The
China Biographical Database

User's Guide

Michael A. Fuller

Revised Version 3.5
October 5, 2023

© China Biographical Database Project: Harvard University, Academia Sinica, Peking University

Preface to the User's Guide

Peter K. Bol

The China Biographical Database, as a relational database, can generate biographical data in response to simple queries (who came from a certain place?) and to far more complex queries (what were the social and kinship connections among all those who entered government through the civil service examination from a certain place within a certain span of years?). Users can query CBDB through an online database (follow the links on the CBDB website, https://projects.iq.harvard.edu/cbdb. Users also can download the entire database, together with query forms and utilities for exporting data for network and spatial analysis, from the CBDB website and explore the database on any computer with Microsoft Access. We also offer a SQLite format database for quantitative researchers and Mac users. This User's Guide explains the structure and application for the downloadable, stand-alone database.

CBDB is a relational database. It categorizes and codes many different aspects of the life histories of men and women in China's past. In using it, there are several considerations that one should bear in mind when reading the Users' Guide's presentation of the specific details of the database, its design, and its use.

A way of thinking about people in context. CBDB is a way of modeling life histories; it is also a way of thinking about how to organize information. The subject of the database is people in society, but we treat people as entities that have relationships to their kin and their social associations, to places where they resided and worked, to times when they lived and moments when they acted, to names they were given and adopted, to books they wrote, to ways in which they entered government or other institutions, and to the modes in which they distinguished themselves from others. In contrast to the narrative of a life, CBDB sees people as entities defined by webs of relationships that can be quantified and analyzed.

Temporal scope. Over ninety percent of CBDB data pertains to the period from the Tang dynasty (618-907) into the early 20th c. As of this writing 1, 2019 it had data on about 472,000 figures with well over 100,000 more in preparation; further data on figures already in the database are frequently added. Tables and trees of place names and official titles will need to be expanded as we incorporate figures from earlier periods.

Factoids versus facts. Like prosopographical databases for other parts of the world, CBDB for the most part deals in "factoids," the assertions of a fact (such as "Su Shi was a person from Meishan") found in the historical sources it references. It relates these assertions, including contradictory assertions when they appear, rather than judging their reliability. However, it does not treat all sources as equal.

Principal sources. CBDB began with research conducted by the late Robert Hartwell focused on the middle period of China's history. Since then, it has been comprehensively incorporating data from published indices, such as Wang Deyi's revised *Index to Biographical Sources for Song Figures* and similar works; from online databases, such as the Name Authority Database of the Ming Qing Archive at Academia Sinica, the Tang Knowledge Base at Kyoto University and the Ming Qing Women's Writings Database directed by Grace Fong at McGill University; from studies of text sources such as collections of epitaphs (墓誌銘); from listings of local officials in local gazetteers and records of appointments; and from biographies in formal dynastic sources. Although CBDB editors at Harvard and Peking University are experimenting with mining data from other sources, it will take some time before the principal sources are exhausted.

Text-mining. The most efficient way to populate CBDB has been through the use of computational text-mining techniques to cull factoids from searchable digital texts that have been provided by the Institute of History and Philology at Academia Sinica or generated by the CBDB project itself. This began in collaboration with computer scientists on an US National Endowment for the Humanities grant. The Harvard editorial team, led first by Professor Song Chen and then Dr. Shih-pei Chen and currently by Mr. Hongsu Wang, who has had the assistance of Dr. Lik Hang Tsui, Mr. Merrick Lex Berman, and Ms Edith Enright has overseen the development of "regular expressions" appropriate to Chinese sources and the process of incorporating new data. The Peking University editorial team reviews the marked-up text, and the managers then oversee the final coding of the data for inclusion in CBDB. This process does not guarantee that all possible factoids are discovered, simply that that those included will accurately reflect the sources being mined.

Margin of error. Machines are more reliable than humans in sifting through large quantities of data but are incapable of interpretation and scholarly judgment. Errors can enter the database. The historical sources themselves can be incorrect. Editors may miss mistakes in tagging. Encoders may fail to properly disambiguate two entities with the same name. A user must always ask if the query to the database produces enough examples to ensure that the margin of error will not undermine confidence in the conclusions that are drawn. The discrepancies between the sources and the original CBDB data were significant, and considerable time was spent correcting the received data; with the adoption of computational techniques the discrepancies appear to be less than one percent. To put this in perspective: an argument based on 1000 examples of which ten are faulty is better than a finding based on ten examples of which one is erroneous.

A database is not a dictionary. CBDB can be used as a guide to biographical factoids about an individual, and it can provide more data about some aspects of a person's connections than would be found in a biographical dictionary. However, the standard

for a dictionary is complete accuracy in all aspects, whereas the expectation for a database is that the cases discovered will be useful because they are extensive in range and number.

CBDB is a joint project of the Center for Research on Ancient Chinese History at Peking University, the Institute of History and Philology at Academia Sinica, and the Fairbank Center for Chinese Studies at Harvard University. At Harvard it is housed in the Institute for Quantitative Social Sciences which provides administrative support. It is guided by a steering committee that includes scholars and collaborators from across the globe. Michael A. Fuller, the author of this User's has designed all iterations of the database.

Since 2005 CBDB has been supported by grants from Harvard University Faculty of Arts and Science and the Harvard University Asia Center, the Institute of History and Philology at Academia Sinica, the Center for Research on Ancient Chinese History at Peking University, the National Endowment for the Humanities, the Tang Research Foundation, the Tang Studies Society, the Henry Luce Foundation, the Chiang Ching-kuo Foundation, the Canadian Social Sciences and Humanities Research Council, the bequest of the late Robert Hartwell to the Harvard-Yenching Institute, and significant support from a licensing arrangement with ChineseAll.com. In China CBDB data, supplemented with extensive biographical data on twentieth century figures, is available through subscription to the Yinde System https://www.inindex.cn provided by ChineseAll.com. Over the years many scholars have visited Harvard and contributed to the project, all participants are recognized on the CBDB website

This User's Guide explains the logic of CBDB as a relational database, the structure of its contents, the primary query interfaces for getting data from the database, and installation procedures for different operating systems. Please also consult Appendix D of the User's Guide for a summary of the most recent changes to the database and to the user interface.

Table of Contents

i Preface

1 Introduction

3 Chapter 1: Relational Databases

A. Relational Database and the Organization of Complex Data

B. Rules for Structuring Data in a Relational Database

C. Relational Databases and the Interactions of Complex Data

9 Chapter 2: The Structure of CBDB

A. An Overview of the Entities in the Database

B. Details of Entities

Summary of Tables in CBDB

27 Chapter 3: CBDB Tools for Analysis

A. The Navigation Pane

B. The Browser

C. Query by Method of Entry

D. Query Associations

E. Query Office Holding

F. Query Kinship

G. Query Social Networks

H. Query Pair-wise Association

I. Query Place Association

J. Query Status

K. Query Texts and Roles

L. Looking up Data on a Group of Individuals

110 Chapter 4: Advanced Query Techniques

A. Kinship Networks for *Jinshi*-degree Holders in Putian

B. Using the Access Query Designer

124 Appendices

A. Installing the MS Access Files

B. Updating the Visual Basic Environment
C. Installing the SQLite CBDB Database on a Mac
D. Kinship Simplification Table
D. Change Log for CBDB

Introduction

The China Biographical Database (CBDB) is a relational database of biographical information for China before the early twentieth century. Through the wide range of data it collects, CBDB offers many ways to examine the lives of past individuals and groups. While CBDB provides detailed information about people and can serve as a biographical dictionary, its more powerful use is as a tool for *prosopography*, the study of the lives of groups of people:

> 'Prosopography' is the investigation of the common background characteristics of a group of actors in history by means of a collective study of their lives. The method employed is to establish a universe to be studied, and then to ask a set of uniform questions – about birth and death, marriage and family, social origins and inherited economic position, place of residence, education, amount and source of personal wealth, occupation, religion, experience of office and so on. The various types of information about the individuals in the universe are then juxtaposed and combined, and are examined for significant variables. They are tested both for internal correlations and for correlations with other forms of behaviour or action. (L. Stone, 'Prosopography', in F. Gilbert and S. Graubard eds., *Historical Studies Today* (New York, 1972)

CBDB also supports a second approach to analyzing the lives of large numbers of people. Social network analysis (SNA) has been a tool for studying group structure in the social sciences for many decades. Scholars, however, also have applied its techniques to data derived from historical documents. Charles Wetherell describes the project of *historical social network analysis* (HSNA):

> Conceptualizing community as collections of personal relationships … provides historians with a blueprint for evaluating when, how and why people in the past used kin and non-kin in the course of their lives. The findings of social network analysts that people need and seek emotional and economic support of different kinds, from different kinds of people, suggest new analytical imperatives. It is not enough now to look solely at how people used kin in times of crisis. Rather, historians need to pursue how people in the past used the kin and friends they had, for different things, throughout the life course, and in the context of the opportunities they enjoyed and the constraints they faced courtesy of demography and culture. Other approaches might be applied to the problem, but HSNA contains the essential perspectives that cannot only advance the debate, but also help historians to meet Tilly's challenge to connect the lives of ordinary people to large-scale change in meaningful ways. (Charles Wetherell, "Historical Social Network Analysis," *International Review of Social History* 43 [1998], Supplement)

In large measure, historians have used SNA approaches on small sample populations where the relations among all the member of the group are known, but CBDB hopes to provide data on relations among individuals in very large populations where the density of relationship data is adequate to produce statistically meaningful results about patterns in the social world of China's past.

Because CBDB records information about where people lived, where they studied, where they served in office, what offices they held, who their parents were, who they married, and who they knew, all these aspects of life can be correlated for very large groups of people. We can ask if local marriage alliance were typical during a particular period or in a particular region, or for a particular level of office-holder or occupation. We can ask about kinship patterns within occupations for any slice of time and/or any region of China. We can look at regional patterns of sponsorship or partisan opposition. We can look at social, kinship and regional factors in promotions within the Buddhist monastic orders. We can ask who associated themselves with certain ideologues and teachers and where they lived. There is almost no limit to the types of questions that can be asked about the people in the database.

The challenge is how to phrase the questions in ways to which CBDB can respond. The goal of this User's Guide is to provide you with enough information about CBDB, first, to use its interface for common types of queries and then to use other tools for more advanced queries of the dataset. Information about CBDB divides into three parts: general information about relational databases, the structure of CBDB in particular (the types of data it contains), and the interface for looking at the data in CBDB.

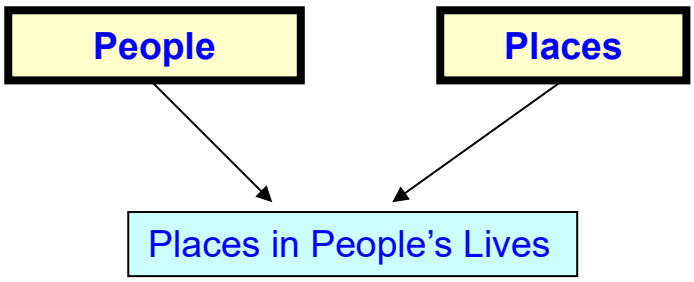
Chapter 1. Relational Databases

A. Relational Database and the Organization of Complex Data

The social historian Robert Hartwell (1932-1996), who was concerned with the kinship and social networks of Song Dynasty officials, first conceived of using a relational database to study collective biographies, and CBDB evolved out of his initial model.

Hartwell's important step was to see that he needed a powerful organizing tool to meet the challenges of the project he proposed. He wanted to look at relations between people, their kinship groups, their social networks, the offices they held, and the places with which they were associated. This is a long list, and the interactions between all of these elements grow complex and difficult to track. Hartwell realized that he could think of the interactions he saw in biographical data as relations between (1) people, (2) places, (3) a bureaucratic system, (4) kinship structures and (5) contemporary modes of social association. He built a relational database precisely to capture biographical data as the relations between these five "things." In the current version of CBDB that grew out of Hartwell's model, we have added three more aspects of social experience through which individuals defined themselves: (6) social institutions like temples, academies, etc., (7) cultural systems for attaining social distinction, and (8) the vast webs of textual production.

This structuring of relationships between *entities*, categories of "things" in the world, is what a relational database does: it allows one to capture multiform relations between complex objects that interact with one another. That is, PLACE is an entity, and under this category we can list any and all places about which we have information and in which we are interested. Similarly, under PEOPLE as yet another entity, we list all the people about whom we have biographical information. Then we can list all the interactions we care to record between people and places: where they were born, where they moved, where they were buried, and so on. We have the abstract model of relations between entities:



This abstract model, when transformed into a relational database, becomes a series of *tables* filled with data divided into *fields*:

PEOPLE

ID	Name	Dates
1	Lü Benzhong 呂本中	1084-1145
2	An Dun 安惇	1042-1101
3	Chao Buzhi 晁補之	1053-1110
4	Chen Jian(5) 陳薦	fl. 1069

PEOPLE-PLACES

Person ID	Place ID	Relation Type ID
1	1	1
1	3	2
1	2	3

PLACES

ID	Place Name
1	Jinhua 金華
2	Shouzhou 壽州
3	Kaifeng 開封

PEOPLE-PLACE TYPES

Relation Type ID	Relation Type
1	Basic Affiliation
2	Moved to
3	Ancestral addr

Note that with this arrangement of tables, there is no limit to the number of people, the number of places, or the number of types of relations between people and places.

From this example of how people and place relate to one another, we see that in relational databases there are three basic types of tables:

1. **Tables that describe the basic "entities."** (The yellow tables "People" and "Places" above) In CBDB, these include people, places, kinship term, bureaucratic structures, and so on. The fields in these tables capture the attributes of these entities that we want to know about. For people, this would include their names, birth and death dates, gender, and the like. For places ("addresses" in CBDB parlance) it would include names, the administrative levels (county, prefecture, etc.), when they were created, and so on.

2. **Tables that describe relations between basic entities.** (The blue "People-Places" table) In CBDB, these translate the relations between people and their social, physical, and cultural environment into a structured format. The fields in these tables capture the features of the relations that are considered important in describing the relationship. For instance, when a person receives a posting to serve in a bureaucratic office, in addition to the basic information of who the person was and what the office was, we also would like to know (1) where the post was, (2) if the person in fact served, and (3) when he served. Other types of entities, however, also have important and often complex relations with one another. For PLACES, for example, it would include its superior or subordinate units, and the period of validity of those relations. For OFFICES, a key relationship is where the office fits into the administrative hierarchy at any particular time.

3. **Tables that describe the *types* of relations between entities**. (The pink "People-Place types" table.) Sometimes, there can be many ways for two "things" to interact in the world, and we need to be able to be more specific in recording the details of the interaction. In the example above, people can have many different ways of being related to a place: it might be the place at which they were formally registered, the place at which they actually lived, or the place where they were buried. We can group these relations into categories to give them structure.

B. Rules for Structuring Data in a Relational Database

In databases, we try to record any particular datum only once. In the example above, the name Lü Benzhong 呂本中 appears in only one record in CBDB, in his basic entry in the table for PEOPLE entities (the table is called BIOG_MAIN). All other records that record information about Lü Benzhong refer to him by his ID number. Thus, if, for example, I mistakenly entered the name Hong Shi for 洪适 (properly romanized as Hong Kuo) because I thought that the second character was the simplified form of *shi* 適, I would need to fix the mistake in only one place. This principle of "one datum, one place" is called *normalization*. There are occasions where CBDB violates this rule in order to speed processing, but if you wish to add additional tables to your own version of CBDB, we strongly recommend that you pay attention to the goal of maintaining a normalized database.

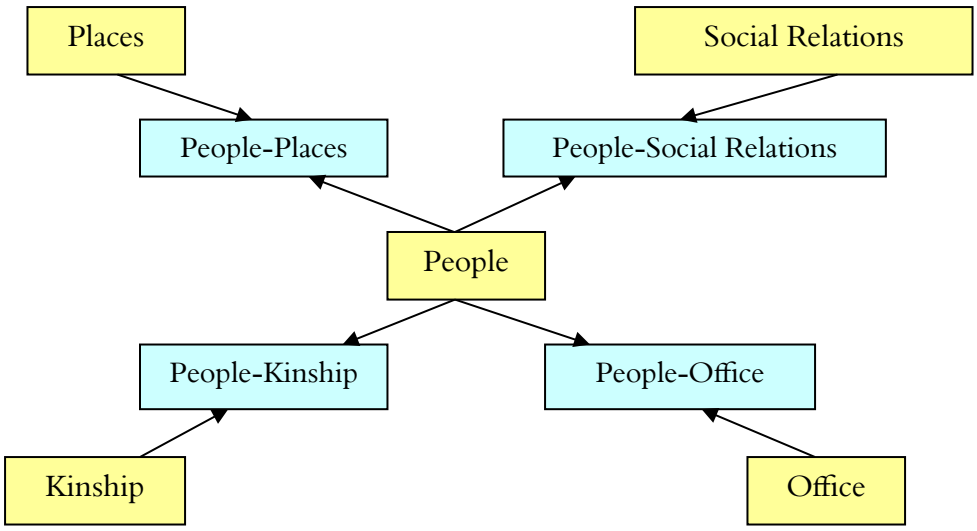
In the example of a person's relationship to places discussed above, we encounter the fact that a person can move to many different places. This is called a "one-to-many" relationship. If one were to try to represent this relationship through a simple table with rows and columns, we either could create a number of columns in the basic biographical table ("Moved to 1", "Moved to 2", and so on), or we could add all entries into a single cell. If we create several columns for "Moved to," we cannot be sure that we will not encounter an individual who moved so many times that it exceeds the number of columns we created. Moreover, every single record in the biographical table would have all of the "Moved to" cells, which would remain empty for most people. If one were to create just one column for "Moved to" information, searching through the entries in the cell for each individual would make retrieving the data very difficult. The disadvantages of these two approaches to keeping the "Moved to" data in the main table leads to the general rule: whenever we find this sort of one-to-many relationship between basic entities (here, PEOPLE and PLACES), we need a separate category of relationship like PEOPLE-PLACES (and a table to represent that relationship) to allow us to capture the interaction.

We encounter a different type of problem when we encode a book like *Record of Things at Hand*, which was edited by Zhu Xi and Lü Zuqian. Writings have a so-called "many-to-many" relationship: one book may have many authors or editors, and each of those writers may have written many books. In CBDB, as in many databases, we treat this situation as a pair of one-to-many relations between PEOPLE and WRITINGS and introduce a new category of relationship (and corresponding table), PEOPLE-WRITINGS, to capture the data.

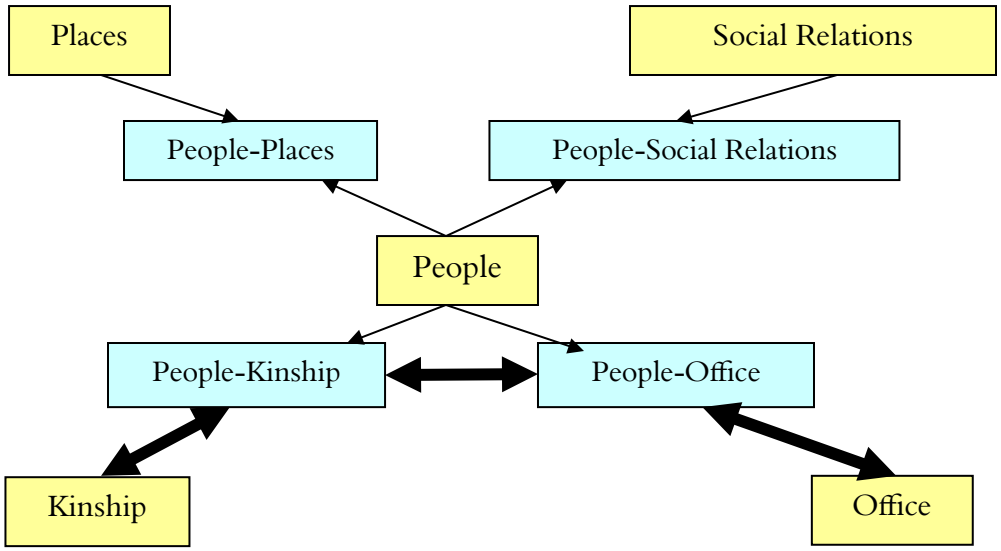
These three rules—normalize data, create new tables for one-to-many relations, and treat many-to-many like one-to-many—are important if you wish to add new data types to CBDB.

C. Relational Databases and the Interactions of Complex Data

CBDB models the interactions between people and the entities—the "things"—that shape their social world. Some of these entities are easily understood in their "thingness:" places are physical entities, and the official bureaucracy has a substantial structure in premodern Chinese society. Kinship is a bit more difficult to conceive. Anthropologists have long considered the kinship relations in a society as a structured system: some kinship ties are particularly strong, and societies are organized around these ties. People, that is, are not simply related to one another: their relationship is part of—and acquires meaning through—the kinship system of the society. "Social relations" as a "thing" is yet more abstract but follows the same principles. If one wants to establish a social relation with another person, the society sets out patterns of what relations are appropriate and significant and what relations are not. Within the system of associations that a society values, "social capital" measures how one has positioned oneself in this network of associations. The categories that CBDB has created for both kinship and social relations reflect the particular systems of significant distinctions we have encountered as we explore the legacy of information on individuals in premodern China. CBDB, as a relational database, then allows users to explore the interactions between these entities in the lives of groups of individuals. For example, consider the following set of entities and their relations with the basic entity PEOPLE:

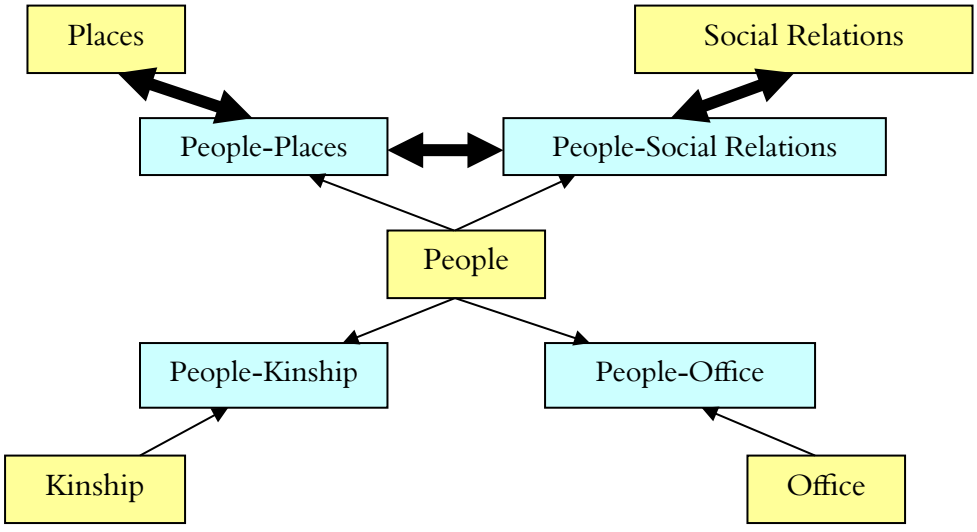


Although there is no direct link between KINSHIP and OFFICE, we still can explore the relation between them through the data we have accumulated about people. We can ask questions like "Was the role of medical officer hereditary, that is, were medical officers the sons or nephews of medical officers, and did the families of medical officers marry their children to one another?" What about men who held mid-level military ranks: were those who moved into civil posts likely to marry daughters of men who held civil posts?



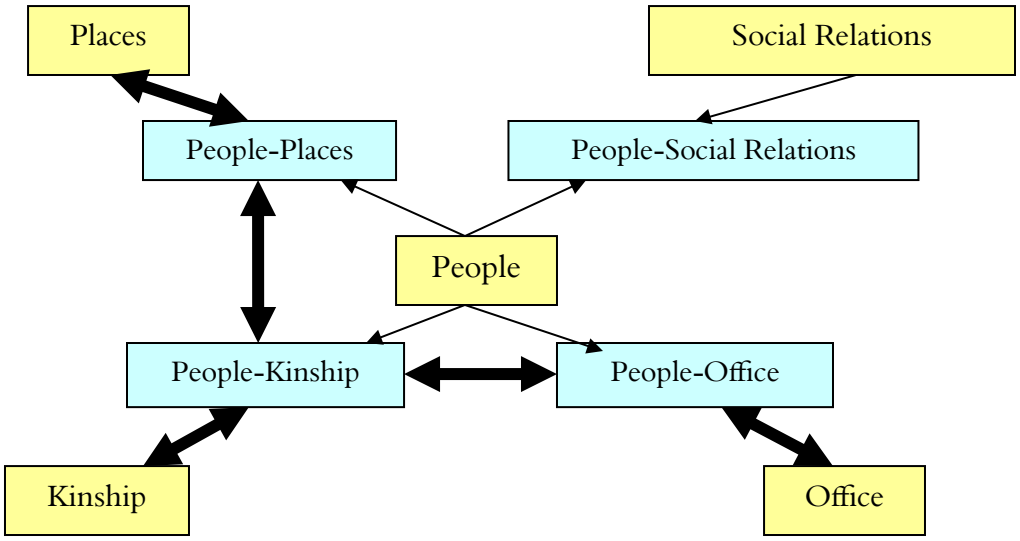
Querying the Relationship between OFFICE and KINSHIP

We can ask many, many questions about the relation of OFFICE and KINSHIP. Were there different patterns of marriage within rank for high civil officials and lower-ranking officials? Did these group form marriage alliances that created different strata? Did these patterns change over time? We can ask similar sorts of questions about PLACE and SOCIAL RELATIONS. Were people from Sichuan, for example, forming local connections, or did they establish empire-wide networks? Did these patterns change from the early to late Northern Song and then again from the late Northern Song to the late Southern Song?



Querying the Relationship of PLACE and SOCIAL RELATIONS

Finally, we can look at the interaction of multiple factors like the role of PLACE in the relationship between KINSHIP and OFFICE:



Querying the Role of PLACE in KINSHIP-OFFICE Relations

Were officials from Fujian more likely to develop local kinship networks than were official from Zhejiang? Did patterns differ depending on the rank, and did the patterns change over time?

In a relational database, the only real constraint on asking questions about the interactions of the entities in CBDB is how well one understands the database and the structure of the data in it.

Chapter 2. The Structure of CBDB

A. An Overview of the Entities in the Database

Database design uses tables to give concrete form to more abstract objects which we simply call "entities." Since the goal of a database is to capture the relational information about entities, it remains useful to keep the abstract objects separate from the tables that represent their relations. That way, one can more easily ask the question of how the tables need to change to better stand in for the entities they represent.

The central entity that defines biography in the database is, of course:

1. **People**

But since a *relational database* tracks the ways in which people form relations with other people, with their society (their political, social, economic and cultural institutions), and with the physical world, we also need entities with which **people** interact. First, relationships with people (these entities will be discussed in greater detail later):

2. **Kinship**
3. **Social (Non-kin) Associations**

Next, with political and socio-cultural institutions and activities:

4. **Status** (modes of social distinction such fame for calligraphy or serving as a monk)
5. **Modes of Entry** into Government or other careers (e.g., passing the civil-service examinations, nepotism or the *yin* protection privilege)
6. **Postings** to office (e.g., a magistrate or general)
7. **Social Institution** in which people collectively participated (from Buddhist temples and Confucian academies to the repair of city walls and bridges)

There also are texts that people produced and through which we know about people:

8. **Texts** (including primary texts, secondary texts, and paleographic data). These include data sources from which CBDB draws its information (primary sources, secondary scholarly compilations, and digital resources).

Then, there are structured aspects of the world with which people interacted that must be included in CBDB. The two aspects on which we have focused are administrative geography and bureaucratic structure:

9. **Administrative Geographic Hierarchy** (defined in political terms as superior and subordinate administrative units)

10. **Bureaucratic Organization** (the changes in bureaucracy and reporting responsibilities over time)

B. Details of Entities

NOTE: The database allows one to record the *source* of information, including the *pages* in the source from which the information comes, and to add additional *notes* as seems appropriate. Every item in the database that records information on an individual has the attributes of *source, pages,* and *notes*. Therefore I will not note these in the discussions below.

1. People

a. **Basic Data:** *name, male* or *female*, date of *birth*, and date of *death*.

Precise dates of birth and death often are not available, and all we have is a period of *years of activity* ("*floruit*" dates). Sometimes, not even that is available: we simply know the *reign period* (*nianhao*) or *dynasty*. In order to capture the level of precision in the data, the database allows the use of reign period information for all dates. One can give a specific year within the reign period, but one also can simply indicate "beginning," "middle", "end", or "unspecified." For analytic purposes, the database will algorithmically produce Western dates from the reign period information for birth, death, years of activity, and any other date given in the traditional Chinese *nianhao* designation, but it will preserve the vagueness in the *nianhao* coding.

b. **Ethnicity and Tribe Affiliation**

CDBD tracks ethnicity, like Han, Uighur, Tibetan, etc. We have over 465 codes at present. These codes are in the table ETHICITY_TRIBE_CODES, which organizes ethnicity and tribe designations by group and subgroup and includes variant forms for ethnicity names.

c. **Choronym**

From the Six Dynasties into the Tang, membership in a clan was of central importance in defining one's social status. From the Song Dynasty onward people did make claims descent from a particular clan from a particular place (like the Cui clan of Boling) but they carried little social or political weight. The combination of place name and clan name defined a *choronym*. The codes for these choronyms are in the table CHORONYM_CODES.

c. **Index Year**

For computational purposes, CBDB needs a single year value to locate a person in time. The *index year* is an artificial value used in analyses. In earlier versions of the

database, index year was based on when the person would have turned 60 *sui*. However, starting with the 2021 dataset, the index year has been based on the known or projected **year of birth.** The rules for calculating the value are complex and based on the following assumptions:

A1: that a man received a Jinshi (進士) degree at age 30, the Juren (舉人) degree at 27, and the Xuicai/licentiate (秀才/生員) degree at 21
A2: that a wife was 3 years younger than her husband
A3: the the first child was born when the father is at age 30 and a mother at age 27 (per assumption A2)
A4: that male children were born 2 years apart
A5: that a man died at age 63 and a woman at age 55

Rules Based on a Person's ***Birth/Death Dates***

Rule 1: Ego's index year = ego's birth year
Rule 2: If we know ego's death year and age at death, then: ego's index year = ego's death year – age at death
Rule 3: If we know just the ego's death year then: (per A5) ego's index year = ego's death year – 63 (for men), ego's death year – 55 (for women)
Rule 4W: Ego's index year = husband's birth year +3 (Note: If the woman was a concubine/second wife, then rule 9W precedes rule 4W.)

Rules Based on ***Degree Dates***

Rule 5: Ego's index year = the year he obtained the Jinshi (進士) - 30
Rule 5W: Ego's index year = the year her husband obtained the Jinshi (進士) - 30+3 = husband's Jinshi year - 27
Rule 6: Ego's index year = the year he obtained the Juren (舉人) - 27
Rule 6W: Ego's index year = the year her husband obtained the Juren (舉人) - 27+3 = husband's Juren year - 24
Rule 7: Ego's index year = the year he obtained the Xuicai (秀才/生員) - 21
Rule 7W: Ego's index year = the year her husband obtained the Xuicai (秀才/生員年) - 21+3 = husband's Xuicai year - 18

Rules based on ***Birth Years of Kin***

Rule 8: If we know the birth year of ego's **father**, then ego's index year is decided per assumption A3: ego's birth year was 30 years later than father's birth (Ego's index year = father's birth year + 30)
Rule 9: If we know the birth year of a **male's oldest child**, then ego's index year is decided per A3: ego's birth year was 30 years earlier than the birth year (Male's index year = oldest child's birth year – 30)
Rule 9W: If we know the birth year of a **female's oldest child**, then ego's index year is decided per A3: ego's birth year was 27 years earlier than the birth

year of her oldest child. (Female's index year = oldest child's birth year - 27)

Rule 10: If we know the birth year of ego's **older brother**, then ego's index year is decided per A4: ego's birth year was 2 years later than older brother's birth year. (Ego's index year = older brother's birth year+2)

Rule 11: If we know the birth year of ego's **younger brother**, then ego's index year is decided per A4: ego's birth year was 2 years earlier than younger brother's birth year. (Ego's index year = younger brother's birth year-2)

Rule 12: If we know the birth year of a **male's oldest son-in-law**, then ego's index year is decided per A3 & A4: ego's birth year was 30 years earlier than the birth year of his oldest daughter, and his oldest son-in-law was 3 years older than oldest daughter. (Male's index year = birth year of oldest son-in-law+3-30 = birth year of oldest son-in-law -27)

Rule 12W: If we know the birth year of a **female's oldest son-in-law**, then ego's index year is decided per A3 & A4: female's birth year was 27 years earlier than the birth year of her oldest daughter, and her son-in-law was 3 years older than her oldest daughter. (Female's index year = birth year of oldest son-in-law+3-27 = birth year of oldest son-in-law - 24)

Rule 13: If we know the birth year of ego's **grandfather**, then ego's index year is decided per assumption A3: ego's birth year was 60 years later than grandfather's birth year. (Ego's index year = grandfather's birth year + 60)

*Rules Based on the **Index Years of Kin***

(Note: CBDB iteratively uses the derived index years for these values.)

Rule 14: If we know the index year of ego's **father**, then we use father's index year to decide ego's index year per A3. (Ego's index year = father's index year + 30)

Rule 15: If we know the index year of a **male's oldest child**, then we use that year to decide ego's index year per A3. (Ego's index year = index year of oldest child - 30)

Rule 15W: If we know the index year of a **female's oldest child**, then we use that year to decide ego's index year per A3. (Female's index year = index year of oldest child - 27)

Rule 16: If we know the index year of ego's **older brother**, then we use that year to decide ego's index year per A4. (Ego's index year = index year of older brother + 2)

Rule 17: If we know the index year of ego's **younger brother**, then we use that year to decide ego's index year per A4. (Ego's index year = index year of younger brother - 2)

Rule 18: If we know the index year of a **male's oldest son-in-law**, then we use that year to decide ego's index year per A3 & A4: ego's birth year was 30 years earlier than the birth year of his oldest daughter, and his oldest son-in-law was 3 years older than oldest daughter. (Ego's index year =

index year of oldest son-in-law +3 -30 = index year of oldest son-in-law - 27)

Rule 18W: If we know the index year of a **female's oldest son-in-law**, then we use that year to decide her index year per A3 & A4: female's birth year was 27 years earlier than the birth year of her oldest daughter, and her son-in-law was 3 years older than her oldest daughter. (Ego's index year = index year of oldest son-in-law +3 -27 = index year of oldest son-in-law - 24)

Rule 19: If we know the index year of ego's **grandfather**, then we use grandfather's index year to decide ego's index year per A3. (Ego's index year = grandfather's index year + 60)

The CBDB table that records this basic biographical information is BIOG_MAIN. BIOG_MAIN assigns each person a unique ID.

d**. Floruit years**

CBDB gives two years: the earliest and the latest. Often when there is no data for index year or for birth and death dates, texts nonetheless provide datable references to individuals. CBDB gives the earliest and the latest known dates given in the textual sources we have examined so far.

2. Kinship

An instance of the **Kinship** relationship for an individual has three components (plus the source information):

person
kin
kinship relation

This relationship is structured as: "Person A has Person B (the kin) as his/her Kinship Relation." E.g. {Wang Anshi, Wang Anli, B-} means Wang Anshi has Wang Anli as a younger brother.

The building-block relations for **Kinship** are the 10 basic categories:

e	Ego (the person whose kinship is being explored)
F	Father
M	Mother
B	Brother
Z	Sister
S	Son
D	Daughter

H	Husband
W	Wife
C	Concubine

There are also variations on the nature of the relationship, as well as additional types of notation to represent types of kinship relations beyond the nuclear family:

+	Older (e.g. older brother B+, 兄)
-	Younger (e.g. younger sister Z-，妹)
⋆	Adopted heir (as in S⋆, adopted son)
°	Adopted
!	Bastard
^	Step- (as in S^ step-son)
½	half- (as in Z½ , half-sister)
~	Nominal (as in M~ , legitimate wife as nominal mother to children of concubine)
%	Promised husband or wife (marriage not completed at time of record)
y	Youngest (e.g., Sy is the youngest known son)
1, 2, 3…	Numbers distinguish sequence (e.g., S1, S2 for first and second sons; W1, W2 for the first and the successor wives)
n	precise generation unknown
G-#, G+#	lineal ancestor (–) or descendant (+) of #th generation
G-n, G+n, Gn	lineal kin of an unknown earlier generation (G-n), or unknown later generation (G+n), or unknown generation (Gn)
G-#B, BG+#	a brother of a lineal ancestor of # generation; a brother's lineal descendant of # generation
K, K-#, K+#, Kn	Lineage kin, of the same, earlier (–), later (+) or unknown (n) generation. CBDB uses "lineage kin" for cases where kinship is attested but the exact relationship is not known. Lineage kin are presumably not *lineal* (direct descent) kin.
K–, K+	Lineage kin of the same generation, younger (-) or elder (+).
P, P-#, P+#, Pn	Kin related via father's sisters or mother's siblings, of the same, earlier (–), later (+) or unknown (n) generation. Signified by the term *biao* (表) in Chinese. (CBDB uses these codes only when the exact relationship is not known).
P–, P+	Kin related via father's sisters or mother's siblings, of the same generation, younger (-) or elder (+).
A	Affine/Affinal kin, kin by marriage

The codes for the types of relationships are in the table KINSHIP_CODES. Although CBDB records all the many variations of kinship, searches for kinship networks in CBDB use an important set of four metrics for kinship distance to simplify the vast proliferation of terms. Each code KINSHIP_CODES table has values for

up, i.e., ancestor generation: *father* = 1, *grandfather* = 2, and so on
down, i.e., descendent generation: *son* = 1, *grandson* = 2, etc.
collateral relation: *brother* = 1, *brother's* wife's *sister*" =2....
marriage relation: *wife* = 1, *wife's* father's *wife* = 2, and so on.

Thus brothers, step-brothers, bastard brothers, and adopted brothers all have set of values {up = 0; down = 0; *collateral = 1*; marriage = 0}. The data recording the kinship relations between people is stored in the table KIN_DATA.

3. Non-kinship Associations

a. Simple Non-kinship Associations

These have a three-part structure: person + association + associate. The major challenge in recording the non-kinship **Associations** that individuals formed over their lives is to control the proliferation of categories derived from the historical texts.

Because associations are between pairs of people, there must be symmetrical types of associations. That is, if {**A** "is the student of" **B**} is in the database, then {**B** "is the teacher of" **A**} also should be so. In fact, the current version of the program automatically generates this second entry. Thus, ASSOCIATIONS as an entity has an internal structure:

Association type
Paired Association type
Association Categories/subcategories (3 levels at present)

When editors for CBDB create a new category of association, they must also create its converse. Mutual associations, of course, are their own converse: {**A** "is friend of" **B**} is the same as {**B** "is friend of" **A**}. In most associations, however, the two people play distinct roles, and CBDB needs the converse category to capture the roles of the two people from their different perspectives: to record for **A** that {**A** "followed" **B**} also means that for **B**, {**B** "*was* followed *by*" **A**}.

b. Mediated Associations

In some important cases, associations form through the mediation of institutions or people. CBDB captures these types of relations by adding additional data to associations. For example, we might know of a relation between X and Y because X asked Y to write a biography for his mother's tomb. In order to record all the variations, the record structure for the table ASSOC_DATA has become rather challenging to understand.

c. **Structure of an Association Record**

Because associations in pre-modern Chinese society often are complex, the table tracking associations in CBDB uses a correspondingly large number of fields:

Basic Information

1. Person ID
2. Associated person ID
3. The kind of association (association code)
4. The number of objects or events establishing the association

Information about Kinship and Other Relations that played a role in the Association

5. The kinship relation, if the association was established through a relative of the person
6. The ID of the person whose kinship relation established the association
7. The kinship relation of the associate, if the association was established through a relative of the *associated* person
8. The ID of the kin of the associate through whom the association was established
9. The ID of the person who claimed the existence of the association: for example, a son claiming it for his father

Time and Place of the Association

10. The ID of the place of the association
11. The sequence of an association, if one does not know the actual date
12. The date of the association (year, month, and day, if known)

Contextual Information

13. The code for the social institution at or through which the association was established
14. The code for the occasion on which the association was established
15. The code for the genre of the writing that establishes the association, if relevant
16. The title of the work that established the association, if relevant
17. The code for the scholarly topic around which the association was formed

Source and Notes

18. Source ID
19. Note

4. Status

CBDB has a table to take note of "social distinctiveness," that for which people were known in society. Since the dating often is uncertain, however, the table has a field to record *sequence* if known. Some forms of social distinctiveness may combine roles (a

Buddhist monk known for his calligraphy, or a literatus who runs a printing firm). At present, CBDB records the different aspects of status under distinct categories. This is a question awaiting future research.

The structure of a Status datum for a person is:

Person ID
Status code
Status sequence
Date
Source information and notes

STATUS as a category of social experience (as opposed to any particular person's status within the structure of social distinction) is a simple entity:

Status code
Status description
Status category and subcategory 1
Status category and subcategory 2

Since social distinctions change over time, CBDB will continue to add to its current list as it draws upon sources for earlier and later periods.

5. Modes of Entry

ENTRY itself is a simple entity, just a name, a type, and a subtype. At present it largely describes entry into government, but CBDB also has begun to track categories like monks' ordinations. Because different routes of entry entail different types of information, the instance of an ENTRY event for an individual is more complex. If a person enters government through the examination system, for example, we would like to know the type of examination and the date of the degree. (CBDB also tracks failed examinations.) If, in contrast, one enters government through the merit of someone else, the person, and the relationship to the person should also be recorded, if known. Thus if Zhang Weisan entered office through *yin* protection privilege deriving from his uncle Zhang Jingyi, the entry would be:

Person: [ID of] Zhang Weisan
Entry type: [code for] *yin*
Entry relation type: [code for] Uncle
Entry relation: [ID of] Zhang Jingyi

Since it is also possible that one can enter office through the *yin* privilege of a non-kin associate, the "entry event" will need to have a way to record the non-kinship relation. In the end, then, the ENTRY event has many attributes, only some of which are relevant to any particular instance:

Person ID
Entry type code
Entry relation type code (for kin)
Entry associate type code (for non-kin)
Entry associate ID (used for both kin and non-kin)
Entry test date (both Western and *nianhao* + year (if known))
Entry test ranking
Entry address ID

6. Offices and Postings

CBDB currently lists over 32,000 office titles and—for the Tang, Song, Yuan, Ming, and Qing—their place in the government bureaucracy. POSTINGS is an entity at the intersection of people, the bureaucracy, and—since most instances will be away from the capital—places. A person serves in an office at a given rank in particular place at a specified time. However, there are instances when a posting includes jurisdiction over more than one administrative unit, and there are times when a single posting entails more than one official position. Following the rule that one-to-many relations (i.e., one posting may have more than one address and one posting may involve more than one office title) require separate tables, information about instances of POSTINGS requires *two* additional categories of one-to-many relations: in addition to the basic **postings** table, there are a **posted-to-office** table (relationship of people to office created in the POSTINGS event) and a **posted-to-office-address** table (relationship of people to office to place).

Posting-Data
Posting ID (this is a unique number)
Person ID
Source and Notes

Posted-to-Office
Posting ID
Office ID
Appointment Type (regular, provisional, honorary, etc.)
Sequence (since often only the order of office is known with no further information about the years for any of the postings)
Year (both Western and *nianhao* + year: a person may have duties added while still serving in a post)
Sources and Notes

Posted-to-Address
Posting ID
Office ID
Address ID

Considerations for Future Development

1. Buddhism and Daoism

Buddhist and Daoist bureaucratic positions eventually will be added to the OFFICE and POSTINGS entities. This, however, also entails significant research to clarify the historical changes in the structure of the Buddhist and Daoist bureaucracies.

2. Tracking Historical Change in Bureaucratic structure

One of the design issues that need to be considered again is how much of the complexity of the Chinese imperial bureaucratic system should be captured in the database. In the Chinese system from the Han through the Qing, the duties of a position may change even though the title of the office remains constant, or the duties may remain constant although the title changes. Scholars have objected that Charles Hucker's *Dictionary of Official Titles* tries to force a continuity of function onto office names when it would have been more useful to simply acknowledge the drifts. Hucker's translated titles are indexed, however, and provide those who do not read Chinese with an easy means of further investigation. CBDB is planning to create tables that will capture the historical changes in the functions designated by any particular office title. (**Office Name** would become one entity and **Office Function** would be another.) Most of the actual duties of an office at any particular time are not relevant to the CBDB because these details contribute little to the analytic power of the database; the attributes of an office that do matter are (1) office as an indication of salary/rank or actual function, (2) the other office to which it reports, and (3) the type of the office (i.e. central military, prefectural civil, etc.) At present, CBDB has captured some of this information, but clarifying the changes in office title is in itself a major research project.

7. Places

CBDB uses a strategy for coding places that derives from the China Historical Geographic Information System (CHGIS) project and relies on the spatial entity ADDRESSES.

Addresses are specifically historical instances of place designation that refer to an administrative jurisdiction. Although administrative jurisdictions such as counties (*xian*) and prefectures (*zhou* and *fu*) were bounded spatial entities, CBDB uses the coordinates for the administrative seat as the address; it does not provide boundaries. Boundaries can be downloaded from CHGIS. If either the boundaries or the name changes, a new address record (and ID) must be created. These historical instances, however, are part of administrative hierarchies: this information is preserved in a "belongs-to" table that serves the same function as the "part-of" table in CHGIS. Since an address ID changes only when the unit changes shape or name, it *does not* change ID simply when it becomes part of a different higher level administrative unit. Thus there are two tables:

Address Code
Address code
Address name
Administrative type
X coordinate
Y coordinate
Address first year
Address last year

Belongs to
Address code
Belongs-to Address code
Belongs-to first year
Belongs-to last year

From these two tables CBDB generates a convenient **Addresses** table that is used in the online database and can be consulted in the stand-alone version to provide information about the role of administrative units in the bureaucratic structure. Its structure is:

Address code
Address name
Address first year (that the address belongs to the superior place)
Address last year (that the address belongs to the superior place)
Administrative type
X coordinate
Y coordinate
belongs1 (the parent: the larger administrative unit it reports to)
belongs2 (the parent of the parent)
belongs3 (etc.)
belongs4
belongs5

To allow the examination of trends *across dynastic boundaries*, the database needs a way to examine what happens in a particular location over long periods of time. For this, CBDB relies on data about physical location, the *x-y coordinates* on the map.[1] The analytic forms allow one to use the x-y data for the addresses one has selected to define squares around those x-y coordinates and locate additional addresses across time that fall within those squares. These addresses then can be searched across the time period one has specified.

To reiterate, CBDB uses the x-y coordinates of the seat of the administrative unit.

[1] In Geographic Information Systems (GIS) research, longitude and latitude typically are referred to as x-y coordinates.

8. Biographical Place Information

People have many connections to place: where they were born, lived, died, and were buried, where they served in office, where they held property and ran businesses, where they visited. Since these relations to place arise out of activities recorded in separate tables in CBDB (e.g., office holding, and possessions), the information appears in these various tables rather than in one place. The tables that record information about people and places are:

Basic biographical information relating to place (BIOG_ADDR_DATA)
Place of official service (POSTED_TO_ADDR_DATA)
The place where a non-kinship relation took place (ASSOC_DATA)
The place where people participated in social institutions (BIOG_INST_DATA)

The CBDB form (LookAtPlace) allows the user to ask questions that integrate all these sources of place information. Note that at present CBDB does not systematically preserve information about places persons briefly visited, where they received their education, or where they wrote texts.

CBDB attempts to associate each person with an *index place*. As with *index year*, CBDB assigns these place associations based on available information, but the data is often incomplete. Therefore CBDB uses a hierarchy of categories of place association to assign a person's index place. CBDB first uses the "basic affiliation" 籍貫, if available. The order of assigning address affiliations is as follows:

1. Basic affiliation 籍貫
2. Household address 戶籍地 (Ming dynasty)
3. Actual residence 落籍
4. Last known address
5. Moved to
6. Eight Banners (Qing dynasty)
7. Alternative basic affiliation
8. Place of exile

However, this hierarchy of codes to use in assigning the index place may not be the most suitable for particular research projects. Thus, CBDB allows the user to change this order. See Appendix X for discussion.

9. Texts

There are three major types of texts of concern to the database: inscriptional and other paleographic material, printed primary texts, and secondary scholarship (in both print and digital form). Since a work like Huang Zongxi's *Song Yuan xue'an* is both a scholarly compendium of earlier writings and a work in its own right, and since the paleographic

materials also were written by authors who are of interest to the database, these distinctions for pre-modern texts of any sort are neither clear nor useful. CBDB accordingly treats all three types as TEXTS. Texts have the attributes one can expect:

title
category of writing (inscription or manuscript/printed)
genre (the bibliographic categories common to that period)
current publication date
current publisher
current publication location

People can relate to the text in a variety of roles:

author
publisher
editor
collator
translator
annotator

The tables for texts are:

Texts Codes
Text ID
Text Name
Date of composition
Current status: extant or not
Current Publication Information (if extant)

Text Data
Text ID
Person ID
Role ID (from the table TEXT_ROLE_CODES)

10. Social Institutions

People participated in the lives of their communities in many ways. A man, for example, may have served for several years as the director of an academy. That academy had students during this period: their respective roles in the academy would have served as important social links between the man and the students. The academy also had donors who contributed to its creation and upkeep and helped to define a community centered on the institution. Similar patterns appeared for Buddhist monasteries and Daoist temples.

CBDB is beginning to track this information in a way that captures the uncertainty we find in the historical sources. There are, for example, thirty-nine temples with the name Kaiyuansi 開元寺. A biographical source may tell us that Wang Anshi contributed funds to repairs at a Kaiyuansi, but we may not know (yet) which Kaiyuansi was the recipient. Other sources eventually may clarify the point, but for the moment CBDB simply records "a Kaiyuansi." There are four tables used to record this information:

Social Institution Names
Institution Name ID
Institution Name

Social Institutions
Institution Name ID
Institution Code (this is a unique ID for each institution: the name is an attribute that may change, but *what the institution is*, identified by the ID, does not. Of course, if the textual data shows that the name change signified to the people at the time that this was a new institution, then CBDB assigns the institution a new *institution code* as well.)
Institution Type ID
Institution Dates (this includes the beginning and ending years, if known, as well as the first known and last known years

Social Institution Addresses
Institution Name ID
Institution Code
Address ID (this gives an approximate location by identifying an administrative unit)
XY-coordinates (this may be more precise than the coordinates associated with the Address ID. An institution may move within its locality.)
Address Type (derived from Address ID or recorded independently)
Address Dates

Relationship of People to Institutions
Person ID
Institution Name ID
Institution Code (if only the name is known, CBDB assigns a 0 to this field)
Institutional Role Code
Role Dates

Summary of Tables in CBDB

1. Basic Entities

These represent the basic elements of the social world of pre-modern China. Each has a complex history and structure that are set out in additional ancillary tables. CBDB records the interaction of people with these aspects of their world in the secondary tables.

Table Name	Description
ADDR_CODES	the units in the administrative geography of China.
ADDRESSES	a convenient reference table that displays the hierarchy
ASSOC_CODES	the non-kinship social relations that connected people
BIOG_MAIN	the people of pre-modern China
ENTRY_CODES	the means by which people entered into institutions
KINSHIP_CODES	the kinship categories of pre-modern China
OFFICE_CODES	the units of the bureaucratic organization of government
SOCIAL_INSTITUTION_CODES	a list of academies, monasteries, temples, etc.
STATUS_CODES	the means by which people attained social distinction
TEXT_CODES	the corpus of pre-modern writings + important secondary works

2. Relations between Basic Entities

Table Name	Description
ADDR_BELONGS_DATA	data for the hierarchical structure of administrative units
ALTNAME_DATA	the many names by which people were known
ASSOC_DATA	the non-kinship relations between people
BIOG_ADDR_DATA	relations between people and administrative geography
BIOG_INST_DATA	the relations of people to social institutions
BIOG_SOURCE_DATA	the list of sources used in defining the CBDB data for a person
BIOG_TEXT_DATA	the relations of people to texts
ENTRY_DATA	the initiating relations between people and institutions
KIN_DATA	the kinship relations connecting people
OFFICE_TYPE_TREE	the hierarchical structure of bureaucratic organizations
POSTED_TO_ADDR	the relations between people, office, and place
POSTING_DATA	the container table for postings: people linked to office
POSTED_TO_OFFICE_DATA	the details of people's connection to office
STATUS_DATA	data on a person's place in the system of social distinctions

3. Relationship Type Information

Table Name	Description
BIOG_ADDR_CODES	the categories of relations between people and places
ALTNAME_CODES	the categories of names by which people were known
APPOINTMENT_TYPE_CODES	the categories of relations between people and postings: regular, acting, probationary, etc.
ASSOC_TYPES	broader categories of social relationships that organize the many non-kinship association codes into groups
ASSUME_OFFICE_CODES	indicating whether a person took up the posting
BIOG_INST_CODES	the roles a person plays in relation to an institution
ENTRY_TYPE	broader categories of entry to organize the entry codes into groups
EXTANT_CODES	indicating degree of the source and its known existence
GENRE_CODES	the bibliographic classifications of texts
GENRE_TYPES	the broader categories of bibliographic classifications
LITERARYGENRE_CODES	the forms of literary composition
OCCASION_CODES	the events in which people participated
OFFICE_TYPES	the categories of offices
SCHOLARLYTOPICS_CODES	the categories of topics of learning and scholarship
SOCIAL_INSTITUTION_ADDR_TYPES	the type of address (actual or derived) used for an institution
SOCIAL_INSTITUTION_TYPES	the categories of social institutions
STATUS_TYPE	The categories of social distinction
TEXT_BIBLCAT_CODES	The fine-grained categories by which texts are organized
TEXT_BIBLCAT_TYPES	The larger units for textual categorization
TEXT_ROLE_CODES	the categories of relations between people and texts
YEAR_RANGE_CODES	the relative degree of exactness of a date

4. Historical Auxiliary Tables

Table Name	Description
CHORONYM_CODES	codes for the place+surname used to identify medieval clans
COUNTRY_CODES	codes for countries appearing in the data
DYNASTIES	codes for dynasties and periods
ETHNICITY_TRIBE_CODES	codes for ethnic groups appearing in the data
GANZHI_CODES	codes for the sixty two-character terms in sexagenary cycle
KIN_MOURNING	codes for all kin relations and mourning obligations in the five degrees of mourning
MEASURE_CODES	codes for quantities of goods, money, books, and space
NIAN_HAO	codes for all reign period titles
SOCIAL_INSTITUTION_ALTNAMES	a list of alternative names for social institutions
SOCIAL_INSTITUTION_ALTNAMES_TYPES	codes for different types of alternative names

5. Analytic Auxiliary Tables

Table Name	Description
ASSOC_CODE_TYPE_REL	the relationship of specific social relations to larger categories of social relations
ENTRY_CODE_TYPE_REL	the relationship of specific modes of entry to larger categories of entry
GENRE_CODE_TYPE_REL	the relationship of specific genre codes to larger categories of genres
OFFICE_CODE_TYPE_REL	the relationship of specific offices to the office hierarchy
OFFICE_CATEGORIES	the categories of offices: rank, honorary, etc.
STATUS_CODE_TYPE_REL	the relation of specific status codes to the larger categories of social distinction
TEXT_BIBLCAT_CODE_TYPE_REL	

6. "Denormalized" Tables

Because the data tables for the relations between basic entities (group 2 above) are in normalized form that uses codes that refer to other tables for the entities, relations, and historical information, they are difficult to use for queries (See Chapter 4). In order to simplify the process of writing queries, CBDB provides a set of tables where the codes have been supplemented by the values (mostly text strings like the names of people, places, official positions, etc.) to which the codes refer. The main tables are listed below:

Table Name	Description
ZZZ_ALT_NAME_DATA	fills in alternate name type
ZZZ_BIOG_ADDR_DATA	fills in address and address type
ZZZ_BIOG_MAIN	fills in nianhao, ethnicity
ZZZ_BIOG_NAME_OFFICE	Links surnames to posted office names (used in searching)
ZZZ_BIOG_TEXT_DATA	fills in the person's name, the person's role, and the text data
ZZZ_ENTRY_DATA	fills in the person's name, entry type, etc.
ZZZ_KIN_BIOG_ADDR	this is the table for kinship, but it also provides the index place
ZZZ_NONKIN_BIOG_ADDR	this is the table for associations, but it also provides the index place
ZZZ_POSTED_TO_ADDR_DATA	fills in person name, office name, address information
ZZZ_POSTED_TO_OFFICE_DATA	fills in person name and office information
ZZZ_STATUS_DATA	fills in person name and status description

Chapter 3. CBDB Tools for Analysis

The China Biographical Database contains large amounts of information, but the information is of little value unless there are ways to analyze it. At present, the Access version of CBDB has seven forms specifically designed to allow the user to query the database about important categories of information. The names of the forms describe their function.

1. **LookAtEntry** allows one to find groups of people who qualified for office through a particular route for a specified period.
2. **LookAtAssociations** allows one to find groups of people who were linked through a particular category of association
3. **LookAtOffice** allows one to look at not only the people who held particular offices but also those who held related offices subordinate to ever higher levels of bureaucratic structure.
4. **LookAtKinship** allows one to examine the kinship networks for individuals. These include both the mourning circle of the traditional Chinese kinship system and more extended sets of relations.
5. **LookAtNetworks** allows one to look at all the networks (both kinship and social relations) for an individual, a group of individuals, or a specified place.
6. **LookAtAssociationPairs** allows one to examine the intersection of the networks for two individuals. It locates both people connected to the two target individuals but also can identify connections at one further remove (i.e. people who had a connection with the first individual who had relations to people somehow related to the second individual).
7. **LookAtPlace** brings together all the types of relations between people and places into a single form. People who formed social relations in a place, served in office there, or whose registry was there all can be part of a single list.
8. **LookAtStatus** allows one to group those individuals identified by particular forms of social distinction.
9. **LookAtTexts** allows the user to explore the people associated with the production of particular categories of texts (unofficial histories, commentaries on canonical ritual texts, etc.)

In more complicated queries, one can explore relations between groups of people by using the results of a search in one form as the input to a second form. Chapter 4, on advanced queries, considers an example of this approach. Beyond the six forms, however, Access also allows the user who is familiar with the structure of the database to make queries that can look at any and all aspects of CBDB data. This process uses Access' built-in Query Designer to create SQL (Structured Query Language) queries to examine the data and is the second topic in Chapter 4.

NOTE: The explanations of the forms in this chapter provide examples of searches, but the results you get will differ from these because CBDB periodically updates the data in the tables.

A. The Navigation Pane

As the name suggests, the **Navigation Pane** is the central console for using the forms developed for the Access version of the database. Clicking on the nine query command buttons opens the browser and the eight analytic forms discussed above. The Navigation Pane also has four additional functions.

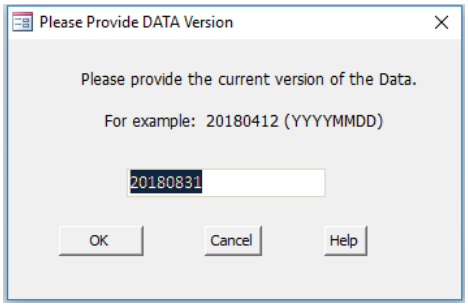
1. **Error Reporting:** The Navigation pane also allows you to report problems with the program. If you click on "Report an Error," the program brings you to a Google form:

Select the type of error and fill in the information requested on the form.

2. **User's Guide:** Clicking on "Users Guide" will open a copy of this User's Guide.

3. **Relinking the Data Tables:** The current version of the Access database splits the CBDB data tables from the user interface. Because the database has grown very large, the size of the files that hold the data were approaching the limit of what Access could handle, and thus the data tables are in three separate files. The user interface then is *linked* to the tables. When you first open the user interface, the program automatically links the interface to the data tables, as long as the three files are in the same folder as the user interface file. However, because the data and the interface are in separate files, it is now possible to update each of these separately.

If there is a new release of the CBDB data, you can download the new data files into your CBDB folder and then link the interface to those new files. The CBDB data release will have a **date-stamp** in the form YYYYMMDD as part of the name of the files. If you click on "Relink Tables," a form will request the date-stamp information:



Simply fill in the new version information and click "OK."

4. **Changing the Index Address Ranking:** While the default setting for how CBDB defines index places works well for most users, scholars pursuing particular topics may need to change how index place is defined. Clicking on this command button opens a form to allow the user to do just this. When one opens the form, it shows the current order for selection of categories of relation to place that is used to define the index place:

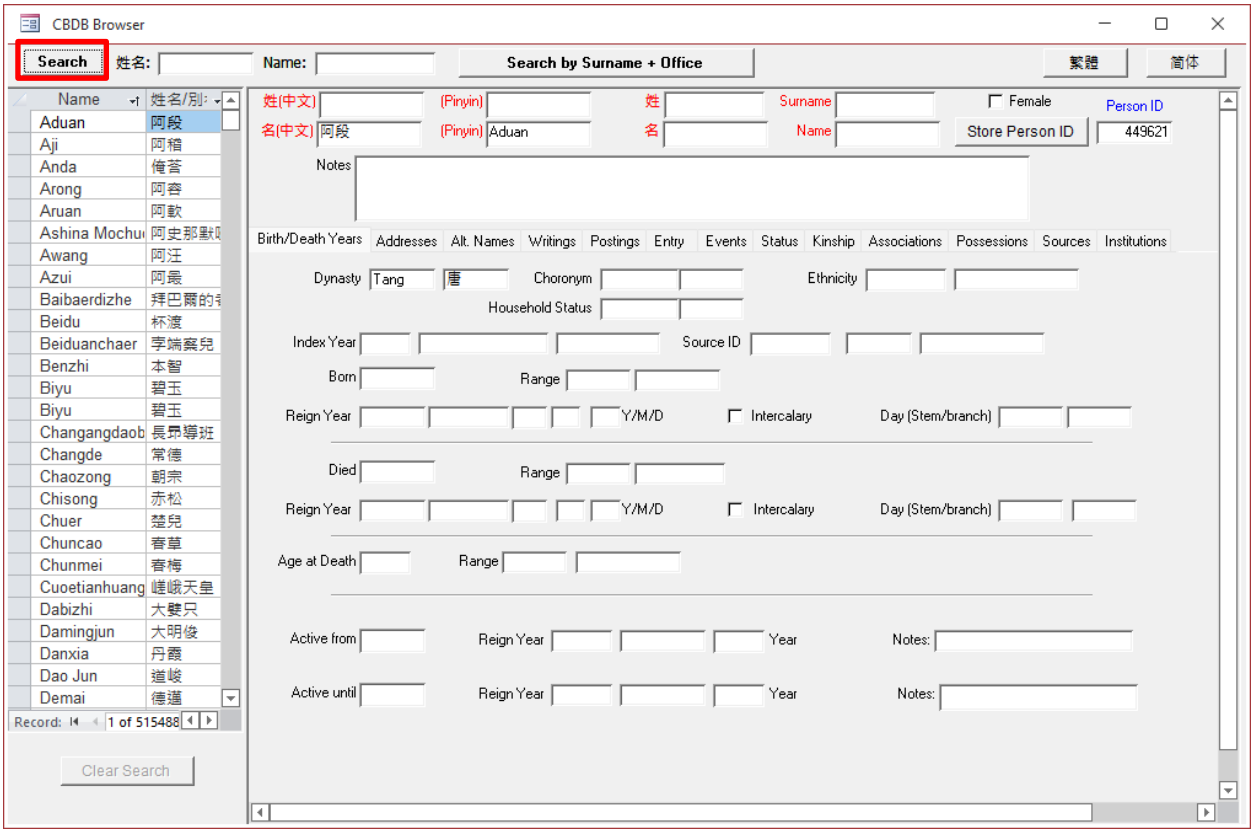
One then can choose a new set of categories to be used to define index place. Clicking on "Disable" limits the selection process to just those categories above the disabled row:

In this example, we set the first choice for index place to "Household Address" (戶籍地) and the second to "Actual Residence" 落籍. All other relations to place are ignored. Clicking on the "Update Index Addresses" then recalculates the index place for BIOG_MAIN and replaces the values for index place in all the tables that use the value.

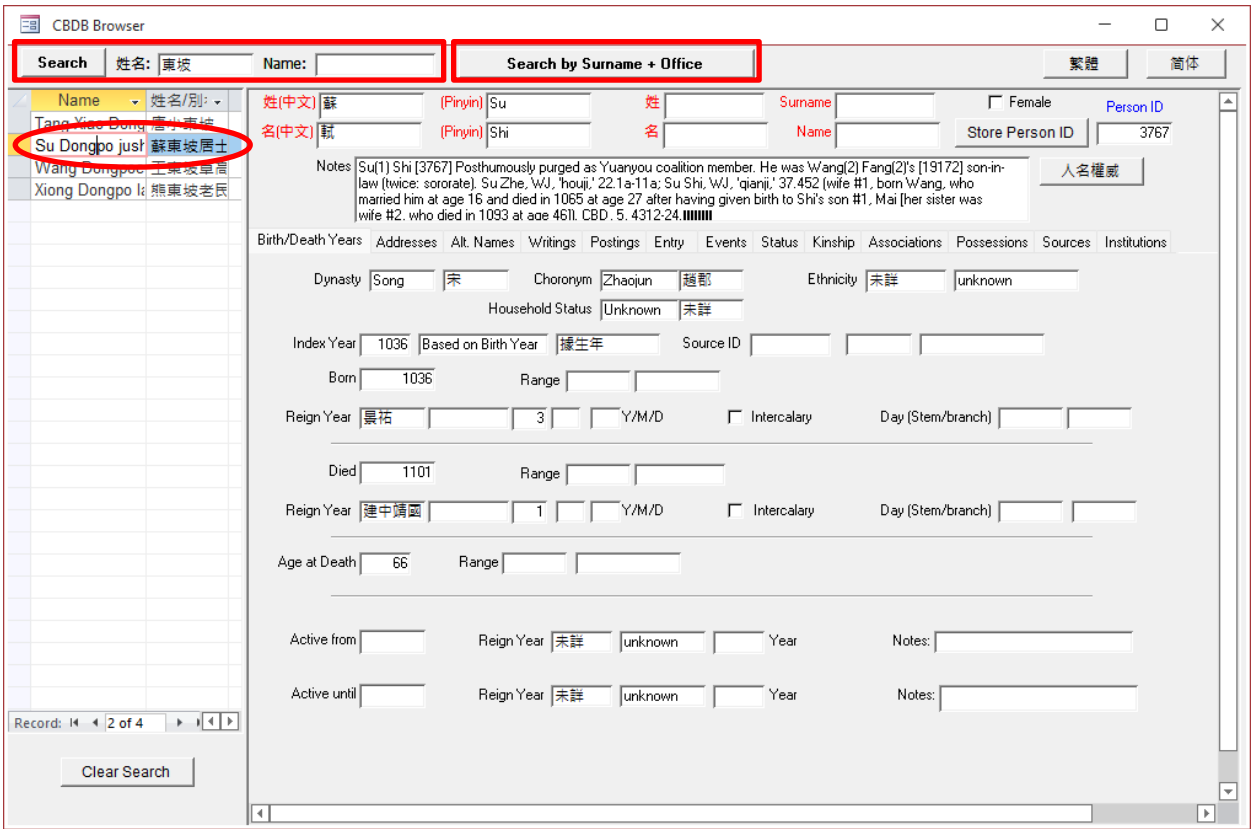
After changing the ranking of place affiliations, one can restore the CBDB default ranking and index place values by simply clicking on the "Reset to Default" command button.

B. The Browser: Looking up Data on an Individual

The browser in CBDB provides a convenient way to explore basic information on individuals in the database. It draws on just the *raw data* for people in the database, so it has no significant analytic or synthetic abilities. The only exception in is the name search functions described below. When one opens the browser, it begins with the first person in the BIOG_MAIN table:



(The sorting by name starts with all people who have just a personal name but no surname.) Since BIOG_MAIN has over 470,000 people, just scrolling through the window on the left is not the most effective way to locate an individual. Therefore, the browser has two search functions. The first is a search by name. "Name," however, includes all the categories of names used in CBDB (courtesy name 字, style name 號, etc.). Thus, if a text provides only Su Shi's 蘇軾 style name, Dongpo 東坡, rather than his full name, one can search by that alone to see how many people share those two characters in any of their names:

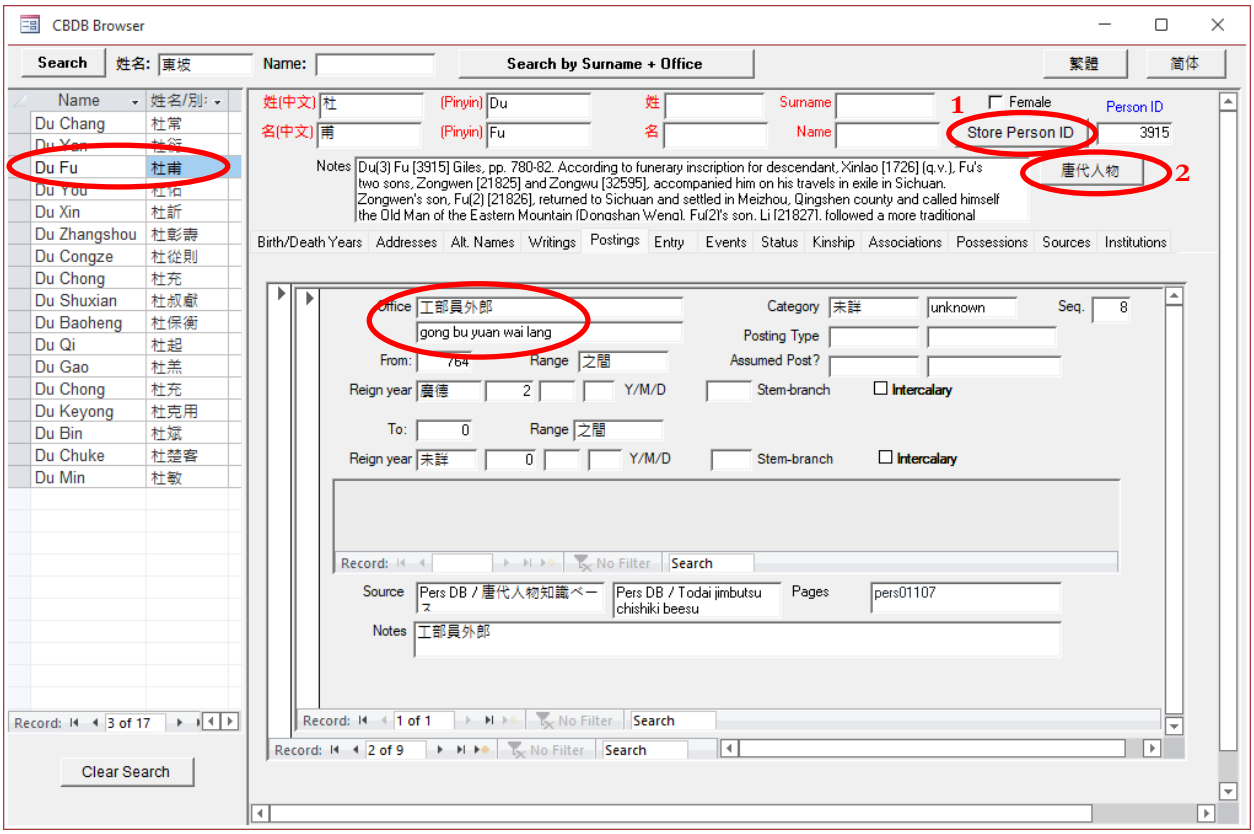


Of course, one can directly **search** by "Su Shi" or "蘇軾" as well. Searching by *pinyin* (i.e., "Su Shi") looks for the search string *anywhere in the name*, but the search can be narrowed. Using a *lower-case* name, the search looks for the phrase in *any part of a name*, so that "hao" will produce "*Zhao* Fang," "*Shao* Yong," and "*Chao* Buzhi," etc., as well as "*Hao* Jing" and "Cheng *Hao*." If the first letter is *capitalized* ("Hao"), the search finds names that *begin with* "Hao" ("*Hao* Jing" and "Cheng *Hao*"). If one adds an "!" at the beginning, the search routine looks at just *surnames*.

If a text provides only *a surname and a title*, the latest version of the browser now allows one to search by those as well. Clicking on the "**Search by Surname + Office**" command button opens a form that allows one to not only specify the name and office but also to narrow the search by providing a range of index years or dynasties:

Note: Using the Chinese name and office produces more certain result, since the form simply looks for all office titles for people with the given surname in the database. (The *pinyin* office titles are in lower case with a space between each character.) Clicking on the **Postings** tab and scrolling through Du Fu's nine posts confirms that he indeed served as a *Gongbu yuanwai lang*.

Once one clicks the "Search" command button, if the form finds any people who match the specified criteria, these results are transferred to the Browser:



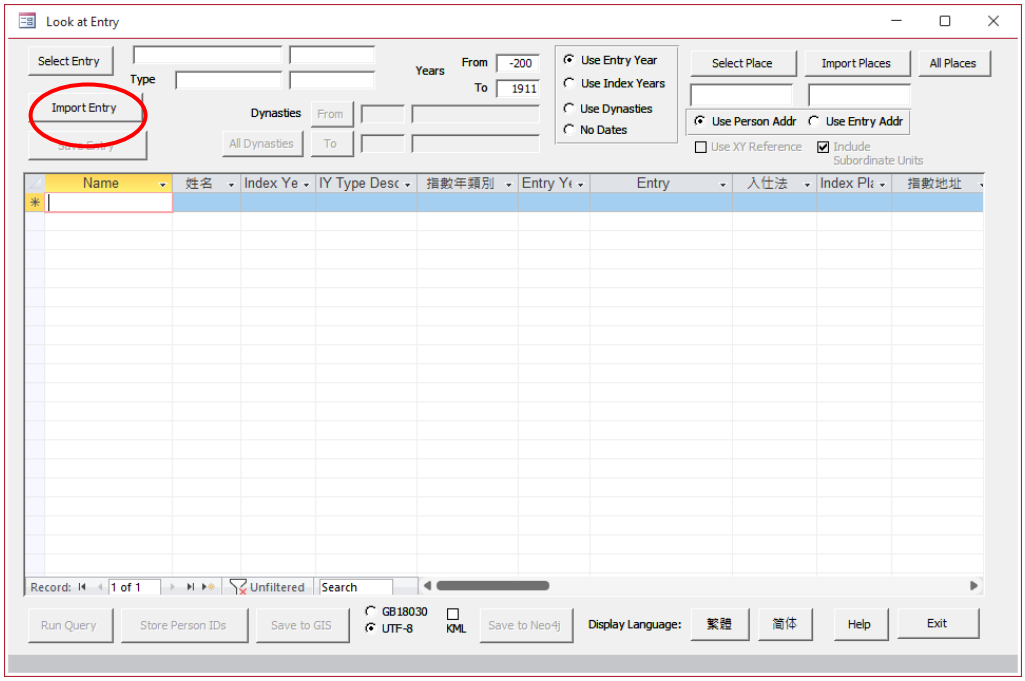
Having located a person, one can use the **Store Person ID (1)** button to save the person's ID to be reused in **LookAtKinship**, **LookAtNetworks**, and **LookAtAssociationPairs**.

The browser itself is fairly self-explanatory. Since the CBDB interface in MS Access aspires to be bilingual, the user can switch between English, traditional characters and simplified characters by clicking on the buttons in the upper right of the form. Each tab provides the basic data in CBDB for the individual in the main categories: addresses, alternative names, writings, postings, mode(s) of entry into service, events, status, kinship relations, social relations, possessions, the sources used for the information, and relations to institutions. The lists of kinship and social relations are just those stored in the basic tables: they are far less complete than the lists created by **LookAtKinship** and **LookAtNewtworks**.

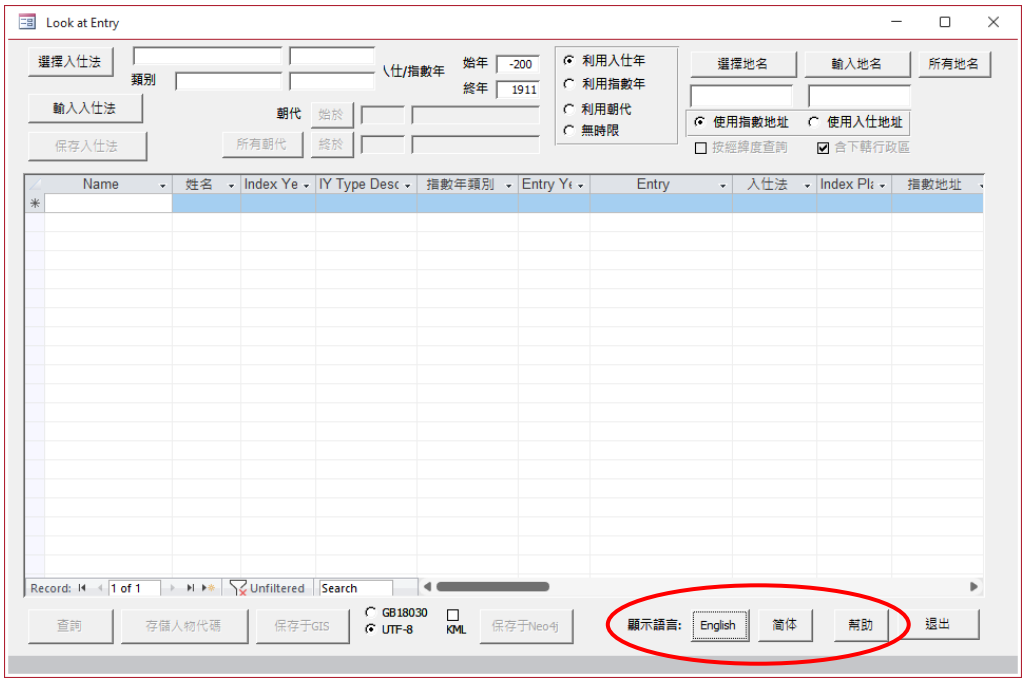
In this particular example, Du Fu 杜甫 has an entry in the Kyoto University *Tōdai jinbutsu chishiki beesu* 唐代人物知識ベース database: clicking on the **button with that label (2)** opens the webpage with his data in that database. CBDB has built-in links for the *Tōdai jinbutsu* database, the McGill *Ming-Qing Women Writers* 明清婦女著作數據庫 database and the Name Authority Database of the Ming Qing Archive 明清人名權威檔案 at Academia Sinica.

C. Using the Form "Query by Methods of Entry into Government

LookAtEntry is the simplest form. One opens it by clicking on "Query by Methods of Entry into Government" on the main page and clicks on the "Select Entry" button to choose a category:

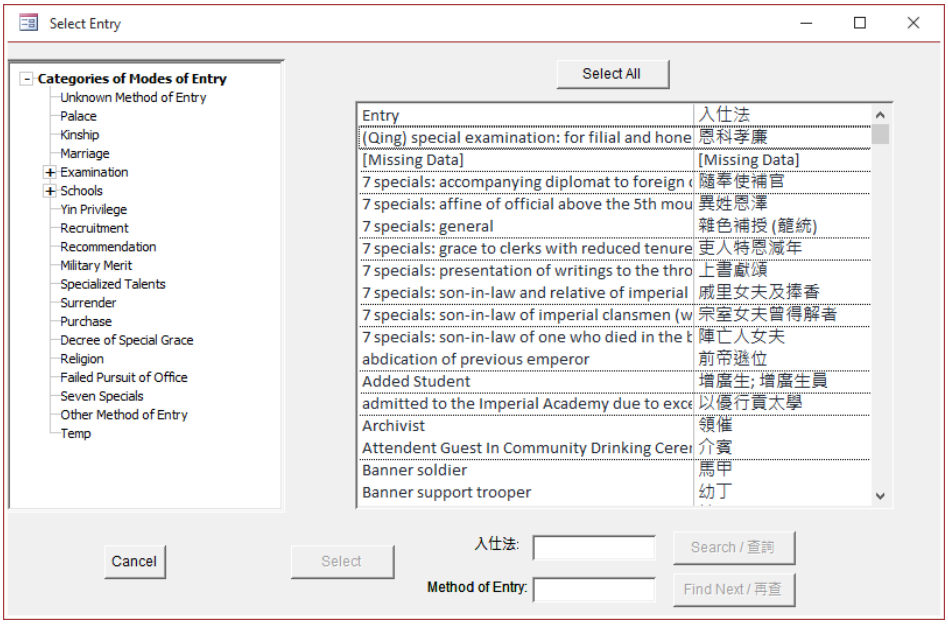


Note that all of the forms have the option to switch between English, traditional or simplified Chinese. When one clicks on the "繁體" label, it then gives one the option to return to English:

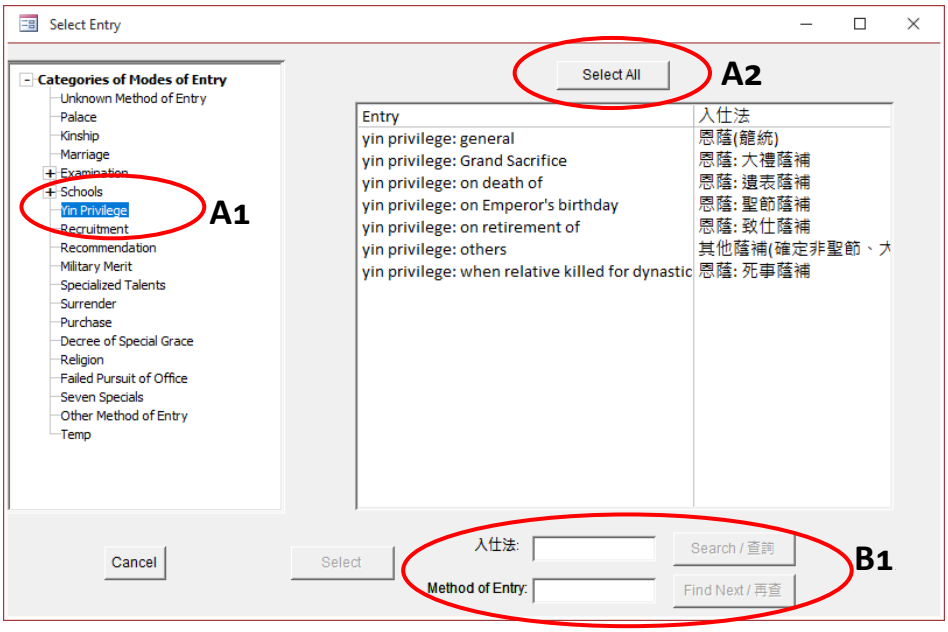


Selecting the Modes of Entry

Clicking on the **Select Entry** button opens a form with a list of options. Since there are many different ways to attain eligibility for office, CBDB uses a collapsible tree to simplify the selection process:

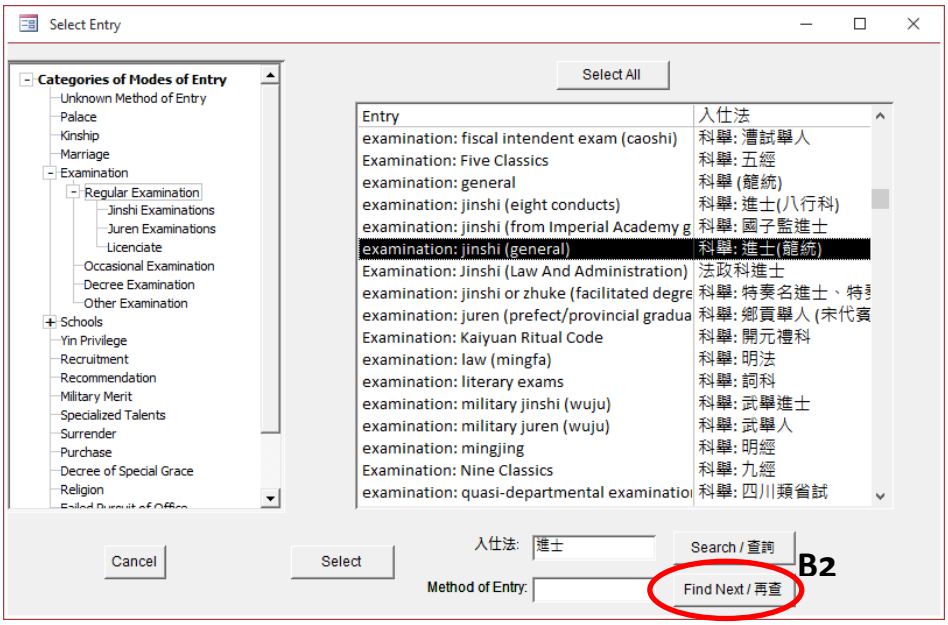


One can narrow the choices by looking at a particular general type of entry which is on the menu on the left of the window (A1):

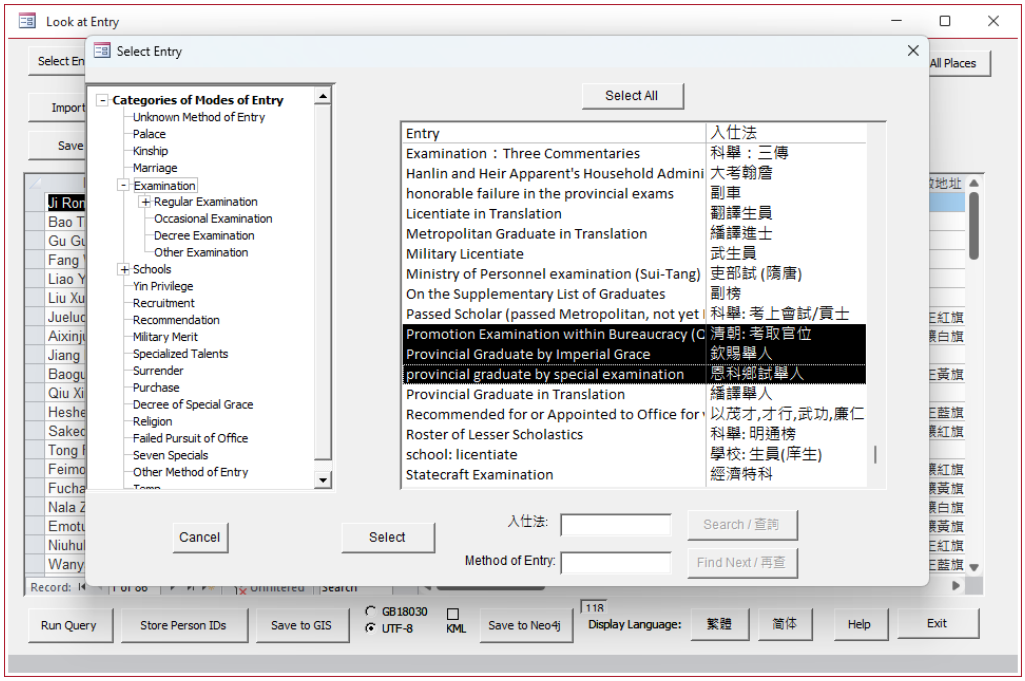


One can either select a specific method of entry from the menu on the right or select all the listed methods (A2). One also can search for a specific method using the search box located on the bottom right corner (B1). The searching rules for CBDB are to first look for the search

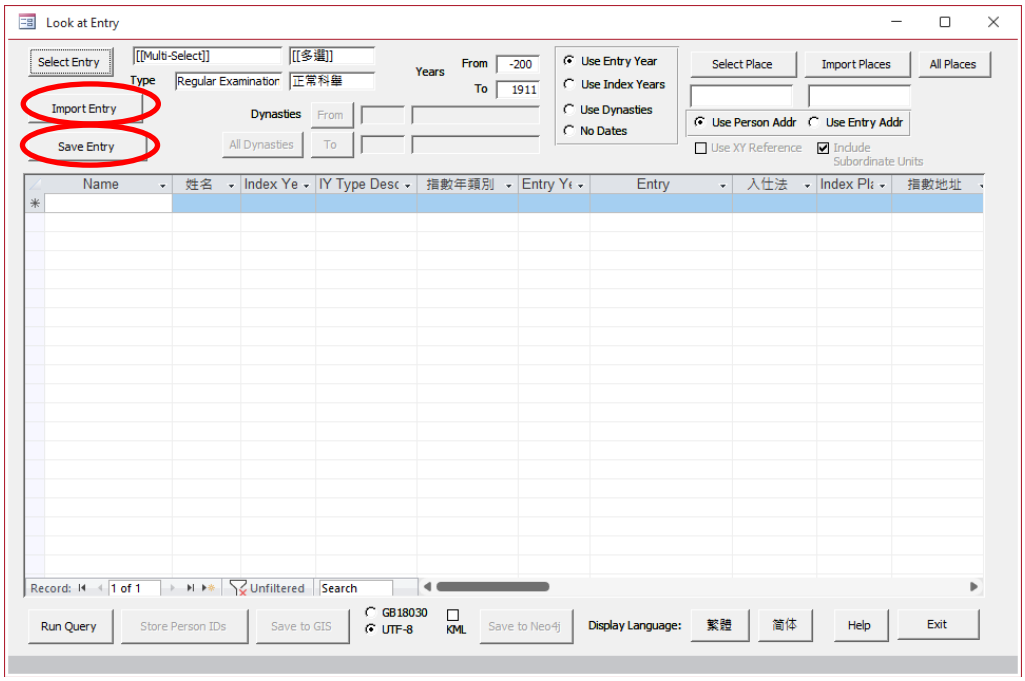
phrase at the *beginning* of the text and then look *within* the text. If CBDB finds the search, you can search for the *next* instance of the phrase (B2), if the first is not what you seek by clicking on the "Find Next" button:



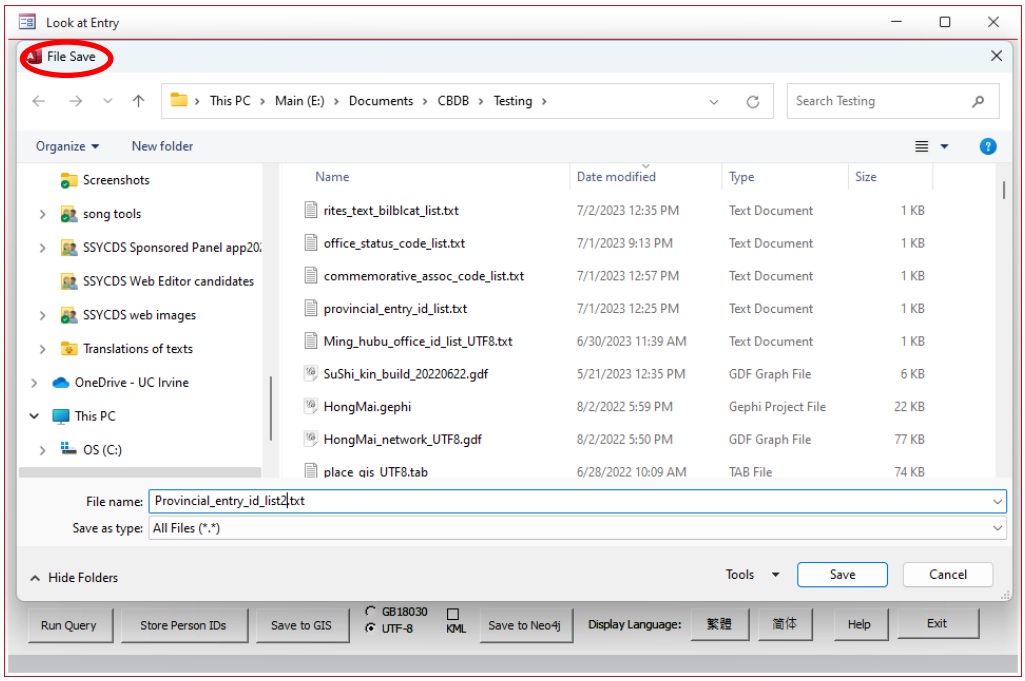
Often one wants to look at several categories within a selected type of mode of entry. CBDB now allows one to select one, two or more method of entry. Simply click to select or to un-select:



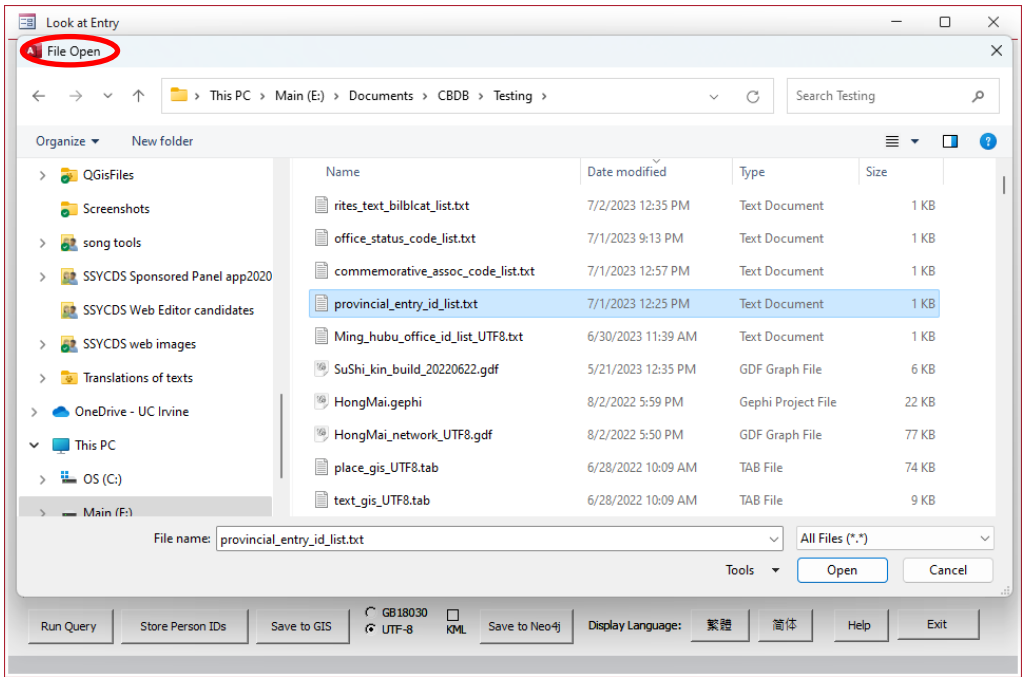
As of version BG of the interface, one can save the list of modes of entry one has selected. Saving the list gives one additional flexibility, since one can combine or edit lists for later use.



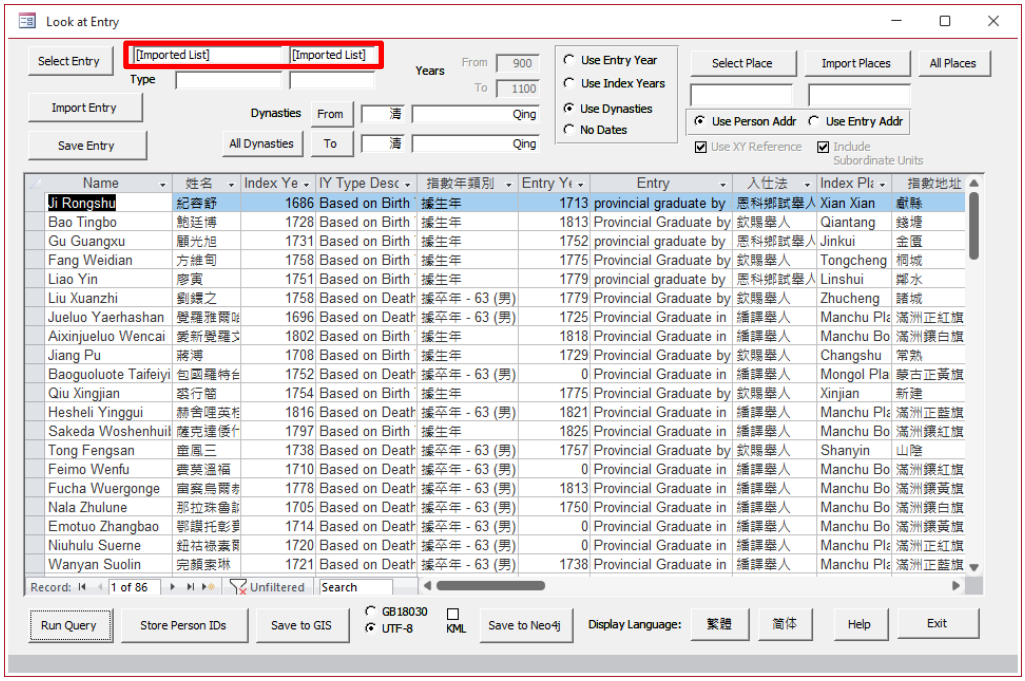
After selecting the combination of modes of entry one wants to explore, one can save them to a text file by clicking on **Save Entry**. This will open a "File Save" dialog box:



Once one has saved the file, it can be imported again by clicking on **Import Entry**. This will open an "File Open" dialog box.



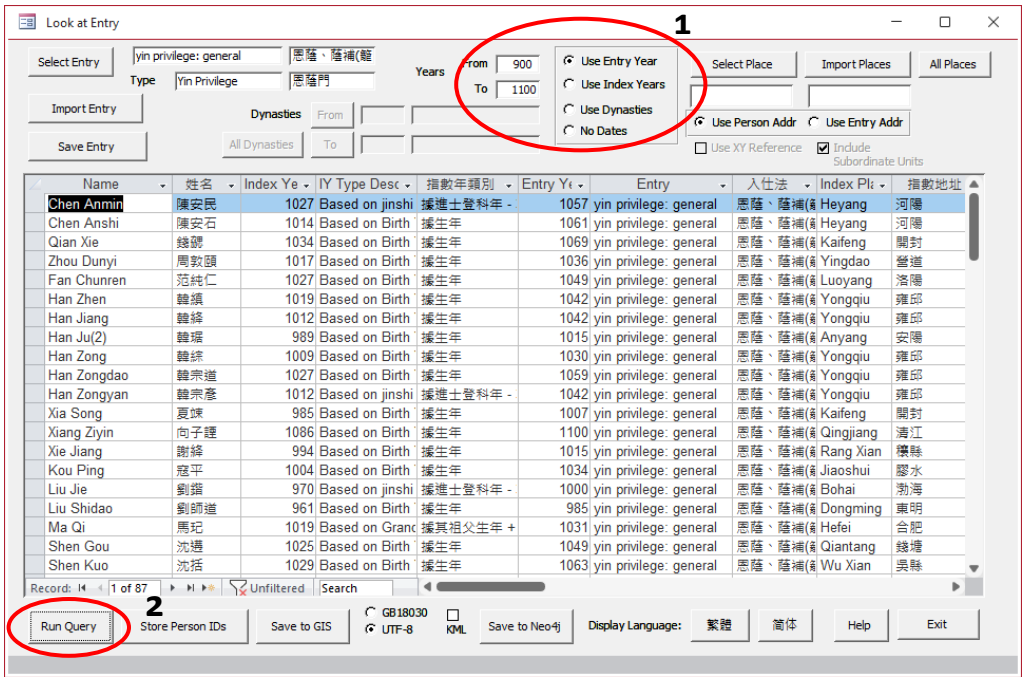
After selecting the file to import and clicking on **Open**, the form will show that it is using an imported list, and one can run the query using the list:



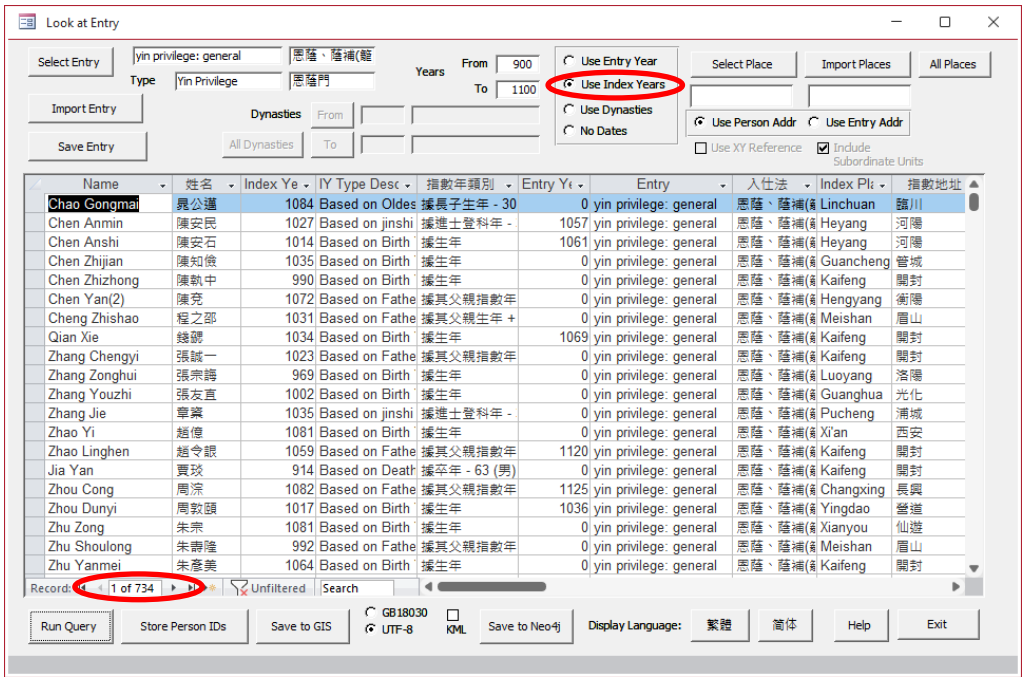
Setting Search Parameters

Setting Time Parameters

After one finds the method(s) of entry and clicks **Select**, one returns to the LookAtEntry form, and can now choose the year range (1) to run the query (2):

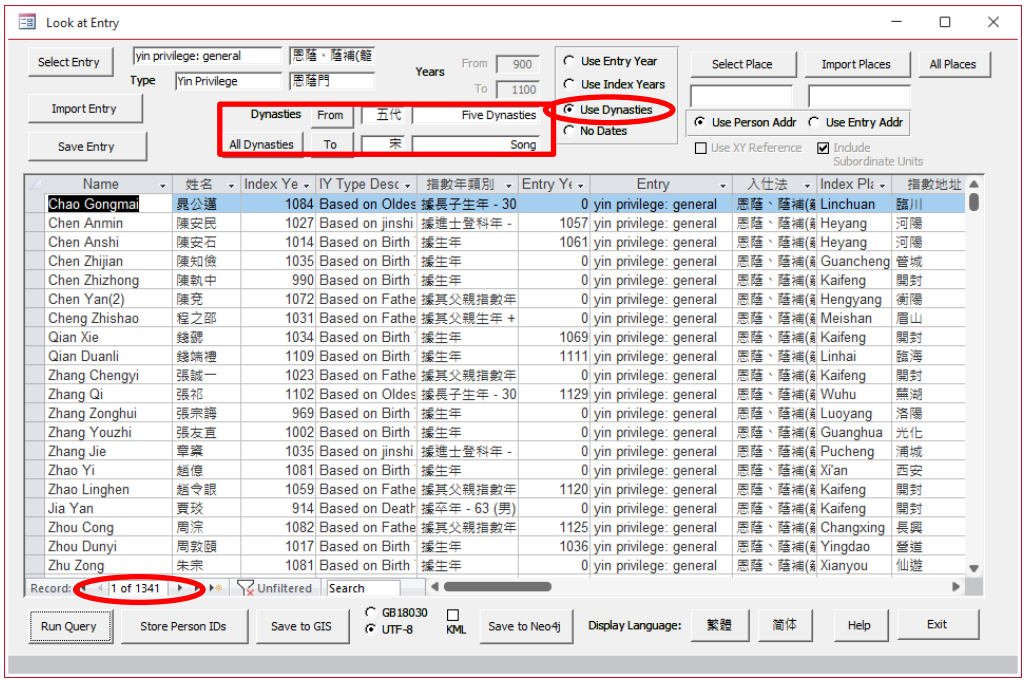


The form allows one to choose either the *entry year* or the *index year* of the person. (The index year is included in the search if the box "Use Index Years" is selected.) Because in many cases we do not know the entry year (given as zero in that case), it may prove useful to run the same query with the "Use Index Years" option selected:



Note the number of 0s in the "Entry Year" column. This approach yields 734 records, compared with just 87 when using the entry year. However, there are people for whom we know the dynasty but do not know their index year. The search by dynasty is less fine-grained:

although we are looking for the Five Dynasties and Northern Song Dynasty, the entire Song Dynasty has just one dynastic code. For this search, we identify 1,341 people, of whom 134 have no index year. However, 10 of those people do have years of entry:

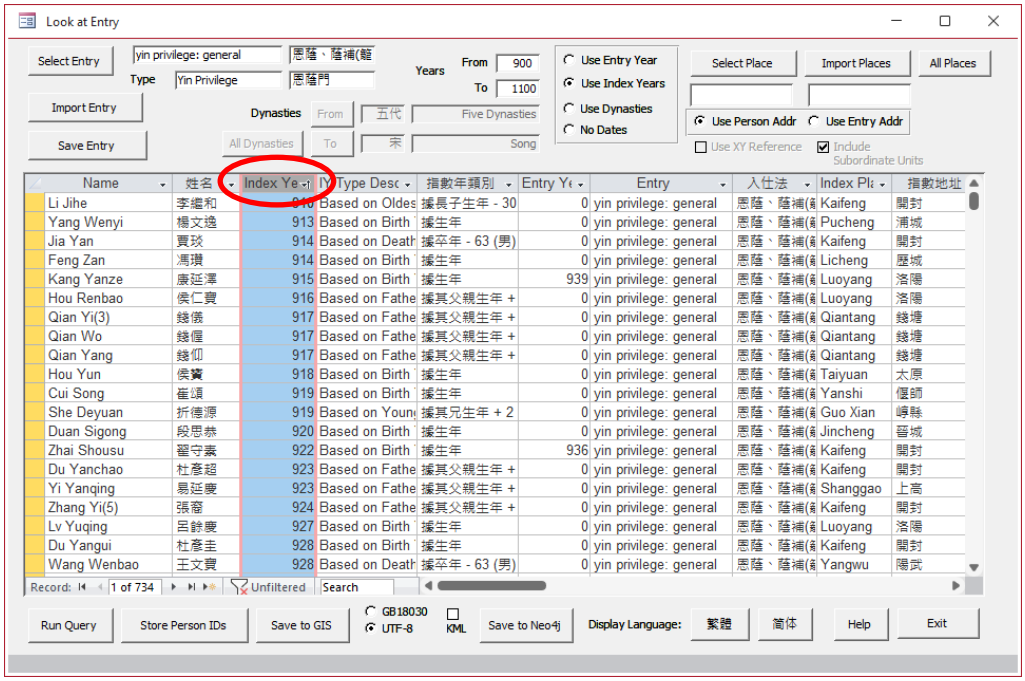


The table the query produces has 32 columns:

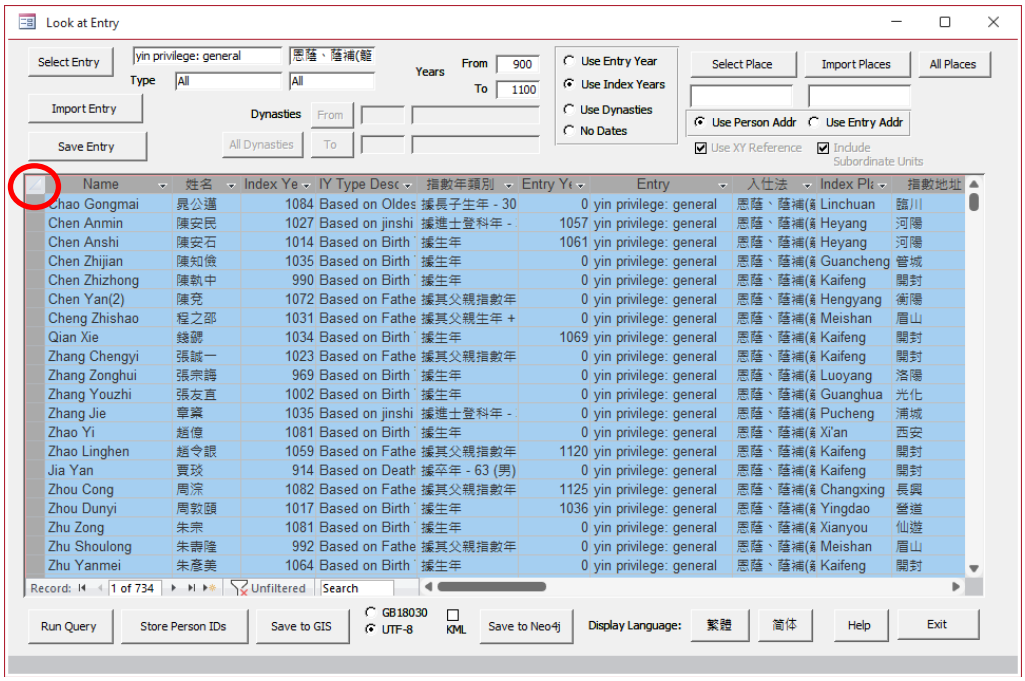
Name (Pinyin)	
Name (Chinese)	
Index Year	
Index Year Type (English)	(how the Index Year was derived)
Index Year Type (Chinese)	
Entry Year	
Description of Entry (English)	
Description of Entry (Chinese)	
Person's Index Place (Pinyin)	See discussion of index place on pages 22-23
Person's Index Place (Chinese)	
Type of Place Association	the type of place association used in assigning an index place
X-coordinate of Index Place	CBDB uses administrative seats
Y-coordinate of Index Place	
Count of XY coordinates	CBDB calculates how many people in the table share the same index place
Exam Rank	Given, if known. It does not apply to entry through *yin* privilege.
Kinship Relation	*Yin* privilege allows a person to become eligible for service based on the merit of a relative. If CBDB knows who that relative was and what the relationship was, the table gives this information
Kin Name (Pinyin)	
Kin Name (Chinese)	

Associate Name (Pinyin)	Sometimes people are granted entry into government through recommendation or through the role of some other non-kin associate
Associate Name (Chinese)	
Association	The type of association
Person ID	
Dynasty (English)	
Dynasty (Chinese)	
Index Year Type Code	
Parental Status (English)	For those sources that note whether the parents are alive at the time of passing the examination.
Parental Status (Chinese)	
Place of Entry (Pinyin)	
Place of Entry (Chinese)	
Place of Entry X-coordinates	
Place of Entry Y-coordinates	
Place of Entry XY Count	

One can **sort** the table using any of these columns. For example, "Index Year" may be useful. *Left-Click* on the column name "Index Year" to select the column and then *Right-Click* to choose from the sorting options:



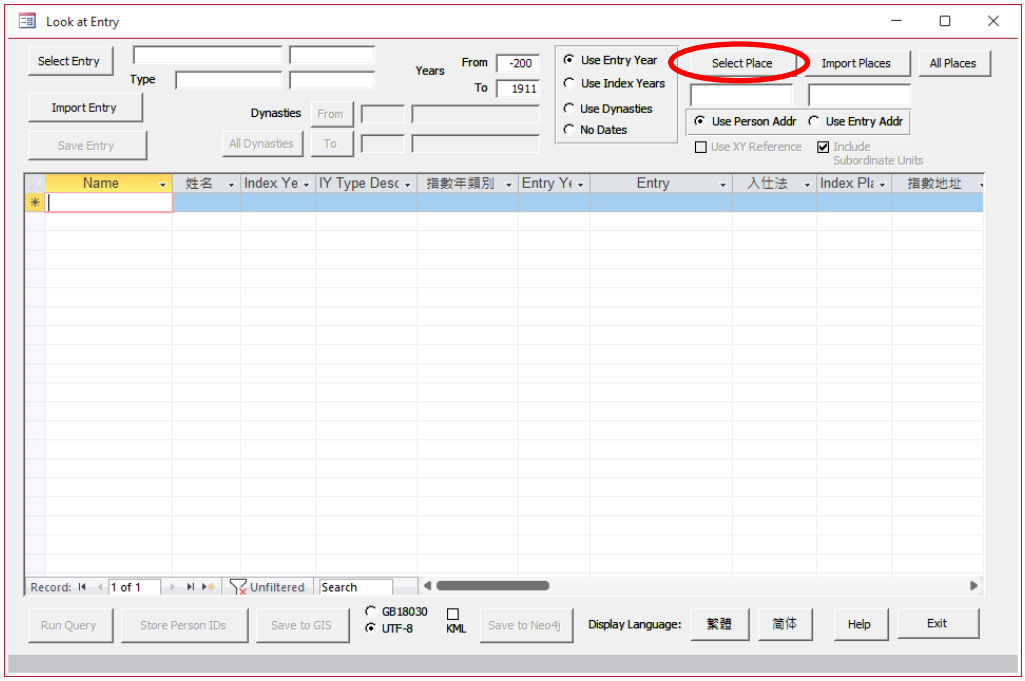
If one wishes to save the table, the simplest method is to **select** the entire table by clicking on the small box in the upper left-hand corner. Then save to the clipboard with **Ctrl-C**.



One then can paste the table into any program that accepts the format.

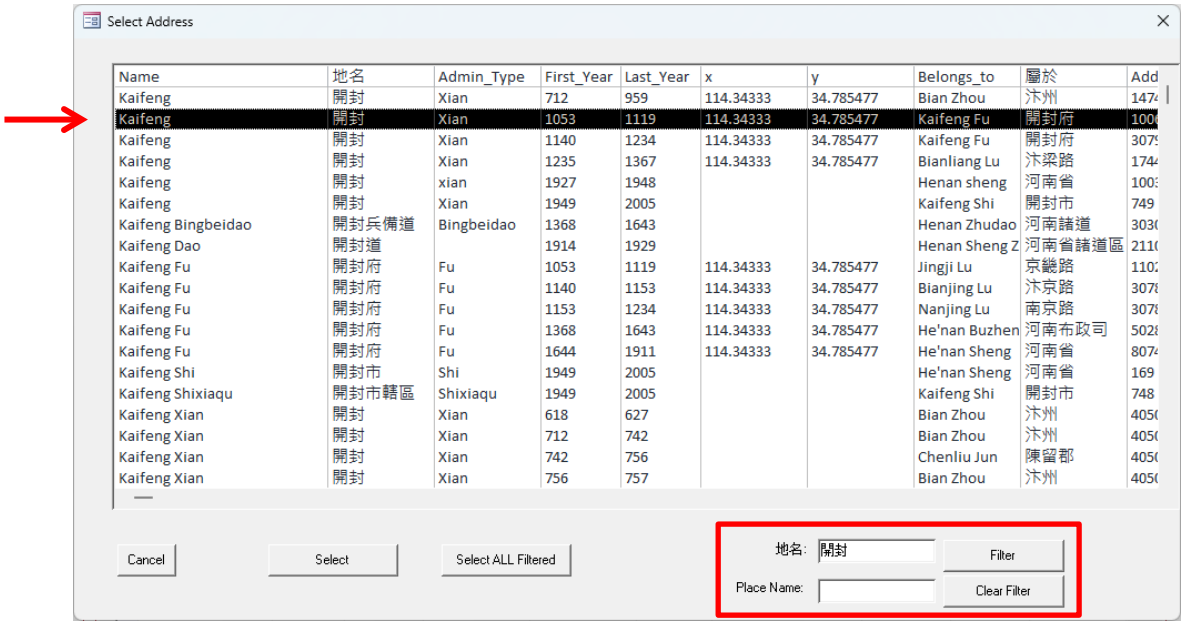
Setting Place Parameters

If one wishes to explore the mode of entry for people from a particular region, one uses the **Select Place** button in the upper right part of the form:

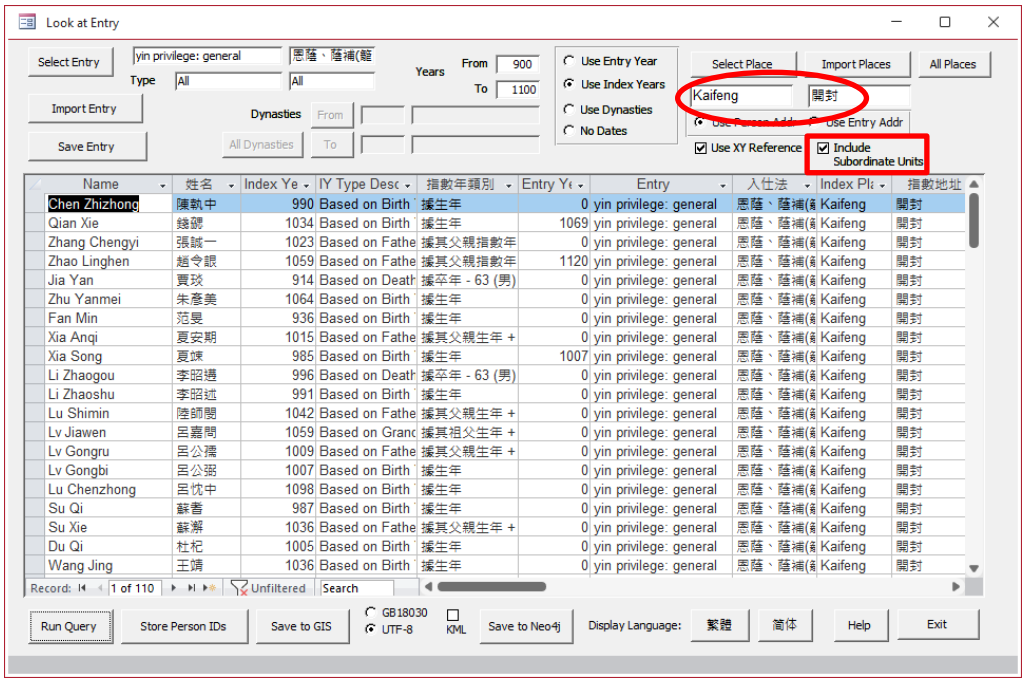


This opens the **Select Address** form. One can search for a place name using the filter box: to filter by Kaifeng 開封, enter "Kaifeng" into the Filter text box and then click the **Filter**

command button. This gives a list of all places that begin with the word "Kaifeng." (Using the Chinese, here 開封, is better to avoid the possibility of homonyms.)

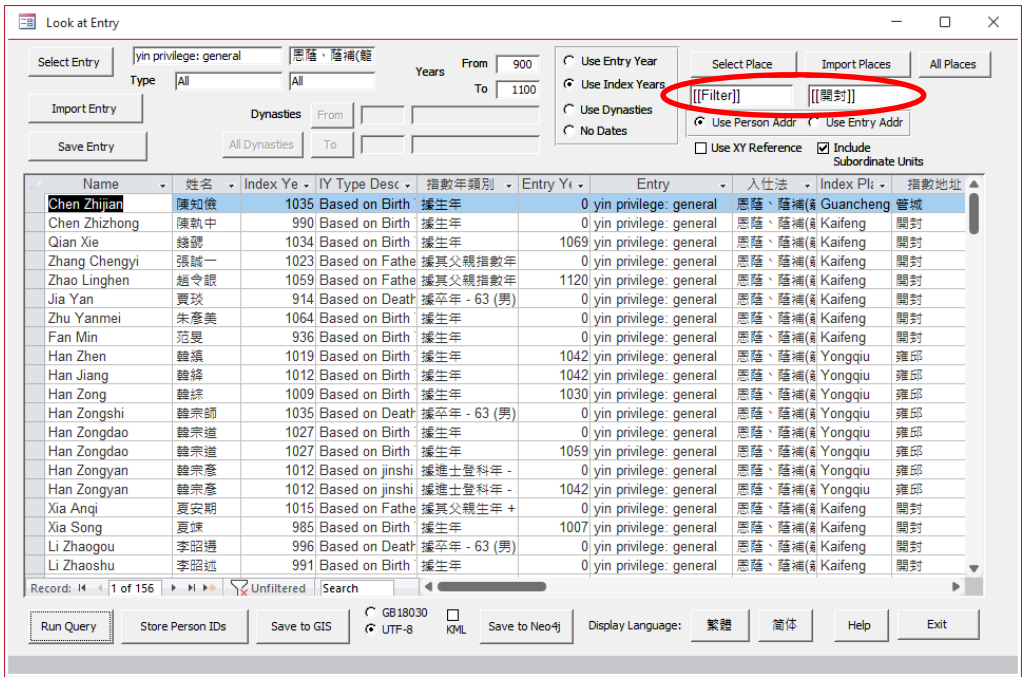


Note that there are many addresses for Kaifeng. We will select the Kaifeng county active from 1053 to 1119 by clicking on the corresponding row and clicking **Select; this** will return us to the main window, where we can run another query.



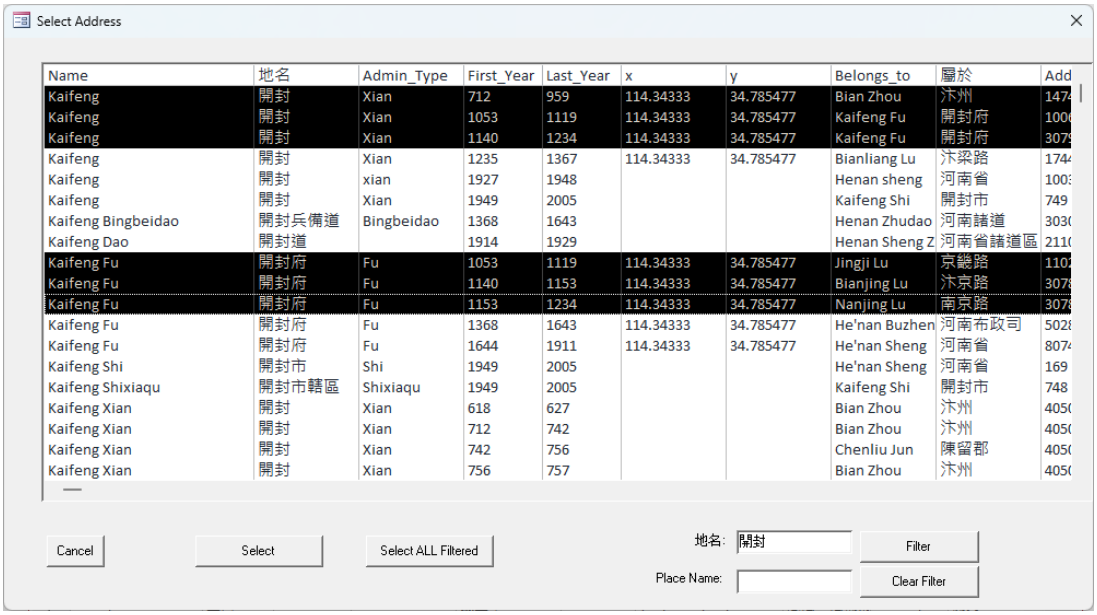
Because the index years start in 900, before the county code, we select "Use XY Reference" to *include codes for the region before 1053*. Using the single address for Kaifeng produces 110 people. If one wishes to look at Kaifeng more broadly, return to the **Select Address** form and once again enter "Kaifeng" into the Filter text box and then click the **Filter** command button. Then

either select ALL the filtered addresses by clicking on the "Select ALL Filtered" button. This will return you to the main LookAtEntry form, with all the Kaifeng codes selected; by including the prefecture (Kaifeng Fu) all its subordinate counties will be included:

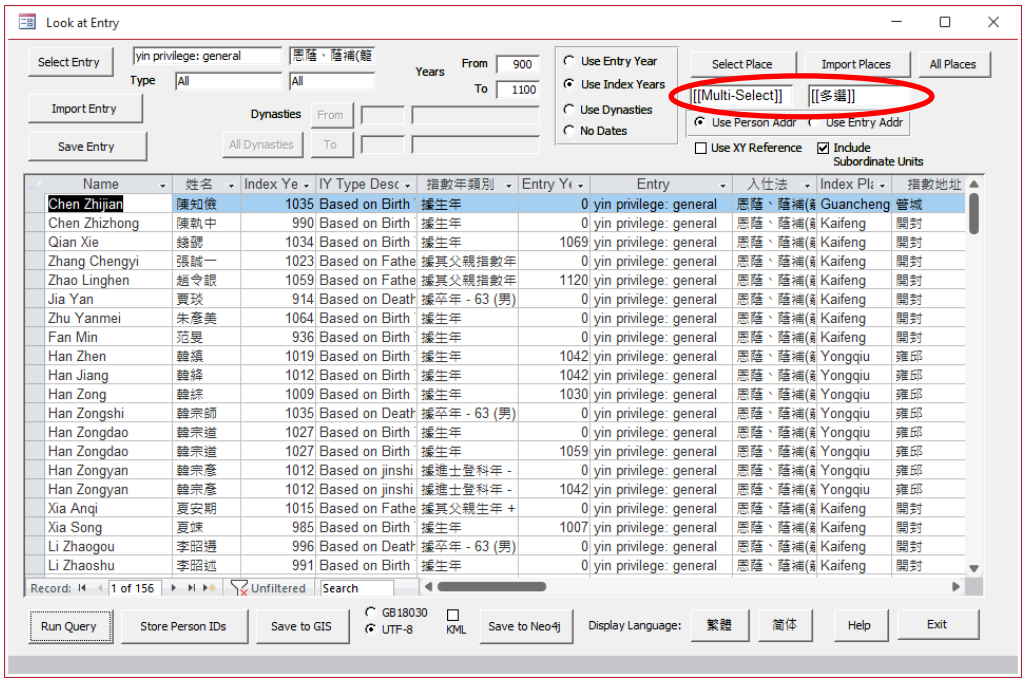


The **Place** text fields will show the filter term in a pair of square brackets, here "[[Kaifeng]]." Running the query now produces 156 records for people from Kaifeng whose index years were between 900 and 1100 and who entered government service through the *yin* privilege.

If *all* of the address IDs for Kaifeng are too many, one can limit the number of codes in the **Select Address** form by selecting just those that are relevant and then click on "Select:"

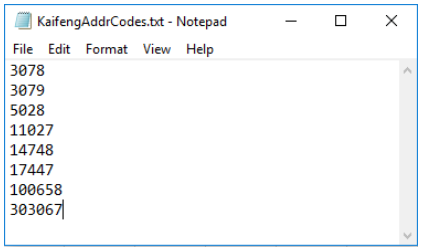


When one selects more than one address, the **Query by Method of Entry** form will have "[[Multi-Select]]/[[多選]]" instead of a place name. A search for *yin* privilege for people with index years between 900 and 1100 using these address codes produces 159 records.



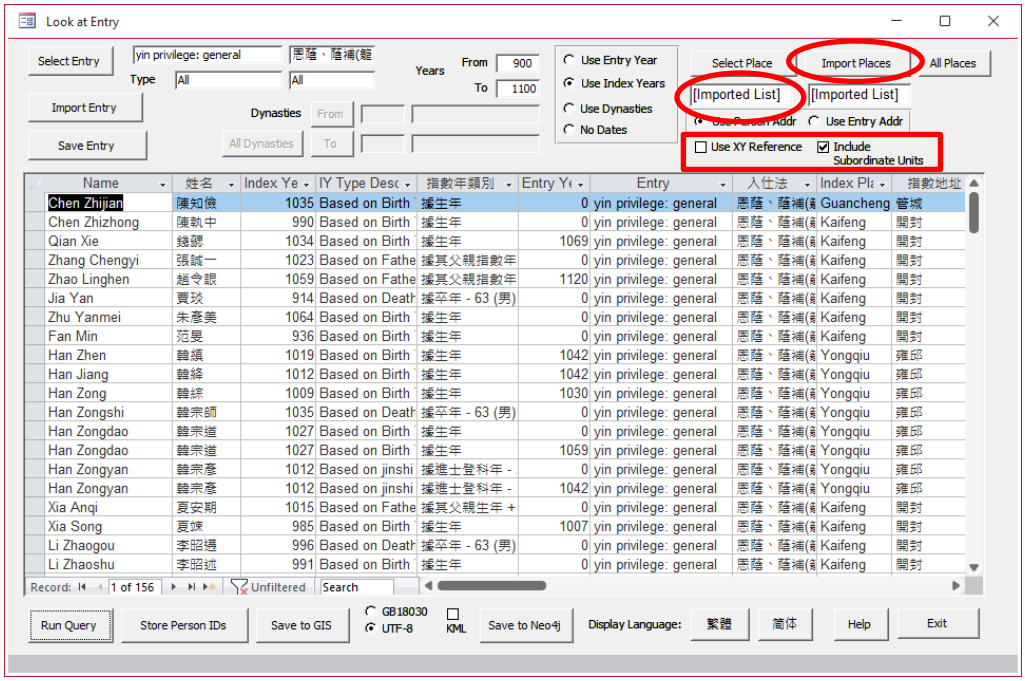
If one wishes more precisely and flexibly to control the address codes in one's search, one can create a text file with a list of address codes. Using the example of Kaifeng, for instance, one can select all the filtered records in the **Select Address** form, paste the records into a new Word or Excel file, delete the records one does not want, and copy the Address IDs to text file.

	A	B	C	D	E	F	G	H	I	J
1	Name	地名	First Year	Last Year	Admin Type	Belongs to	屬於	X	Y	Address ID
2	Kaifeng Fu	開封府	1127	1234	Fu	Nanjing Lu	南京路	114.34333	34.785477	3078
3	Kaifeng	開封	1127	1234	Xian	Kaifeng Fu	開封府	114.34333	34.785477	3079
4	Kaifeng Fu	開封府	1368	1643	Fu	He'nan Buzhengsi	河南布政司	114.34333	34.785477	5028
5	Kaifeng Fu	開封府	1053	1119	Fu	Jingji Lu	京畿路	114.34333	34.785477	11027
6	Kaifeng	開封	712	959	Xian	Bian Zhou	汴州	114.34333	34.785477	14748
7	Kaifeng	開封	1235	1367	Xian	Bianliang Lu	汴梁路	114.34333	34.785477	17447
8	Kaifeng	開封	960	1126	Xian	Kaifeng Fu	開封府	114.34333	34.785477	100658
9	Kaifeng Bingbeidao	開封兵備道	1368	1643	Bingbeidao	Henan Zhudao	河南諸道			303067



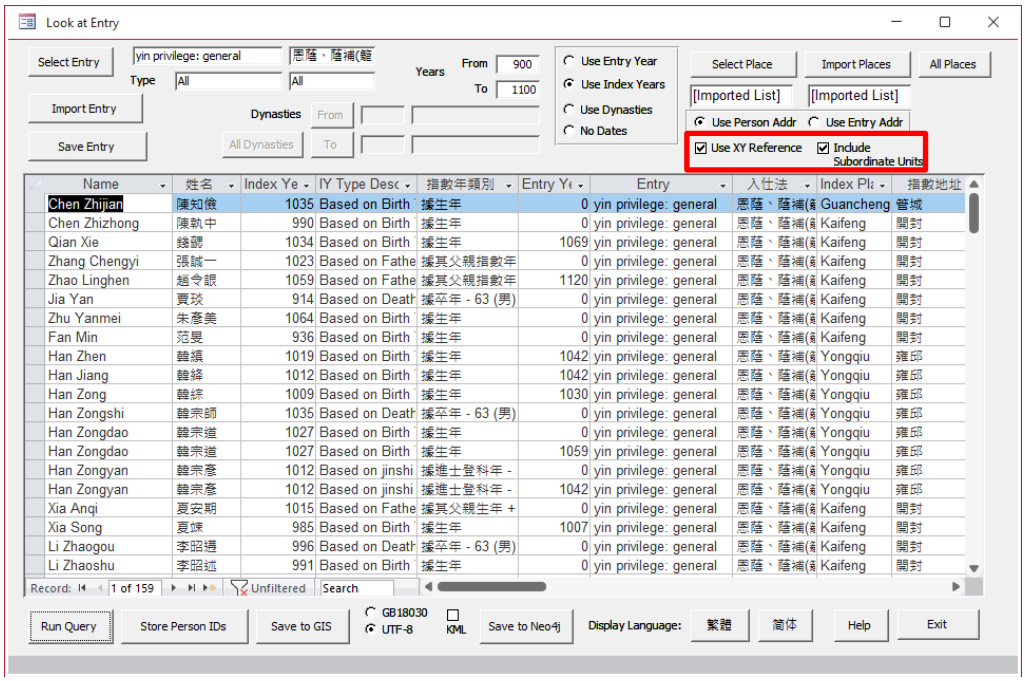
The importing routine checks the list against the address codes in ADDR_CODES and moves invalid codes to an ImportErrorList table for your inspection. (The table ImportErrorList is listed on the left-hand part of the Access screen. To view it, just double-click on it.)

Now click on the **Import Places** button in the LookAtEntry form and select the file to be imported. (CBDB gives a warning when it reads the list of IDs and finds an invalid ID.) If the import has been successful, one will see "[Imported List]" in the Place Information text boxes. Once the list has been imported, set the other parameters, and run the query.



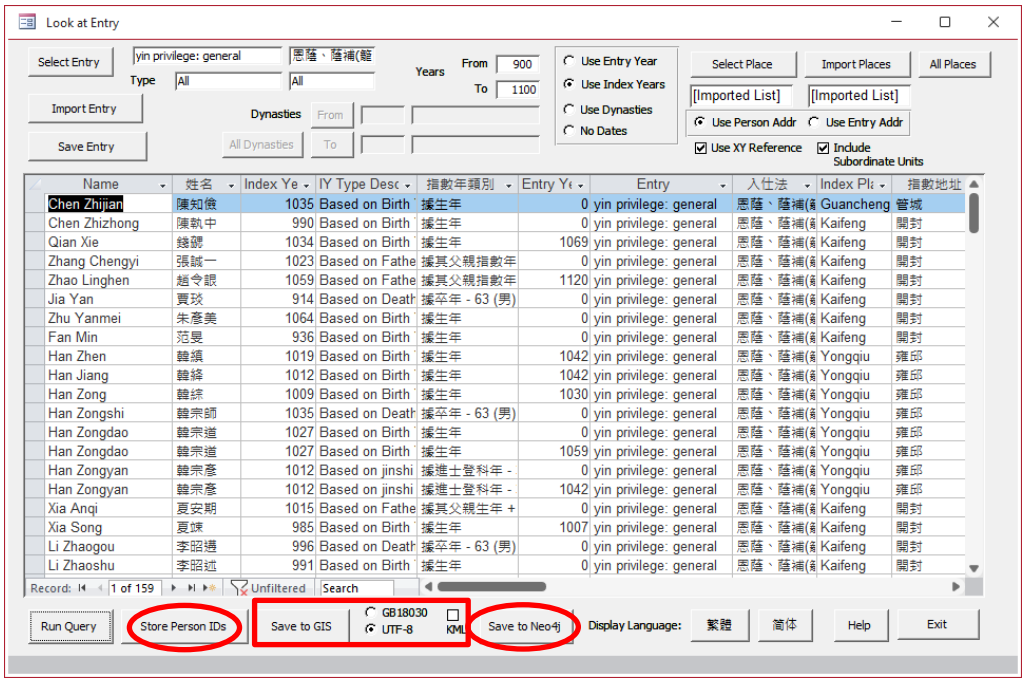
This approach produces 156 people, the same as the filtered version. However, note the **Include Subordinate Units** checkbox in the upper right corner. One of the places on the imported list was the Kaifeng Superior Prefecture 開封府: it has other counties subordinate to it that are included in the search when the checkbox is selected. This is the default setting. Note that the table includes people from Yongqiu 雍邱 and Guancheng 管城: these are administrative units subordinate to Kaifeng Superior Prefecture and included in the search. If one unclicks the **Include Subordinate Units** checkbox, these counties disappear from the search, which then produced just 105 people.

There is one additional approach to searching by name that handles *the problem of when a place name might change and thus be excluded from a list using names*. During the Later Jin (936-947), for example, Kaifeng was called by its old name, Bianzhou. CBDB allows one to pick one address (or a filtered or imported list of addresses) and, based on its longitude and latitude, to find all the administrative units throughout the specified time period that were close to that unit. If one imports the list of address codes for Kaifeng above and checks the **Use the XY Reference** checkbox as well as the **Include Subordinate Units checkbox**, one finds 159 people.



Saving Results

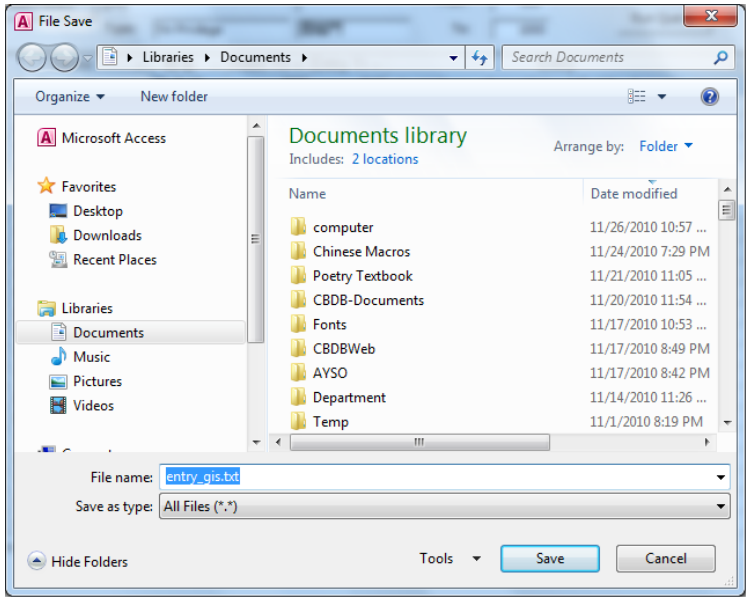
If one has created a query that produces a list of people who one wants to reuse in other queries—for example, if one wants to look at the kinship networks for the officials from Kaifeng who entered government service through *yin* privilege—one can store the list of people for reuse in the forms that use Person IDs as input (LookAtKinship, LookAtNetworks, LookAtAssociationPairs).



Simply click on the **Store Person IDs** button in this form and then click on the **Recall Person IDs** in the other form.

If one wishes to save the results to a file readable by a GIS program, one selects the coding for the file and clicks on the **Save to GIS** button at the bottom of the form. The table is saved to a text file, by default, "entry_gis.txt." One can specify the encoding of the text file as either GB18030 or UTF-8.

CBDB results also can be saved in KML format, the standard for importing CBDB query results into Google Earth.



Starting with user interface version BF, CBDB now also allows one to save the results of a search to a set of files that can be opened with Neo4j or other graph databases (see https://neo4j.com/developer/graph-database/). One selects the encoding and then clicks on **Save to Neo4j**. For the Kaifeng search in our example, the program creates *seven* .csv ("comma-separated values") files:

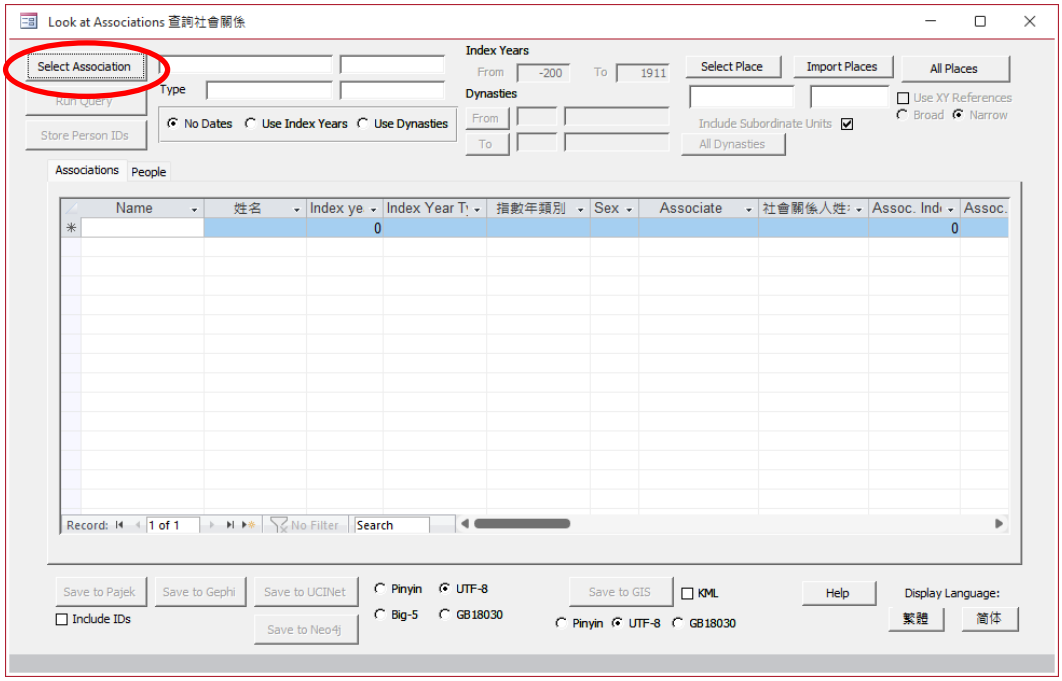
EntryCode_UTF8.csv
KinshipCodes_UTF8.csv
People_UTF8.csv
PeopleEntry_UTF8.csv
PeoplePlaces_UTF8.csv
PeoplePlaceCodes_UTF8.csv
Places_UTF8.csv

The seven files create graph representations of the various aspects of the data created through the search.

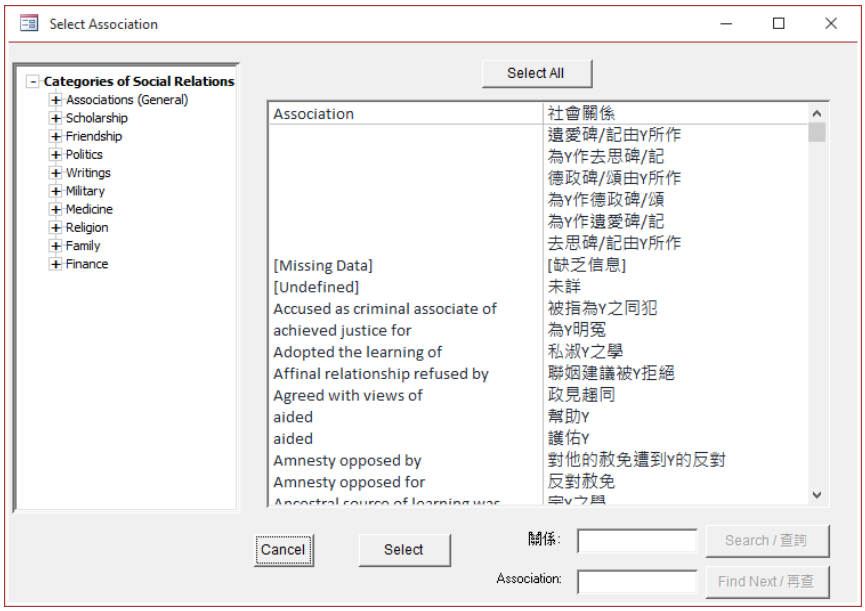
All the search forms support output to Neo4j but vary in the number of files created, depending on the nature of the data collected through the forms' queries.

D. Using the Form "Query Associations"

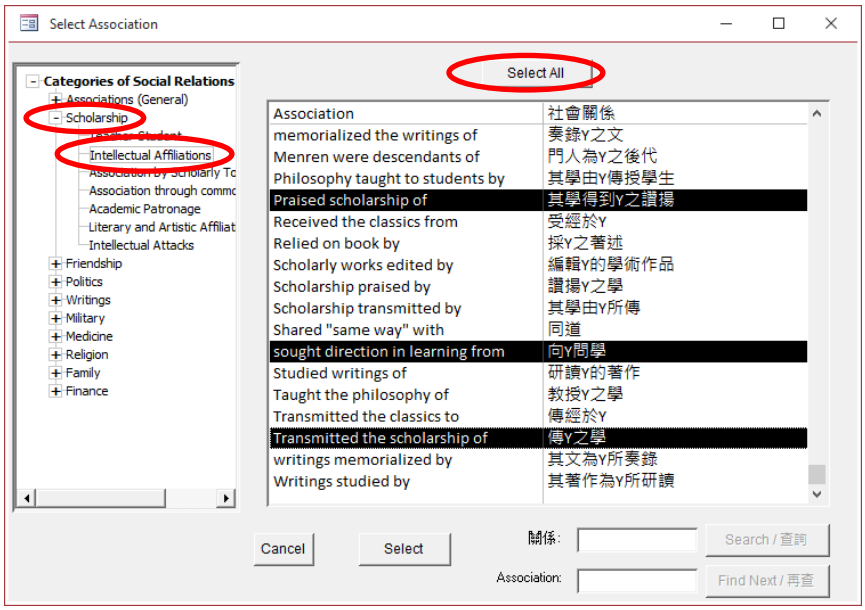
LookAtAssociations allows one to look at the people who have participated in particular associations or categories of associations. After opening the form, one clicks on "Select Association" to choose the type of association one wants to investigate.



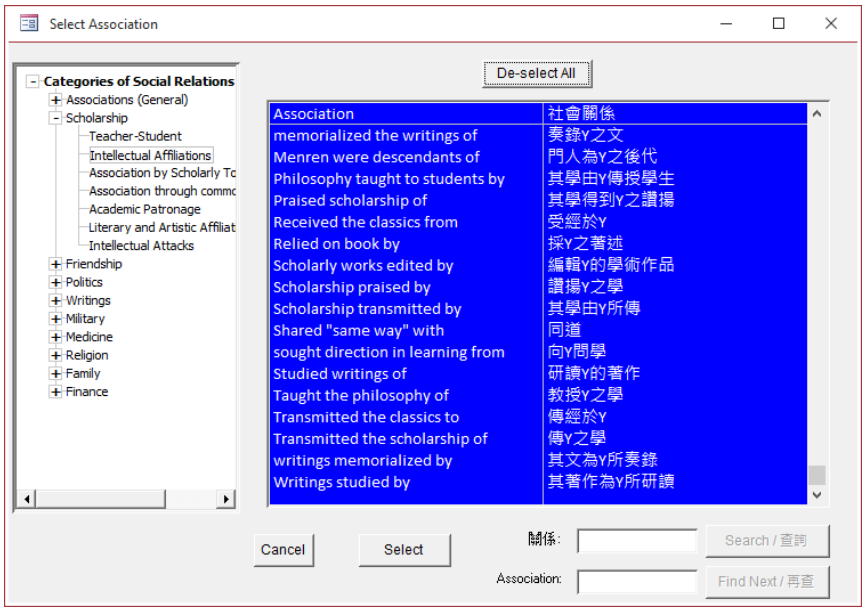
There are over four hundred categories of associations, so CBDB allows one to pick by type and subtype.



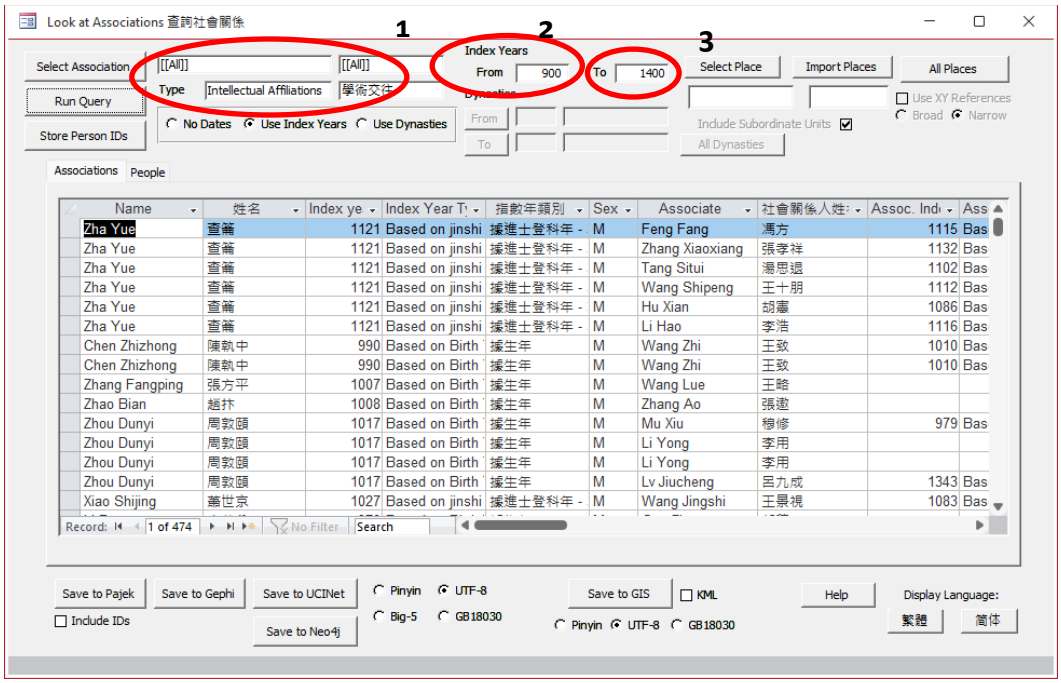
Consider the "Scholarship" associations from the list on the left. Under "Scholarship" there are seven subtypes. The subtype "intellectual affiliations" in turn has seventeen categories of associations. As with selecting entry codes, one can select the relevant codes.



One also can select all of the subtypes simply by clicking on the "Select All" command button at the top and then clicking on the "Select" button at the bottom center:

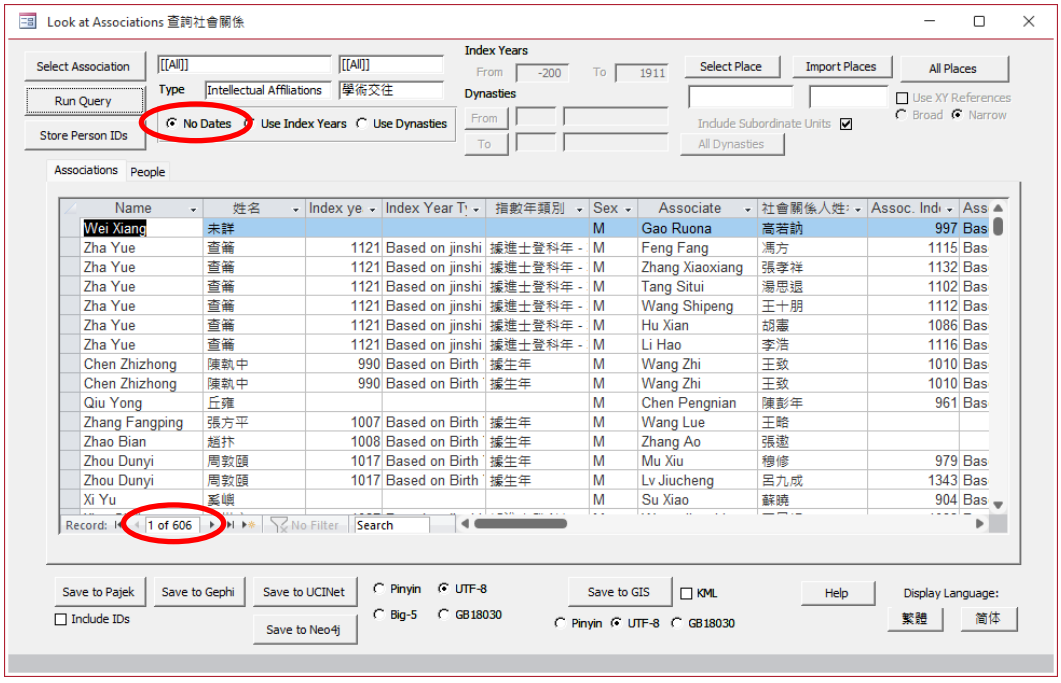


Note that, as with entry, one can search for terms in the table of associations in both English and Chinese (using the search box at the bottom right corner) and search again if the first item found is not what you are looking for. In the screenshot below, I have selected *all intellectual affiliation associations* (1) as the subtype of association. I then chose the year between 900 and 1400 (2), and ran the query (3):



Although dates are a part of the ASSOC_DATA table, we do not have date information for most associations, and **LookAtAssociations** uses the ***index year*** of the individuals to see whether they fall within the specified beginning and end dates.

Using the index year of people, however, introduces a significant limitation at the same time that it allows one to focus on specific timeframes: people for whom CBDB does not have an index year simply disappear from the results. This CBDB allows one to search without using the index years by unchecking the **Use Index Years** box directly below the input boxes for years:



Note that the results rise from 474 to 606. One can sort on index years after doing the search to look for the relevant associations.

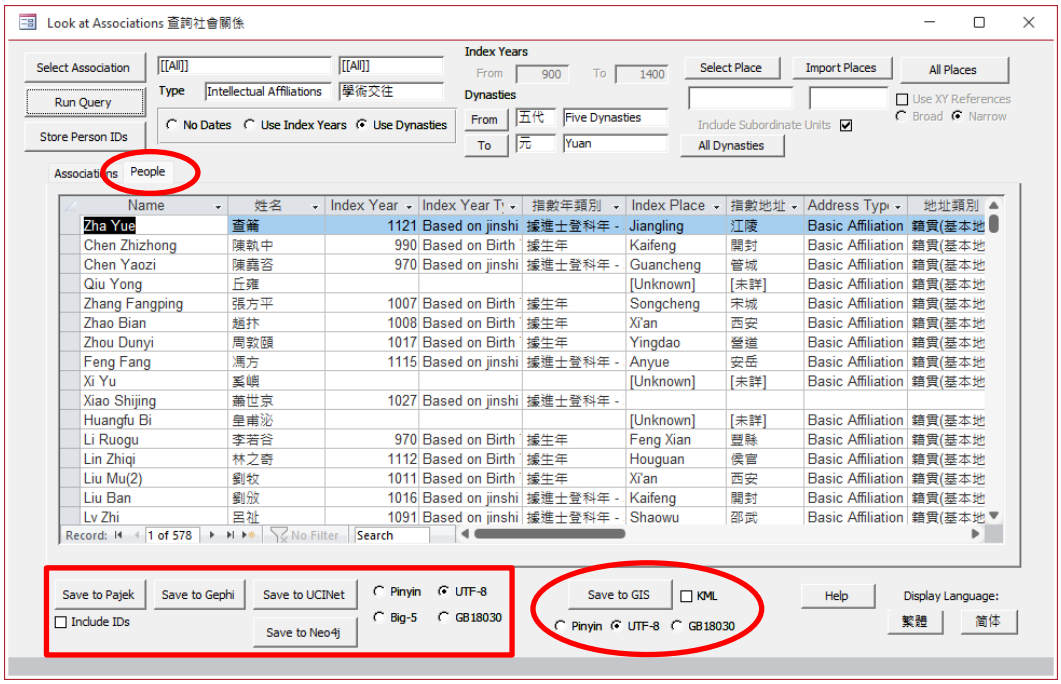
The **Associations** table in **LookAtAssociations** has 40 columns to display the types of information recorded in theASSOC_DATA table:

Name (Pinyin)	
Name (Chinese)	
Index Year	
Sex	
Associated Person's Name (Pinyin)	
Associated Person's Name (Chinese)	
Associated Person's Index Year	
Associated Person's Sex	
Association Category (English)	
Association Category (Chinese)	
Association Count	This gives the number of objects or events that established the association
Address (English)	This is the index place, if known.
Address (Chinese)	
X-coordinate	These are the coordinates for the address above.
Y-coordinate	
Associate's Address (English)	
Associate's Address (Chinese)	
Associate's X-coordinate	
Associate's Y-coordinate	
Kinship Relation (English)	The next four columns are for associations created through actions for the sake of a kin
Kinship Relation (Chinese)	
Kin Name (pinyin)	
Kin Name (Chinese)	
Associate's Kinship Relation (English)	The next four columns are for associations created through actions for the sake of the associate's kin
Associate's Kinship Relation (Chinese)	
Associate's Kin Name (pinyin)	
Associate's Kin Name (Chinese)	
Index Year Type (English)	
Index Year Type (Chinese)	
Dynasty (Pinyin)	
Dynasty (Chinese)	
Associate Index Year Type (English)	
Associate Index Year Type (Chinese)	
Associate Dynasty (Pinyin)	
Associate Dynasty (Chinese)	
Distance	If CBDB has the coordinates for the place identification for both people, it calculates the great-circle arc distance between them (in kilometers).
Index Type Code	
Dynasty Code	
Associate Index Type Code	
Associate Dynasty Code	

In addition to the table of associations, **LookAtAssociations** also provides a table listing all the people involved in the association one is investigating. One views this table simply by clicking on the **People in Association tab**. This table provides information about association with place.

This table has 19 columns:

Name (pinyin)
Name (Chinese)
Index Year
Index Year Type (English)
Index Year Type (Chinese)
Dynasty (Pinyin)
Dynasty (Chinese)
Sex
Index Place (pinyin)
Index Place (Chinese)
Index Place Type (English)
Index Place Type (Chinese)
X-coordinate
Y-coordinate
XY-count
Person ID
Index Year Type Code
Dynasty Code
Index Place Type Code



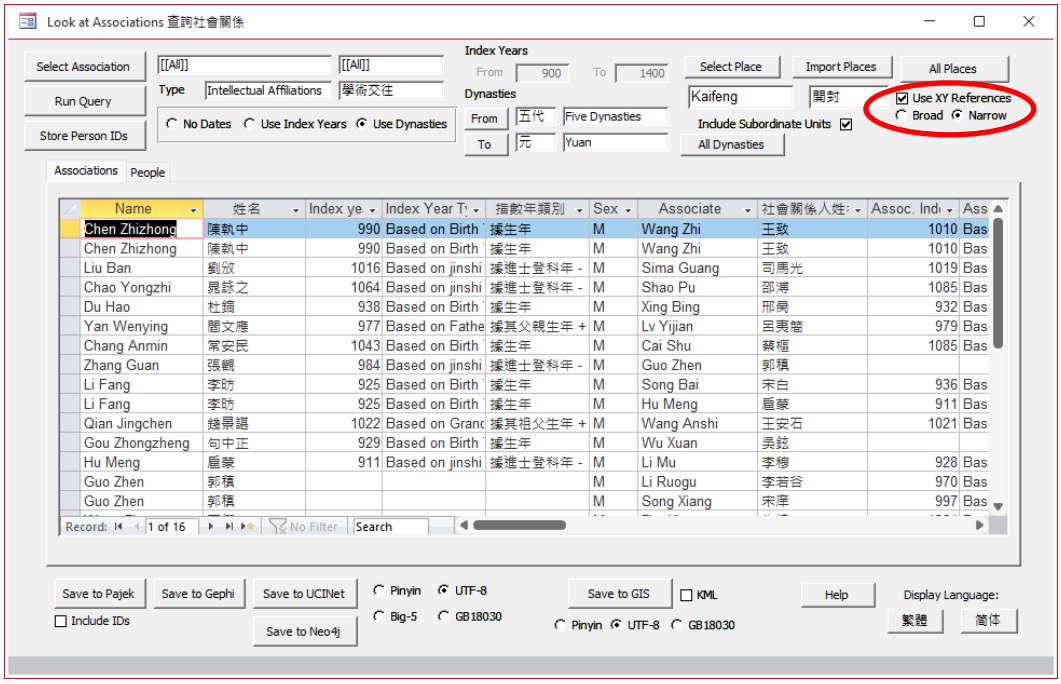
One can save the address information for display through a GIS program by clicking on the **Save to GIS**. Since association data provides an implicit social network linking the groups of people connected by the category of association being examined, one can save the network for

analysis in the Pajek format, for example, by clicking on the **Save to Pajek** button. Pajek is one standard format for visualization in social network analysis (SNA). In addition data can be saved to Gephi or UCINet, and many programs can read it and convert it to other formats. CBDB allows files for both GIS programs and for Pajek to be saved in different text encodings to enable the use of Chinese characters. Note that there is an option to include the Person ID with the node information in the Pajek files. Finally, one can also save the data to Neo4j files.

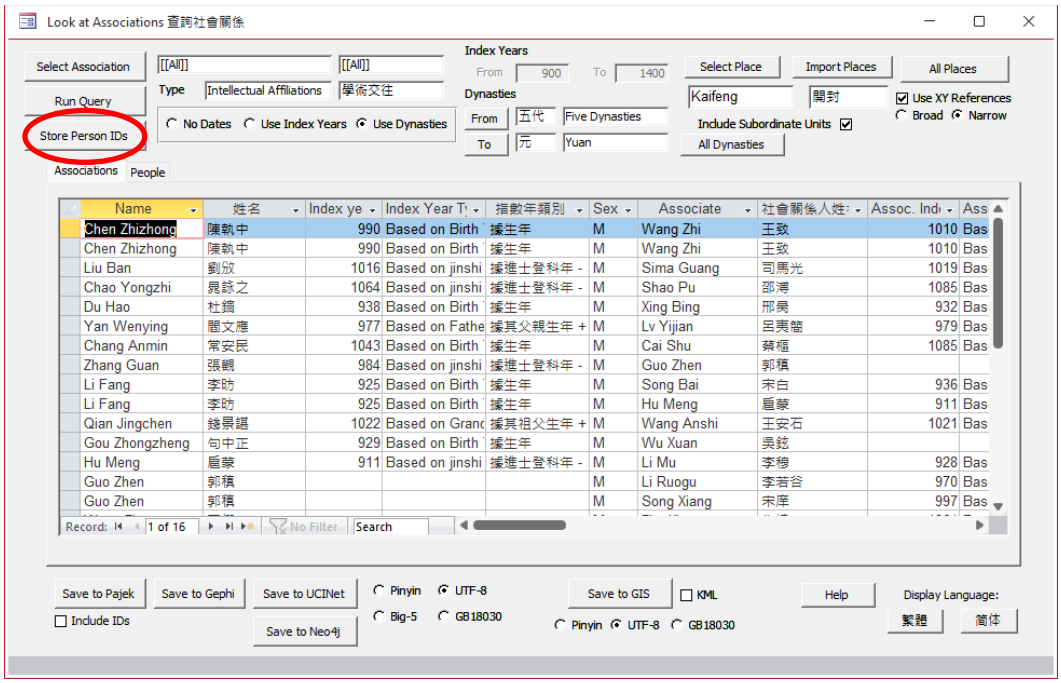
Search by Place

Like the **LookAtEntry** form, **LookAtAssociations** allows one to look at associations for people from a particular place or from a particular list of places.

Like the **LookAtEntry** form, **LookAtAssociations** has an option when searching for a specific place: search by **XY Reference**. One uses the XY coordinates of the selected administrative unit(s) to locate other units through the specified time span whose coordinates are close to those of selected place(s). In **LookAtAssociations**, one choose either a *narrow* bounding box to define administrative units close to the units one has chosen, or one can choose a slightly larger box that may include additional units by clicking on the radio button labelled **Broad** under the "Use XY References" check box. This feature is particularly useful when administrative units change name in a way that cannot be caught by simply filtering by name. In this case, CBDB uses the Kaifeng administrative unit in the Northern Song, and it turns out that the results are the same if one chooses either "Narrow" or "Broad:"

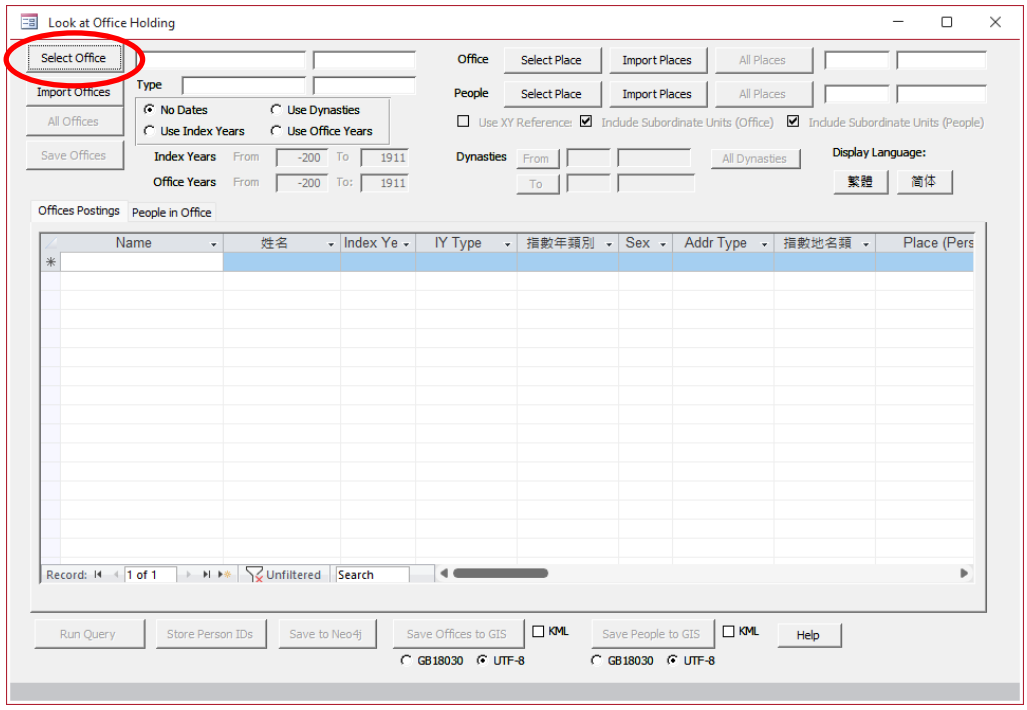


Like all other tables, LookAt Associations allows one to store the results of a query for later use in another form. One clicks on the "Store Person IDs" button.


Look at Associations 查詢社會關係
Select Association
Run Query
Store Person IDs
Type
Intellectual Affiliations
學術交往
No Dates
Use Index Years
Use Dynasties
Index Years
From
900
To
1400
Dynasties
From
五代
Five Dynasties
To
元
Yuan
Select Place
Import Places
All Places
Kaifeng
開封
Use XY References
Broad
Narrow
Include Subordinate Units
All Dynasties
Associations
People
Name
姓名
Index ye
Index Year Ty
指數年類別
Sex
Associate
社會關係人姓名
Assoc. Indi
Ass
Chen Zhizhong 陳執中 990 Based on Birth 據生年 M Wang Zhi 王致 1010 Bas
Chen Zhizhong 陳執中 990 Based on Birth 據生年 M Wang Zhi 王致 1010 Bas
Liu Ban 劉放 1016 Based on jinshi 據進士登科年 - M Sima Guang 司馬光 1019 Bas
Chao Yongzhi 晁詠之 1064 Based on jinshi 據進士登科年 - M Shao Pu 邵溥 1085 Bas
Du Hao 杜鎬 938 Based on Birth 據生年 M Xing Bing 邢昺 932 Bas
Yan Wenying 閻文應 977 Based on Fathe 據其父親生年 + M Lv Yijian 呂夷簡 979 Bas
Chang Anmin 常安民 1043 Based on Birth 據生年 M Cai Shu 蔡確 1085 Bas
Zhang Guan 張觀 984 Based on jinshi 據進士登科年 - M Guo Zhen 郭稹
Li Fang 李昉 925 Based on Birth 據生年 M Song Bai 宋白 936 Bas
Li Fang 李昉 925 Based on Birth 據生年 M Hu Meng 扈蒙 911 Bas
Qian Jingchen 錢景諶 1022 Based on Grand 據其祖父生年 + M Wang Anshi 王安石 1021 Bas
Gou Zhongzheng 句中正 929 Based on Birth 據生年 M Wu Xuan 吳鉉
Hu Meng 扈蒙 911 Based on jinshi 據進士登科年 - M Li Mu 李穆 928 Bas
Guo Zhen 郭稹 M Li Ruogu 李若谷 970 Bas
Guo Zhen 郭稹 M Song Xiang 宋庠 997 Bas
Record: 1 of 16
No Filter
Search
Save to Pajek
Include IDs
Save to Gephi
Save to UCINet
Save to Neo4j
Pinyin
UTF-8
Big-5
GB18030
Save to GIS
KML
Pinyin
UTF-8
GB18030
Help
Display Language:
繁體
简体

E. Using the Form "Query Offices Holding"

The bureaucratic system of imperial China was complex, and it evolved over time. As a result, CBDB at present has over six thousand office codes and will certainly have many more as the database extends its coverage to all of pre-modern China. Thus a central challenge in offering a useful approach to the examination of people's roles in office is how to aggregate the plethora of offices into larger units for analysis. **LookAtOffice** provides both hierarchical and functional groupings. When one opens **LookAtOffice**, it looks much like the other simple analytic forms. One clicks on the **Select Office** button on the top left to begin.

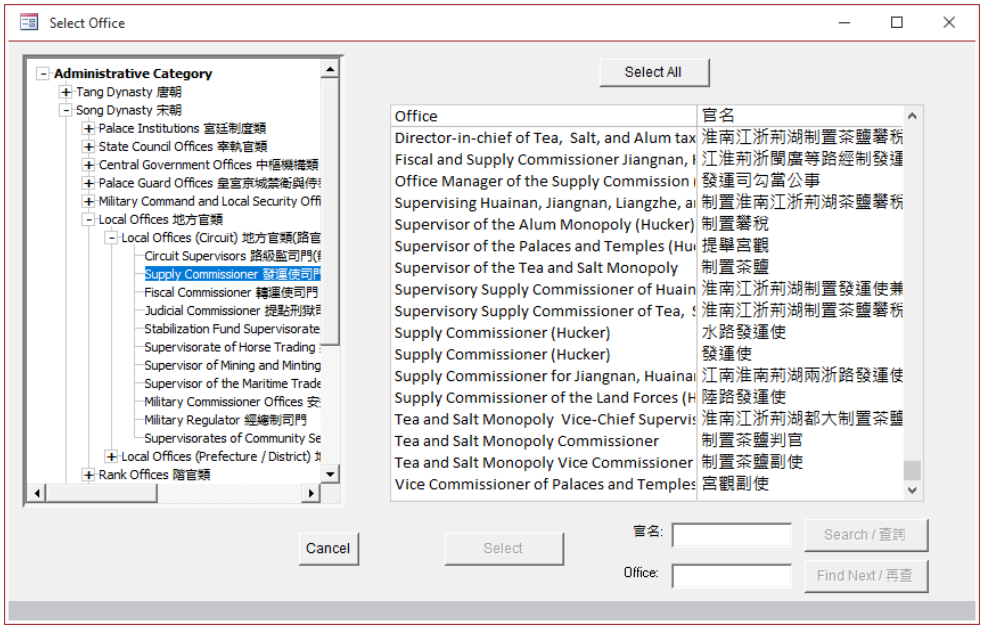


Selecting the Offices

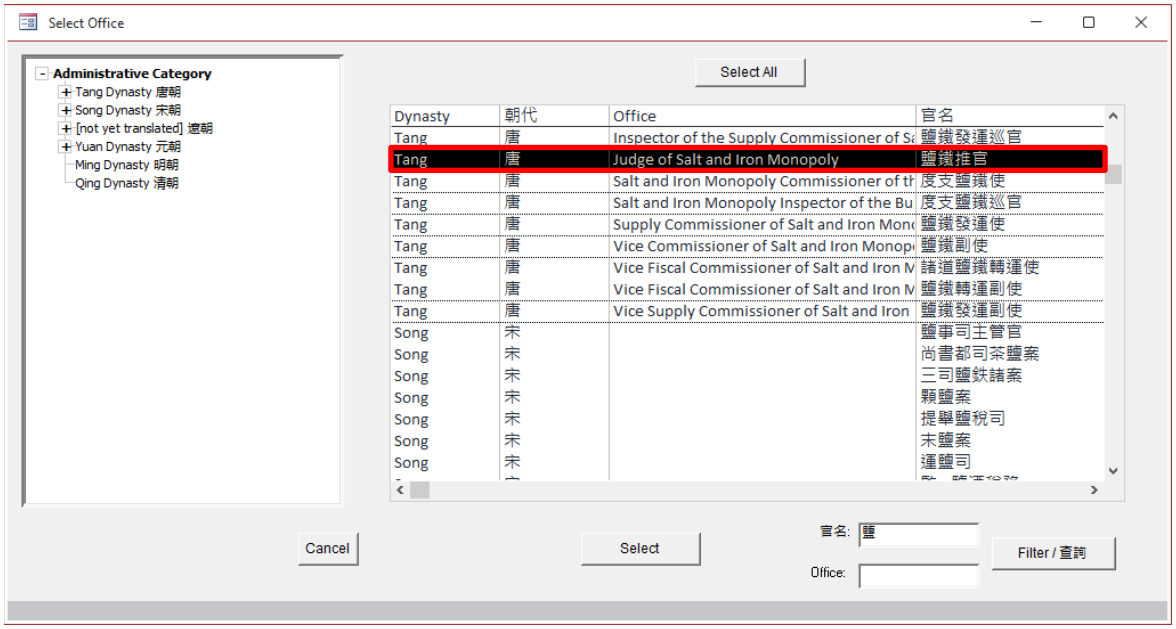
The "Select Office" form displays a tree of office categories. The first node, "Bureaucratic Structure" shows the organizational hierarchy organized by dynasty. One can view—and select—the offices at any level of structure. The first form shows all the offices associated with the Supply Commissioner at the circuit level.

One also can search for particular terms. Because there are not yet tree structures to explore bureaucratic organization of dynasties other than the Tang, Song, and Yuan, filtering by the Chinese term (most records do not yet have English equivalents) is the best way to find particular offices in other dynasties. Filtering for "salt" 鹽 (see the second form below) lists all titles of offices *in all dynasties*. One can scroll up (the form initially goes to the last record in the table) to find the Tang dynasty offices. As in other forms, one can pick more than one office title.

Offices in the Supply Commissioner's Office in the Song Dynasty



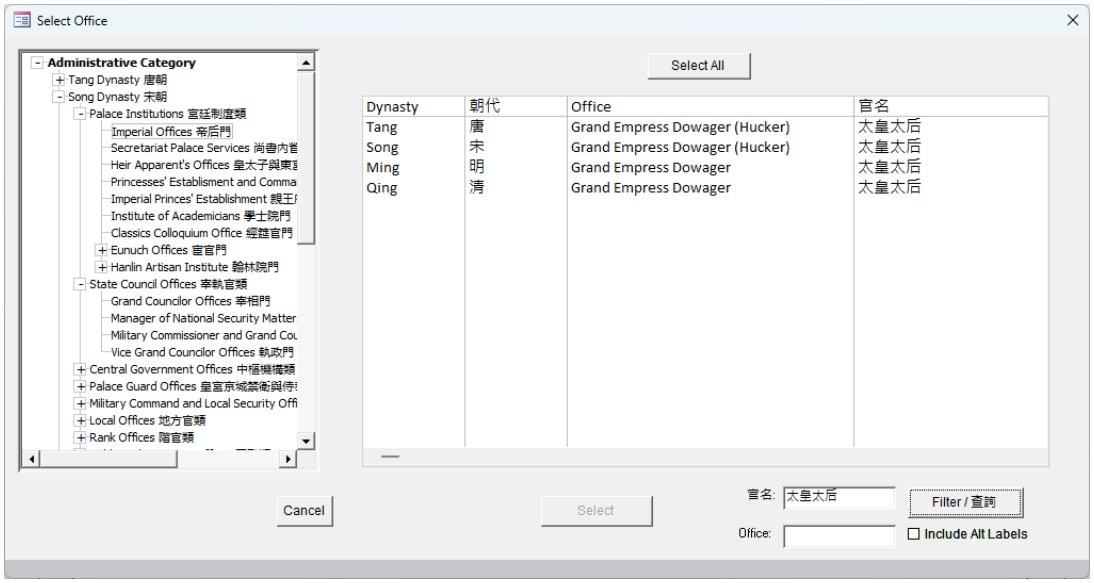
A Tang office containing the character 鹽 ("salt")



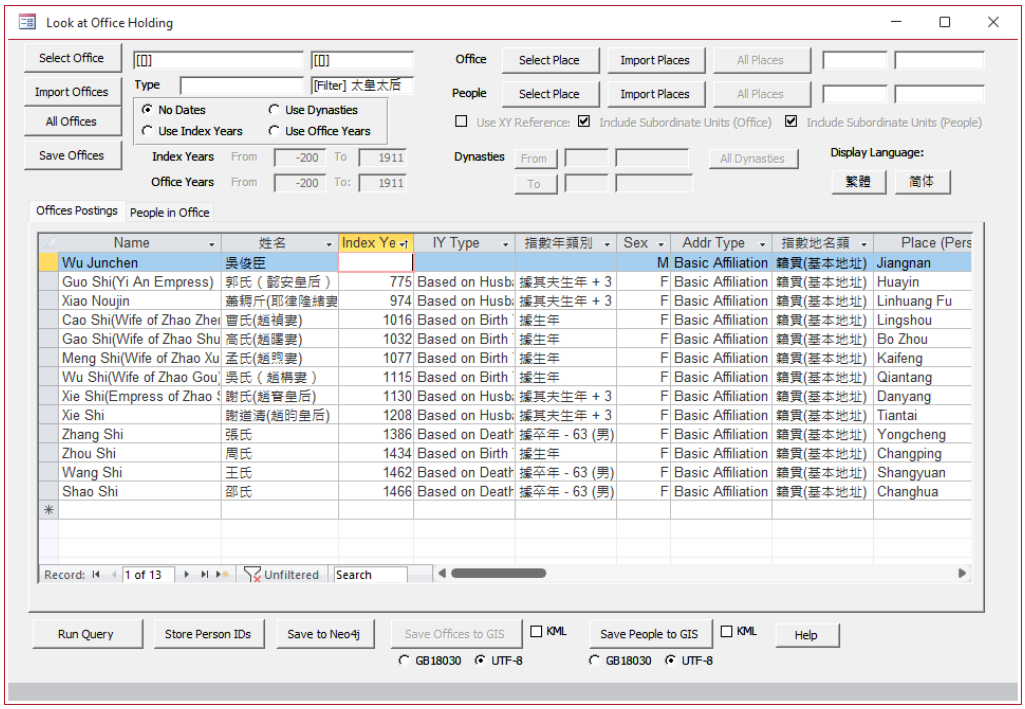
The filter allows one to select related offices across dynastic boundaries. One can select individual offices related to salt, or one can select all offices.

Cross-dynastic Office Selection

One can use the filter for office names to find a set of offices that crosses dynastic boundaries and, therefore, allows one to make cross-dynastic comparisons. For example, one can search for all the Grand Empress Dowagers:

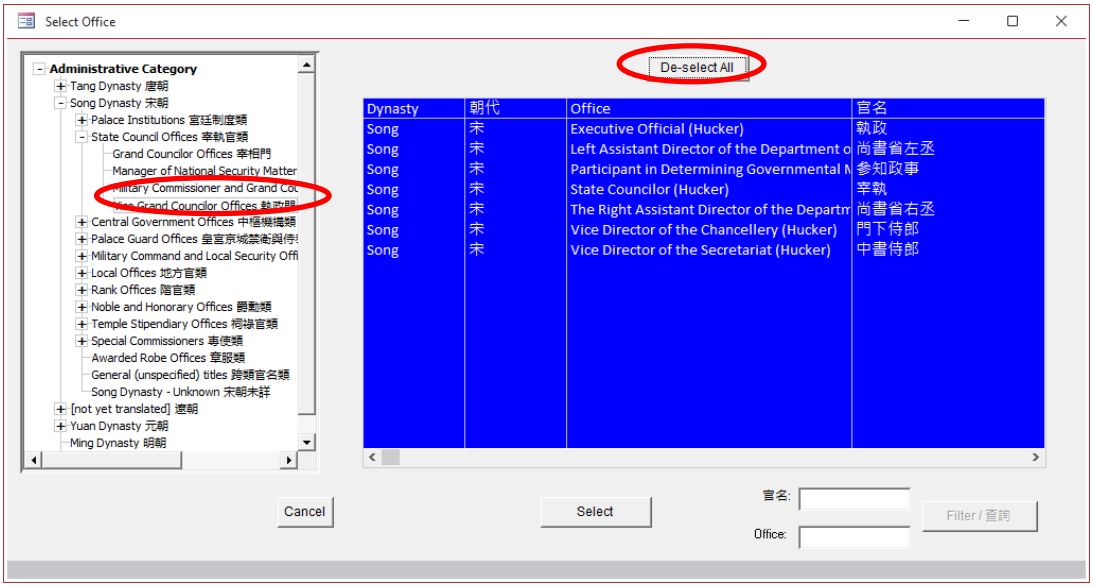


Selecting these office titles, one can locate all the women in CBDB who served as Grand Empress Dowager:

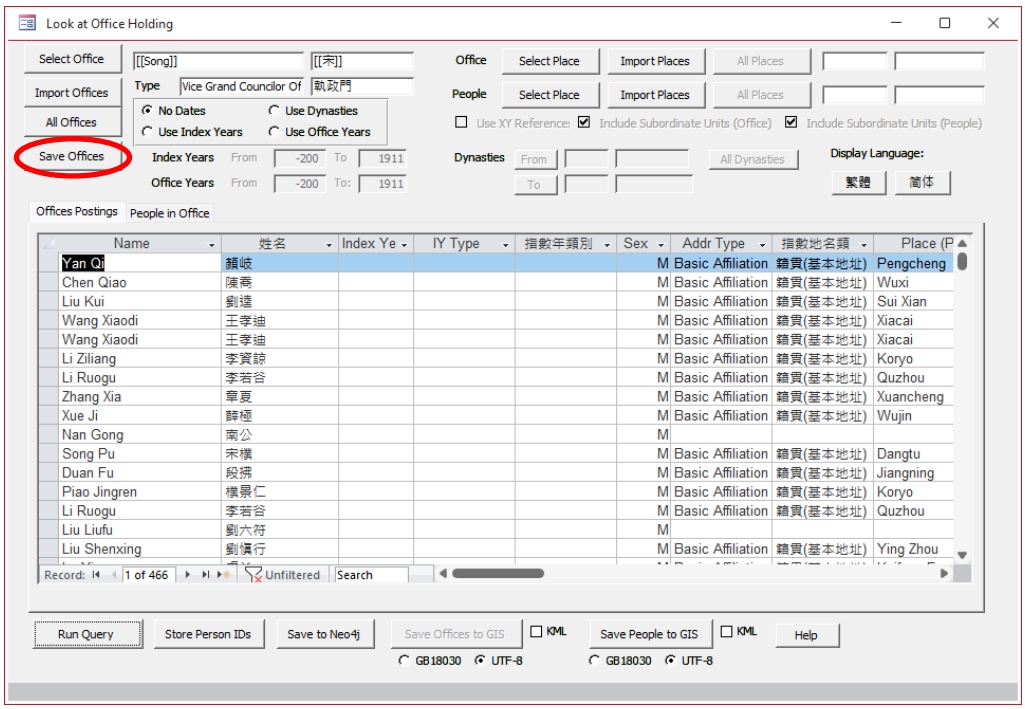


Saving, Editing and Importing Lists of Offices

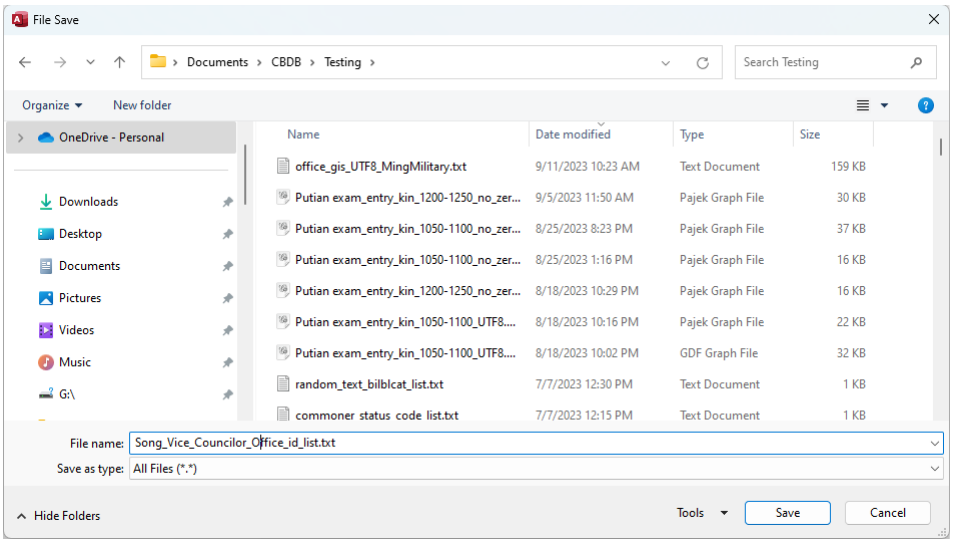
Because of the complexity of office structure—and especially structures across dynastic boundaries, CBDB provides a way to build reusable lists of offices that one might want to investigate. One begins by selecting an initial set of offices: for example, the Song dynasty offices in the Vice Grand Councilor's Office:



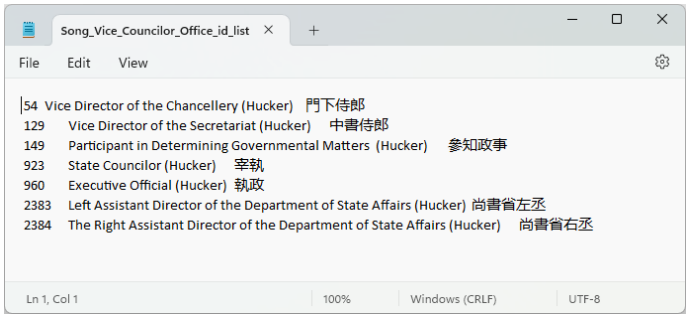
One can save this list by clicking on the **Save Offices** button:



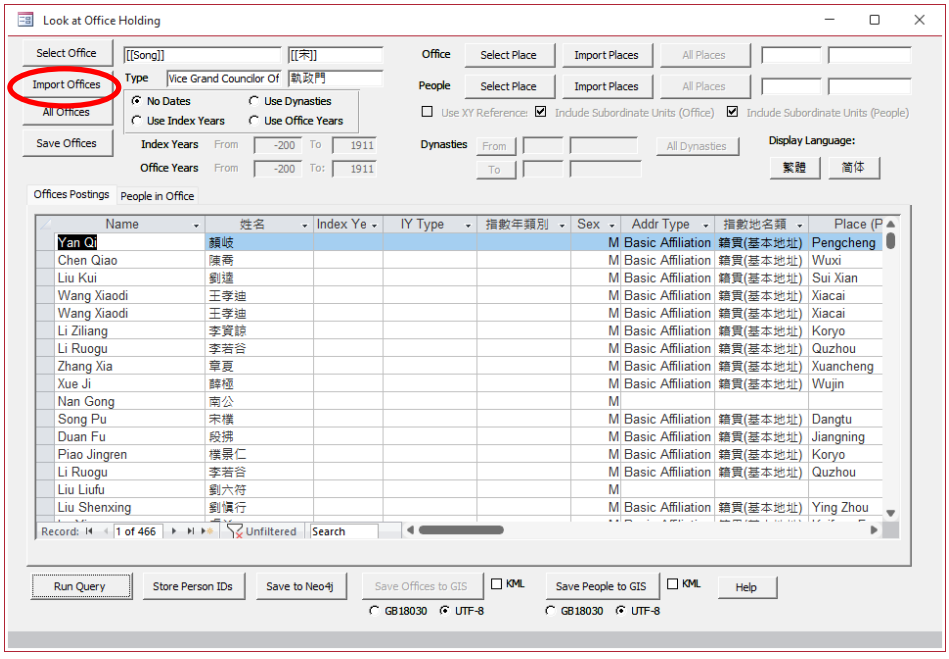
This command opens a "Save File" dialog box:



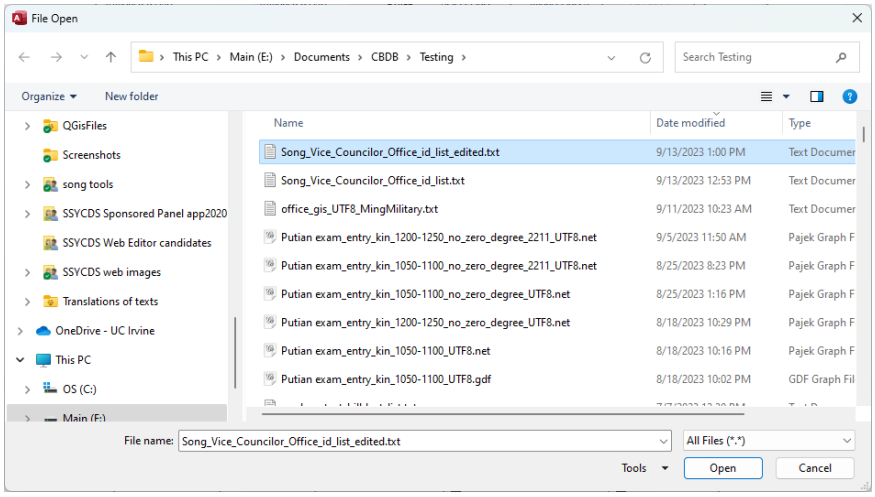
Once one has saved the file (as a text file), one can edit it in the usual manner:



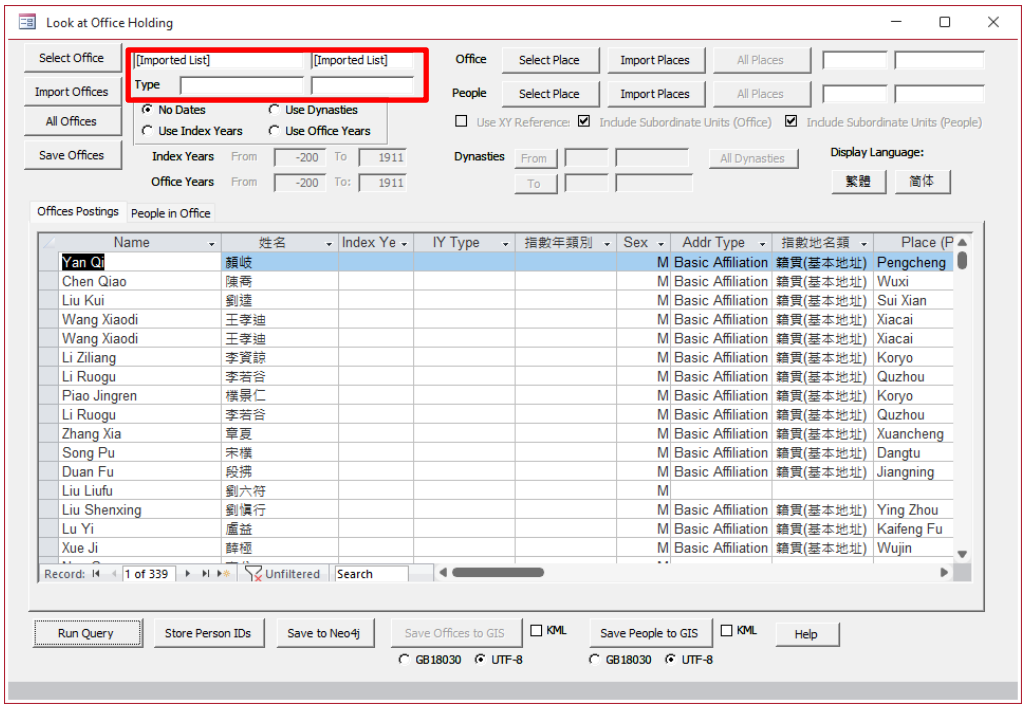
The file begins with the office ID, then a <Tab> character, then the rest of the information. Lists of office IDs from different files can be combined and imported back into the form.



Clicking the button will open an "Open file" dialog box:

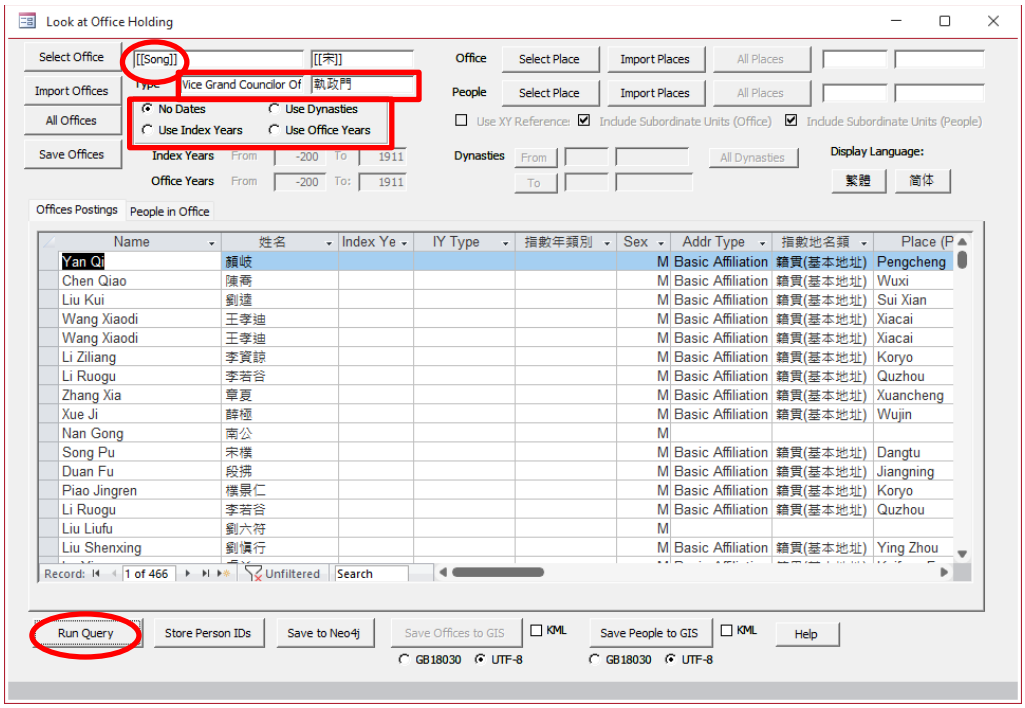


The form then displays "[[Imported List]]," and one can run a query using the list:

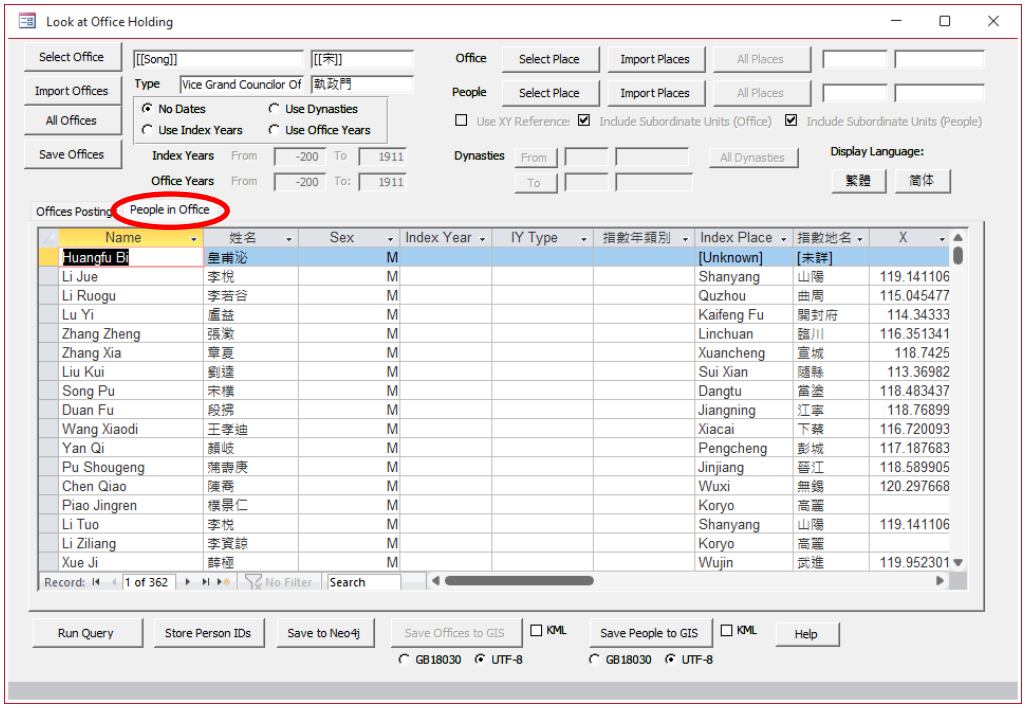


Querying Office

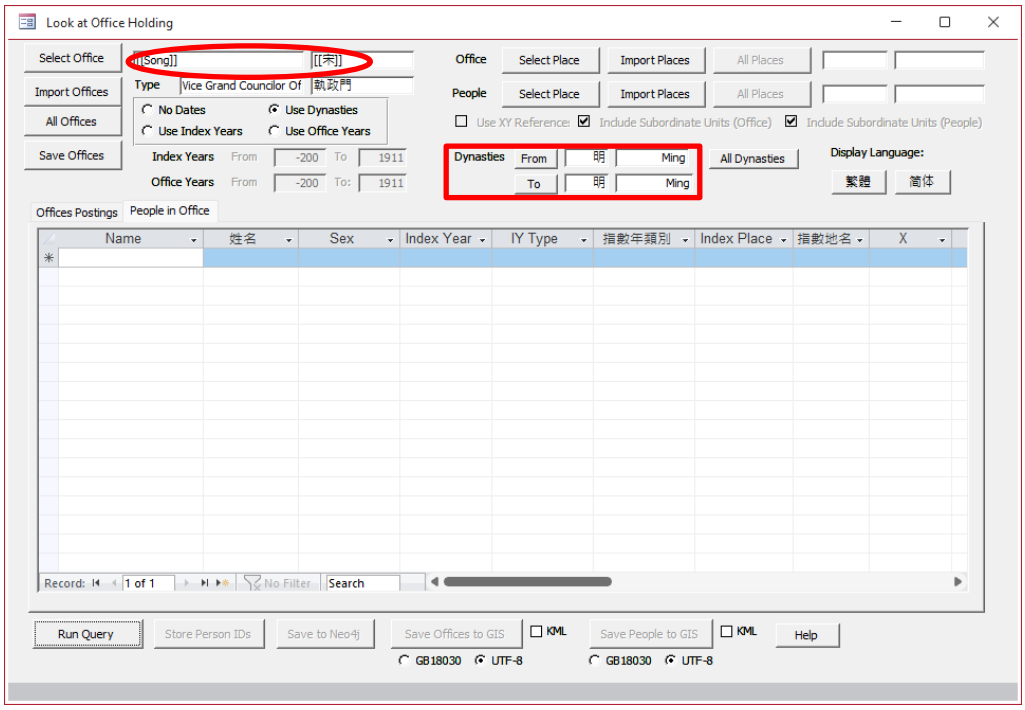
The form below shows a query selecting all postings to offices associated with the Vice Grand Councilor in CBDB and does not use either index years or dynasties. Because at present *office codes are tied to dynasty*, in fact all the results are from the Song (although one can use filtering by office name to create a cross-dynastic list of office codes):



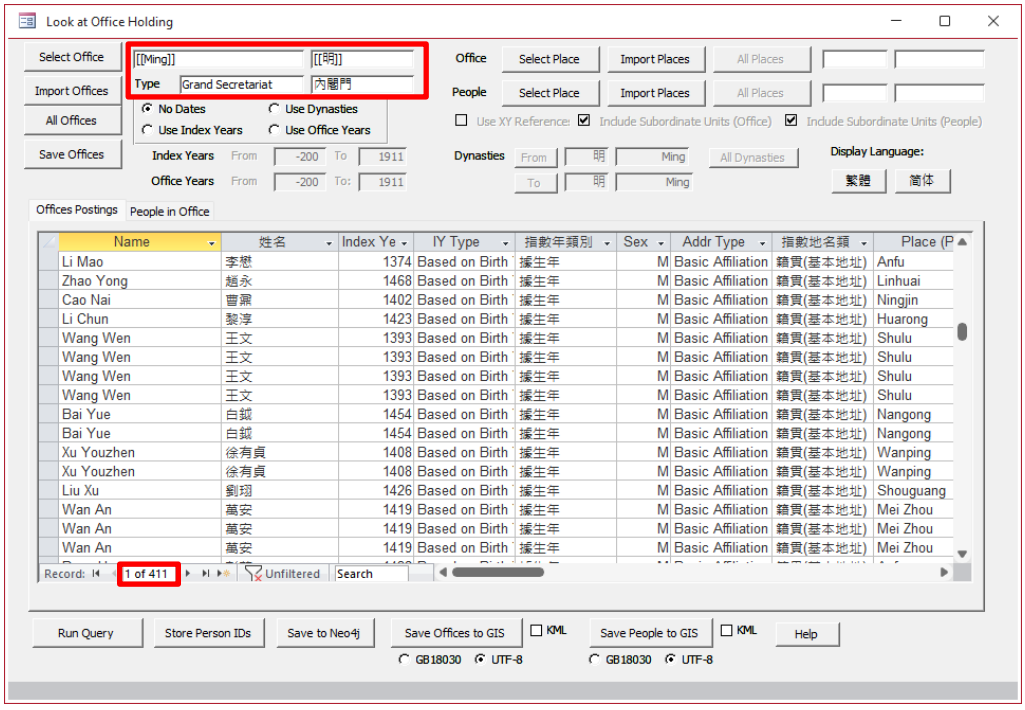
The query generates two tabbed pages of results. The first, **Office Postings**, displays information about all the postings to the offices being examined. The second, **People in Office**, lists the people who were appointed to the offices. This list of people is particularly useful if one wishes to then import it into the **LookAtNetworks** form to explore the social networks connecting the people who held a particular office. (One clicks on the square in the upper left corner to select all the records, copies them (Ctrl-c), and pastes them to a text file.



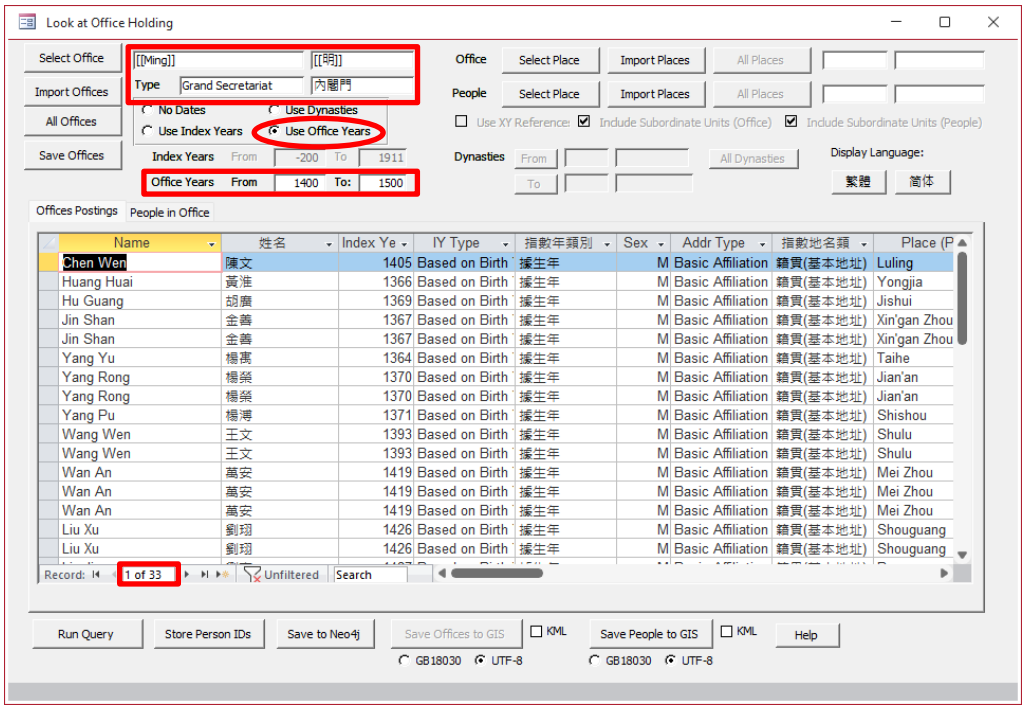
If one sets a time filter to the Ming dynasty for offices from the Song dynasty, as expected, one gets no results:



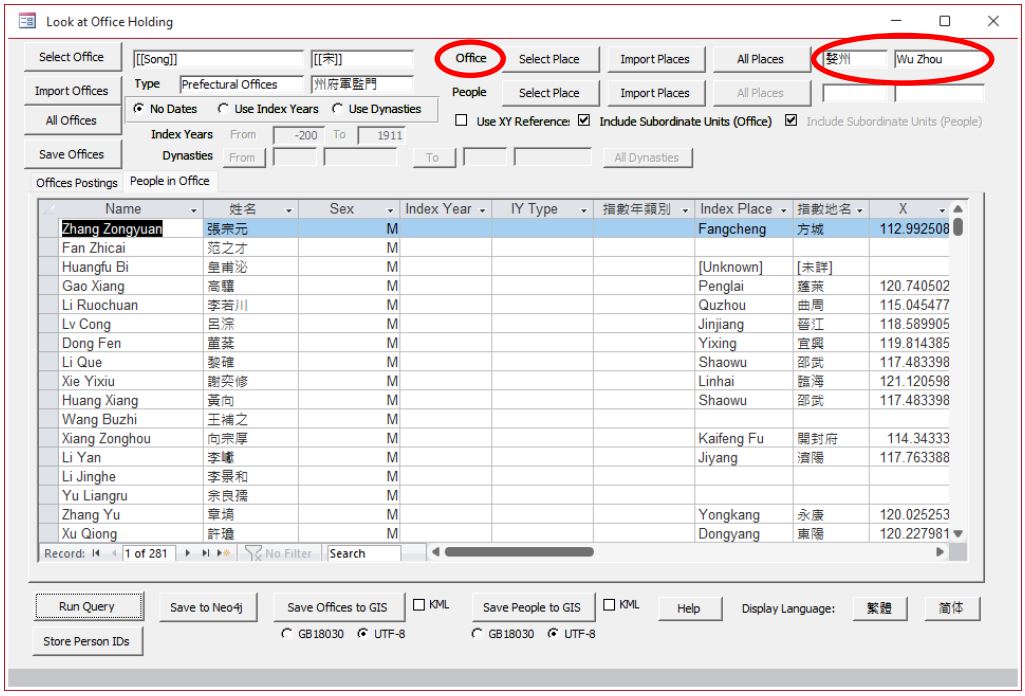
One can also set a filter for a range of years for which appointments to selected offices were made. It turns out that this option is most useful for the Ming, where data on years for office postings is more abundant. Without limiting the years, CBDB finds 411 postings for:



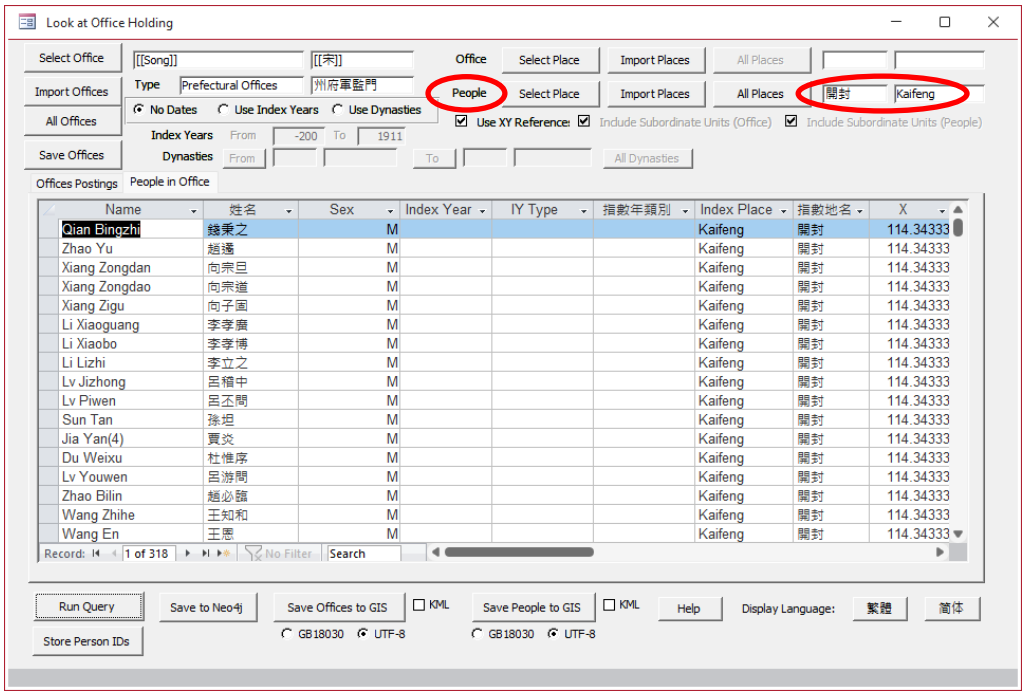
Adding the filter limits the results to the years 1400-1500 to 33 postings:



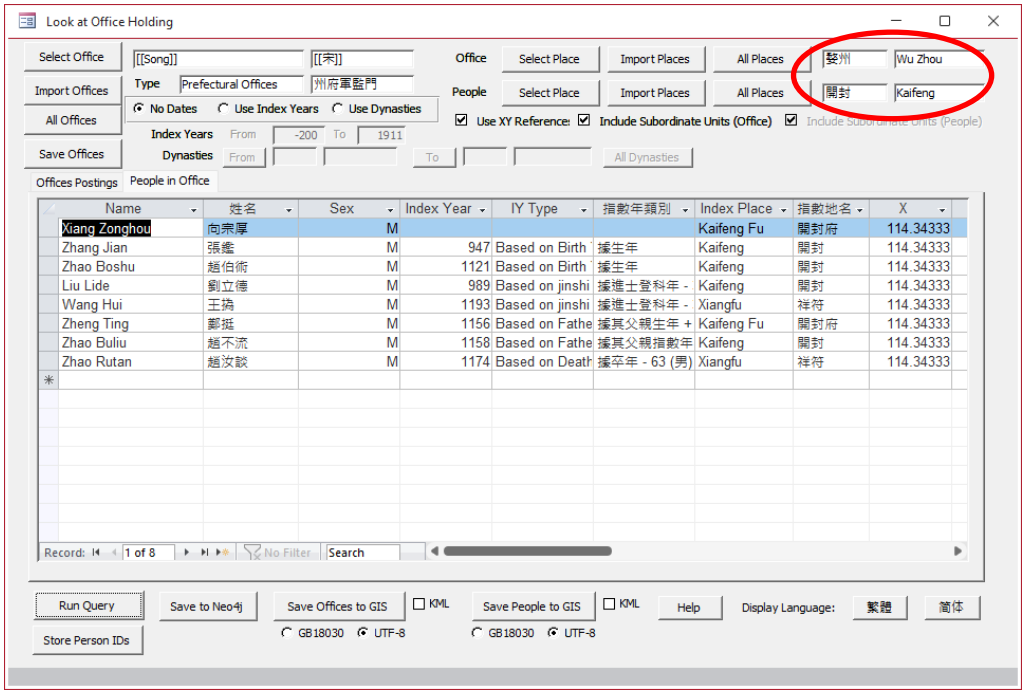
If one wishes to look at people who held office at a particular place or places, the form allows the user to select a place through the procedures discussed above. One can select a single place, use a filter for name, or import a list of address IDs. Then one runs the query in the usual way. Below is a query about the people who served in prefectural offices in Wuzhou 婺州 during the Song dynasty.



One can also explore where people from a particular place (or list of places) held particular types of office. Below is a query about where people from Kaifeng held prefectural office during the Song dynasty.

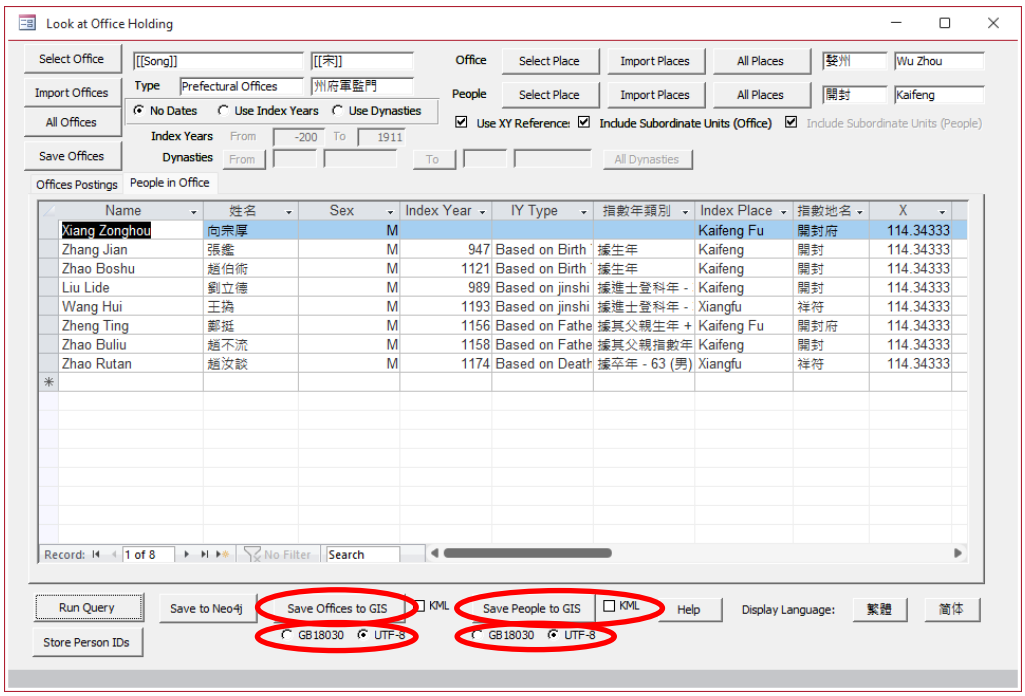


And one can combine the two restrictions and explore who from Kaifeng served in prefectural office in Wuzhou during the Song:



Exporting to GIS

Because one might want to look at the spatial distribution of either the postings or the people who held the posts, the **LookAtOffice** form provides ways to save both to files that can be read by GIS software. One can specify either UTF-8 or GB18030 encoding at the bottom left of the form:



Note that if the results do not have any place information with X-Y coordinates, then one cannot save information to a GIS file. For example, the office records for Vice Grand-Counselor Offices does not have any coordinates associated with them because the office location is simply "Song Dynasty."

As in all othr forms, one can save the results to a set of CSV (comma-separated values) files for use with Neo4j.

The table "Office Postings" has 30 fields:

Person Name (pinyin)
Person Name (Chinese)
Index Year
Sex (M or F)
Person Index Address Type (English)
Person Index Address Type (Chinese)
Person Index Address (pinyin)
Person Index Address (Chinese)
X coordinate of Person Index Address
Y coordinate of Person Index Address
Office (translation)
Office (Chinese)

First year of appointment
Last year of appointment
Dynasty (Pinyin) (useful in cases where the years are very uncertain)
Dynasty (Chinese)
Office Address (pinyin)
Office Address (Chinese)
X coordinate of Office Address
Y coordinate of Office Address
XY count (number of postings) for the Office Address
Notes
Person ID
Posting ID
Office Code
Appointment type (regular, provisional, etc.)
Information on assumption of office (accepted, declined, etc.)
Office Address ID
Person Address ID
Dynasty Code

The table "People in Office" contains the usual information about people:

Person ID
Person Name (pinyin)
Person Name (Chinese)
Index Year
Sex
Dynasty (Pinyin)
Dynasty (Chinese)
Index Address ID
Index Address (pinyin)
Index Address (Chinese)
X coordinate of Index Address
Y coordinate of Index Address
Index Address Type (English)
Index Address Type (Chinese)
XY count (number of people) for the Index Address

F. Using the Form "Query Kinship"

Queries involving kinship are more complex than queries examining categories of association or modes of attaining eligibility for office. Since the information on kinship for an individual usually contains just a few records, CBDB begins with those records and then looks at the kinship information available for all the kin listed for the initial person. CBDB repeats this search for the kin, the kin of the kin, the kin of the kin of the kin, and so on, until specified criteria are met. First is simply a limit to the number of search iterations to allow. Usually 5000 loops are adequate. Second are limits on the distance of the kinship relations being examined:

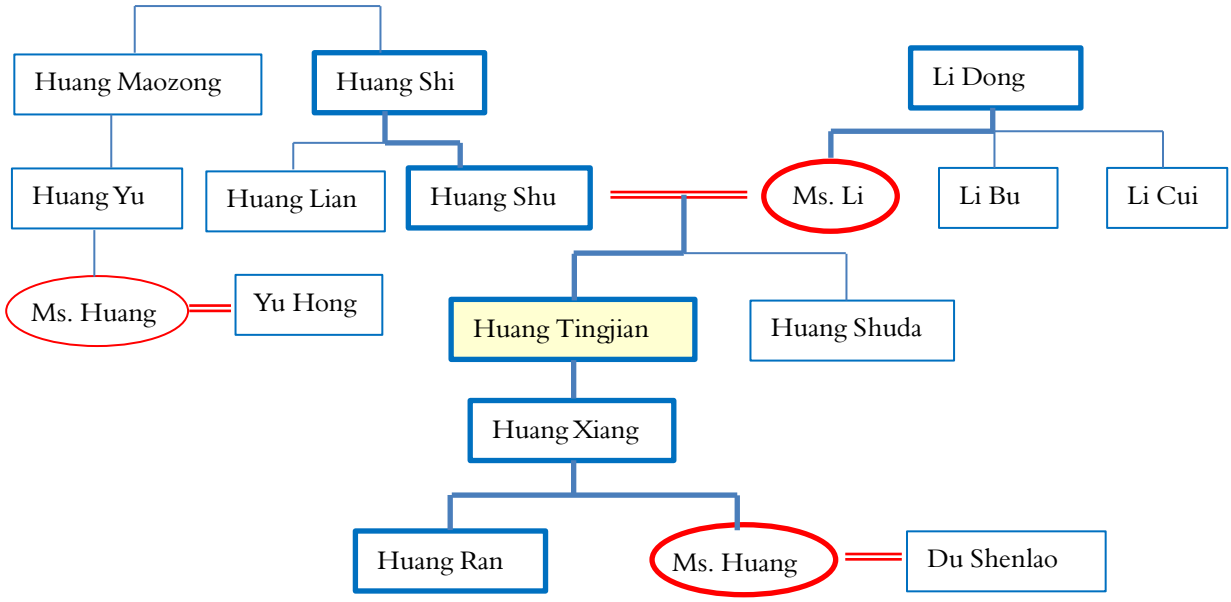
Max. Ancestor Gen. specifies how many generations of ancestors to include. One's father's generation is 1; the grandfather is 2, great-grandfather 3, and so on.

Max. Descend. Gen. specifies how many generations of descendants to include. One's children's generation is 1, grandchildren 2, great-grandchildren 3, and so on.

Max. Collateral Kin limits how many horizontal moves are allowed. For example, one's wife's sister has one unit of "marriage" distance and one unit of "collateral" distance. One's wife's sister's husband's brother has two units of "marriage" distance and two units of "collateral" distance.

Max. Marriage Dist. limits how many links defined by marriage are allowed in the search. One's wife's sister's husband has two units of "marriage" distance.

To visualize these distinctions, consider the partial kinship network:



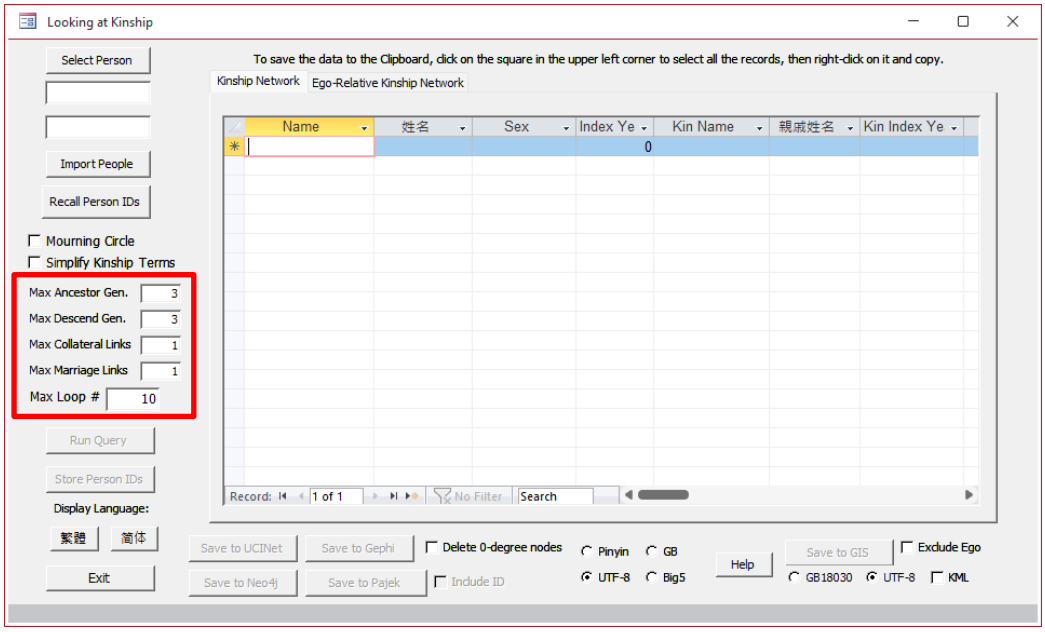
For Huang Tingjian, the squares and ovals with thick lines show direct lineal descent (fathers and mothers, sons and daughters). The double lines are marriage links. All other single lines

mark collateral relations. In the measurement system used in **LookAtKinship**:

Huang Yu 黃育	is FFBS	(Up = 2, Down = 1, Collateral = 1)
Yu Hong 余宏	is FFBSDH	(Up = 2, Down = 2, Collateral = 1, Marr. = 1)
Li Cui 李萃	is MB	(Up = 1, Collateral = 1, Marr. = 1)
Du Shenlao 杜莘老	is SDH	(Down = 2, Marr. = 1)

Because LookAtNetwork keeps looking through a very large table of kinship relations until the distance limits are reached, the kinship table produced by the search can grow very large. Therefore please note:

> **WARNING: searching for extended degrees of collateral and marriage distance may result in a very large dataset**

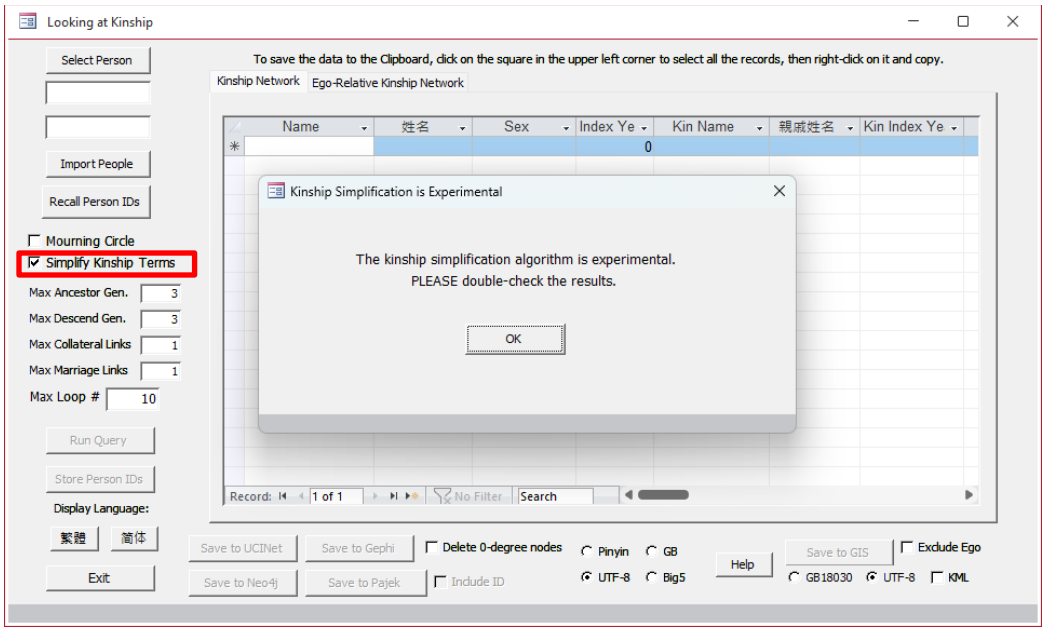


Please note that when one searches, CBDB automatically simplifies a small group of relations created by concatenating relationships through iterative searches (B = Brother; D = Daughter; S = Son; Z = Sister):

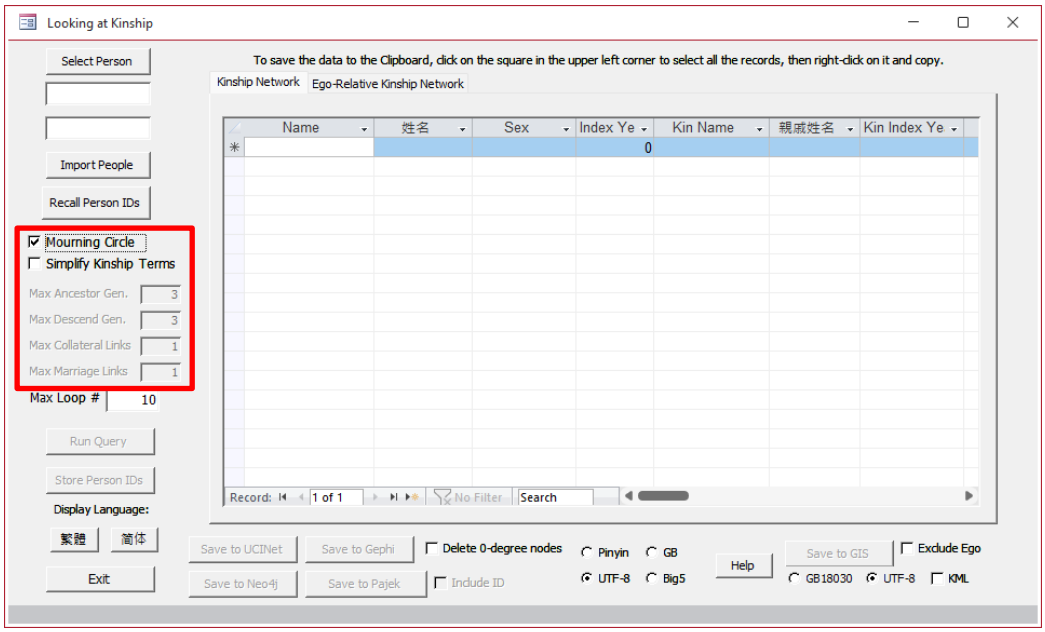
BB ⇨ B	BZ ⇨ Z	ZB ⇨ B	ZZ ⇨ Z
SB ⇨ S	SZ ⇨ D	DB ⇨ S	DZ ⇨ D

These simplifications reduce the **collateral** distance by 1.

CBDB also provides a more experimental option for simplifying concatenated kinship terms that the user should use with *caution*. The user can activate a CBDB kinship simplification algorithm by clicking on the **Simplify Kinship Terms** check box. (**Appendix D** lists the kinship terms simplified by the CBDB algorithm.) When one clicks on this option, CBDB warns that this approach should be double-checked:

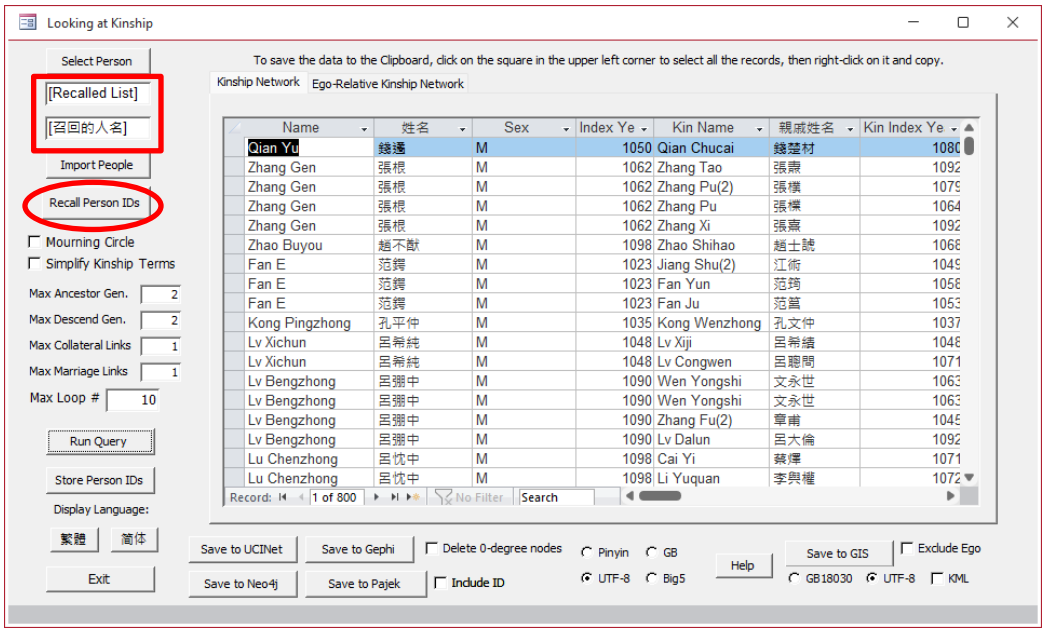


Another standard concern in Chinese kinship studies is to examine the so-called "mourning circle" defined by five degrees of kinship relation. LookAtKinship allows one to simply click on the "Mourning Circle" check-box to reconstruct what is known in the database about kin who are part of an individual's mourning circle. When one selects "Mourning circle," however, the four limit parameters are preset and therefore deactivated.



To examine kinship relations, one first selects the person or group of people whose kinship networks one seeks to explore. There are **three** different ways to select people. First is to **recall** an individual person (from the Browser) or group of people stored in the database as the result of an earlier query (see LookAtEntry for an example of storing the list). If the

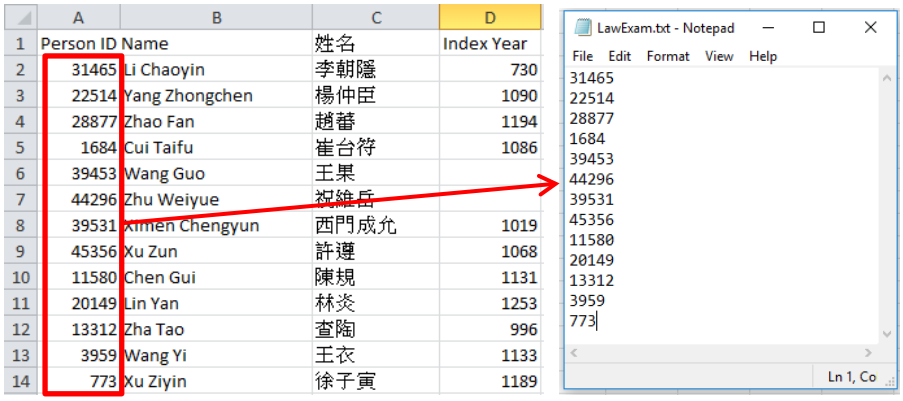
"Recall Person IDs" button is enabled, this means that there is a person ID or group of IDs created earlier that can be used now.



When one clicks on "Recall Person IDs," the form either loads the person (if there is just one ID) or the list of ID (and displays "[Recalled List]") in the box for the selected person's name. One then sets the desired parameters and runs the query.

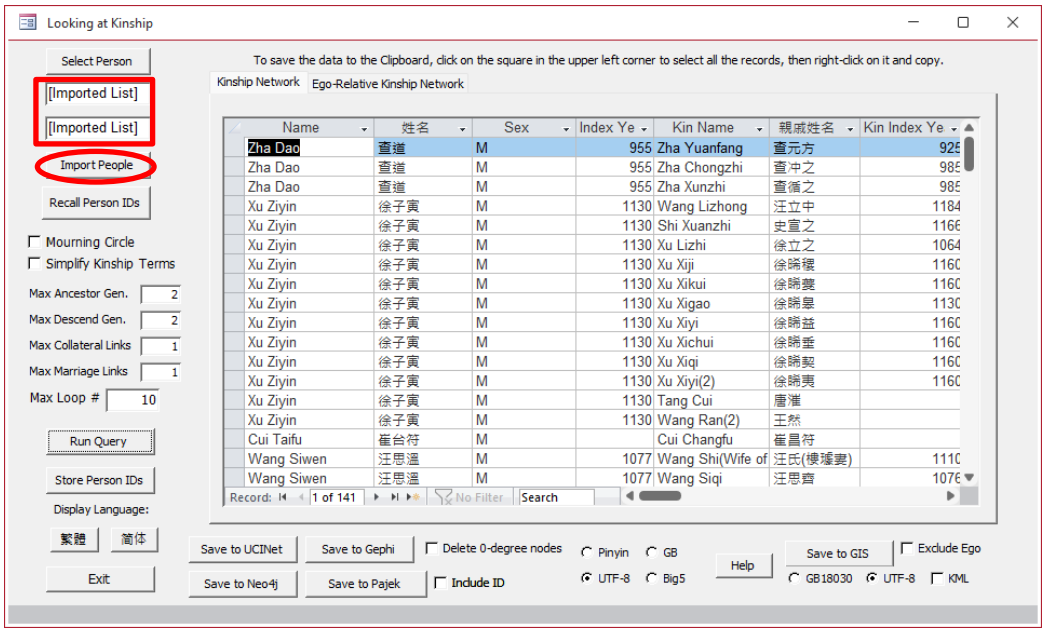
The second approach is through **importing** a list of people sharing common characteristics identified by other queries. For example, one could start with people in the Song dynasty who became eligible for office through a legal examination. One copies the results of the **LookAtEntry** query to a **Word** or **Excel** file, edits the results, and copies the person IDs to a text file.

Note that this is a change from earlier versions of the program. The text file should contain nothing more than a list of person IDs and needs to be in ANSI text formatting.



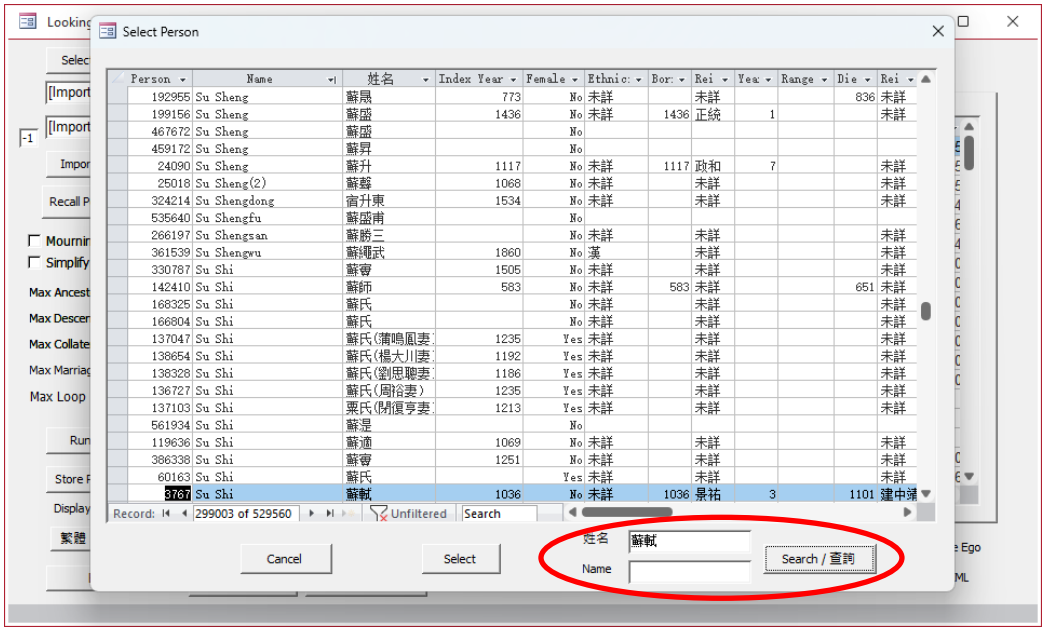
	A	B	C	D
1	Person ID	Name	姓名	Index Year
2	31465	Li Chaoyin	李朝隱	730
3	22514	Yang Zhongchen	楊仲臣	1090
4	28877	Zhao Fan	趙蕃	1194
5	1684	Cui Taifu	崔台符	1086
6	39453	Wang Guo	王果	
7	44296	Zhu Weiyue	祝維岳	
8	39531	Ximen Chengyun	西門成允	1019
9	45356	Xu Zun	許遵	1068
10	11580	Chen Gui	陳規	1131
11	20149	Lin Yan	林炎	1253
12	13312	Zha Tao	查陶	996
13	3959	Wang Yi	王衣	1133
14	773	Xu Ziyin	徐子寅	1189

After one clicks on the **Import People** command button, selects the file, and **LookAtKinship** successfully reads the file, the form will look like:



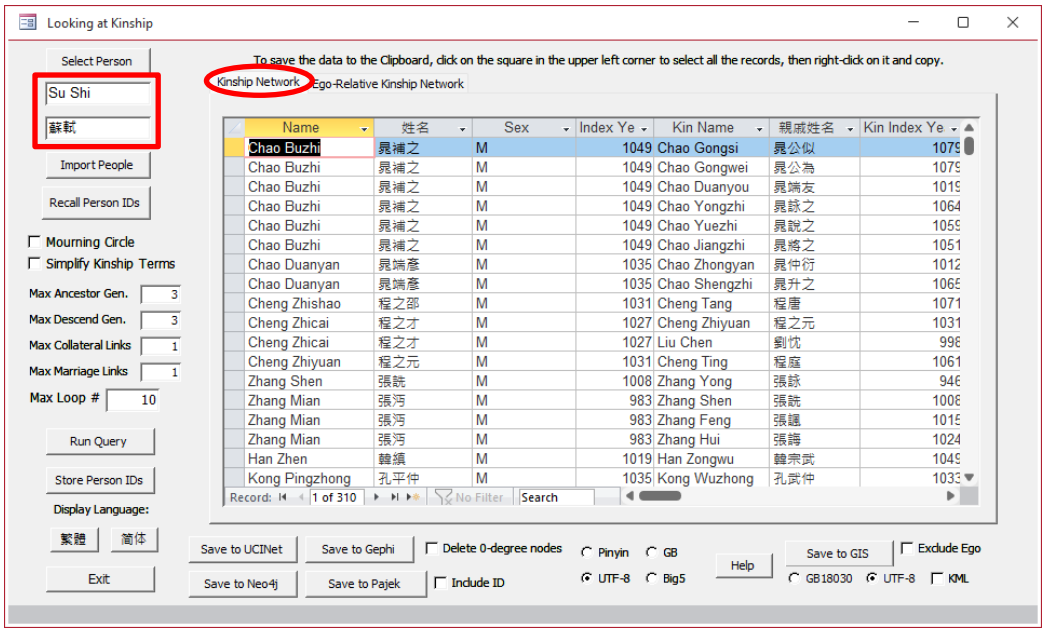
One then sets the desired parameters and runs the query.

The third approach is the simplest and most direct: one clicks on the "Select Person" command button on the top left corner, which will open a form with a list of all the people in the database. One can search for a person using either Chinese characters or *pinyin*. (Sometimes, for a second search, the form finds an unrelated person: just click **Search** again to reset the results.)



Once one has selected the person, one sets the search limits (or chooses the Mourning Circle) and clicks the **Run Query** command button to start the search.

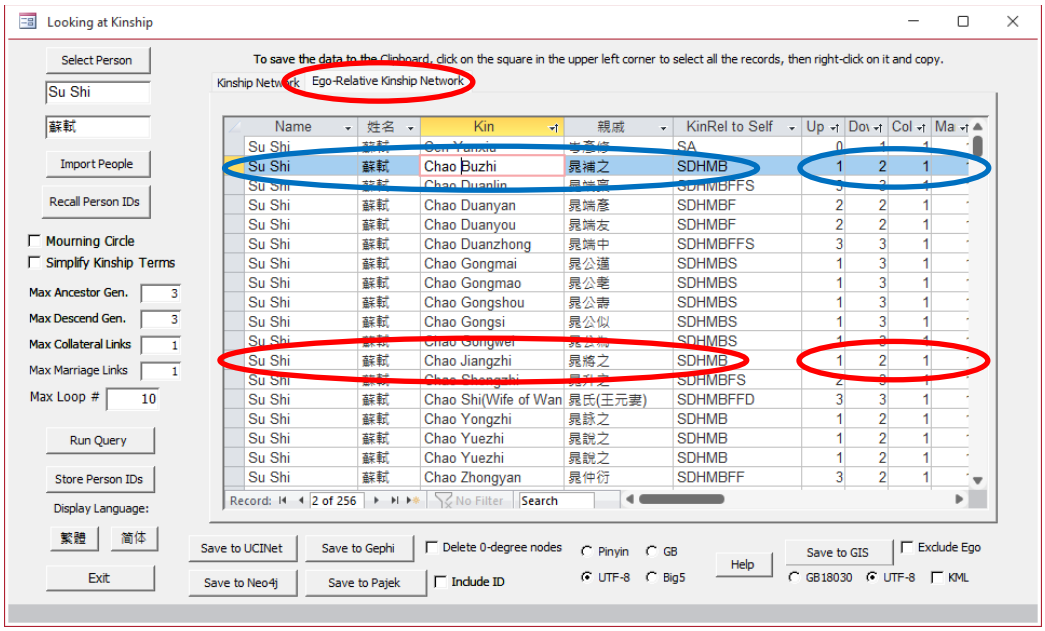
When the search finishes, there are two tables one can examine. The first, **Kinship Network**, lists all the kinship relations discovered through the search:



This table has 27 columns:

Name (pinyin)
Name (Chinese)
Kin Name (pinyin)
Kinship Relation
Index Address of Person (pinyin)
Index Address of Person (Chinese)
X-Coordinate
Y-Coordinate
Index Address of Kin (pinyin)
Index Address of Kin (Chinese)
X-Coordinate of Kin Index Address
Y-Coordinate of Kin Index Address
Notes
Index Address Type
Kin Name (Chinese)
Index Year of Kin
Sex of Kin
Index Address Type (Chinese)
Address Type of Kin Index Address
Address Type of Kin Index Address (Chinese)
Distance (great-circle distance between the addresses)
Person ID
Kin ID
Index Year Type (English)
Index Year Type (Chinese)
Kin Index Year Type (English)
Kin Index Year Type (Chinese)

The second table, **Ego-Relative Kinship**, describes the kinship relation between each person in the first table and the person selected at the very beginning:

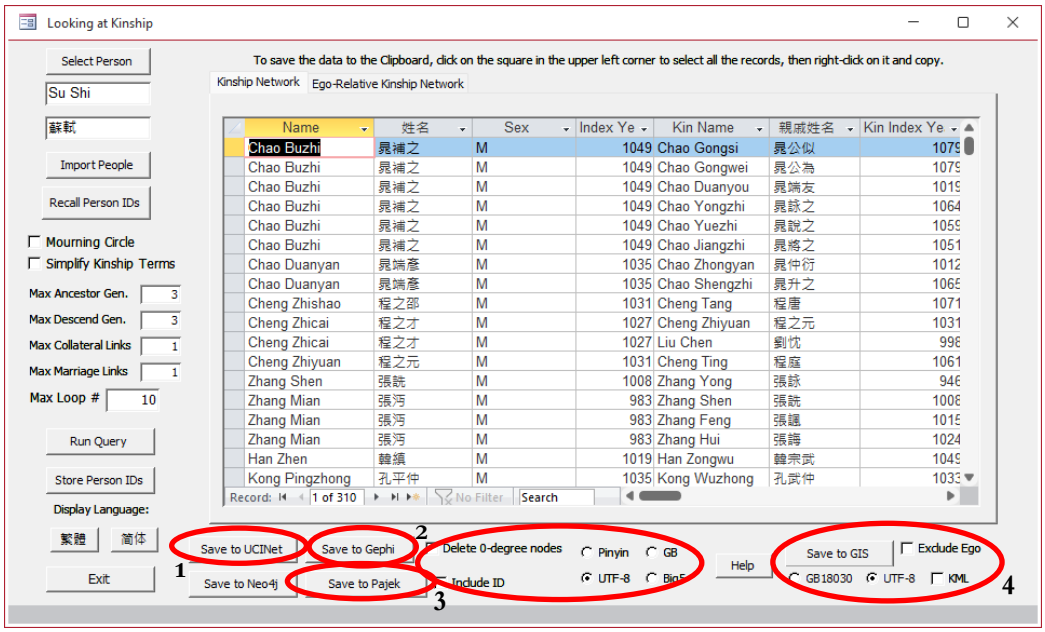


For example, Chao Buzhi 晁補之 is Su Shi's third son's second daughter's husband's mother's brother (S3D2HMB) with a metric of {1,2,1,1}. The path one traverses to reach Chao Buzhi's younger brother Chao Jiangzhi 晁將之 is first to locate Chao Buzhi and then find all of Chao Buzhi's brothers. Their metrics would then be that of Chao Buzhi, {1,2,1,1} + one more collateral step, for the result {1,2,2,1}, which would exceed the search parameter for collateral distance, set to just 1. However, the search algorithm automatically reduces **BB** (in S3D2HM**B+B**) to **B**, since they, as Chao Buzhi's brothers, are also brothers to the husband's mother. They then fall within the 1 collateral link distance and are included in the search results. (The "Ego-Relative Kinship" table has an additional column that gives a raw path that shows how CBDB simplified the kinship relations, but, as explained above, CBDB simplifies only the simplest relations (e.g., BZ ⇨ Z). More complex simplifications require correspondingly complex algorithm that CBDB does not implement.)

As is true for all the other tables in all the other forms, if one clicks on the upper left corner of either table in this form, one can select all the records in the table, which then can be cut and pasted into other programs. Also, right-clicking on any of the column headings allows one to sort on that column.

Finally, one can export the kinship data to four different types of files. The first three are different formats of Social Network Analysis (SNA) files: **UCINet** (1), **Gephi** (2), and **Pajek** (3) with various character code options and the option to include ID in the labels. For Gephi and UCINet, the program can also remove *zero-degree nodes*, those nodes without connections to any other nodes. (This sometimes occurs when one imports a list of people to search for, and some of those people have no kinship information in CBDB.) The fourth type of file is for GIS visualization: the program can save the file as a file readable GIS software (4) or in **KML** format with two different code options. The output includes the **xy_count** field, which is *the count of the number of people associated with a particular set of coordinates*. This field is very useful as a parameter for displaying results in GIS software. Note that the form allows one to exclude the *ego-records* in the GIS output. When one has searched for the kinship network of a

single, selected person, checking this box just removes the selected person from the output with little impact on the results. However, if one looks for the kinship networks of a list of people, including the people on the list can distort the data, and especially the xy_count, and it may prove useful to filter those people out of the GIS data and focus only on their kin.



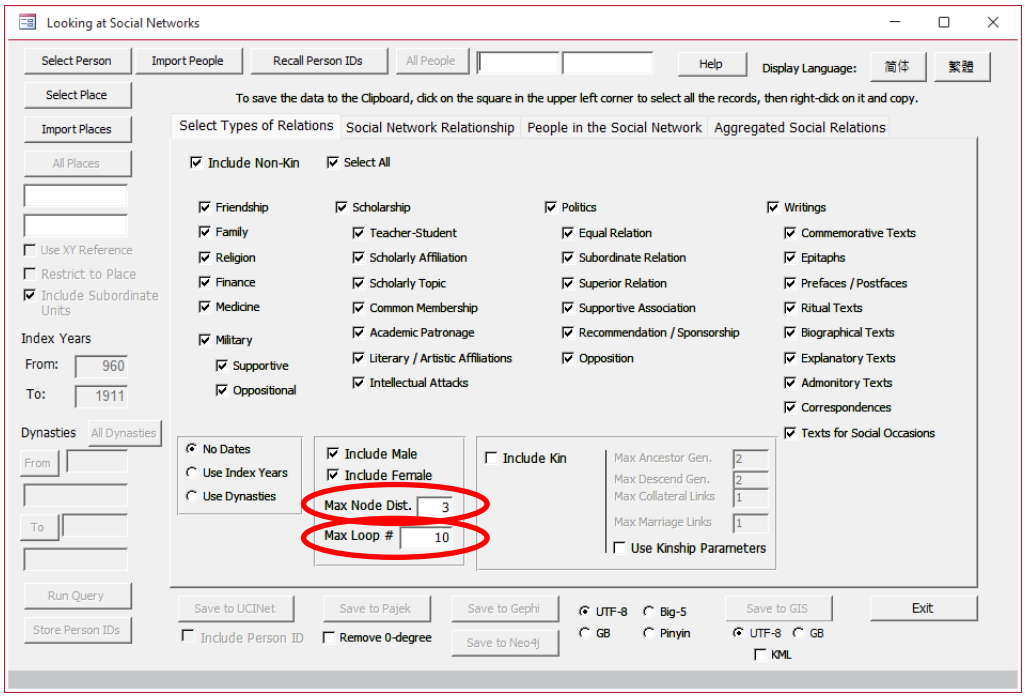
The default display for both nodes and edges in the Pajek output files uses color-coding to indicate degree of distance from the target person:

Black = the target node;
Blue = nodes at a summed kinship distance of 1
Green = nodes at a summed kinship distance of 2
Orange = nodes at a summed kinship distance of 3
Yellow = nodes at a summed kinship distance of 4
Red = nodes at a summed kinship distance of 5 or more

G. *Using the Form "Query Social Networks"*

LookAtNetworks is the most powerful querying interface developed for the Access version of CBDB. It allows the user to explore social networks defined both by kinship ties and by other forms of social relations. It allows the user to select a person or to import a list of people produced by other queries. It similarly allows the user to start with a place or to import a list of places. **LookAtNetworks** allows the user to select the particular forms of social association to investigate, and it allows one to set the range of years to consider. As in **LookAtKinship**, the queries in **LookAtNetwork** are iterative: the query produces an initial group of people and then looks at the relevant connections between these people and others in the database. Each cycle adds more people, whose associations then produce yet more people. **LookAtKinship** has five metrics to limit the search, but **LookAtNetworks** has just two: a *maximum loop count* (how many times the query iterates through the list of people), and a *maximum node distance*. This distance is the number of links between a person in the network and members of the group of people identified by the first step in the search process. If the user selects a particular person, then all distances are measured from that person. If one starts with a list of people, then all the people on that list serve as starting points. If one starts with a place or list of places, then the people initially identified as associated with that place or those places serve as the starting points.

WARNING: Higher node distances may result in a very large dataset



NOTE: A query set at a node distance of 1 will result in a) all the people associated with the selected person(s) and b) all the associations between the people in the network. This particular kind of network is called an ego network: it is important because it sometimes reveals that even within the network of one person there were rival networks. One can sort

these relationships in the query results table, and one can delete any records one does not wish to export for further analysis.

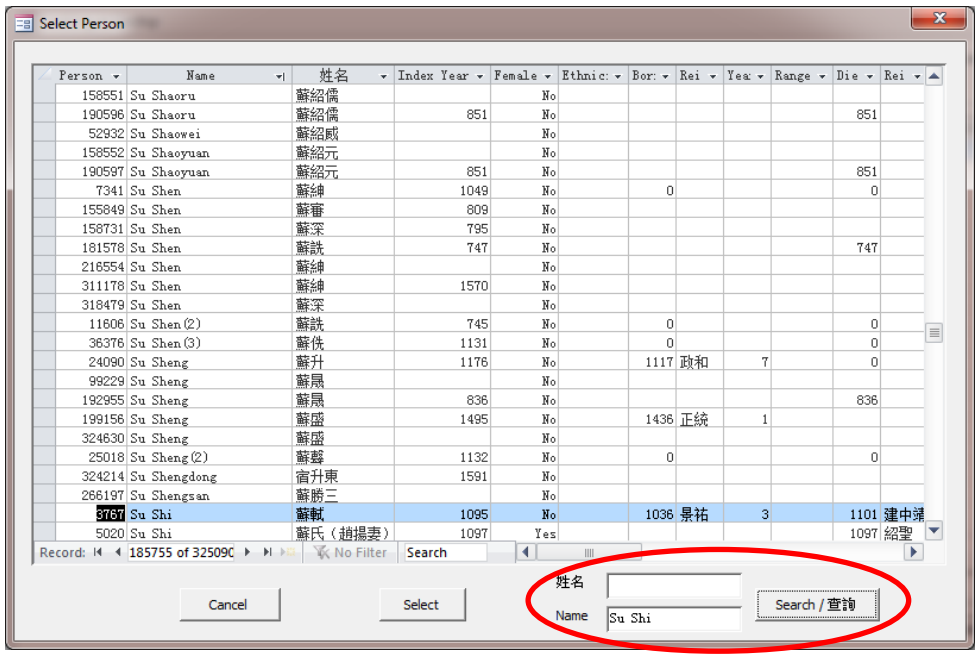
Basic Query Functions

Running a query begins with selecting the elements to investigate.

1. Begin with People

A. Select a Person

If one wants to look at the social networks which link a particular person to others, one can click on the **Select Person** command button to open a list of people:



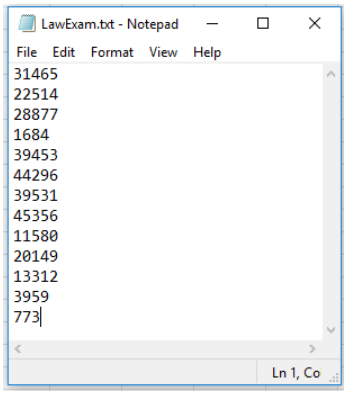
Since the list contains all the people in the database, one can search for a particular person using the search box in the lower right corner.

B. Import a List of People

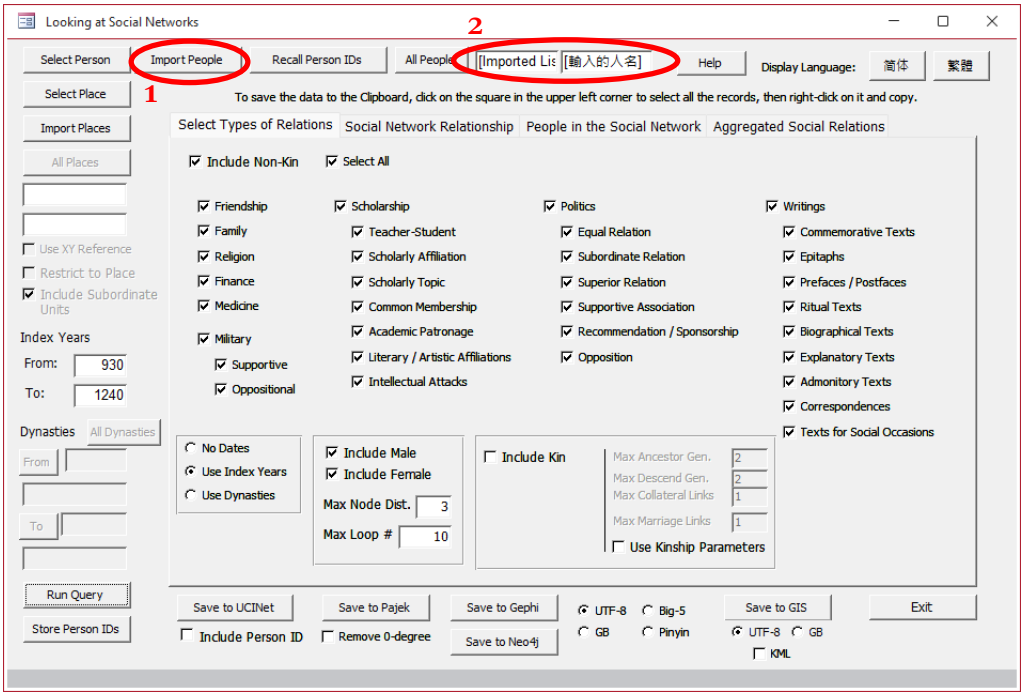
A second, very useful way to consider social networks is to import a group of people sharing common characteristics identified by other queries. For example, one could start with people in the Song dynasty who became eligible for office through a legal examination. One copies the results of the **LookAtEntry** query to a **Word** or **Excel** file, edits the results, and copies the person IDs to a text file.

Note that this is a change from earlier versions of the program. The text file should contain nothing more than a list of person IDs and needs to be in ANSI text formatting.

	A	B	C	D
1	Person ID	Name	姓名	Index Year
2	31465	Li Chaoyin	李朝隱	730
3	22514	Yang Zhongchen	楊仲臣	1090
4	28877	Zhao Fan	趙蕃	1194
5	1684	Cui Taifu	崔台符	1086
6	39453	Wang Guo	王果	
7	44296	Zhu Weiyue	祝維岳	
8	39531	Ximen Chengyun	西門成允	1019
9	45356	Xu Zun	許遵	1068
10	11580	Chen Gui	陳規	1131
11	20149	Lin Yan	林炎	1253
12	13312	Zha Tao	查陶	996
13	3959	Wang Yi	王衣	1133
14	773	Xu Ziyin	徐子寅	1189



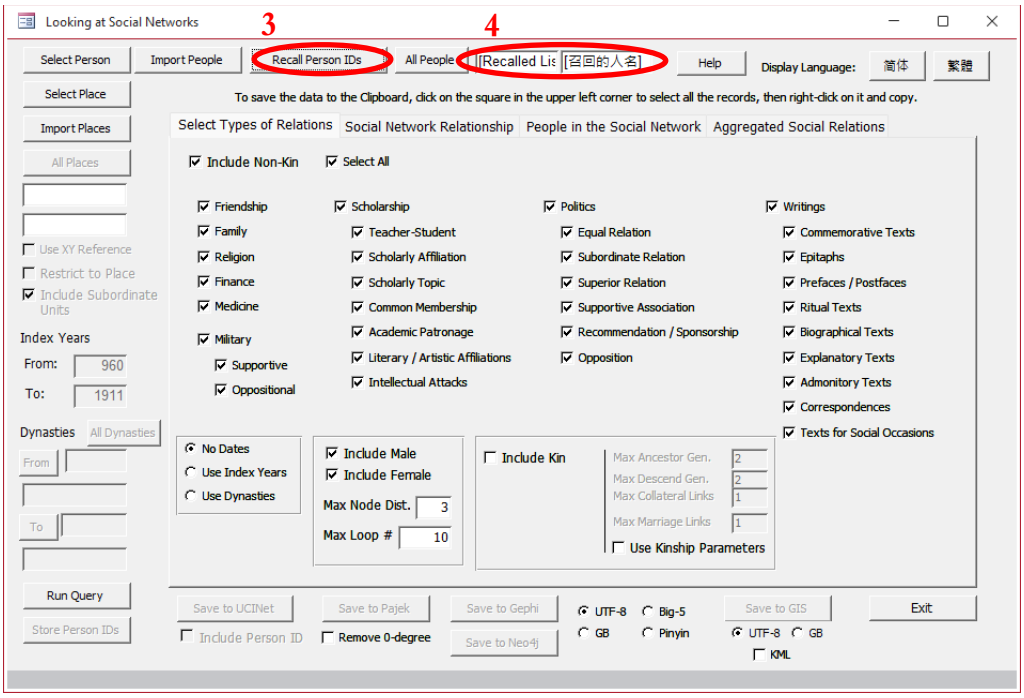
After one clicks on the **Import People (1)** command button, selects the file, and **LookAtNetworks** successfully reads the file, the form will look like:



The two boxes that give the person's name **(2)** will state "[Imported List] and [輸入的人名]."

C. ***Recall a Person or a Group of People from a Previous Stored Search Result***

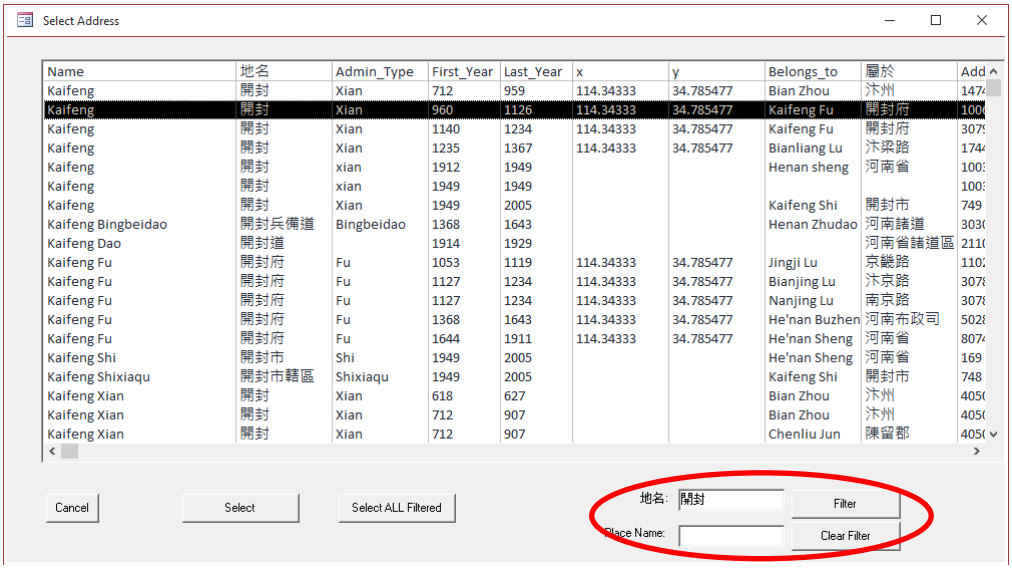
The third way to select people for analysis is to recall either a single ID that was stored from the **Browser** or a list of IDs saved from a previous query. One simply clicks on the **Recall Person IDs (3: next page)** command button. If there is just *one saved ID*, the form displays the person's name. If one recalls *a list of IDs*, the form displays "[Recalled List]" and "[召回的人名]" instead of a person's name **(4: next page)**:



2. Begin with Place

A. Select a Place

When one clicks on the **Select Place** command button, one opens a form to allow one to select a particular place. As described in the section on **LookAtEntry**, , the form provides a **Filter** function to select a group of addresses all beginning with a specified word or phrase.



B. Import a List of Places

Sometimes it is more useful to work with a set of Address IDs to precisely define the area for which one wants to study the social networks. Importing a list of Address ID works the same way as importing Person IDs.

Unless one clicks the **Restrict to Place** check box, the selection of a place or list of places *only influences the first step* of locating an initial group of people around whom to build a social network. After the first round of locating people with a connection to the specified place(s), CBDB searches for the sorts of associations selected as the next part of the query process.

If one uses both people and place as the starting point for a query, CBDB looks for people from the specified place who had connections to the selected person or group of people that matched the categories of association specified for the search.

C. Use XY Reference

As in the other forms, CBDB allows the one to use the longitude and latitude of the place(s) one has selected to identify other relevant administrative units for the specified time period. One clicks on the **Use XY Reference** check box to activate this feature.

3. Determine the Time Period

This is straight-forward: simply fill in the beginning and ending years for the *index years* of people to be considered for the search.

4. Select the Node Distance

One needs to be careful: the number of people found by the search procedure can grow exponentially with the increase in node distance. It is a good practice to start conservatively with a small node distance. In the example search discussed below, using the nine people who became eligible for service through law examinations, a node distance of three produces over 5000 relationships.

5. Set the Maximum Number of Iterations

The search procedure is slow with CBDB's large dataset, and one might want to experiment with a relatively small "Max Loop #."

6. Select Kin, Non-Kin, Male and Female

The LookAtKinship form does not allow one to look at kinship relations for a group of people imported through a list, so LookAtNetworks provides an alternative approach to examining kinship. One selects "Kin" and de-selects "Non-Kin." There also may be times when one wants to eliminate associations (kinship or social) based on females, or one may want to examine networks strictly among women. LookAtNetworks allows the user to select these options.

7. Select Types of Non-Kinship Relations

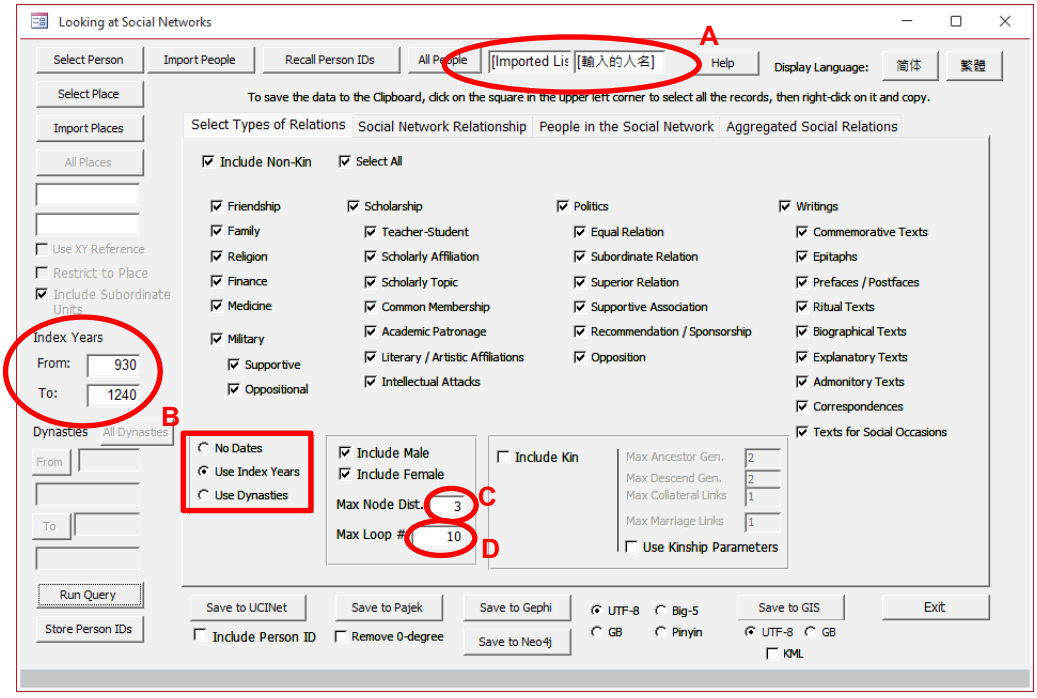
Because there are many, many categories of non-kinship relationship, most of which are of little importance in a particular query, one can limit the search to selected large groups of

associations. These are:

Friendship
Family
Religion
Finance
Medicine
Military
Scholarship
Politics
Writings

The last four types of non-kin associations have further selectable subdivisions. "Military" has two, "Scholarship" seven, "Politics" six, and "Writings" nine. One can mix the types of associations as one wishes. Once selected, these limits to the range of associations remain active through the entire search process.

Once all of these decisions have been made, one runs the query. The example examined below uses the list of people (A) who entered service through the law examination. The first version selects the years 930 through 1240 (B) with a maximum node distance of 3 (C) and a maximum loop count of 10 (D) but does not constrain either the kinship or the non-kinship associations and allows all possible types of association.



The result is a network with 6,441 people participating in 24,782 relations.

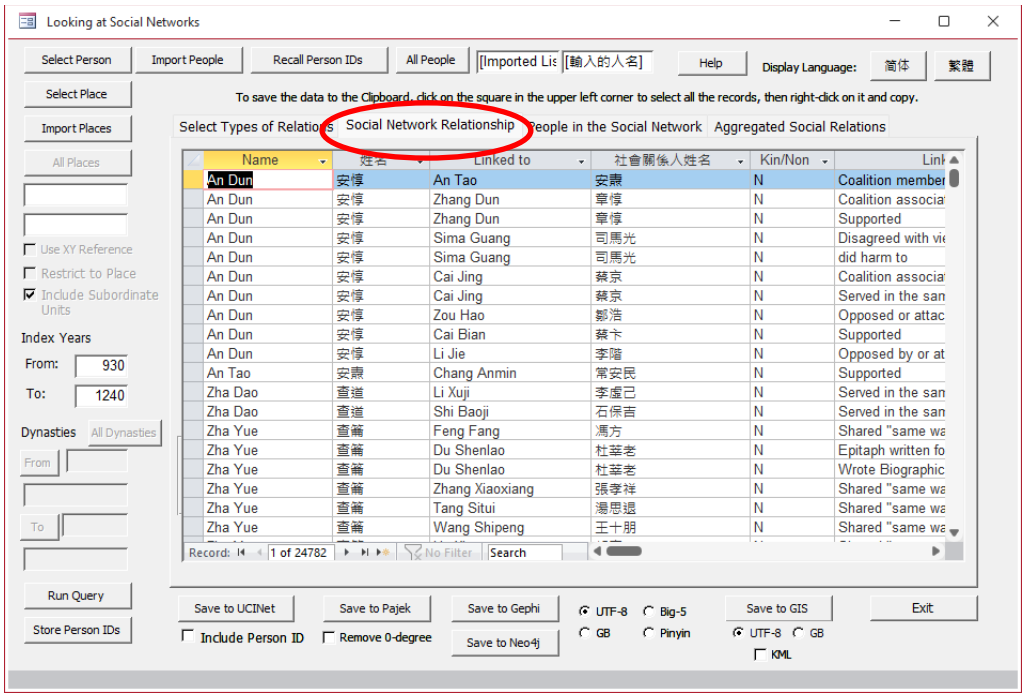


Table of Associations in the Social Network

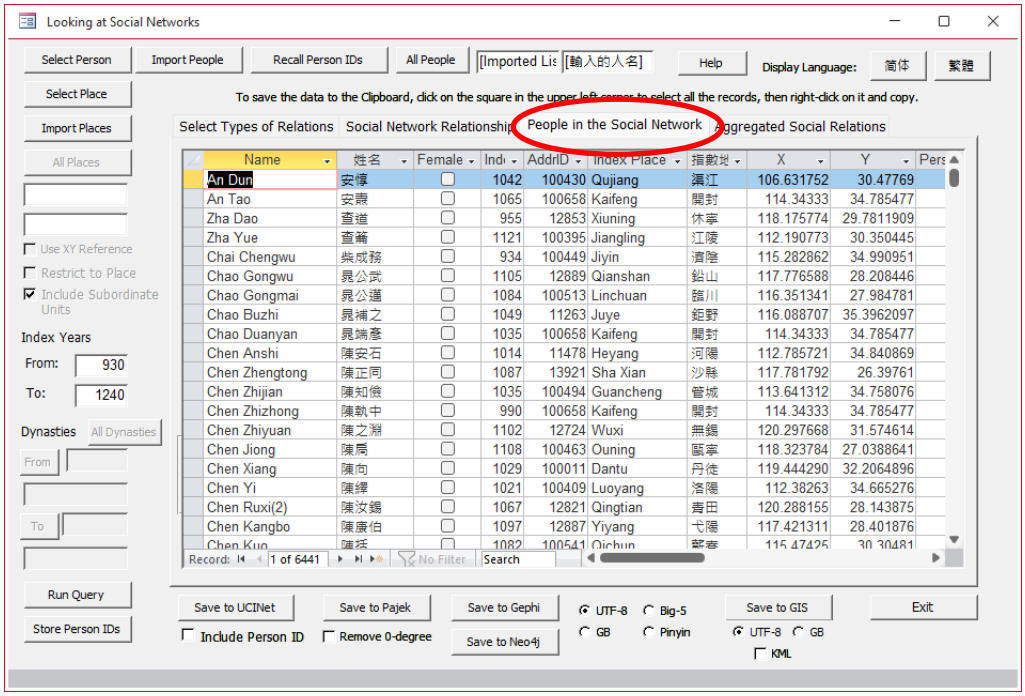
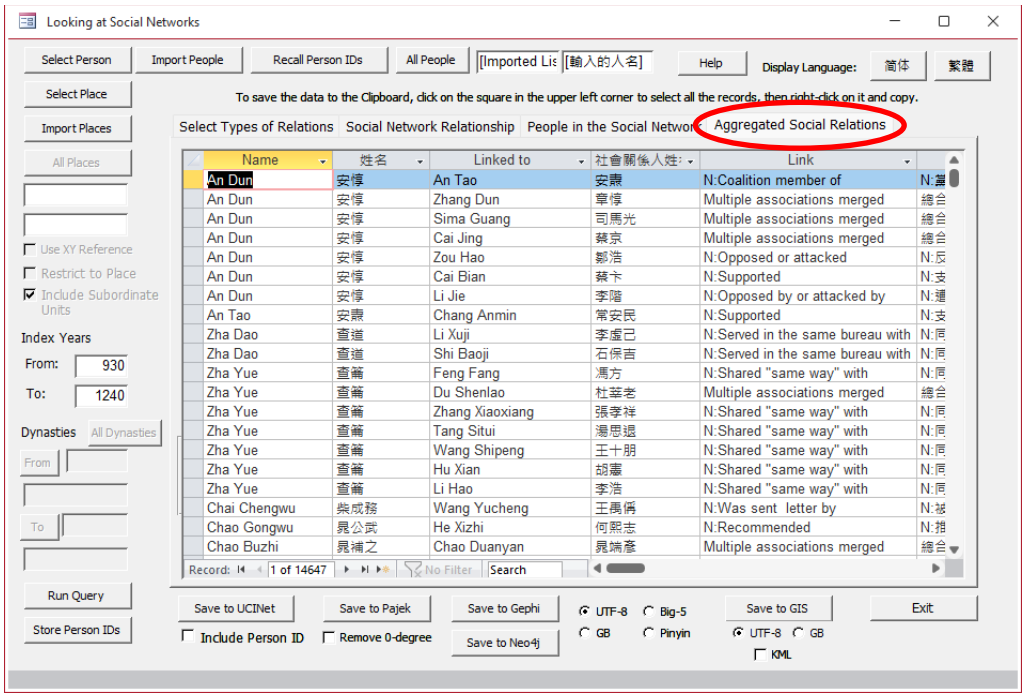
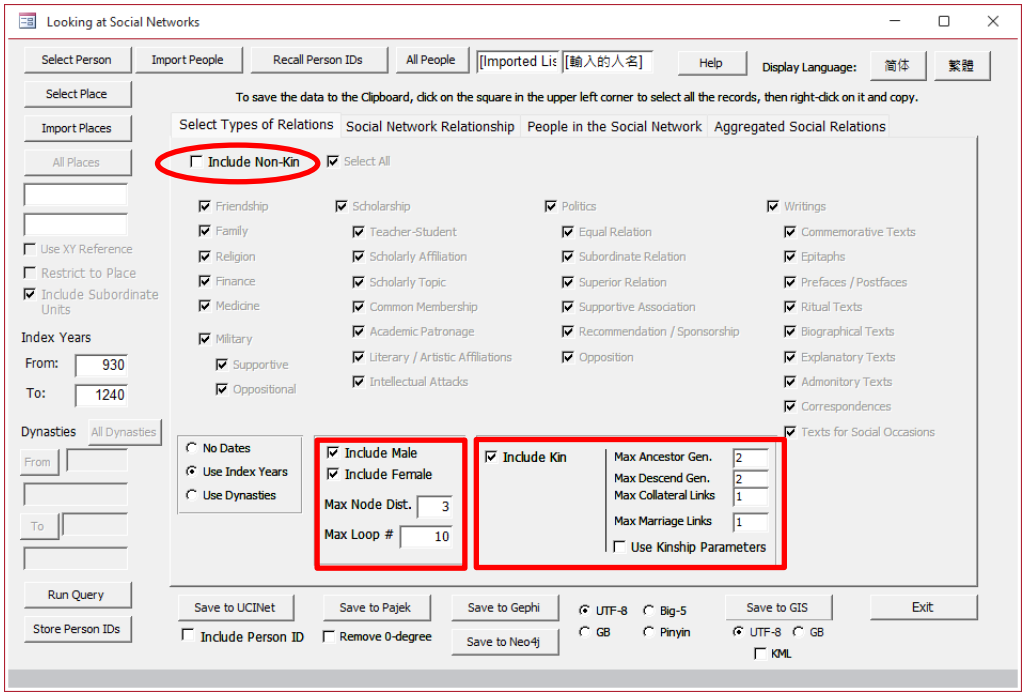


Table of People Participating in the Social Network

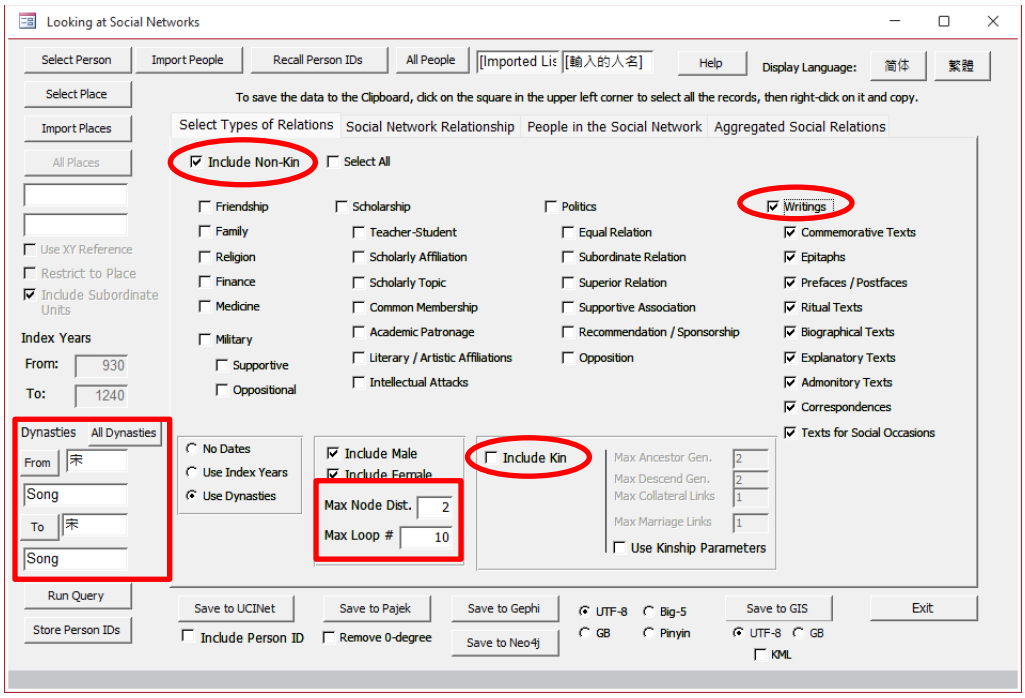
Many of the pairs of people in this list have more than one relationship between them, so CBDB also produces a table in the **Aggregated Social Relations** tab with just one record for each pair of people that gives the number of relations between them:



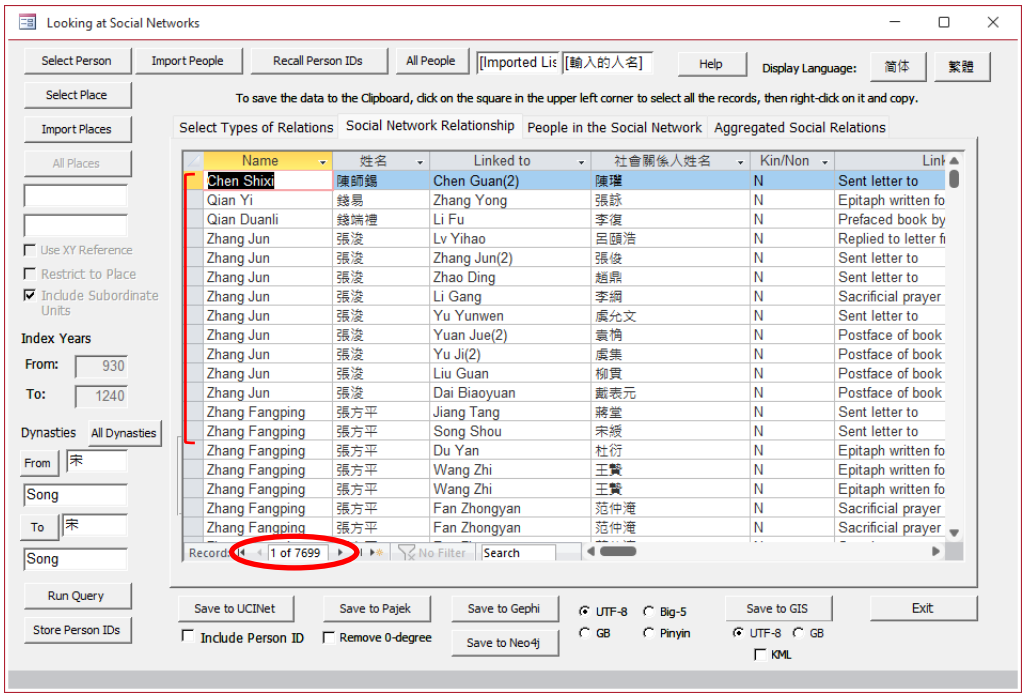
If the network is too large, one can examine more narrowly defined networks. If one looks *just at kinship relations* for the group, CBDB finds 117 people linked through 144 relations (with total node distance of 3 and constraints on the kinship distance for the relationships).



If one looks just at associations formed through writing with a maximum node distance of 2, excludes kinship, and uses dynasty rather than index year, CBDB discovers 7,699 relations (with 3,897 aggregated relations) among 1,379 people:



The results seem promising: not too many links, and not too few:



However, if one scrolls to the right in the table of results and right-clicks on the header of the field called “Edge Distance” to sort the records, one will discover that only the first nine records connect the initial group of people who became eligible for service through legal examinations with other individuals. (These are relations with an “edge distance” of 0, i.e., directly linked to the original list.) Only five of the initial thirteen people have any associations

defined by writings, and these links are to only seven people. Of those seven associations, five are to people (Liu Zhi, Yang Jian, Zhu Xi, Zhang Shi, and Lou Yue) who have vast social networks who contribute most of the relations in the social network. Thus, it perhaps is better to return to the larger set of unrestricted relations among 3,897 people and use the tools of social network analysis to sort through the data.

Requerying

Some users have discovered that it is useful to reuse the people identified in one query in **LookAtNetworks** to serve as the basis for additional queries in the same form. For example, the search for the kinship relations of the men who passed the law examination produced 146. We can look to see if they wrote to one another by first clicking on the **Store Person IDs** command button and then directly clicking on the **Recall Person IDs** command button. This loads the current results as a list of person IDs. One then restricts the non-kin relationships to writing and reruns the query. This has the added virtue that the list of people is now available for use in other forms as well.)

Outputting Results

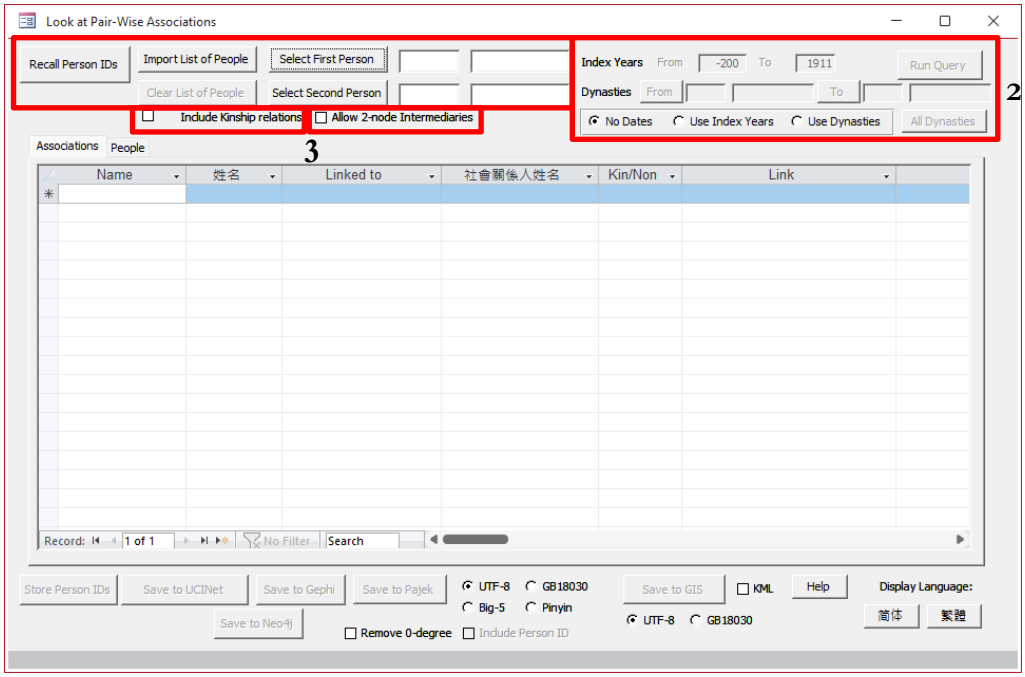
LookAtNetworks provides ways to output the results of a query to three different SNA programs: **UCINet**, **Pajek**, and **Gephi**. Because Pajek supports Chinese characters, CBDB allows the output to Pajek to be in either of three coding systems-UTF-8, Big-5, and GB-or in *pinyin* without characters. GIS software also supports Chinese characters, but how they are handled differs depending on the regional settings of one's computer. The default display for both nodes and edges in the Pajek output files uses color-coding to indicate degree of distance from the target person:

Black = the target node
Blue = nodes at a summed distance of 1
Green = nodes at a summed distance of 2
Orange = nodes at a summed distance of 3
Yellow = nodes at a summed distance of 4
Red = nodes at a summed distance of 5 or more

One also can export a set of CSV (comma-separated values) files for use with Neo4j.

H. Using the Form "Query Pair-wise Associations"

At times one wants to consider whether there were any social links between two individuals or among members of a group of people identified through criteria other than those of kinship or social network. One could use **LookAtNetworks** to generate the social network of one person and see at what point the other person or people appear as part of the network. However, the Access version of CBDB provides a tool to directly examine if there were any connections without going through the general network search.



The form is simple. First one (1) either chooses two individuals or imports a list of people, or recalls either *a single person* stored from the **Browser** [who becomes the "First Person"] or *a list of people* from earlier, saved query results using the procedure described for other forms above, then (2) the range of dynasties or index years for the people in the relations, if desired, and finally, (3) the type of permissible relations. The options for relationships are:

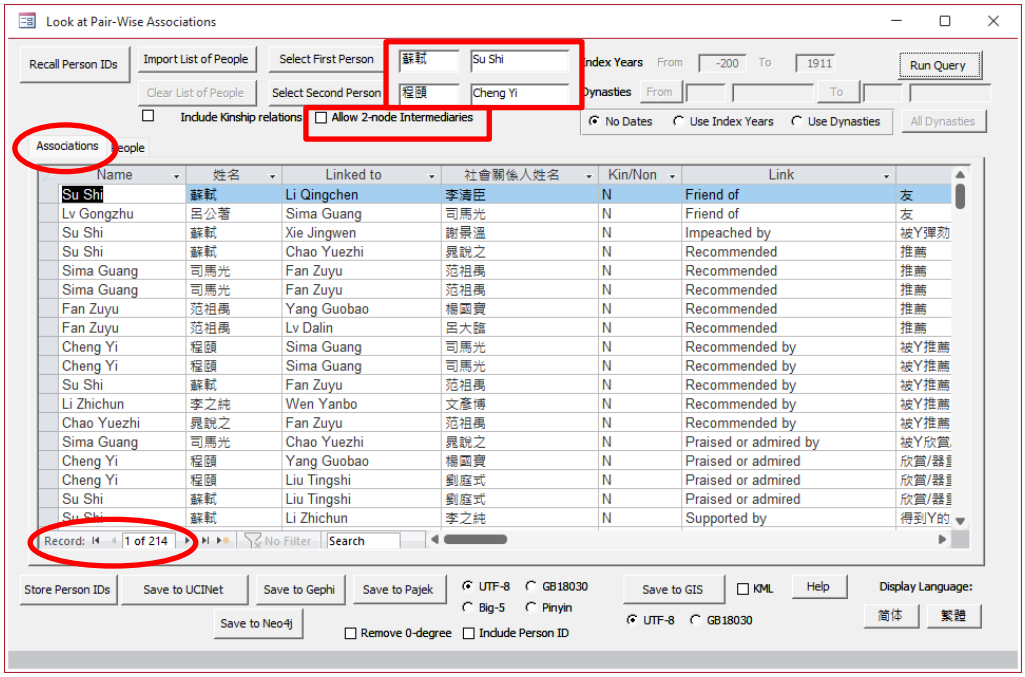
1. **Allow 1-node Intermediaries**: That is, people who are directly linked to both (or, for imported lists, two) of the selected people: Person A — Node1 — Person B. In this case one leaves the check box for two-node intermediaries unchecked.

2. **Allow 2-node Intermediaries**: Here one allows people linked to one person who in turn have links to people linked to the second person (or to another person on the imported list): Person A — Node1 — Node2 — Person B. In this case one clicks on the check box for two-node intermediaries to select the option.

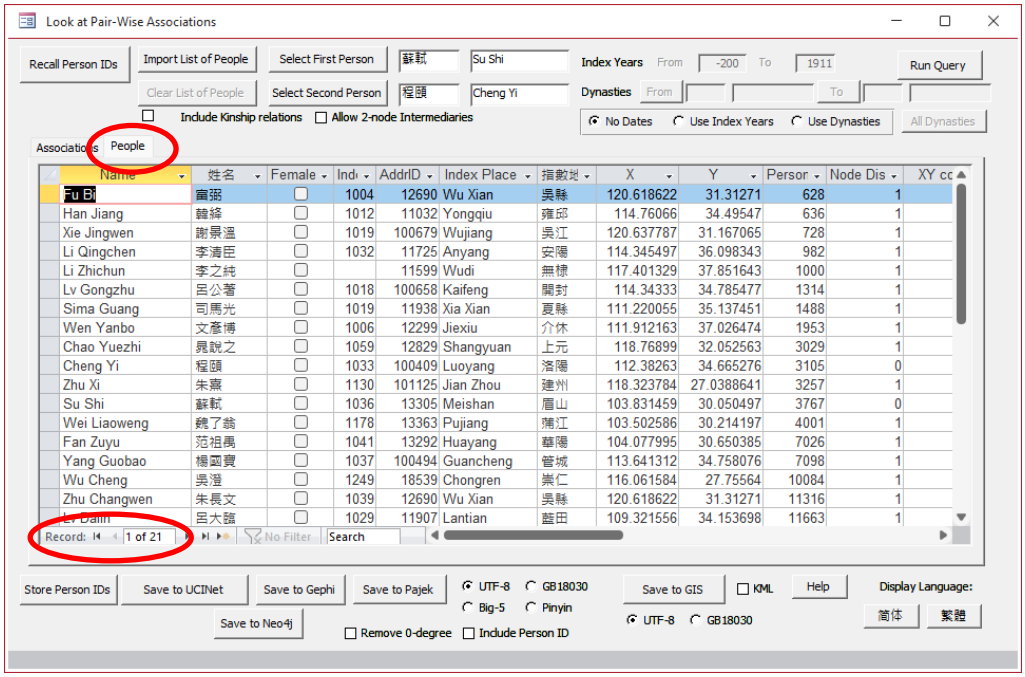
3. **Include Kinship relations**: The default is simply to look at social (non-kinship) relations connecting people, but kinship also can be important, and the form allows one to examine the role of kinship relations in the social network.

One Node Intermediary Searches

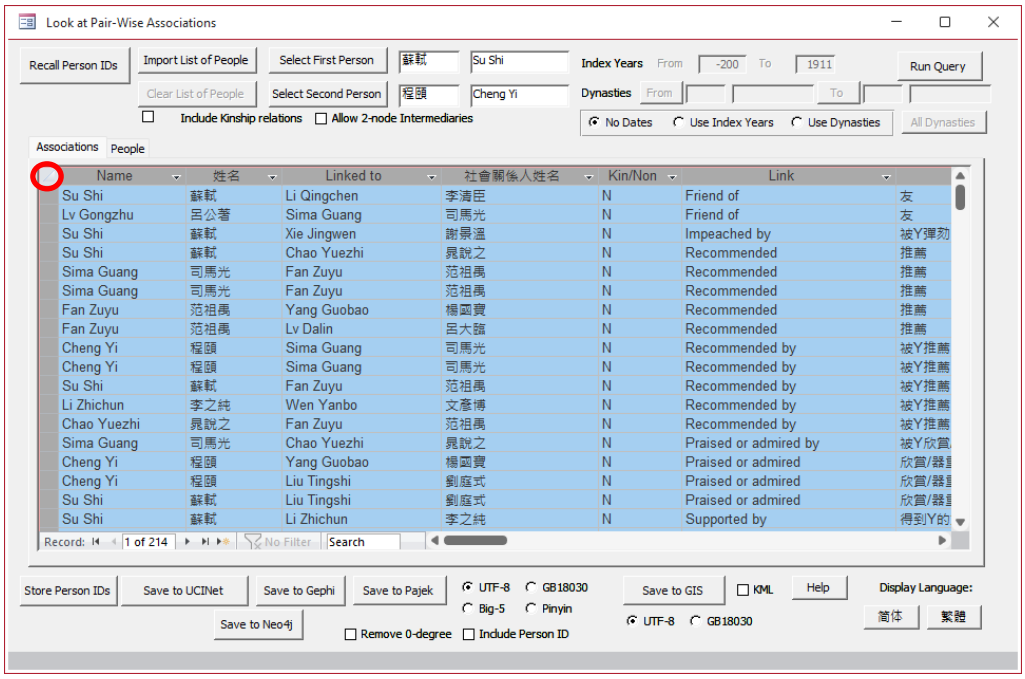
For example, if one explores the links between Su Shi 蘇軾 and Cheng Yi 程頤, allowing only people directly linked to both of them finds 214 associations among 21 people.



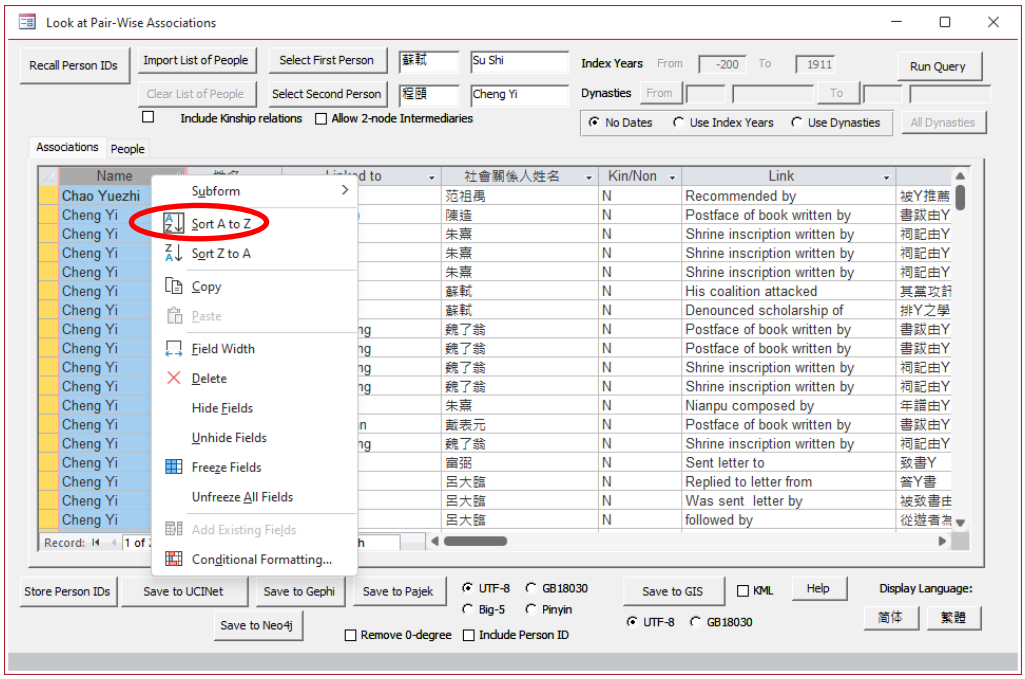
As in **LookAtNetworks**, the form provides two output tables: "Associations" for the relationships, and another, "People in Associations," for the people in the relations.



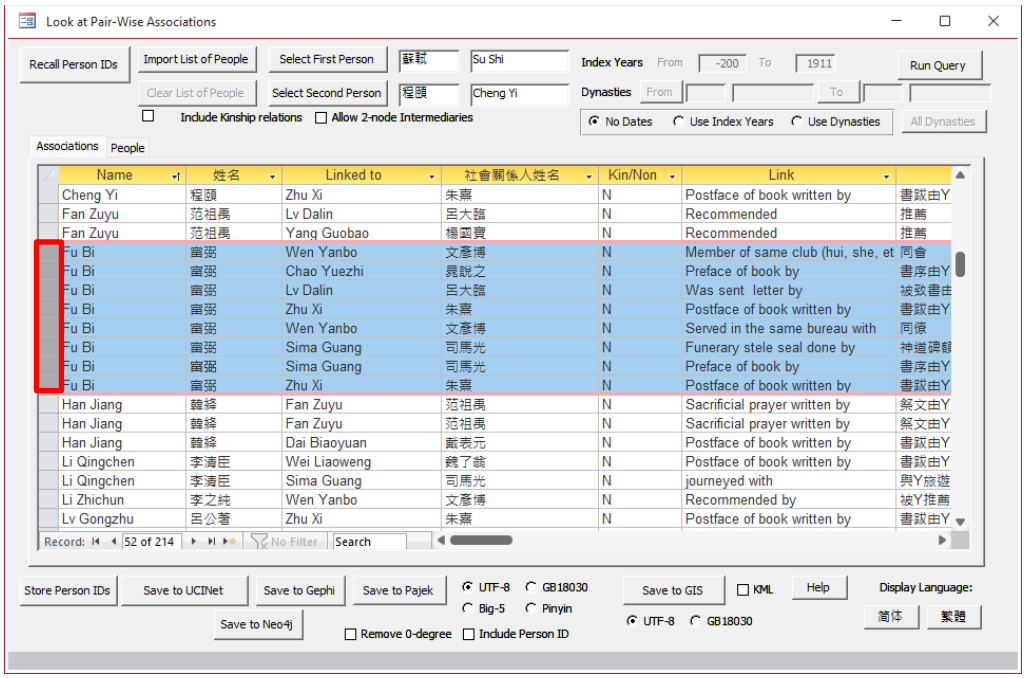
As with the other forms, one can save the results of a search by clicking on the grey square in the upper left hand corner of the table to select all the records and then using Ctrl-C:



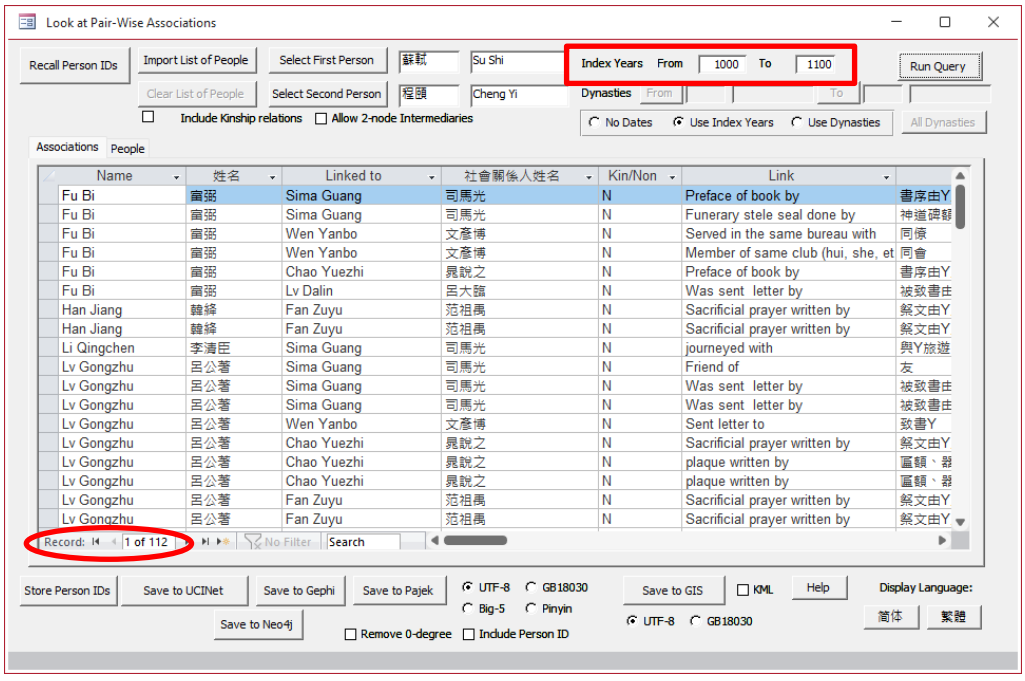
One also can sort on a column of the table by clicking on the column (in this case, "Name") to select it, then right-clicking to choose the type of sort:



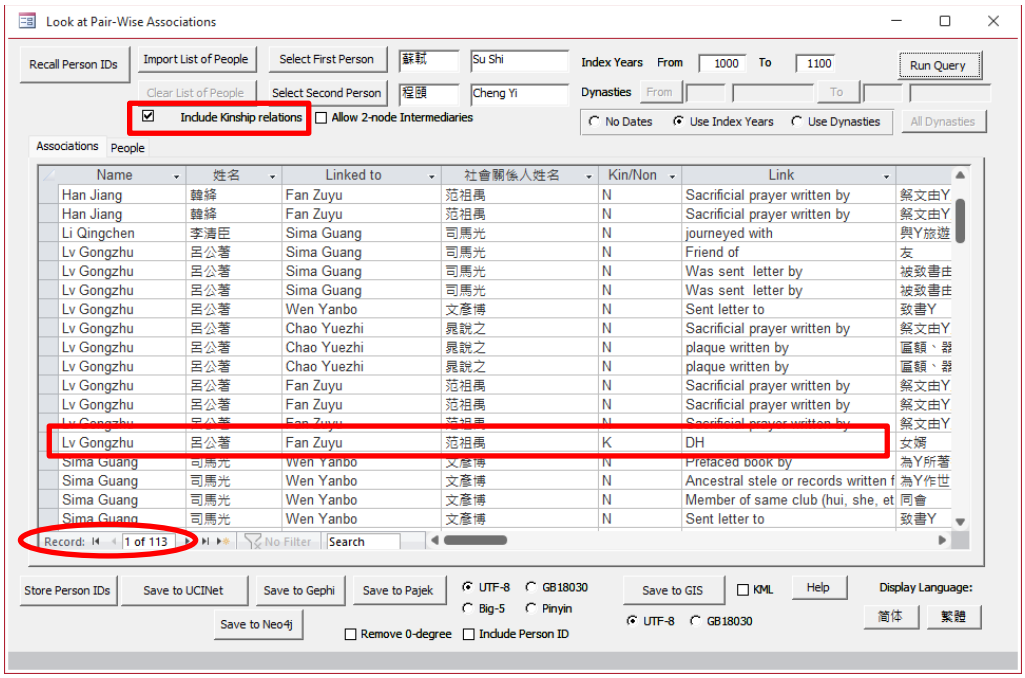
One also can select a block of records to save by clicking the mouse on the left-hand grey column of the first record in the block and then, with the left-click button still held down, dragging the mouse down the grey column to the last record in the desired group:



However, note that the entry directly below the selected block includes Dai Biaoyuan 戴表元 (1244-1310), a late Southern Song figure. If one wishes to narrow the search to intermediate nodes who are roughly contemporaneous with the target people, one can use **index years** to limit the search. (Using **dynasty** as a filter does not help.) If one limits the index years to a range between 1000 and 1100, one finds fourteen people with 112 relations connecting them:

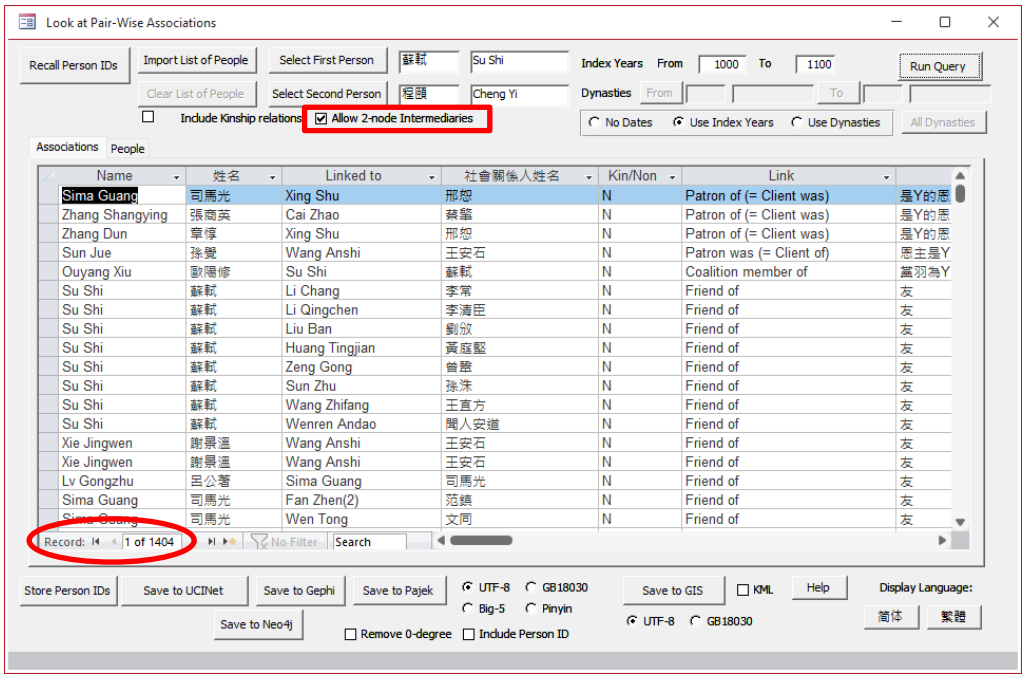


If one then includes kin of either Su Shi or Cheng Yi who have a social connection to the other, then one discovers one additional connection but, in this case, no additional people:



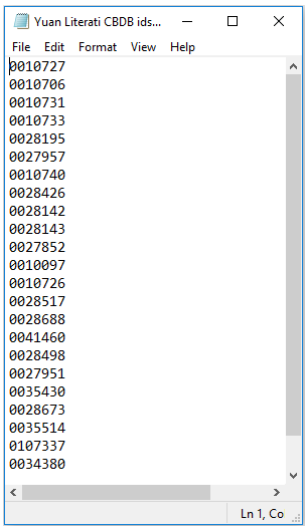
Two Node Intermediary Searches

If one broadens the search to allow two intermediary links to connect the target people, the network becomes more complicated: The program reveals 1404 relations among 123 people with index years between 1000 and 1100:

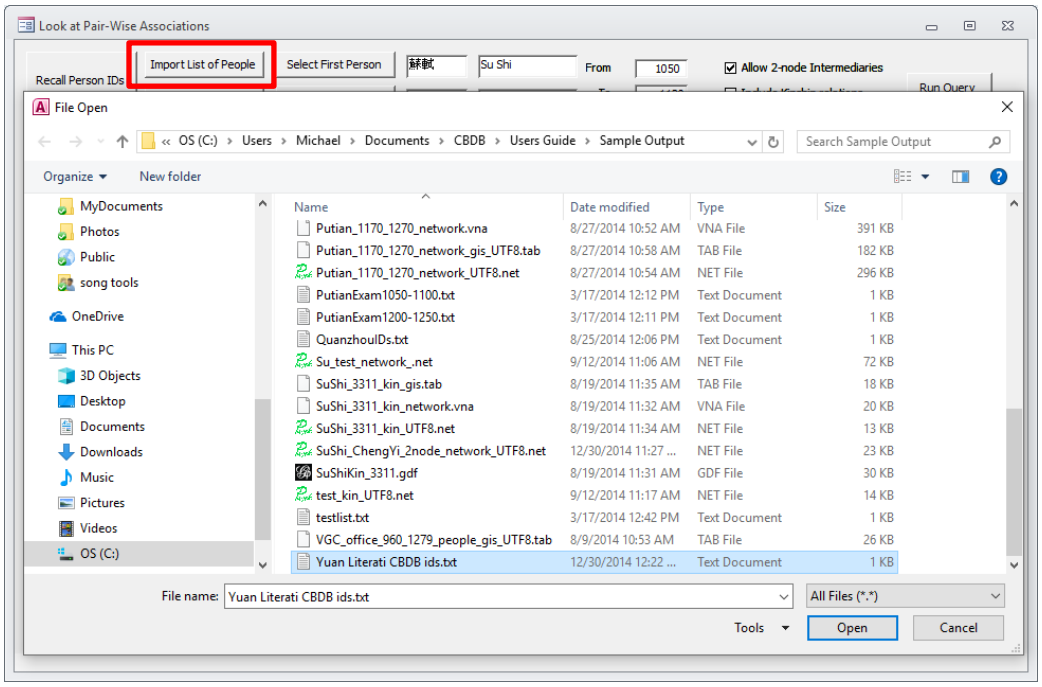


Searches Using Lists

If one wants to look for connections within a larger group of people chosen by other criteria, the form allows one to import a list of person IDs. Here one looks at Jinhua men who from the Yuan dynasty who have extant collections. As in all lists for importing people, CBDB requires a single column of IDs in ANSI encoding:

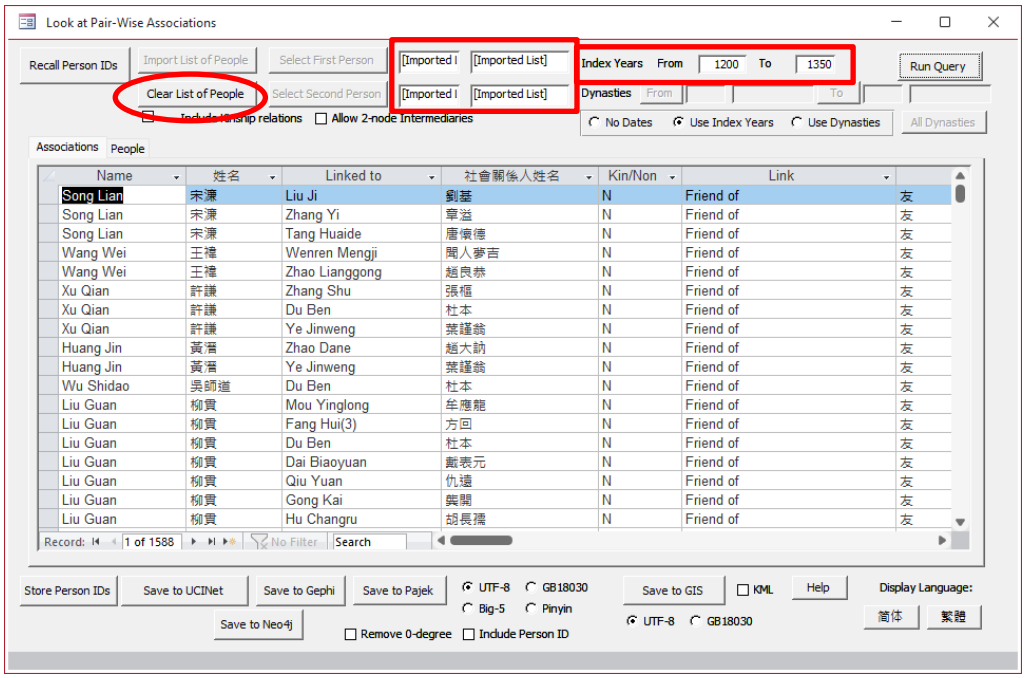


One clicks on the **Import List of People** command button and locates the file:



If the file is successfully read, the form indicates that the names are from an imported list. To clear the list and return to selecting people through the two **Select** command buttons, simply click on the **Clear List of people** command button.

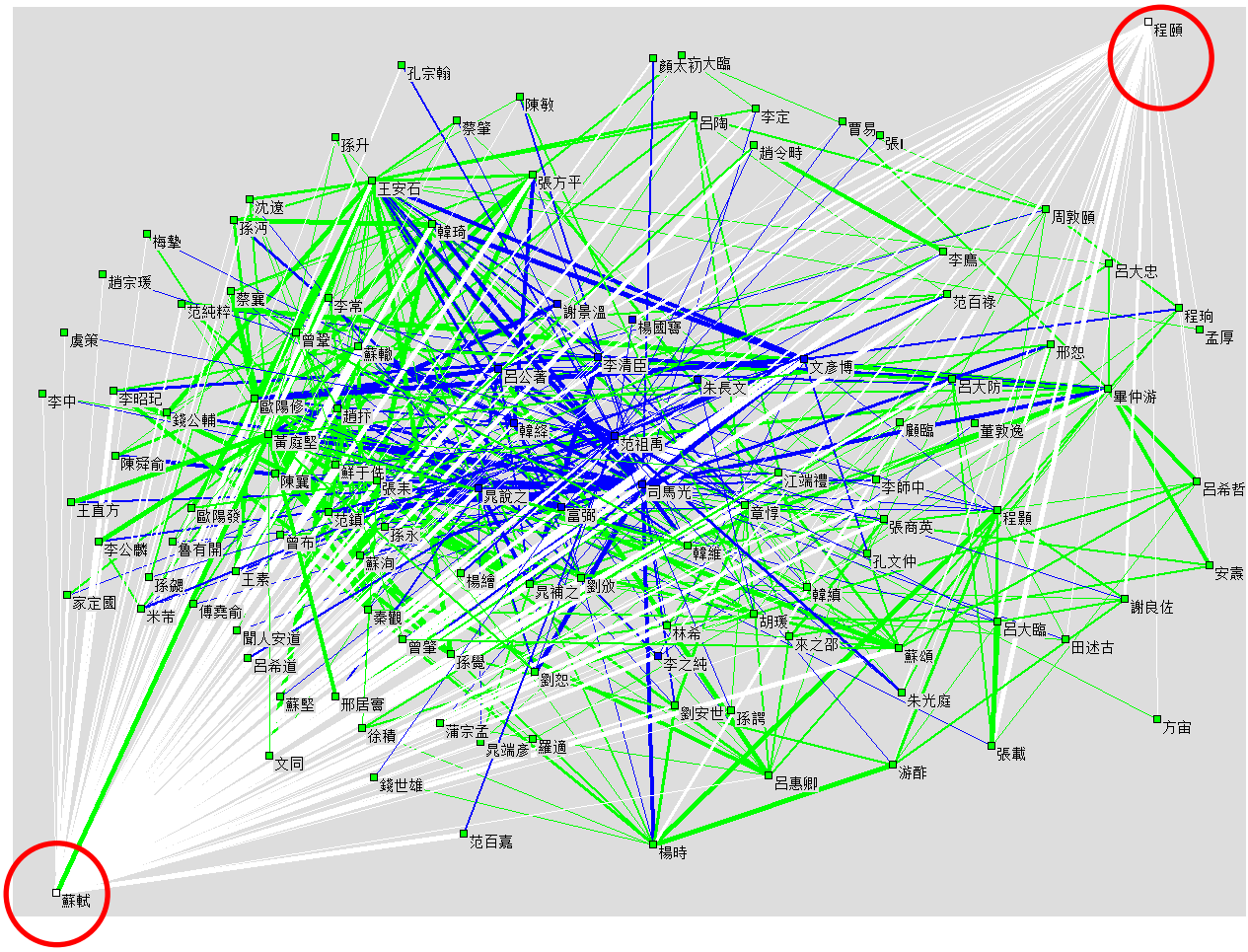
Once one has imported the list, the search procedures are the same. In this case, the query is set to look for one-node intermediaries with index years between 1200 and 1350 and produces 1,588 associations among 187 people:



Output to SNA and GIS Programs

Like the other forms, LookAtAssociationPairs can generate files for use with Pajek and with GIS visualization programs. The output tables for Associations and People are the same as those in LookAtAssociations. Please consult the information in that section of the User's Guide.

Allowing the form to list all the relations between the 1-node and 2-node intermediaries between Su Shi and Cheng Yi who have index years between 1050 and 1120 intermediaries produces a network that can be imported into Pajek.



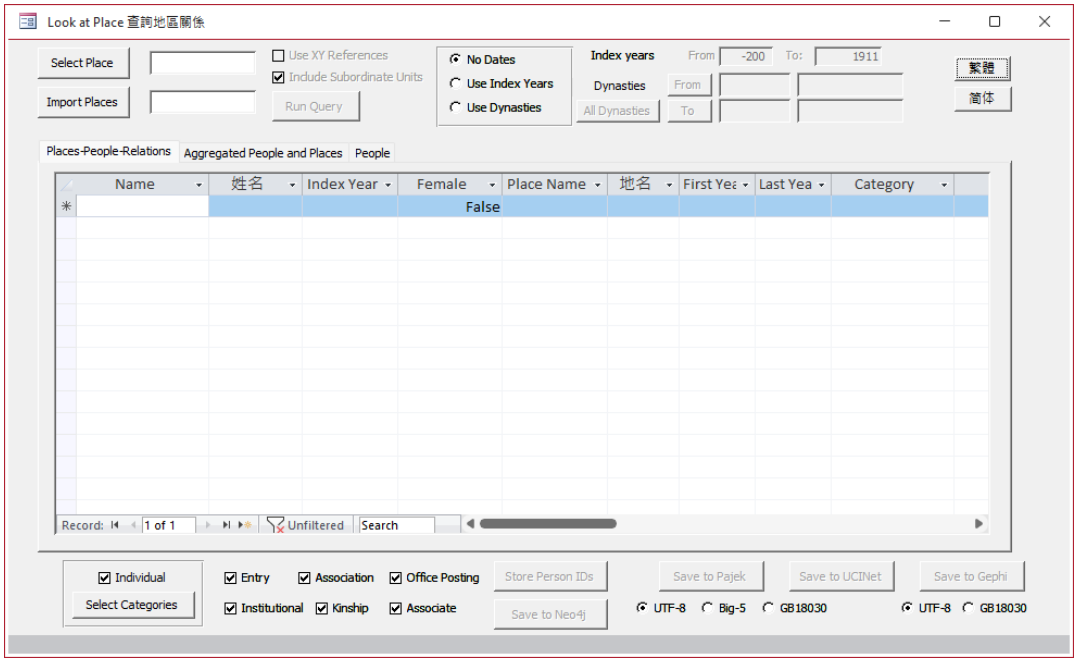
The default display for both nodes and edges in the SNA output files uses color-coding to indicate degree of distance from the target person and the type of connections:

	Nodes	**Edges**
White	= the target nodes;	from target nodes
Blue	= nodes that serve as 1-node intermediaries	from 1st order to 2nd order nodes
Green	= nodes that serve as 2-node intermediaries	between 2nd order nodes (except for one mysterious line to Su Shi)

The output files aggregate the associations between people, and the width of the lines reflects the number of associations between nodes.

I. Using the Form "Query Place Associations"

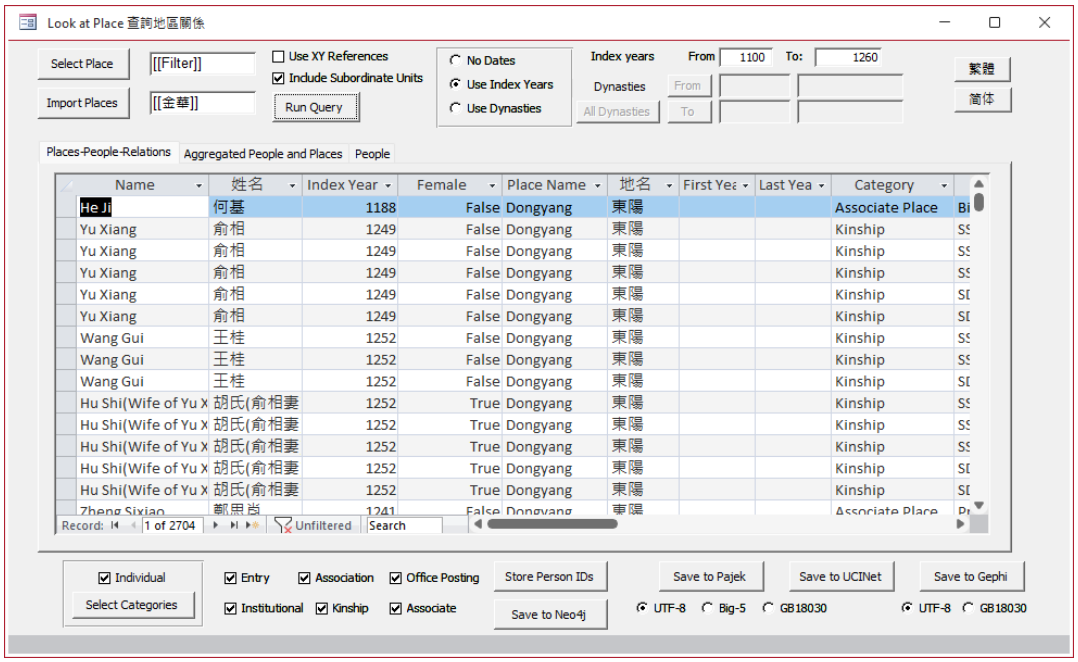
The forms discussed above produce information about the relationship between people and places in the contexts of kinship and social relations, office holding, and entry into government. It may be useful to see how people and place come together in a more synoptic view. For example, one person may have been in office at a place which was the place of registry of the kin of a friend. This sort of drawing together of connections proves difficult without a way to aggregate information about a place over time.



Thus CBDB provides the form **LookAtPlace**. The form can trace seven types of relationship to place:

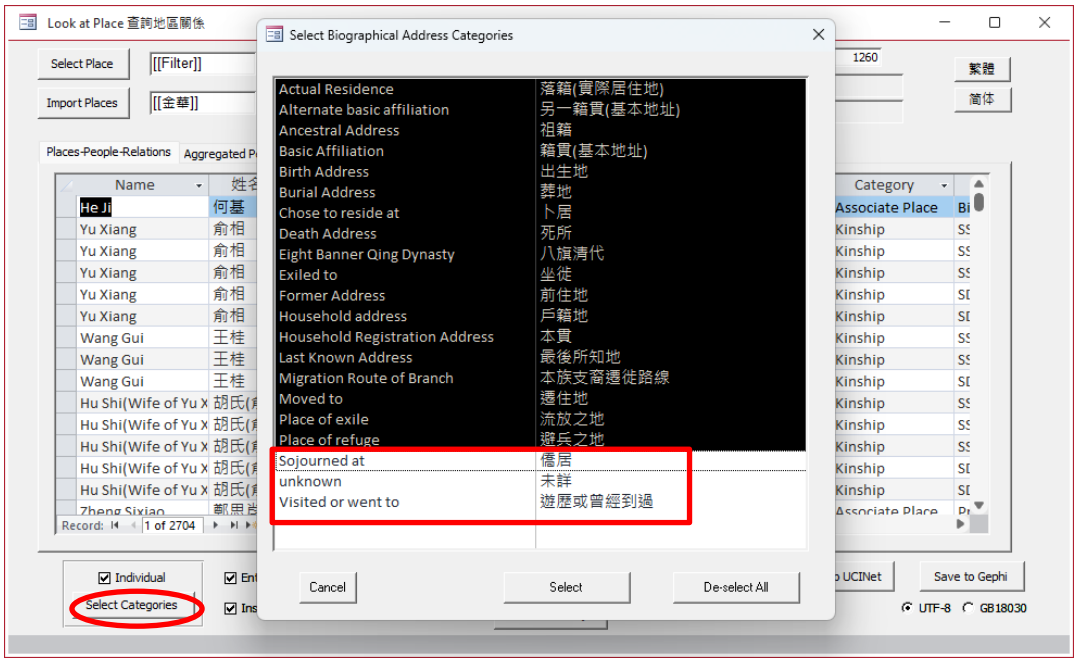
1. Biographical Data: was this place the index place of the person? Did he or she move there?
2. Entry Data: did the person take an examination at this place, or was this place otherwise associated with the person's entry into government service? (At present CBDB has very little data on this type of relationship to place.)
3. Connection via Kinship: who were the kin of people from this place?
4. Connection via Association: who had associations with people from this place?
5. Place of Association: what social connections were created at this place? (At present CBDB has very little data on this type of relationship to place.)
6. Office Posting Data: who held office at this place?
7. Institutional Connection: who were associated with social institutions at this place?

The query below looks at Jinhua for people with index years between 1100 and 1260.

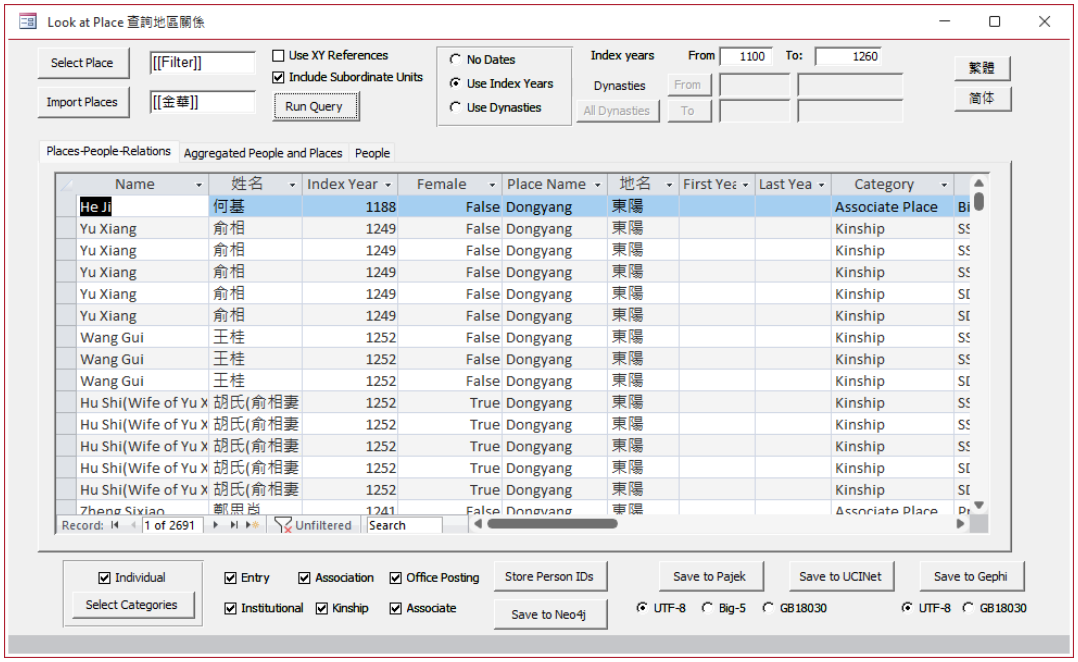


One can select which relationship to place to include in the search and can specify the usual sorts of parameters (use of *dynasty*, *index years* and the use of XY references). As with the other forms, one also can use a filtered list of place names or import a list of address IDs.

In addition, because the categories of relationship of people to place include that may not be relevant to the particular query, the form allows the user to select the categories of relationship to be used in the search:



One clicks on the **Select Categories** button, which opens a form. One can "Select All" and then click on those categories to *not* be included, clicks on the **Select** button to close the form and runs the query:



In this particular search, removing the categories eliminates just 13 records.

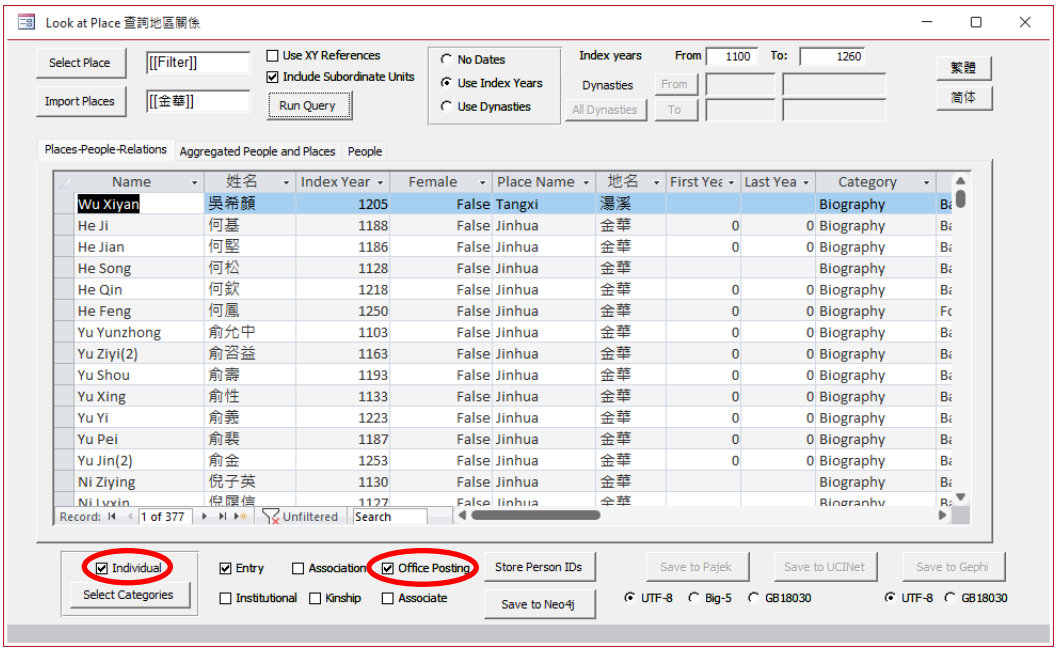
The output table has 17 fields:

1. Person name (Pinyin)
2. Person name (Chinese)
3. Index year
4. Place Name (Pinyin)
5. Place Name (Chinese)
6. Associate Name (Pinyin)
7. Associate Name (Chinese)
8. First year
9. Last year
10. Category of Place Association
11. Relation to Place within Category (English)
12. Relation to Place within Category (Chinese)
13. X coordinate
14. Y-coordinate
15. Index Year Type (English)
16. Index Year Type (Chinese)
17. Index Year Type Code

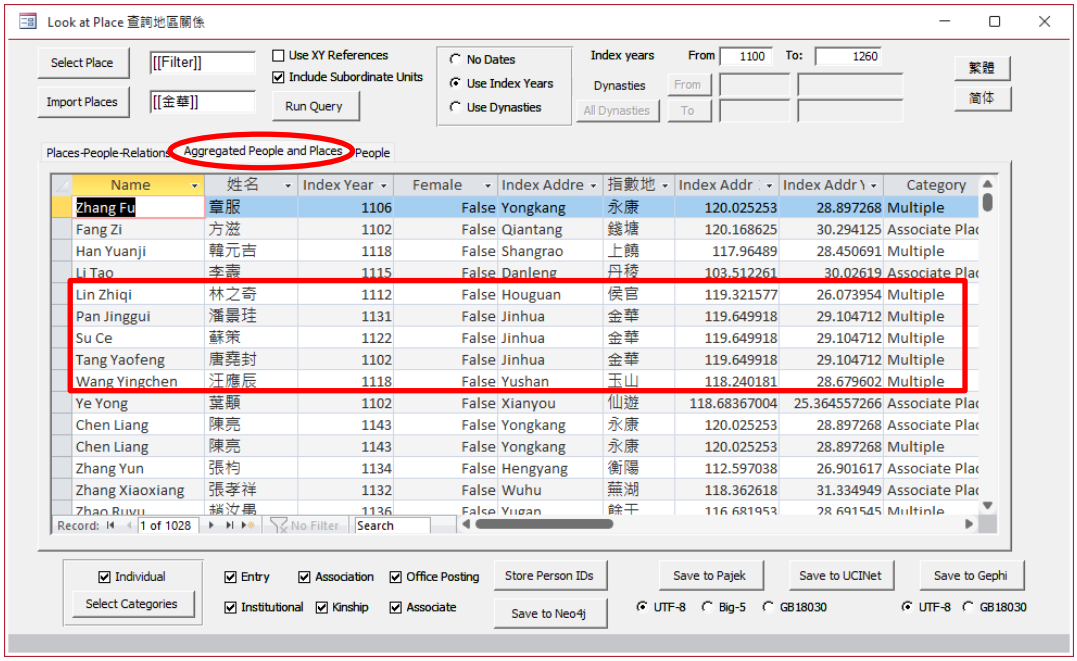
The *Category* specifies which of the seven types of relations to place is recorded for the person, while the *Relation* gives the specific information within the category. Thus the *Category* of

"Biography" indicates the person's immediate biographical relationship to place, and the *Relation* provides the detail ("basic affiliation," "moved to," etc.). Similarly, the *Category* of "Associate Place" records that the person is from the selected place, the *Associate* has a social connection to the person, and *Relationship* provides the details of the relationship.

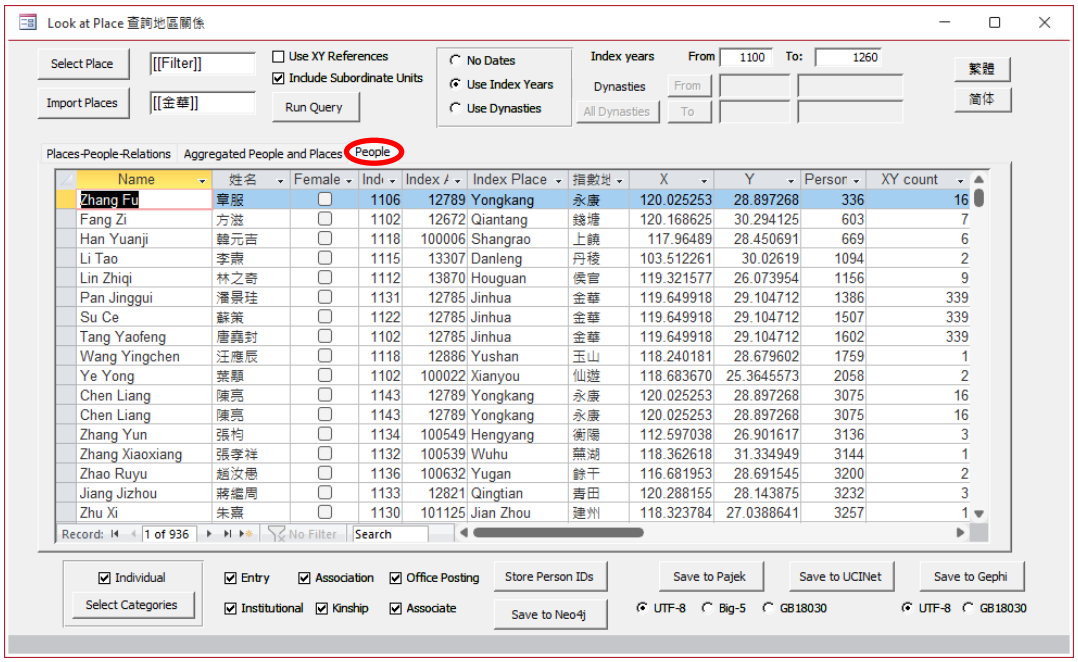
If one wishes to look at specific types of relationship to place, one can select the type of relationship(s). Below is the result when one chooses *Individual* and *Office Postings*:



People may have more than one relationship to a place, and the form provides a table that list the **Aggregated People and Places** relations and reveals many people with multiple types of relationships to Jinhua:



Finally, the form provides a table that lists the people who participate in the relationships:



At present, there are two ways to write the results of a search to a file:

a. The first is as **SNA** data in Pajek, UCINet, or Gephi format. This option is available only when one has selected *Association, Associate,* or *Kinship*.
b. The second option is to write a series of CSV files for **Neo4j**. The number of files produced through this option depends on the types of relationship to place the user has selected.

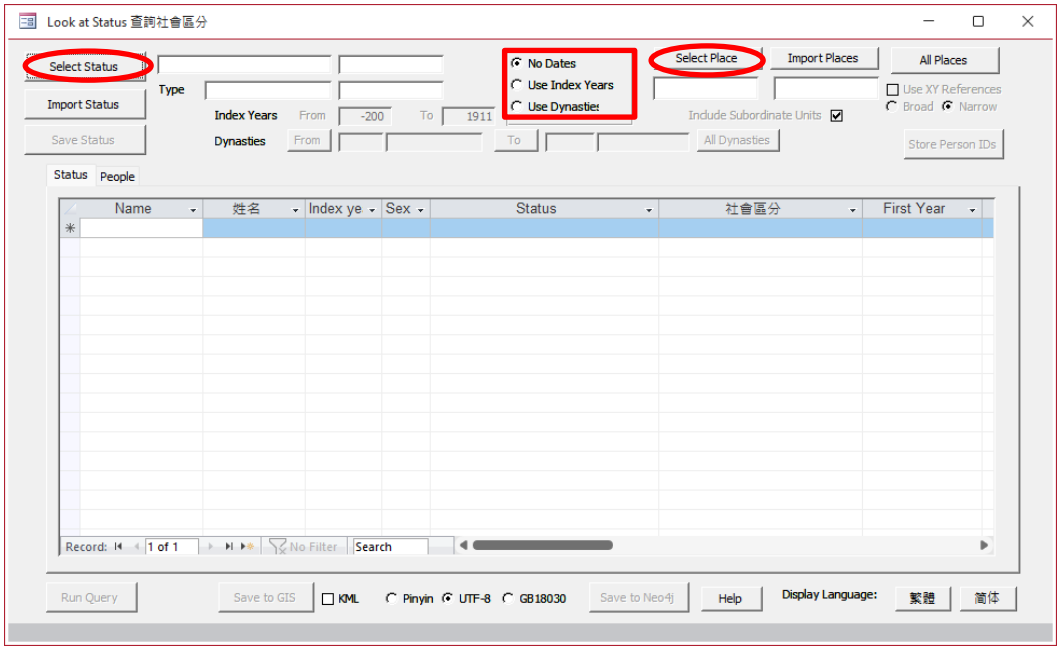
If there is a need to save the data in GIS form, this functionality can be added in future versions of the software.

J. Using the Form "Query Status"

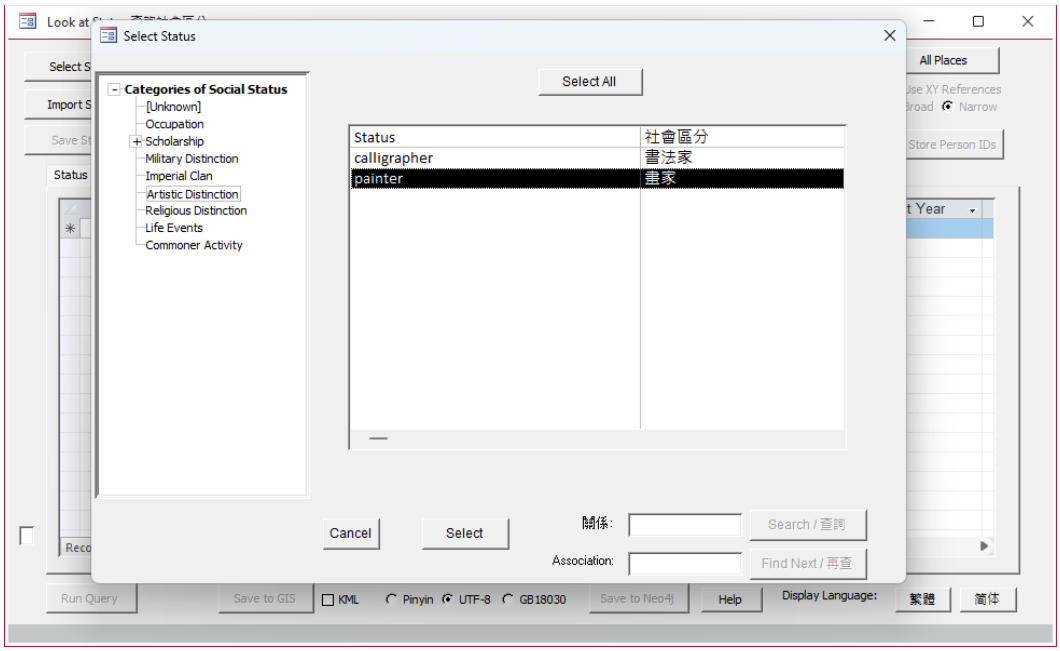
LookAtStatus is a recent addition to the forms for exploring the CBDB data. It allows users to examine CBDB information on social distinctions recorded for members of the database. As explained in Chapter 2, **status** records ways in which individuals gained reputations in their communities. At present we have 285 codes divided into 7 categories:

Occupation	事業
Scholarship	學術
Military Distinction	武功
Imperial Clan	宗社
Artistic Distinction	藝術
Religious Distinction	宗教
Life Events	時事
Commoner Activity	布衣事

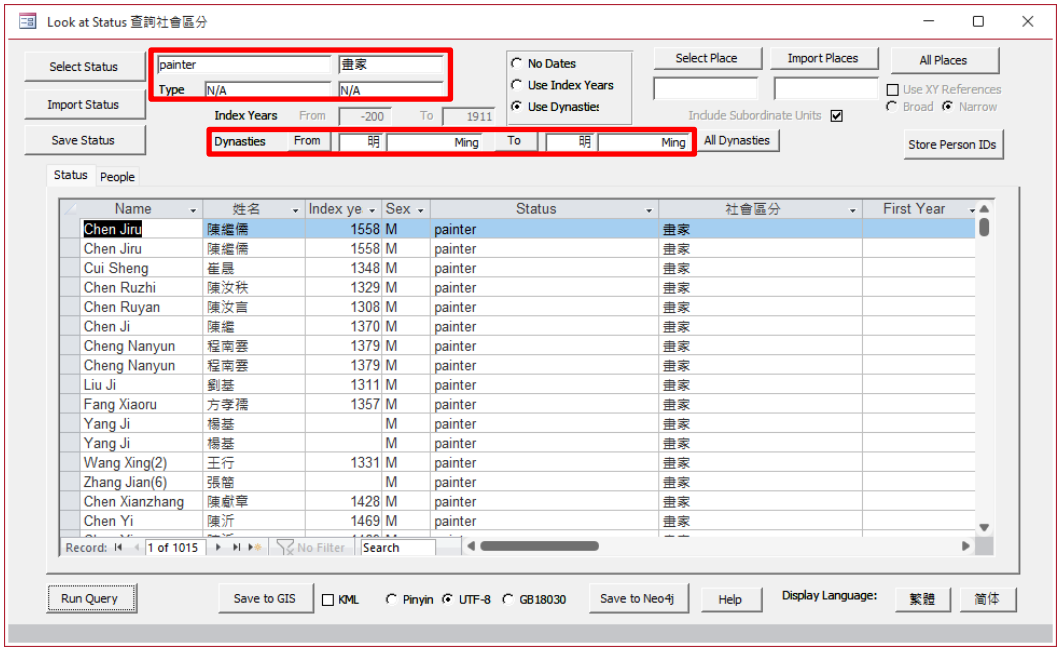
The form shares the features of the other forms. One can filter by *dynasty* or *index year*. One can select an *index place* (or group of index places) to explore. And one can store the person IDs to use in other forms.



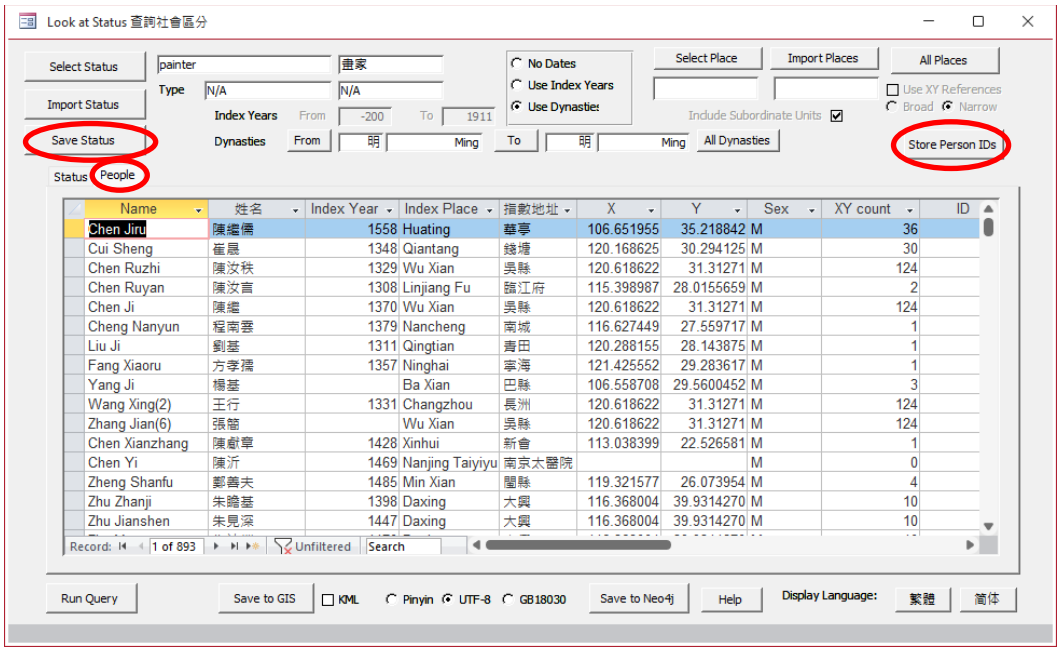
One begins by selecting the category of status one seeks to explore. Since, at present, there are 275 codes for status, the **Select Status** form, like the other forms, organizes the codes into larger groups of types of status. As with the other selection forms, one can select an entire category of status relations, or one can choose one or more specific status relations.



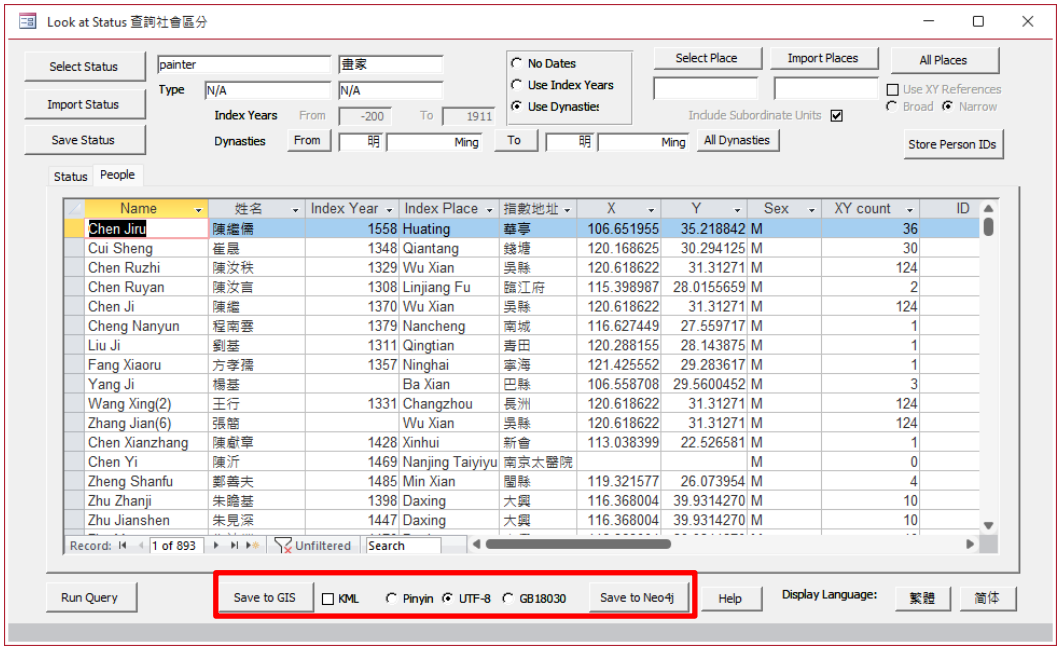
Below is the list of 1,015 records for social distinction through painting for individuals in the Ming dynasty.



If one has selected just one type of status, in theory, the number of records in **Status** and **People** should be the same. Note, however, that there are *duplicate records* in the **Status** table. This is a bug that will be fixed in the next release of the data. As the **People** table shows, there are only 893 individuals who have status as painters in the Ming dynasty.



One can save the IDs of the people for use in other forms by clicking on the **Store Person IDs** button, and one can save the list of status codes from the query to a file for reuse later by clicking on the **Save Status** button and saving the file.



The form provides output to GIS data files as well as to Neo4j files, which capture the bipartite person-status relationship (that is, people are connected as nodes to status types as nodes rather than as people connected to other people).

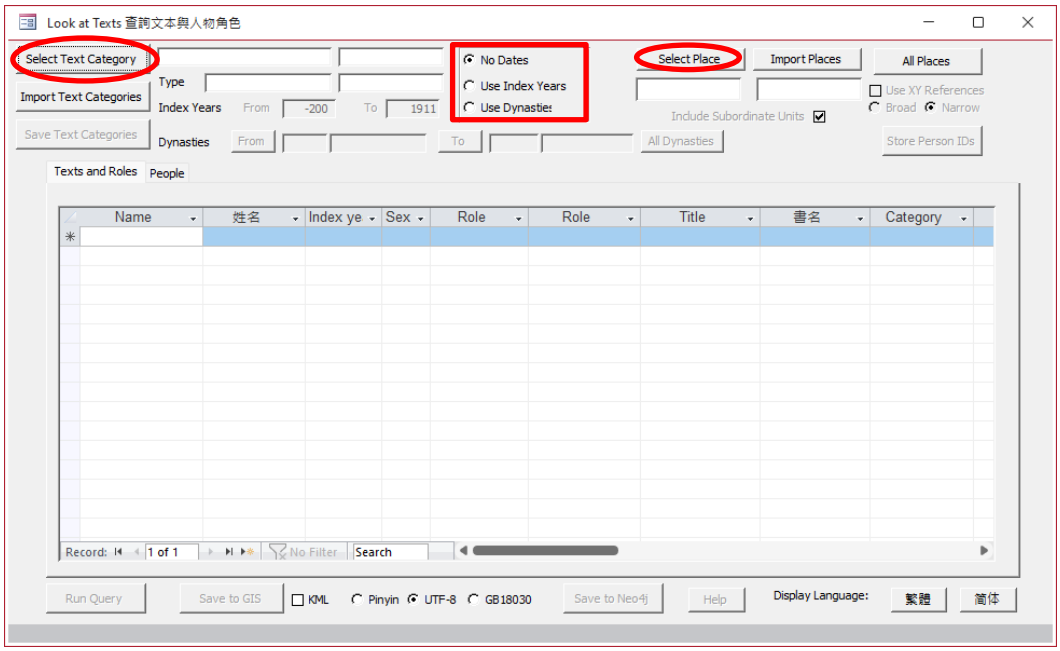
K. Using the Form "Query Texts and Roles"

The form **LookatTextRoles** enables users to investigate people who have roles in the production of premodern Chinese texts. CBDB uses the classification of texts used in the *Siku quanshu*. The roles connecting people to texts in CBDB are:

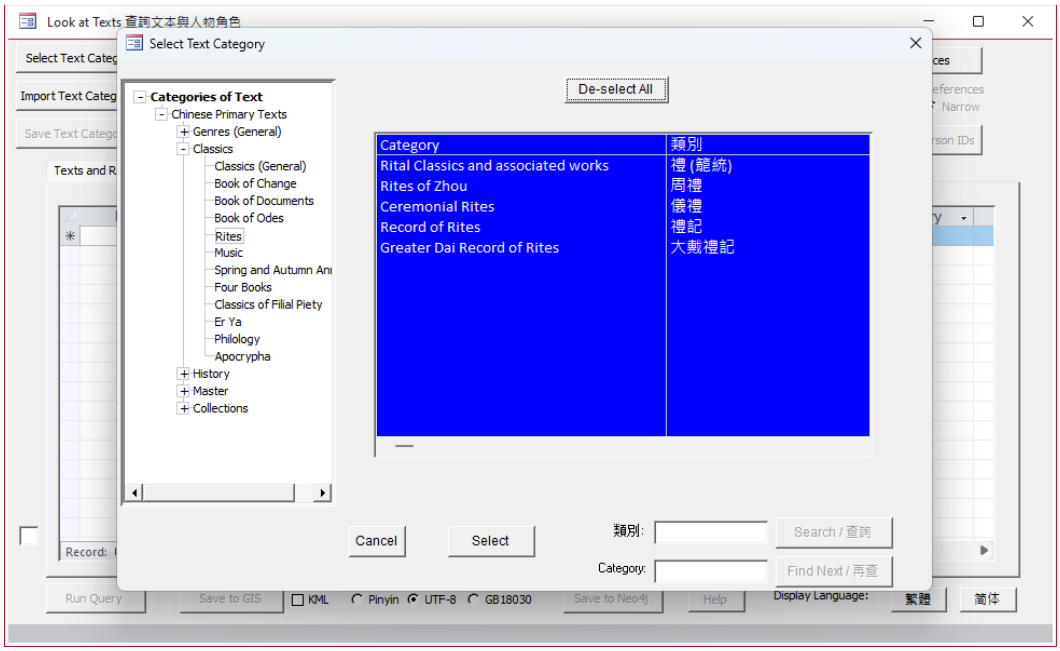
Annotator	註疏者
Author	撰著者
Commentator	註釋者(含評點者)
Compiler	編纂者
Donor	捐助者
Editor	編輯者
Editorial Associate	編輯助理
Proofreader	校對者
Publisher	出版者
Translator	翻譯者
Work included in	收入Y集

At present CBDB has approximately 31,000 records for people in relation to texts.

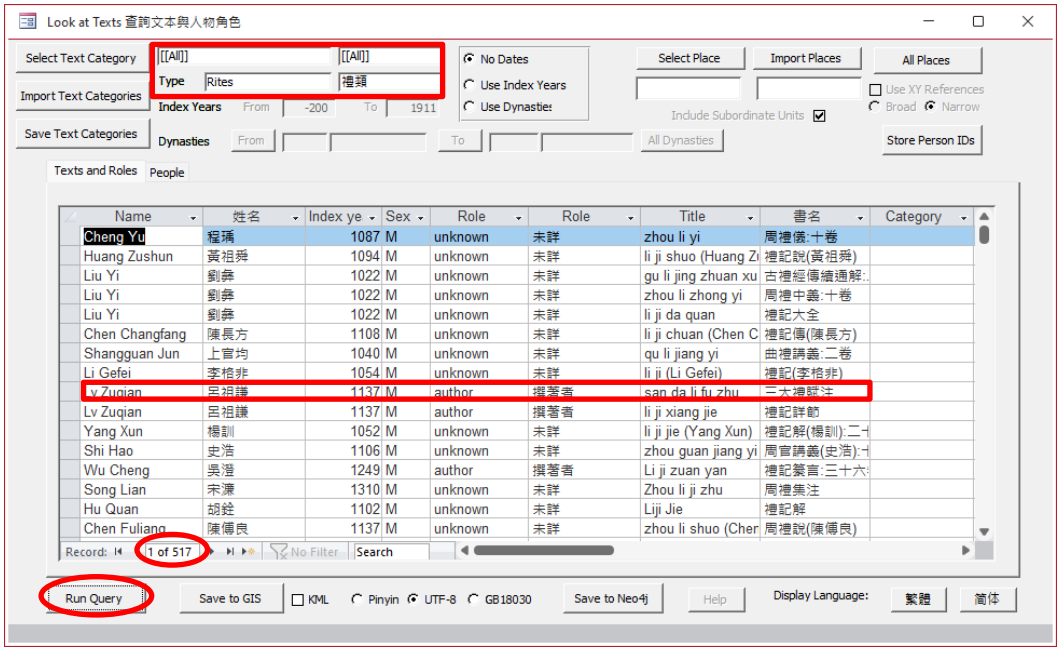
The design of **LookAtTextRoles** is very similar to that of other forms. It uses the same sorts of filters: by *index year*, by *dynasty*, and by *index address*.



One first selects a category of texts:

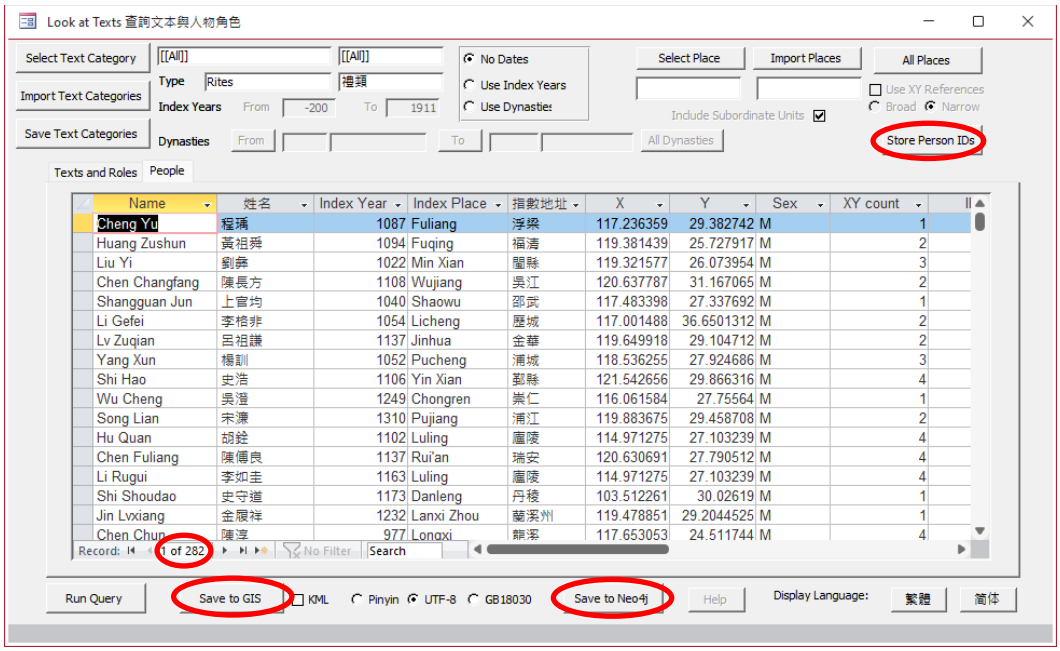


In this example, the user selects all the texts in the category of "Rites." In the simplest query, one uses just the category without additional filters:



Running the query produces a list of all the role in which people participated in the production of texts of the selected category for which CBDB has data. There are 517 **roles** in which 282 people participated in producing texts on the ritual classics in the Confucian canon. In the example, Lv Zuqian 呂祖謙 is identified as the author of the *San Da li fuzhu* 三大禮賦注.

The form also provides a list of all the people who participated in these roles:

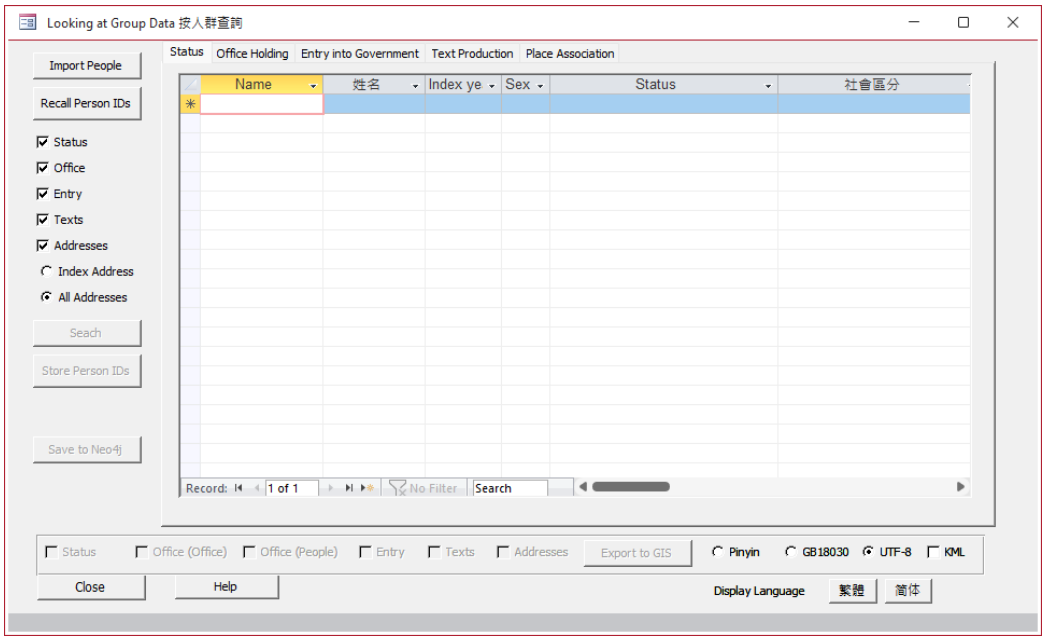


These people can be stored for further analysis using other forms by clicking on the **Store Person IDs** button and can be saved to a GIS file to look at their geospatial distribution with the **Save to GIS** button.

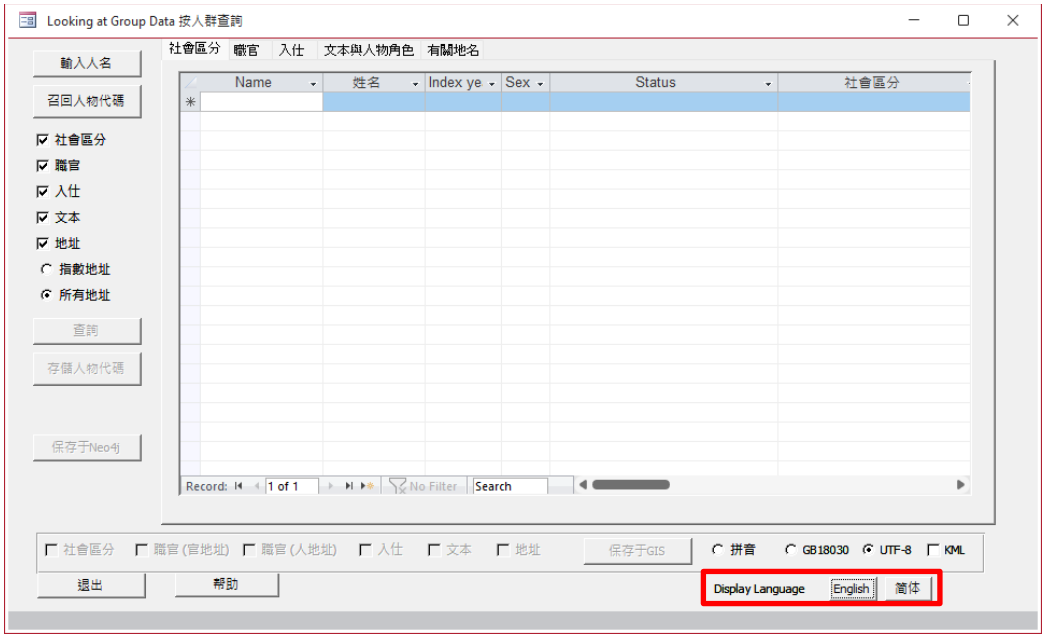
Output to a set of Neo4j files by clicking on **Save to Neo4j** gives one a way to further explore the bipartite relationship between people and texts using Neo4j.

L. Using the form "Looking up Data on a Group of People

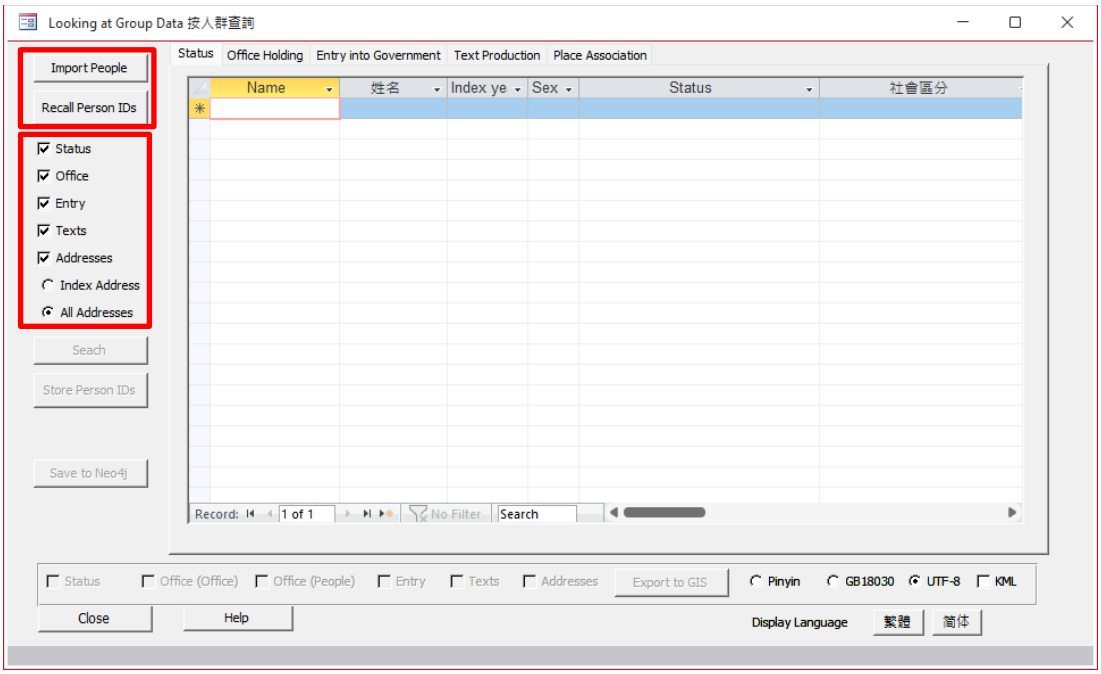
As shown in the description of the other forms, CBDB allows the user to identify groups of people according to specified characteristics. The form **LookAtGroupData** allows the user then to quickly get additional data on these groups of people.



Like the other forms, the user can change the labels from English to either traditional or simplified Chinese:



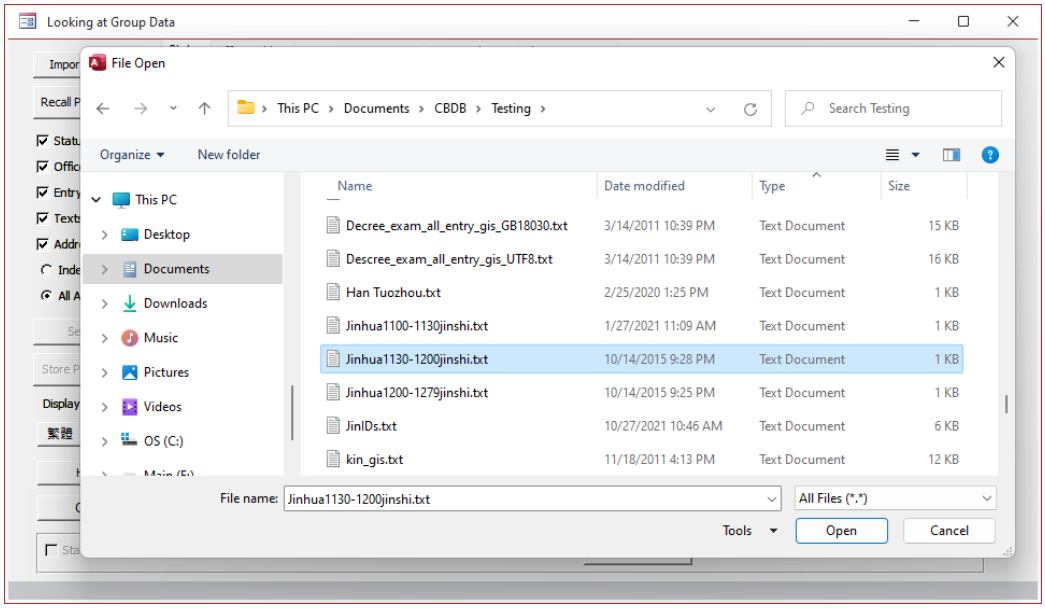
To input the group of IDs to be examined, one can either recall a *stored list* or *import a list* from a file:



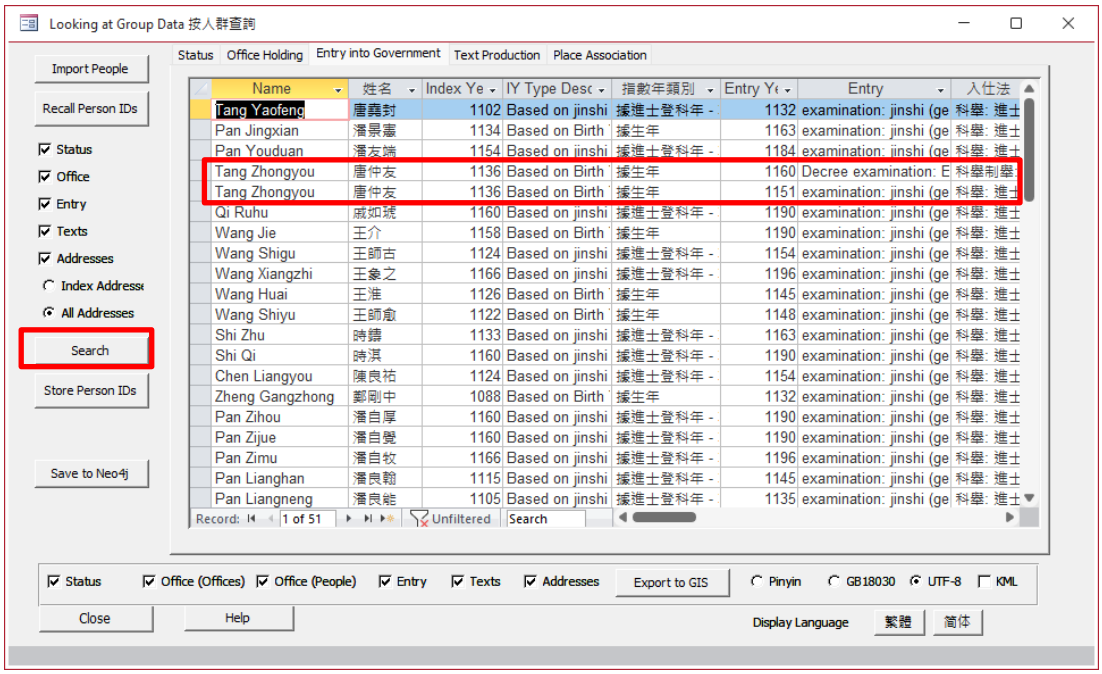
Once one has selected the group of IDs, one then selected the types of information to explore. The choices are:

a. status,
b. office holding,
c. mode of entry into government,
d. textual production, and
e. associations with place (In some cases, one wants to know just the index addresses for the people in the group, and the form allows the user to select this option.)

One can, for instance, import a list of the people from Jinhua County who earned *jinshi* degrees between1130 and 1200:

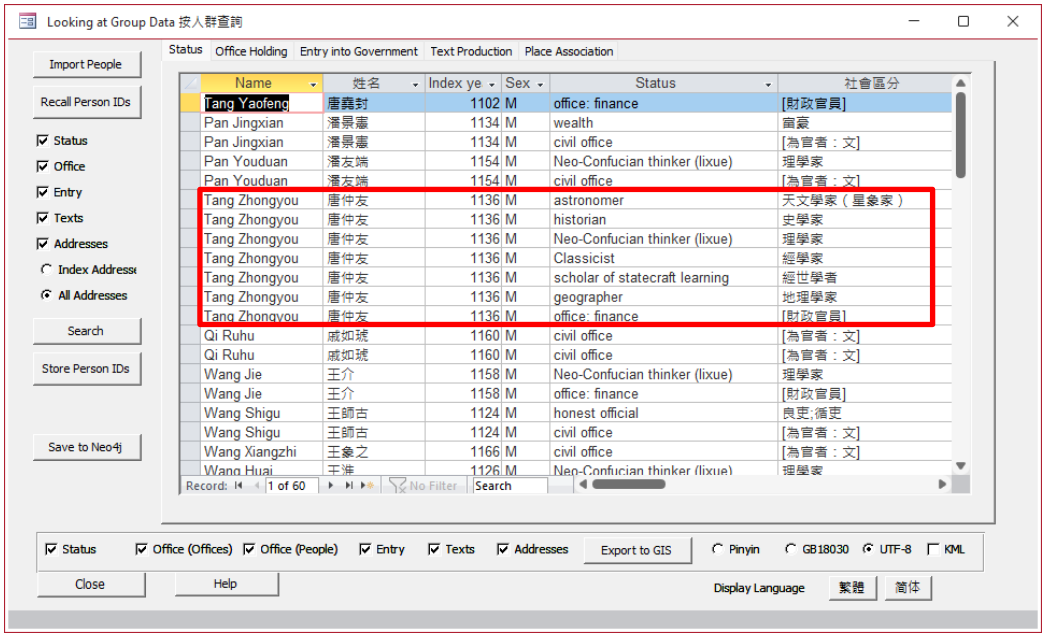


One selects the types of data and clicks the **Search** button:

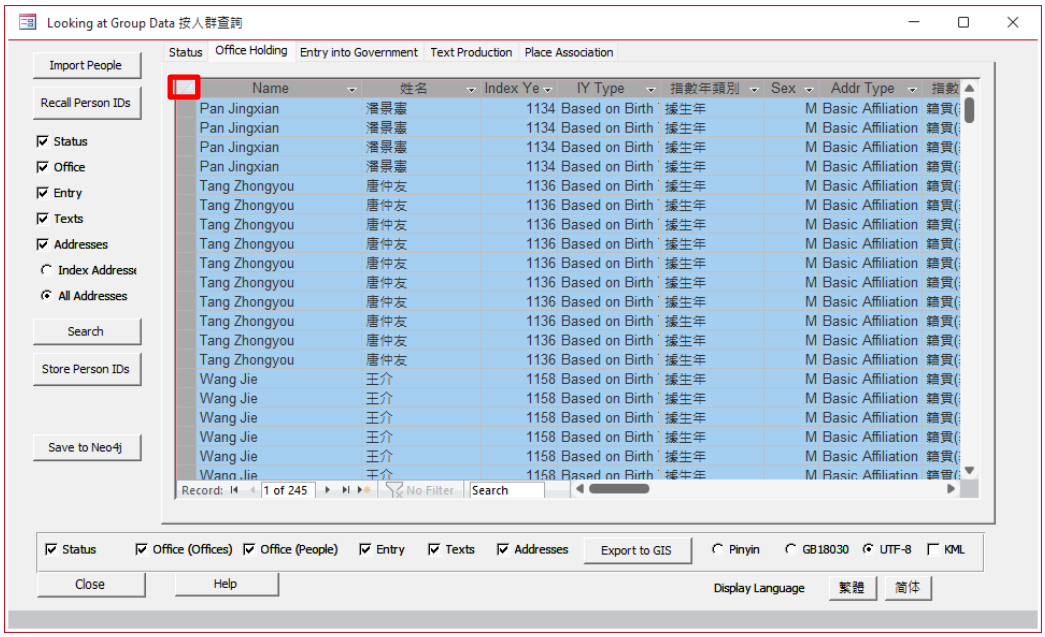


The *Entry* table shows that some *jinshi* degree holders also used other paths to enter government service in addition to the *jinshi* examination. Tang Zhongyou is an example.

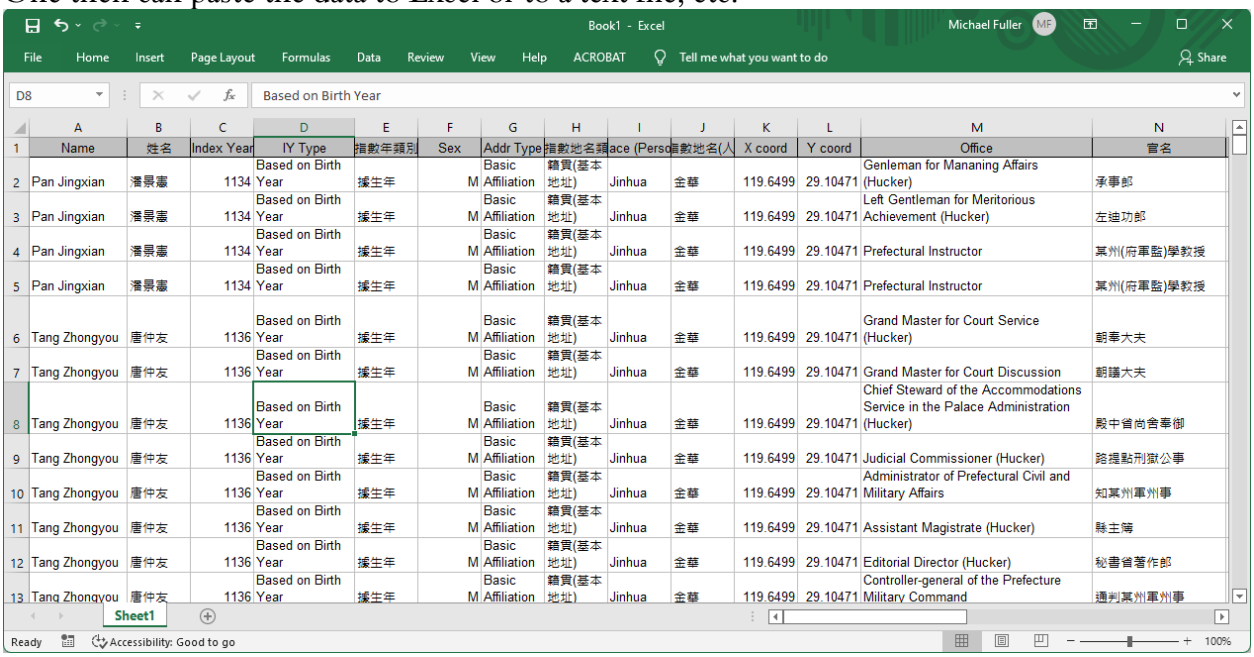
The *Status* table additionally shows the range of forms of social distinction achieved by the degree-holders. Tang Zhongyou is, again, a good example:



To save the data from the search, there are three options. One is to select the data in a table by clicking on the small square in the upper left corner of the table and copying it:

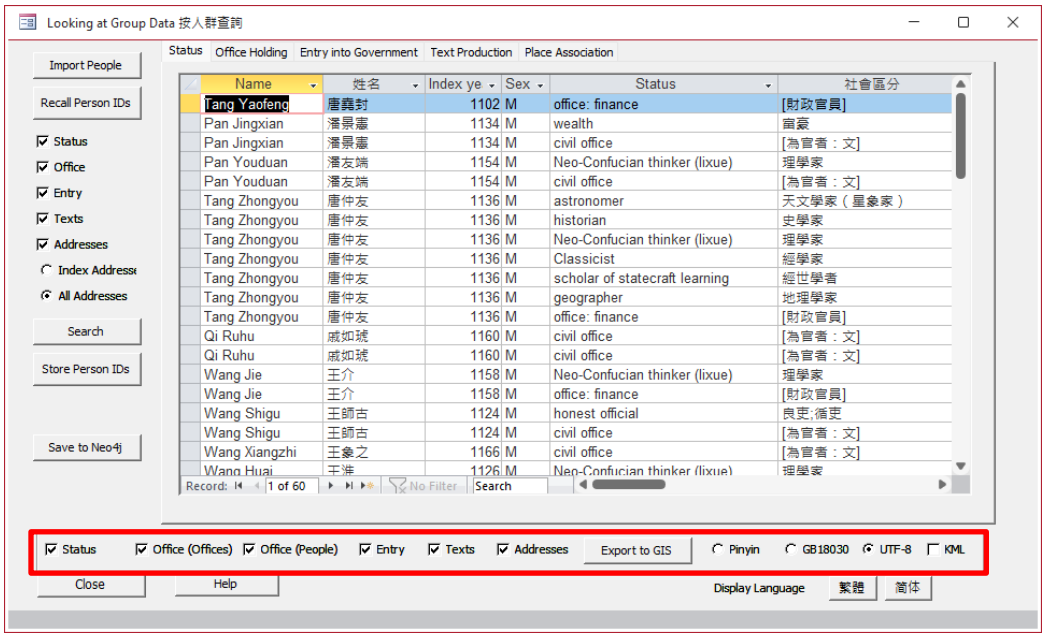


One then can paste the data to Excel or to a text file, etc.



The second way to export the data is to save it to a set of Neo4j files. The number of files produced by the form depends on the number of categories of information one has selected for export (see below).

The third way to export the data is to save it to a file that can be opened by GIS software or to a KML file:



As with exporting to Neo4j, the user selects what type of data to export and the encoding For GIS, one also chooses the file format. When the user clicks the **Export to GIS** button, the form creates a separate file for each type of information.

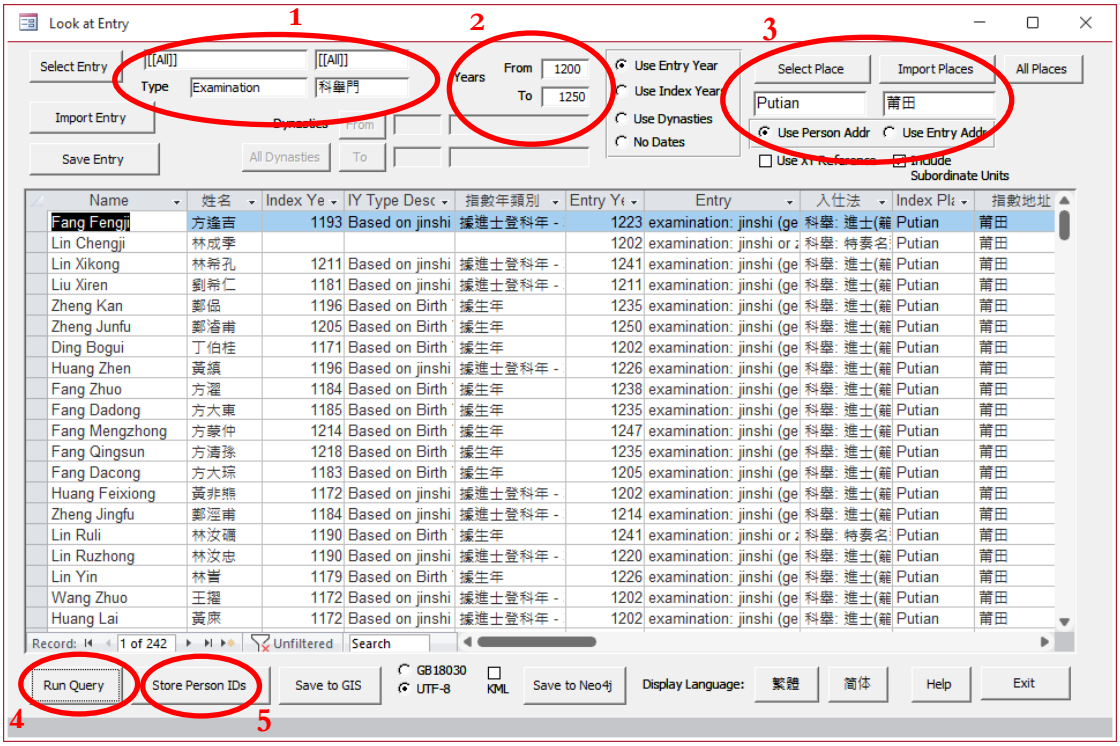
If the user has imported a list of IDs, this list can be saved by clicking on the **Store Person IDs** button.

Chapter 4: Advanced Query Techniques

The Access version of CBDB permits a variety of increasingly complex and powerful approaches to analyzing the data. The first level of advanced query simply is to use the output from one form as the input for a second search. The next step, taken when one has become relatively familiar with the data structures in CBDB, is to use the Access built-in Query Design form to create free-form queries. As one's command of the concepts of SQL (Structured Query Language) deepens, one can create ever more sophisticated queries. This chapter considers one example of using the output from CBDB forms as input for other queries and then introduces the basic ideas of SQL and illustrates them through an example that requires two steps in query design.

A. Kinship Networks for Examination Graduates in Putian, Fujian during the Song

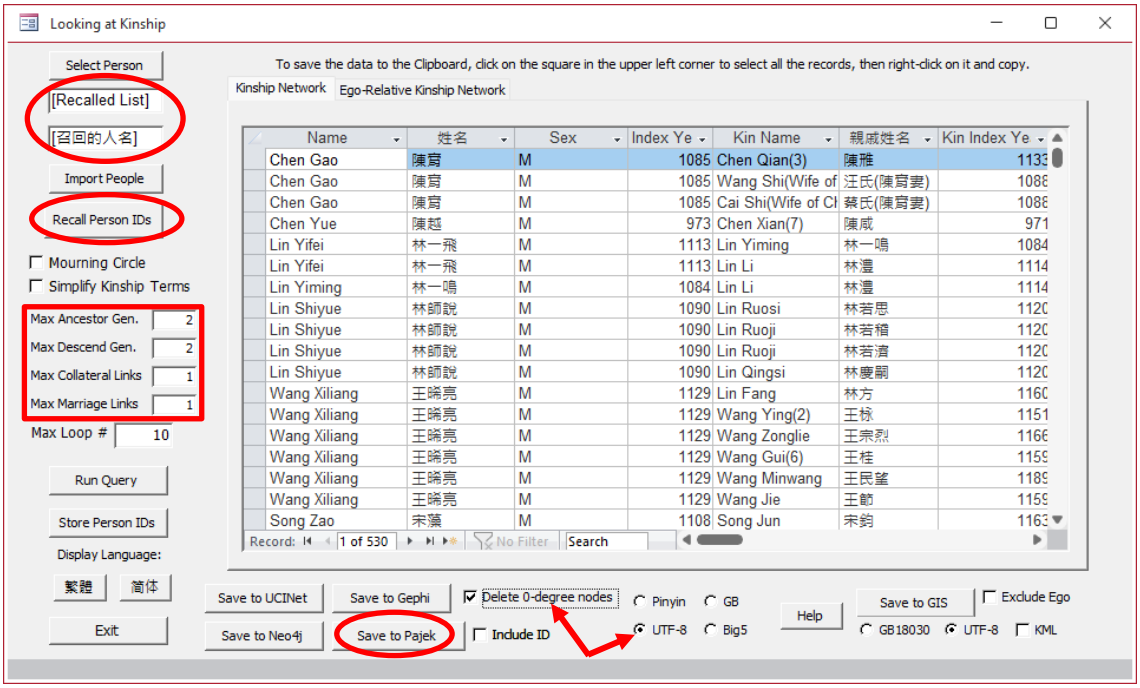
One question in the study of social history during the Song dynasty is whether local elites remained stable and controlled access to the cultural resources needed to gain entrance to official status or whether there was in fact social mobility where marginal families managed to join the elite stratum through the educational success of their sons. To explore this question, one can look at the kinship structures for those who entered government service through examinations in localities at different times during the Song and see if there is any change in organization. In our example, we consider Putian in Fujian during two periods: 1050-1100 and 1200-1250. We first use the **LookAtEntry** form:



The procedure is:

(1) Use **Select Entry** to choose all types in the category of "Examination" 科舉門.
(2) Set the range of examinations first to 1050-1100. (Here I show 1200-1250.)
(3) Use **Select Place** to choose Putian 莆田 during the Song Dynasty.
(4) Run the Query
(5) Use **Store Person IDs** to copy the IDs of the selected people into a temporary table.

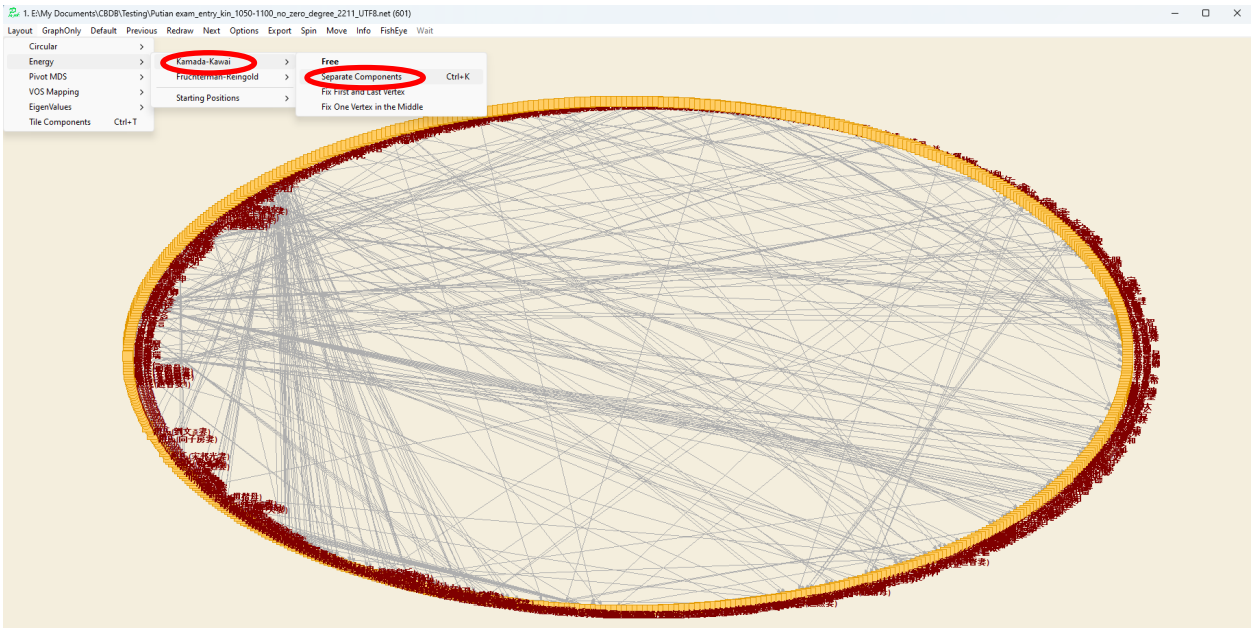
Once you have the table of the IDs of people from Putian who entered government through examination for the specified period, open the form **LookAtKinship** and have the form read the stored table of people for 1050-1100:



Here, the procedure is:

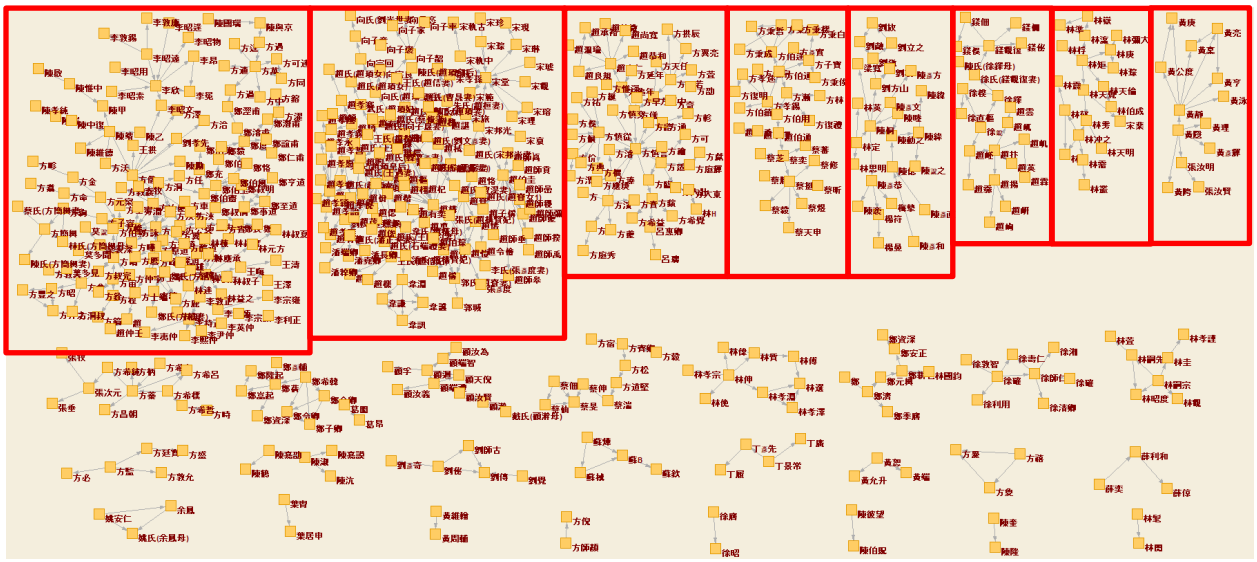
(1) Recall the list of people IDs with the **Recall Person IDs** command button at the top of the form.
(2) You will get confirmation that the table was correctly imported when you see "[Recalled List]."
(3) Set the kinship parameters to 2 up, 2 down, 1 collateral, and 1 marriage.
(4) After you run the query, save the results into a **Pajek** file that uses UTF-8 encoding. Set the output to not include 0-degree nodes (nodes with no connections to other nodes).
(5) Repeat the process for the people from 1200-1250 and create a second Pajek file.

Open your Social Network program and "Draw" the results. In this example we use **Pajek**:

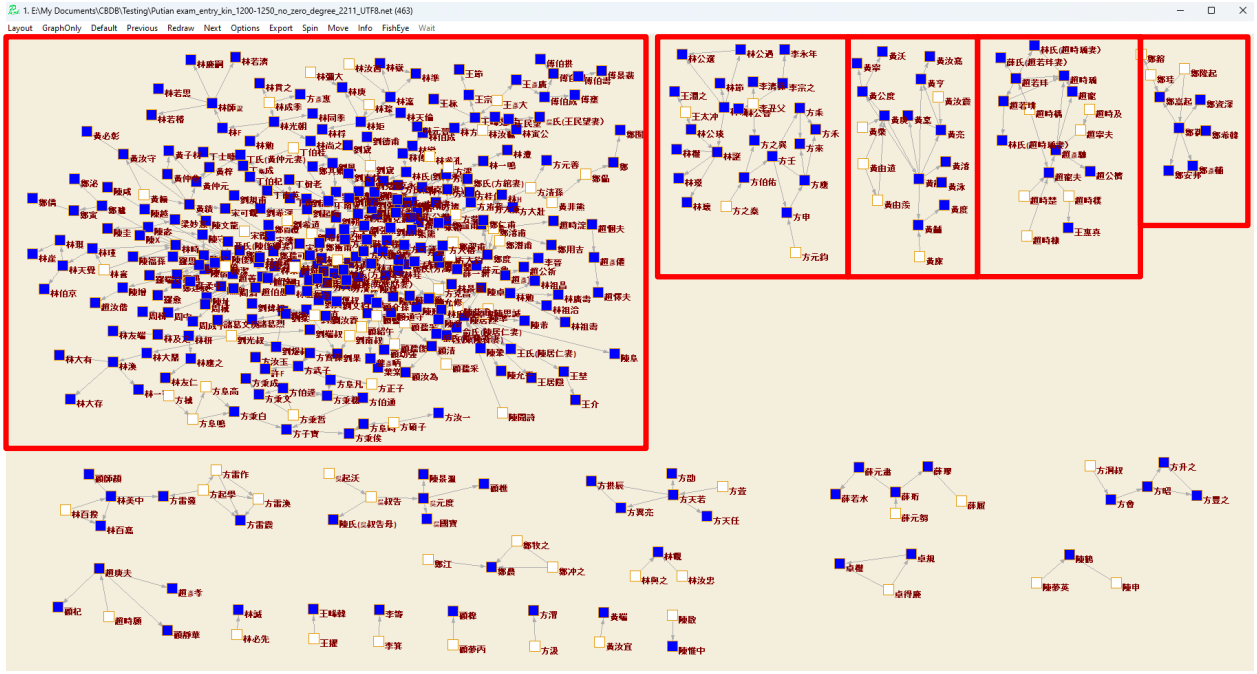


The initial layout for visualizing networks in Pajek is "Circular." A more useful layout for looking at groups of kinship networks is to select "Separate Components" in the "Kamada-Kawai" layout listings.

When one select and closely looks at the components of the kinship networks for men from Putian who passed an examination for the years 1050-1100 and 1200-1250, one gets:



Putian Examination Kinship Networks, 1050-1100



Putian Examination Kinship Networks, 1200-1250

Note that by the later period, the "principal component" (the largest component in the network) has grown to include not only a Fang 方, Chen 陳, and Lin 林 clan, but also members of Zheng 鄭 and Gu 顧 clans. The Song 宋 surname largely has disappeared. In the diagrams, the white nodes are the men who passed the examinations, and the blue squares are their kin.

B. Using the Access Query Designer

Another extremely powerful capacity built into Access is the ability to design SQL queries to look at the CBDB data from whatever angle you wish. There are a few concepts to master, but the **Query Designer** in Access allows end-users to begin to explore the data without any knowledge of **SQL** (**Structured Query Language**). As you become more familiar with queries, you can learn more about the formalisms to help you work with the data better.

In order to use the **Query Designer**, you will need some knowledge of the tables in CBDB and their relations to one another. We have simplified some of the tasks by creating a set of tables that are "denormalized," that is, where we have added descriptive fields to explain the codes in the fields that rely on IDs. For example, the table BIOG_ADDR_DATA records lists of places associated with individuals: where they were born, where their "basic affiliation" was, where they moved, where they were buried, etc. The key information for each record, however, is a set of three codes: a person ID, an address ID, and an address type ID. We have created a second table, **ZZZ_BIOG_ADDR_DATA**, that takes information from other tables (BIOG_MAIN, ADDR_CODES, BIOG_ADDR_CODES) to give the name of the person, the name of the place, and the description of the type of address, along with other

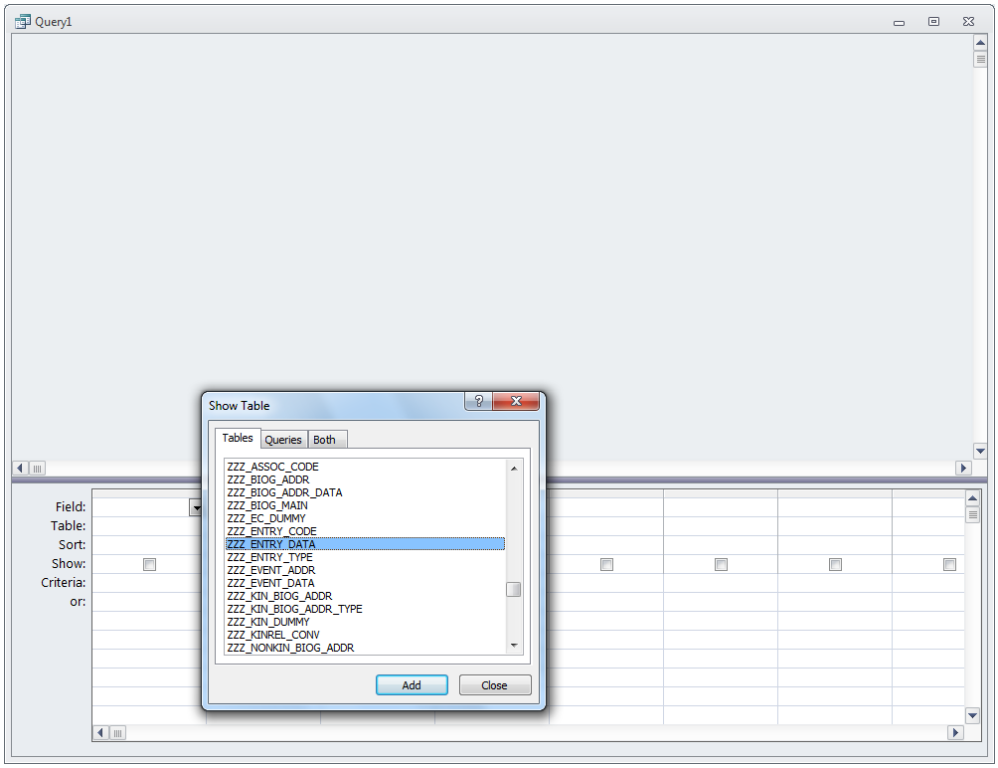
useful data. Using these tables with descriptions and codes simplifies the task of building a useful query. The tables are:

1. ZZZ_ALT_NAME_DATA
   (fills in alternate name type)
2. ZZZ_BIOG_ADDR_DATA
   (fills in address and address type)
3. ZZZ_BIOG_MAIN
   (fills in nianhao, ethnicity)
4. ZZZ_ENTRY_DATA
   (fills in entry type)
5. ZZZ_KIN_BIOG_ADDR
   (this is the table for kinship, but it also provides the main entry for biographical address)
6. ZZZ_NONKIN_BIOG_ADDR
   (this is the table for associations, but it also provides the main entry for biographical address)
7. ZZZ_POSTED_TO_ADDR_DATA
   (fill in address information)
8. ZZZ_POSTED_TO_OFFICE_DATA
   (fills in office information)
9. ZZZ_TEXT_DATA
   (fills in text data)

I. An Example:
The mode of entry into government of near kin of the successful *jinshi* degree candidates of the 1148 examination

How might one use an SQL query to determine how many of the people who passed the *jinshi* examination in 1148 (for which we have a complete list) had close relatives who had entered government service?

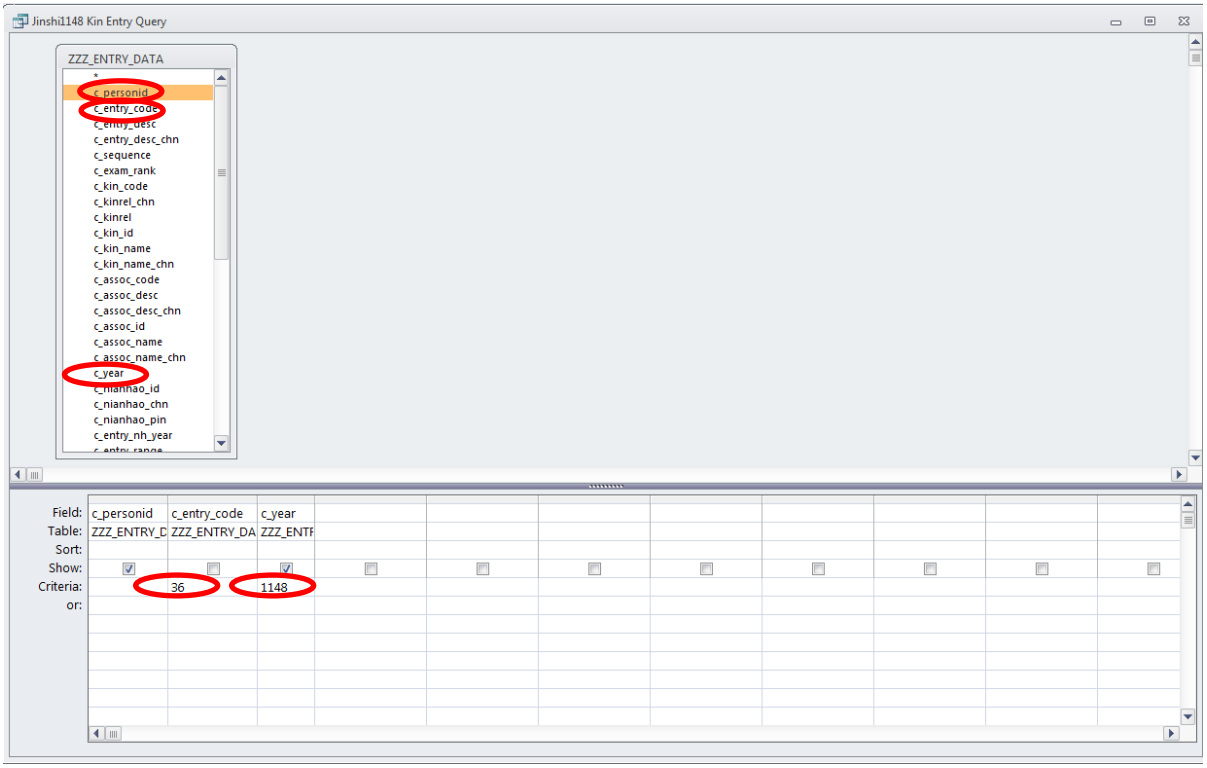
1. In the **Create** menu (next to the **Home** tab at the top of the main screen) , Select **Query Design**:



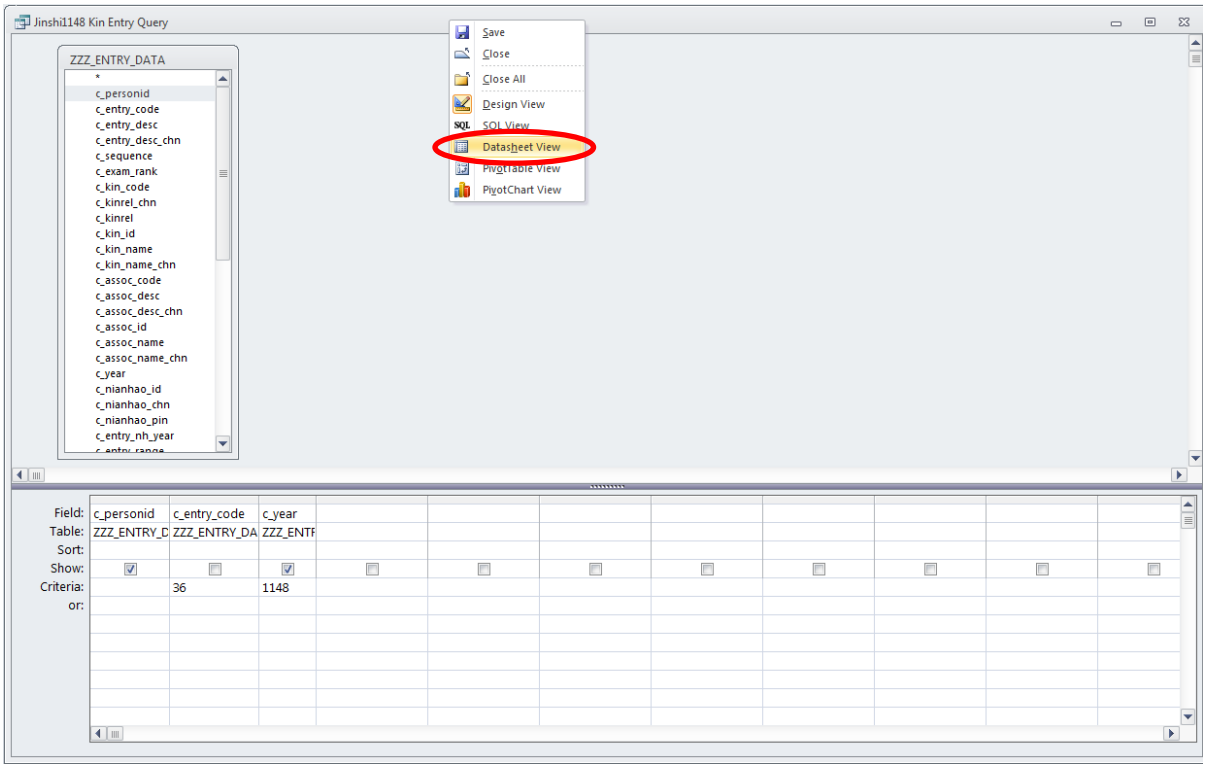
In the "Show Table" window, select ZZZ_ENTRY_DATA and click **Add**

2. Double-click on **c_personid**, **c_entry_code**, and **c_year** to add them to the query. Unclick the "Show" check-box for **c_entry_code** so that you can next specify a value but have the field not appear in the results of the query, since in every record, the value of the field will be the same.

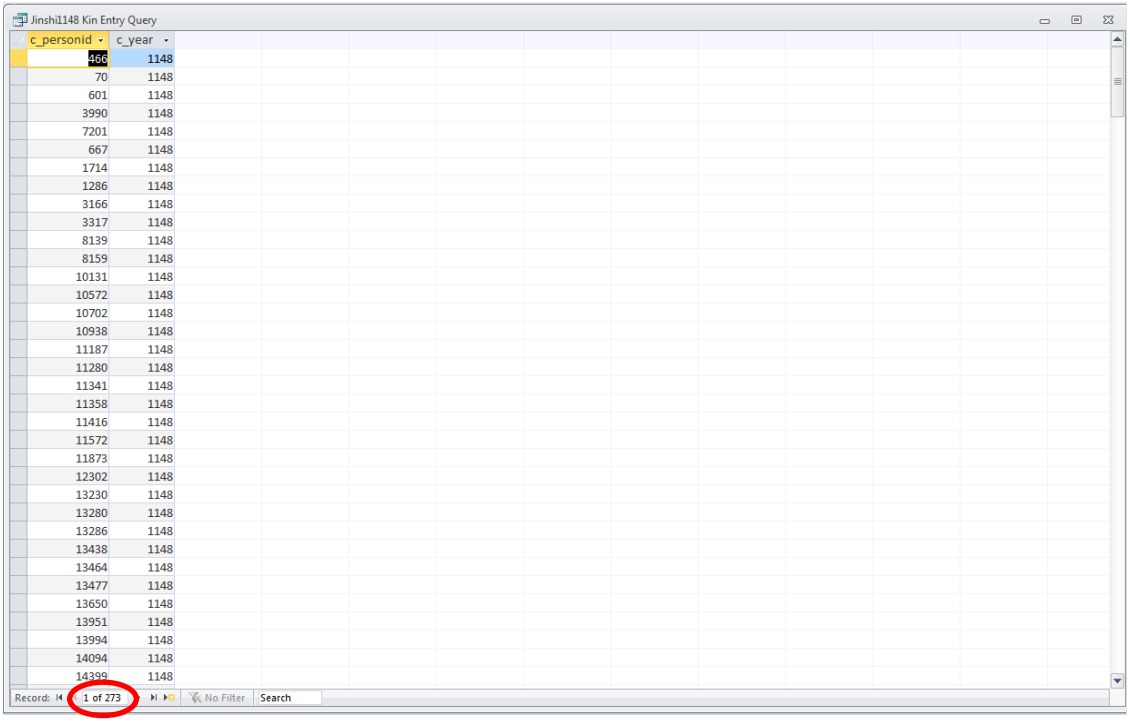
3. Then in "Criteria" specify the value 36 for c_entry_code (*jinshi*), and the year 1148.



4. Check the results by right-clicking on the top border of the Query form and switching to **Datasheet View**:



5. There are 273 records. (Please note that as CBDB adds data, these results will change.)



6. Now add the kinship table ZZZ_KIN_BIOG_ADDR by clicking on **Show Table** along the Query Tools menu at the top of the screen and select ZZZ_KIN_BIOG_ADDR from the "Show Table" window:

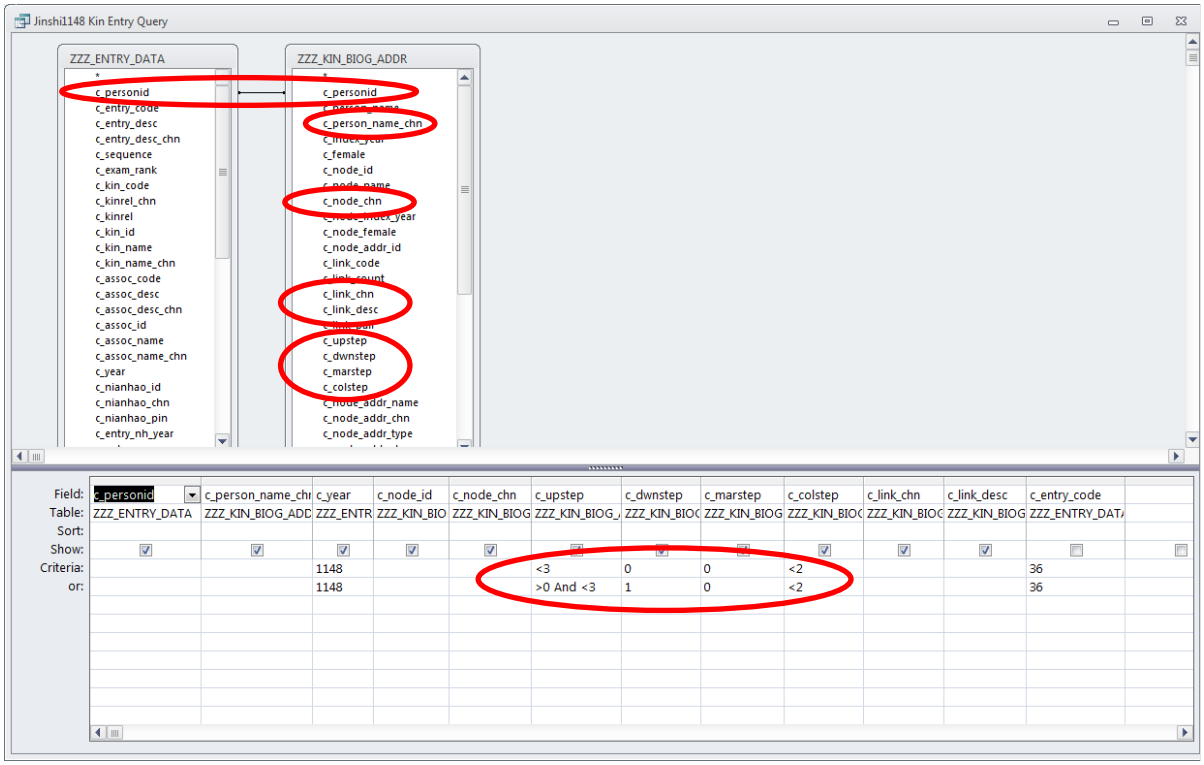
a. Create a **link** between the two tables by clicking on c_personid in ZZZ_ENTRY_DATA and dragging it to the c_personid in ZZZ_KIN_BIOG_ADDR. The query builder may ask you to confirm that you want to select only those pairs of records from the two tables which share the same person IDs.

b. From the kinship table, add the following fields:
c_person_name_chn (the name of the person identified by c_personid)
c_node_id (the ID of the relative)
c_node_chn (the name of the relative)
c_upstep (the number of generations up in the kinship relation)
c_dwnstep (the number of generations down in the kinship relation)
c_marstep (the number of marriage relations involved in the kinship relation)
c_colstep (the number of brother/sister relations involved in the kinship relation)
c_link_desc (the English description of the kinship relation)
c_link_chn (the Chinese description of the kinship relation)

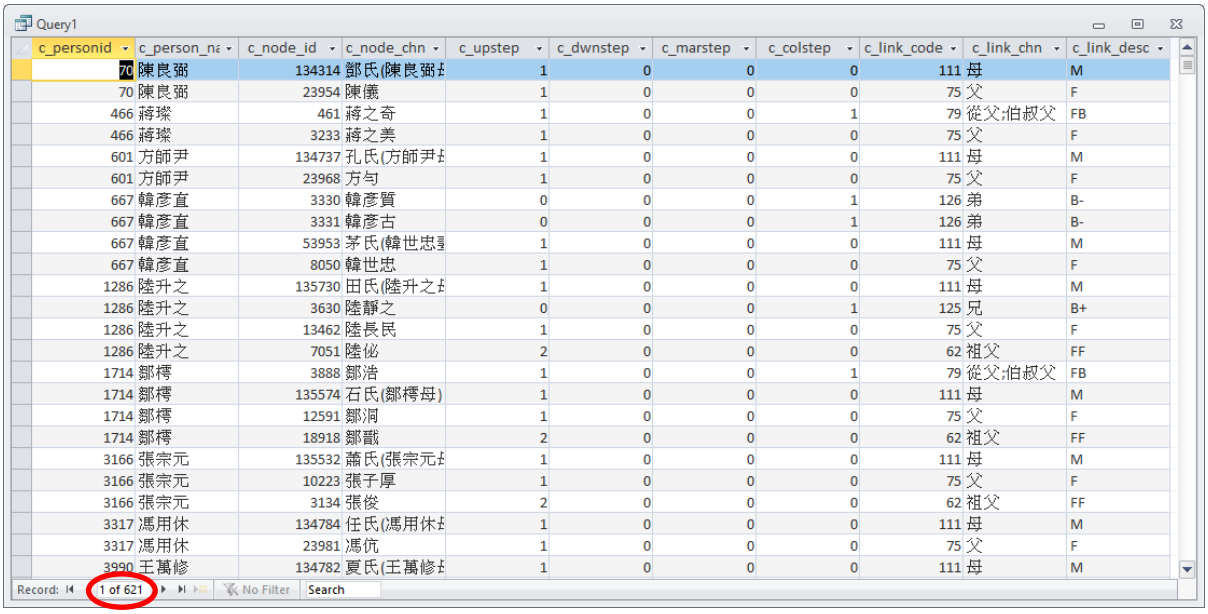
c. Set the limit for generations up (c_upstep) to 2 (i.e., FF, FFB, etc.)
Set the limit for generations down (c_dwnstep) to 0 (i.e, we want to look only at ancestors)
Set the limit for affines (c_marstep) to 0

Set the limit for brother/sister (c_colstep) to at most 1

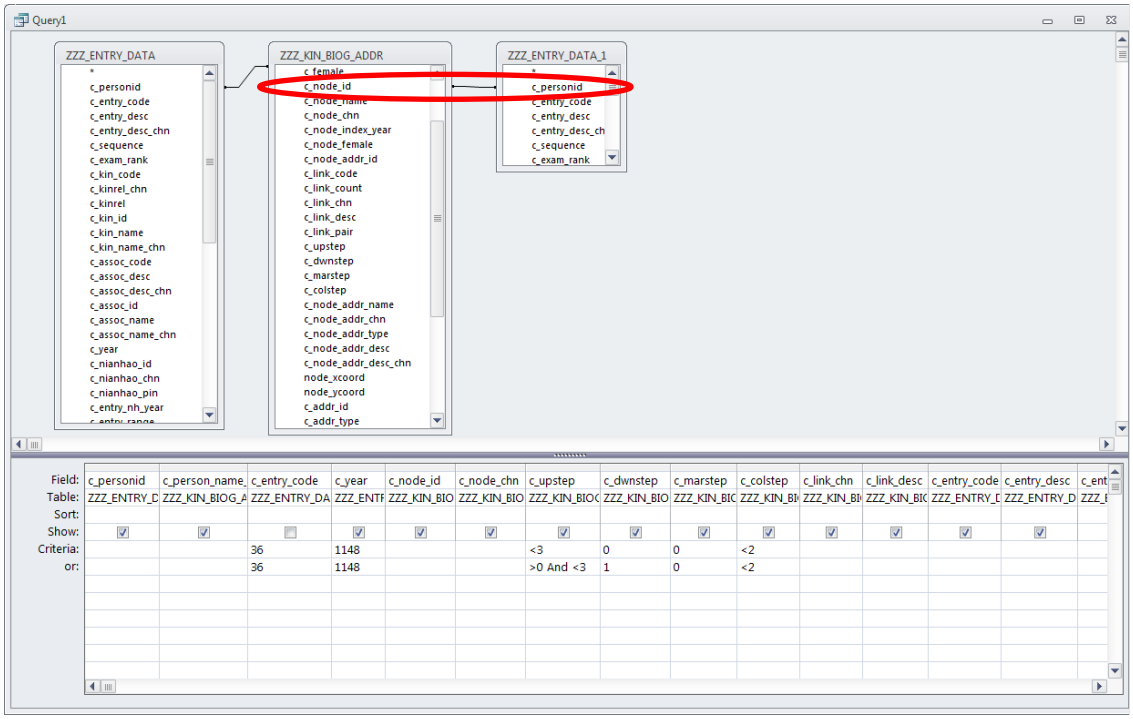
d. Repeat this process, but allow cousins (i.e. FBS or FFBS: 1 down step, at least 1 up)



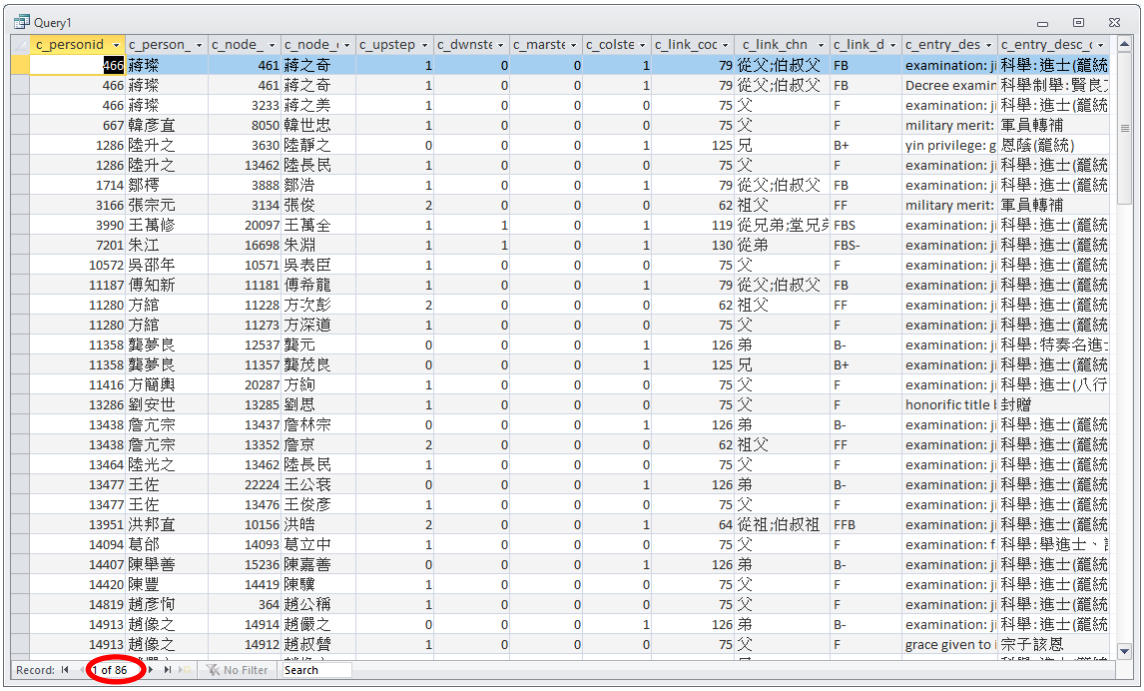
6. Check the results: There are 621 relatives that meet the criteria



7. Now add a ***second*** version of the ZZZ_ENTRY_DATA table and link that table to the ZZZ_KIN_BIOG_ADDR table by making c_node_id = c_personid:

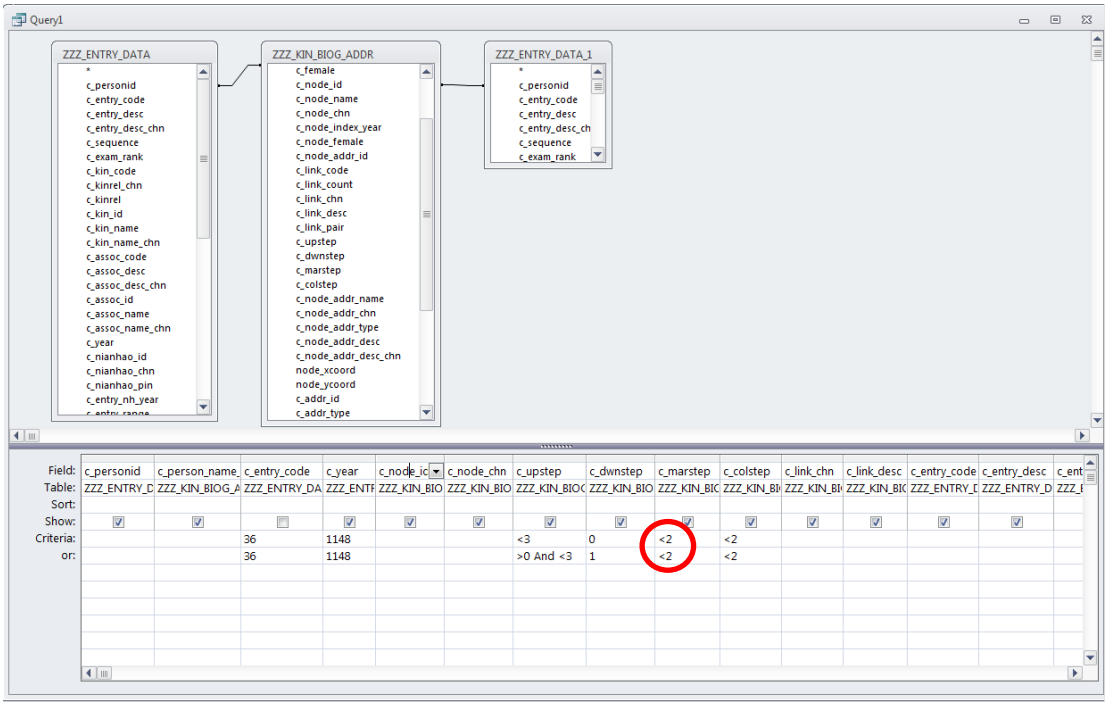


8. Add the two fields c_entry_desc and c_entry_desc_chn from ZZZ_ENTRY_DATA_1 (to get the mode of entry of the kin) and check the results:

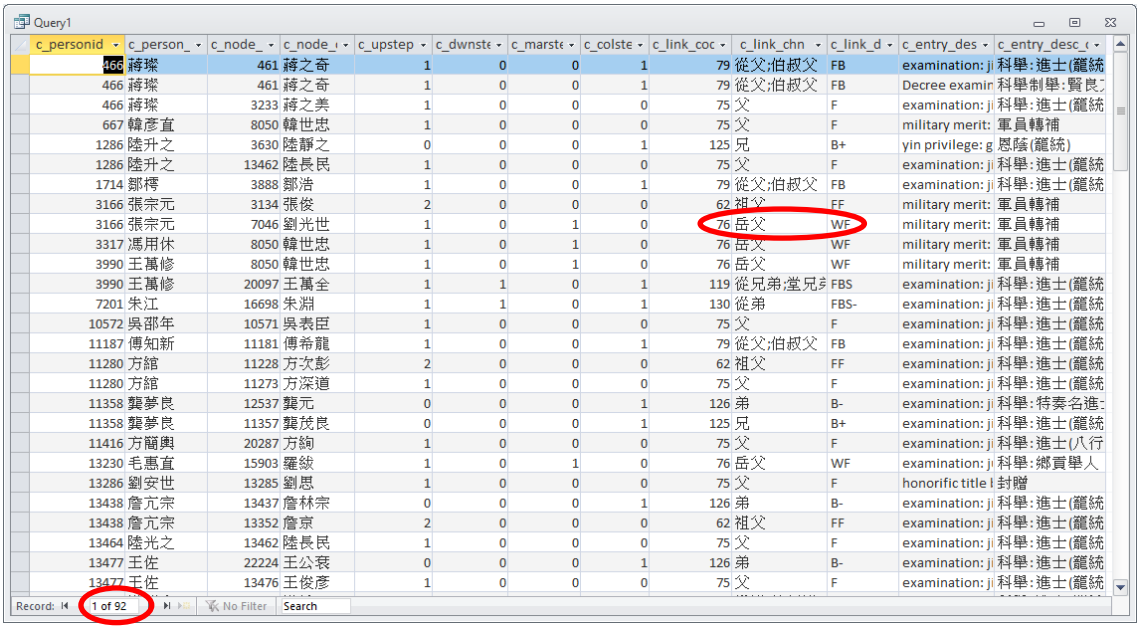


86 kin from the 273 initial degree recipients have data on how they entered officialdom

9. Simply adding a 1 to the c_marstep will allow one to look at affinal relations as well. Using the criterion "<2" means that a c_marstep of either 0 or 1 in the record will be acceptable:



This produces 6 additional records for a total of 92.

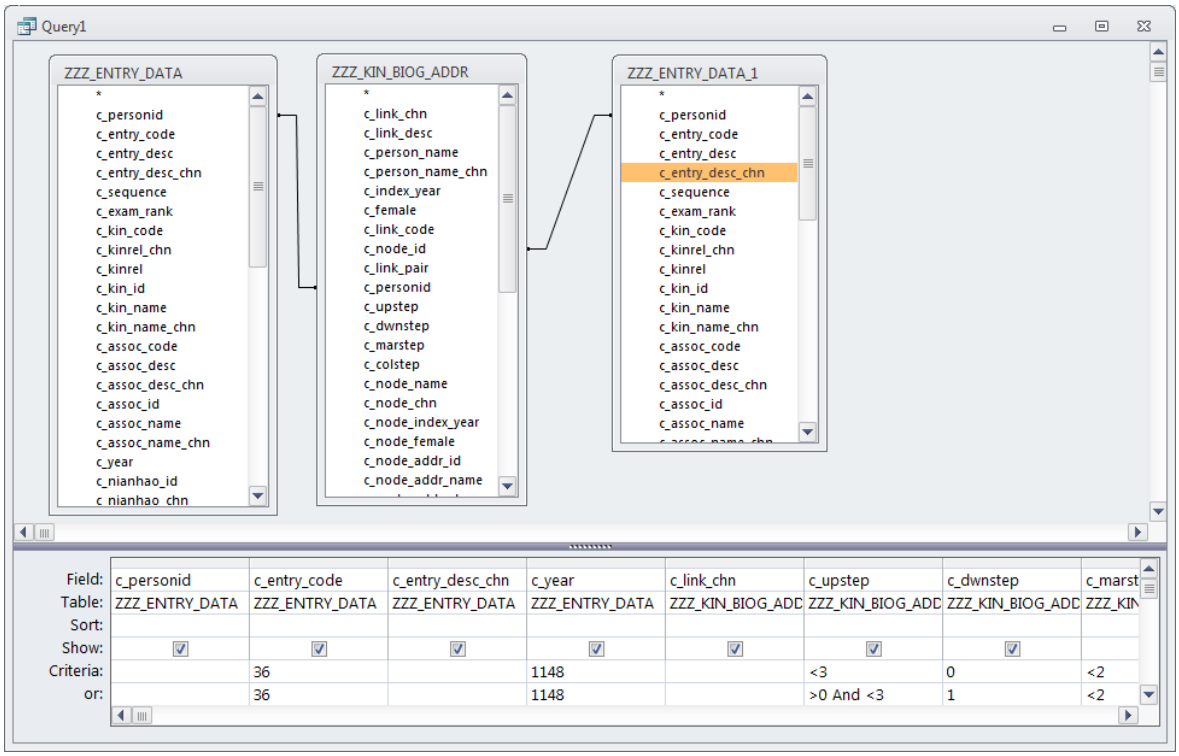


II. Some Useful Additional Procedures for Queries

A. Null Information can be Useful

In the above query, we dealt only with those relatives for whom information about their mode of entering government service was known. Suppose, however, that we wanted a list of *all* the relatives *as well as* the available information about their mode of entering service. Such a list helps clarify the percentage for whom we have data.

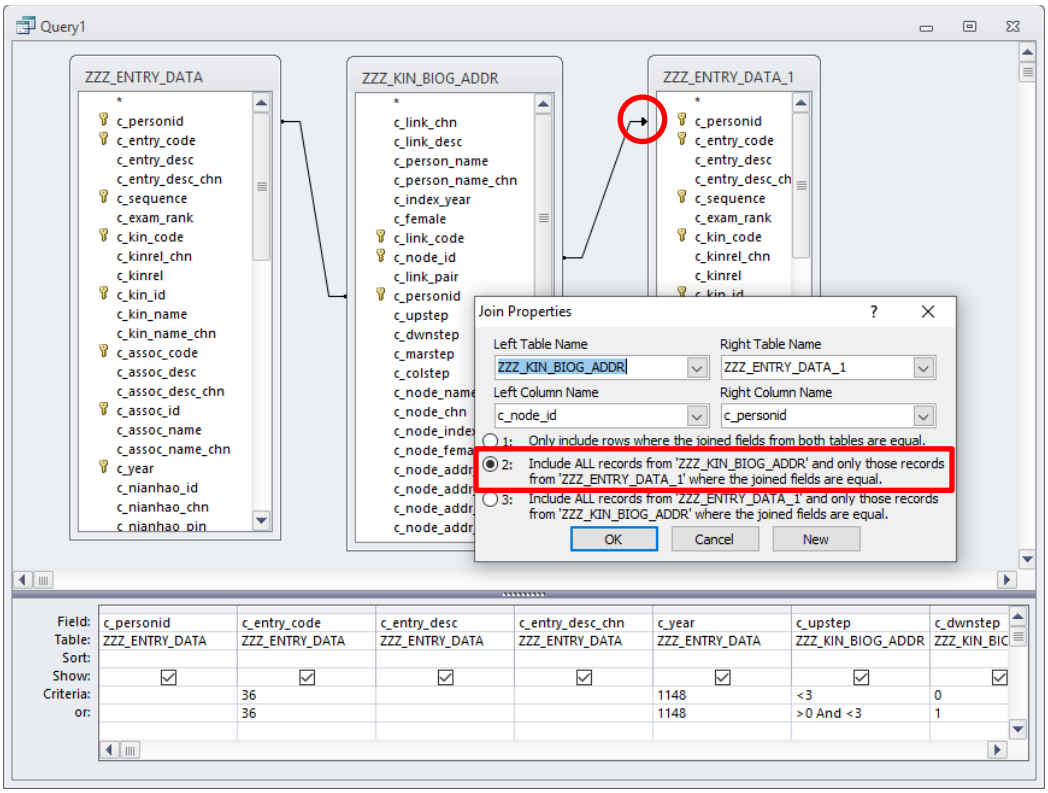
Our initial design looked like:



We need to change the way Access selects its records. To do this we need to modify the **link between the entry data for the kin and the kin** themselves, which we created by equating c_node_id (i.e., the ID for the kin) in ZZZ_KIN_BIOG_ADDR with c_personid in ZZZ_ENTRY_DATA_1, the second copy of ZZZ_ENTRY_DATA you added to the query.

ZZZ_KIN_BIOG_ADDR. c_node_id = ZZZ_ENTRY_DATA_1. c_personid

To modify that link, double-click on the line connecting c_node_id and c_personid. This will open a dialog box:



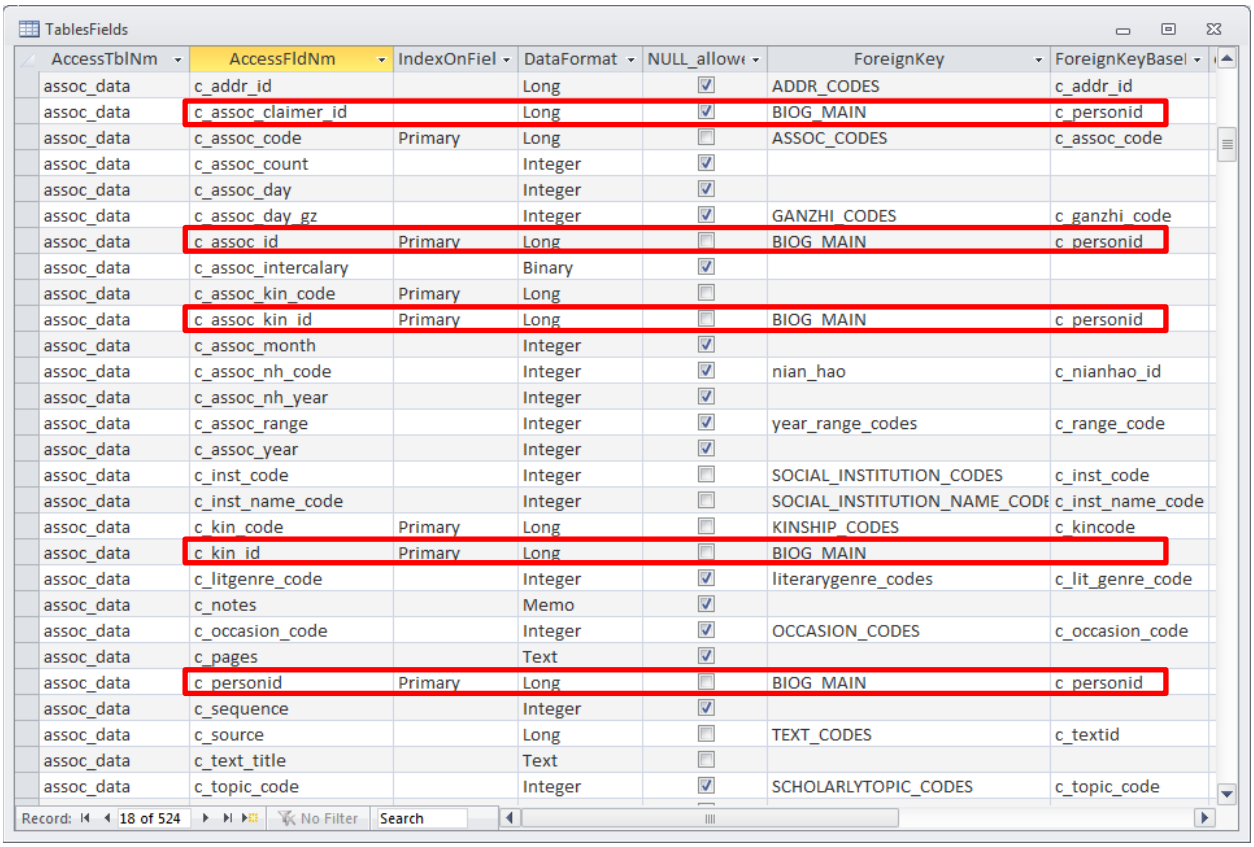
Select option 2 and click OK. Note the arrow pointing to c_personid. This arrow indicates a "left join" in the language of SQL. This **left join** includes all the records from ZZZ_KIN_BIOG_ADDR (the left table) that match the other query criteria as well as the fields from ZZZ_ENTRY_DATA_1 (the right table) where there is a match in kin IDs and entry IDs. (Left and Right are determined by the order in which the tables are linked.)

When we execute the query, we get records for all the initial 621 kin.

B. The TablesFields *Table*

For getting information on additional people involved in various types of social interactions, you need to know which fields in a table refer to IDs for people. When in doubt, you can open the TablesFields table from the list of tables on the left of the main Access interface and look for the fields in the table you want to explore. Those that have "BIOG_MAIN" in the "foreign key" column and "c_personid" in the ForeignKeyBase column refer to people.[2] For example, in ASSOC_DATA, we have:

[2] In a normalized database, "foreign key" simply refers to those fields that use the IDs defined (as primary keys) in other tables.



TablesFields

AccessTblNm	AccessFldNm	IndexOnFiel	DataFormat	NULL_allow	ForeignKey	ForeignKeyBasel
assoc_data	c_addr_id		Long	☑	ADDR_CODES	c_addr_id
assoc_data	c_assoc_claimer_id		Long	☑	BIOG_MAIN	c_personid
assoc_data	c_assoc_code	Primary	Long	☐	ASSOC_CODES	c_assoc_code
assoc_data	c_assoc_count		Integer	☑		
assoc_data	c_assoc_day		Integer	☑		
assoc_data	c_assoc_day_gz		Integer	☑	GANZHI_CODES	c_ganzhi_code
assoc_data	c_assoc_id	Primary	Long	☐	BIOG_MAIN	c_personid
assoc_data	c_assoc_intercalary		Binary	☑		
assoc_data	c_assoc_kin_code	Primary	Long	☐		
assoc_data	c_assoc_kin_id	Primary	Long	☐	BIOG_MAIN	c_personid
assoc_data	c_assoc_month		Integer	☑		
assoc_data	c_assoc_nh_code		Integer	☑	nian_hao	c_nianhao_id
assoc_data	c_assoc_nh_year		Integer	☑		
assoc_data	c_assoc_range		Integer	☑	year_range_codes	c_range_code
assoc_data	c_assoc_year		Integer	☑		
assoc_data	c_inst_code		Integer	☐	SOCIAL_INSTITUTION_CODES	c_inst_code
assoc_data	c_inst_name_code		Integer	☐	SOCIAL_INSTITUTION_NAME_CODE	c_inst_name_code
assoc_data	c_kin_code	Primary	Long	☐	KINSHIP_CODES	c_kincode
assoc_data	c_kin_id	Primary	Long	☐	BIOG_MAIN	
assoc_data	c_litgenre_code		Integer	☑	literarygenre_codes	c_lit_genre_code
assoc_data	c_notes		Memo	☑		
assoc_data	c_occasion_code		Integer	☑	OCCASION_CODES	c_occasion_code
assoc_data	c_pages		Text	☑		
assoc_data	c_personid	Primary	Long	☐	BIOG_MAIN	c_personid
assoc_data	c_sequence		Integer	☑		
assoc_data	c_source		Long	☐	TEXT_CODES	c_textid
assoc_data	c_text_title		Text	☐		
assoc_data	c_topic_code		Integer	☑	SCHOLARLYTOPIC_CODES	c_topic_code

Record: 18 of 524 No Filter Search

Among all these, the following are IDs of people:

c_assoc_claimer_id (the ID of the person claiming the existence of the association)
c_assoc_id (the ID of the associate)
c_assoc_kin_id (the ID of the kin of the associate through who the association exists, if any)
c_kin_id (the ID of the kin of the main person in the record through who the association exists, if any)
c_personid (the person whom the record is about)

Appendices

Appendix A: Installing the MS Access Files

In order to keep the database files within the two gigabyte limit for Microsoft Access files, CBDB is divided into four files: three "Base" files with the tables of data, and a "User" file with the user interface. The User file draws on the tables in the Base files as "linked tables." When you install the CBDB files, the Access program will automatically create the links between the User and Base files that you have installed in a shared directory. If that link fails or you need to recreate the link when you download new data files, the Navigation pane provides a way to recreate the links.

To install the MS Access database

1. Create a folder into which to extract the four files that you have downloaded from the CBDB website. Extract the files.

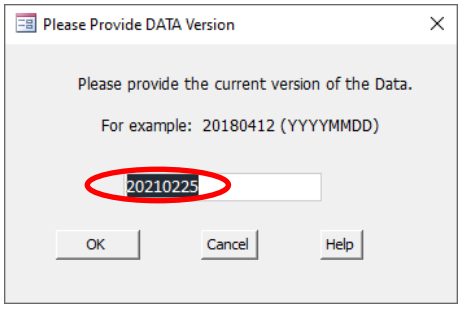
2. Double-click on the User file to open it in Microsoft Access. You will see:

Note the arrows next to most of the tables in the list on the left side of the screen. The arrow indicates that the table is a *linked* table from the Base files.

3. Double-click on any linked table, and if the table is successfully linked, it will open. If the link is broken, you will see the message:

4. If you get an error message, double-check that the three data files are in the same directory as the User file. If they are, write down the name of one of the data files, e.g. CBDB_20210225_DATA1.mdb. The date "20210225" (in YYYYMMDD format) gives the date of the data release.

5. Next, click on the "Relink Tables" command button in the Navigation Pane. This will open a form that will ask for the date of the data release:



Write the date into the form and click "OK." The form will relink the tables.

6. The User file is now ready to use.

Appendix B: Updating the Visual Basic Environment (if necessary)

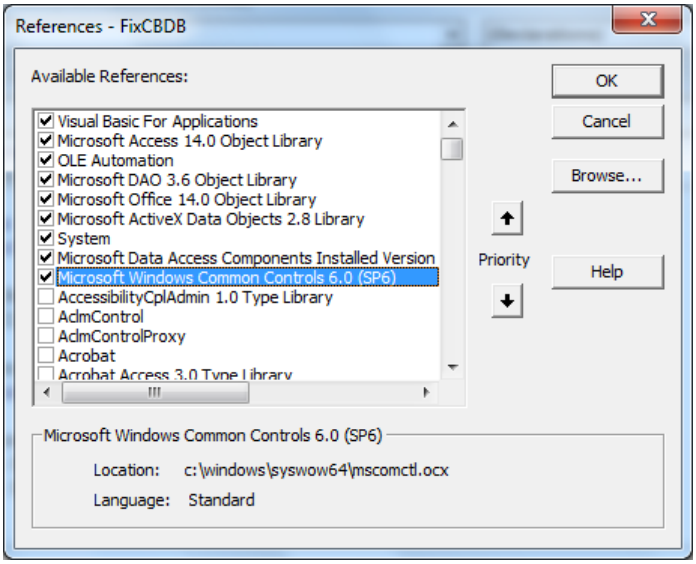
Adding References

CBDB uses a variety of Visual Basic resources that are not part of the default MS Access Visual Basic environment. If your effort to run a routine produces an error about an undefined VB object, you may need to double-check the "References" used by Visual Basic.

To do this:

1. Under "Database Tools" in the main Access window, select Visual Basic. This will launch the Visual Basic editor.

2. In the VB editor, click on the menu item "Tools" and then "References…" You will see something like:



3. If you do not see the same references checked, please scroll down the list and make your "References" list match this one. You may encounter a complaint about duplicated resources. In that case, you will see that your initial checked list has components that are *not* on this list. Uncheck them and try again.

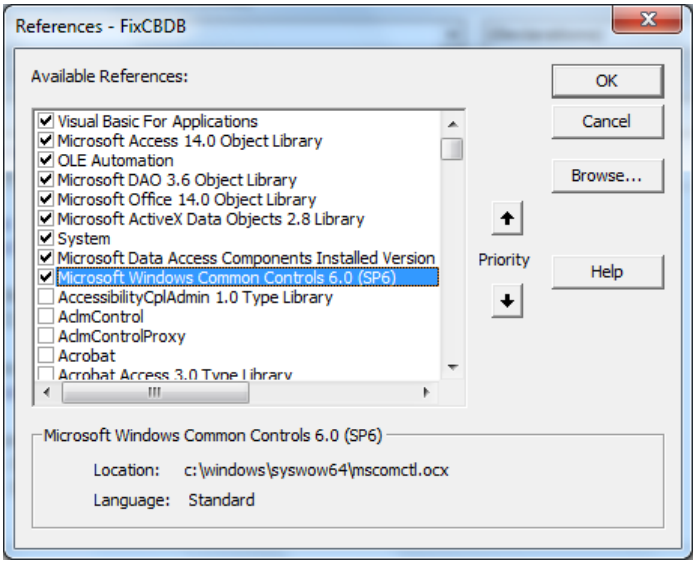
Adding TreeView to Visual Basic

If your copy of Access gives you an error when you try to select an office in LookAtOffices or select an association in LookAtAssociations, this is because you do not have a file (MSCOMTL.ocx) added to your Visual Basic environment.

To Fix:

1. Under "Database Tools" in the main Access window, select Visual Basic. This will launch the Visual Basic editor.

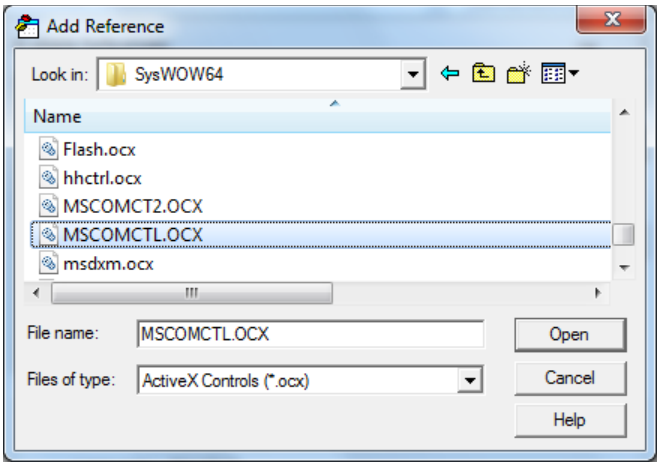
2. In the VB editor, click on the menu item "Tools" and then "References…" You will see something like:



3. If you see "Microsoft Windows Common Controls 6.0 (SP6)," then your problem may something else. Please uncheck the check box, close the window, exit the VB editor, close Access, then reopen Access, return to the editor, and go to step 5 below. If this does not let TreeView work, please let us know.

4. If you do NOT see the line, please scroll down the list. If you find the line, click on it to check the box. Click OK.

5. If you do not find the Common Controls 6.0 on the list, you will need to add it.
   a. Click on "Browse…"
   b. If you are using **Windows 7**, go to the subdirectory SysWOW64 in the Windows directory.
      If you are using **Windows XP**, go to the subdirectory System32.
   c. Change the "Files of type" to: "ActiveX Controls (*.ocx)"
   d. You should see:



e. Click on "MSCOMCTL.OCX"
f. Click on "Open"
g. Make sure the check-box for Common Controls 6.0 is checked in the References window, then click "OK."

6. If you do not find MSCOMCTL.OCX in SysWOW64, you will need to add it.
   a. The CBDBPatch.rar file that you downloaded from the CBDB website contains a copy of the OCX file as well as these instructions.
   b. Copy the file MSCOMCTL.OCX to C:\WINDOWS\SysWOW64
   c. Now you will need to register the file:
      1. Click on the Windows "Start" Button.
      2. Select "All Programs" and then "Accessories"
      3. Right-click on "Command Prompt" and click on "Run as Administrator."
      4. Click "yes" when the system asks you if it can proceed.
      5. In the Command Prompt window, type:
         REGSVR32 C:\Windows\sysWOW64\MSCOMCTL.OCX
      6. Hit "Enter" to run the program.
      7. Close the Command Prompt window.
   d. Now perform the steps listed in (1) - (5) on the first page.

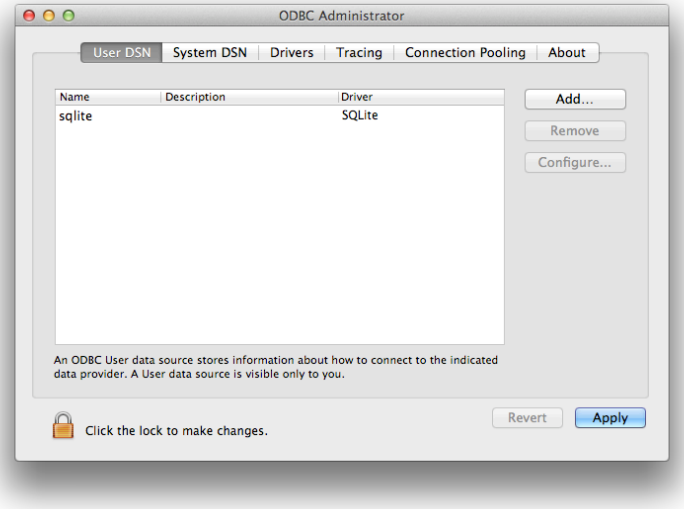
7. To exit the **Visual Basic Editor**, click on the menu item "File" and then on "Close and Return to Microsoft Access."

Appendix C: Installing the SQLite CBDB database on a Macintosh

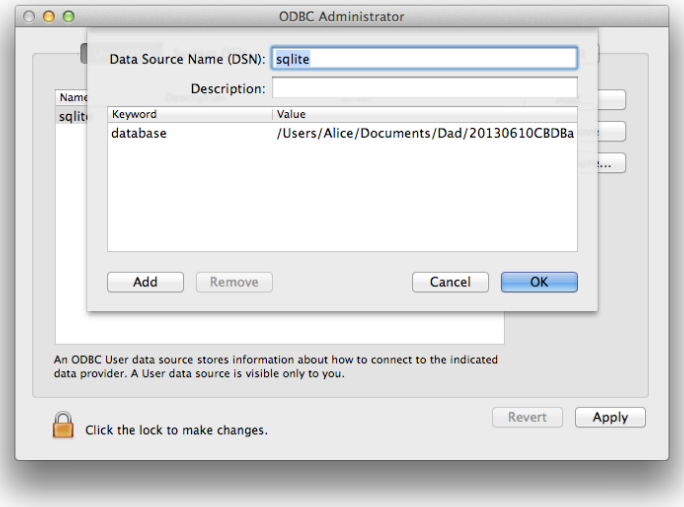
For Apple users (or Linux users, who probably do not need these instructions), there is a stand-alone version of the CBDB database using the SQLite format.

For any database file to be used in a Macintosh system, the operating system needs a connector between the file and the standard ODBC (Open Database Connectivity) interface. In order to make this connection, you will need the Mac ODBC Administrator and the ODBC driver for SQLite. (You may need to download these from the web, or you may decide to leave these steps to your information technology specialist, if you have access to one. The ODBC driver for SQLite can be downloaded from http://www.ch-werner.de/sqliteodbc/).

1. Install the Macintosh ODBC Administrator and the driver for SQLite.
2. In "Finder," go to Utilities and open the ODBC Administrator.
3. Go to User DSN and add "CBDBFull" as an SQLite database:



4. Click on "Configure" to set up the connector:



5. Add the keyword “database” and use the full path for the database file as the “value.”
6. Click on OK. The window will close. Then click on Apply.
7. The SQLite version of CBDB should be ready to use with OpenOffice or whatever software interface you prefer.

Appendix D: Kinship Simplification Table

Simplifications that are applied in all Searches:

Kinship Term	Male/Female/Both	Simplified Kinship term
BB	B	B
BZ	B	Z
DB	B	S
DZ	B	D
SB	B	S
ZB	B	B
ZZ	B	Z

Simplifications that are applied when the user requests algorithmic

Kinship Term	Male/Female/Both	Simplified Kinship term
BDF	B	B
BDM	B	BW
BF	B	F
BM	B	M
BSF	B	B
BSM	B	BW
DDF	B	DH
DDM	B	D
DF	F	H
DF	M	-
DM	F	-
DM	M	W
DSF	B	DH
DSM	B	D
FBF	B	FF
FD	F	Z
FD	M	Z
FDF	B	F
FDM	B	M
FFD	B	FZ
FFS	B	FB
FS	F	B
FS	M	B
FSF	B	F
FSM	B	M
FW	B	M
FZF	B	FF
HD	F	D
HDF	F	H

HDM	F	-
HS	F	S
HSF	F	H
HSM	F	-
HW	F	-
MD	F	Z
MD	M	Z
MDF	B	F
MDM	B	M
MH	B	F
MS	F	B
MS	M	B
MSF	B	F
MSM	B	M
SDF	B	S
SDM	B	SW
SF	F	H
SF	M	-
SM	F	-
SM	M	W
SSF	B	S
SSM	B	SW
SZ	B	D
WD	M	D
WDF	M	-
WDM	M	W
WH	M	-
WS	M	S
WSF	M	-
WSM	M	W
ZDF	B	ZH
ZDM	B	Z
ZF	B	F
ZM	B	M
ZSF	B	ZH
ZSM	B	Z

Appendix E. Change Log for CBDB

The Access version of CBDB was significantly reorganized with the **AW** release: the data and the querying interface are now completely separate and can be updated independently. For all more recent releases, changes in the data and in the interface are reported separately.

Versions:

CBDB SQL Server Version 1

Announcement

We are delighted to announce the introduction of CBDB_SS, a version of CBDB that is identical to Version BC in its appearance and functionality, except that the data for CBDB is now stored on a SQL Server Express database platform. SQL Server does not have the limitation in file size in MS Access that required us to split the CBDB data into three separate Access files and will allow us to continue to use the PC-based version of CBDB for the foreseeable future.

The code for CBDB has been rewritten so that all queries are performed by the SQL Server database and the results then become available to the CBDB user interface. There are only a few insignificant (we hope) differences in behavior, and in general, queries should run more quickly on the SQL Server version.

Users who create their own queries with MS Access Query Builder may encounter a few problems created by inconsistencies between the versions of SQL in Access and in SQL Server. Advanced users should consider downloading and installing MS SQL Server Management Studio, which gives on direct access to the SQL Server CBDB database.

Build 20211110

Design Change

In LookAtOffice, selecting an office now provides more information about the dynasty of the selected office.

Build 20210826

Design Change

In the Browser for individuals, using pinyin now has three options:
(1) lower case: it looks for the string anywhere in the name;
(2) upper case: it looks for the string only at the beginnings of words ("Hao" will select both Zhang Hao and Hao Jing;
(3) string beginning with "!:" it looks only at the beginning of the surname (just "Hao Jing")

Build 20210609

Bug Fixes:

1. Added additional fields to ZZ_SCRATCH_BIOG_MAIN (these had been deleted in recreating the SQL Server database from the Access database. I have added the fields to the table in the SupplementalTablesSQL_Server.mdb file in the Migration subfolder.
2. Removed all references to ZZZ_ALL_BIOG_ADDR, which has been replaced in functionality by ZZZ_BIOG_MAIN
3. Added the field c_kin_rel_count to ZZ_SCRATCH_KINNET_EDGE.
4. Because Access does not allow sorting of SQL Server tables opened as recordsets, replaced all such recordset with sorted queries of the tables.

Build 20210603

Design Change

1. Created a filter-by-office-title function for selecting offices
2. In order to filter by office title, I added c_dy, c_dynasty, and c_dynasty_chn both to ZZ_OFFICE_CODE and to Z_SCRATCH_DUMMY_OC.

Bug Fixes

1. Cleared the scratch table for status codes when clicking on the highest tree level to fix a duplicate-value bug.

Build 20210601

This is the initial version created by using the new CBDB dataset in Access. Since some scratch tables are unique to the SQL Server version, I have created an additional Access file SupplementalTablesSQL_Server.mdb with those tables, which need to be included in the migration

CBDB Interface Version BG:

Changes:

1. The ability to save and import lists of codes to be used in search has been added to **Query Associations**, **Query Office Holding**, **Query by Methods of Entry**, **Query Status** and **Query Texts and Roles**.
2. A bug in which one could not use multi-select from the root list in **Query by Methods of Entry** has been fixed.

CBDB Interface Version BF:

Changes:

1. A Neo4j export function has been added to all forms.
2. There is an accumulation of bug fixes in this release.

CBDB Interface Version BE:

Changes:

1. **LookAtTexts**, a form for looking at people's role in the production of texts, has been added.
2. **LookAtGroupData**, a form to quick collect data on groups of people, has been added.

Build 20220627

Design Changes

1. Made the changes necessary for users to export dynasty information from LookAtKinship:
   a. Added c_dy, c_dynasty, c_dynasty_chn, c_kin_dy, c_kin_dynasty, and c_kin_dynasty_chn to ZZ_SCRATCH_KIN, ZZ_SCRATCH_KINNET, ZZ_SCRATCH_KINNET_EDGE, ZZ_SCRATCH_GEPHI_NODE, and ZZ_SCRATCH_GEPHI_NODE_DISTINCT
   b. Added the new fields to the forms frmZZ_SCRATCH_KIN and frmZZ_SCRATCH_KINNET
   c. Revised the code for exporting to UCINet and Gephi
2. Made the changes necessary for users to export dynasty information from LookAtNetworks by revising the code for exporting to UCINet and Gephi
3. Created the form **LookAtGroupData**.
   a. Created tables Z_SCRATCH_DUMMY_OF, Z_SCRATCH_DUMMY_ENTRY, Z_SCRATCH_DUMMY_BA
   b. Added c_sequence, etc. to ZZ_SCRATCH_ENTRY
   c. Created the (sub)forms:
      ZZ_SCRATCH_GROUP_STATUS
      ZZ_SCRATCH_GROUP_OFFICE
      ZZ_SCRATCH_GROUP_ENTRY
      ZZ_SCRATCH_GROUP_TEXT
      ZZ_SCRATCH_GROUP_PLACE
   d. Added fields to ZZZ_BIOG_ADDR_DATA
      Created the table ZZ_SCRATCH_BIOG_ADDR_DATA
   e. Added a routine to rebuild ZZZ_BIOG_ADDR_DATA to DATA3

Build 20220425

1. A "Simplify Kinship Terms" option has been added to **Query Kinship**.
2. A "Store Person ID" button has been added to the **Browser**, so that once the user has selected a person, that person can be recalled for use in **Query Kinship**, **Query Social Networks**, and **Query Pair-wise Association** by simply clicking on the "Recall Person IDs" command button.

Build 20220315

1. Information on sources for records has been added to all the forms.

CBDB Interface Version BD:

Changes:

1. The forms for selecting codes for association, entry into government service, office titles, and status now support the multi-select option. The user can click on items in a list to select those that are of interest.
2. The user can now filter office titles by name when selecting office. This feature replaces the earlier search function. Because the filtering produces a list of offices from different dynasties, the form now includes information about the dynasty associate with each office name.

Build 20211102

Design Change

1. In **LookAtOffice**, selecting an office now provides more information about the dynasty of the selected office.

2. In the **Browser** for individuals, one can now search by Surname + Office Title in either Chinese or in *pinyin*.

Build 20210826

Design Change

In the **Browser** for individuals, using pinyin now has three options:
(1) lower case: it looks for the string anywhere in the name;
(2) upper case: it looks for the string only at the beginnings of words ("Hao" will select both Zhang Hao and Hao Jing;
(3) string beginning with "!:" it looks only at the beginning of the surname (just "Hao Jing")

Build 20210609

Bug Fixes

1. Removed all references to ZZZ_ALL_BIOG_ADDR, which has been replaced in functionality by ZZZ_BIOG_MAIN
2. Added the field c_kin_rel_count to ZZ_SCRATCH_KINNET_EDGE.

Build 20210606

Bug Fixes

1. Changed all references to TEXT_DATA to BIOG_TEXT_DATA in the Admin Version of BD (the Used BC version had already been corrected).

20210603 (Initial build for BD)

Design Change

1. Created a filter-by-office-title function for selecting offices
2. In order to filter by office title, I added c_dy, c_dynasty, and c_dynasty_chn both to ZZ_OFFICE_CODE and to Z_SCRATCH_DUMMY_OC.

Bug Fixes

1. Cleared the scratch table for status codes when clicking on the highest tree level to fix a duplicate-value bug.

CBDB Interface Version BC:

Changes:

1. Index Place is now handled the way Index Year is: both are added to BIOG_MAIN. Index Year derives from birth year or death year (values in BIOG_MAIN), if they are known, and, if not, from data elsewhere in the system. A source code for the index year value is included in BIOG_MAIN. Similarly, Index Place is derived from data in BIOG_ADDR_DATA, and the type code for Index Place is included in BIOG_MAIN. As a result, users should use ZZZ_BIOG_MAIN instead of ZZZ_ALL_BIOG_ADDR as the table for building queries about people
2. Because scholars may prefer to use different categories of place association to define the Index Place, the BC version of CBDB includes a form accessible from the Navigation Pane through which the user can change the hierarchy of places associations used to define Index Place. (See the explanation under the heading "Navigation Pane."
3. Searching by dynasty behaves a bit differently. When one specifies that the "from" and "to" dynasties are the same, for example from Yuan 元 to Yuan, the search routines now look for data with that particular *code* (Yuan = 18) rather than for all dynasties that have a temporal overlap with the selected dynasty (for instance, the Yuan [1234-1367], overlaps with the Song dynasty [960-1279]).
4. To select codes for **Association**, **Entry**, **Office**, **Place**, and **Status**, CBDB now allow the user to select *more than one category* at a time. As a result, the form behaves a bit differently than before: the form always moves to the *bottom* of the list for any type in the right-hand list box, so that the user will need to scroll up to see the entire list. Also, while the search function still works, it does not highlight the target record.
5. Various bugs were fixed in the behavior of the forms. In particular, the address tree now does a better job checking and sorting out subordinate relations between administrative units.

CBDB Interface Version BB:

Changes:

1. The **Index Year** has been significantly revised. It now represents the ***birth year*** of the individual. For individuals for whom the year of birth is not known, CBDB uses a series of calculations based on other data (see the main text for a detailed explanation). While CBDB has derived the index year for individuals in the past, it now uses *derived* index years to derive yet more index years when it is possible. The **Index Year Type Code** preserves the steps in the derivation. Please note that each iteration is yet more inaccurate, but we believe that for running queries an index year that is off by a decade is still better than having no index year at all.

2. CBDB is now explicit that the address codes used for searches is an **Index Place**, a construct analogous to **Index Year**. While the address codes used for searches always have been assigned according to a hierarchy of place information, we believe that it is better to be explicit about the status of index place. Even the "basic place affiliation" (*jiguan* 籍貫) has problems in its historical interpretation, so that it always remains useful to be circumspect about the assignment of index places. They are largely—but not entirely—reliable.
3. The approach to kinship searches has been revised. In concatenating kinship relations in iterative searches, CBDB now automatically simplifies eight relationships:
   BB (brother's brother) ➔ Brother
   ZB (sister's brother) ➔ Brother
   BZ (brother's sister) ➔ Sister
   ZZ (sister's sister) ➔ Sister
   SB (son's brother) ➔ Son
   SZ (son's sister) ➔ Daughter
   DB (daughter's brother) ➔ Son
   DZ (daughter's sister) ➔ Daughter
   The effect of this change is that the "collateral" parameter in the relationship *decreases by 1*, so that the relationship (and the person identified through the relationship) may now remain within the search limits specified by the user. Moreover, CBDB may identify additional new relations of the newly permitted individual who would not have appeared in the earlier version of the search.
4. CBDB now has a MS Access "Look at Status" form to allow users to explore categories of social distinction.
5. All the MS Access query forms now permit using **dynasty** as a search parameter. There remain many individuals for which CBDB lacks the data to assign an index year, and while searches by dynasty define a rather broad time period, still it provides some temporal specification that we believe may prove useful.

CBDB Interface Version BA:

Changes:

1. This release fixes a major bug in the way that the XY count is counted when outputting data to Gephi.
2. This release adds the ability to export to Gephi in the **Query Associations** and **Query Pair-Wise Associations** forms.
3. The output to Gephi now includes the XY coordinates to allow users to take advantage of the Geographic Distribution visualization add-on in Gephi.

CBDB Interface Version AZ:

Changes:

1. Removal of the ability to filter by superior administrative unit when selecting places.
2. Addition of the ability to include or disallow the inclusion of subordinate administrative units when running queries that involve restrictions to specific places.

CBDB Interface Version AY:

This release is effective as of 2019-04-29. Additions include:

1, Michael Fuller updated address selector to allow users to filter place names by superior administrative units.

2, Edith Enright systematically refined our label translations in Access query interface.

CBDB Data Release 20220312

Changes:

1. [To be Added]

Build 20220315

1. Source information was added to ZZZ_NONKIN_BIOG_ADDR

CBDB Data Release 20211222

Changes:

1. 19,286 new persons and their 3,689 alternative names, 19,576 records of entry data from the most complete list of civil service examination passers for the Song period.(Contributor: Yang Xu)
2. 34,574 pairs of Ming and Qing dynasties kinship relationships and 15,312 records of postings data in Ming Dynasty from the Academia Sinica Name Authority Database.
3. 3,267 new Koryo and Choson persons in Korea history with their 6,939 alternative names, 3,031 records of social status from Chronicles and Biographies of Choson. (Contributor: Yafei Chen)
4. 22,363 records of data from the biography chapter from Ming and Qing local gazetteers. (Contributor: CBDB Crowdsourcing group)

CBDB Data Release 20210525

Changes:

1. 17,000 new persons names and 10,486 social association for Ming and Qing dynasties from Academia Sinica Name Authority Database.

2. 17,560 book titles from the Union Catalogue of Chinese Collectanea. (Contributor: Edith Enright)
3. 20,678 entries, 2,180 alternative names and 16,020 postings information for the Ming and Qing dynasties from local gazetteers.
4. 79 schools with 379 scholars belonging information. (Contributor: Mengxi Bi)
5. 600 new Ming and Qing persons with their data from the biography chapters in local gazetteers were created by CBDB crowdsourcing contributors.

CBDB Data Release 20201110

Changes:

1. The Index Year has been significantly revised. It now represents the birth year of the individual. For individuals for whom the year of birth is not known, CBDB uses a series of calculations based on other data. While CBDB has derived the index year for individuals in the past, it now uses derived index years to derive yet more index years when it is possible. The Index Year Type Code preserves the steps in the derivation. Please note that each iteration is yet more inaccurate, but we believe that for running queries an index year that is off by a decade is still better than having no index year at all.
2. 417 garrison addresses for Ming dynasty. (Contributor: Ruoran Cheng)
3. 376 social status records for thinkers. (Contributor: Mengxi Bi)
4. Kinship and entry data mistakes were fixed. (Contributors: Moqin Zhou, Song Chen)
5. Added TEXT_INSTANCE_DATA table to collect version information of books. (Contributors: Edith Enright, Song Chen)
6. Change TEXT_DATA table name to BIOG_TEXT_DATA.

CBDB Data Release 20190424

Changes:

1. 18,124 new social assignations for Tang and Five Dynasties from *The communication poems for Tang and Five Dynasties figures* 唐五代人交往詩索引 with 4,380 new figures, 702 new alternative names and 671 new kinship relationships etc. (contributor: Shuhua Zhang 張淑華, Qiong Yang 楊瓊, Yongqin Li 李永琴, Chengguo Pei 裴成國)
2. 5,895 new Tang addresses with 11,844 belongs data from *General History of Chinese Administrative Divisions* 中國行政區劃通史. (contributor: Chao Wei 魏超, Yifan Wang 王一帆, Yun Xing 邢雲, Wen Luo 駱文, Yuying Yuan 袁鈺瑩)
3. 1,200 new address names with 670 new address belongs data for Jin Dynasty. (contributor: Jingjia Qiu 邱靖嘉)

4. 1,765 new office titles for Jin dynasty. (contributor: Jingjia Qiu 邱靖嘉)

CBDB Data Release 20180831

Changes:

1. 5,300 new persons added with 5,300 entries *jiguan* data, 4,000 other entries, and 2,300 alternative names from the Name Authority Database at Academia Sinica;
2. 8,000 person ID entries are mapped between CBDB and the Name Authority Database;
3. Bugs were fixed in pinyin entries and *jiguan* data etc.

CBDB Interface Version AX

This release is effective as of 2018-12-14. Additions include:

1. An important feature of kinship network algorithm was added. The duplicate records for kinship relationships can be calculated correctly in this new algorithm.
2. The query forms now have a **Store Person IDs** button to save the list of people created in a query. That stored list of IDs can be recalled for use in other forms (where relevant) through a new **Recall Person IDs** button.

CBDB AW Version:

This release is effective as of 2018-09-01. Changes to the interface include:

1. Michael Fuller created **Relink Tables** button on the Navigation panel as a new and more efficient mechanism to connect the user interface and the backend data which is now in three separate files with name that indicate the date of release of the data, for example CBDB_20190424_DATA1.mdb, CBDB_20190424_DATA2.mdb, CBDB_20190424_DATA3.mdb.
2. The database was thoroughly cleaned with the foreign key mechanism (contributor: Fu Qunchao 傅群超);

20170829CBDB AV Version:

This release is effective as of 2017-09-07. Additions include:

Data

1. 51,551 new persons with 34,447 posting from local gazetteers;
2. 467 Wuzhou jinshi degree holders from Song to Yuan dynasties;
3. 841 figures with 1,725 kinship associations and 381 social associations from 全元文，宋濂全集，遜志齋集 etc. (contributor: Yu Wen 于文);

Interface

1. Michael Fuller and Chen Song has designed a **Rerun** function in Query Social Networks to run queries using the results from the previous query.
2. A new query function named Query Place Associations.
3. The Office holding query form now allows the user to select both the place of the posting and the index place of the office-holder.

20170424CBDB AU Version：

This release is effective as of 2017-04-25. The Access interface has not changed: It remains the AU version, but the data has been updated to the 2010425 release. Additions include:

Data

1. 789 Wuzhou figures with 500 biographical address data, 1,800 kinship relations and other data from 全宋文 and 金華府志 (contributor : Du Feiran 杜斐然);
2. 700 biographical addresses, 3,000 kinship relations, 500 postings and other data from 全元文, 宋濂全集 and 藥房樵唱 (contributor : Yu Wen 于文);
3. 6,700 figures were connected to the 明清人名權威檔案 database (contributor: Institute of History and Philology, Academia Sinica);
4. Tang bureaucratic tree added (contributor: Lik Hang Tsui 徐力恆)
5. Fixed several mistakes in the bureaucratic and biographical data. Thanks to Chu Pingtzu 祝平次 and Yang Guang 楊光's for reporting them.

20170310CBDB AU Version:

This release is effective as of 2017-03-13. Additions include:

Data

1. Data on 8,836 Tang figures and their 15,138 postings (source: 唐九卿考, 唐刺史考全編);
2. 5,921 Tang personid were disambiguated (contributor: Wen Xin 文欣);
3. 770 figures from 全元文 (contributor: Yu Wen 于文);
4. 1498 social status data from the Tang Dynasty (source: 唐五代人物傳記資料綜合索引);

Interface

1. Updated User Guide with English and Chinese versions (collated by Lik Hang Tsui 徐力恆);

2. Michael Fuller and Chu Ping-tzu rewrote several critical codes in CBDB Access Database so that it can run on both 32-bit and 64-bit MS Windows;
3. Michael Fuller added import person id list function to the Query Mediated Associations interface.

20150202CBDB AS Version:

This release is effective as of 2015-03-18. Additions to previous versions include:

Data

1. 36,826 new persons and 38,565 new entry records of Ming and Qing Civil Service Jinshi Degree holder (source: 明清人物題名碑);
2. 3,142 Liao Dynasty office titles with Liao office tree (contributor: Cao Liu 曹流);
3. Yuan office tree (contributor: Yi Ding 丁一, Yu Yue 于月);
4. 1,004 Song Yuan Academies (contributor: Stephen P. Ford);
5. 272 China emperors with their Posthumous Name (謚號), Honorific name (廟號);

Interface

1. Revised Help Files.
2. Place name filter to select a set of places for search
3. Searching places based on geographic coordinates and proximity

20140310CBDB AR Version

This release, on date 2014-03-10, is built upon the Oct. 8 2013 dataset. Major changes in this version include:

Data

1. 27,000 association data from Ming Biographical Materials (明人傳記資料索引)(contributor: Qiaomei Tang 唐巧美 and Hui Cheng 程卉)
2. 5,000 entry data from Ming civil service high degree holders (jinshi)
3. 3,700 posting data from Ming civil service high degree holders (jinshi)
4. 3,300 books from the Ming Qing Women Writers database (MQWW) and Ming Biographical Materials (明人傳記資料索引)
5. 2,800 address codes were updated (contributor: Yi Ding 丁一)

Interface

1. This release also fixed minor mistakes in the posted_to_office data and altname data in the previous standalone database.
2. In addition, new search and selection features have been added to the "LookAt" forms as well as greater flexibility in choosing whether to use index years. All the search routines have been rewritten in SQL to greatly speed up the searches.

20131008CBDB AQ Version:

This release 20131008CBDBaq.mdb, on date 2013-10-08, is built upon the Sep. 21 2013 dataset. This version adds biographical data on *200,000 new men and women* to the dataset from the 7th to the 20th century, resulting in a total number of 325,000 individuals. Major new additions include:

Data

1. 50,000 principals and kin from Tang and Five Dynasties tomb biographies
2. data on 14,000 civil service high degree holders (jinshi) and 130,000 of their kin from 52 Ming dynasty examination years
3. principals and kin from the 1148 and 1256 examinations
4. selected biographical data from the *Index of Ming Biographical Materials* (明人傳記資料索引)
5. new data on the kin and social relations of women writers
6. a variety of new and expanded code tables
7. New data was developed through the contributions by and in collaboration with Profs. Ping Yao, Nicolas Tackett, Liu Cheng-yun, and Grace Fong.

CBDB Patch:

[Important!] This is the patch for fixing the TreeView selection problem. If your copy of Access gives you an error when you try to select an office via the TreeView in **LookAtOffices** (**Query Office Holding**) or select an association in **LookAtAssociations** (**Query Associations**), this is because you do not have the correct version of the "Microsoft Windows Common Controls 6.0 (SP6)" added to your Visual Basic environment.

We have prepared a document to walk you through the steps for fixing this problem. Please download this RAR file, unzip it, and follow the instructions in the PDF file.

20130610CBDB AN Version:

This release, on date 2013-07-08, is built upon the June 10th 2013 dataset which adds biographical information for 12,773 new individuals to the January 2012 dataset and results in 128,923 as the total number of individuals. The following lists the details of the addition:

Data

1. Incorporated individuals, their kin and their associates from: the Ming Qing Women Writers database (MQWW) (contributor: Professor Grace Fong and the CBDB Beida editors), *Quan Song Wen letters* 全宋文書信 (contributor: Pingtzu Chu 祝平次, Beida, Chen Liu 劉晨), *Song Lian Quan Ji* 宋濂全集 (contributor: Qiaomei Tang 唐巧美), Ji Yun 紀昀's associates (contributor: Clea Walford), Lu You 陸游's associates (contributor: Ziyu Zhou 周子鈺), and the 1148 紹興十八年 exam passers (contributor: Ziyu Zhou 周子鈺).
2. Collaborated with IHP, Academia Sinica 中研院史語所 to incorporated the basic information, alternative names, and entry data for 2,912 individuals from the 明清檔案人名權威資料 database (system number 13197 to 16110). It results in 2,134 new individuals (because some of the them already exist in CBDB), 6,540 alternative names, and 2,515 entry data.
3. Collaborated with IHP to incorporate the basic biographical data, alternative names, and address data for the 9,900 individuals in the Ming Ren Chuan Ji Zi Liao Suo Yin 明人傳記資料索引, which has given us 7,400 new individuals, 15,000 alternative names, and 8,600 biog address data.
4. Added 987 new individuals who were the kin of the subjects in the biographies section of Song Shi 宋史.
5. Added 8,800 social association data from the Quan Song Wen letters 全宋文書信 and 114,000 associations from *Index to Song Biographical Materials* 宋人傳記資料索引.
6. Added 14,447 posting data from the Kyoto Tang database 唐代人物知識ベース and 22,067 from *Index to Yuan Biographical Materials* 元人傳記資料索引.

Interface

1. From the system side, in this release we also refactored a bunch of database tables (for example, social institutions) in order to accommodate more detailed information about one's life and to enable such queries.

20120105CBDB AM Version:

This release, on date 2013-03-14, is built upon the January 2012 dataset and the 20120105CBDBal.mdb. Major changes in this version:

Data

1. Addition of 18,000 Tang-Wudai, Yuan, Ming, and Qing office codes.
2. Restructure of Social Institution tables: 8 code tables and 1 data table where we can record the relation between a person and a social institution.

20120105CBDB AL Version:

This release 20120105CBDBal.mdb, on date 2012-08-27, is built upon the January 2012 dataset. It contains the biographical information for 116,149 historical figures in the Chinese history. It also comes with the most up-to-date built-in queries, including the latest revision of the Query Kinship and Query Social Network functionalities. Major changes in this version:

Data

1. It includes Han addresses (漢代地名) and a new Ethnicity/Tribe code table.
2. It uses the new ethnicity coding for people.

Interface

1. "Look up Data on an Individual 按人查詢" now accepts search via alternative names. E.g. You are able to find 蘇軾 via 蘇東坡 now.

3. Bug fixed in "Query Association 查詢社會關係" and improve the search performance.

NOTE: It is known that some of the CBDB built-in queries do not function on 64-bit version of Microsoft Office 2010. It is because the 64-bit Office is not compatible with former VBA programs (see the official annoucement here), which the CBDB queries were built with. Therefore, if you are running a 64-bit Office, please consider to re-install a 32-bit version Office 2010 on your 64-bit Windows machine. (Yes, you can still run the 64-bit Windows Operating System). Not sure which version are you running? Follow this link.

20110705CBDB AF Version:

This release, on date 2012-02-07, is the last release for the July 2011 dataset.

Data

1. It does not add significant new data to the July 2011 release but some code tables have been improved and duplicates have been removed.

Interface

NOTE: Some of the built in queries do not function on 64 bit machines. This will be corrected in the near future.

1. Bug fixed for the "Enter Biographical Data 輸入傳記資料"

2. Bug fixed in the "Look up Data on an Individual 按人查詢" buttons.