CSCI 5333.4 DBMS
Suggested Solution to Homework #5

(1)

(a) What format of the data have you downloaded and why?

The site provides various download options:

- Structured: XML: offer no advantage as the data is still flat and XML files need additional parsing.
- Value separated:
    - Excel CSV: not text file. Need to open in Excel and save before using. The format may be very good for analyzing some aspects of data. Loading database may take extra effort.
    - Semicolon, Pipe: text file, all values are double quoted. May be easier to be loaded by using the MySQL load statement. Data analysis can be aided by a good text editor.

We downloaded the Pipe version (data analysis and preparation) and the Excel version (used for part of data analysis).

(b) What properties have you found about the five datasets that can be used to prepare and simplify the data?

Some general observations:

- All five datasets have a flat structure and the same six columns: "Reference Area"|"Time Period"|"Sex"|"Age group"|"Units of measurement"|"Observation Value".
- Some columns are not appropriate for a given dataset (e.g. Sex and Age group for GNI and expenditure.)
- There are columns in which the values are all the same and they are candidates for removal.
- There are “0” in "Observation Value". They should be interpreted as null values, and not the integer 0.
- There is redundancy in the data.

(c) What mechanisms have you employed to clean and prepare the data and why?

We use text editors to clean the data because of its sufficiency and efficiency for the datasets. It also provides an easy way for populating the MySQL database.

The process and result of cleaning the five datasets are shown below.

Primary Education:

- An initial total of 3781 records: that means not all country-year pairs have both female and all gender data.
- Years range from 1999 to 2009.

- An empty line removed at the end of the file.
- All values of Age group are "Not applicable" => removed.
- All values of Unit of measurement are “Number” => removed.
- For “Sex”, use “F” for Female, “A” for “All genders” and “M” for “Male” for storage efficiency. The Pipe file is changed accordingly. 1900 records were changed to “A”. 1881 records were changed to “F”.
- 24 occurrence of “0”. That means missing information, and not really 0 students. Such records (e.g. "Afghanistan"|"2001"|"F"|"0") do not include any useful information. There are two approaches, delete the record or change “0” to “\N”, the notation for null values for MySQL load file format. We deleted the record.
- Final result: 1889 “A” records and 1868 “F” records.

Secondary Education:

- An initial total of 3387 records in which the value of genders can be female, male or all genders.
- Years range from 1999 to 2009.
- An empty line removed at the end of the file.
- All values of Age group are "Not applicable" => removed.
- All values of Unit of measurement are “Percent” => removed. Note that the dataset does not include the actual number of students.
- For “Sex”, use “F” for Female, “A” for “All genders” and “M” for “Male” for storage efficiency. The Pipe file is changed accordingly. 1163 records were changed to “A”. 1112 records were changed to “F”. 1112 records were changed to “M”.
- 36 occurrences of “0”. We deleted these records.
- Final result: 3351 records: 1151 all genders, 1100 females and 110 males.

Tertiary Education:

- An initial total of 4319 records.
- Years range from 1999 to 2009.
- An empty line removed at the end of the file.
- All values of Age group are "Not applicable" => removed.
- Counts of records with “Percent”: 1412; counts of records with “Number”: 2907. Note that there are 2824 female records. Thus, we conclude that every country-year pair with female entries comes in a pair of number and percentage. We used Excel to check and confirm that the percentage entry is computed from the number entries of “All genders” and “Female”. Thus, 1412 of these records are removed as they can be computed later. (Note: we used the regular expression ^".*Percent.*$ for the record deletion using the text editor.)
- For “Sex”, use “F” for Female, “A” for “All genders” and “M” for “Male” for storage efficiency. The Pipe file is changed accordingly. 1495 records were changed to “A”. 2824 records were changed to “F”. We deleted the record.
- 36 occurrence of “0”. These records were deleted.

- Data cleaning:

It is noted that there are data such as:

Argentina	2003	All genders	Not applicable	Number	2101437
Argentina	2003	Female	Not applicable	Number	1253533.737
Argentina	2003	Female	Not applicable	Percent	59.65126

The number of female students cannot be 1253533.737. It must be an integer. We round it to the closest integer. This is reasonable because:

- The percentages of female students of Argentina are similar in years close to 2003.
- The computed percentage from the all genders and female numbers is 59.65126423 so the same formula is used.

We cleaned two records this way:

"Argentina"|"2003"|"F"|"1253533.73747175" to

"Argentina"|"2003"|"F"|"1253534"

and

"Brazil"|"2003"|"F"|"2254291.44725734"

to

"Brazil"|"2003"|"F"|"2254292"

- 274 records with value “0” deleted.
- Final number of records: 1275 “F” records and 1358 “A” records.

GNI:

- A total of 1141 records.
- Year ranges from 1999 to 2009.
- Two empty lines removed at the end of the file.
- All values of Age group and gender are "Not applicable" => removed.
- All values of units are “Percent” and are removed.
- One value of “0” (Holy See) deleted.
- Final number of records: 931 records.

Expenditure:

- A total of 932 records: that means not all country-year pairs have both female and all gender data.

- Years range from 1999 to 2009.
- An empty line removed at the end of the file.
- All values of Age group and gender are "Not applicable" => removed.
- All values of units are “Percent” and are removed.
- Final number of records: 931 records.

(d) What are the relation schemas of the tables you have designed to store the data? Identify all candidate keys.

(e) What are the SQL commands you have used to create the tables and populate them?

The relation schemas are encapsulated by the SQL commands listed below. The candidate keys of the relations primary, secondary and tertiary are the same: {country, year, gender}. The candidate keys of the relations gni and expenditure are the same: {country, year}.

```
create table f1h5_primary (
  country varchar(53) not null,
  year integer(4) not null,
  gender char(1),
  prim_number integer(10) unsigned
);

LOAD DATA LOCAL INFILE "Primary_Pipe_Cleaned.txt"
INTO TABLE f1h5_primary
FIELDS ENCLOSED BY "\"" TERMINATED BY "|"
LINES TERMINATED BY '\r\n'
IGNORE 1 LINES
;

create table f1h5_secondary (
  country varchar(53) not null,
  year integer(4) not null,
  gender char(1),
  second_percent decimal(8,5) unsigned
);

LOAD DATA LOCAL INFILE "Secondary_Pipe_Cleaned.txt"
INTO TABLE f1h5_secondary
FIELDS ENCLOSED BY "\"" TERMINATED BY "|"
LINES TERMINATED BY '\r\n'
IGNORE 1 LINES
;

create table f1h5_tertiary (
  country varchar(53) not null,
  year integer(4) not null,
  gender char(1),
```

```
  tert_number integer(10) unsigned
);

LOAD DATA LOCAL INFILE "tertiary_Pipe_Cleaned.txt"
INTO TABLE f1h5_tertiary
FIELDS ENCLOSED BY "\"" TERMINATED BY "|"
LINES TERMINATED BY '\r\n'
IGNORE 1 LINES
;

create table f1h5_gni (
  country varchar(53) not null,
  year integer(4) not null,
  gni_percent decimal(8,5) unsigned
);

LOAD DATA LOCAL INFILE "gni_Pipe_Cleaned.txt"
INTO TABLE f1h5_gni
FIELDS ENCLOSED BY "\"" TERMINATED BY "|"
LINES TERMINATED BY '\r\n'
IGNORE 1 LINES
;

create table f1h5_expenditure (
  country varchar(53) not null,
  year integer(4) not null,
  exp_percent decimal(8,5) unsigned
);

LOAD DATA LOCAL INFILE "expenditure_Pipe_Cleaned.txt"
INTO TABLE f1h5_expenditure
FIELDS ENCLOSED BY "\"" TERMINATED BY "|"
LINES TERMINATED BY '\r\n'
IGNORE 1 LINES
;
```

Notes:

- The use of “IGNORE 1 LINES” to skip the column header line at the beginning of the cleaned file.
- The use of “LINES TERMINATED BY '\r\n'” to handle new line symbols in windows.

(f) What are the format of the cleaned data files you used to populate the tables, and why?

The formats of the primary, secondary and tertiary are the same using Pipe separated values:

"Reference Area"|"Time Period"|"Sex"|"Observation Value"

The formats of the gni and expenditure are the same using Pipe separated values:

"Reference Area"|"Time Period"|"Observation Value"

Bonus

Additional processing of the relations may lead to improvement. We merged all five relations into one relation with its schema encapsulated by the SQL statement below.

```
create table f1h5 (
  country varchar(53) not null,
  year integer(4) not null,
  female_primary integer(10) unsigned,
  all_primary integer(10) unsigned,
  female_secondary decimal(8,5) unsigned,
  male_secondary decimal(8,5) unsigned,
  all_secondary decimal(8,5) unsigned,
  female_tertiary integer(10) unsigned,
  all_tertiary integer(10) unsigned,
  gni_percent decimal(8,5) unsigned,
  exp_percent decimal(8,5) unsigned,
  unique key (country, year)
 );
```

The advantages of a single relation include faster processing (no join operations) and easier SQL statements for application development. In production, because of its small size and static in nature, the web server application may select to load the entire database into memory for performance optimization. It is also likely that the data is just a very small part of a large database and design adjustment is needed.

To populate the table, there are many approaches:

1. Use other tools to prepare a single text file containing all data and use the MySQL LOAD command to load it.
2. Write a script program to read the separate text files to load the master relation.
3. Write a script program to read the separate relations to load the master relation.
4. Write SQL commands to use the separate relations to load the master relation.

Each approach has pros and cons. As a demonstration, a command line PHP program is written to implement approach (3). The program also creates a single text file containing the data (which is the input for approach (1)).

The table ends up with 1986 records.