rect="375,344,405,351" printed_text="" empty="1"/>
<zone rect="375,351,405,361" printed_text="" empty="1"/>
<zone rect="375,361,405,368" printed_text="" empty="1"/>
<zone rect="375,368,405,376" printed_text="" empty="1"/>
<zone rect="375,376,405,384" printed_text="" empty="1"/>
<zone rect="375,384,405,392" printed_text="" empty="1"/>
<zone rect="375,392,405,400" printed_text="" empty="1"/>
<zone rect="375,400,405,409" printed_text="" empty="1"/>
<zone rect="375,409,405,416" printed_text="" empty="1"/>
<zone rect="375,416,405,425" printed_text="" empty="1"/>
<zone rect="375,425,405,433" printed_text="" empty="1"/>
<zone rect="375,433,405,441" printed_text="" empty="1"/>
<zone rect="375,441,405,449" printed_text="" empty="1"/>
<zone rect="375,449,405,456" printed_text="" empty="1"/>
<zone rect="375,456,405,464" printed_text="" empty="1"/>
<zone rect="375,464,405,474" printed_text="" empty="1"/>
<zone rect="375,474,405,481" printed_text="" empty="1"/>
<zone rect="375,481,405,490" printed_text="" empty="1"/>
<zone rect="375,490,405,497" printed_text="" empty="1"/>
<zone rect="375,497,405,506" printed_text="" empty="1"/>
<zone rect="375,506,405,514" printed_text="" empty="1"/>
<zone rect="375,514,405,522" printed_text="" empty="1"/>
<zone rect="375,522,405,530" printed_text="" empty="1"/>
<zone rect="375,530,405,538" printed_text="" empty="1"/>
<zone rect="375,538,405,545" printed_text="" empty="1"/>
<zone rect="375,545,405,554" printed_text="" empty="1"/>
<zone rect="375,554,405,561" printed_text="" empty="1"/>
<zone rect="375,561,405,569" printed_text="" empty="1"/>
<zone rect="375,569,405,579" printed_text="" empty="1"/>
<zone rect="405,161,433,169" printed_text="26"empty="0"/>
<zone rect="405,169,433,178" printed_text="" empty="1"/>
<zone rect="405,178,433,186" printed_text="" empty="1"/>
<zone rect="405,186,433,194" printed_text="" empty="1"/>
<zone rect="405,194,433,203" printed_text="" empty="1"/>
<zone rect="405,203,433,212" printed_text="" empty="1"/>
<zone rect="405,212,433,221" printed_text="" empty="1"/>
<zone rect="405,221,433,229" printed_text="" empty="1"/>
<zone rect="405,229,433,237" printed_text="" empty="1"/>
<zone rect="405,237,433,245" printed_text="" empty="1"/>
<zone rect="405,245,433,254" printed_text="" empty="1"/>
<zone rect="405,254,433,261" printed_text="" empty="1"/>
<zone rect="405,261,433,269" printed_text="" empty="1"/>
<zone rect="405,269,433,278" printed_text="" empty="1"/>
<zone rect="405,278,433,286" printed_text="" empty="1"/>
<zone rect="405,286,433,293" printed_text="" empty="1"/>
<zone rect="405,293,433,302" printed_text="" empty="1"/>
<zone rect="405,302,433,310" printed_text="" empty="1"/>
<zone rect="405,310,433,317" printed_text="" empty="1"/>
<zone rect="405,317,433,326" printed_text="" empty="1"/>
<zone rect="405,326,433,334" printed_text="" empty="1"/>
<zone rect="405,334,433,344" printed_text="" empty="1"/>
<zone rect="405,344,433,351" printed_text="" empty="1"/>
<zone rect="405,351,433,361" printed_text="" empty="1"/>
<zone rect="405,361,433,368" printed_text="" empty="1"/>
<zone rect="405,368,433,376" printed_text="" empty="1"/>
<zone rect="405,376,433,384" printed_text="" empty="1"/>
<zone rect="405,384,433,392" printed_text="" empty="1"/>
<zone rect="405,392,433,400" printed_text="" empty="1"/>
<zone rect="405,400,433,409" printed_text="" empty="1"/>
<zone rect="405,409,433,416" printed_text="" empty="1"/>
<zone rect="405,416,433,425" printed_text="" empty="1"/>
<zone rect="405,425,433,433" printed_text="" empty="1"/>
<zone rect="405,433,433,441" printed_text="" empty="1"/>
<zone rect="405,441,433,449" printed_text="" empty="1"/>
<zone rect="405,449,433,456" printed_text="" empty="1"/>
<zone rect="405,456,433,464" printed_text="" empty="1"/>
<zone rect="405,464,433,474" printed_text="" empty="1"/>
<zone rect="405,474,433,481" printed_text="" empty="1"/>
<zone rect="405,481,433,490" printed_text="" empty="1"/>
<zone rect="405,490,433,497" printed_text="" empty="1"/>
<zone rect="405,497,433,506" printed_text="" empty="1"/>
<zone rect="405,506,433,514" printed_text="" empty="1"/>
<zone rect="405,514,433,522" printed_text="" empty="1"/>
<zone rect="405,522,433,530" printed_text="" empty="1"/>
<zone rect="405,530,433,538" printed_text="" empty="1"/>
<zone rect="405,538,433,545" printed_text="" empty="1"/>
<zone rect="405,545,433,554" printed_text="" empty="1"/>
<zone rect="405,554,433,561" printed_text="" empty="1"/>
<zone rect="405,561,433,569" printed_text="" empty="1"/>
<zone rect="405,569,433,579" printed_text="" empty="1"/>
</table>

# APPENDIX 2 - SQL STATEMENTS

```
INSERT INTO PERSON (Person_Identifier, Full_Name, Age, Gender, Occupation, Race, Family_Identifier, Birth_Identifier) (1,
        '109,455,267,478', '314,456,336,479','291,456,314,478', '505,457,637,480', '267,456,291,478', 1, 1)
INSERT INTO FAMILY (Family_Identifier) (1)
INSERT INTO MOTHER_CHILD (Mother_Identifier, Child_Identifier) (3, 1)
INSERT INTO FATHER_CHILD (Father_Identifier, Child_Identifier) (2, 1)
INSERT INTO EVENT (Event_Identifier, Location) (1, '894,460,997,483')
INSERT INTO EVENT (Event_Identifier, Location) (2, '997,460,1076,483')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (2, 2)
INSERT INTO EVENT (Event_Identifier, Location) (3, '1076,461,1153,484')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (3, 3)

INSERT INTO PERSON (Person_Identifier, Full_Name, Age, Gender, Relationship, Race, Family_Identifier, Birth_Identifier) (4,
        '109,477,267,500', '314,478,336,501', '291,478,314,501', '363,479,442,501', '267,478,291,500', 1, 4)
INSERT INTO MOTHER_CHILD (Mother_Identifier, Child_Identifier) (6, 4)
INSERT INTO FATHER_CHILD (Father_Identifier, Child_Identifier) (5, 4)
INSERT INTO EVENT (Event_Identifier, Location) (4, '894,482,997,505')
INSERT INTO EVENT (Event_Identifier, Location) (5, '997,483,1076,506')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (5, 5)
INSERT INTO EVENT (Event_Identifier, Location) (6, '1076,483,1153,506')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (6, 6)

INSERT INTO PERSON (Person_Identifier, Full_Name, Age, Gender, Relationship, Race, Family_Identifier, Birth_Identifier) (7,
        '109,499,267,521', '314,501,336,522', '291,500,314,522', '363,501,442,523', '267,500,291,522', 1, 7)
INSERT INTO MOTHER_CHILD (Mother_Identifier, Child_Identifier) (9, 7)
INSERT INTO FATHER_CHILD (Father_Identifier, Child_Identifier) (8, 7)
INSERT INTO EVENT (Event_Identifier, Location) (7, '894,504,997,527')
INSERT INTO EVENT (Event_Identifier, Location) (8, '997,505,1076,527')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (8, 8)
INSERT INTO EVENT (Event_Identifier, Location) (9, '1076,506,1153,528')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (9, 9)

INSERT INTO PERSON (Person_Identifier, Full_Name, Age, Gender, Relationship, Race, Family_Identifier, Birth_Identifier) (10,
        '109,520,267,544', '314,522,336,544', '291,522,314,544', '363,522,442,544', '267,521,291,544', 1, 10)
INSERT INTO MOTHER_CHILD (Mother_Identifier, Child_Identifier) (12, 10)
INSERT INTO FATHER_CHILD (Father_Identifier, Child_Identifier) (11, 10)
INSERT INTO EVENT (Event_Identifier, Location) (10, '894,526,997,547')
INSERT INTO EVENT (Event_Identifier, Location) (11, '997,527,1075,547')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (11, 11)
INSERT INTO EVENT (Event_Identifier, Location) (12, '1076,527,1153,547')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (12, 12)

INSERT INTO PERSON (Person_Identifier, Full_Name, Age, Gender, Relationship, Race, Family_Identifier, Birth_Identifier) (13,
        '109,543,267,566', '314,544,336,567', '291,544,314,566', '363,544,442,567', '267,544,291,566', 1, 13)
INSERT INTO MOTHER_CHILD (Mother_Identifier, Child_Identifier) (15, 13)
INSERT INTO FATHER_CHILD (Father_Identifier, Child_Identifier) (14, 13)
INSERT INTO EVENT (Event_Identifier, Location) (13, '894,546,996,571')
INSERT INTO EVENT (Event_Identifier, Location) (14, '997,547,1075,571')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (14, 14)
INSERT INTO EVENT (Event_Identifier, Location) (15, '1075,547,1152,572')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (15, 15)

INSERT INTO PERSON (Person_Identifier, Full_Name, Age, Gender, Relationship, Occupation, Race, Family_Identifier,
        Birth_Identifier) (16, '109,565,267,588', '314,566,336,588', '291,566,313,588', '363,567,441,589', '504,568,637,590',
        '267,566,291,588', 1, 16)
INSERT INTO MOTHER_CHILD (Mother_Identifier, Child_Identifier) (18, 16)
INSERT INTO FATHER_CHILD (Father_Identifier, Child_Identifier) (17, 16)
INSERT INTO EVENT (Event_Identifier, Location) (16, '894,570,996,592')
INSERT INTO EVENT (Event_Identifier, Location) (17, '996,571,1075,593')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (17, 17)
INSERT INTO EVENT (Event_Identifier, Location) (18, '1075,571,1152,593')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (18, 18)
```

```
INSERT INTO PERSON (Person_Identifier, Full_Name, Age, Gender, Occupation, Race, Family_Identifier, Birth_Identifier) (19,
        '109,587,267,609', '313,588,336,610', '291,588,313,610', '504,589,637,612', '267,588,290,609', 2, 19)
INSERT INTO FAMILY (Family_Identifier) (2)
INSERT INTO MOTHER_CHILD (Mother_Identifier, Child_Identifier) (21, 19)
INSERT INTO FATHER_CHILD (Father_Identifier, Child_Identifier) (20, 19)
INSERT INTO EVENT (Event_Identifier, Location) (19, '893,592,996,614')
INSERT INTO EVENT (Event_Identifier, Location) (20, '996,592,1075,615')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (20, 20)
INSERT INTO EVENT (Event_Identifier, Location) (21, '1075,593,1152,615')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (21, 21)

INSERT INTO PERSON (Person_Identifier, Full_Name, Age, Gender, Relationship, Race, Family_Identifier, Birth_Identifier) (22,
        '109,608,267,631', '313,610,336,632', '290,609,313,632', '363,610,441,632', '267,609,290,631', 2, 22)
INSERT INTO MOTHER_CHILD (Mother_Identifier, Child_Identifier) (24, 22)
INSERT INTO FATHER_CHILD (Father_Identifier, Child_Identifier) (23, 22)
INSERT INTO EVENT (Event_Identifier, Location) (22, '893,614,996,636')
INSERT INTO EVENT (Event_Identifier, Location) (23, '996,614,1075,637')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (23, 23)
INSERT INTO EVENT (Event_Identifier, Location) (24, '1075,615,1152,637')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (24, 24)

INSERT INTO PERSON (Person_Identifier, Full_Name, Age, Gender, Relationship, Race, Family_Identifier,
        Birth_Identifier) (25, '109,630,267,655', '313,632,336,655', '290,631,313,655', '363,632,441,655', '267,631,290,655', 2, 25)
INSERT INTO MOTHER_CHILD (Mother_Identifier, Child_Identifier) (27, 25)
INSERT INTO FATHER_CHILD (Father_Identifier, Child_Identifier) (26, 25)
INSERT INTO EVENT (Event_Identifier, Location) (25, '893,635,996,658')
INSERT INTO EVENT (Event_Identifier, Location) (26, '996,636,1075,658')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (26, 26)
INSERT INTO EVENT (Event_Identifier, Location) (27, '1075,637,1152,658')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (27, 27)

INSERT INTO PERSON (Person_Identifier, Full_Name, Gender, Relationship, Race, Family_Identifier, Birth_Identifier) (28,
        '109,654,267,675', '290,655,313,676', '363,655,441,677', '267,655,290,676', 2, 28)
INSERT INTO MOTHER_CHILD (Mother_Identifier, Child_Identifier) (30, 28)
INSERT INTO FATHER_CHILD (Father_Identifier, Child_Identifier) (29, 28)
INSERT INTO EVENT (Event_Identifier, Location, Year) (28, '893,657,996,681', '336,655,363,676')
INSERT INTO EVENT (Event_Identifier, Location) (29, '996,658,1074,682')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (29, 29)
INSERT INTO EVENT (Event_Identifier, Location) (30, '1075,658,1152,682')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (30, 30)

INSERT INTO PERSON (Person_Identifier, Full_Name, Age, Gender, Relationship, Race, Marital_Status, Family_Identifier,
        Birth_Identifier) (31, '109,674,267,697','313,676,335,697', '290,676,313,697', '363,676,441,698', '267,675,290,697',
        '456,677,471,698', 2, 31)
INSERT INTO MOTHER_CHILD (Mother_Identifier, Child_Identifier) (33, 31)
INSERT INTO FATHER_CHILD (Father_Identifier, Child_Identifier) (32, 31)
INSERT INTO EVENT (Event_Identifier, Location) (31, '893,680,996,703')
INSERT INTO EVENT (Event_Identifier, Location) (32, '996,681,1074,703')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (32, 32)
INSERT INTO EVENT (Event_Identifier, Location) (33, '1074,682,1152,704')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (33, 33)

INSERT INTO PERSON (Person_Identifier, Full_Name, Age, Gender, Relationship, Race, Family_Identifier, Birth_Identifier) (34,
        '109,695,267,719', '313,697,335,719', '290,697,313,719', '363,697,441,720', '267,697,290,719', 2, 34)
INSERT INTO MOTHER_CHILD (Mother_Identifier, Child_Identifier) (36, 34)
INSERT INTO FATHER_CHILD (Father_Identifier, Child_Identifier) (35, 34)
INSERT INTO EVENT (Event_Identifier, Location) (34, '893,702,996,725')
INSERT INTO EVENT (Event_Identifier, Location) (35, '996,703,1074,726')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (35, 35)
INSERT INTO EVENT (Event_Identifier, Location) (36, '1074,703,1151,727')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (36, 36)
```

```
INSERT INTO PERSON (Person_Identifier, Full_Name, Age, Gender, Relationship, Race, Family_Identifier,
        Birth_Identifier) (37, '109,717,267,741', '313,719,335,742', '290,719,313,742', '363,720,441,742', '267,719,290,741', 2, 37)
INSERT INTO MOTHER_CHILD (Mother_Identifier, Child_Identifier) (39, 37)
INSERT INTO FATHER_CHILD (Father_Identifier, Child_Identifier) (38, 37)
INSERT INTO EVENT (Event_Identifier, Location) (37, '893,725,995,746')
INSERT INTO EVENT (Event_Identifier, Location) (38, '996,725,1074,747')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (38, 38)
INSERT INTO EVENT (Event_Identifier, Location) (39, '1074,726,1151,747')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (39, 39)

INSERT INTO PERSON (Person_Identifier, Full_Name, Age, Gender, Race, Family_Identifier, Birth_Identifier) (40,
        '109,740,267,764', '313,742,335,764', '290,741,313,764', '267,741,290,764', 2, 40)
INSERT INTO MOTHER_CHILD (Mother_Identifier, Child_Identifier) (42, 40)
INSERT INTO FATHER_CHILD (Father_Identifier, Child_Identifier) (41, 40)
INSERT INTO EVENT (Event_Identifier, Location) (40, '893,746,995,767')
INSERT INTO EVENT (Event_Identifier, Location) (41, '995,746,1074,768')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (41, 41)
INSERT INTO EVENT (Event_Identifier, Location) (42, '1074,747,1151,768')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (42, 42)

INSERT INTO PERSON (Person_Identifier, Full_Name, Age, Gender, Relationship, Race, Marital_Status, Family_Identifier,
        Birth_Identifier) (43, '108,763,267,785','313,764,335,785', '290,764,313,785', '363,764,441,786', '267,764,290,785',
        '471,765,487,786', 2, 43)
INSERT INTO MOTHER_CHILD (Mother_Identifier, Child_Identifier) (45, 43)
INSERT INTO FATHER_CHILD (Father_Identifier, Child_Identifier) (44, 43)
INSERT INTO EVENT (Event_Identifier, Location) (43, '893,767,995,790')
INSERT INTO EVENT (Event_Identifier, Location) (44, '995,767,1074,791')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (44, 44)
INSERT INTO EVENT (Event_Identifier, Location) (45, '1074,768,1151,791')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (45, 45)

INSERT INTO PERSON (Person_Identifier, Full_Name, Age, Gender, Relationship, Occupation, Race, Family_Identifier,
        Birth_Identifier) (46, '108,783,266,805', '313,785,335,806', '290,785,312,806', '363,785,440,807', '503,786,636,809',
        '267,785,290,805', 2, 46)
INSERT INTO MOTHER_CHILD (Mother_Identifier, Child_Identifier) (48, 46)
INSERT INTO FATHER_CHILD (Father_Identifier, Child_Identifier) (47, 46)
INSERT INTO EVENT (Event_Identifier, Location) (46, '893,789,995,812')
INSERT INTO EVENT (Event_Identifier, Location) (47, '995,790,1073,813')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (47, 47)
INSERT INTO EVENT (Event_Identifier, Location) (48, '1074,791,1151,814')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (48, 48)

INSERT INTO PERSON (Person_Identifier, Full_Name, Age, Gender, Relationship, Occupation, Race, Family_Identifier,
        Birth_Identifier) (49, '108,804,266,828', '312 ,806,335,828', '290,805,312,828', '362,806,440,829', '503,808,636,831',
        '266,805,289,828', 2, 49)
INSERT INTO MOTHER_CHILD (Mother_Identifier, Child_Identifier) (51, 49)
INSERT INTO FATHER_CHILD (Father_Identifier, Child_Identifier) (50, 49)
INSERT INTO EVENT (Event_Identifier, Location) (49, '892,811,995,834')
INSERT INTO EVENT (Event_Identifier, Location) (50, '995,812,1073,835')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (50, 50)
INSERT INTO EVENT (Event_Identifier, Location) (51, '1073,813,1151,835')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (51, 51)

INSERT INTO PERSON (Person_Identifier, Full_Name, Age, Gender, Occupation, Race, Marital_Status, Family_Identifier,
        Birth_Identifier) (52, '108,826,266,850', '312,828,335,851', '289,828,312,850', '503,830,636,853', '266,828,289,850',
        '440,829,455,852', 3, 52)
INSERT INTO FAMILY (Family_Identifier) (3)
INSERT INTO MOTHER_CHILD (Mother_Identifier, Child_Identifier) (54, 52)
INSERT INTO FATHER_CHILD (Father_Identifier, Child_Identifier) (53, 52)
INSERT INTO EVENT (Event_Identifier, Location) (52, '892,833,995,857')
INSERT INTO EVENT (Event_Identifier, Location) (53, '995,834,1073,857')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (53, 53)
INSERT INTO EVENT (Event_Identifier, Location) (54, '1073,835,1151,858')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (54, 54)
```

INSERT INTO PERSON (Person_Identifier, Full_Name, Age, Gender, Occupation, Race, Marital_Status, Family_Identifier, Birth_Identifier) (55, '108,848,266,872', '312,850,335,873', '289,850,312,873', '503,852,636,875', '266,850,289,872', '455,852,471,874', 4, 55)
INSERT INTO FAMILY (Family_Identifier) (4)
INSERT INTO MOTHER_CHILD (Mother_Identifier, Child_Identifier) (57, 55)
INSERT INTO FATHER_CHILD (Father_Identifier, Child_Identifier) (56, 55)
INSERT INTO EVENT (Event_Identifier, Location) (55, '892,856,995,877')
INSERT INTO EVENT (Event_Identifier, Location) (56, '995,857,1073,878')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (56, 56)
INSERT INTO EVENT (Event_Identifier, Location) (57, '1073,857,1150,878')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (57, 57)

INSERT INTO PERSON (Person_Identifier, Full_Name, Age, Gender, Relationship, Race, Marital_Status, Family_Identifier, Birth_Identifier) (58, '108,871,266,893','312,873,335,894', '289,872,312,894', '362,873,440,895', '266,872,289,894', '455,874,471,895', 4, 58)
INSERT INTO MOTHER_CHILD (Mother_Identifier, Child_Identifier) (60, 58)
INSERT INTO FATHER_CHILD (Father_Identifier, Child_Identifier) (59, 58)
INSERT INTO EVENT (Event_Identifier, Location) (58, '892,877,994,899')
INSERT INTO EVENT (Event_Identifier, Location) (59, '995,877,1073,899')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (59, 59)
INSERT INTO EVENT (Event_Identifier, Location) (60, '1073,878,1150,900')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (60, 60)

INSERT INTO PERSON (Person_Identifier, Full_Name, Age, Gender, Relationship, Occupation, Race, Marital_Status, Family_Identifier, Birth_Identifier) (61, '108,892,266,916', '312,894,335,916', '289,894,312,916', '362,894,440,917', '502,895,636,919', '266,893,289,916', '440,895,455,917', 5, 61)
INSERT INTO FAMILY (Family_Identifier) (5)
INSERT INTO MOTHER_CHILD (Mother_Identifier, Child_Identifier) (63, 61)
INSERT INTO FATHER_CHILD (Father_Identifier, Child_Identifier) (62, 61)
INSERT INTO EVENT (Event_Identifier, Location) (61, '892,898,994,922')
INSERT INTO EVENT (Event_Identifier, Location) (62, '994,899,1073,922')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (62, 62)
INSERT INTO EVENT (Event_Identifier, Location) (63, '1073,899,1150,923')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (63, 63)

INSERT INTO PERSON (Person_Identifier, Full_Name, Age, Gender, Occupation, Race, Marital_Status, Family_Identifier, Birth_Identifier) (64, '108,914,266,938', '3 12,916,334,939', '289,916,312,939', '502,918,635,941', '266,916,289,938', '440,917,455,940', 5, 64)
INSERT INTO MOTHER_CHILD (Mother_Identifier, Child_Identifier) (66, 64)
INSERT INTO FATHER_CHILD (Father_Identifier, Child_Identifier) (65, 64)
INSERT INTO EVENT (Event_Identifier, Location) (64, '892,921,994,943')
INSERT INTO EVENT (Event_Identifier, Location) (65, '994,922,1072,944')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (65, 65)
INSERT INTO EVENT (Event_Identifier, Location) (66, '1073,922,1150,944')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (66, 66)

INSERT INTO PERSON (Person_Identifier, Full_Name, Age, Gender, Occupation, Race, Marital_Status, Family_Identifier, Birth_Identifier) (67, '108,937,266,962', '312,939,334,962', '289,938,312,962', '502,940,635,964', '266,938,289,962', '440,939,455,963', 5, 67)
INSERT INTO MOTHER_CHILD (Mother_Identifier, Child_Identifier) (69, 67)
INSERT INTO FATHER_CHILD (Father_Identifier, Child_Identifier) (68, 67)
INSERT INTO EVENT (Event_Identifier, Location) (67, '892,942,994,966')
INSERT INTO EVENT (Event_Identifier, Location) (68, '994,943,1072,966')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (68, 68)
INSERT INTO EVENT (Event_Identifier, Location) (69, '1072,944,1150,966')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (69, 69)

INSERT INTO PERSON (Person_Identifier, Full_Name, Age, Gender, Occupation, Race, Marital_Status, Family_Identifier, Birth_Identifier) (70, '108,961,266,983', '312,962,334,984', '289,962,312,984', '502,963,635,986', '266,962,289,983', '455,963,470,985', 6, 70)
INSERT INTO FAMILY (Family_Identifier) (6)
INSERT INTO MOTHER_CHILD (Mother_Identifier, Child_Identifier) (72, 70)
INSERT INTO FATHER_CHILD (Father_Identifier, Child_Identifier) (71, 70)
INSERT INTO EVENT (Event_Identifier, Location) (70, '892,965,994,988')
INSERT INTO EVENT (Event_Identifier, Location) (71, '994,966,1072,989')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (71, 71)
INSERT INTO EVENT (Event_Identifier, Location) (72, '1072,966,1150,989')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (72, 72)

```
INSERT INTO PERSON (Person_Identifier, Full_Name, Age, Gender, Relationship, Race, Marital_Status, Family_Identifier,
        Birth_Identifier) (73, '108,982,266,1005', '312,984,334,1005', '289,983,311,1005', '362,984,439,1006', '266,983,289,1005',
        '455,985,470,1006', 6, 73)
INSERT INTO MOTHER_CHILD (Mother_Identifier, Child_Identifier) (75, 73)
INSERT INTO FATHER_CHILD (Father_Identifier, Child_Identifier) (74, 73)
INSERT INTO EVENT (Event_Identifier, Location) (73, '892,987,994,1009')
INSERT INTO EVENT (Event_Identifier, Location) (74, '994,988,1072,1010')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (74, 74)
INSERT INTO EVENT (Event_Identifier, Location) (75, '1072,989,1150,1010')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (75, 75)

INSERT INTO PERSON (Person_Identifier, Full_Name, Age, Gender, Relationship, Occupation, Race, Marital_Status,
        Family_Identifier, Birth_Identifier) (76, '108,1004,266,1027', '311,1005,334,1028', '289,1005,311,1027',
        '362,1006,439,1028', '502,1006,635,1029', '266,1005,288,1027', '439,1006,455,1028', 6, 76)
INSERT INTO MOTHER_CHILD (Mother_Identifier, Child_Identifier) (78, 76)
INSERT INTO FATHER_CHILD (Father_Identifier, Child_Identifier) (77, 76)
INSERT INTO EVENT (Event_Identifier, Location) (76, '891,1009,994,1032')
INSERT INTO EVENT (Event_Identifier, Location) (77, '994,1009,1072,1032')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (77, 77)
INSERT INTO EVENT (Event_Identifier, Location) (78, '1072,1010,1150,1033')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (78, 78)

INSERT INTO PERSON (Person_Identifier, Full_Name, Age, Gender, Relationship, Race, Marital_Status, Family_Identifier,
        Birth_Identifier) (79, '108,1026,266,1047', '311,1027,334,1048', '288,1027,311,1048', '362,1028,439,1048',
        '266,1027,288,1047', '439,1028,455,1049', 6, 79)
INSERT INTO MOTHER_CHILD (Mother_Identifier, Child_Identifier) (81, 79)
INSERT INTO FATHER_CHILD (Father_Identifier, Child_Identifier) (80, 79)
INSERT INTO EVENT (Event_Identifier, Location) (79, '891,1031,993,1052')
INSERT INTO EVENT (Event_Identifier, Location) (80, '994,1032,1072,1053')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (80, 80)
INSERT INTO EVENT (Event_Identifier, Location) (81, '1072,1032,1149,1053')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (81, 81)

INSERT INTO PERSON (Person_Identifier, Full_Name, Age, Gender, Relationship, Race, Marital_Status, Family_Identifier,
        Birth_Identifier) (82, '108,1046,266,1071', '311,1048,334,1071', '288,1047,311,1071', '362,1048,439,1071',
        '266,1047,288,1071', '439,1048,455,1071', 6, 82)
INSERT INTO MOTHER_CHILD (Mother_Identifier, Child_Identifier) (84, 82)
INSERT INTO FATHER_CHILD (Father_Identifier, Child_Identifier) (83, 82)
INSERT INTO EVENT (Event_Identifier, Location) (82, '891,1052,993,1074')
INSERT INTO EVENT (Event_Identifier, Location) (83, '993,1052,1072,1074')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (83, 83)
INSERT INTO EVENT (Event_Identifier, Location) (84, '1072,1053,1149,1074')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (84, 84)

INSERT INTO PERSON (Person_Identifier, Full_Name, Age, Occupation, Race, Marital_Status, Family_Identifier, Birth_Identifier)
        (85, '108,1070,266,1090', '311,1071,334,1091', '501,1072,635,1093', '266,1071,288,1090', '439,1071,455,1092', 7, 85)
INSERT INTO FAMILY (Family_Identifier) (7)
INSERT INTO MOTHER_CHILD (Mother_Identifier, Child_Identifier) (87, 85)
INSERT INTO FATHER_CHILD (Father_Identifier, Child_Identifier) (86, 85)
INSERT INTO EVENT (Event_Identifier, Location) (85, '891,1073,993,1095')
INSERT INTO EVENT (Event_Identifier, Location) (86, '993,1074,1071,1096')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (86, 86)
INSERT INTO EVENT (Event_Identifier, Location) (87, '1072,1074,1149,1096')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (87, 87)

INSERT INTO PERSON (Person_Identifier, Full_Name, Age, Race, Marital_Status, Family_Identifier, Birth_Identifier) (88,
        '108,1089,266,1111', '311,1091,334,1112','266,1090,288,1112', '455,1092,470,1113', 8, 88)
INSERT INTO FAMILY (Family_Identifier) (8)
INSERT INTO MOTHER_CHILD (Mother_Identifier, Child_Identifier) (90, 88)
INSERT INTO FATHER_CHILD (Father_Identifier, Child_Identifier) (89, 88)
INSERT INTO EVENT (Event_Identifier, Location) (88, '891,1095,993,1117')
INSERT INTO EVENT (Event_Identifier, Location) (89, '993,1095,1071,1117')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (89, 89)
INSERT INTO EVENT (Event_Identifier, Location) (90, '1071,1096,1149,1118')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (90, 90)
```

```
INSERT INTO PERSON (Person_Identifier, Full_Name, Age, Gender, Relationship, Race, Marital_Status, Family_Identifier,
        Birth_Identifier) (91, '108,1110,266,1134', '311,1112,334,1135', '288,1112,311,1134', '362,1112,439,1136',
        '266,1111,288,1134', '455,1113,470,1136', 8, 91)
INSERT INTO MOTHER_CHILD (Mother_Identifier, Child_Identifier) (93, 91)
INSERT INTO FATHER_CHILD (Father_Identifier, Child_Identifier) (92, 91)
INSERT INTO EVENT (Event_Identifier, Location) (91, '891,1116,993,1141')
INSERT INTO EVENT (Event_Identifier, Location) (92, '993,1117,1071,1142')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (92, 92)
INSERT INTO EVENT (Event_Identifier, Location) (93, '1071,1117,1149,1142')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (93, 93)

INSERT INTO PERSON (Person_Identifier, Full_Name, Age, Relationship, Race, Family_Identifier, Birth_Identifier) (94,
        '108,1132,266,1155', '311,1134,334,1156', '362,1135,439,1157', '266,1134,288,1155', 8, 94)
INSERT INTO MOTHER_CHILD (Mother_Identifier, Child_Identifier) (96, 94)
INSERT INTO FATHER_CHILD (Father_Identifier, Child_Identifier) (95, 94)
INSERT INTO EVENT (Event_Identifier, Location, Year) (94, '891,1140,993,1162', '334,1135,362,1156')
INSERT INTO EVENT (Event_Identifier, Location) (95, '993,1141,1071,1163')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (95, 95)
INSERT INTO EVENT (Event_Identifier, Location) (96, '1071,1142,1149,1164')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (96, 96)

INSERT INTO PERSON (Person_Identifier, Full_Name, Age, Gender, Race, Marital_Status, Family_Identifier, Birth_Identifier) (97,
        '108,1154,266,1178', '311,1156,334,1178', '288,1155,311,1178', '266,1155,288,1178', '455,1157,470,1179', 9, 97)
INSERT INTO FAMILY (Family_Identifier) (9)
INSERT INTO MOTHER_CHILD (Mother_Identifier, Child_Identifier) (99, 97)
INSERT INTO FATHER_CHILD (Father_Identifier, Child_Identifier) (98, 97)
INSERT INTO EVENT (Event_Identifier, Location) (97, '891,1161,993,1183')
INSERT INTO EVENT (Event_Identifier, Location) (98, '993,1162,1071,1184')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (98, 98)
INSERT INTO EVENT (Event_Identifier, Location) (99, '1071,1163,1149,1184')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (99, 99)

INSERT INTO PERSON (Person_Identifier, Full_Name, Age, Gender, Relationship, Race, Marital_Status, Family_Identifier,
        Birth_Identifier) (100, '108,1176,266,1200', '311,1178,334,1201', '288,1178,311,1201', '362,1178,439,1202',
        '266,1178,288,1200', '455,1179,470,1202', 9, 100)
INSERT INTO MOTHER_CHILD (Mother_Identifier, Child_Identifier) (102, 100)
INSERT INTO FATHER_CHILD (Father_Identifier, Child_Identifier) (101, 100)
INSERT INTO EVENT (Event_Identifier, Location) (100, '891,1182,992,1205')
INSERT INTO EVENT (Event_Identifier, Location) (101, '993,1183,1071,1206')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (101, 101)
INSERT INTO EVENT (Event_Identifier, Location) (102, '1071,1184,1148,1207')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (102, 102)

INSERT INTO PERSON (Person_Identifier, Full_Name, Age, Gender, Relationship, Race, Marital_Status, Family_Identifier,
        Birth_Identifier) (103, '108,1199,265,1221', '311,1201,333,1222', '288,1200,310,1222', '362,1201,438,1223',
        '266,1200,287,1222', '439,1202,454,1223', 9, 103)
INSERT INTO MOTHER_CHILD (Mother_Identifier, Child_Identifier) (105, 103)
INSERT INTO FATHER_CHILD (Father_Identifier, Child_Identifier) (104, 103)
INSERT INTO EVENT (Event_Identifier, Location) (103, '891,1205,992,1227')
INSERT INTO EVENT (Event_Identifier, Location) (104, '992,1205,1070,1228')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (104, 104)
INSERT INTO EVENT (Event_Identifier, Location) (105, '1071,1206,1148,1228')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (105, 105)

INSERT INTO PERSON (Person_Identifier, Full_Name, Age, Gender, Relationship, Race, Marital_Status, Family_Identifier,
        Birth_Identifier) (106, '107,1220,265,1243', '310,1222,333,1243', '287,1222,310,1243', '361,1222,438,1244',
        '265,1221,287,1243', '438,1223,454,1244', 9, 106)
INSERT INTO MOTHER_CHILD (Mother_Identifier, Child_Identifier) (108, 106)
INSERT INTO FATHER_CHILD (Father_Identifier, Child_Identifier) (107, 106)
INSERT INTO EVENT (Event_Identifier, Location) (106, '890,1226,992,1249')
INSERT INTO EVENT (Event_Identifier, Location) (107, '992,1227,1070,1249')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (107, 107)
INSERT INTO EVENT (Event_Identifier, Location) (108, '1070,1228,1148,1250')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (108, 108)
```

```
INSERT INTO PERSON (Person_Identifier, Full_Name, Age, Gender, Race, Family_Identifier, Birth_Identifier) (109,
        '107,1241,265,1265', '310,1243,333,1266', '287,1243,310,1266', '265,1243,287,1265', 9, 109)
INSERT INTO MOTHER_CHILD (Mother_Identifier, Child_Identifier) (111, 109)
INSERT INTO FATHER_CHILD (Father_Identifier, Child_Identifier) (110, 109)
INSERT INTO EVENT (Event_Identifier, Location) (109, '890,1248,992,1270')
INSERT INTO EVENT (Event_Identifier, Location) (110, '992,1249,1070,1271')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (110, 110)
INSERT INTO EVENT (Event_Identifier, Location) (111, '1070,1249,1148,1271')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (111, 111)

INSERT INTO PERSON (Person_Identifier, Full_Name, Age, Gender, Race, Marital_Status, Family_Identifier, Birth_Identifier)
        (112, '107,1264,265,1287', '310,1266,3 33,1288', '287,1265,310,1288', '265,1265,287,1287', '454,1267,469,1289', 10, 112)
INSERT INTO FAMILY (Family_Identifier) (10)
INSERT INTO MOTHER_CHILD (Mother_Identifier, Child_Identifier) (114, 112)
INSERT INTO FATHER_CHILD (Father_Identifier, Child_Identifier) (113, 112)
INSERT INTO EVENT (Event_Identifier, Location) (112, '890,1270,992,1292')
INSERT INTO EVENT (Event_Identifier, Location) (113, '992,1270,1070,1293')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (113, 113)
INSERT INTO EVENT (Event_Identifier, Location) (114, '1070,1271,1148,1293')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (114, 114)

INSERT INTO PERSON (Person_Identifier, Full_Name, Age, Gender, Relationship, Race, Marital_Status, Family_Identifier,
        Birth_Identifier) (115, '107,1286,265,1307', '310,1288,333,1307', '287,1287,310,1307', '361,1288,438,1308',
        '265,1287,287,1307', '454,1288,469,1308', 10, 115)
INSERT INTO MOTHER_CHILD (Mother_Identifier, Child_Identifier) (117, 115)
INSERT INTO FATHER_CHILD (Father_Identifier, Child_Identifier) (116, 115)
INSERT INTO EVENT (Event_Identifier, Location) (115, '890,1291,992,1313')
INSERT INTO EVENT (Event_Identifier, Location) (116, '992,1292,1070,1313')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (116, 116)
INSERT INTO EVENT (Event_Identifier, Location) (117, '1070,1293,1148,1314')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (117, 117)

INSERT INTO PERSON (Person_Identifier, Full_Name, Age, Gender, Relationship, Family_Identifier, Birth_Identifier) (118,
        '107,1305,265,1329', '310,1307,333,1329', '287,1307,310,1329', '361,1308,438,1330', 10, 118)
INSERT INTO MOTHER_CHILD (Mother_Identifier, Child_Identifier) (120, 118)
INSERT INTO FATHER_CHILD (Father_Identifier, Child_Identifier) (119, 118)
INSERT INTO EVENT (Event_Identifier, Location) (118, '890,1312,992,1335')
INSERT INTO EVENT (Event_Identifier, Location) (119, '992,1313,1070,1336')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (119, 119)
INSERT INTO EVENT (Event_Identifier, Location) (120, '1070,1313,1148,1337')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (120, 120)

INSERT INTO PERSON (Person_Identifier, Full_Name, Age, Gender, Relationship, Race, Marital_Status, Family_Identifier,
        Birth_Identifier) (121, '107,1327,265,1353', '310,1329,333,1353', '287,1329,310,1353', '361,1330,438,1354',
        '265,1329,287,1353', '438,1330,454,1354', 10, 121)
INSERT INTO MOTHER_CHILD (Mother_Identifier, Child_Identifier) (123, 121)
INSERT INTO FATHER_CHILD (Father_Identifier, Child_Identifier) (122, 121)
INSERT INTO EVENT (Event_Identifier, Location) (121, '890,1334,992,1358')
INSERT INTO EVENT (Event_Identifier, Location) (122, '992,1335,1069,1359')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (122, 122)
INSERT INTO EVENT (Event_Identifier, Location) (123, '1070,1336,1148,1359')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (123, 123)

INSERT INTO PERSON (Person_Identifier, Full_Name, Age, Gender, Race, Marital_Status, Family_Identifier, Birth_Identifier)
        (124, '107,1351,265,1374', '310,1353,333,1375', '287,1353,310,1374', '265,1353,287,1374', '438,1354,454,1375', 11, 124)
INSERT INTO FAMILY (Family_Identifier) (11)
INSERT INTO MOTHER_CHILD (Mother_Identifier, Child_Identifier) (126, 124)
INSERT INTO FATHER_CHILD (Father_Identifier, Child_Identifier) (125, 124)
INSERT INTO EVENT (Event_Identifier, Location) (124, '890,1357,991,1379')
INSERT INTO EVENT (Event_Identifier, Location) (125, '992,1358,1069,1379')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (125, 125)
INSERT INTO EVENT (Event_Identifier, Location) (126, '1069,1359,1147,1380')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (126, 126)
```

INSERT INTO PERSON (Person_Identifier, Full_Name, Age, Gender, Race, Marital_Status, Family_Identifier, Birth_Identifier)
(127, '107,1373,265,1396', '310,1374,333,1396', '287,1374,310,1396', '265,1374,287,1396', '484,1376,499,1397', 12, 127)
INSERT INTO FAMILY (Family_Identifier) (12)
INSERT INTO MOTHER_CHILD (Mother_Identifier, Child_Identifier) (129, 127)
INSERT INTO FATHER_CHILD (Father_Identifier, Child_Identifier) (128, 127)
INSERT INTO EVENT (Event_Identifier, Location) (127, '890,1378,991,1400')
INSERT INTO EVENT (Event_Identifier, Location) (128, '991,1379,1069,1400')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (128, 128)
INSERT INTO EVENT (Event_Identifier, Location) (129, '1069,1379,1147,1401')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (129, 129)

INSERT INTO PERSON (Person_Identifier, Full_Name, Age, Gender, Relationship, Marital_Status, Family_Identifier,
Birth_Identifier) (130, '107,1395,265,1416', '310,1396,333,1417', '287,1396,310,1417', '361,1396,438,1418',
'483,1397,499,1418', 12, 130)
INSERT INTO MOTHER_CHILD (Mother_Identifier, Child_Identifier) (132, 130)
INSERT INTO FATHER_CHILD (Father_Identifier, Child_Identifier) (131, 130)
INSERT INTO EVENT (Event_Identifier, Location) (130, '890,1399,991,1421')
INSERT INTO EVENT (Event_Identifier, Location) (131, '991,1400,1069,1422')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (131, 131)
INSERT INTO EVENT (Event_Identifier, Location) (132, '1069,1400,1147,1422')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (132, 132)

INSERT INTO PERSON (Person_Identifier, Full_Name, Age, Gender, Race, Marital_Status, Family_Identifier, Birth_Identifier)
(133, '107,1415,265,1437', '310,1417,333,1437', '287,1416,309,1437', '265,1416,286,1437', '483,1418,499,1438', 13, 133)
INSERT INTO FAMILY (Family_Identifier) (13)
INSERT INTO MOTHER_CHILD (Mother_Identifier, Child_Identifier) (135, 133)
INSERT INTO FATHER_CHILD (Father_Identifier, Child_Identifier) (134, 133)
INSERT INTO EVENT (Event_Identifier, Location) (133, '890,1421,991,1441')
INSERT INTO EVENT (Event_Identifier, Location) (134, '991,1421,1069,1442')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (134, 134)
INSERT INTO EVENT (Event_Identifier, Location) (135, '1069,1422,1147,1442')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (135, 135)

INSERT INTO PERSON (Person_Identifier, Full_Name, Age, Gender, Relationship, Race, Marital_Status, Family_Identifier,
Birth_Identifier) (136, '107,1436,265,1458', '309,1437,333,1458', '286,1437,309,1458', '361,1438,437,1459',
'265,1437,286,1458', '483,1438,499,1459', 13, 136)
INSERT INTO MOTHER_CHILD (Mother_Identifier, Child_Identifier) (138, 136)
INSERT INTO FATHER_CHILD (Father_Identifier, Child_Identifier) (137, 136)
INSERT INTO EVENT (Event_Identifier, Location) (136, '889,1441,991,1462')
INSERT INTO EVENT (Event_Identifier, Location) (137, '991,1441,1069,1462')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (137, 137)
INSERT INTO EVENT (Event_Identifier, Location) (138, '1069,1442,1147,1462')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (138, 138)

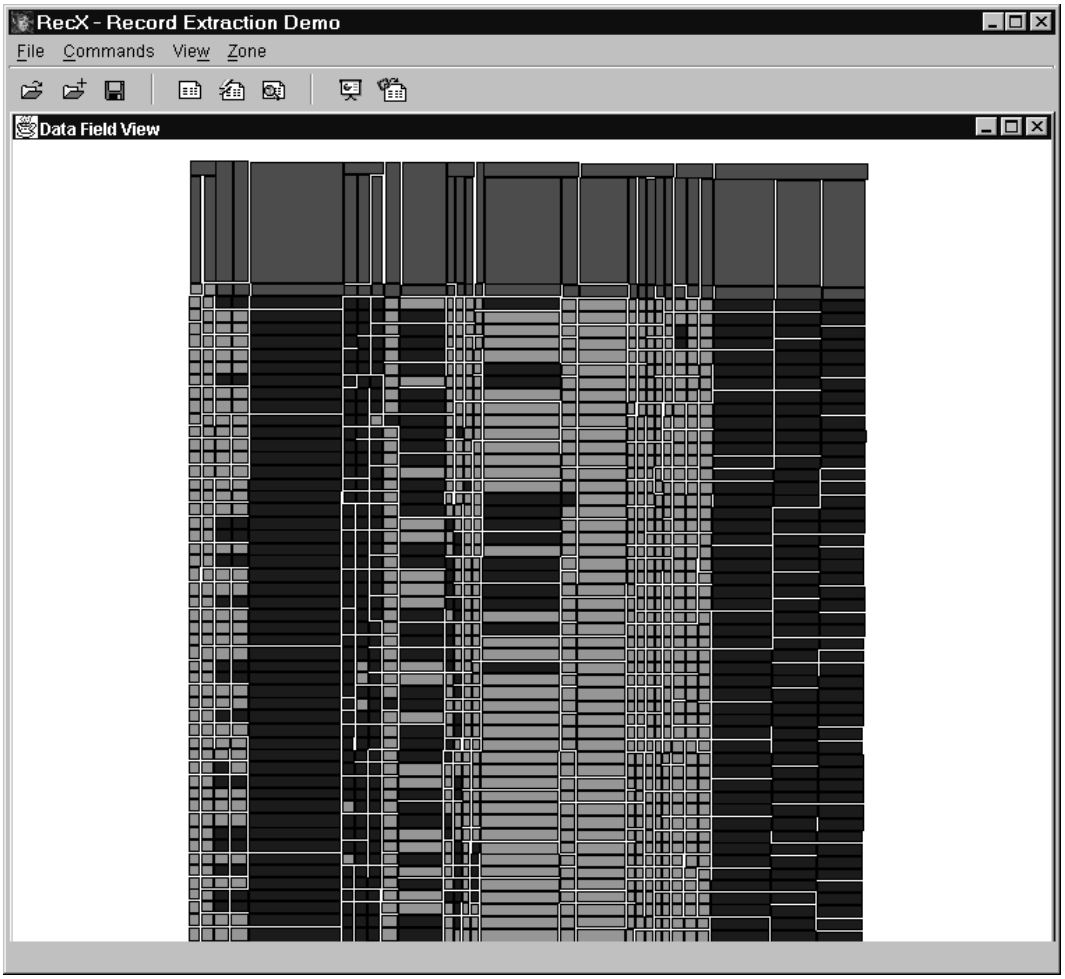
INSERT INTO PERSON (Person_Identifier, Full_Name, Age, Gender, Race, Marital_Status, Family_Identifier, Birth_Identifier)
(139, '107,1457,265,1478', '309,1458,332,1479', '286,1458,309,1478', '265,1458,286,1478', '483,1459,499,1480', 14, 139)
INSERT INTO FAMILY (Family_Identifier) (14)
INSERT INTO MOTHER_CHILD (Mother_Identifier, Child_Identifier) (141, 139)
INSERT INTO FATHER_CHILD (Father_Identifier, Child_Identifier) (140, 139)
INSERT INTO EVENT (Event_Identifier, Location) (139, '889,1461,991,1483')
INSERT INTO EVENT (Event_Identifier, Location) (140, '991,1462,1068,1483')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (140, 140)
INSERT INTO EVENT (Event_Identifier, Location) (141, '1069,1462,1147,1484')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (141, 141)

INSERT INTO PERSON (Person_Identifier, Full_Name, Age, Gender, Race, Marital_Status, Family_Identifier,
Birth_Identifier) (142, '107,1477,265,1499', '309,1478,332,1500', '286,1478,309,1500', '265,1478,286,1500',
'454,1479,468,1501', 15, 142)
INSERT INTO FAMILY (Family_Identifier) (15)
INSERT INTO MOTHER_CHILD (Mother_Identifier, Child_Identifier) (144, 142)
INSERT INTO FATHER_CHILD (Father_Identifier, Child_Identifier) (143, 142)
INSERT INTO EVENT (Event_Identifier, Location) (142, '889,1482,991,1505')
INSERT INTO EVENT (Event_Identifier, Location) (143, '991,1483,1068,1506')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (143, 143)
INSERT INTO EVENT (Event_Identifier, Location) (144, '1068,1483,1147,1506')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (144, 144)

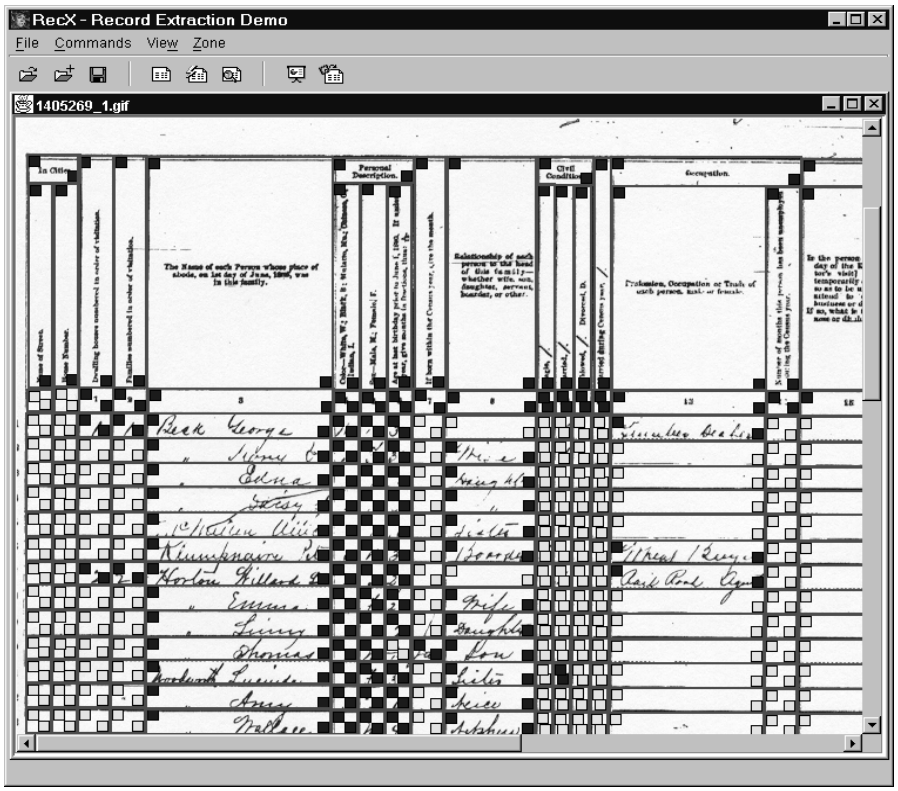
```
INSERT INTO PERSON (Person_Identifier, Full_Name, Age, Gender, Relationship, Race, Marital_Status, Family_Identifier,
        Birth_Identifier) (145, '107,1498,265,1521', '309,1500,332,1522', '286,1500,309,1522', '361,1500,437,1522',
        '265,1499,286,1521', '454,1501,468,1523', 15, 145)
INSERT INTO MOTHER_CHILD (Mother_Identifier, Child_Identifier) (147, 145)
INSERT INTO FATHER_CHILD (Father_Identifier, Child_Identifier) (146, 145)
INSERT INTO EVENT (Event_Identifier, Location) (145, '889,1504,990,1526')
INSERT INTO EVENT (Event_Identifier, Location) (146, '991,1505,1068,1527')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (146, 146)
INSERT INTO EVENT (Event_Identifier, Location) (147, '1068,1506,1146,1527')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (147, 147)

INSERT INTO PERSON (Person_Identifier, Full_Name, Age, Gender, Relationship, Race, Family_Identifier, Birth_Identifier) (148,
'107,1520,265,1546', '309,1522,332
,1547', '286,1521,309,1547', '361,1522,437,1548', '265,1521,286,1547', 15, 148)
INSERT INTO MOTHER_CHILD (Mother_Identifier, Child_Identifier) (150, 148)
INSERT INTO FATHER_CHILD (Father_Identifier, Child_Identifier) (149, 148)
INSERT INTO EVENT (Event_Identifier, Location) (148, '889,1525,990,1552')
INSERT INTO EVENT (Event_Identifier, Location) (149, '990,1526,1068,1553')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (149, 149)
INSERT INTO EVENT (Event_Identifier, Location) (150, '1068,1527,1146,1553')
INSERT INTO PERSON (Person_Identifier, Birth_Identifier) (150, 150)
```

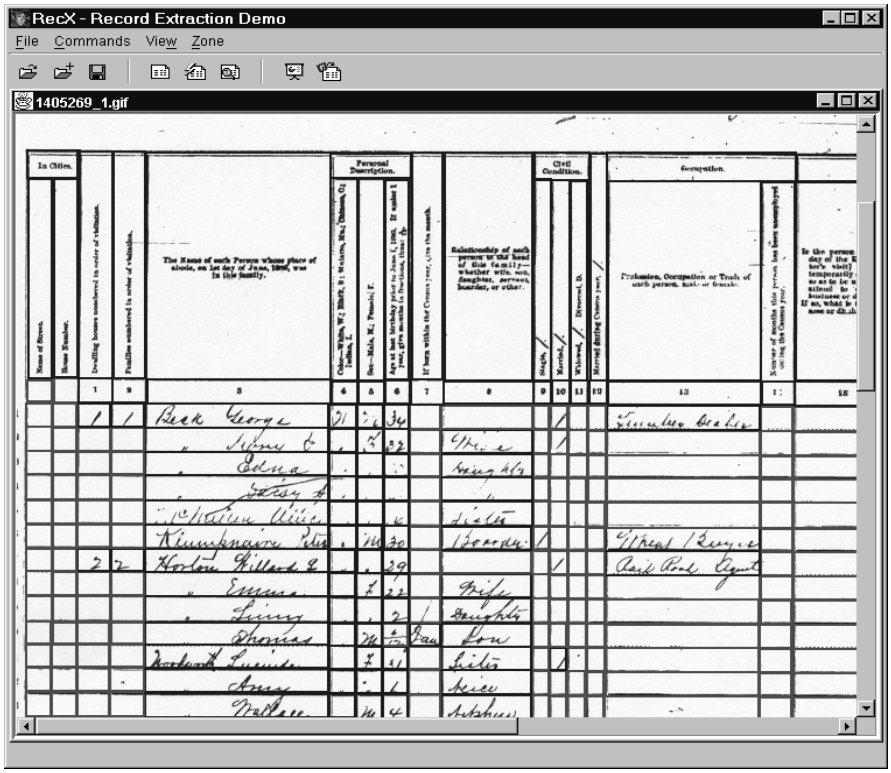
# APPENDIX 3 – SCREEN SHOTS OF TOOL

The tool shows a visual representation of the coordinates of each cell in a table.



Using the tool, an individual can correct the coordinates of each cell by adjusting the top-left and bottom-right corners of the cell.



APPENDIX 4 – INDEX OF MICROFILM ROLLS

We gathered the following 25 rolls of microfilm from the Utah Valley Regional Family History Center on the Brigham Young University campus.

## Training Set

1. Roll No. 0444770
2. Roll No. 0545677
3. Roll No. 0823344
4. Roll No. 0833737
5. Roll No. 1405269

## Test Set

1. Roll No. 0833736
2. Roll No. 0833785
3. Roll No. 0833791
4. Roll No. 0836135
5. Roll No. 0853198
6. Roll No. 0853204
7. Roll No. 0877494
8. Roll No. 1005002
9. Roll No. 1021499
10. Roll No. 1032701
11. Roll No. 1041607
12. Roll No. 1240879
13. Roll No. 1304781
14. Roll No. 1305264
15. Roll No. 1375585