The
China Biographical Database

User's Guide

Michael A. Fuller

Revised Version 3.1
March 15, 2020

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 January 1, 2019 it had data on about 422,000 figures with well over 100,000 more in preparation; further data on figures already in the data are frequently added. We are now preparing tables of place names and official titles to facilitate the systematic incorporation of 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 and Mr. Merrick Lex Berman, 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, and the bequest of the late Robert Hartwell to the Harvard-Yenching Institute. 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. ChineseAll is currently the leading funder of the CBDB project. Over the years many scholars have visited Harvard and contributed to the project.

This User's Guide explains the logic of CBDB as a relational database, the structure of its contents, and the primary query interfaces for getting data from the database.

Table of Contents

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 that has begun to emerge in recent years. Social network analysis (SNA) has been a tool for studying group structure in the social sciences for many decades. Recently, however, scholars 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.

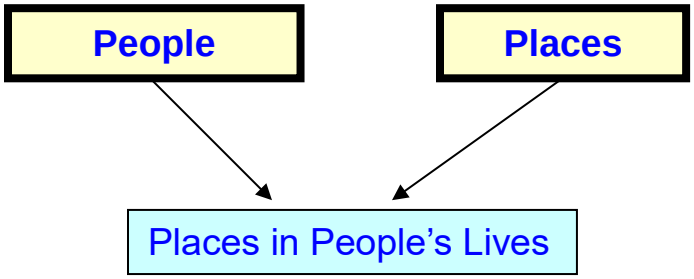
Chapter 1. Relational Databases

A. Relational Database and the Organization of Complex Data

The social historian Robert Hartwell, 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."

This structuring of relationships between entities is what a relational database does: it allows one to capture multiform relations between complex objects in the world that interact with one another. That is, "place" is a category of "things" in the world, but 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 category of "thing" in the world, 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:

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 structure, 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 the administrative level of a place, its superior or subordinate units, and the period of validity. For offices this would include where the office fit in the administrative hierarchy during a particular dynastic period.

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 served, and (3) when he served.

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 Sima Guang 司馬光 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 Sima Guang 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 a normalized database.

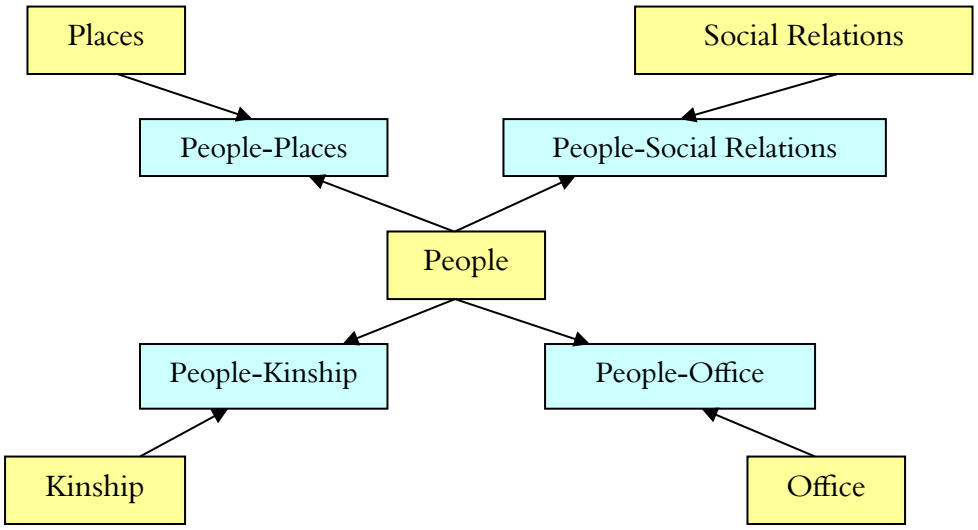
In the example of a person's relationship to places discussed above, we encounter the fact that a person can move to many different places. This is called a "one-to-many" relationship. If one were to try to represent this relationship through a simple table with rows and columns, we either could create 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 entity like PEOPLE-PLACES (and a table to represent that entity) 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 Ziqian. 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 entity, PEOPLE-WRITINGS, to capture the data.

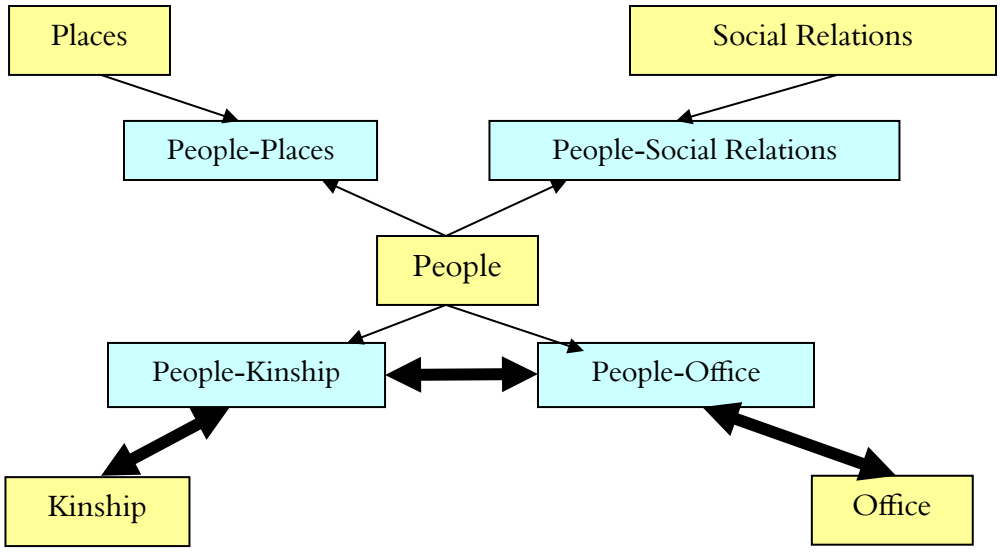
These three rules—normalize data, create new tables for one-to-many relations, and treat many-to-many like one-to-many—are important if you wish to add new data types to CBDB.

C. Relational Databases and the Interactions of Complex Data

Consider the following set of entities and their relations with the basic entity PEOPLE:

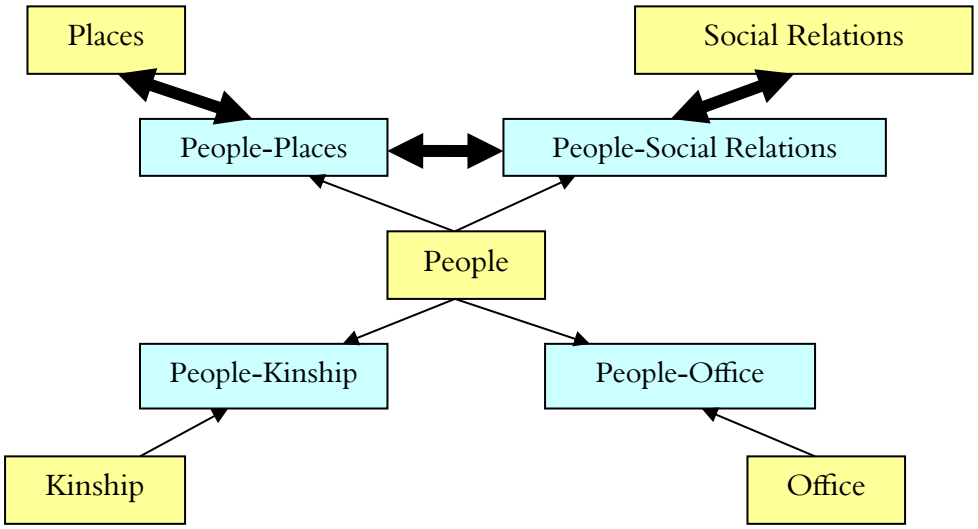


Although there is no direct link between KINSHIP and OFFICE, we still can explore the relation between them through the data we have accumulated about people. We can ask questions like "Was the role of medical officer hereditary, that is, were medical officers the sons or nephews of medical officers, and did the families of medical officers marry the 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?



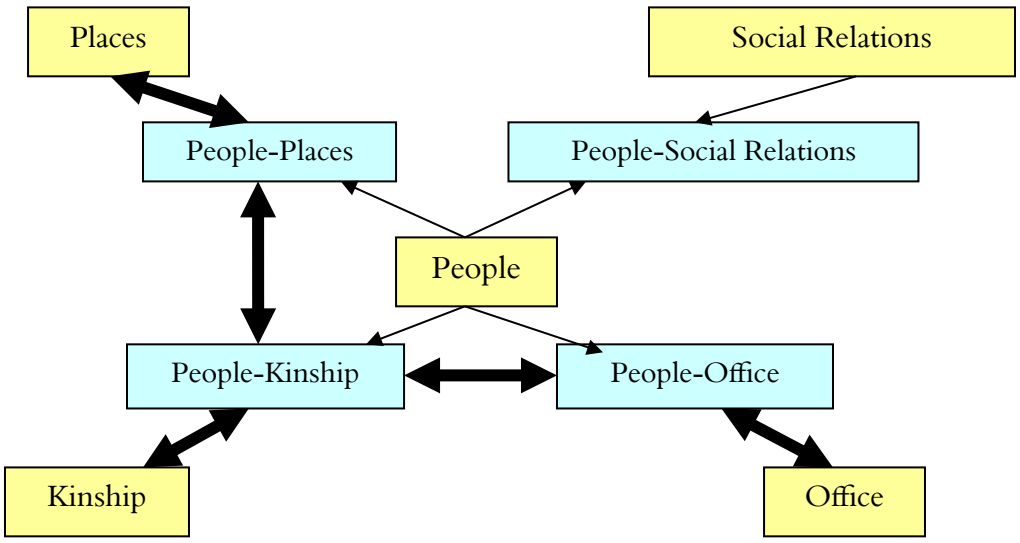
Querying the Relationship between OFFICE and KINSHIP

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. **Events** of significance in which a person participates
8. **Possessions**
9. **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:

10. **Texts** (including primary texts, secondary texts, and paleographic data)
11. **Data Sources** from which CBDB draws its information (this includes 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 three aspects on which we have focused are administrative geography, physical location, and bureaucratic structure:

12. **Geographic Administrative Hierarchy** (defined in political terms as administrative units)
13. **Longitude and Latitude** (fixed locations in space required for historical comparisons)
14. **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, the claims people made about their ancestral membership in a particular clan from a particular place (like the Cui clan of Boling) were difficult to verify. The combination of place name and clan name defined a *choronym*. The codes for these choronyms are in the table CHORONYN_CODES.

c. **Index Year**

For computational purposes, CBDB needs a single year value to locate a person in time. The *index year* (the "sixtieth year of age") is an artificial value used in analyses. The rules for calculating the value are complex:

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

Rule 1: If ego's death year < (ego's birth year + 60 - 1) , then: ego's index year = ego's death year; otherwise: ego's index year = (ego's birth year + 60 - 1)

Rule 2: If we only know ego's death year, then: ego's index year = ego's death year

Rule 3: If we only know ego's birth year, then: ego's index year = ego's birth year + 60 - 1

Rule 4W: Ego's index year = (husband's birth year +3)+60-1 = husband's birth year + 62 (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) + (60-30) = his Jinshi year + 30

Rule 5W: Ego's index year = the year her husband obtained the Jinshi (進士) +30+3 = husband's Jinshi year + 33

Rule 6: Ego's index year = the year he obtained the Juren (舉人) + (60-27) = his Juren year +33

Rule 6W: Ego's index year = the year her husband obtained the Juren (舉人) +33+3 = husband's Juren year + 36

Rule 7: Ego's index year = the year he obtained the Xuicai (秀才/生員) + (60-21) = his Xuicai year + 39

Rule 7W: Ego's index year = the year her husband obtained the Xuicai (秀才/生員年) +39+3 = husband's Xuicai year + 42

*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) + 60 - 1 = father's birth year + 89)

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) + 60 - 1 = oldest child's birth year + 29)

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) + 60 - 1 = oldest child's birth year + 32)

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)+60-1 = older brother's birth year + 61)

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)+60-1 = younger brother's birth year + 57)

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)+60-1 = birth year of oldest son-in-law + 32)

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)+60-1 = birth year of oldest son-in-law + 35)

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)+60-1 = grandfather's birth year + 119)

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

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 # 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's generation" = 1, "grandfather's generation = 2, and so on), "down, i.e.,

descendent generation" (son = 1, grandson = 2, etc.), "collateral relation" ("brother" = 1, "brother's wife's sister" =2...), and "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.

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 one creates a new category of **Association**, one 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**}. Some associations are not dyadic because the relation is not to a person but to a more abstract or general object. The most important type of association of this type is the faction. Thus we have {**A** "is member of the purged Yuanyou group" ∅} (∅ here is the *Null* (Empty) element.) A few rare relations are **triadic**: {**A** "ordered" **B** "to execute" **C**}. CBDB has added a third field to capture such situations but does not yet process such triadic relations in its analytic routines.

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.

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
2. Associated person
3. The association
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 person whose kinship relation established the association
7. The kinship relation, if the association was established through a relative of the *associated* person
8. The kin of the associate through whom the association was established
9. The name 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 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 social institution at or through which the association was established
14. The occasion on which the association was established
15. 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 scholarly topic around which the association was formed

Source and Notes

18. Source
19. Note

d. **Future Developments** (in planning)

1. **Mourning Associations**

In China, a person can choose to participate in mourning for a teacher or a person connected through some other form of non-kinship association. Mourning has five aspects that are represented in the database:

Mourner
Mourned

Length of Mourning
Color of Mourning Robe
Date

It may prove that this category of association is most important in early China, but it should be expected that as Chinese culture transformed over time, sources for biography stressed different forms of activities and relationships. The database needs to be able to accommodate these historical shifts. At present CBDB has no way to record the details of mourning association: once we have more data from earlier individuals, we will assess the best way to proceed.

2. **Gift Giving**

Another type of association is created through the giving of gifts. This practice also has five aspects:

Giver
Recipient
Gift
Value (or quantity) of gift
Date

At present, CBDB does not record this information, which is perhaps more important in early biographical texts. As with mourning, we will assess the situation once we have more information.

4. Status

CBDB has a separate table to take note of a person's "social distinctiveness," that for which they are 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: it may be better to aggregate the two social roles, but this is largely an empirical question of how often such merged roles appear and whether they seem to have been viewed as a single "status" rather than two. The structure of a Status datum for a person is:

Person
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 need 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

6. Offices and Postings

CBDB currently lists over 24,000 office titles and—at least for the Song—their place in the government bureaucracy. **Postings** are entities at the intersection of people, the bureaucracy, and—since most 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 <> many addresses and one posting <> many office titles) require separate tables, information about postings requires *three* entities: a basic **postings** table, a **posted-to-office** table, and a **posted-to-office-address** table. Because the administrative unit is tied specifically to the *office* rather than the posting more generally, the address table records the relation of posting-office information and place:

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

Posted-to-Office
Posting ID
Office ID
Office Type
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 Name/Office Function/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 CHGIS project and relies on the spatial entity **Addresses**.

Addresses are specifically historical "instances" of place designation that refer to an administrative jurisdiction bounded in space with a particular name. They have a location defined by a point using x and y coordinates. If either the boundaries or the name changes, a new **address** 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. 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.

Robert Hartwell in his initial version of the database had co-located historical districts (or counties, *xian*) with the district boundaries in 1990. However, rather than using misleading boundaries, CBDB now uses the x-y coordinates of the seat of the administrative unit.

In sum, there are two tables that are part of the basic representation of places in CBDB:

Administrative Units: ADDR_CODES
Administrative hierarchy: ADDR_BELONGS_DATA

Considerations for Future Development

The **addresses** in CBDB's **Addresses** table have specific beginning and ending years. For historical data, one often does not know the specific year of the relationship that connects a person to a place. At present CBDB has been using a proliferation of ad hoc codes to deal with this historical uncertainty. On occasion, the historical information confronts one with a yet more fundamental uncertainty: sometimes one knows the name of the place related to the person, but there may be more than one **address** with that name. In the near

future, CBDB will restructure the coding of **addresses** to account for these uncertainties by converting all present **addresses** to a three-part code:

Address Name Code
Address Location Code
Address Instance Code

Because there is a historical continuity for place names, there usually are only a handful of places that share a particular name, and these names recur at more or less the same place throughout history, even if there are minor changes. Thus the *Address Location Code* identifies the series of historical instances of a name that appear at approximately the same place. For example, there are counties with the name "Ningyuan" 寧遠 in four separate locations (Hunan, Shaanxi, Shanxi, and Guangdong), and each has several address codes associated with that location (3 codes in Hunan, 4 in Shaanxi, 4 in Shanxi, and 5 in Guangdong). The *Address Instance Code* will distinguish the different administrative entities currently assigned different **address** codes in each location.

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 about 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 of an event in which people participate (EVENT_ADDR)
Places where people's possessions are (POSSESSION_ADDR)
The place where people participated in social institutions (BIOG_INST_DATA)

Note that at present CBDB does not preserve information about where people briefly visited, where they received their education, or where they wrote texts. CBDB now has a form (LookAtPlace) to allow the user to ask questions that integrate all these sources of place information.

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

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 ways:

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 (CBDB includes all attributions of a text name to a person)
Role ID
Text Role Codes
Role ID
Role description

Considerations for Future Development

Inscriptional materials have a few additional attributes recorded in separate tables that have not yet been developed because we have not yet dealt with this type of data:

alternate names
place where discovered
date of discovery
current location
source of information

These materials also can involve additional people, i.e., donors and recipient.

Since the texts can serve as sources for biographical information, CBDB records the publication information for the modern edition used, since source information for entries includes page numbers. However, CBDB does not aspire to serve as a standard reference for bibliographic information. It (at least at present and in the near future) will not list all the extant editions of texts for authors nor adjudicate which are the most reliable among those extant editions. Part of the future plans for CBDB on the web is to develop links between the database and other web resources: bibliographic sites certainly will be among such links.

10. Events

CBDB only recently has begun to make a concerted effort to record information about groups of people linked together by shared participation in important events. At present the list of events, which comes from the *Song shi jishi benmo* 宋史紀事本末, is heterogeneous and includes such matters as the successful crafting of peace treaties, rebellions, the removal and restoration of empresses, and various factional disputes. Since communities were forged by common purposes discovered in these events, they are worth preserving. CBDB records events through three tables:

Event-Codes
Event ID
Event name
Event date

Event-Data
Event ID
Event-Record ID (this is a unique number to track instances)
Person ID
Date (sometimes it takes a while for events to catch up with people)

Event-Address
Event-Record ID
Address ID (in case a person's involvement in an event is linked to more than one place.)

11. Possessions

CBDB has barely begun collecting data about possessions. One aspect of the information is the type of transaction (purchase, donation, etc.) recorded in a historical text through which we know about the possession. Although CBDB tables exist, we may change their structure to accommodate the type of information we get as we collect relevant material.

12. 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_Name_Codes
Institution Name ID
Institution Name

Social_Institution_Codes
Institution Name ID
Institution Code (this is a unique ID for each institution: the name may change, but the ID does not.)
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_Addr
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

Biog_Inst_ Data
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
EVENT_CODES	significant events
KINSHIP_CODES	the kinship categories of pre-modern China
OFFICE_CODES	the units of the bureaucratic organization of government
PLACE_ID	a stand-in for standard units of longitude and latitude
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
ENTRY_DATA	the initiating relations between people and institutions
EVENTS_ADDR	relations between people, events, and places
EVENTS_DATA	relations between people and events
KIN_DATA	the kinship relations connecting people
OFFICE_TYPE_TREE	the hierarchical structure of bureaucratic organizations
POSSESSION_ADDR	the relations between people, material goods, and place
POSSESSION_DATA	"Possessions" remain at the descriptive level at present
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
TEXT_DATA	the relations of people to texts
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

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
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
POSSESSION_ACT_CODES	the categories of relations between people and goods
SCHOLARLYTOPICS_CODES	the categories of topics of learning and scholarship
SOCIAL_INSTITUTION_TYPES	the categories of social institutions
TEXT_ROLE_CODES	the categories of relations between people and texts
YEAR_RANGE_CODES	the relative degree of exactness of a date
BIOG_INST_CODES	the roles a person plays in relation to an institution

SOCIAL_INSTITUTION_ADDR_ TYPES	the type of address (actual or derived) used for an institution

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.

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 ego-nets 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.

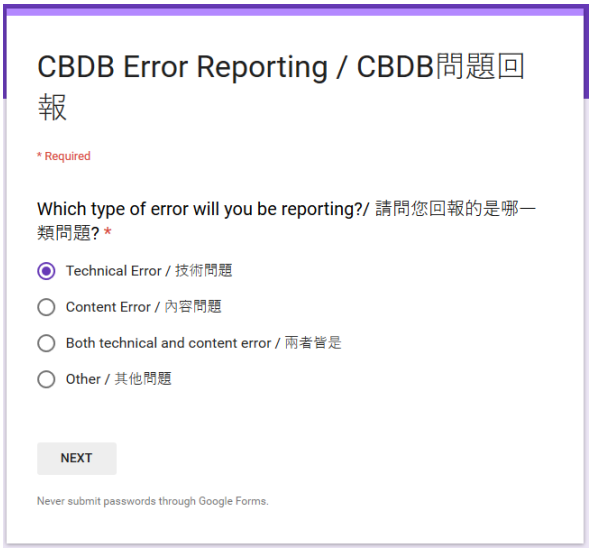
In more complicated queries, one can explore relations between groups of people by using the results of a search in one form as the input to a second form. Chapter 4, on advanced queries, considers an example of this approach. Beyond the six forms, however, Access also allows the user who is familiar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

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 eight command buttons open the browser and the seven analytic forms discussed above. The Navigation Pane also has three 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:

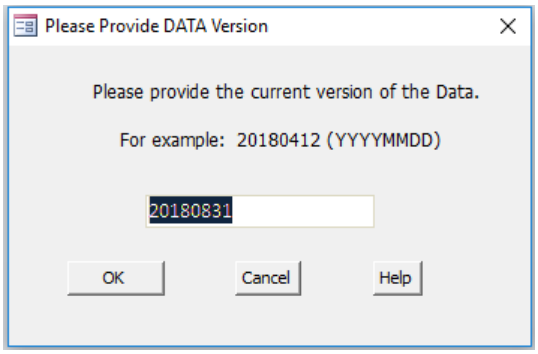


Select the type of error and fill in the information requested on the form.

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.

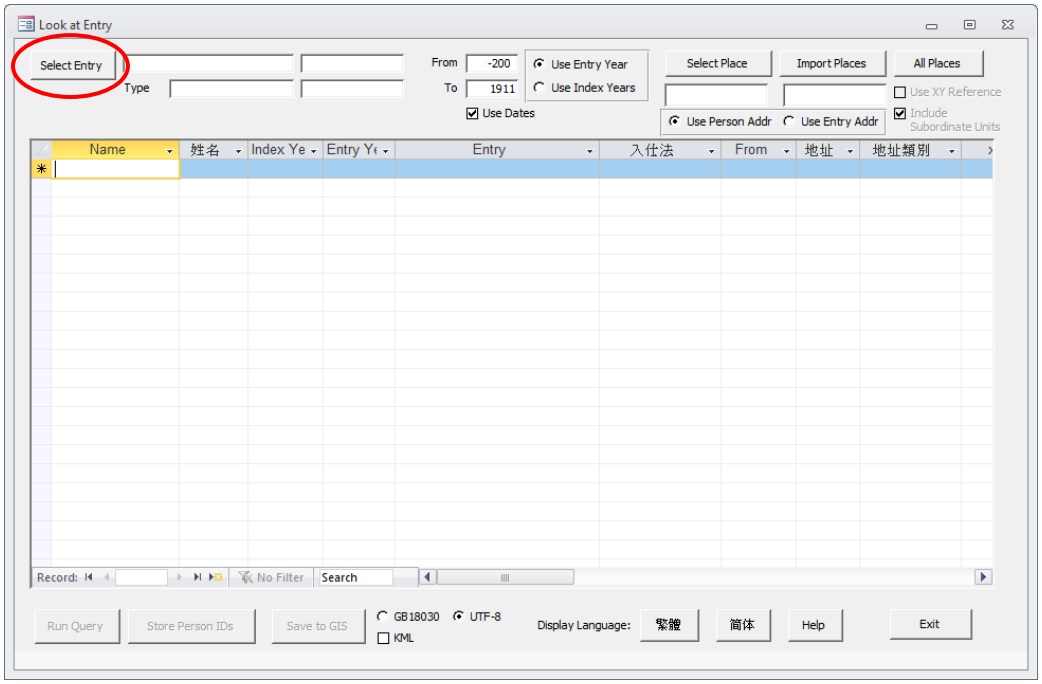
If there is a new release of the CBDB data, you can download the new data files into your CBDB folder and then link the interface to those new files. The CBDB data release will have a **date-stamp** in the form YYYYMMDD as part of the name of the files. If you click on "Relink Tables," a form will request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



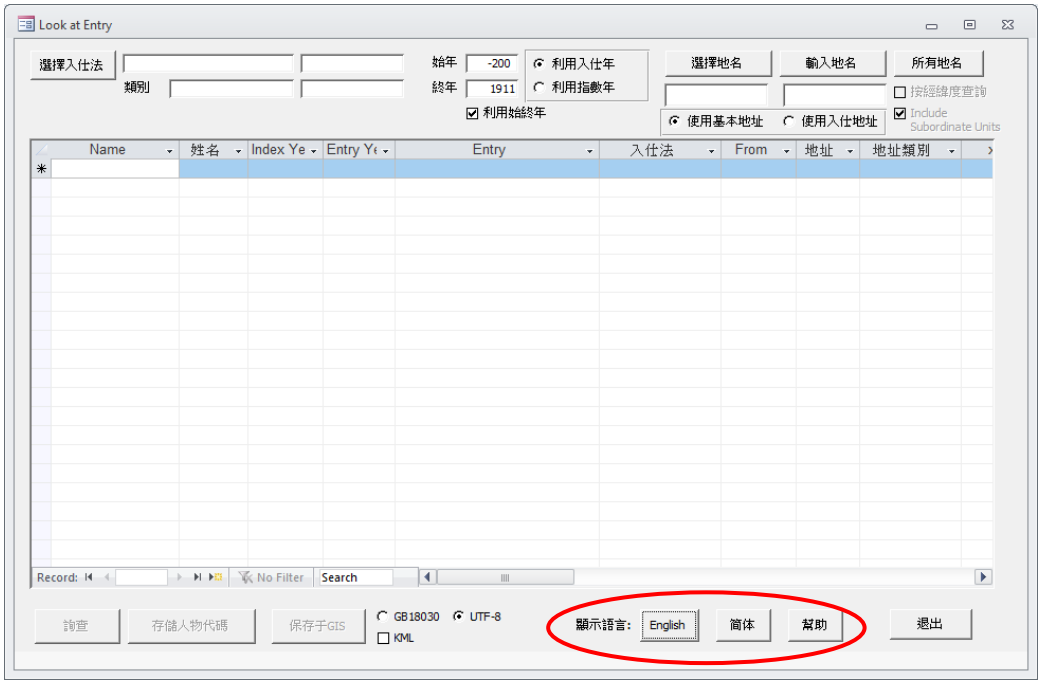
Simply fill in the new version information and click "OK."

A. Using the Form LookAtEntry

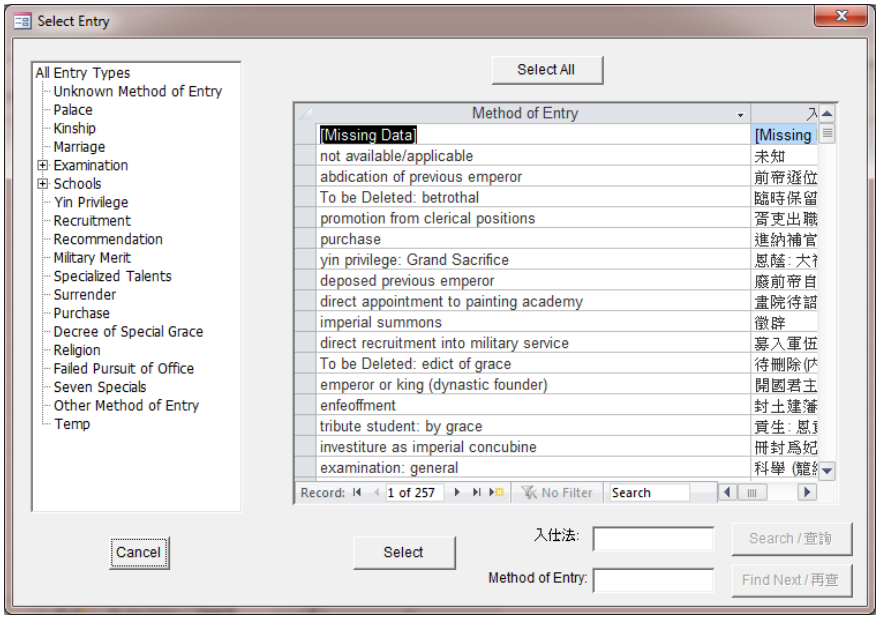
LookAtEntry is the simplest form. One opens it by clicking on "Query by Methods of Entry into Government" on the main page and clicks on the "Select Entry" button to choose a category:



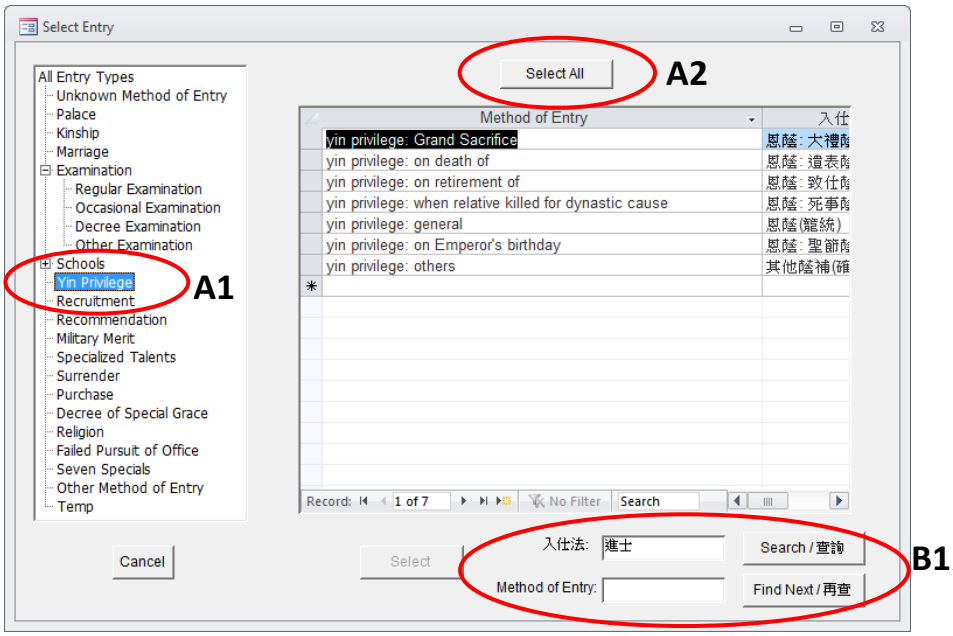
Note that all of the forms have the option to switch between English, traditional or simplified Chinese. When one clicks on the "繁體" label, it then gives one the option to return to English:



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:

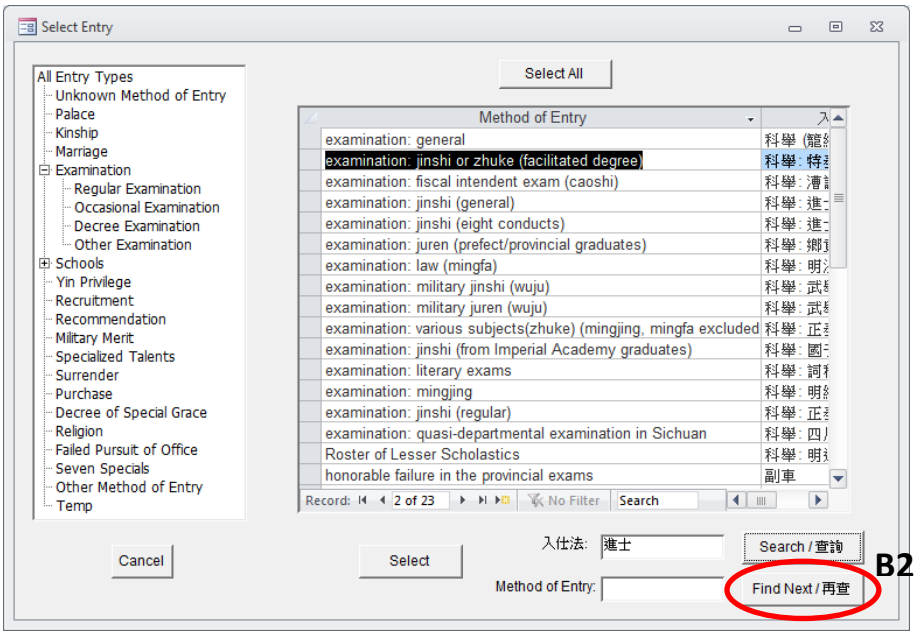


One can narrow the choices by looking at a particular general type of entry which is on the menu on the left of the window (A1):

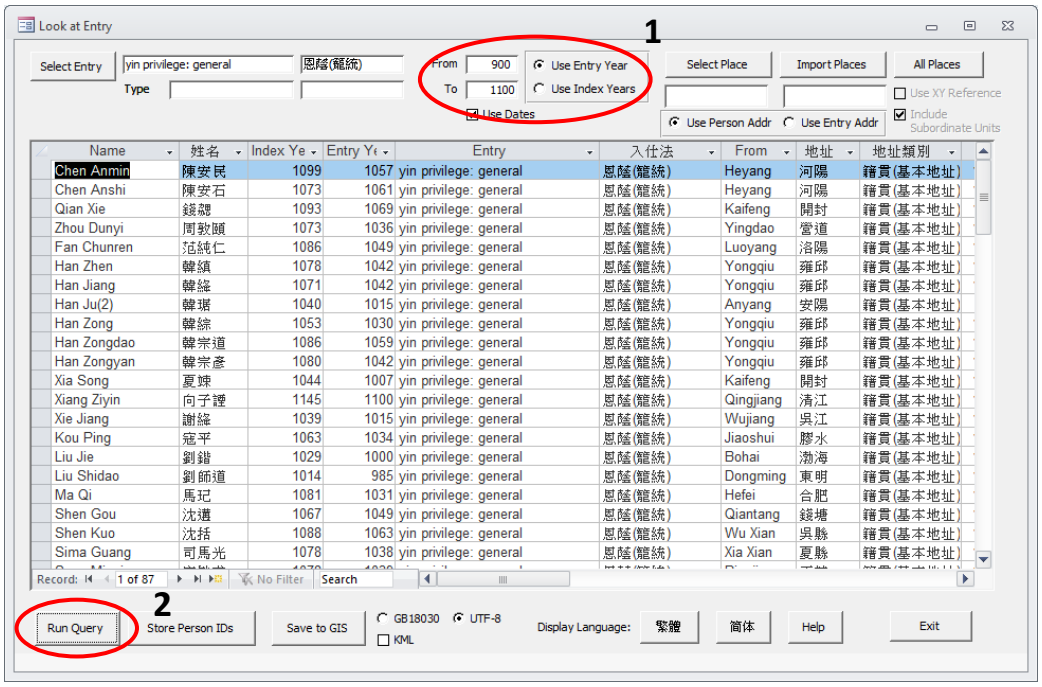


One can either select a specific method of entry from the menu on the right or select all the listed methods (A2). One also can search for a specific method using the search box located on the bottom right corner (B1). The searching rules for CBDB are to first look for the search

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:

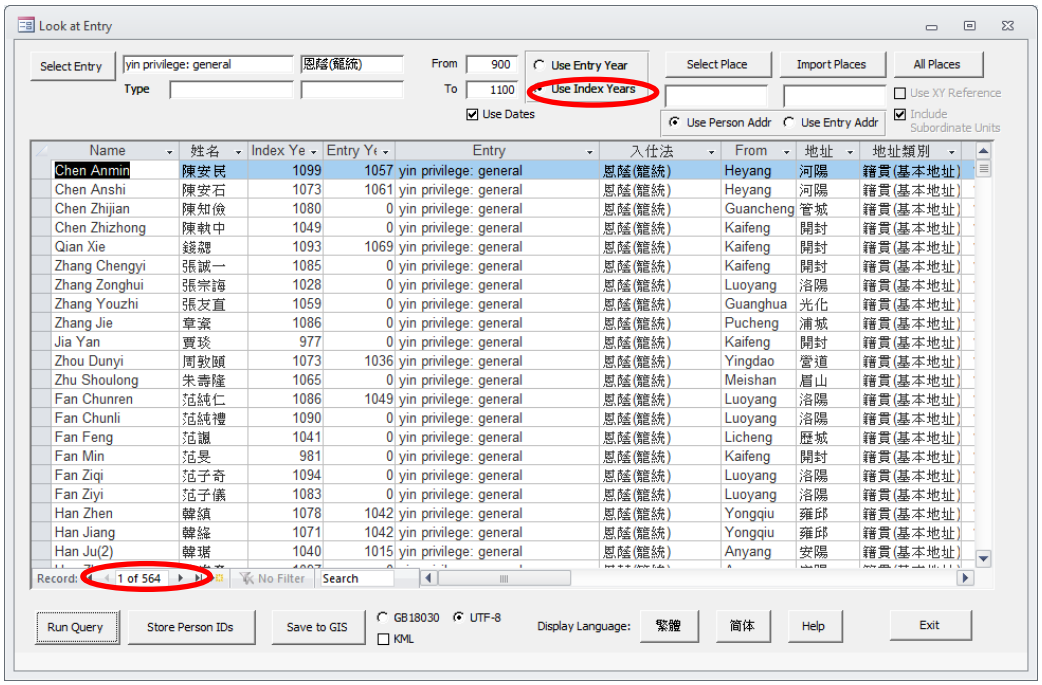


After one finds the method of entry and clicks **Select**, one returns to the LookAtEntry form, and can now choose the year range (1) to run the query (2):



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:



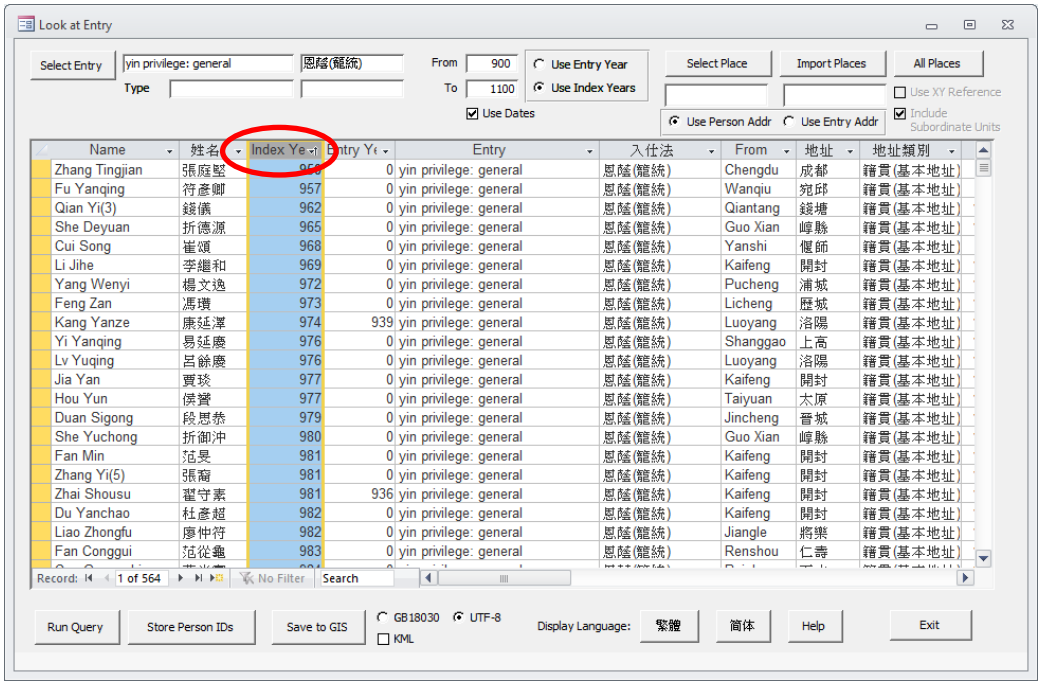
Note the number of 0s in the "Entry Year" column. This approach yields 564 records, compared with just 87 when using the entry year.

The table the query produces has 26 columns:

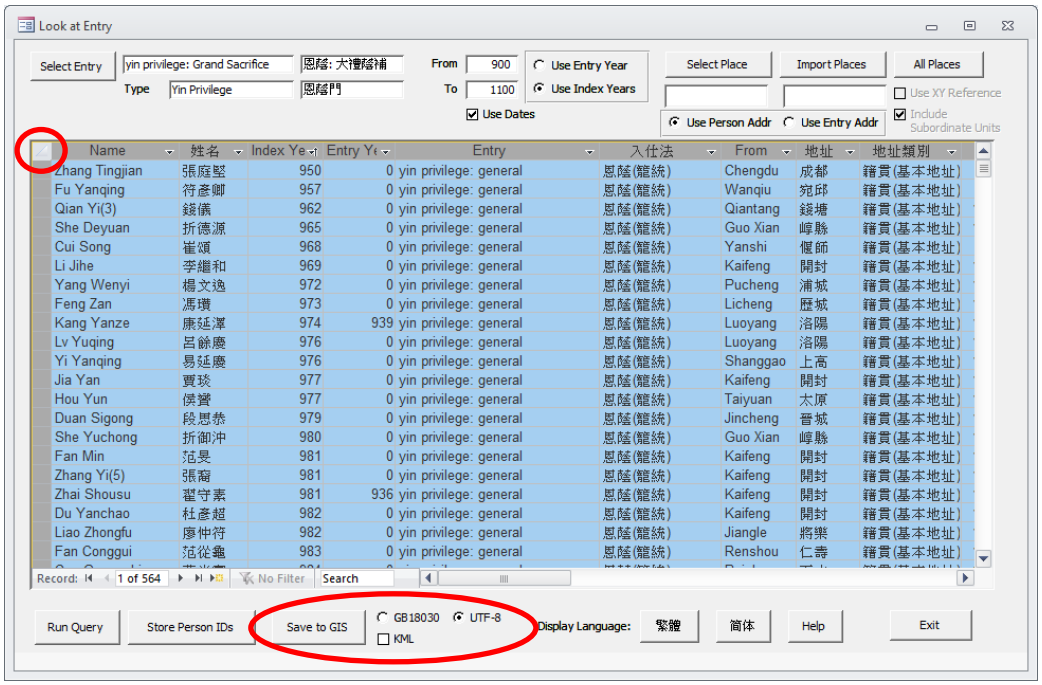
Name (Pinyin)	
Name (Chinese)	
Index Year	
Entry Year	
Description of Entry (English)	
Description of Entry (Chinese)	
Person's Index Place (Pinyin)	See discussion of index place on pages 22-23
Person's Index Place (Chinese)	
Type of Place Association	the type of place association used in assigning an 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 obviously is irrelevant for people who enter 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	
Parental Status (English)	The sources note whether the parents are alive.
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	

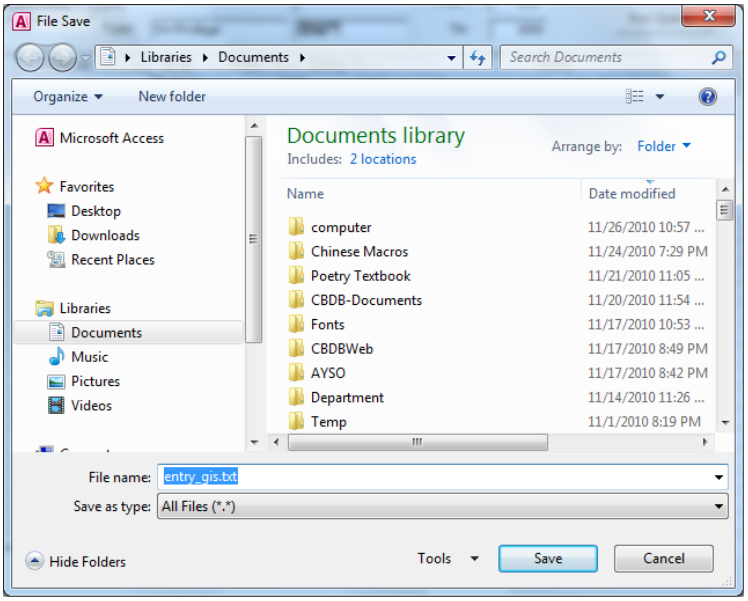
One can **sort** the table using any of these columns. For example, "Index Year" may be useful. *Left-Click* on the column name "Index Year" to select the column and then *Right-Click* to choose from the sorting options:



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**.

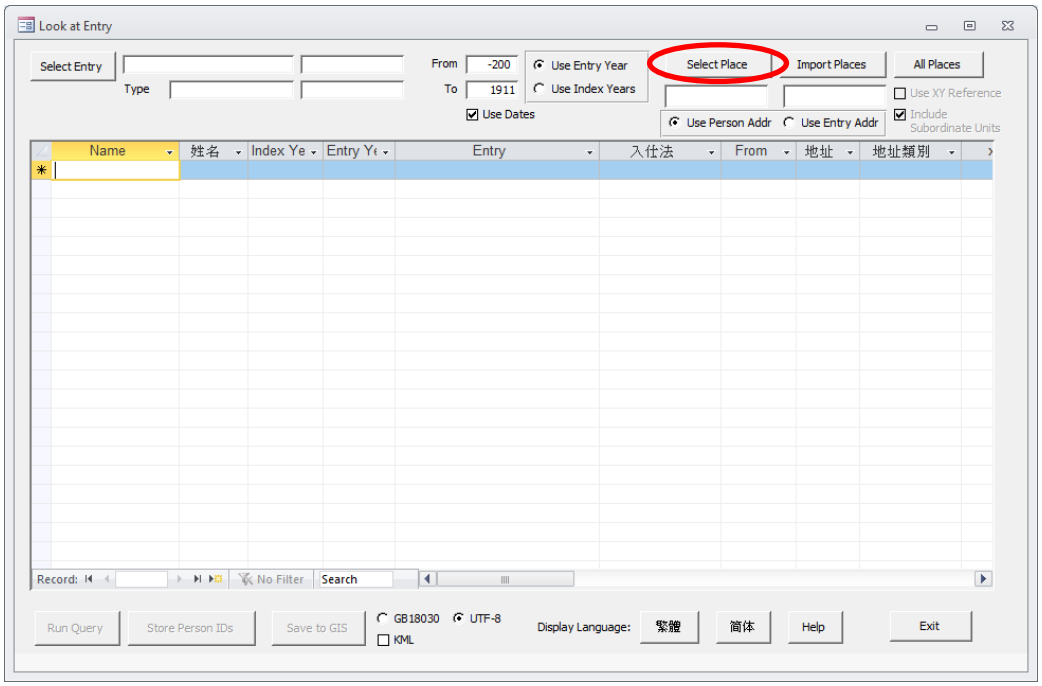


One then can paste the table into any program that accepts the format. If one wishes to save the results to a file readable by ArcGIS, one selects the coding for the file and clicks on the **Save to GIS** button at the bottom of the form. The table is saved to a text file, by default, "entry_gis.txt."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

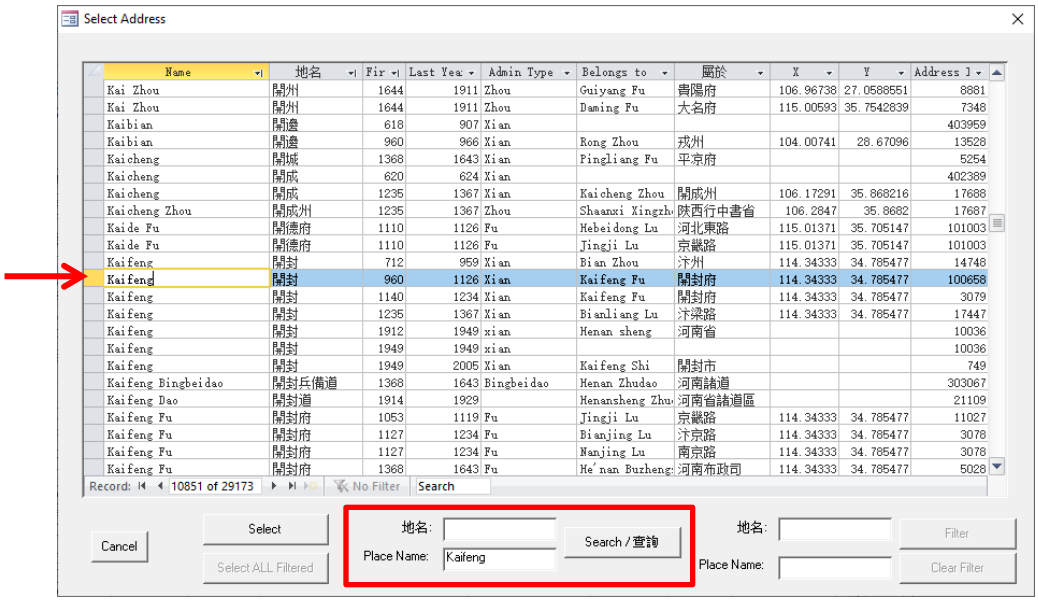


KML has become an important standard for GIS software, and the CBDB results also can be saved in KML format.

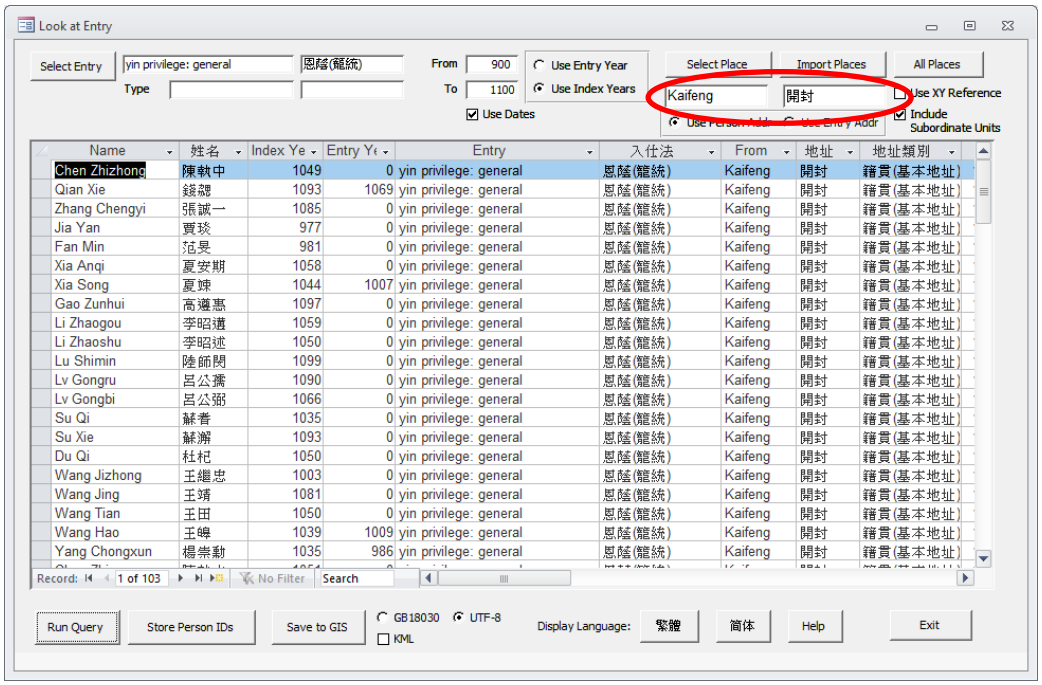
If one wishes to explore the mode of entry for people from a particular region, one uses the **Select Place** button in the upper right part of the form:



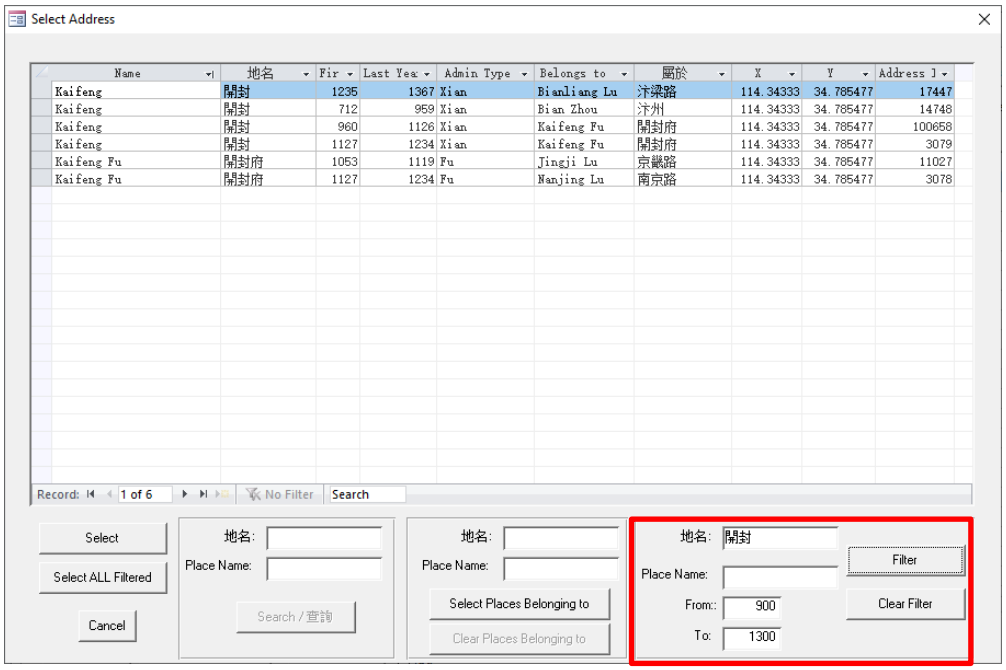
This opens the **Select Address** form. One can search for a place name using the search box:



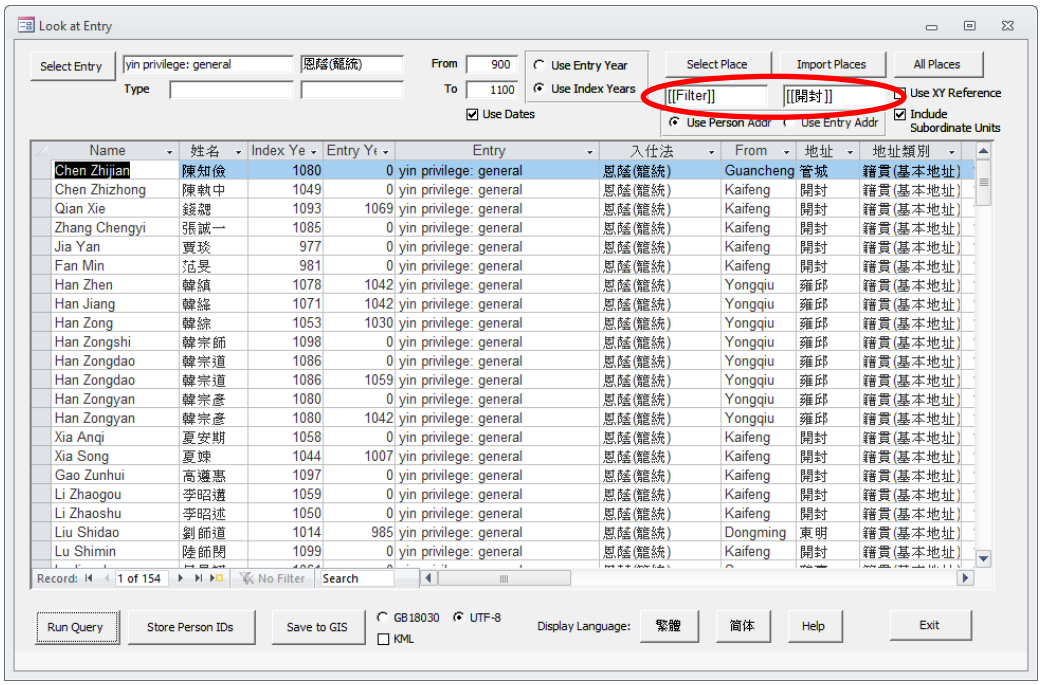
For example, if one searches for the address "Kaifeng", note that there are many addresses for Kaifeng. We will select the one active from 960 to 1126 by clicking on the corresponding row and clicking **Select** will return back to the main window, where we can run another query.



Using the single address for Kaifeng from 960 to 1126 produces 103 people. If one wishes to look at Kaifeng more broadly, return to the **Select Address** form and enter "Kaifeng" into the Filter text box and then click the **Filter** command button (on the right of the search box). 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.) To narrow the search yet more, one can limit the years to a range from 900 to 1300. Then click the **Select ALL Filtered** command button:



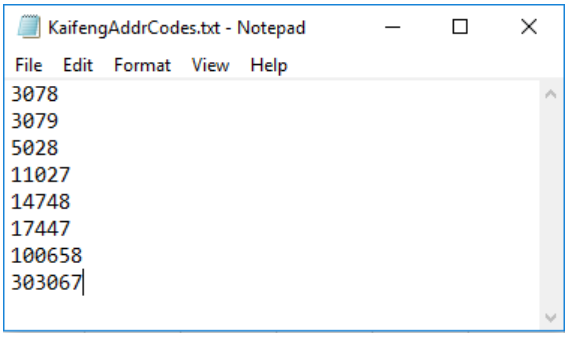
This will return you to the main LookAtEntry form, with all the Kaifeng codes selected:



The **Place** text fields will show the filter term in a pair of square brackets, here "[[Kaifeng]]." Running the query now produces 154 records for people from Kaifeng whose index years were between 900 and 1100 and who entered government service through the *yin* privilege.

If one wishes to limit the number of codes for Kaifeng even further, 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.

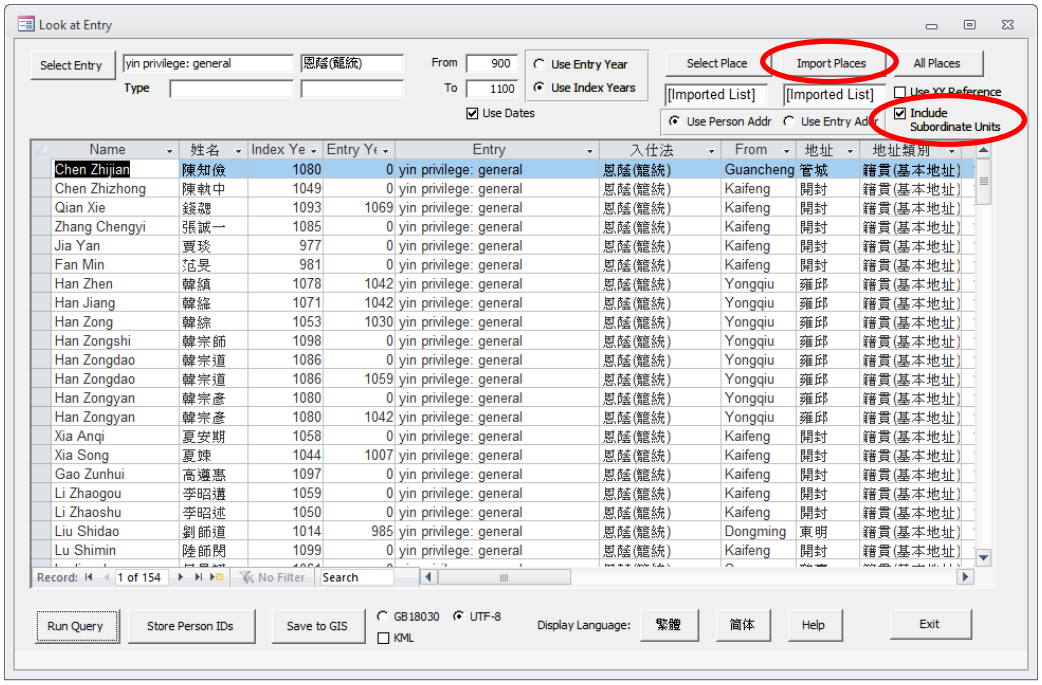
	A	B	C	D	E	F	G	H	I	J
1	Name	地名	First Year	Last Year	Admin Type	Belongs to	屬於	X	Y	Address ID
2	Kaifeng Fu	開封府	1127	1234	Fu	Nanjing Lu	南京路	114.34333	34.785477	3078
3	Kaifeng	開封	1127	1234	Xian	Kaifeng Fu	開封府	114.34333	34.785477	3079
4	Kaifeng Fu	開封府	1368	1643	Fu	He'nan Buzhengsi	河南布政司	114.34333	34.785477	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



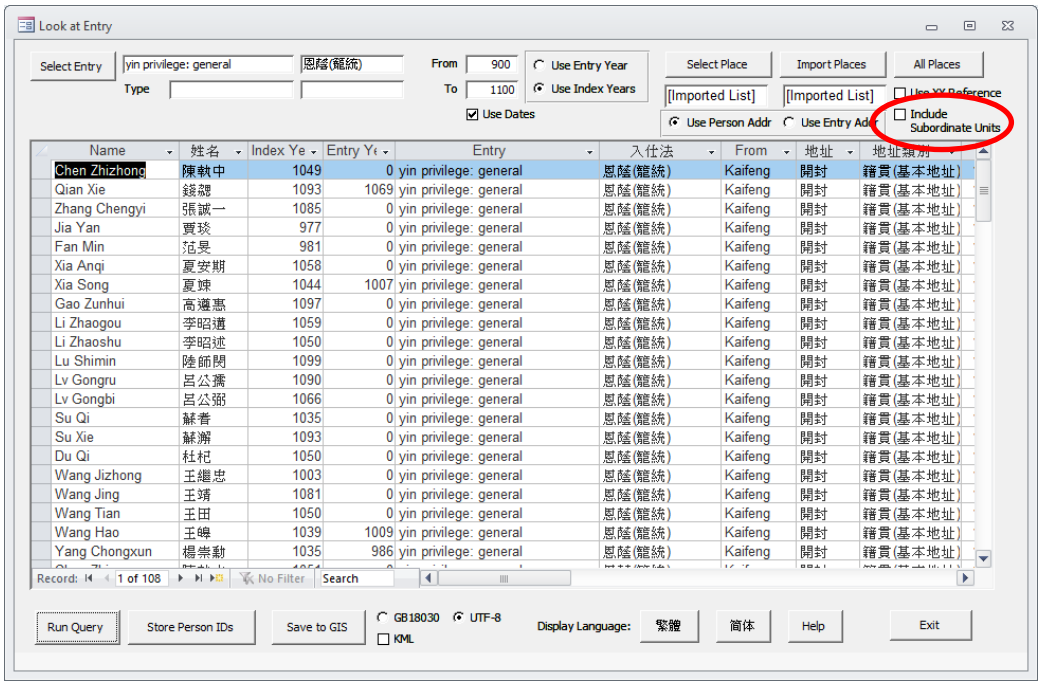
Please note that **this format is different from earlier versions of the interface.** Because scholars have been importing very long list of place ID and people IDs into the various analytic forms, we have rewritten the way in which the system imports IDs to make it much faster. However, it is now best to use a text file (in **ANSI** encoding) with just a column of IDs.

The new 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.)

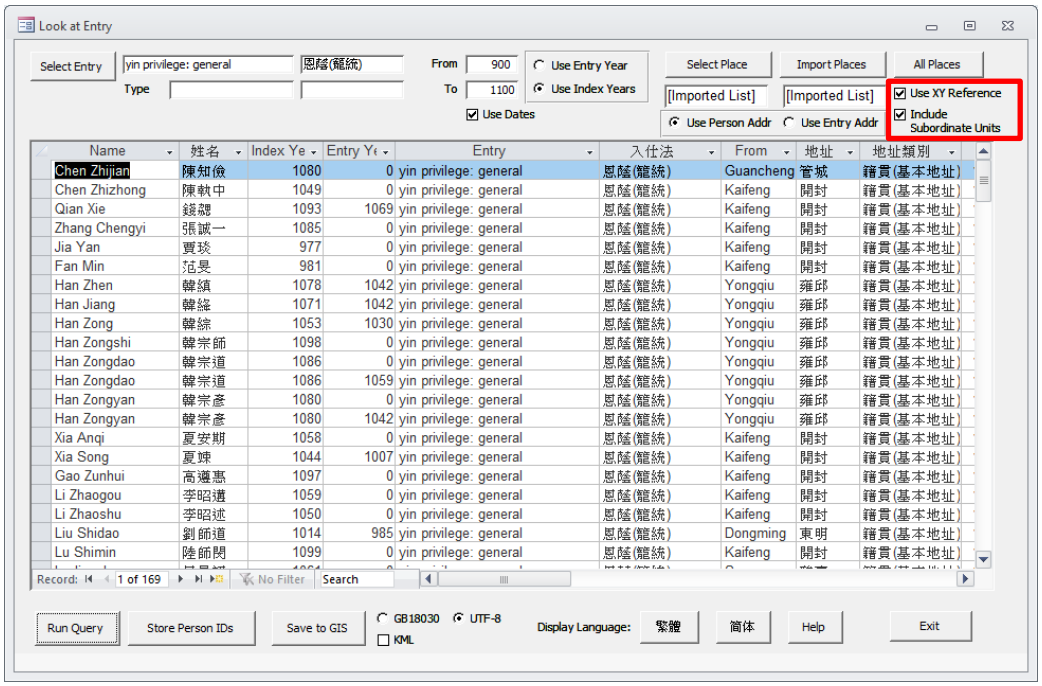
Now click on the **Import Places** button in the LookAtEntry form and select the file to be imported. (CBDB gives a warning when it reads the list of IDs and finds an invalid ID.) If the import has been successful, one will see "[Imported List]" in the Place Information text boxes. Once the list has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



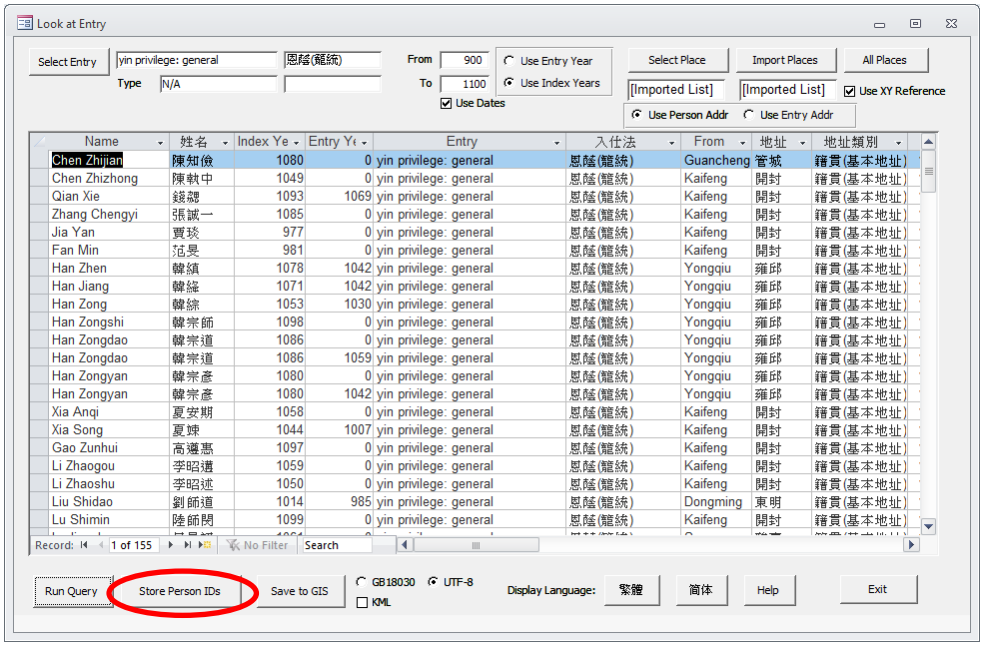
This approach produces 154 people, the same as the filtered version. However, note the **Include Subordinate Units** checkbox in the upper right corner. One of the places on the imported list was the Kaifeng Superior Prefecture 開封府: it has other counties subordinate to it that are included in the search when the checkbox is selected. This is the default setting. Note that the table includes people from Yongqiu 雍邱 and Dongming 東明: 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 108 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 169 people (a gain of 15).



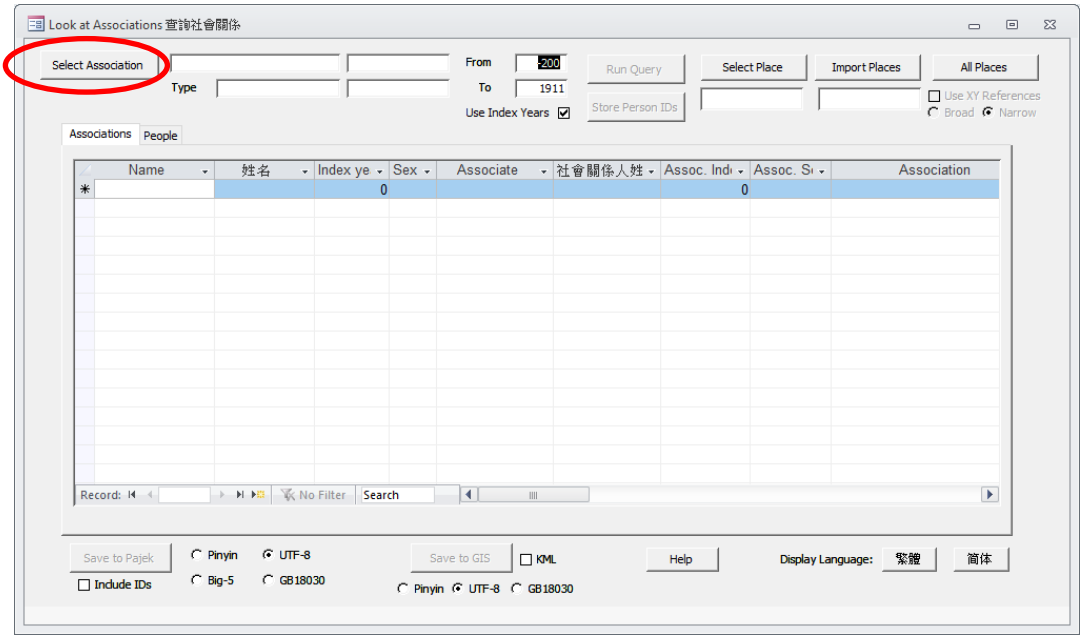
If one has created a query that produces a list of people who one wants to reuse in other queries—for example, if one wants to look at the kinship networks for the officials from Kaifeng who entered government service through *yin* privilege—one can store the list of people for reuse in the forms that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



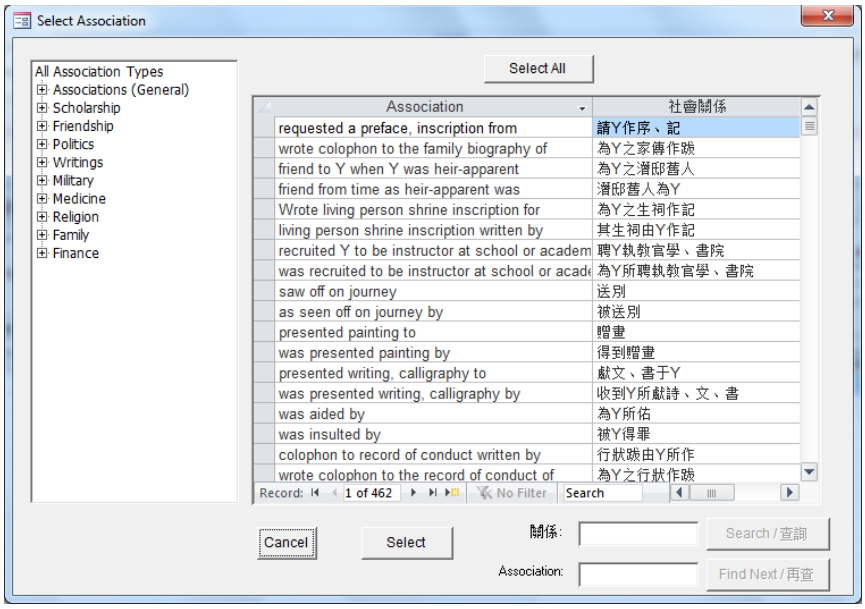
Simply click on the “Store Person IDs” button in this form and then click on the “Recall Person IDs” in the other form.

B. Using the Form LookAtAssociations

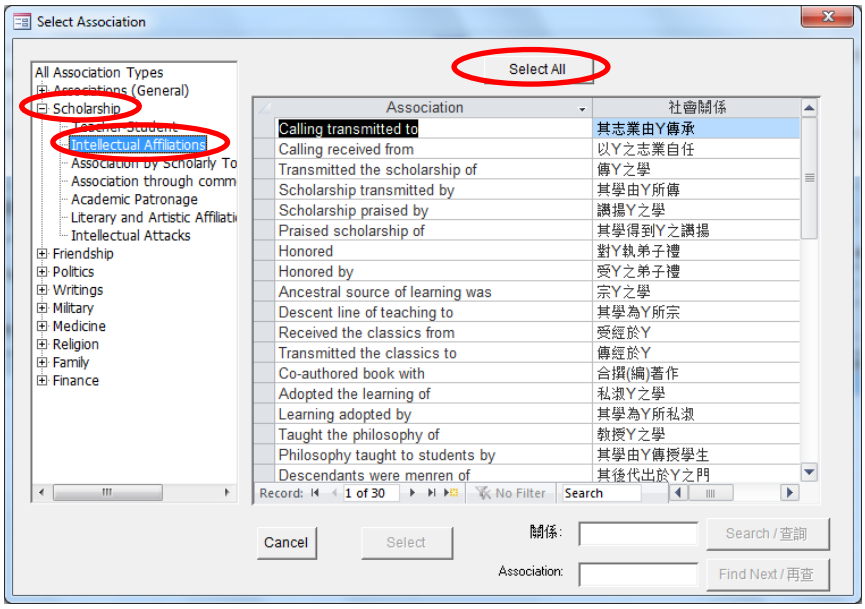
LookAtAssociations allows one to look at the people who have participated in particular associations or categories of associations. After opening the form, one clicks on "Select Association" to choose the type of association one wants to investigate.



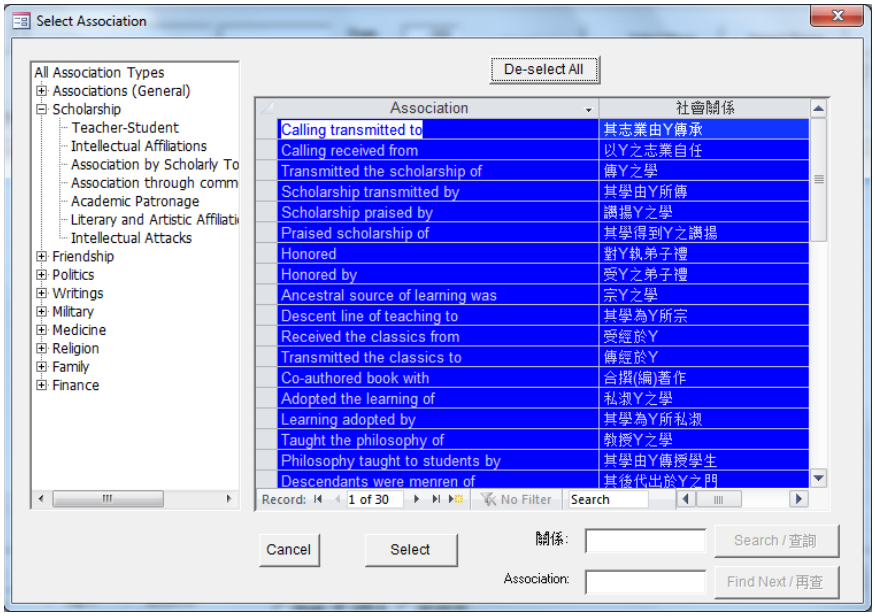
There are over four hundred categories of associations, so CBDB allows one to pick by type and subtype.



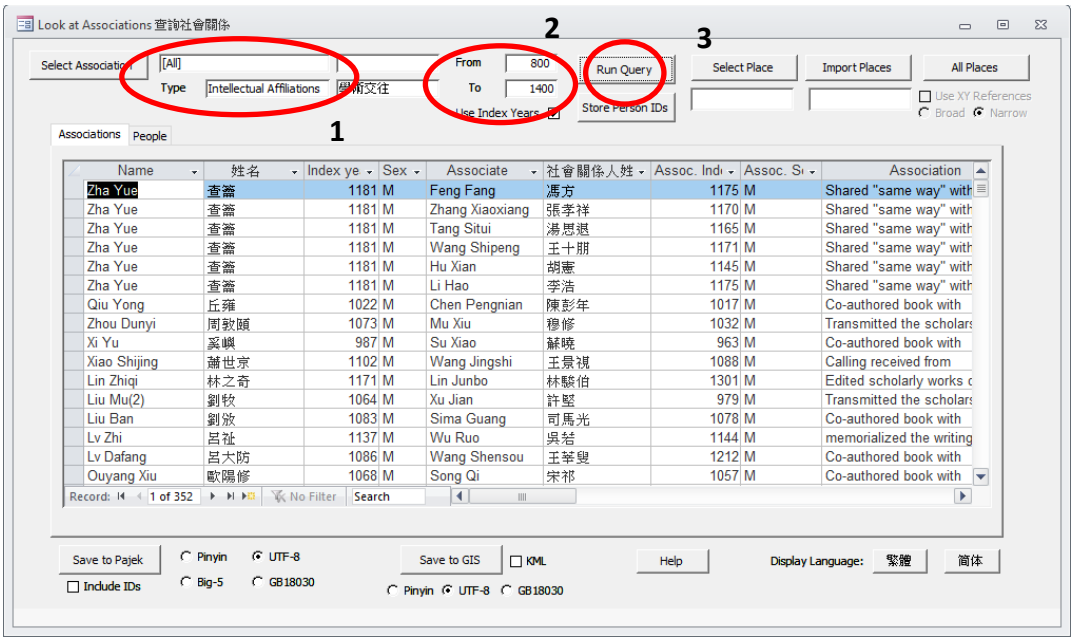
Consider the "Scholarship" associations from the list on the left. Under "Scholarship" there are seven subtypes. The subtype "intellectual affiliations" in turn has eight categories of associations:



One 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:

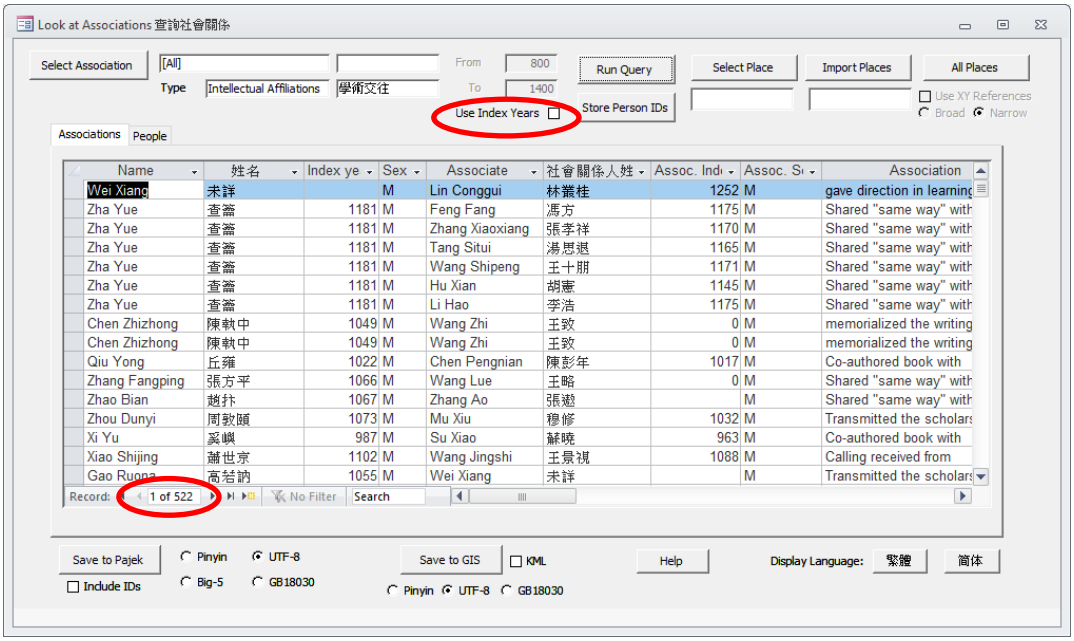


Note that, as with entry, one can search for terms in the table of associations in both English and Chinese (using the search box at the bottom right corner) and search again if the first item found is not what you are looking for. In the screenshot below, I have selected *all intellectual affiliation associations* (1) as the subtype of association. I then chose the year between 800 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 352 to 522. One can sort on index years after doing the search to look for the relevant associations.

The **Associations** table in **LookAtAssociations** has 28 columns to display the types of information recorded in the ASSOC_DATA table:

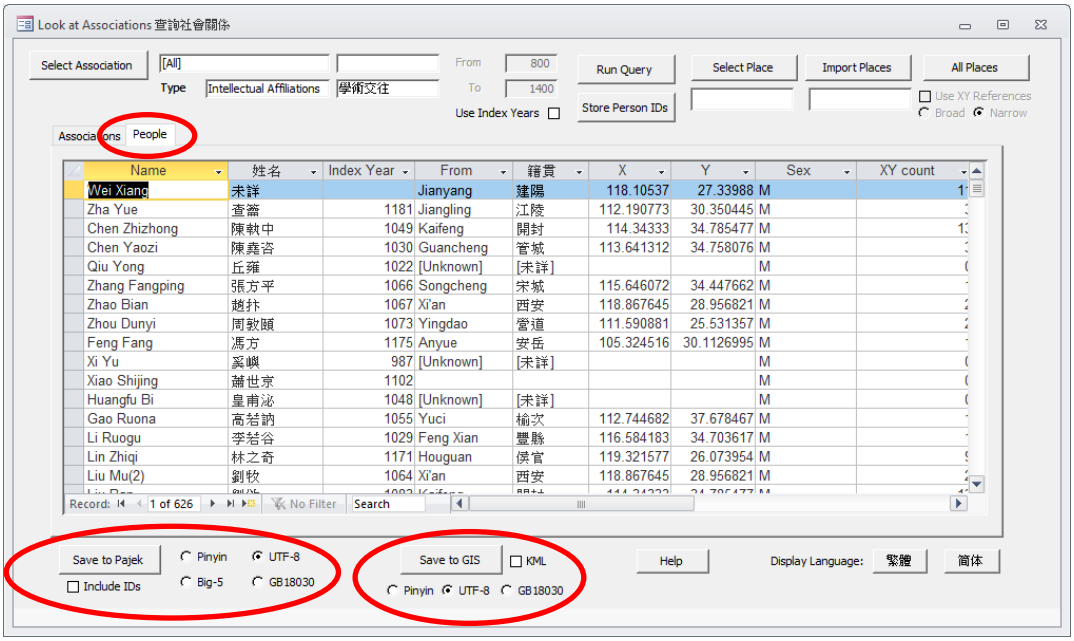
Name (Pinyin)	
Name (Chinese)	
Index Year	
Sex	
Associated Person's Name (Pinyin)	
Associated Person's Name (Chinese)	
Associated Person's Index Year	
Associated Person's Sex	
Association Category (English)	
Association Category (Chinese)	
Association Count	This gives the number of objects or events that established the association
Address (English)	This is the index address, if known. Otherwise, CBDB selects the first record of address information on the person.
Address (Chinese)	
X-coordinate	These are the coordinates for the centroid 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)	
Distance	If CBDB has the coordinates for the place identification for both people, it calculates the great-circle arc distance between them (in kilometers).

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 ten columns:

Name (pinyin)
Name (Chinese)
Index Year
Sex
Place Association (pinyin)
Place Association (Chinese)
X-coordinate

Y-coordinate
XY-count
Person ID



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 by clicking on the **Save to Pajek** button. The Pajek format is one standard format for visualization in social network analysis (SNA), 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.

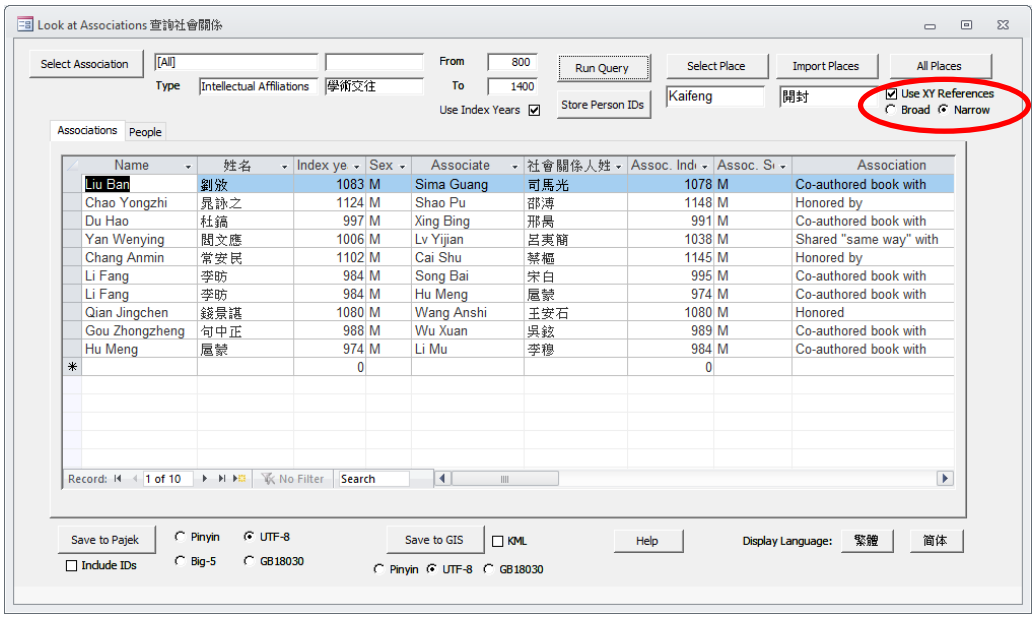
The default display for both nodes and edges uses color-coding to indicate degree of distance from the target person or people:

White = the target nodes;
Blue = nodes directly connected to them
Green = node distance of 2
Orange = node distance of 3
Yellow = node distance of 4
Red = node distance of 5 or more

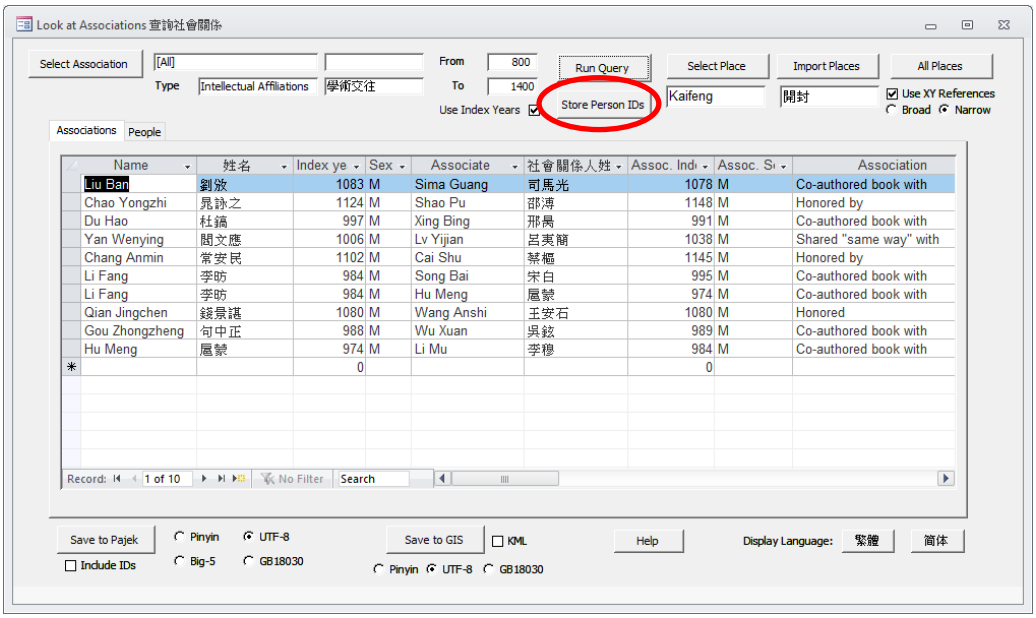
Like the **LookAtEntry** form, **LookAtAssociations** allows one to look at associations for people from a particular place or from a particular list of places.

However, **LookAtAssociations** has an additional 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). 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:”

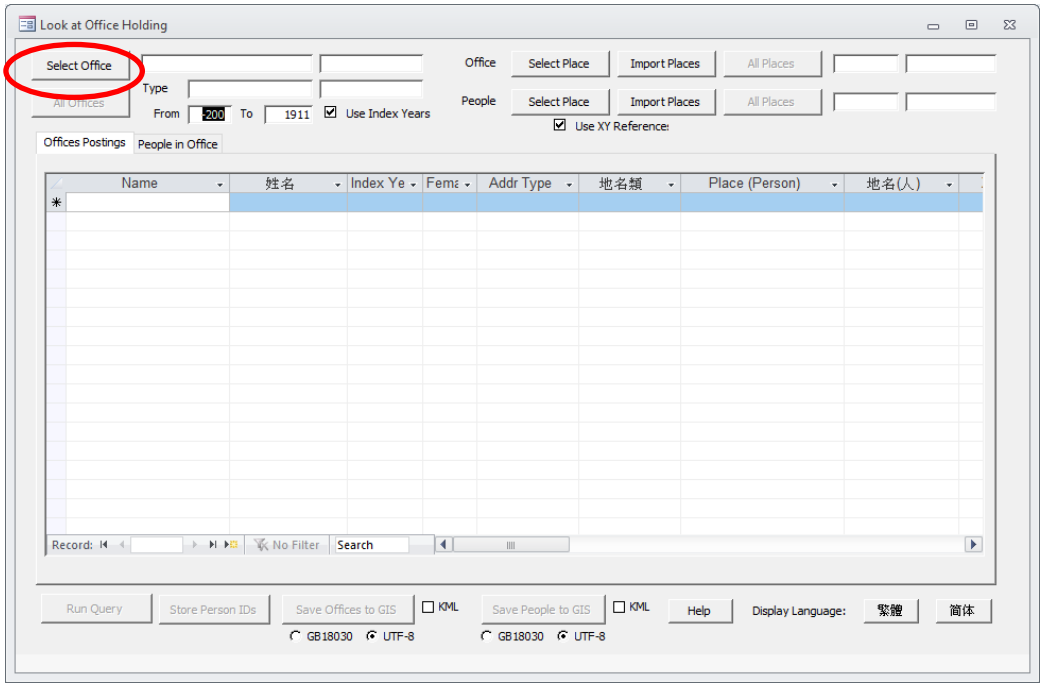


Like all other tables, LookAt Associations allows one to store the results of a query for later use in another form. One clicks on the “Store Person IDs” button.



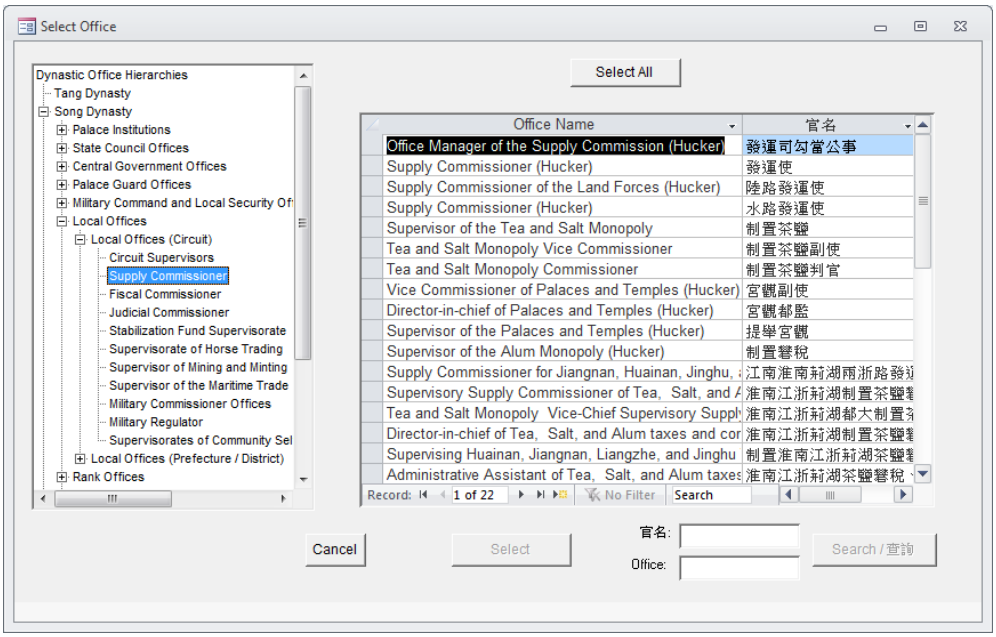
C. Using the Form LookAtOffices

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.

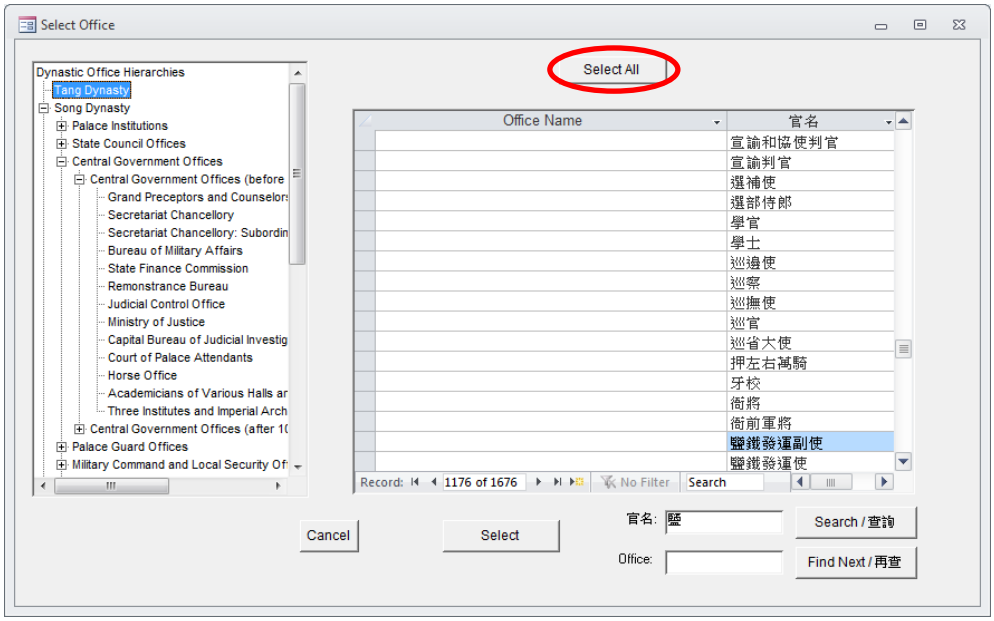


The "Select Office" form displays a tree of office categories. The first node, "Bureaucratic Structure" shows the organizational hierarchy organized by dynasty. One can view—and select—the offices at any level of structure. The first form shows all the offices associated with the Supply Commissioner 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 Song, searching by the Chinese term (most records do not yet have English equivalents) is the best way to find particular offices in other dynasties. Searching for "salt" 鹽 (see the second form below) eventually leads one to Tang dynasty offices. (Because the Song offices were added first, they are the first offices found in the search routine.) One can determine which node on the tree is current by jiggling one of the scroll bars (not a great design, but it works):

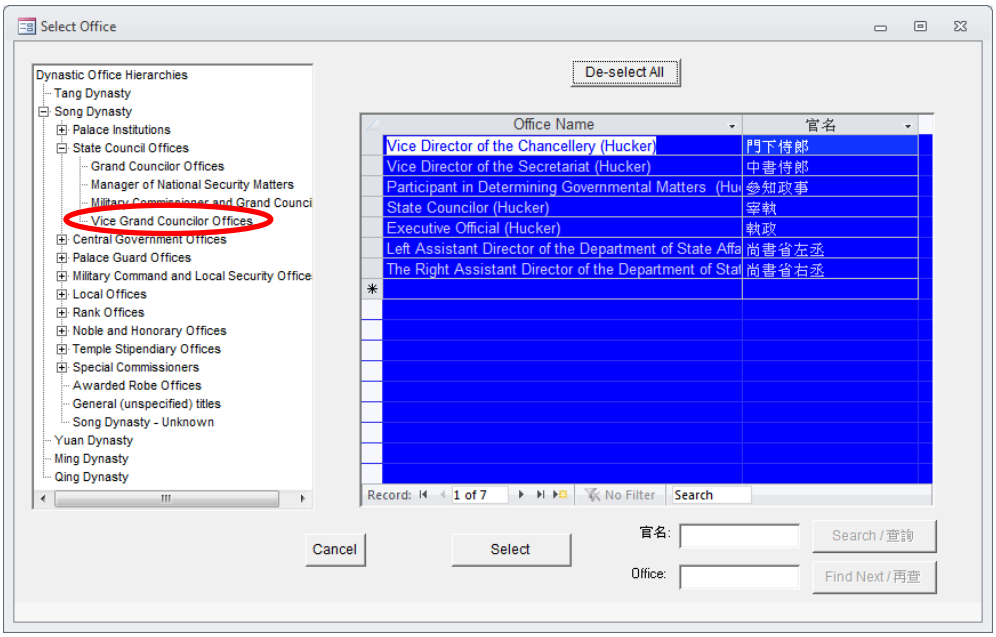


Offices in the Supply Commissioner's Office in the Song Dynasty

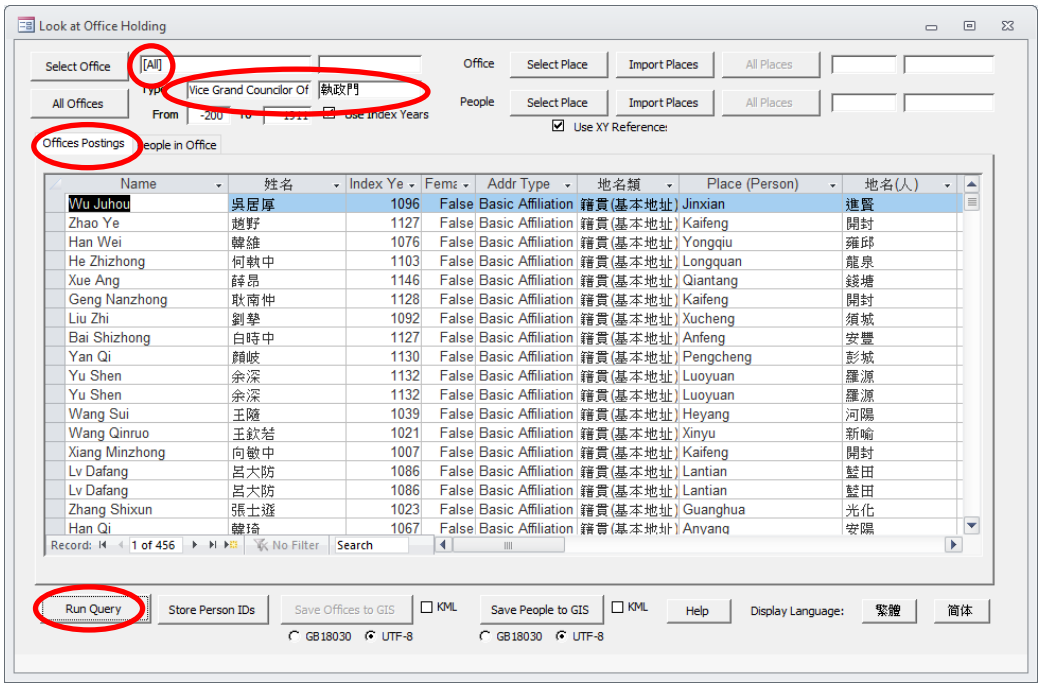


A Tang office containing the character 鹽 ("salt")

If one looks for the word "Chancellery," the *second* record brings one to the Vice Grand Councilor's Office in the Song dynasty. One then can select all of the records for offices in the Vice Grand Councilor's Office by clicking on "Select All:"

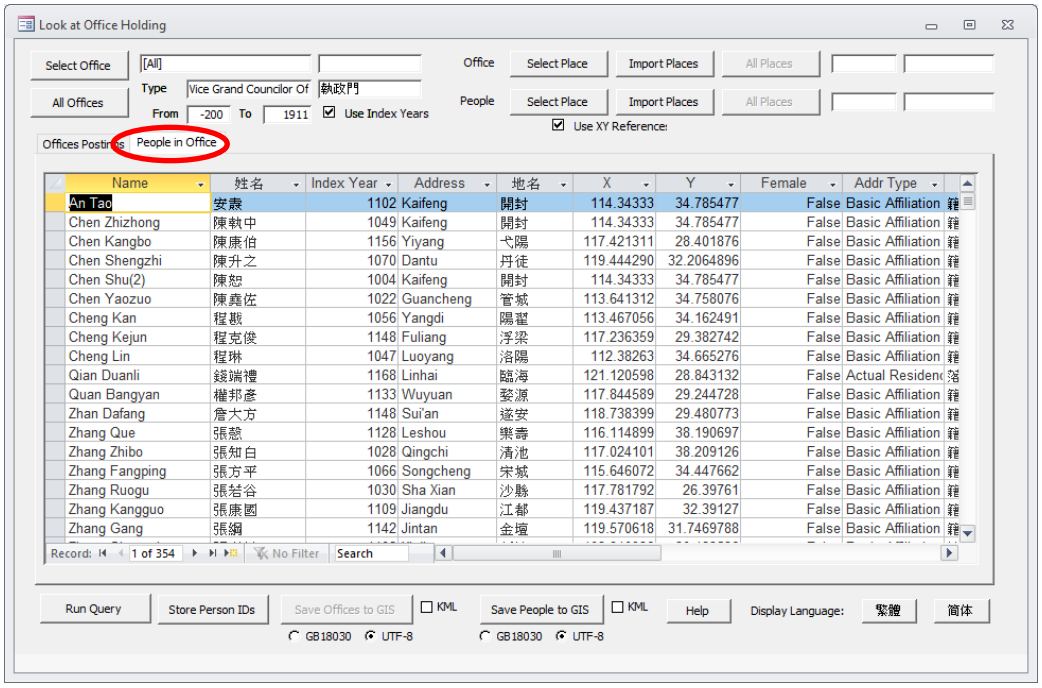


The form below shows a query selecting all postings to offices associated with the Vice Grand Councilor in CBDB, beginning in 200 B.C. and ending in 1911 and does not use index years. Because at present office codes are tied to dynasty, in fact all the results are from the Song:



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.



The table "Office Postings" has 28 fields:

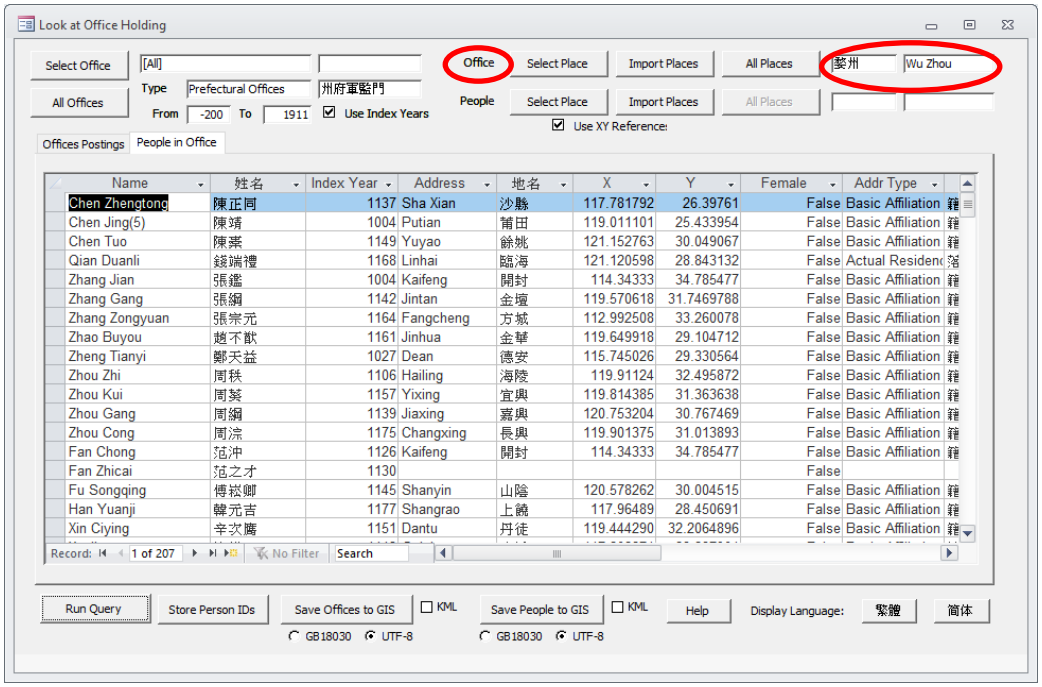
Person Name (pinyin)
Person Name (Chinese)
Index Year
Female (True or False)
Person Index Address Type (English)
Person Index Address Type (Chinese)
Person Index Address (pinyin)
Person Index Address (Chinese)
X coordinate of Person Index Address
Y coordinate of Person Index Address
Office (translation)
Office (Chinese)
First year of appointment
Last year of appointment
Dynasty (useful in cases where the years are very uncertain)
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

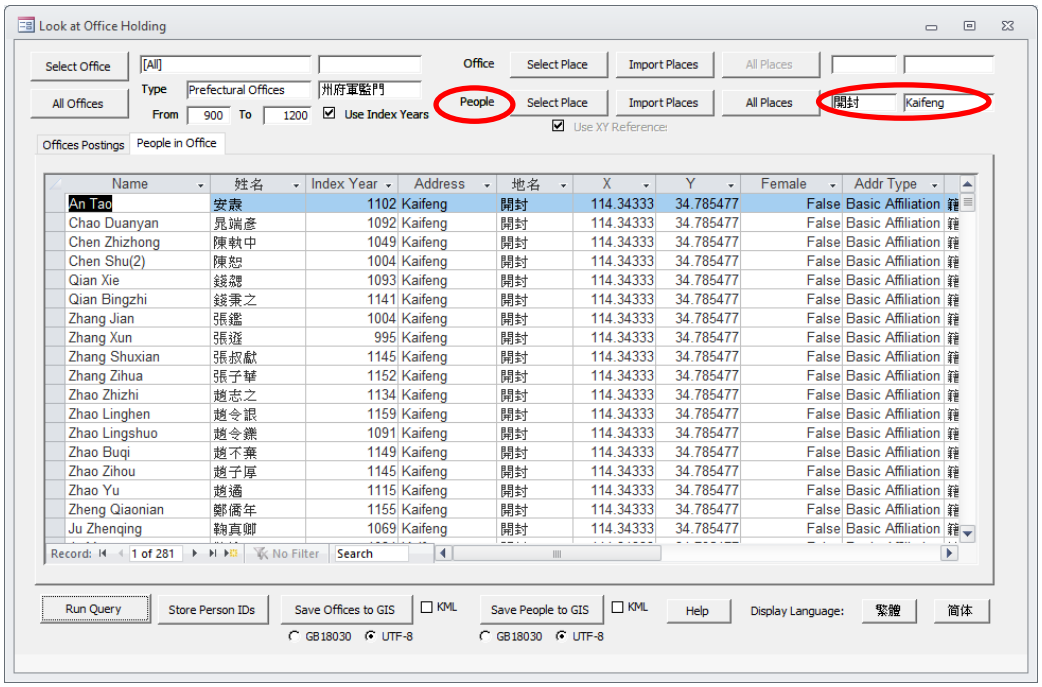
The table "People in Office" contains the usual information about people:

Person ID
Person Name (pinyin)
Person Name (Chinese)
Index Year
Sex
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

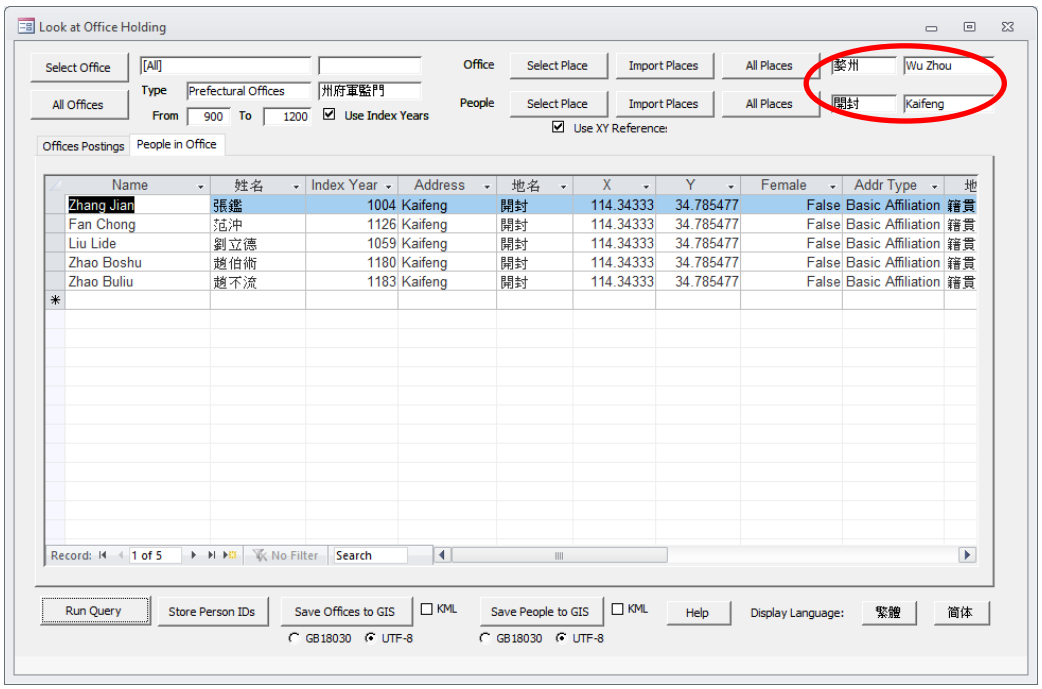
If one wishes to look at people who held office at a particular place or places, the form allows the user to select a place through the procedures discussed above. One can select a single place, use a filter for name, or import a list of address IDs. Then one runs the query in the usual way. Below is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



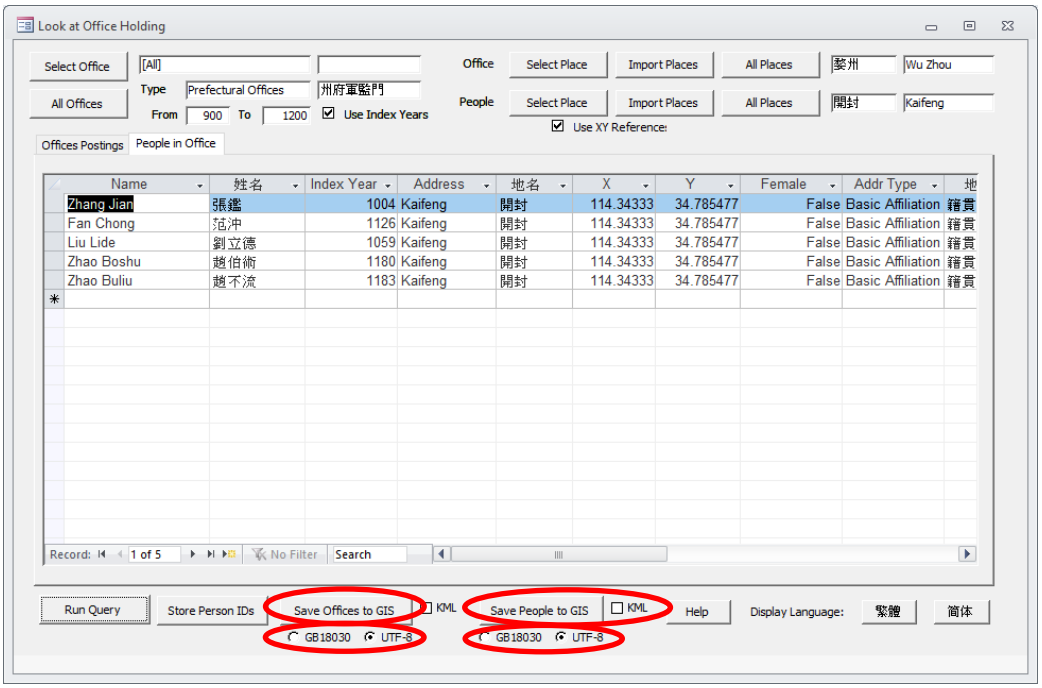
One can also explore where people from a particular place (or list of places) held particular types of office. Below is a query about where people from Kaifeng held prefectural office during the Song dynasty.



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



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."

D. Using the Form LookAtKinship

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:

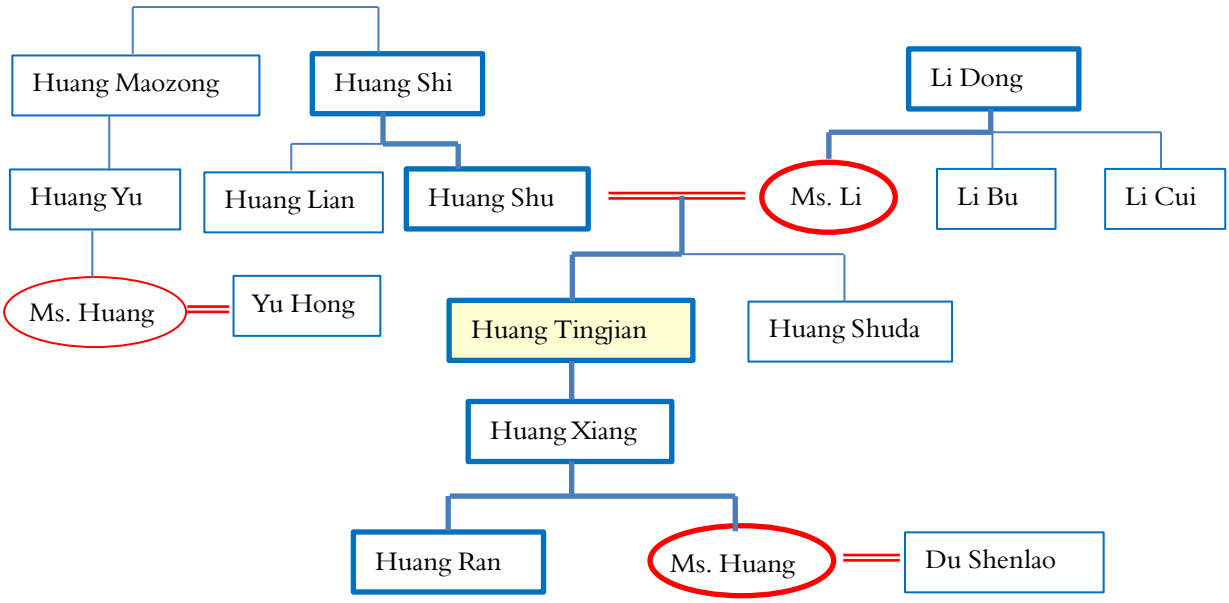
Max. Ancestor Gen. specifies how many generations of ancestors to include. One's father's generation is 1; the grandfather is 2, great-grandfather 3, and so on.

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:

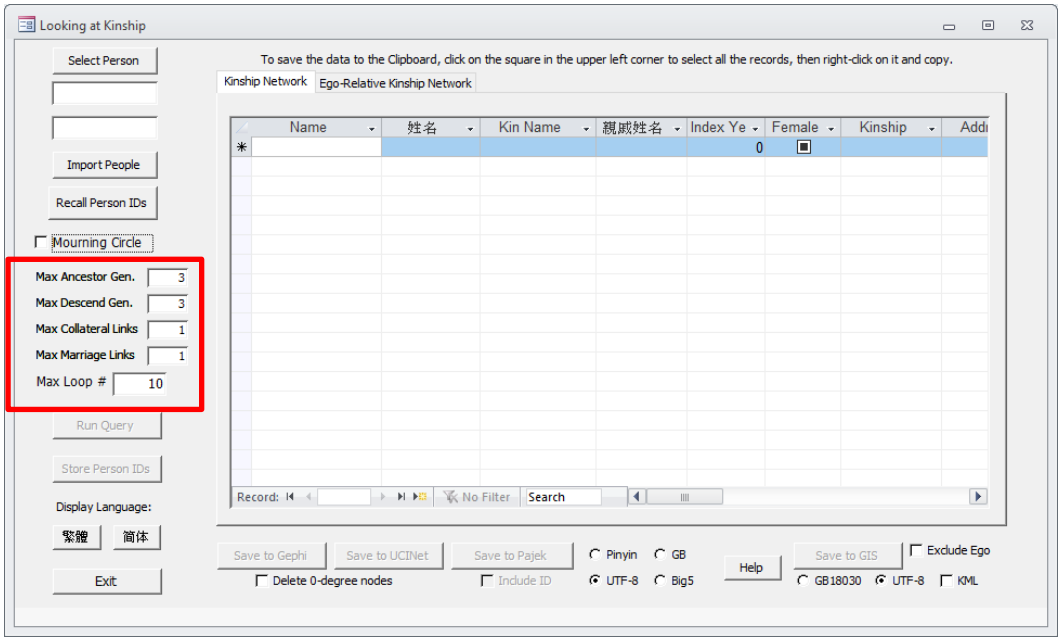


For Huang Tingjian, the squares and ovals with thick lines show direct lineal descent (fathers and mothers, sons and daughters). The double lines are marriage links. All other single lines mark collateral relations. In the measurement system used in **LookAtKinship**:

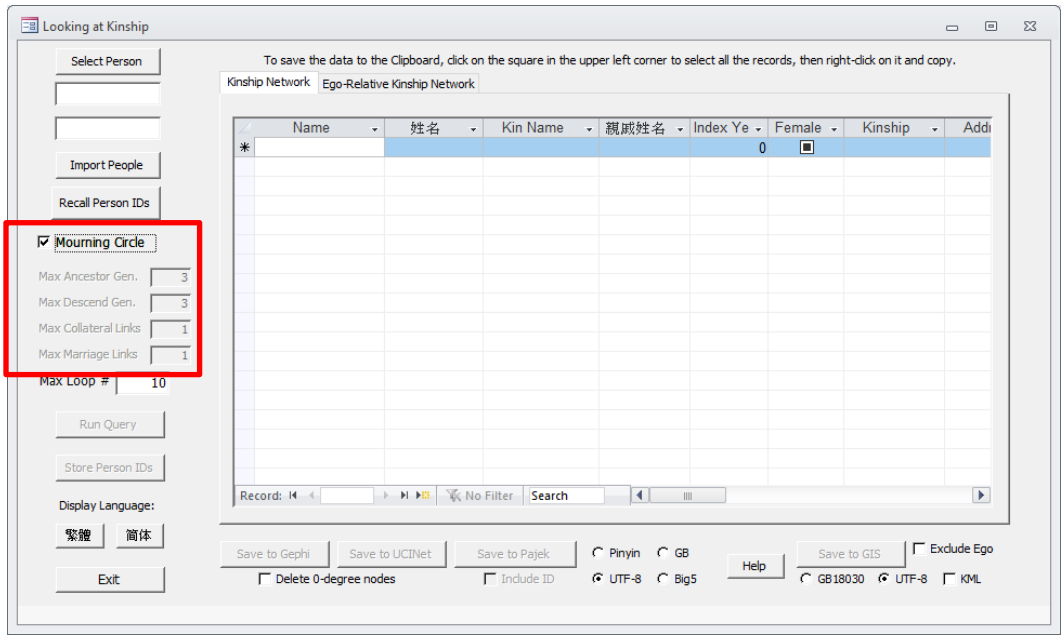
Huang Yu 黃育	is FFBS	(Up = 2, Down = 1, Collateral = 1)
Yu Hong 余宏	is FFBSDH	(Up = 2, Down = 2, Collateral = 1, Marr. = 1)
Li Cui 李萃	is MB	(Up = 1, Collateral = 1, Marr. = 1)
Du Shenlao 杜莘老	is SDH	(Down = 2, Marr. = 1)

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:

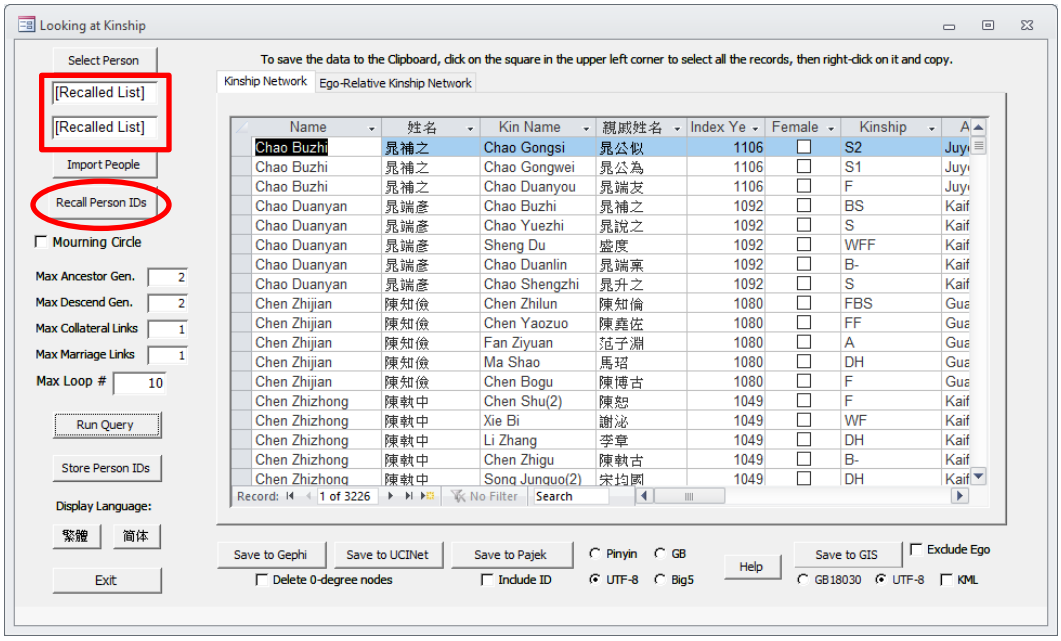
WARNING: searching for extended degrees of collateral and marriage distance may result in a very large dataset



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.



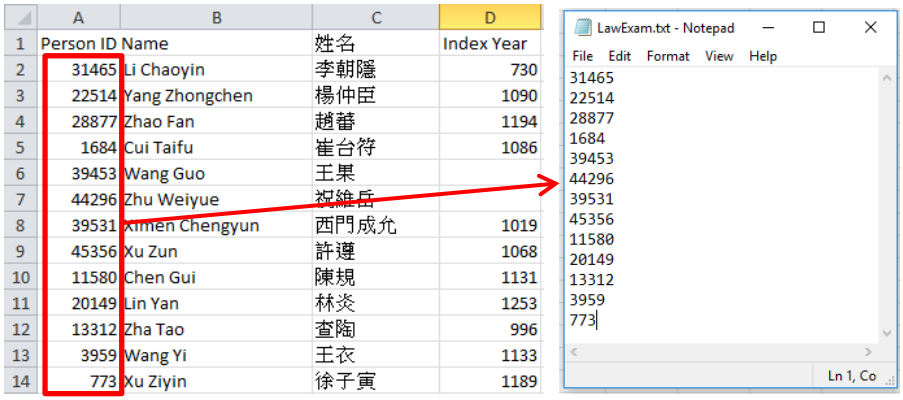
To examine kinship relations, one first selects the person or group of people whose kinship networks one seek to explore. There are **three** different ways to select people. First is to **recall a group of people** stored in the database as the result of an earlier query (see LookAtEntry for storing the list). If the "Recall Person IDs" button is enabled, this means that there is a group of IDs created earlier that can be used now.



When one clicks on "Recall Person IDs," the form loads the list and displays "[Recalled List]" in the box for the selected person's name. One then sets the desired parameters and runs the query.

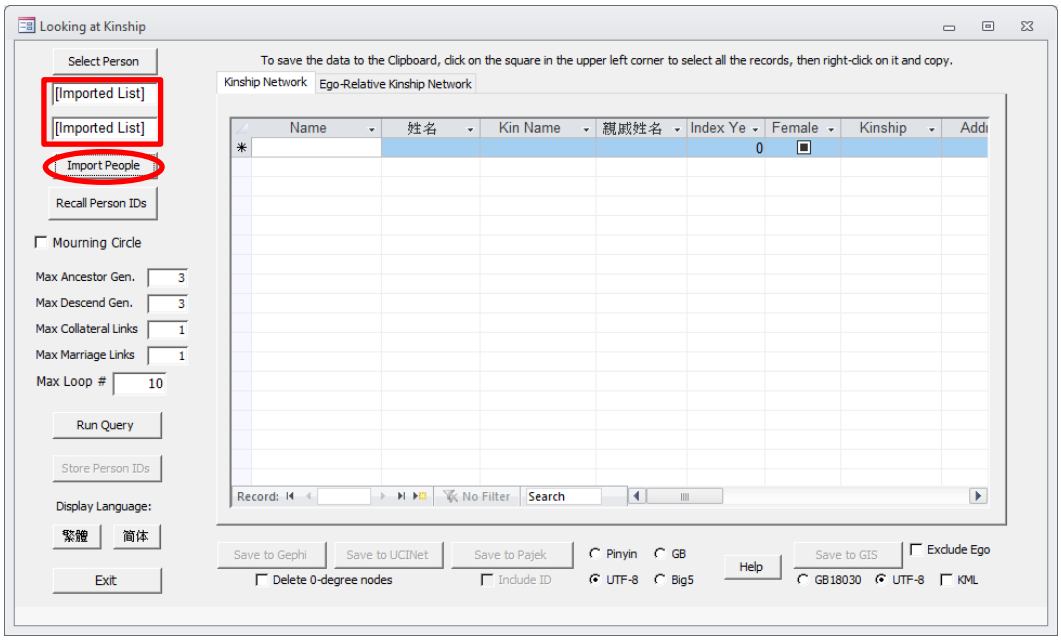
The second approach is through **importing** a list of people sharing common characteristics identified by other queries. For example, one could start with people in the Song dynasty who became eligible for office through a legal examination. One copies the results of the **LookAtEntry** query to a **Word**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



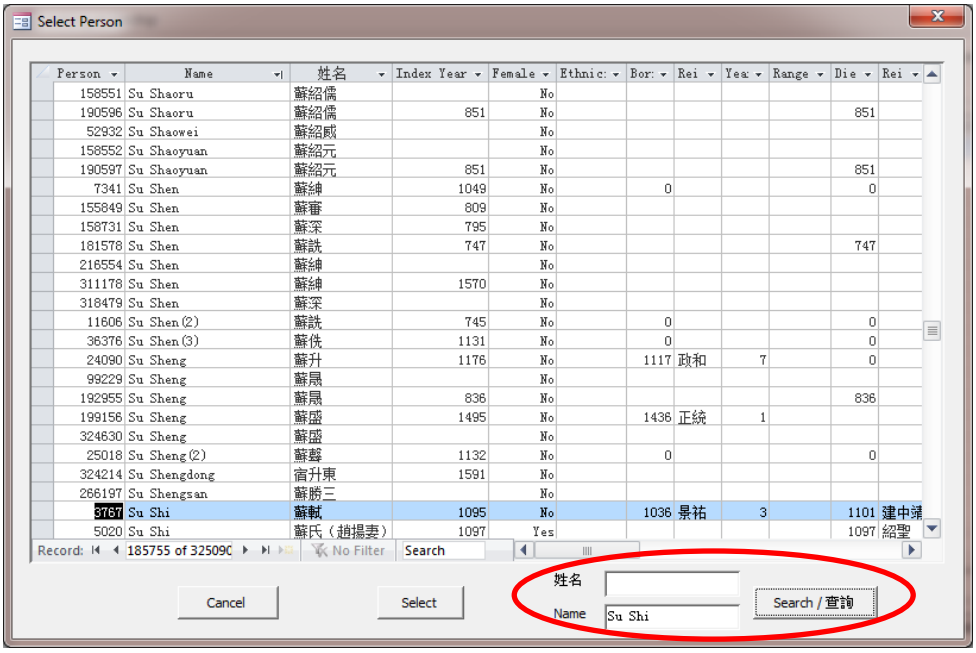
	A	B	C	D
1	Person ID	Name	姓名	Index Year
2	31465	Li Chaoyin	李朝隱	730
3	22514	Yang Zhongchen	楊仲臣	1090
4	28877	Zhao Fan	趙蕃	1194
5	1684	Cui Taifu	崔台符	1086
6	39453	Wang Guo	王果	
7	44296	Zhu Weiyue	祝維岳	
8	39531	Ximen Chengyun	西門成允	1019
9	45356	Xu Zun	許遵	1068
10	11580	Chen Gui	陳規	1131
11	20149	Lin Yan	林炎	1253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



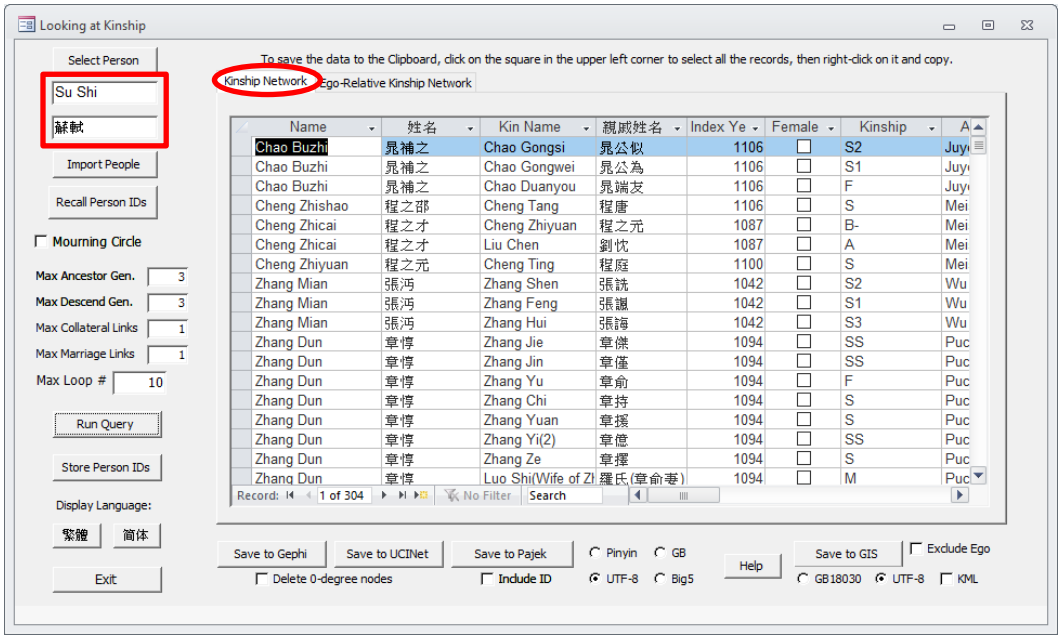
One then sets the desired parameters and runs the query.

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.

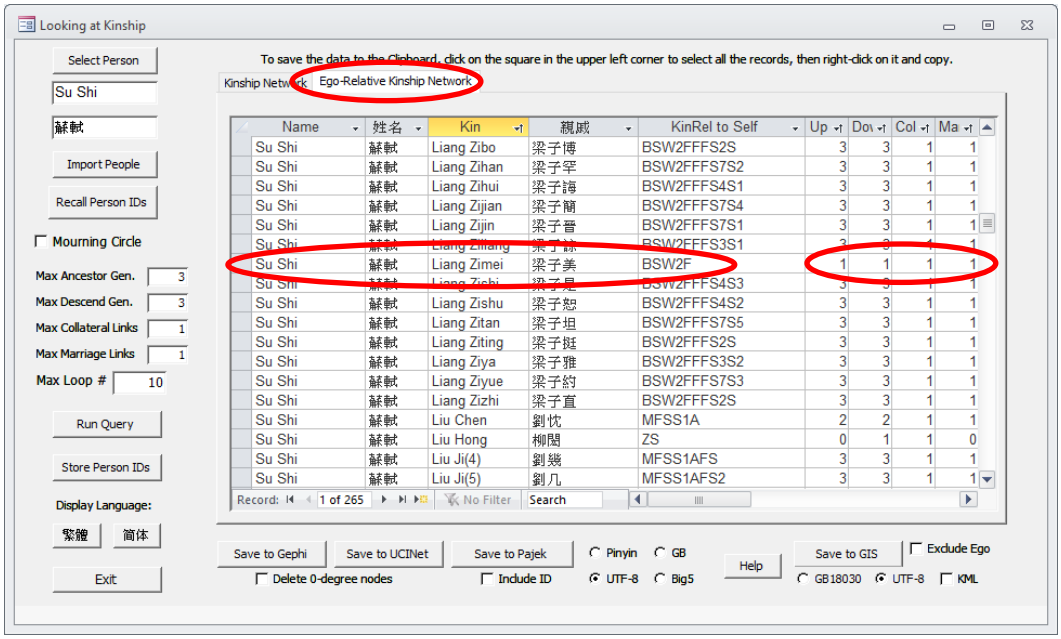
When the search finishes, there are two tables one can examine. The first, **Kinship Network**, lists all the kinship relations discovered through the search:



This table has 23 columns:

Name (pinyin)
Name (Chinese)
Kin Name (pinyin)
Kin Name (Chinese)
Index Year of Kin
Sex of Kin
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
Index Address Type (Chinese)
Address Type of Kin Index Address
Address Type of Kin Index Address (Chinese)
Distance (great-circle arc distance in kilometers between the addresses)
Person ID
Kin ID

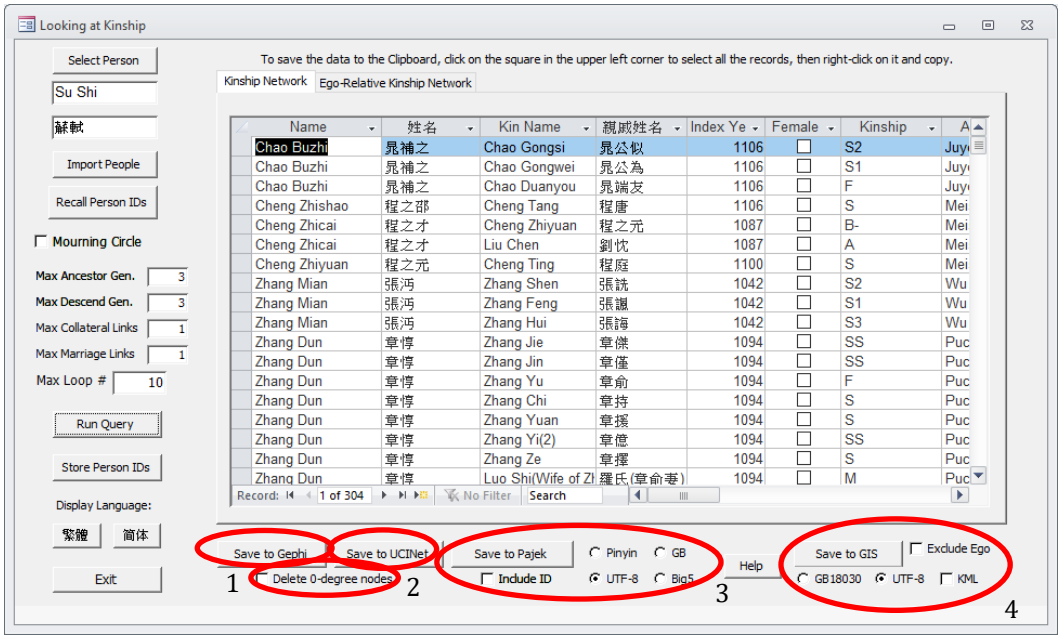
The second table, **Ego-Relative Kinship**, describes the kinship relation between each person in the first table and the person selected at the very beginning:



For example, Liang Zimei 梁子美 is Su Shi's younger brother's son's wife's father. The path one traverses to reach Liang Zimei requires one to go over one collateral link (B), down one generational link (S), over one marriage link (W2), and finally up one generation (F). Thus the kinship distance metrics are {1,1,1,1}. (The "Ego-Relative Kinship" table has an additional column that gives a raw path that shows how CBDB simplified the kinship relations, but since this task proves to beyond any simple algorithm, CBDB has disabled the function.)

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: **Gephi** (1), **UCINet** (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 for **ArcGIS** (4) or in **KML** format with two different code options. 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 count of the number of people associated with a particular set of coordinates (the **xy_count**), and it may prove useful to filter those people out of the GIS data and focus only on their kin.



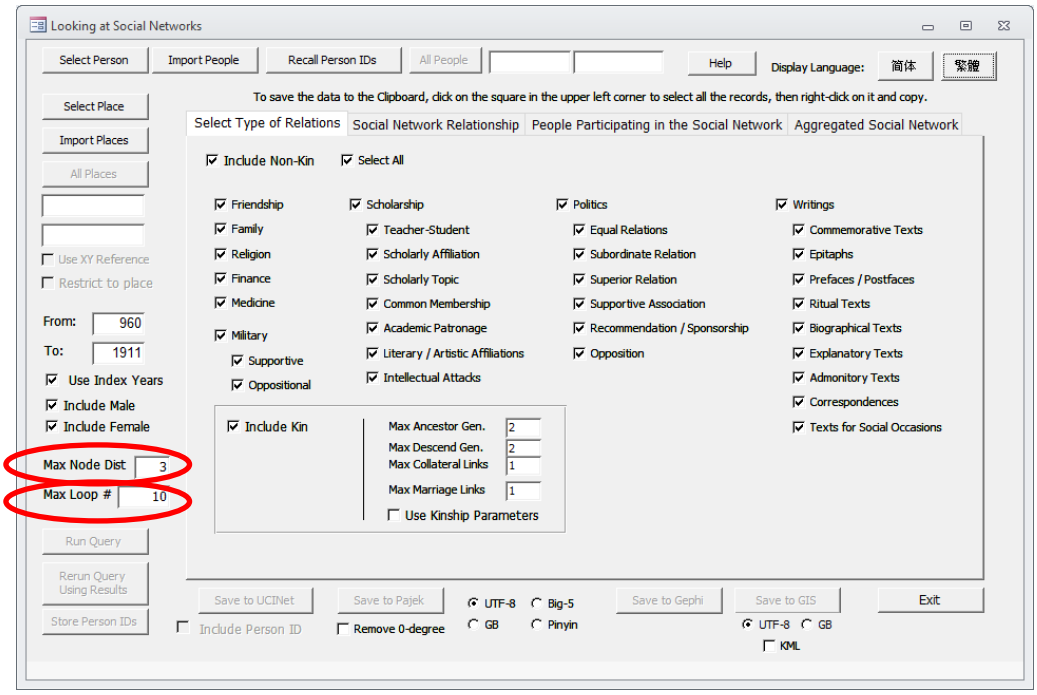
The default display for both nodes and edges in the SNA output files uses color-coding to indicate degree of distance from the target person:

White = 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

E. Using the Form LookAtNetworks

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 network is called an ego network. 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.

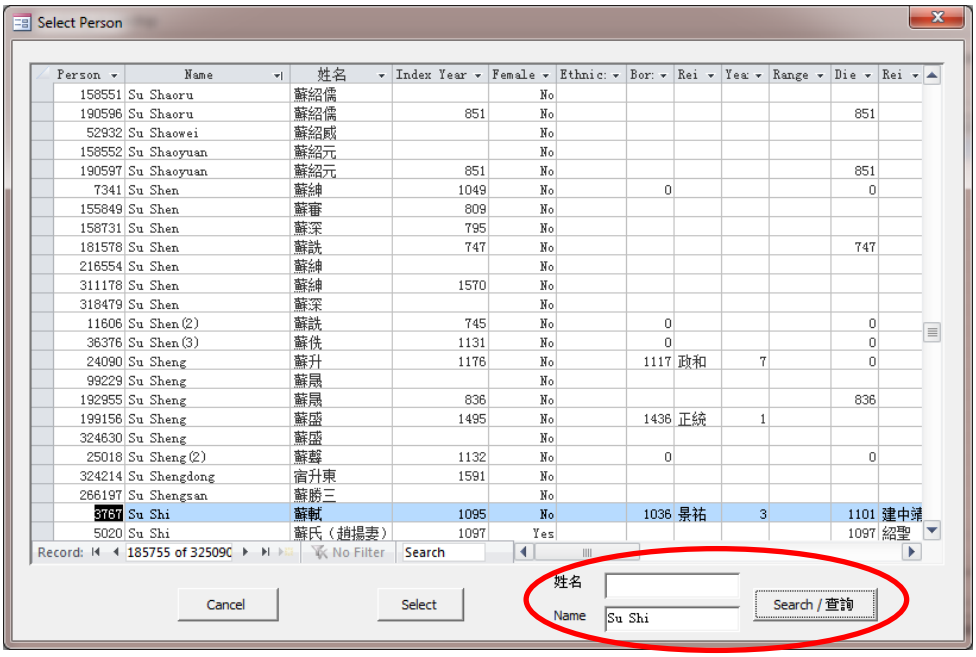
Basic Query Functions

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:



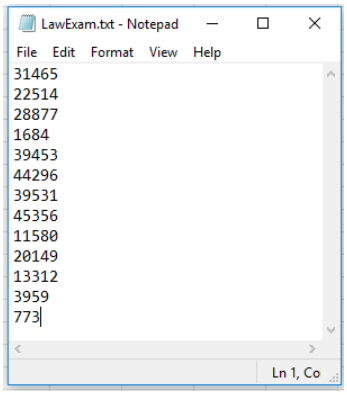
Since the list contains all the people in the database, one can search for a particular person using the search box in the lower right corner.

B. Import a List of People

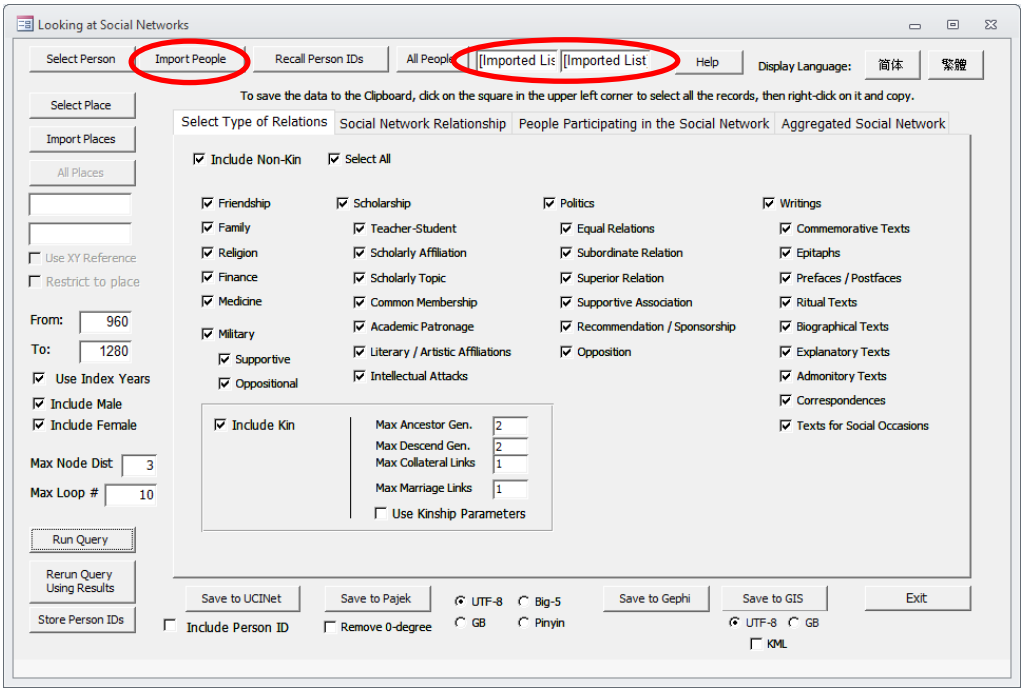
A second, very useful way to consider social networks is to import a group of people sharing common characteristics identified by other queries. For example, one could start with people in the Song dynasty who became eligible for office through a legal examination. One copies the results of the **LookAtEntry**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



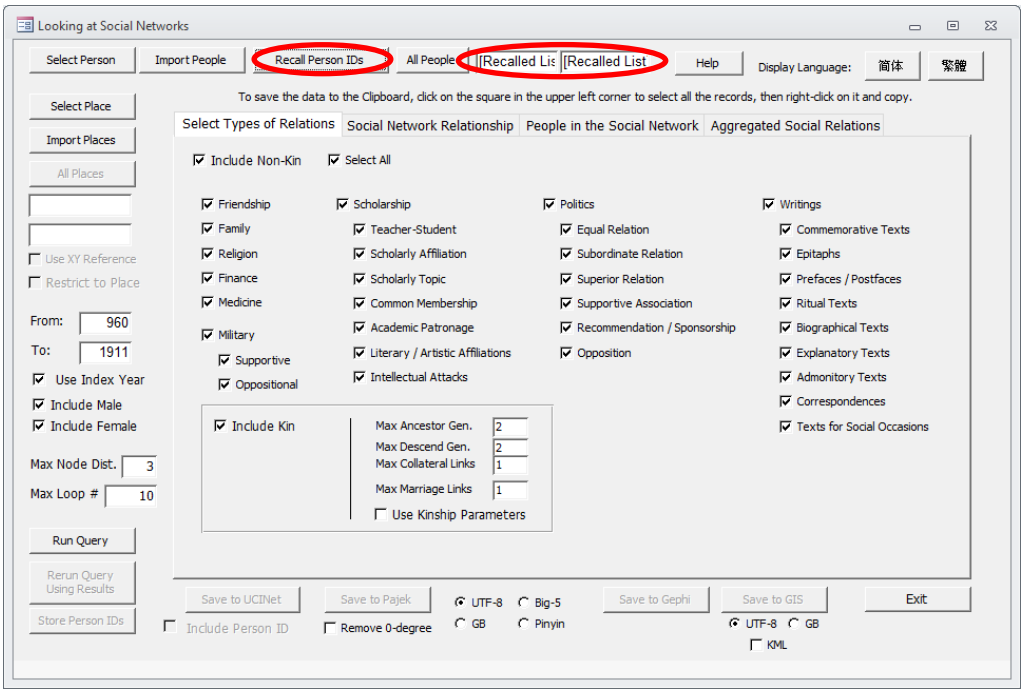
After one clicks on the **Import People** command button, selects the file, and **LookAtNetworks** successfully reads the file, the form will look like:



The two boxes that give the person's name in pinyin and Chinese will state "[Imported List]."

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

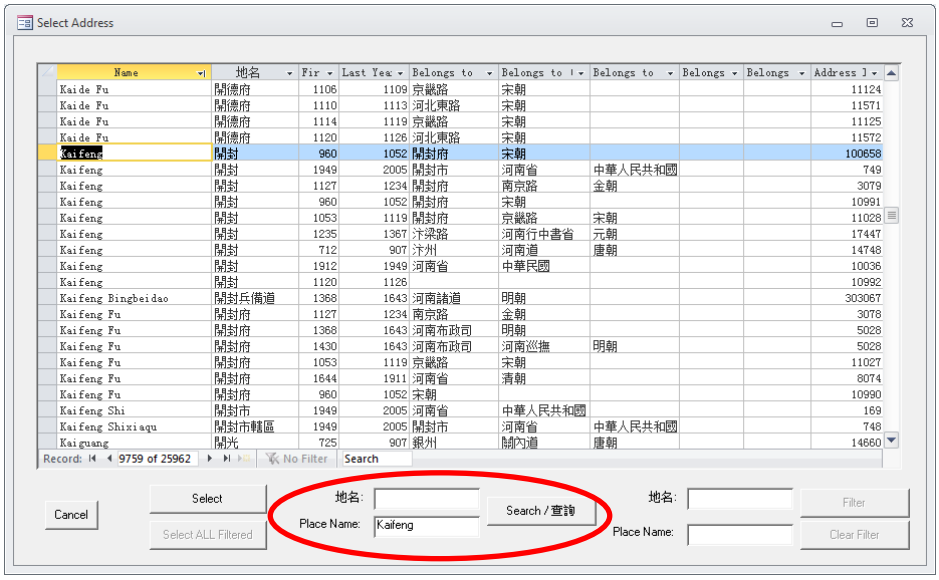
The third way to select people for analysis is to recall a list of IDs saved from a previous query. One simply clicks on the **Recall Person IDs** command button. Instead of a person's name, the form has "[Recalled List]."



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 **Search** utility to find a name either in Chinese or *pinyin*. One can also use the **Filter** feature 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

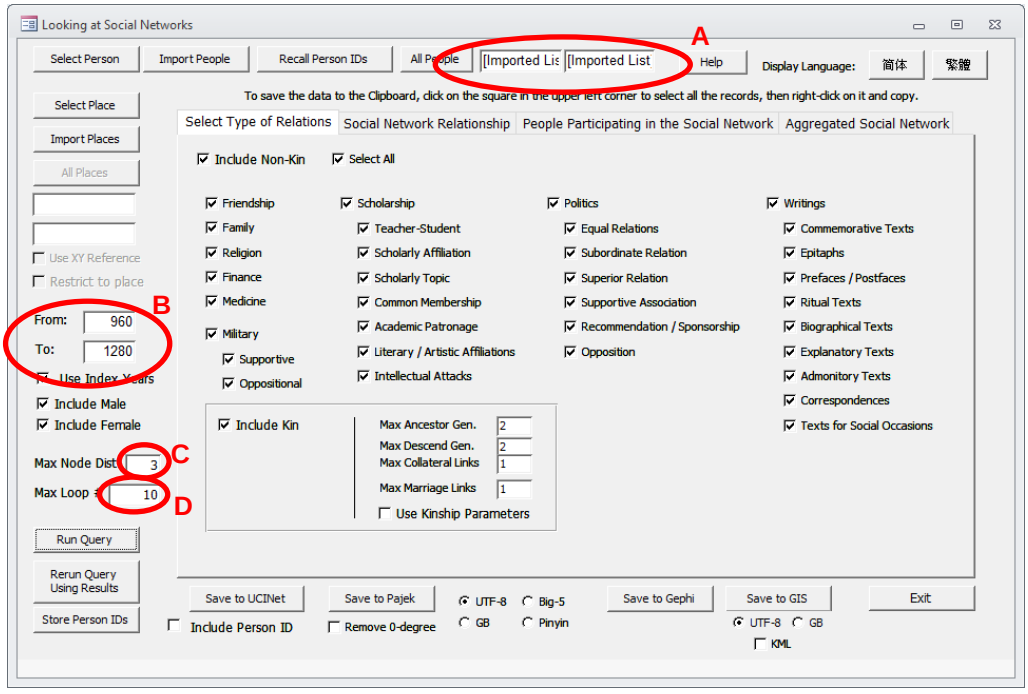
Because there are many, many categories of non-kinship relationship, most of which are of little importance in a particular query, one can limit the search to selected large groups of

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 960 through 1280 (B) with a maxi-mum 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 11,006 people participating in 39,197 relations.

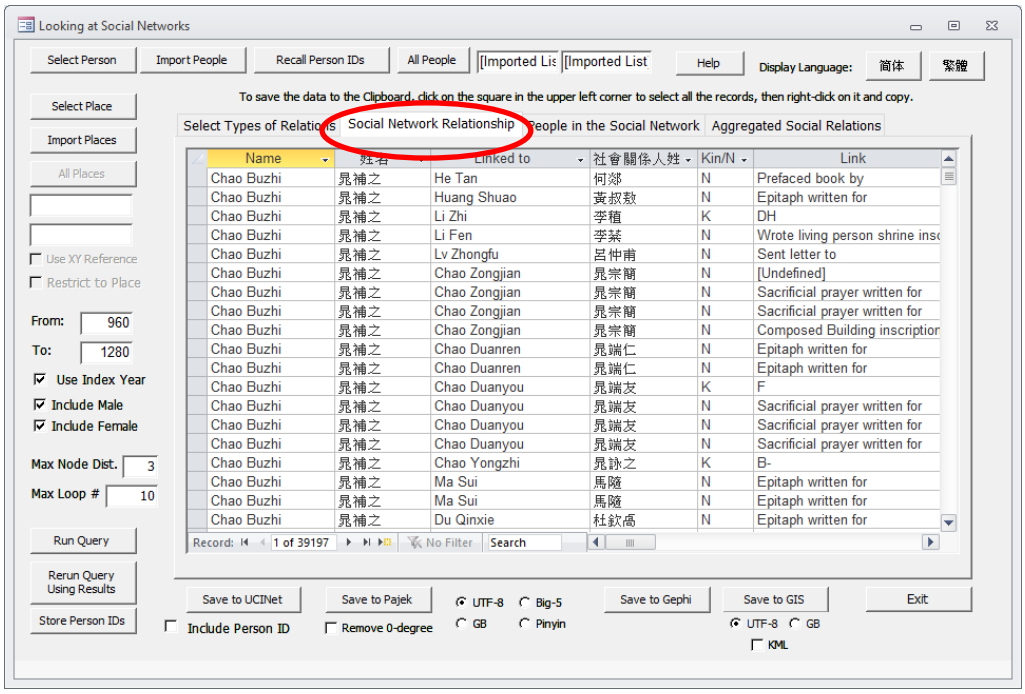


Table of Associations in the Social Network

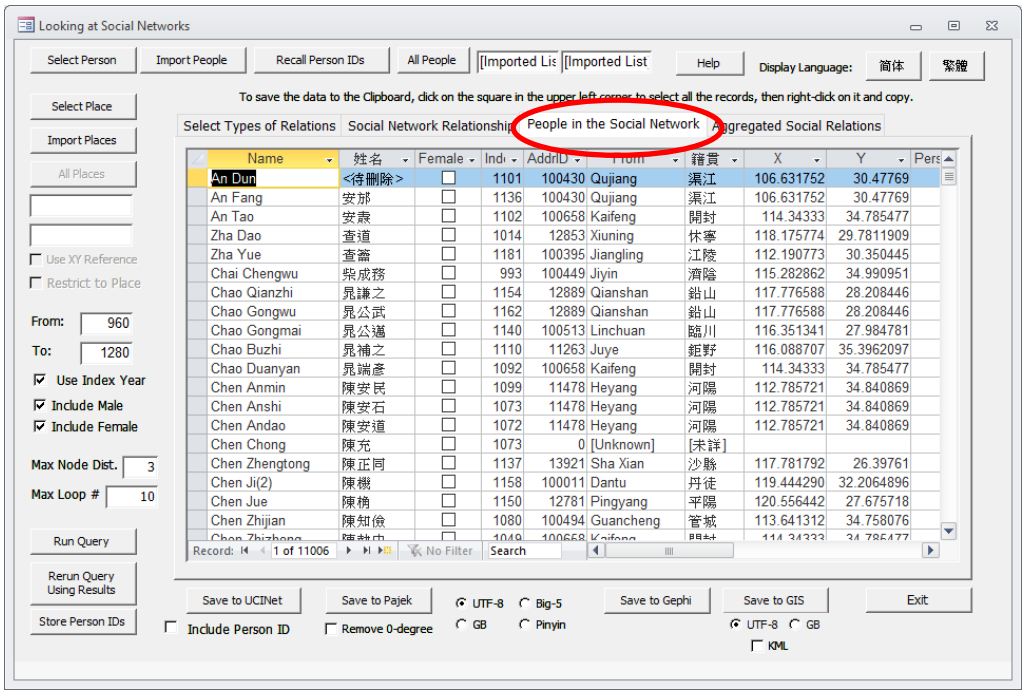
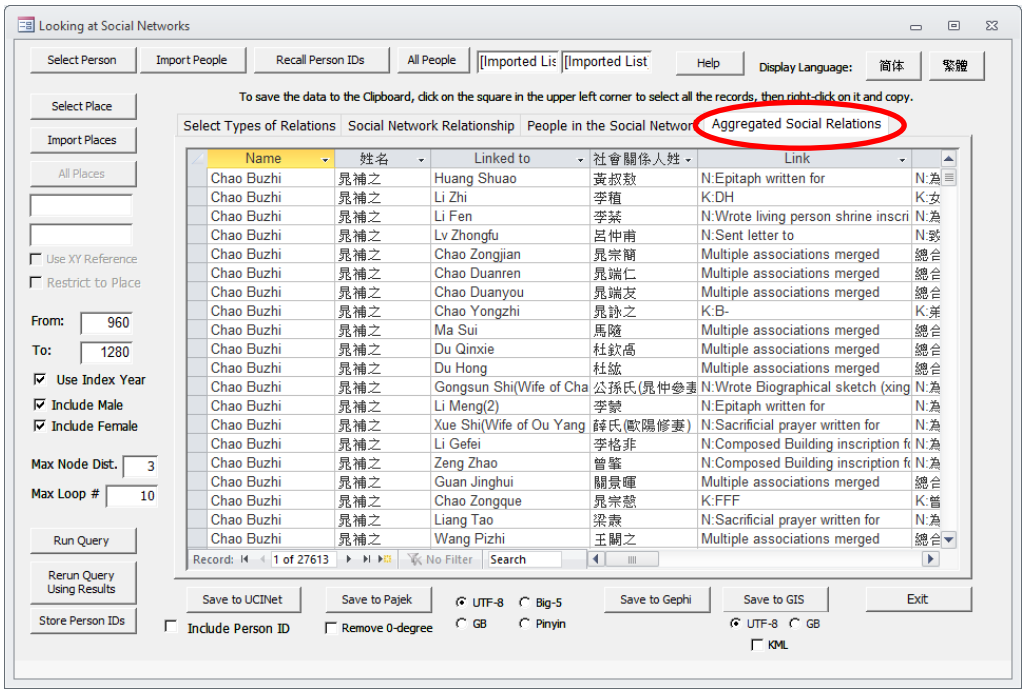
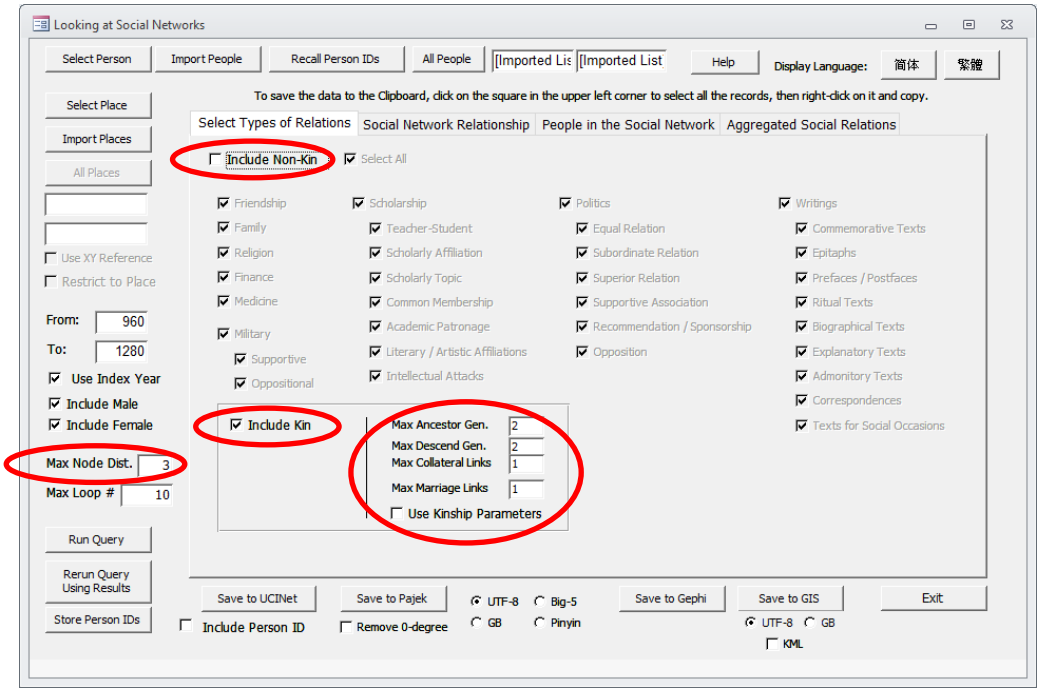


Table of People Participating in the Social Network

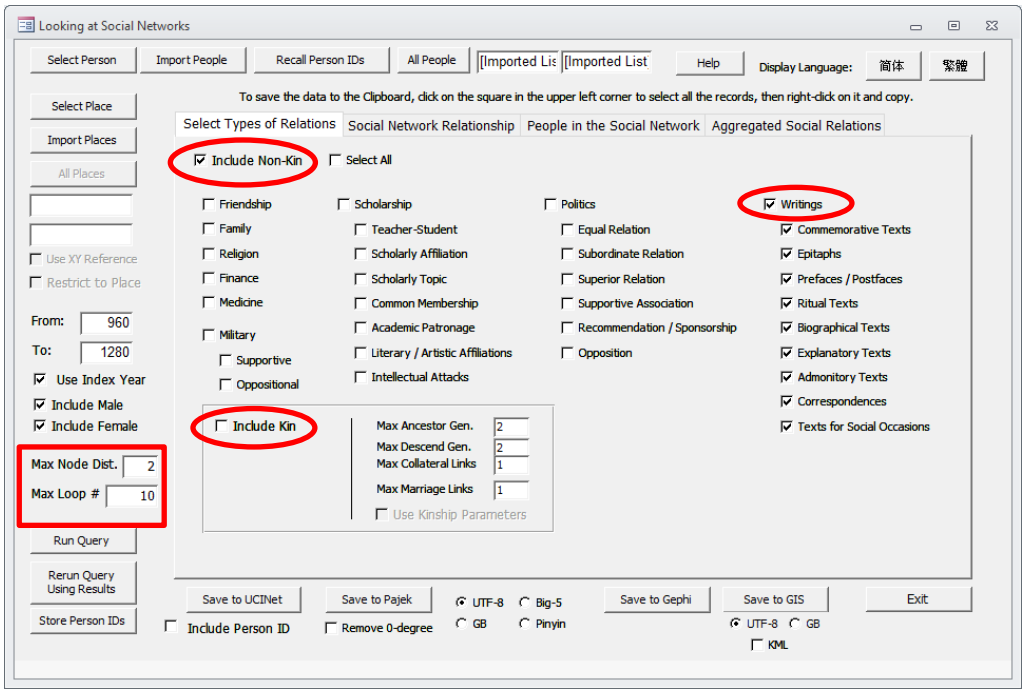
Many of the pairs of people in this list have more than one relationship between them, so CBDB also produces a table in the **Aggregated Social Relations** tab with just one record for each pair of people that gives the number of relations between them:



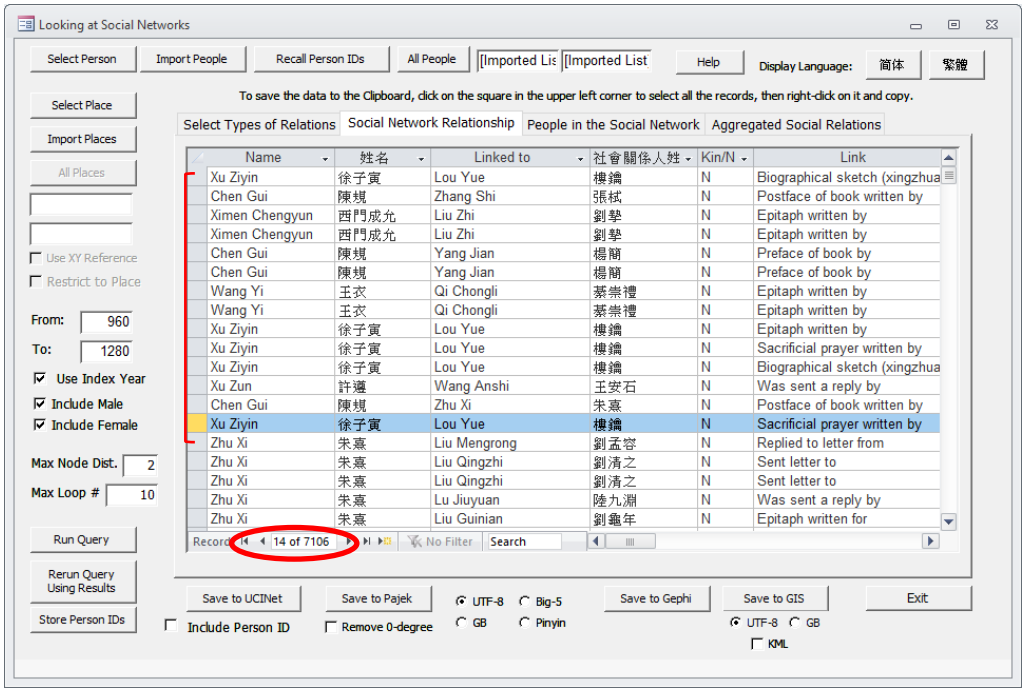
If the network is too large, one can examine more narrowly defined networks. If one looks just at kinship relations for the group, CBDB finds 146 people linked through 169 relations (with total node distance of 3 and constraints on the kinship distance for the relationships).



If one looks just at associations formed through writing with a maximum node distance of 2 and excludes kinship, CBDB discovers 3469 aggregated relations among 1195 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 3964 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 restricting the non-kin relationships to writing and click on the "Rerun Query Using Results" button. (Note that one achieves the same result by clicking on the **Store Person IDs** command button and then directly clicking on the **Recall Person IDs** command button. This has the added virtue that the list of people is now available for use in other forms as well.)

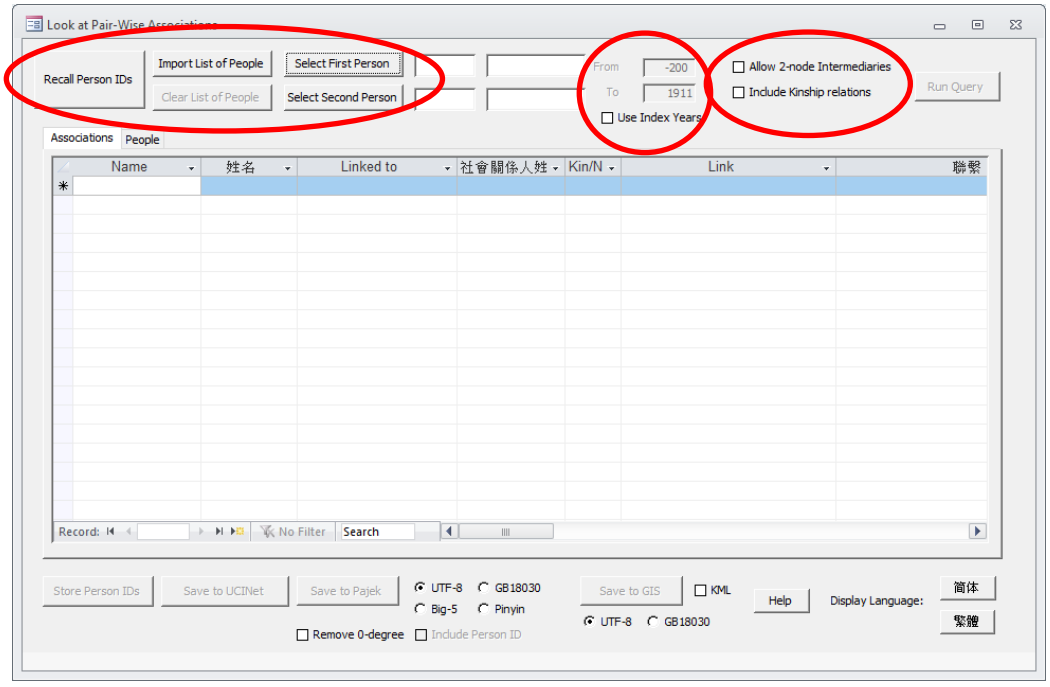
Outputting Results

LookAtNetworks provides ways to output the results of a query to three different SNA programs: **UCINet**, **Pajek**, and **Gephi**. Because Pajek has begun to support 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 SNA output files uses color-coding to indicate degree of distance from the target person:

White = the target node (the UCINet files set these to Black)
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

F. Using the Form LookAtAssociationPairs

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 a list of people from earlier, saved query results using the procedure described for other forms above, then (2) the range of index years for the people in the relations, if desired, and finally, (3) the type of permissible relations. The options for relationships are:

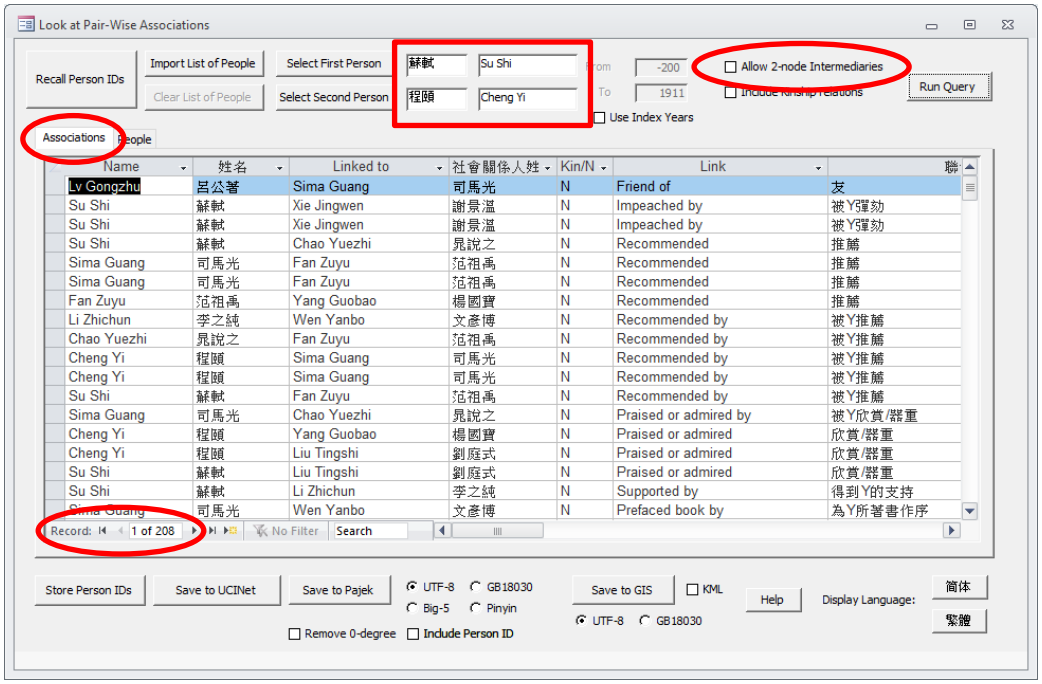
1. **Allow 1-node Intermediaries**: That is, people who are directly linked to both (or, for imported lists, two) of the selected people: Person A — Node1 — Person B. In this case one leaves the check box for two-node intermediaries unchecked.

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.

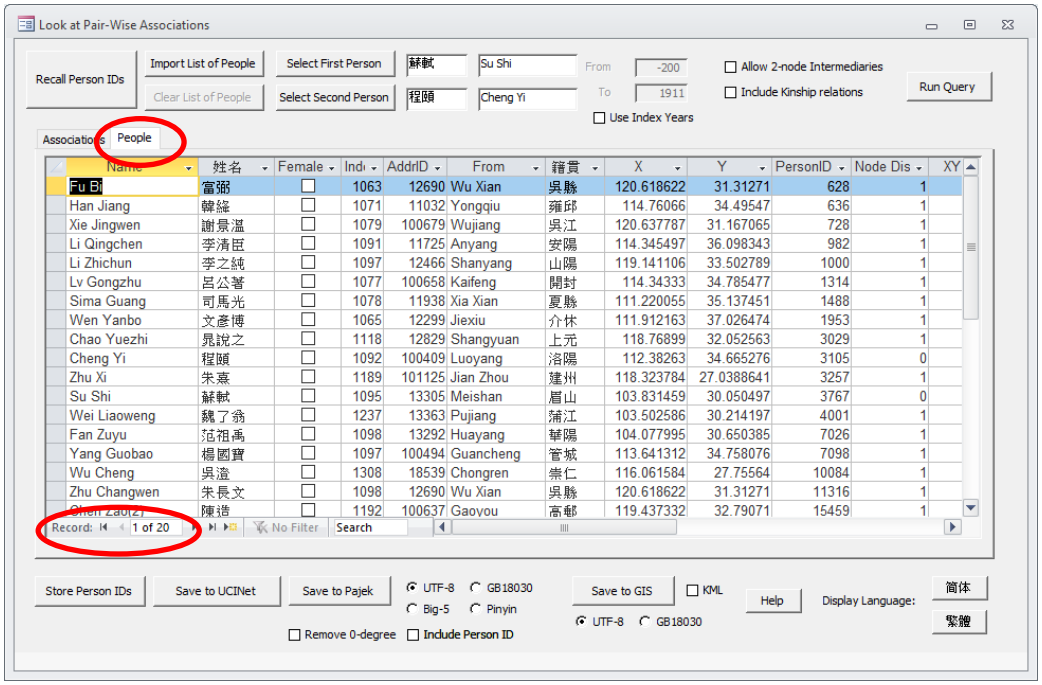
3. **Include Kinship relations**: The default is simply to look at social (non-kinship) relations connecting people, but kinship also can be important, and the form allows on to examine the role of kinship relations in the social network.

One Node Intermediary Searches

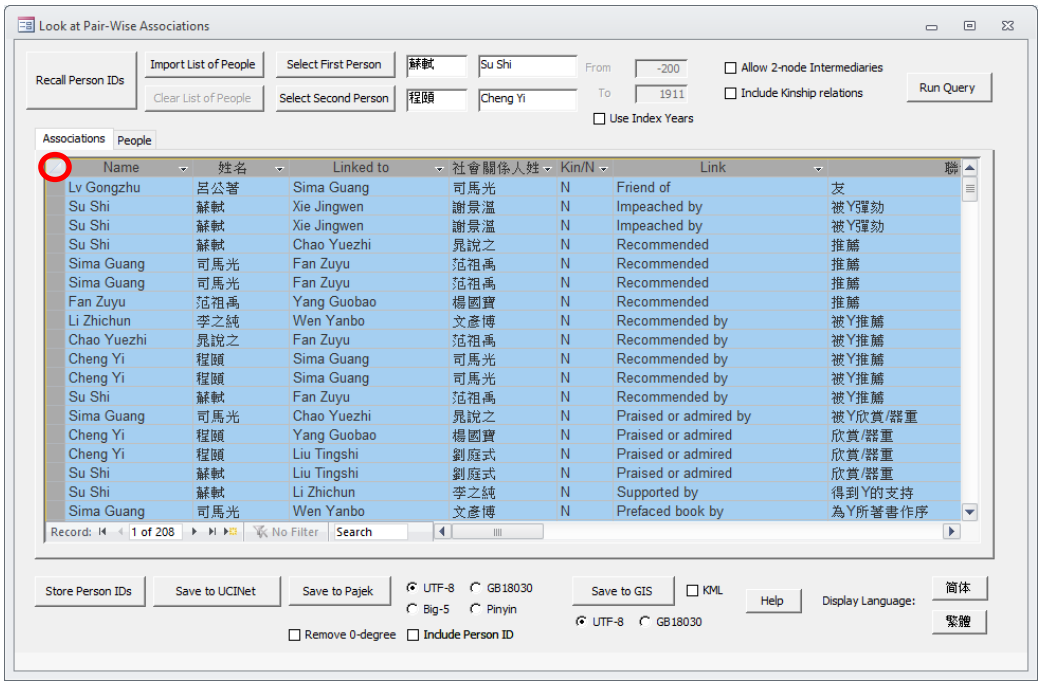
For example, if one explores the links between Su Shi 蘇軾 and Cheng Yi 程頤, allowing only people directly linked to both of them finds 208 associations among twenty people.



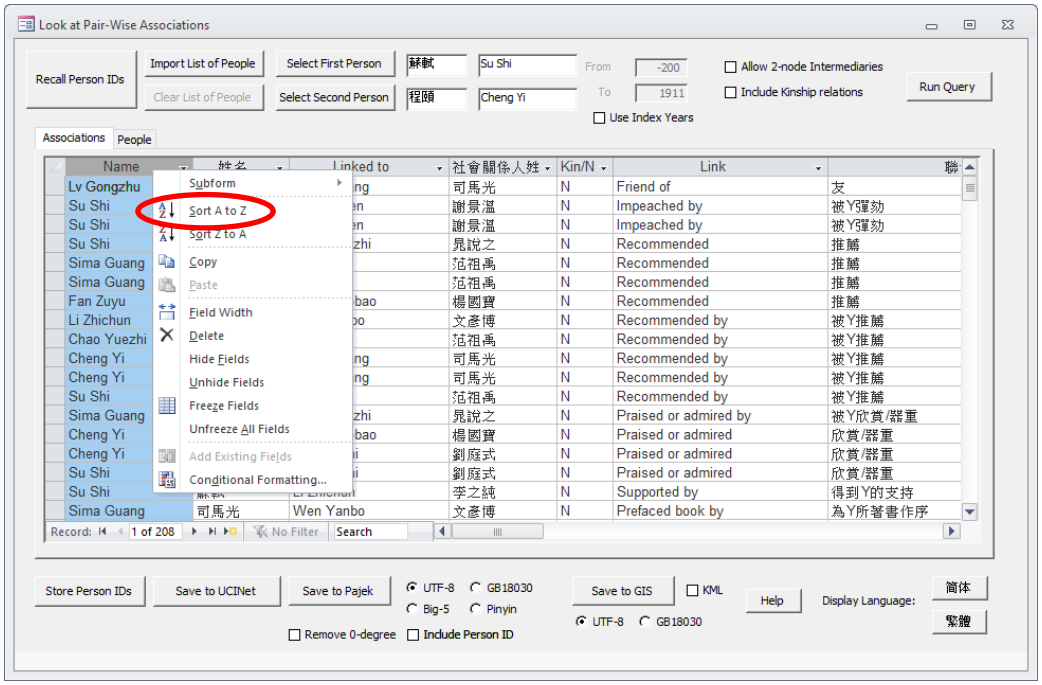
As in **LookAtNetworks**, the form provides two output tables: "Associations" for the relationships, and another, "People in Associations," for the people in the relations.



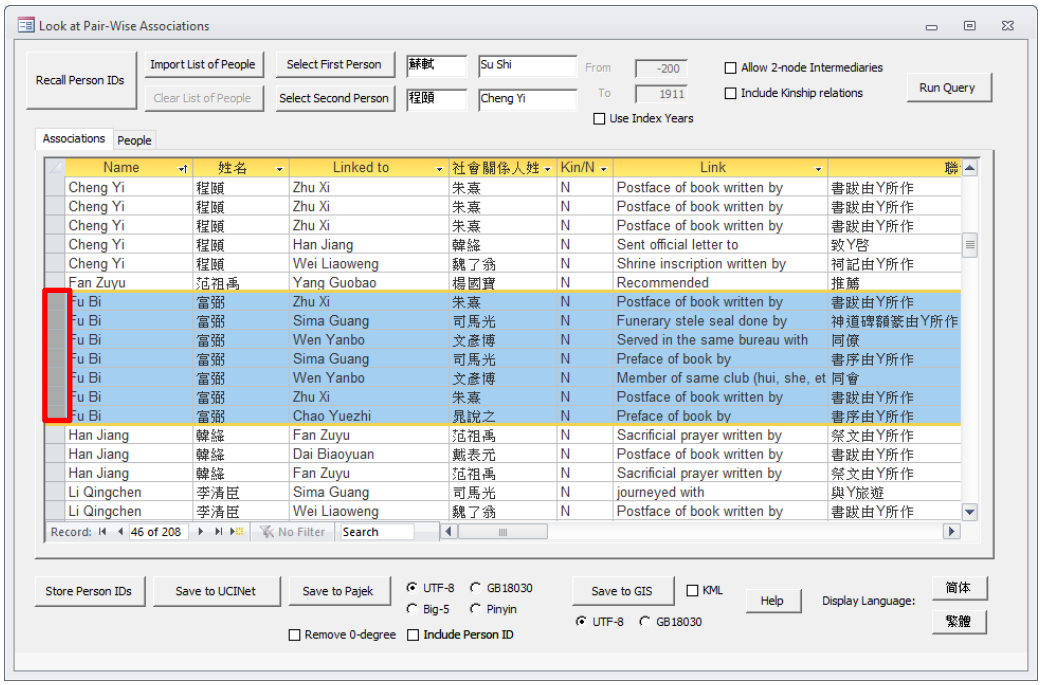
As with the other forms, one can save the results of a search by clicking on the grey square in the upper left hand corner of the table to select all the records and then using Ctrl-C:



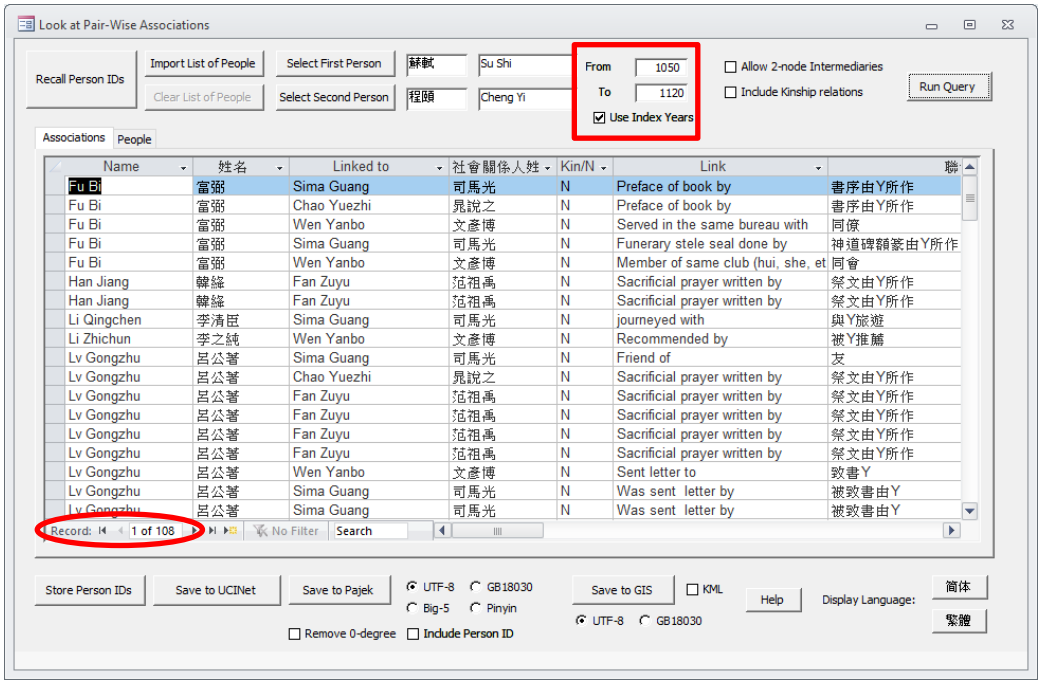
One also can sort on a column of the table by clicking on the column (in this case, "Name") to select it, then right-clicking to choose the type of sort:



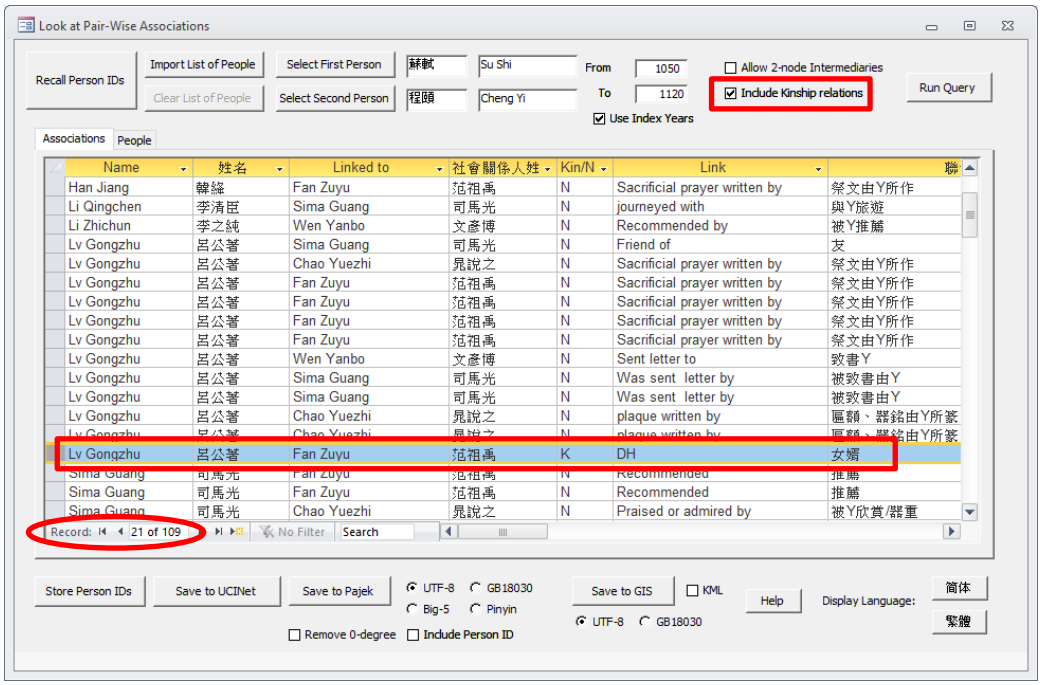
One also can select a block of records to save by clicking the mouse on the left-hand grey column of the first record in the block and then, with the left-click button still held down, dragging the mouse down the grey column to the last record in the desired group:



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. If one limits the index years to a range between 1050 and 1120, one finds fourteen people with 108 relations connecting them:

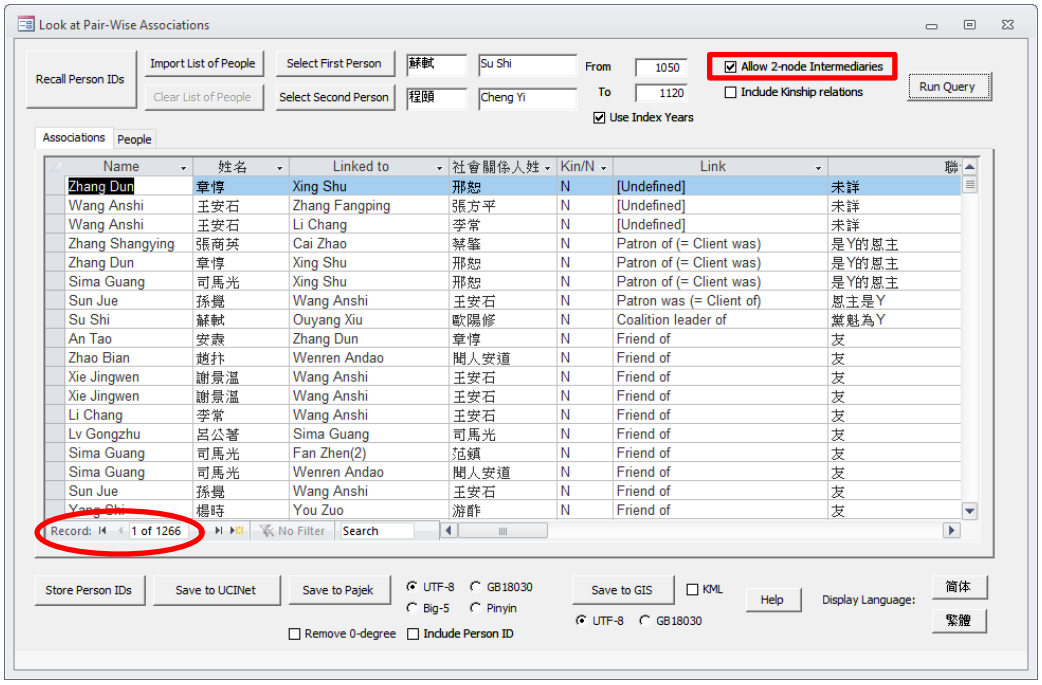


If one then includes kin of either Su Shi or Cheng Yi who have a social connection to the other, then one discovers one additional connection but, in this case, no additional people:



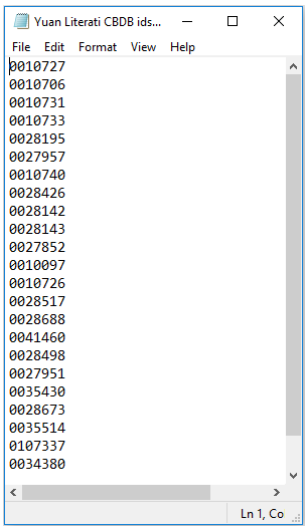
Two Node Intermediary Searches

If one broadens the search to allow two intermediary links to connect the target people, the network becomes more complicated: The program reveals 1266 relations among 115 people with index years between 1050 and 1120:

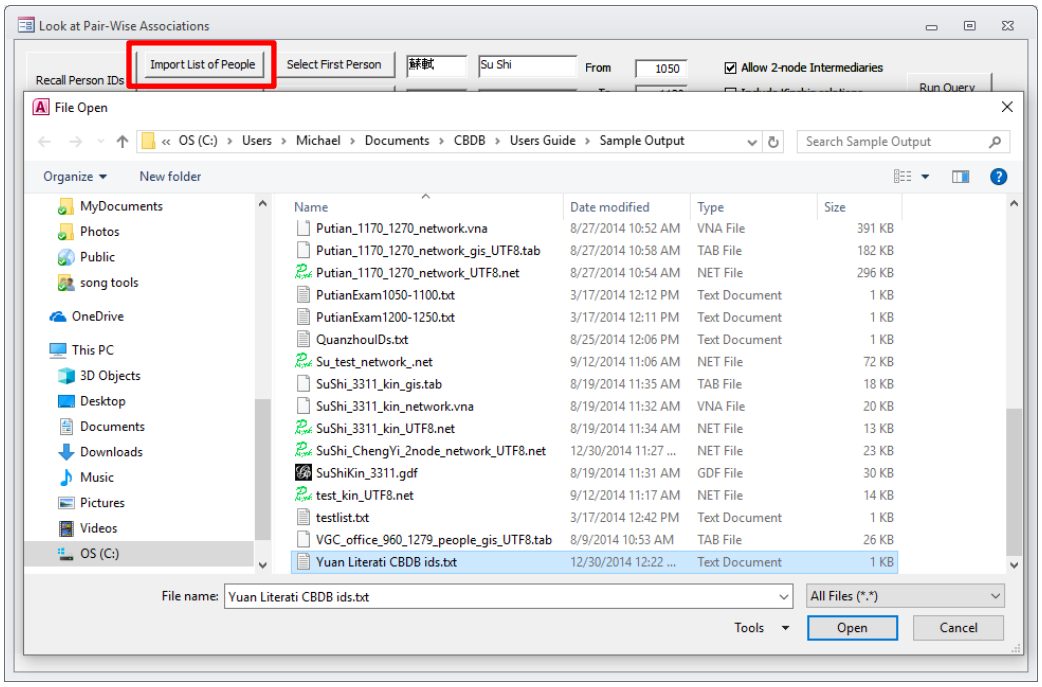


Searches Using Lists

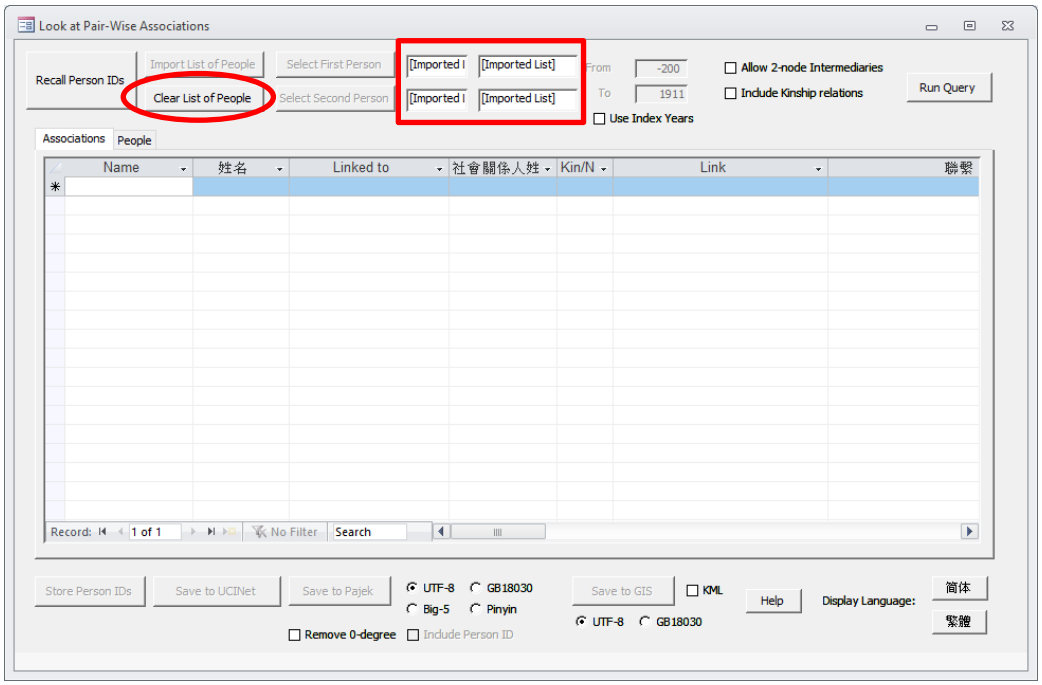
If one wants to look for connections within a larger group of people chosen by other criteria, the form allows one to import a list of person IDs. Here one looks at Jinhua men who from the Yuan dynasty who have extant collections. As in all lists for importing people, CBDB requires a single column of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



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

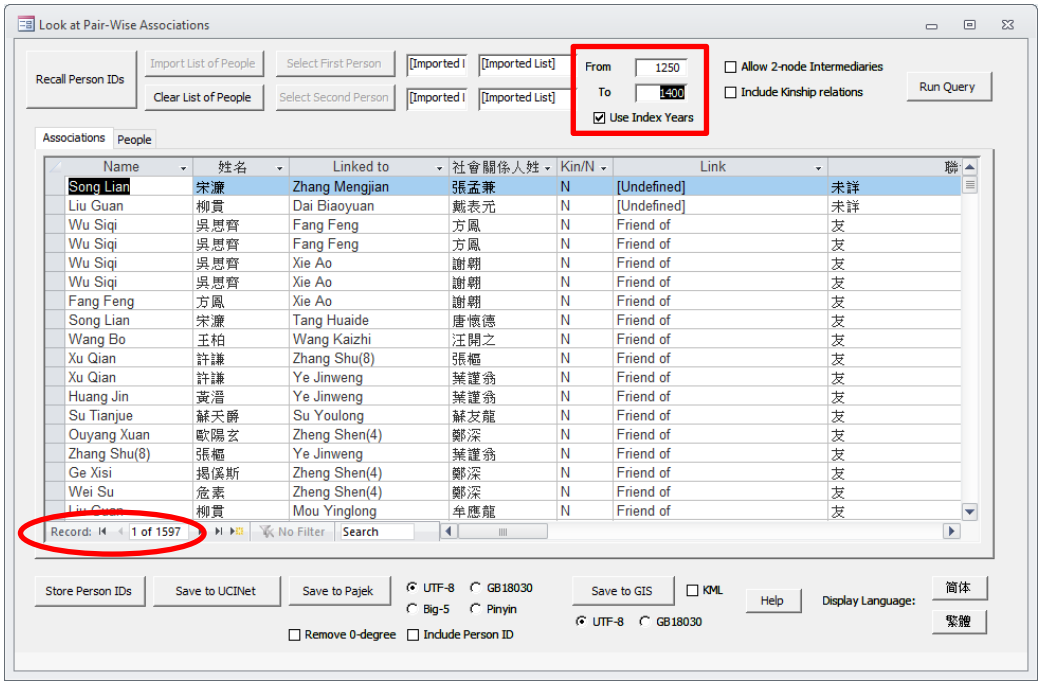


If the file is successfully read, the form indicates that the names are from an imported list:



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.

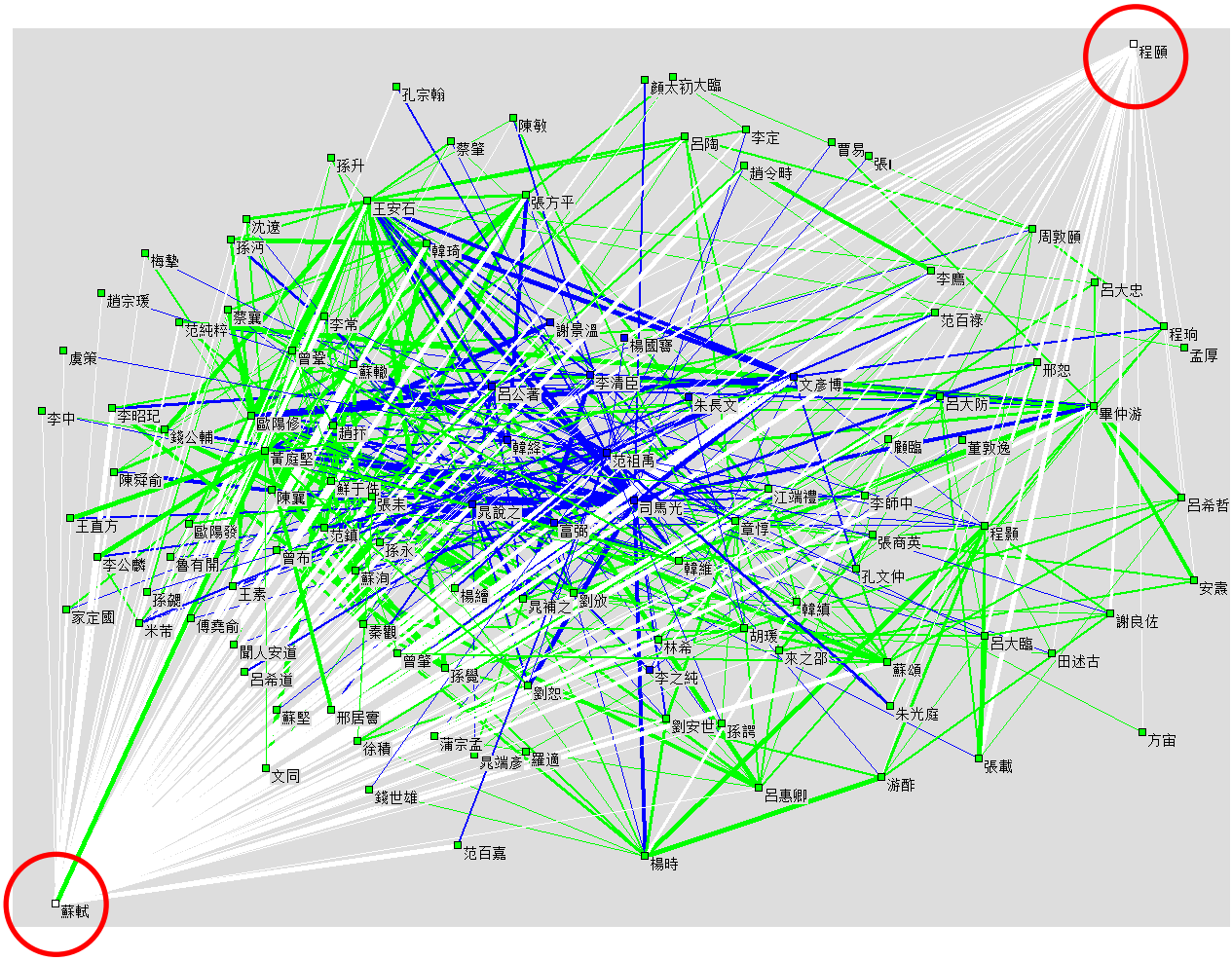
Once one has imported the list, the search procedures are the same. In this case, the query is set to look for one-node intermediaries with index years between 1250 and 1400 and produces 1,597 associations among 180 people:



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)

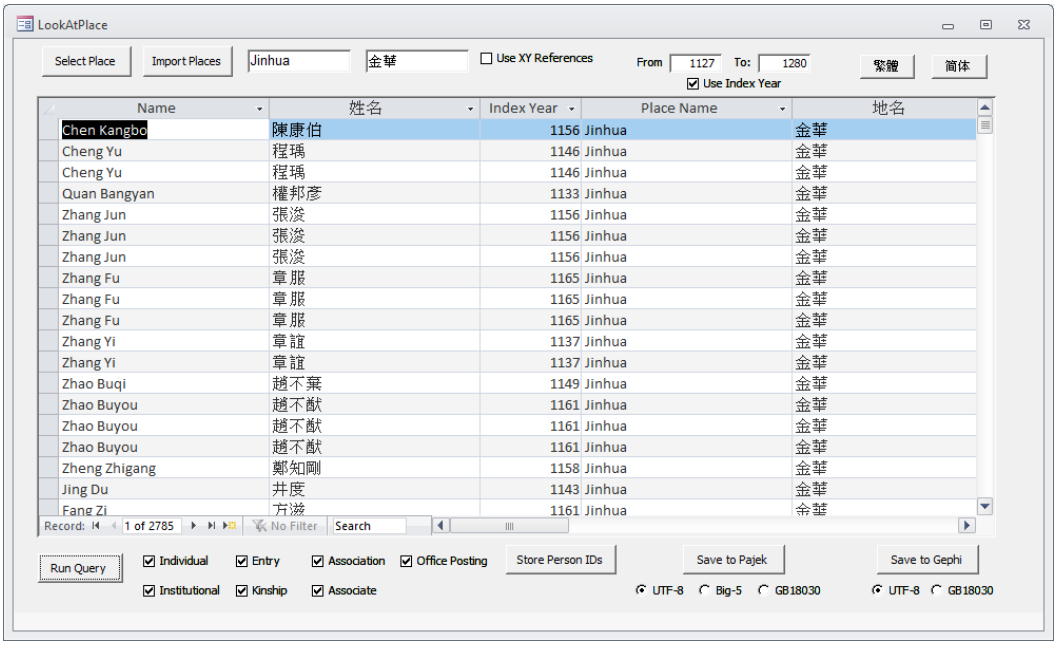
The output files aggregate the associations between people, and the width of the lines reflects the number of associations between nodes.

G. Using the Form LookAtPlace

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 in the Southern Song dynasty.



One can select which relationship to place to include in the search and can specify the usual sorts of parameters (use of *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.

The output table has 14 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 Association
11. Relation (English)
12. Relation (Chinese)
13. X coordinate
14. Y-coordinate

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.

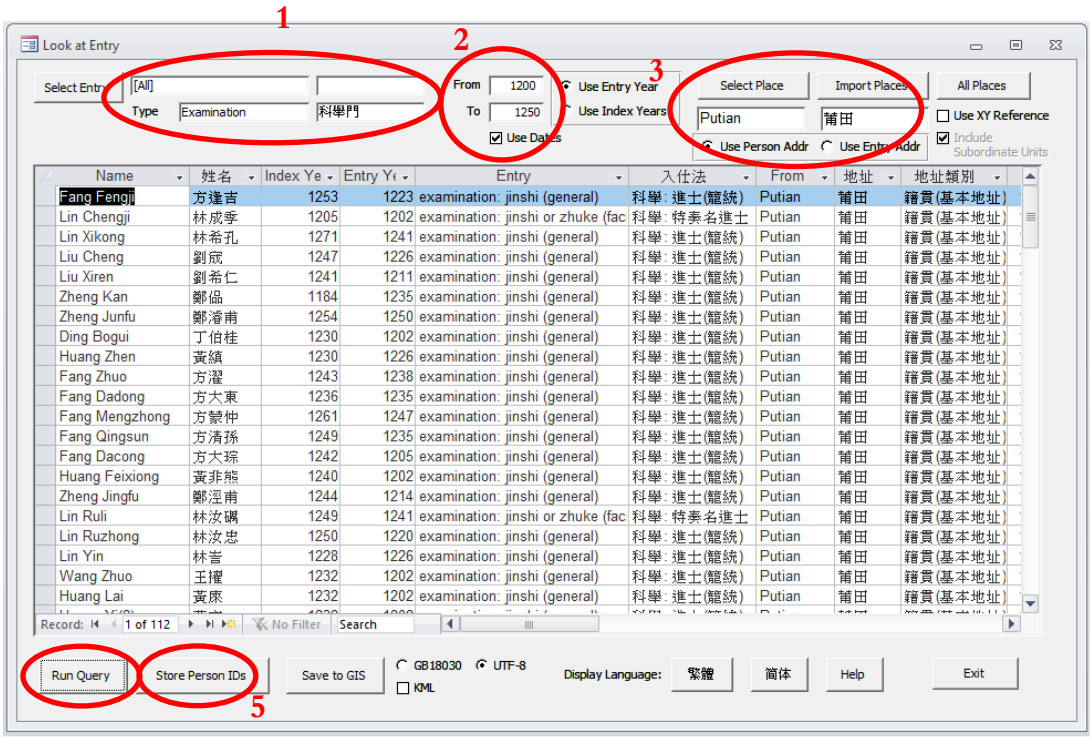
At present, the only way to write the results of a search to a file is as SNA data in Pajek or Gephi format. If there is a need to save the data in GIS form, this functionality can be added in future versions of the software. It should be stressed that this form is still somewhat experimental, and suggestions are welcome about its functionality and design.

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

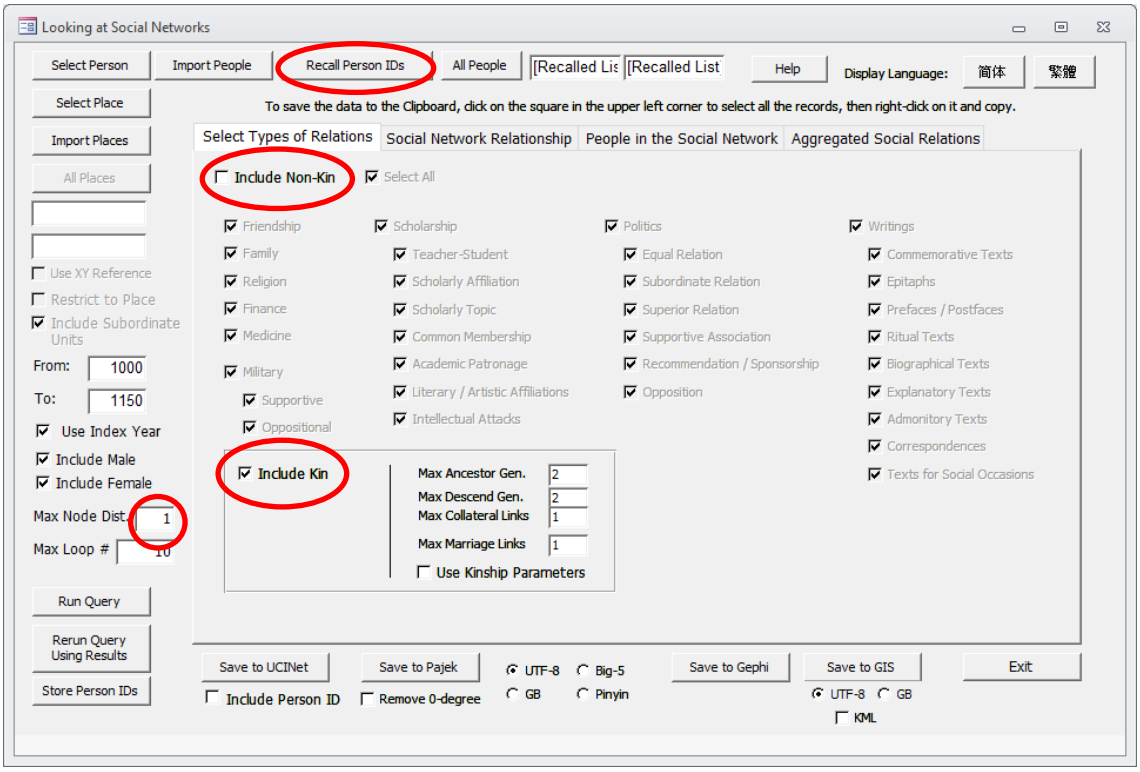
One question in the study of social history during the Song dynasty is whether local elites remained stable and controlled access to the cultural resources needed to gain entrance to official status or whether there was in fact social mobility where marginal families managed to join the elite stratum 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.

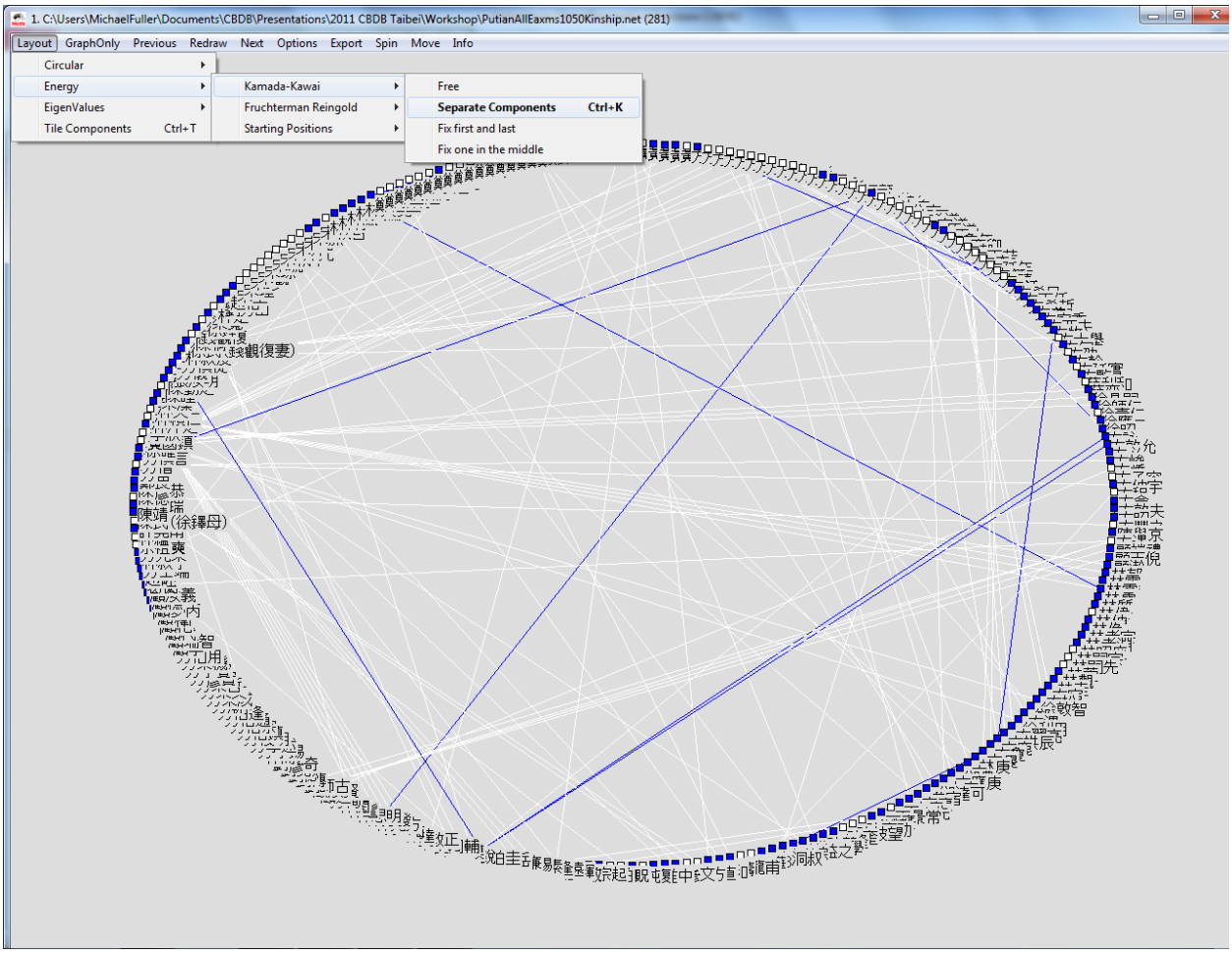
Once you have the table of the IDs of people from Putian who entered government through examination for the specified period, open the form **LookAtNetworks** and have the form read the stored table of people for 1050-1100:



Here, the procedure is:

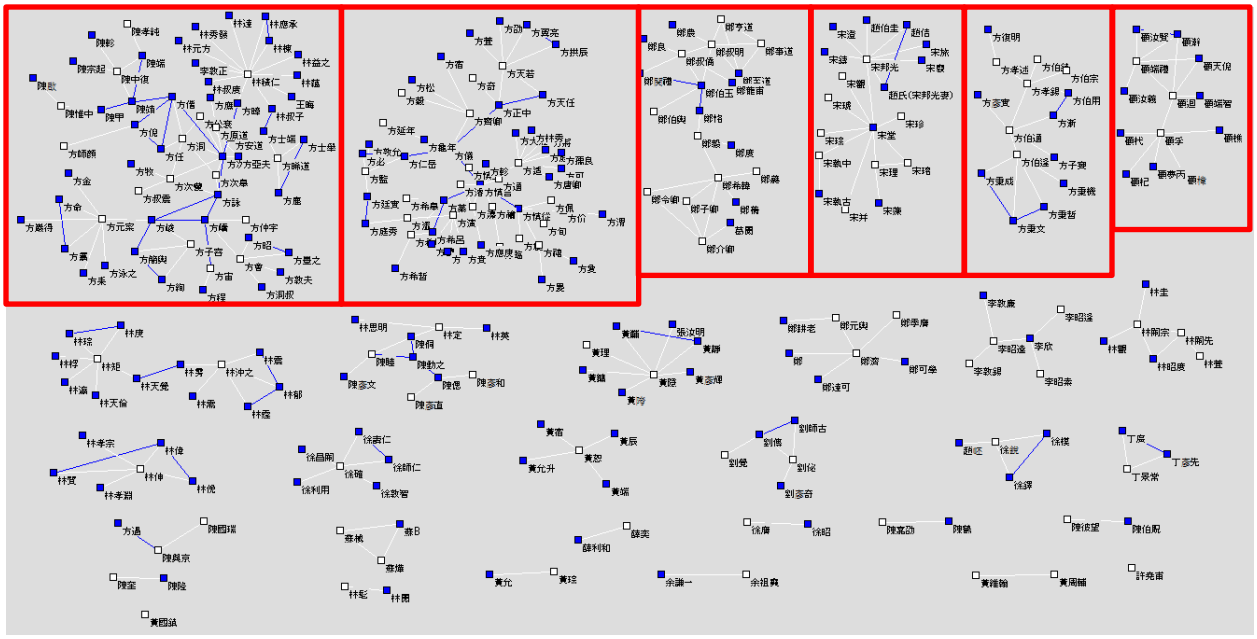
(1) Recall the list of people IDs with the **Recall Person IDs** command button at the top of the form.
(2) You will get confirmation that the table was correctly imported when you see "[Recalled List]."
(3) Set the node distance to 1: we want to look only at directly connected people.
(4) In this case, we want to look at just kinship networks, so unclick "Non-Kin."
(5) After you run the query, save the results into a **Pajek** file that uses UTF-8 encoding.
(6) Repeat the process for the people from 1200-1250 and create a second Pajek file.

Open **Pajek** and "Draw" the results:

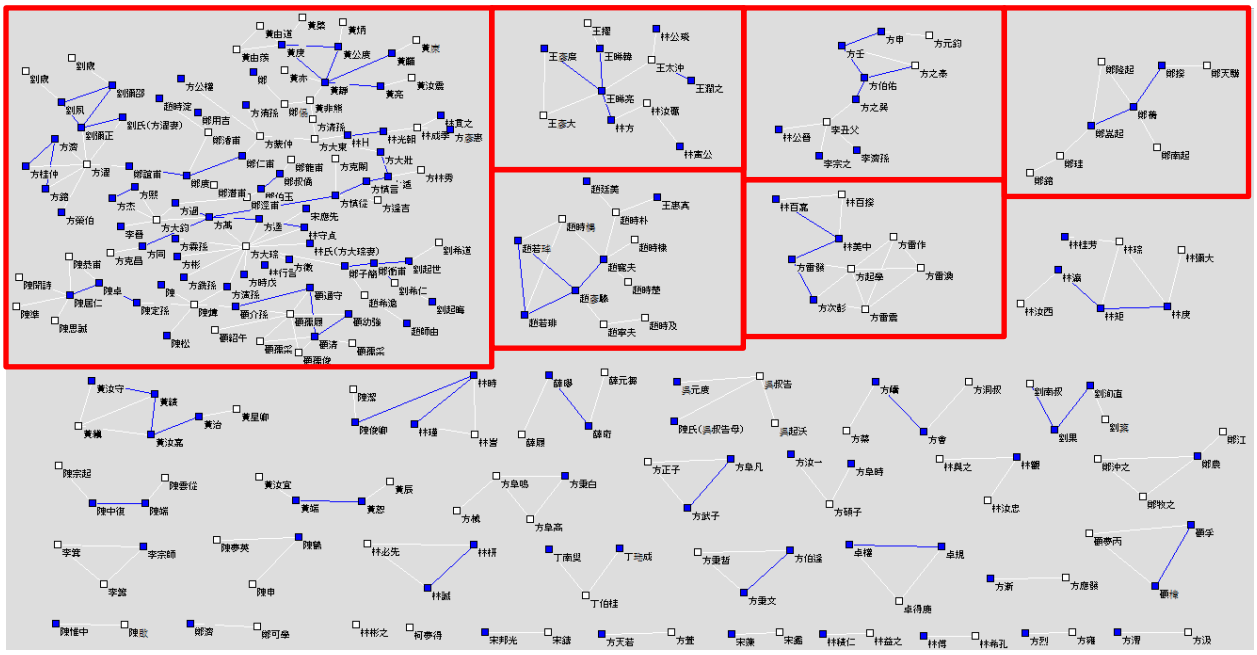


The initial layout for visualizing networks in Pajek is "Circular." A more useful layout for looking at groups of kinship networks is to select "Separate Components" in the "Kamada-Kawai" layout listings.

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. The white lines connect the examinees to their kin and to each other, while the blue lines connect kin to one another.

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:

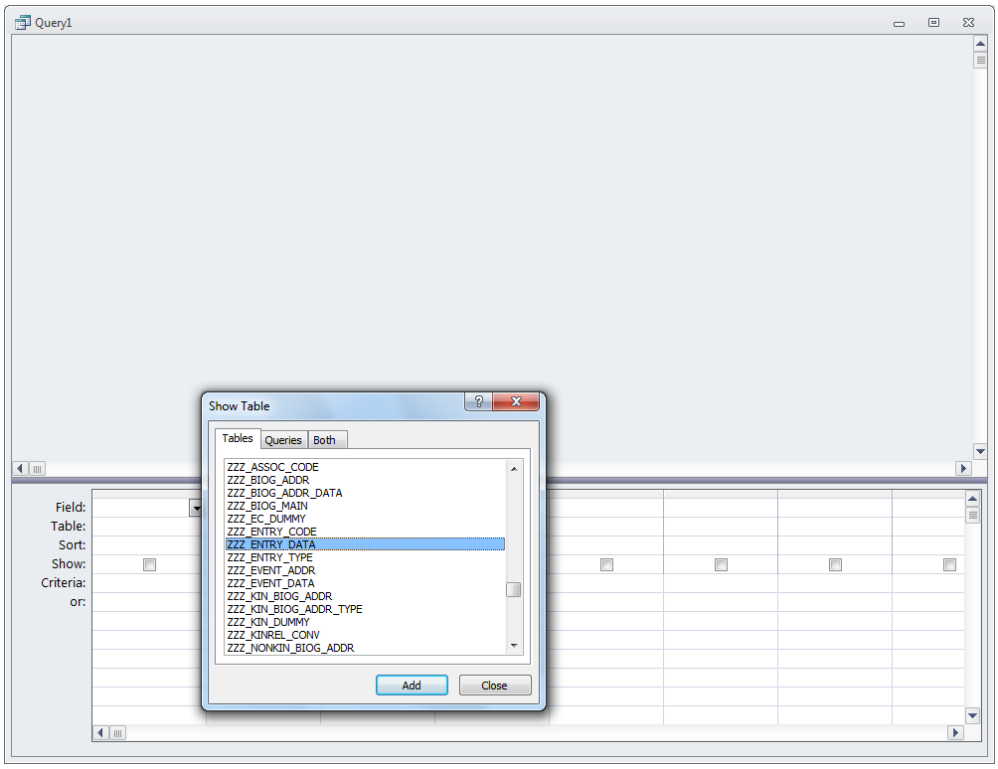
1. ZZZ_ALT_NAME_DATA
   (fills in alternate name type)
2. ZZZ_BIOG_ADDR_DATA
   (fills in address and address type)
3. ZZZ_BIOG_MAIN
   (fills in nianhao, ethnicity)
4. ZZZ_ENTRY_DATA
   (fills in entry type)
5. ZZZ_KIN_BIOG_ADDR
   (this is the table for kinship, but it also provides the main entry 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?

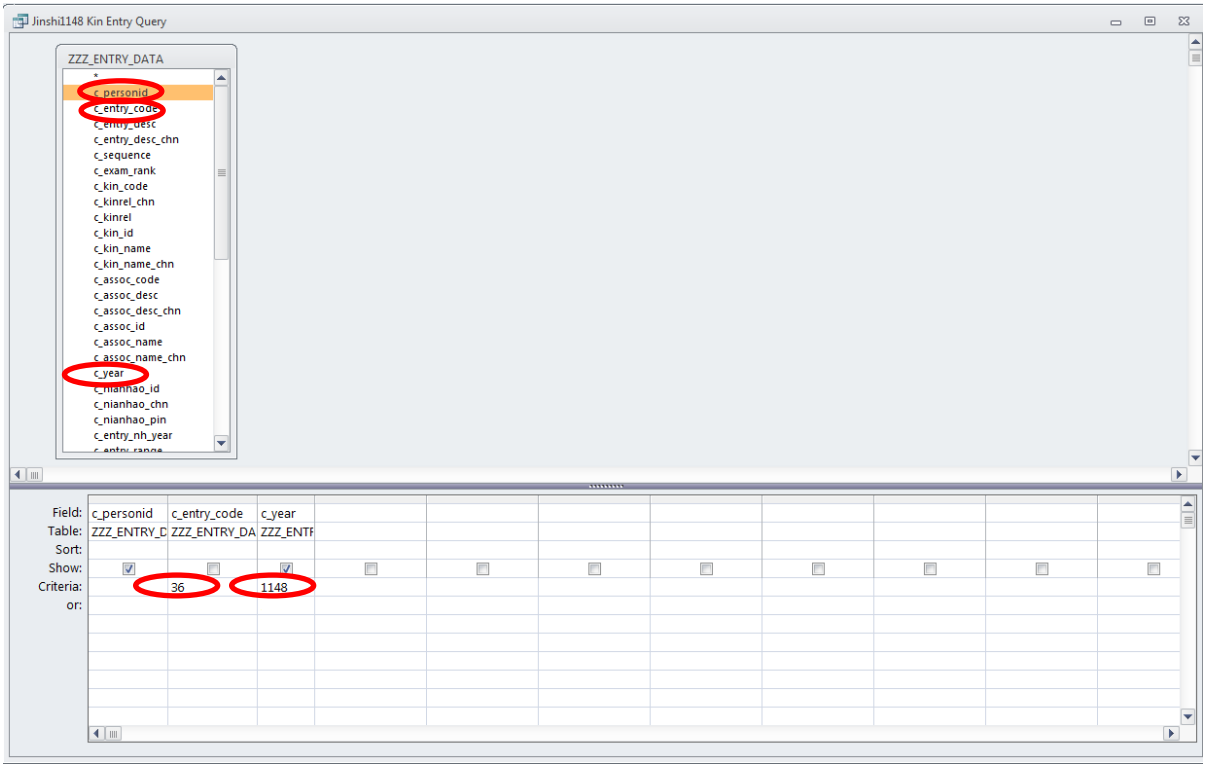
1. In the **Create** menu (next to the **Home** tab at the top of the main screen) , Select **Query Design**:



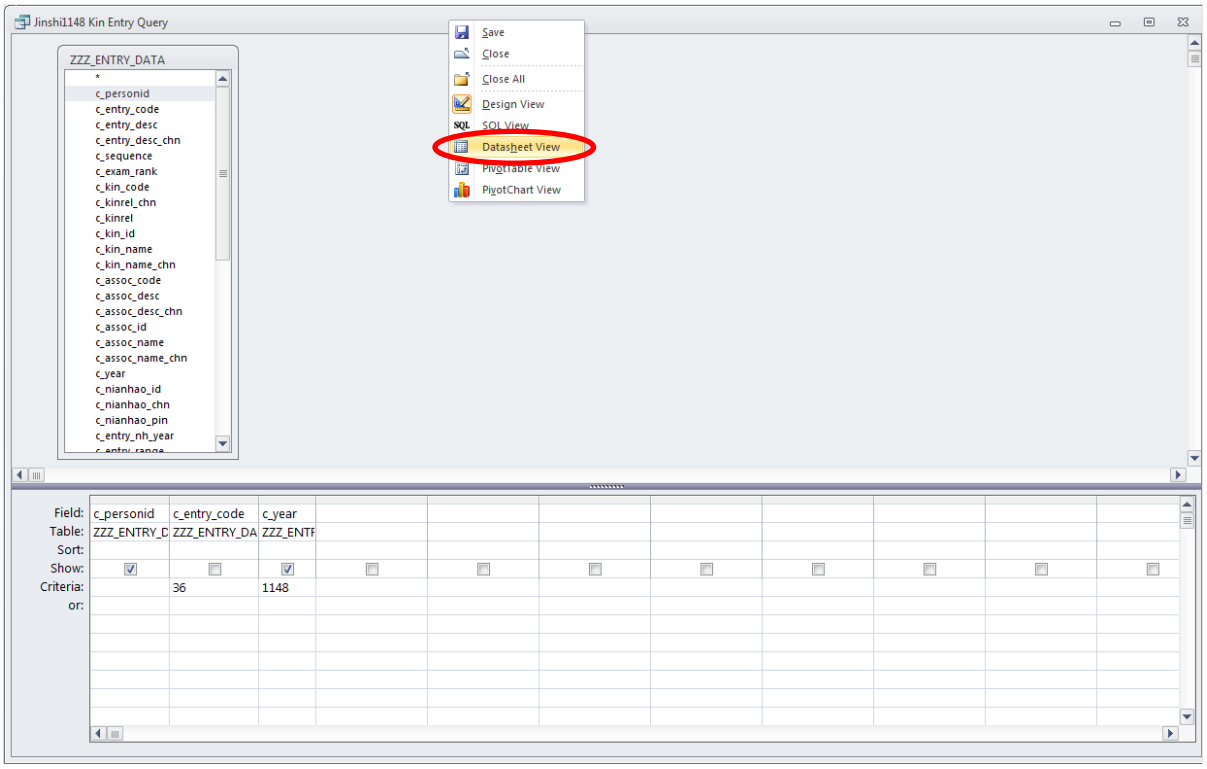
In the "Show Table" window, select ZZZ_ENTRY_DATA and click **Add**

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.

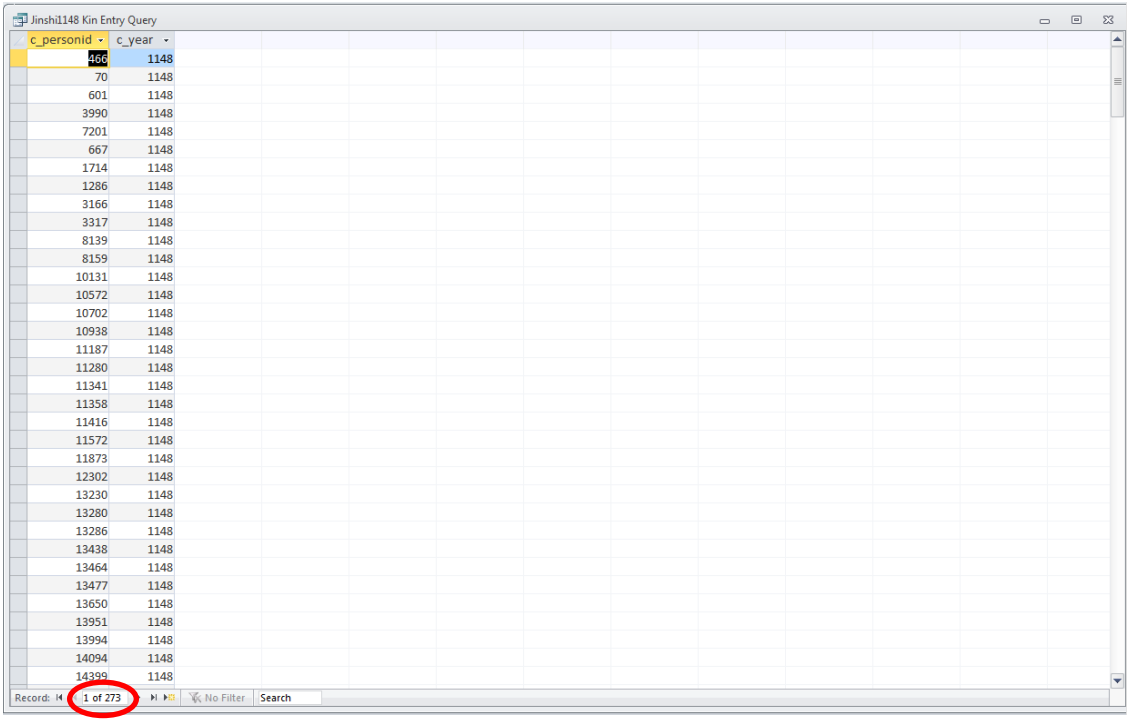
3. Then in "Criteria" specify the value 36 for c_entry_code (*jinshi*), and the year 1148.



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.)



c_personid	c_year
466	1148
70	1148
601	1148
3990	1148
7201	1148
667	1148
1714	1148
1286	1148
3166	1148
3317	1148
8139	1148
8159	1148
10131	1148
10572	1148
10702	1148
10938	1148
11187	1148
11280	1148
11341	1148
11358	1148
11416	1148
11572	1148
11873	1148
12302	1148
13230	1148
13280	1148
13286	1148
13438	1148
13464	1148
13477	1148
13650	1148
13951	1148
13994	1148
14094	1148
14399	1148

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:

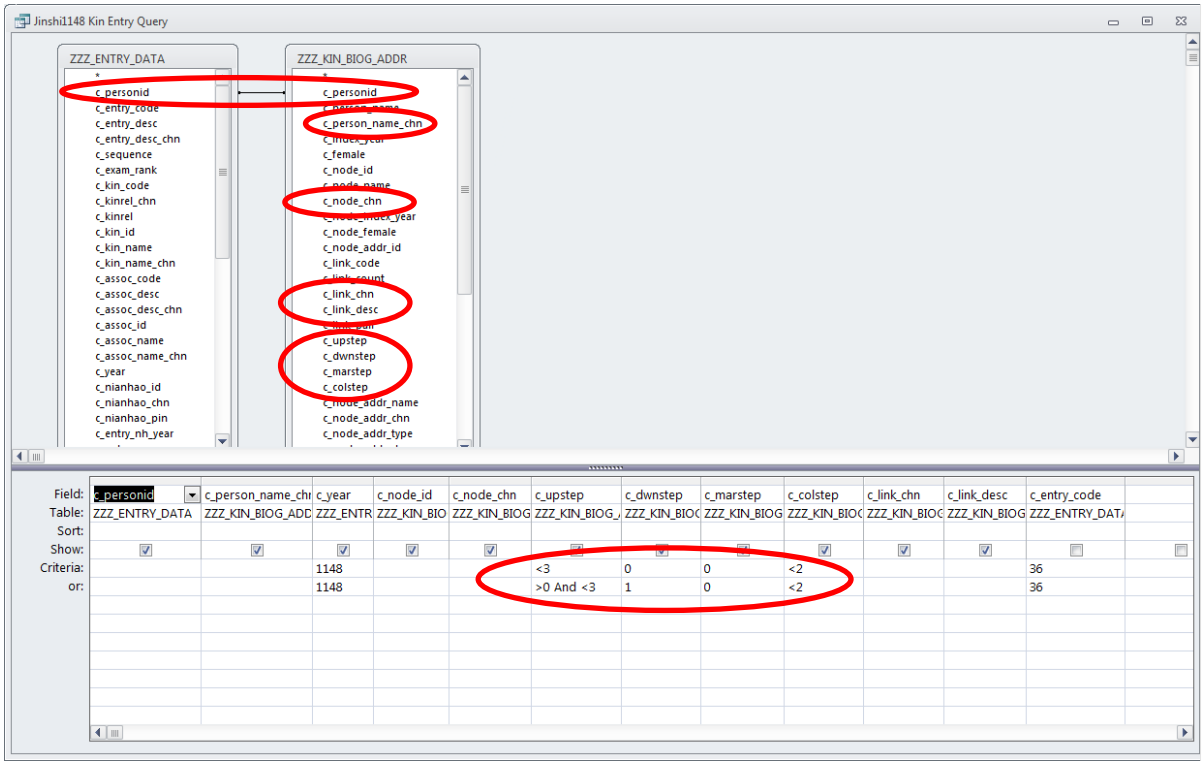
a. Create a **link** between the two tables by clicking on c_personid in ZZZ_ENTRY_DATA and dragging it to the c_personid in ZZZ_KIN_BIOG_ADDR. The query builder may ask you to confirm that you want to select only those pairs of records from the two tables which share the same person IDs.

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)

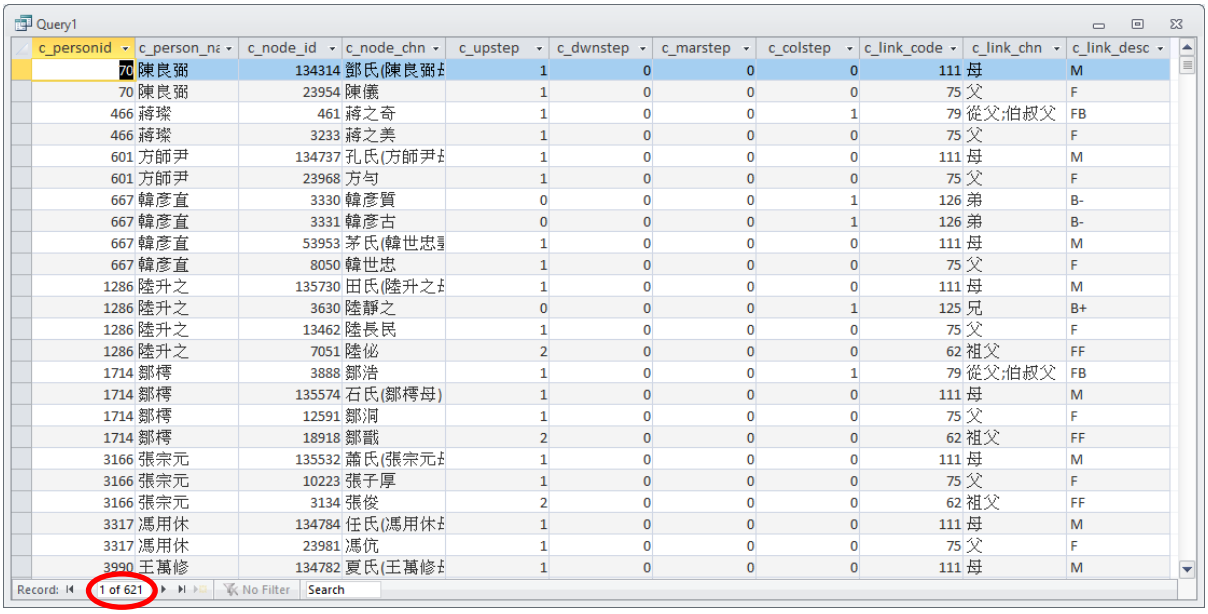
c. Set the limit for generations up (c_upstep) to 2 (i.e., FF, FFB, etc.)
Set the limit for generations down (c_dwnstep) to 0 (i.e, we want to look only at ancestors)
Set the limit for affines (c_marstep) to 0

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

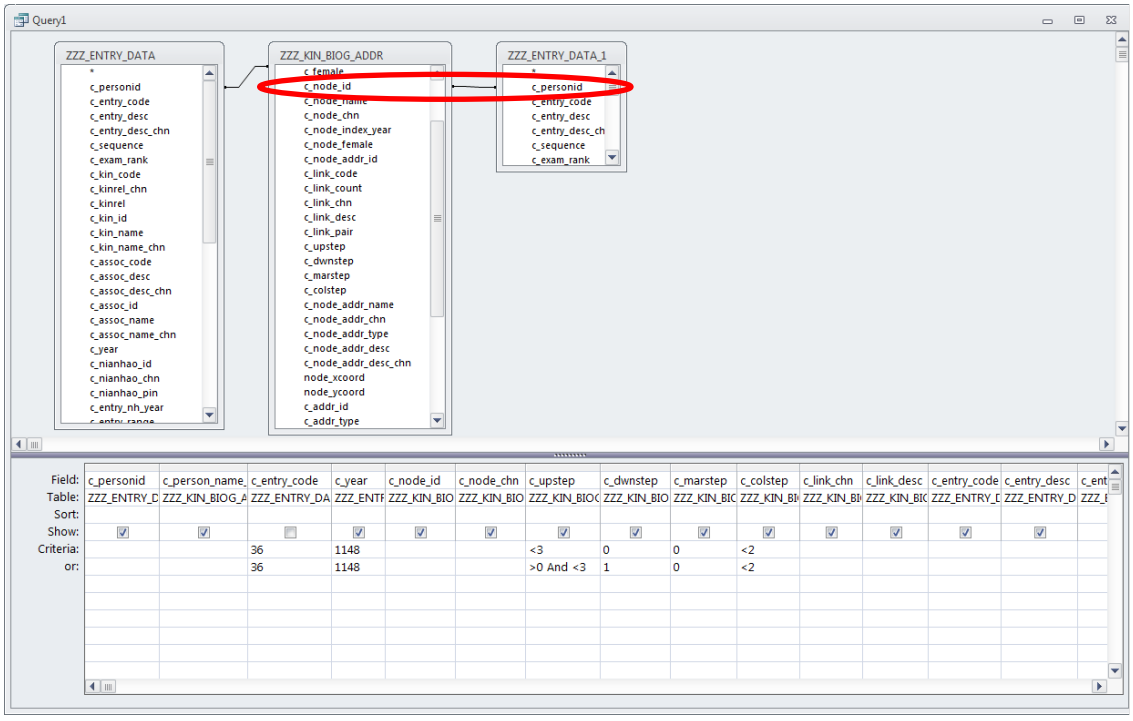
d. Repeat this process, but allow cousins (i.e. FBS or FFBS: 1 down step, at least 1 up)



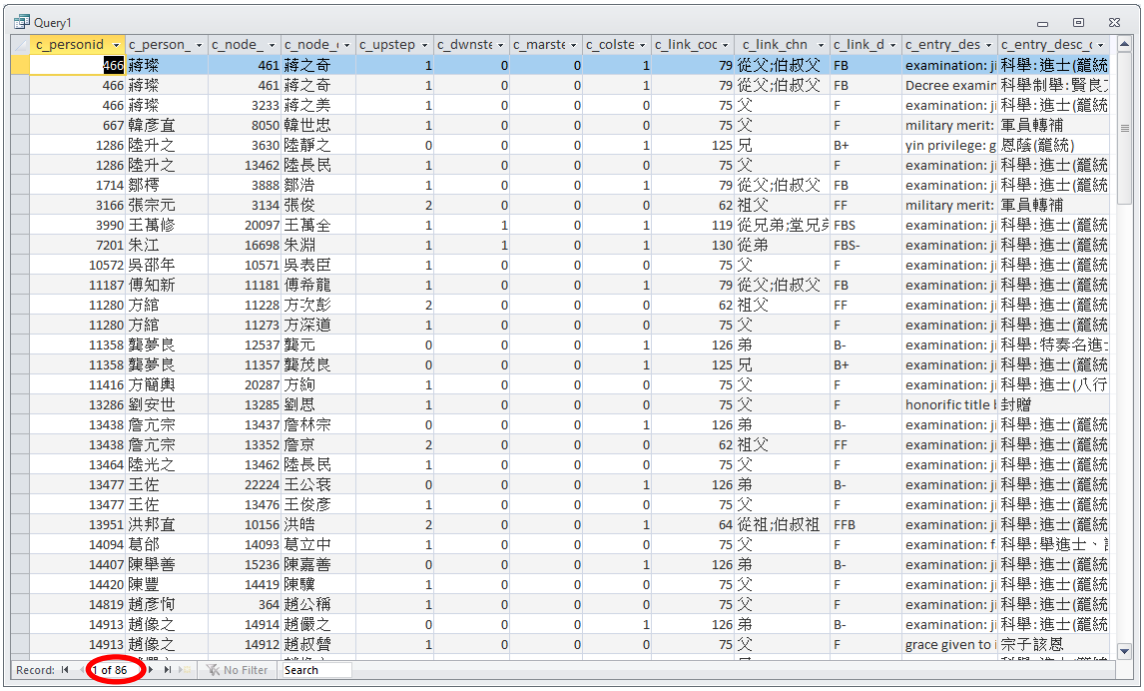
6. Check the results: There are 621 relatives that meet the criteria



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:

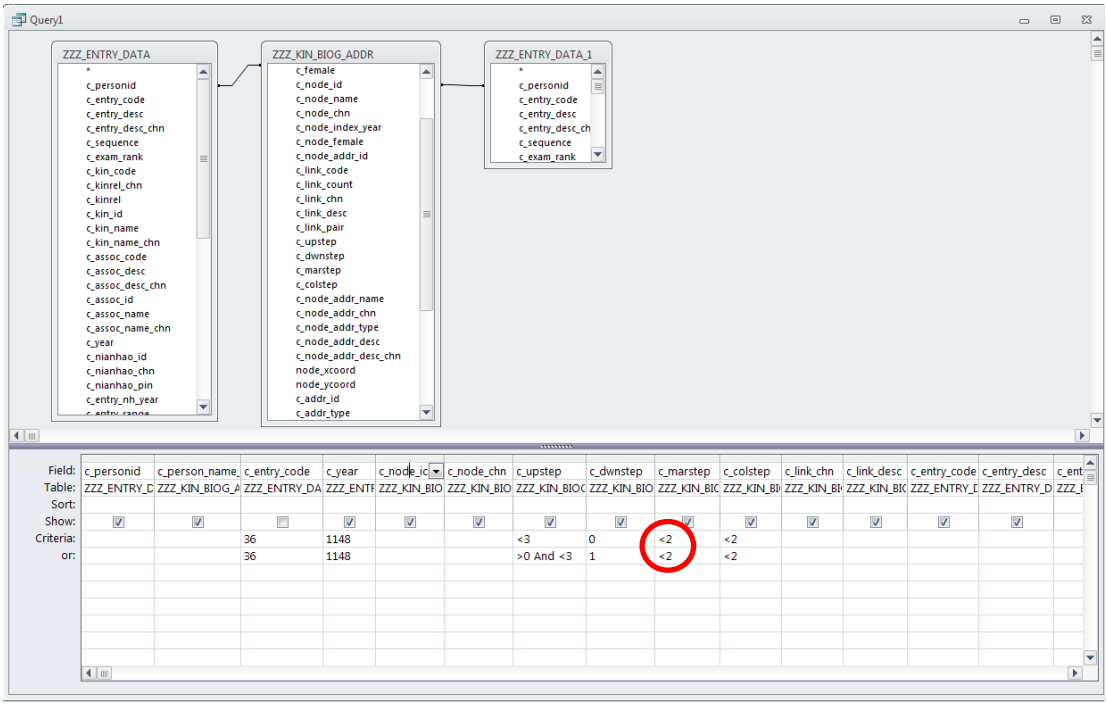


8. Add the two fields c_entry_desc and c_entry_desc_chn from ZZZ_ENTRY_DATA_1 (to get the mode of entry of the kin) and check the results:

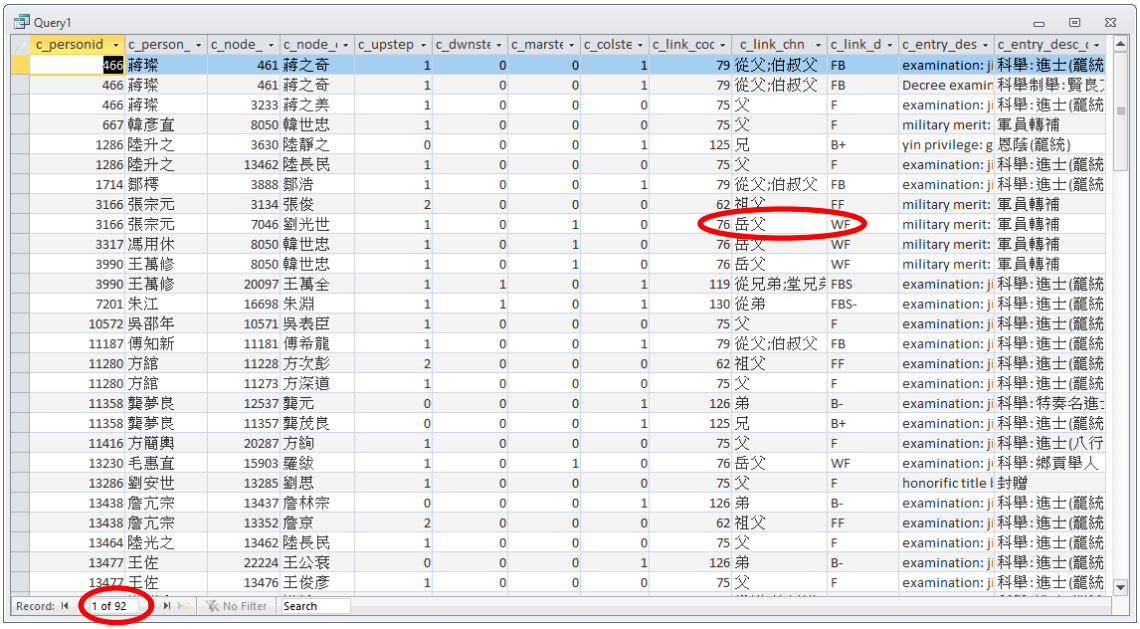


86 kin from the 273 initial degree recipients have data on how they entered officialdom

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.

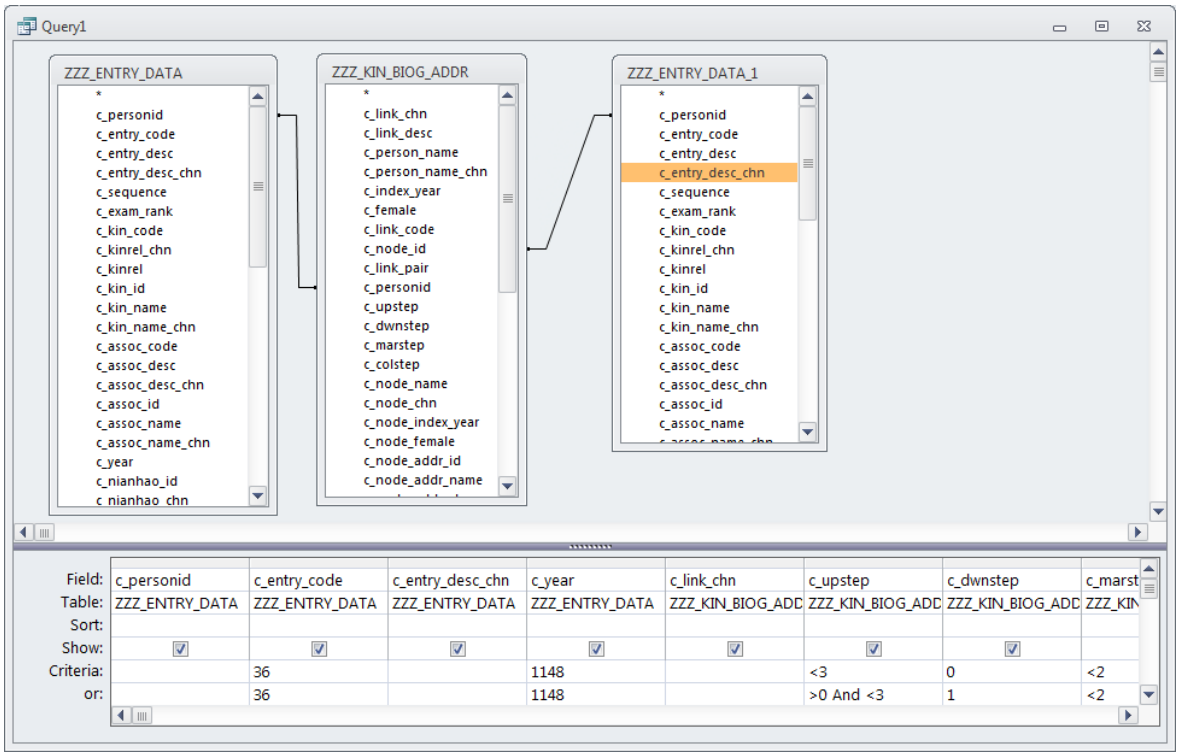


II. Some Useful Additional Procedures for Queries

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.

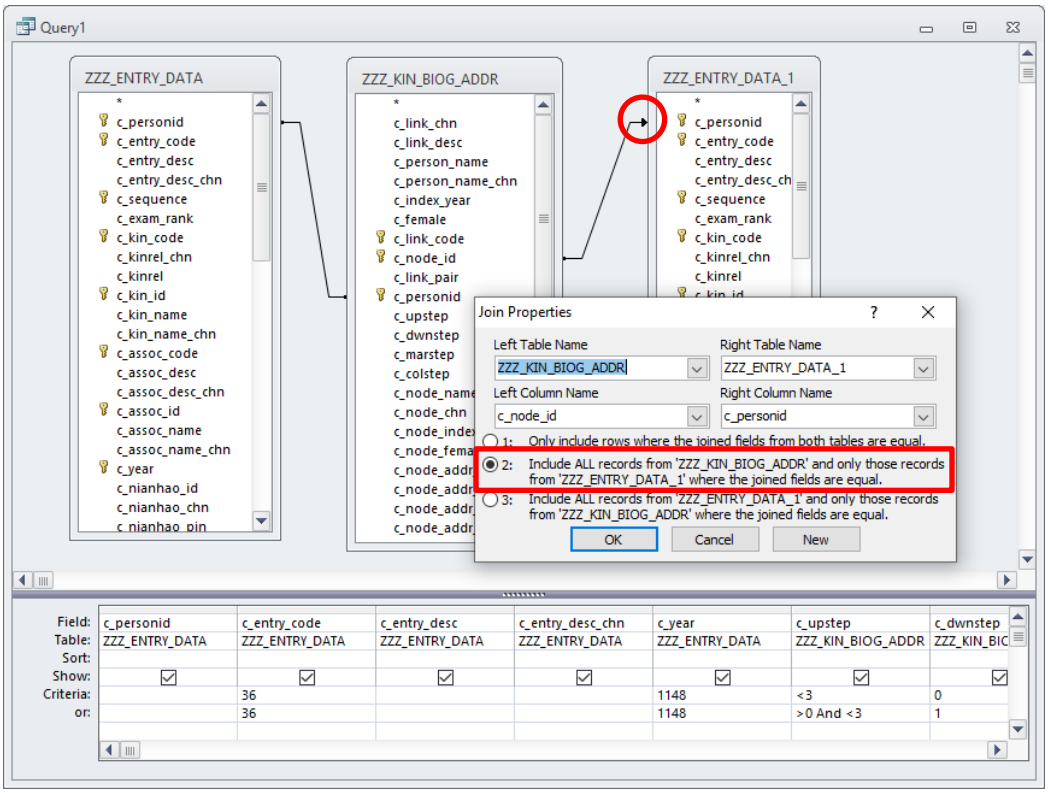
Our initial design looked like:



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:



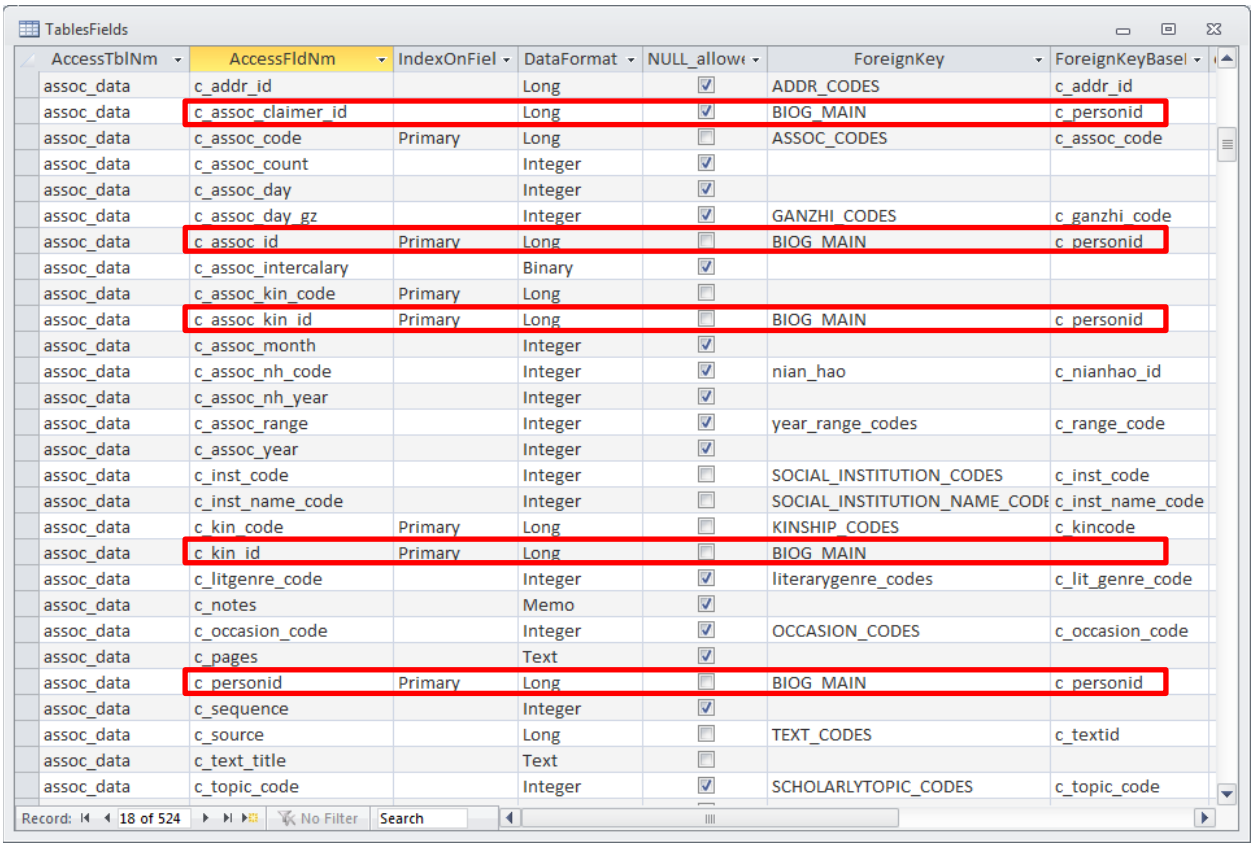
Select option 2 and click OK. Note the arrow pointing to c_personid. This arrow indicates a "left join" in the language of SQL. This **left join** includes all the records from ZZZ_KIN_BIOG_ADDR (the left table) that match the other query criteria as well as the fields from ZZZ_ENTRY_DATA_1 (the right 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 a refer to people.[1] For example, in ASSOC_DATA, we have:

[1] 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)

Appendix: Installing CBDB Files

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.

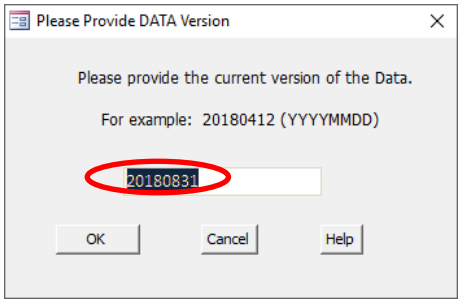
2. Double-click on the User file to open it in Microsoft Access. You will see:

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_20180831_DATA1.mdb. The date "20180831" (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.

B. Updating the Visual Basic Environment (if necessary)

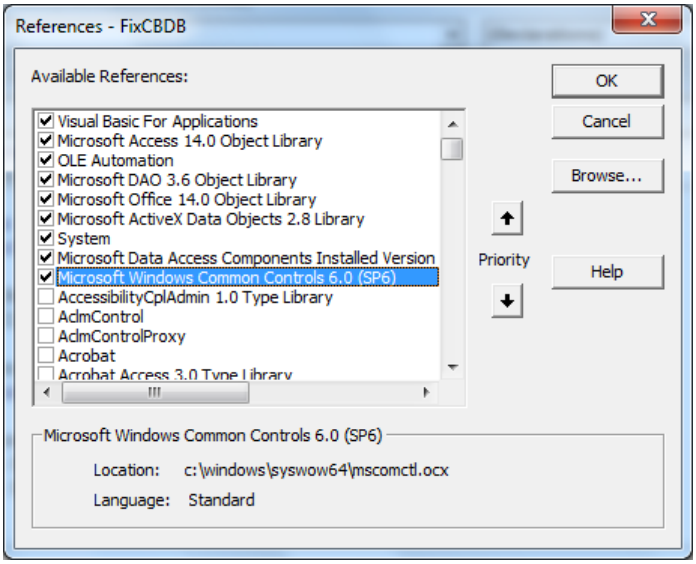
Adding References

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.

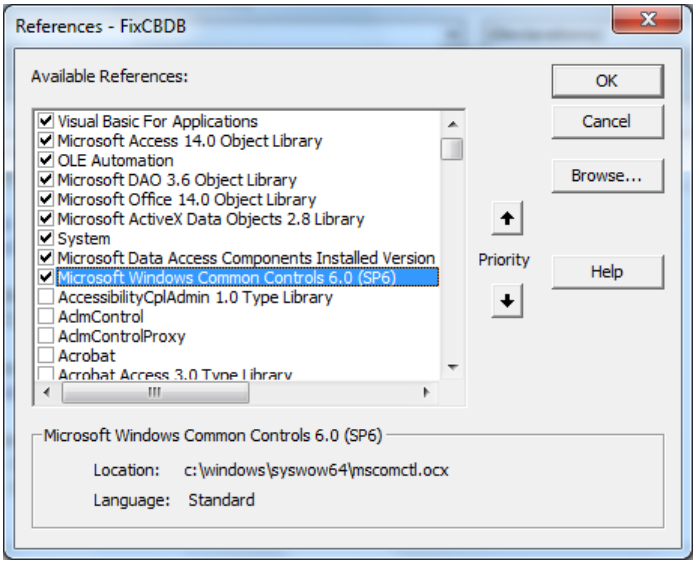
Adding TreeView to Visual Basic

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.

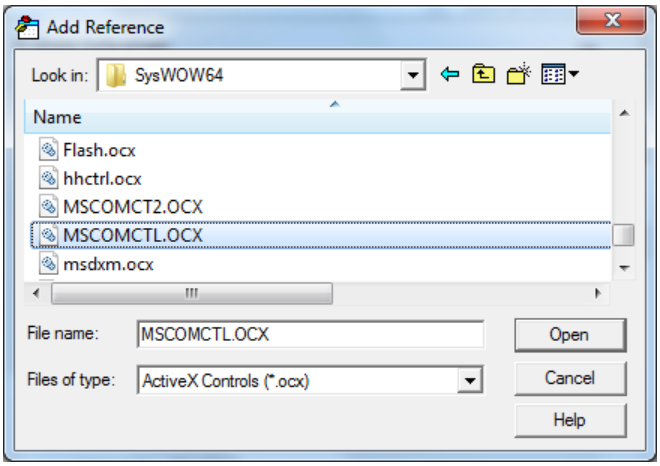
2. In the VB editor, click on the menu item "Tools" and then "References…" You will see something like:



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."

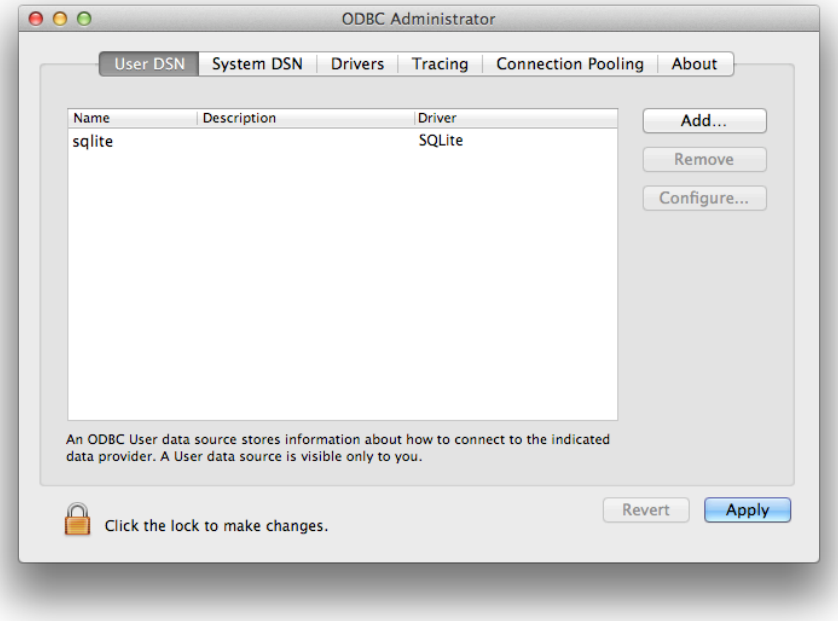
C. Installing the SQLite CBDB database on a Macintosh

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.

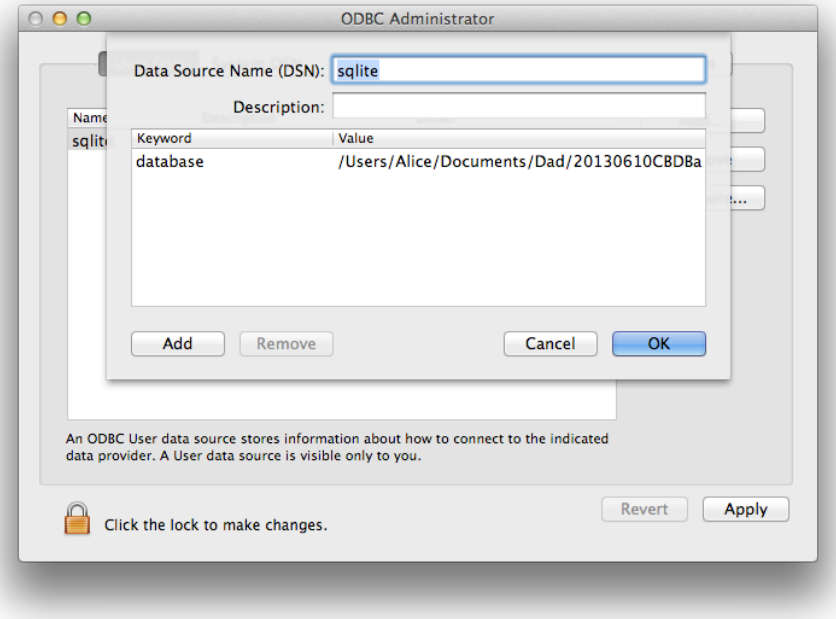
For any database file to be used in a Macintosh system, the operating system needs a connector between the file and the standard ODBC (Open Database Connectivity) interface. In order to make this connection, you will need the Mac ODBC Administrator and the

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.