**Report on the “Hokkien” project for the “Working with Digital Data for Historians” class by Sebestyen Hompot**

1. **Excel spreadsheets**

First of all, regarding the contents of the submitted work I would note that during my studies in China I investigated the history of Hokkien-Foreign linguistic interaction until the end of the Chinese Civil War (ca. 1950) only, therefore at the moment I am not familiar with works published in the post-Civil War period. For this reason, to works published in the post-Civil War period I did not add any new information apart from what appears in Chiung’s (2015)[[1]](#footnote-1) table (title, year of publication, author, publisher, city of publication) and did not add new post-Civil War works. I had much more knowledge regarding pre-1950 works and based on that knowledge I decided to create the following tables: countries, cities, missionary\_societies, publishers, people (authors of the works), transcription\_methods, publications, pub\_author (a join table of the `people` and `publications` tables). I needed to add information not included in the Chiung table manually in Excel, including a large number of relevant works I came across during my research not appearing in that table.

Transporting the data from the Chiung article (.pdf format) into my excel file was not as difficult as I thought first. I managed to do it via Microsoft Word, which preserved the table format and the Asian characters, although without the separation lines of the pdf file. Text from Word could be selected and inserted into Excel on a cell-by-cell basis, without any change to the Asian characters. In order to demonstrate it I copy the first two rows of the Chiung table here:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1837 | Dictionary of the Hok-keen Dialect of the Chinese Languages, According to the Reading and Colloquial Idioms (福建方言字典) | W. H. Medhurst (麥都思) | 閩南語、英語 | 福建話 | 澳門 | Honorable East India Company (英國東印度公司) |
| 1838 | A vocabulary of the Hok-keen Dialect as spoken in the county of Tsheang-tshew (漳州語彙) | S. Dyer | 漳州話、英語 | 福建漳州話 | Malacca | Anglo-  Chinese College Press |
|  |  |  |  |  |  |  |

I converted title and author names in foreign key referenced columns into their respective ID’s inside Excel by using the “Replace all” function. I obtained the future auto-incremented ID’s for each item by adding a temporary column in each table next to the item’s name and auto-filling it with an increasing series starting from 1. Originally, I planned to have a ‘publisher’ column (INT, referenced to publishers.id) in the `publications` table, since the city and country of the publisher is included in the `publishers` table. However, since I wanted to make direct queries on the amount of publications in one given country or city, I added the ‘city\_pub’ and ‘country\_pub’ columns as well. Without this, queries to investigate the number of publications by country or city would need a triple join query, this way a double join query is enough (e.g. ‘SELECT countries.name, COUNT(pubications.id) FROM countries JOIN publications ON publications.country = countries.id GROUP BY countries.name ORDER BY COUNT(publications.id) DESC;).

1. **Transporting Excel data into SQL via Jupyter Notebook**

I created the database “hokkien” through MySQL Workbench using the script I provided

in my “project plan” earlier. Following this, I imported the Excel file into Jupyter Notebook following the method described in the “Excel in Python” notebook (import pandas as pd -> pd.ExcelFile(file\_name) -> xl.parse(‘sheet\_names’). Following this, however, it was not clear to me how is it possible to export the data into an SQL database. After some online search I found the following method, for which I first had to convert my Excel file from .xlsx format into its older version .xls and which ultimately proved to be unfit for my purposes:

1. import xlrd
2. book = xlrd.open\_workbook("hokkien.xls")
3. sheet\_ctr = book.sheet\_by\_name("countries")

# *Here I provide the example of the “countries” table, for which I used the abbreviation “ctr”. The other abbreviations I used in Jupyter Notebook are “cit” (cities), “mis” (missionary\_societies), “pbr” (publishers), “ppl” (people), “tra” (transcription\_methods), “pub” (publications), “pau” (pub\_author).*

1. import pymysql, cursor = […], cnx = […]
2. query\_ctr= ‘INSERT INTO countries ([column1, …]) VALUES (%s, …)’
3. for r in range (1, sheet\_ctr.nrows):

column1 = sheet\_ctr.cell(r,0).value

column2 = sheet\_ctr.cell(r,1).value

values\_ctr = ([column1, …])

cursor.execute(query\_ctr, values\_ctr)

cnx.commit()

This method produced several errors, most notably it was unable to correctly interpret empty cells as null values in the case of columns in the SQL database where the data type was set to INT (even though I did not put a NOT NULL constraint on most of those columns). Jupyter Notebook repeatedly displayed the error message “Incorrect integer value: ‘ ‘ for “column\_name” in row [number]”. There was no such problem with empty cells in cases where datatype was set to VARCHAR.

After further online search I found out that Pandas has a .to\_sql function to export data frames into SQL. Using this, I could smoothly accomplish my task through the steps documented in the attached notebook.

1. **Creating the dump version**

As far as I remember we did not discuss how to use the mysqldump function in MySQL Command Line or any program other than MySQL Workbench. I tried “mysqldump –u [root] –p hokkien (name of my database) > hokkien\_dump.sql” in MySQL Command Line and following some online suggestions in Windows Command Prompt and Windows Power Shell as well, but the general error message was that the program has no such function. Therefore I decided to create the dump version through MySQL Workbench.

1. **Future opportunities**

The database can be expanded in the future by adding further data, especially related to works published in the post-Civil War period. This database can also be used as a starting point of a larger database of Sino-Foreign interaction in the South China Sea region, a tool for research on the role of the region in global history. This larger database would be realized by expanding the thematic and temporal scope of the original database. On the one hand, separate tables for works other than Hokkien linguistic materials (e.g. travelogues) would be created. On the other hand, authors of dictionaries and other sorts of works were often the same, therefore a unified `people` table would be suitable for the expanded database as well. The temporal scope of the database could be enlarged to include works published in pre-colonial times characterized by trade between China and other parts of Asia (especially the Muslim world). This enlarged database could also be reformatted using Cypher for graphical display in Neo4J, in order to illustrate the complex relationship between actors, publications, places and organizations (see e.g. transregional missionary societies).

1. Chiung, Wi-vun Taiffalo 2015. “Taiwanese or Southern Min? On the Controversy of Ethnolinguistic Names in Taiwan” <http://uibun.twl.ncku.edu.tw/chuliau/lunsoat/english/2015/taiwanese.pdf> Appendix 1 (pp. 24-29, last update: 2014-07-27). [↑](#footnote-ref-1)