DATABASE DESIGN
Peru School System

Nicolas Fajardo for Stephanie Majerowicz

March 1, 2021

1. SOURCES

1.1. Sistema de Información de Apoyo a la Gestión de la Institución Educativa (SIAGIE)

This database provides three tables for each year 2013, 2014, and 2015 respectively. The variables are renamed as defined within the built-in dictionary included in the function `translate_names()`. Then, the reported ID number is converted to numeric, after deleting all non-digit characters. Sex is translated to a dummy 1 if female, 0 if male called "mujer". Finally, empty variables "grupo_unico", "sit_mat", and "sit_final" are dropped.

1.2. Evaluación Censal de Estudiantes (ECE)

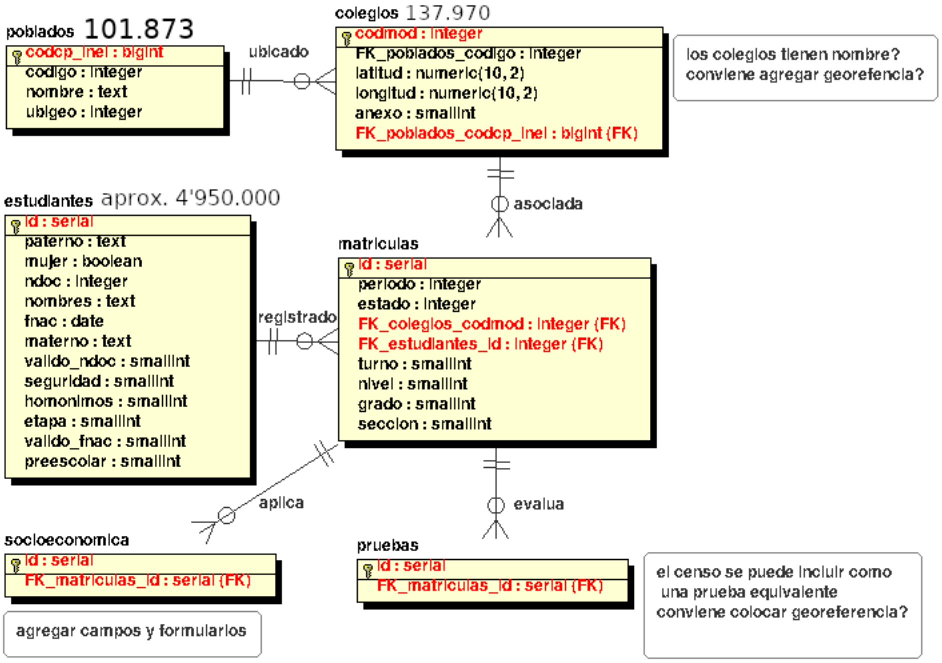
1.3. Sistema de Focalización de Hogares (SISFOH)

This source provides one table comprising approximately 24 million observations. The variables are renamed as defined within the built-in dictionary included in the function `translate_names()`. Then, the reported ID number is converted to numeric, after deleting all non-digit characters. Sex is translated to a dummy 1 if female, 0 if male called "mujer".

1.4. Estadística de la Calidad Educativa

1.5. Directorio Nacional de Centros Poblados

2. DESIGN



3. PROCEDURES

3.1. Identification of students

STEP 1: Depurate enrollment tables for each year by deleting observations with both missing ID and birthdate. Then, group by ID, father's surname, mother's surname, name, birthdate, and female, and create a variable called "translado_*", where * corresponds to each year, and equals to the number of appearances of each existing combination of the grouping variables minus 1.[1] Afterwards, the registry with the lowest matriculation status is chosen as well as those with ID status equal to 1.[2]

STEP 2: Create an enrollment table full-joining the previous tables yielded in step 1 by ID, father's surname, mother's surname, name, birthdate, and female. The joins preserves the code of the school, the branch (if applicable), the school grade and the matriculation status for each year (2013, 2014, and

[1] The idea behind this approach is that SIAGIE tables serve as a log of matriculation for all Peruvian schools, if a registry (person) appears more than once, it means that the matriculation status somehow changed, being a relocation between schools the most common reason for such duplicates.

[2] ID status is a variable indicating if the ID number was checked in the National Registry of Identification and Civil Status databases. However, even when is equal to 1 if ID number does exist in the official registries, it is no necessarily a name - ID match.

2015).

STEP 3: Create a variable that counts the number of existing homonyms (grouping by last names and name) and perform some information checks, assigning 1 if condition holds, 0 otherwise. *Check 1*, Student appears on all observed years (2013, 2014, 2015). *Check 2*: Student appears on 2013 and 2014, it is defined as most secure if the student is in last year of schooling, [$grado\_2014 = 16$]. *Check 3*: Student appears on 2014 and 2015, it is defined as most secure if student is in first year of schooling, [$grado\_2014 = 2$]. *Check 4:* Student appears on 2013 and 2015 but not in 2014, it is not secure, as possible name typos or mismatches exist.

STEP 4: Drop observations with missing either father's last name, mother's last name, names or female and keep all observations with unique ID numbers. Recalculate number of homonyms and drop all registries where the same combination of father's surname, mother's surname, name, birthdate, and female appears more than once.[3] Save the obtained output into a new table called "students_siagie". Number of students identified: $7,920,788$.

STEP 5 Perform a weak inner-join between each of the tables and table "students_siagie", and a weak full-join between the two outputs obtained before. Weak refers to the fact that nor ID number or birthdate are used in the join.[4] Afterwards, generate validity variables according to table 1 combinations. Finally, group by ID number, drop repeated observations and keep observations with *Validity of ID number* ≤ 3 and *Validity of birthdate* $\neq 4$,[5] set *Filter's stage* variable to 1 and copy to "student_id" table. Number of students identified: $3,975,594$. Those students appear at least in two sources simultaneously (between ECE, SISFOH and SIAGIE). In this particular step, the joins are more robust to ID number mismatches, since the probability of having a typo in any of the (last) names is higher than the probability of erring only the ID number, while also taking into account that for early ECE tests, no ID numbers were collected.

STEP 6: Perform a strong inner-join between each of the tables and table "students_siagie", and a weak full-join between the two outputs obtained before. Strong refers to the fact that only ID number and female are used in the join.[6] Afterwards, generate validity variables according to table 1 combinations. Finally, group by ID number, drop repeated observations and keep observations with *Validity of ID number* ≤ 3 and *Validity of birthdate* $\neq 4$, set *Filter's stage* variable to 2 and copy to "student_id" table. Number of students identified: $1,844,219$. Those students appear at least in two sources simultaneously (between ECE, SISFOH and SIAGIE).

Until this point $5,819,813$ unique students (SIAGIE) were identified, with at least one register either in ECE or SISFOH.

[3] If ID is added into the grouping, 233 more observations are recovered, meaning that less than 233 individuals share the same name, sex and birthdate. However, the grouping without ID is more robust to ID's mismatches or errors.

[4] Therefore the join variables are father's surname, mother's surname, name, and female.

[5] If *Validity of birthdate* ≤ 3 is chosen instead, 2637 less observations are identified.

[6] To ensure that the unique constraint on ID number in the "student_id" table is satisfied, all ID numbers identified in step 5, are eliminated before the strong inner-joins using an appropriate anti-join. Moreover, since names could differ between tables, even when ID numbers match, only those of SIAGIE are preserved. Finally, if two consecutive strong full-join are used, the resulting table is identical, since the registries on SIAGIE are identical, the only advantage is that with a weaker join the number of duplicates is reduced (more efficient).

DIFFERENCE OF BIRTHDATE	
Diff	Value
Identical	1
One unit (Upwards)	2
One unit (Downwards)	-2
Very different	3

VALIDITY OF BIRTHDATE			
SIAGIE	SISFOH	Diff	Value
Match	Match	–	1
Match	Match	2	2
Match	Match	-2	3
No match	No match	–	4
Not Missing	Missing	–	5

VALIDITY OF ID NUMBER			
SIAGIE	ECE	SISFOH	Value
Match	Match	Match	1
Match	Missing	Match	2
Match	Match	Missing	3
Match	Match	No match	4
Match	No match	Match	5
No match	Match	Match	6
No match	No match	Missing	7
No match	Missing	No match	8
No match	No match	No match	9
Not Missing	Missing	Missing	10

Table 1 If two or more “Match” appears in a row, the variable is equal in the referenced tables. “No match” refers that the reported variable is different from the other non-missing registries. If *Difference of birthdate* = ±2, it corresponds most likely to a typo, since the distance is exactly one time unit (year, month or day).