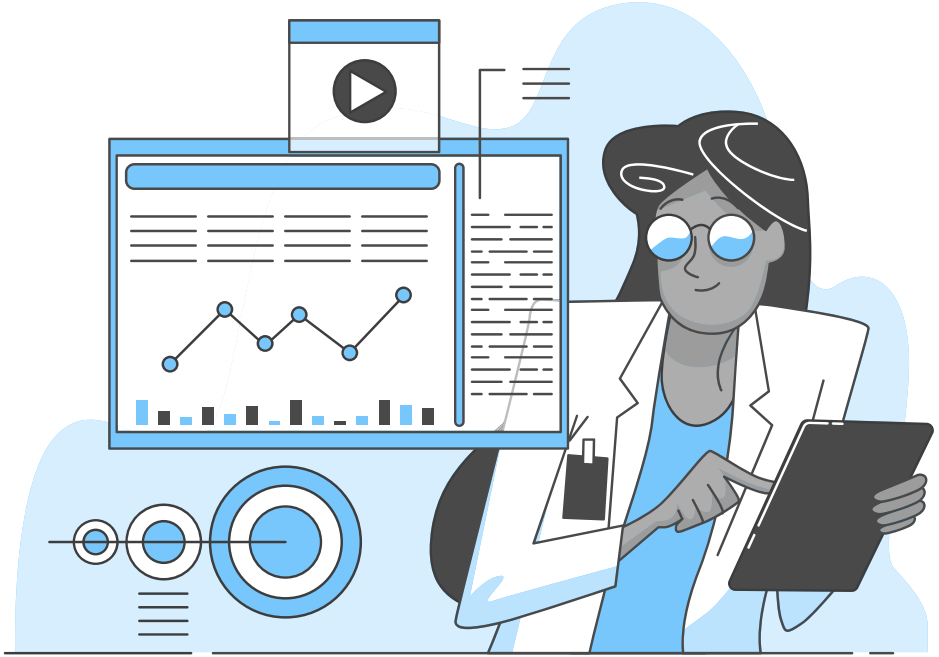
Exercise 5:

Create a database and family tree from XML file

Storage and Data Recovery
Master's Degree in Intelligent Systems

Iván López Muñoz

Table of Contents



Input – XML file data from 483 persons

```
<?xml version="1.0" encoding="UTF-8"?>
<people>
 - <person>
        <gender>D</gender>
        <name>Francisca</name>
        <alter>Ana</alter>
        <surname1>Figuerola</surname1>
        <surname2>Llompard</surname2>
        <birth>1817-1-17</birth>
        <fathername>Ramon</fathername>
        <fathersurname1>Figuerola</fathersurname1>
        <fathersurname2>Alberti</fathersurname2>
        <mothername>Cathalina</mothername>
        <mothersurname1>Llompard</mothersurname1>
        <mothersurname2>Rayo</mothersurname2>
        <father_grandfathername>Juan</father_grandfathername>
        <father_grandmothername>Theresa</father_grandmothername>
        <mother_grandfathername>Francisco</mother_grandfathername>
        <mother_grandmothername>Francisca</mother_grandmothername>
    </person>
 - <person>
        <gender>D</gender>
        <name>Esperanza</name>
        <alter>-</alter>
```

Parse XML file to SQL with python program

- Define table
- Extract and write data for each person
- Generate SQL insert statements

```python
import xml.etree.ElementTree as ET

# Parse the XML file
tree = ET.parse('AllPeople.xml')
root = tree.getroot()

# Open a file to write SQL statements
with open('people_data.sql', 'w') as sql_file:
    # Create table definition SQL statement
    table_definition = (
        "CREATE TABLE people_table ("
        "id INT AUTO_INCREMENT PRIMARY KEY, "
        "gender CHAR(1), "
        "name VARCHAR(255), "
        "alter_name VARCHAR(255), "
        "surname1 VARCHAR(255), "
        "surname2 VARCHAR(255), "
        "birth DATE, "
        "fathername VARCHAR(255), "
        "fathersurname1 VARCHAR(255), "
        "fathersurname2 VARCHAR(255), "
        "mothername VARCHAR(255), "
        "mothersurname1 VARCHAR(255), "
        "mothersurname2 VARCHAR(255), "
        "father_grandfathername VARCHAR(255), "
        "father_grandmothername VARCHAR(255), "
        "mother_grandfathername VARCHAR(255), "
        "mother_grandmothername VARCHAR(255)"
        ");\n"
    )
    sql_file.write(table_definition)

    # Iterate through each person in the XML
    for person in root.findall('person'):
        # Extract data for each person
        gender = person.find('gender').text
        name = person.find('name').text
        alter = person.find('alter').text if person.find('alter') is not None else ''
        surname1 = person.find('surname1').text
        surname2 = person.find('surname2').text
        birth = person.find('birth').text
        fathername = person.find('fathername').text
        fathersurname1 = person.find('fathersurname1').text
        fathersurname2 = person.find('fathersurname2').text
        mothername = person.find('mothername').text
        mothersurname1 = person.find('mothersurname1').text
        mothersurname2 = person.find('mothersurname2').text
        father_grandfathername = person.find('father_grandfathername').text
        father_grandmothername = person.find('father_grandmothername').text
        mother_grandfathername = person.find('mother_grandfathername').text
        mother_grandmothername = person.find('mother_grandmothername').text

        # Generate SQL insert statement
        sql_insert = (
            f"INSERT INTO people_table (gender, name, alter_name, surname1, surname2, birth, "
            f"fathername, fathersurname1, fathersurname2, mothername, mothersurname1, mothersurname2, "
            f"father_grandfathername, father_grandmothername, mother_grandfathername, mother_grandmothername) "
            f"VALUES ('{gender}', '{name}', '{alter}', '{surname1}', '{surname2}', '{birth}', "
            f"'{fathername}', '{fathersurname1}', '{fathersurname2}', '{mothername}', '{mothersurname1}', '{mothersurname2}', "
            f"'{father_grandfathername}', '{father_grandmothername}', '{mother_grandfathername}', '{mother_grandmothername}');\n"
        )

        # Write the SQL insert statement to the file
        sql_file.write(sql_insert)
```

Output – Text file with SQL data inserts

```
CREATE TABLE people_table
(id INT AUTO_INCREMENT PRIMARY KEY, gender CHAR(1),
name VARCHAR(255),
alter_name VARCHAR(255),
surname1 VARCHAR(255),
surname2 VARCHAR(255),
birth DATE,
fathername VARCHAR(255),
fathersurname1 VARCHAR(255),
fathersurname2 VARCHAR(255),
mothername VARCHAR(255),
mothersurname1 VARCHAR(255),
mothersurname2 VARCHAR(255),
father_grandfathername VARCHAR(255),
father_grandmothername VARCHAR(255),
mother_grandfathername VARCHAR(255),
mother_grandmothername VARCHAR(255));
```

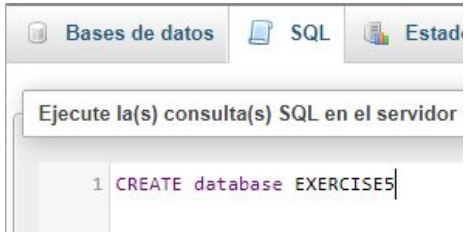
```
INSERT INTO people_table
(gender,
name,
alter_name,
surname1,
surname2,
birth,
fathername,
fathersurname1,
fathersurname2,
mothername,
mothersurname1,
mothersurname2,
father_grandfathername,
father_grandmothername,
mother_grandfathername,
mother_grandmothername)
VALUES ('D',
'Francisca',
'Ana',
'Figuerola',
'Llompard',
'1817-1-17',
'Ramon',
'Figuerola',
'Alberti',
'Cathalina',
'Llompard',
'Rayo',
'Juan',
'Theresa',
'Francisco',
'Francisca');
```

Steps to create a database with phpMyAdmin

- Start Apache and MySQL app in XAMPP Control Panel

- Search http://localhost/phpmyadmin/ in your internet browser

- Execute the next command and press continue:



- Select the database created in left panel

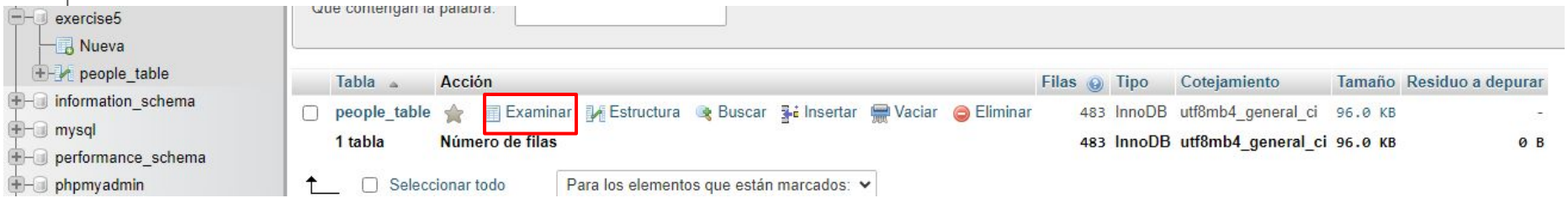
Insert all data in the database from previous output txt file

Estructura | SQL | Buscar | Generar una consulta | Exportar | Importar | Operaciones | Privilegios | Rutinas | Eventos | Disparadores | Seguimiento | Diseñador | Columnas cer

Ejecutar la(s) consulta(s) SQL en la base de datos exercise5:

```
1 CREATE TABLE people_table (id INT AUTO_INCREMENT PRIMARY KEY, gender CHAR(1), name VARCHAR(255), alter_name VARCHAR(255), surname1 VARCHAR(255), surname2 VARCHAR(255), birth DATE, fathername VARCHAR(255),
fathersurname1 VARCHAR(255), fathersurname2 VARCHAR(255), mothername VARCHAR(255), mothersurname1 VARCHAR(255), mothersurname2 VARCHAR(255), father_grandfathername VARCHAR(255), father_grandmothername
VARCHAR(255), mother_grandfathername VARCHAR(255), mother_grandmothername VARCHAR(255));
2 INSERT INTO people_table (gender, name, alter_name, surname1, surname2, birth, fathername, fathersurname1, fathersurname2, mothername, mothersurname1, mothersurname2, father_grandfathername,
father_grandmothername, mother_grandfathername, mother_grandmothername) VALUES ('D', 'Francisca', 'Ana', 'Figuerola', 'Llompard', '1817-1-17', 'Ramon', 'Figuerola', 'Alberti', 'Cathalina', 'Llompard', 'Rayo',
'Juan', 'Theresa', 'Francisco', 'Francisca');
3 INSERT INTO people_table (gender, name, alter_name, surname1, surname2, birth, fathername, fathersurname1, fathersurname2, mothername, mothersurname1, mothersurname2, father_grandfathername,
father_grandmothername, mother_grandfathername, mother_grandmothername) VALUES ('D', 'Esperanza', '-', 'Carbonell', 'Coch', '1817-1-21', 'Miguel', 'Carbonell', 'Ramis', 'Francisca', 'Coch', 'Bassa', 'Miguel',
'Cathalina', 'Pedro', 'Francisca');
4 INSERT INTO people_table (gender, name, alter_name, surname1, surname2, birth, fathername, fathersurname1, fathersurname2, mothername, mothersurname1, mothersurname2, father_grandfathername,
father_grandmothername, mother_grandfathername, mother_grandmothername) VALUES ('D', 'Margarita', 'Maria', 'Carbonell', 'Coch', '1817-1-21', 'Miguel', 'Carbonell', 'Ramis', 'Francisca', 'Coch', 'Bassa',
'Miguel', 'Cathalina', 'Pedro', 'Francisca');
5 INSERT INTO people_table (gender, name, alter_name, surname1, surname2, birth, fathername, fathersurname1, fathersurname2, mothername, mothersurname1, mothersurname2, father_grandfathername,
father_grandmothername, mother_grandfathername, mother_grandmothername) VALUES ('D', 'Rosa', 'Maria', 'Barcelo', 'Gual', '1817-1-21', 'Vicente', 'Barcelo', 'Bernat', 'Cathalina', 'Gual', 'Ferrer', 'Andres',
'Margarita', 'Miguel', 'Rosa');
```

If the command is correct we are able to visualize the table by clicking 'Examinar':



We can visualize all information in the table and that allows us to clean the mistakes in the data

```
SELECT * FROM `people_table`
```

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]

1 > >> | Mostrar todo | Número de filas: 25 Filtrar filas: Buscar en esta tabla Ordenar según la clave: Ninguna

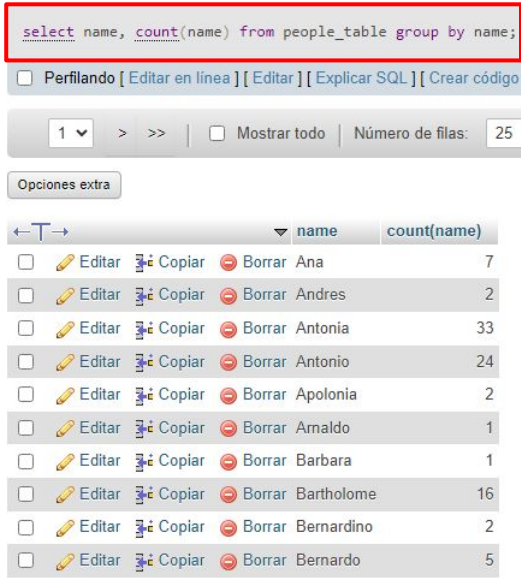
Opciones extra

		id	gender	name	alter_name	surname1	surname2	birth	fathername	fathersurname1	fathersurname2	mothername	mothersurname1	mothersurname2	father_grandfathername	father_grand
☐	Editar Copiar Borrar	1	D	Francisca	Ana	Figuerola	Llompard	1817-01-17	Ramon	Figuerola	Alberti	Cathalina	Llompard	Rayo	Juan	Theresa
☐	Editar Copiar Borrar	2	D	Esperanza	-	Carbonell	Coch	1817-01-21	Miguel	Carbonell	Ramis	Francisca	Coch	Bassa	Miguel	Cathalina
☐	Editar Copiar Borrar	3	D	Margarita	Maria	Carbonell	Coch	1817-01-21	Miguel	Carbonell	Ramis	Francisca	Coch	Bassa	Miguel	Cathalina
☐	Editar Copiar Borrar	4	D	Rosa	Maria	Barcelo	Gual	1817-01-21	Vicente	Barcelo	Bernat	Cathalina	Gual	Ferrer	Andres	Margarita
☐	Editar Copiar Borrar	5	H	Bartholome	Juan, Pedro, Gabriel, Abdon, Sennen	Marti	Pascual	1817-01-22	Sebastian	Marti	Pol	Antonia	Pascual	Alemañy	Juan	Antonia
☐	Editar Copiar Borrar	6	D	Francisca	Ana, Maria	Pujades	Llompard	1817-01-27	Juan	Pujades	Matheu	Cathalina	Llompard	Coll	Antonio	Antonia
☐	Editar Copiar Borrar	7	D	Cathalina	Maria	Rexach	Bertran	1817-01-01	Juan	Rexach	Segui	Cathalina	Bertran	Llabres	Miguel	Cathalina
☐	Editar Copiar Borrar	8	D	Antonia	-	Estrañy	Pou	1817-01-04	Jose	Estrañy	Llompard	Antonia	Pou	Gari	Bartholome	Margarita
☐	Editar Copiar Borrar	9	H	Gabriel	Abdon, Sennen	Matheu	Llompard	1817-02-04	Miguel	Matheu	Pujol	Francisca	Llompard	Llinas	Rafael	Francisca
☐	Editar Copiar Borrar	10	H	Matheu	Mariano	Truyol	Melis	1817-02-14	Juan	Truyol	Santandreu	Juana	Melis	Perello	Gonzales	Maria

How can we clean our data?

- Filter names to find those that are similar
  - The phpMyAdmin page itself allows us to filter the database table with different options

- Count the names of all persons and those that are similar, update them to the version that has bigger number
  - select name, count(name) from people_table group by name;
  - update person set name="Theresa" where name="Teresa";



Data updates in our database:

- Wrong dates -> births on the 31st with months that do not have 31 days
- Similar names and surnames have been changed to the biggest number one:

Names:

- Catalina 10 -> Cathalina 268
- Jaime 4 -> Jayme 118
- Bartolome 7 -> Bartholome 106
- Anna 3 -> Ana 403
- Sense nom -> " "
- Josef 56, Joseph 3 -> Jose 60
- Margharita 6 -> Margarita 142
- Rafel 2 -> Rafael 42
- Mateo 2, Matheo 31 -> Matheu 49

Surnames:

- Llompart 15 -> Llompard 88
- Beltran 10 -> Bertran 108
- Desconegut 100 -> " "
- x 1 -> " "

Once we have our cleaned database we are able to create a family tree with the following conditions:

- Split table into 2 different tables: ‘Child’ and ‘Family’

- Create relationship between:

  - Child – Parents
  - Parents – Grandparents

The problem is that parents have to belong to ‘child’ table and ‘family’ table, since they are parents of their children and they themselves are children of their grandparents.

To solve it, like our tree is following Spanish surnames rules tradition, child table will contain each child name and surnames of their parents, and also parent name and the first surname. Family's table will contain parents names and both surnames, grandparents name and first surname.

Create both tables in our database:

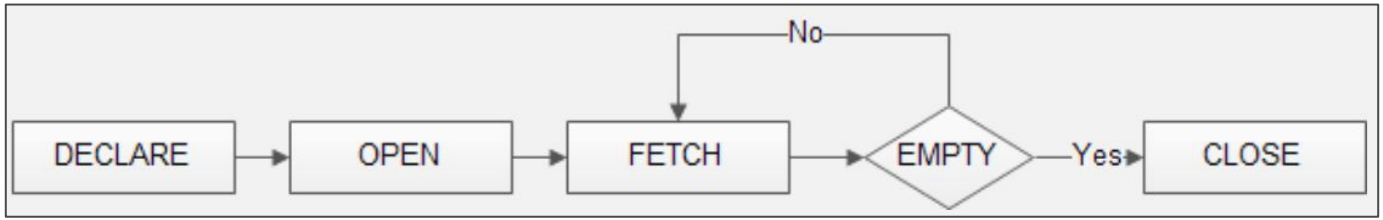
- Child table: child name and surnames / parents name and first surname
- Family table: parents name and surnames / grandparents name and first surname

```sql
CREATE TABLE child (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(255),
    surname1 VARCHAR(255),
    surname2 VARCHAR(255),
    father_name VARCHAR(255),
    father_surname1 VARCHAR(255),
    mother_name VARCHAR(255),
    mother_surname1 VARCHAR(255)
);
```

```sql
CREATE TABLE family (
    id INT AUTO_INCREMENT PRIMARY KEY,
    father_name VARCHAR(255),
    father_surname1 VARCHAR(255),
    father_surname2 VARCHAR(255),
    father_grandfather_name VARCHAR(255),
    father_grandfather_surname1 VARCHAR(255),
    father_grandmother_name VARCHAR(255),
    father_grandmother_surname1 VARCHAR(255),
    mother_name VARCHAR(255),
    mother_surname1 VARCHAR(255),
    mother_surname2 VARCHAR(255),
    mother_grandfather_name VARCHAR(255),
    mother_grandfather_surname1 VARCHAR(255),
    mother_grandmother_name VARCHAR(255),
    mother_grandmother_surname1 VARCHAR(255)
);
```

Split our database into both tables

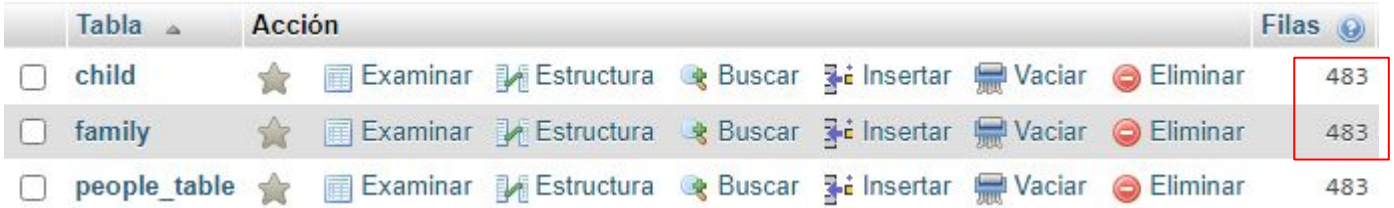
- To be able to execute this function for multiple rows we have to create a stored procedure function with a cursor to iterate through multiple query results:
  - DECLARE with all variables to hold the fetched data
  - CURSOR that allows to assign a name to a SELECT into a Procedure and manipulate the SELECT result
  - FETCH retrieves the next row from the cursor and assigns the values to variables



Stored procedure function with a cursor:

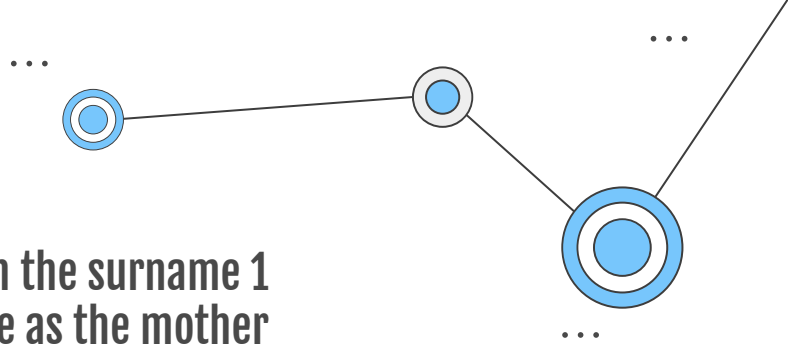
```
DELIMITER //
CREATE PROCEDURE SplitDataToTables()
BEGIN
    DECLARE done INT DEFAULT FALSE;
    DECLARE child_id INT;
    DECLARE child_name VARCHAR(255);
    DECLARE child_surname1 VARCHAR(255);
    DECLARE child_surname2 VARCHAR(255);
    DECLARE father_name VARCHAR(255);
    DECLARE father_surname1 VARCHAR(255);
    DECLARE father_surname2 VARCHAR(255);
    DECLARE mother_name VARCHAR(255);
    DECLARE mother_surname1 VARCHAR(255);
    DECLARE mother_surname2 VARCHAR(255);
    DECLARE father_grandfather_name VARCHAR(255);
    DECLARE father_grandfather_surname1 VARCHAR(255);
    DECLARE father_grandmother_name VARCHAR(255);
    DECLARE father_grandmother_surname1 VARCHAR(255);
    DECLARE mother_grandfather_name VARCHAR(255);
    DECLARE mother_grandfather_surname1 VARCHAR(255);
    DECLARE mother_grandmother_name VARCHAR(255);
    DECLARE mother_grandmother_surname1 VARCHAR(255);

    DECLARE cur CURSOR FOR
    SELECT
        id,
        name,
        surname1,
        surname2,
        fathername,
        fathersurname1,
        fathersurname2,
        mothername,
        mothersurname1,
        mothersurname2,
        father_grandfathername,
        father_grandfather_surname1,
        father_grandmothername,
        father_grandmother_surname1,
        mother_grandfathername,
        mother_grandfather_surname1,
        mother_grandmothername,
        mother_grandmother_surname1
    FROM people_table;

    DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;
    OPEN cur;

    read_loop: LOOP
        FETCH cur INTO
            child_id,
            child_name,
            child_surname1,
            child_surname2,
            father_name,
            father_surname1,
            father_surname2,
            mother_name,
            mother_surname1,
            mother_surname2,
            father_grandfather_name,
            father_grandfather_surname1,
            father_grandmother_name,
            father_grandmother_surname1,
            mother_grandfather_name,
            mother_grandfather_surname1,
            mother_grandmother_name,
            mother_grandmother_surname1;

        IF done THEN
            LEAVE read_loop;
        END IF;

        -- Insert data into child table
        INSERT INTO child (
            name,
            surname1,
            surname2,
            father_name,
            father_surname1,
            mother_name,
            mother_surname1
        )
        VALUES (
            child_name,
            child_surname1,
            child_surname2,
            father_name,
            father_surname1,
            mother_name,
            mother_surname1
        );

        -- Insert data into family table
        INSERT INTO family (
            father_name,
            father_surname1,
            father_surname2,
            father_grandfather_name,
            father_grandfather_surname1,
            father_grandmother_name,
            father_grandmother_surname1,
            mother_name,
            mother_surname1,
            mother_surname2,
            mother_grandfather_name,
            mother_grandfather_surname1,
            mother_grandmother_name,
            mother_grandmother_surname1
        )
        VALUES (
            father_name,
            father_surname1,
            father_surname2,
            father_grandfather_name,
            father_surname1,
            father_grandmother_name,
            father_surname2,
            mother_name,
            mother_surname1,
            mother_surname2,
            mother_grandfather_name,
            mother_surname1,
            mother_grandmother_name,
            mother_surname2
        );
    END LOOP;

    CLOSE cur;
END //

DELIMITER ;
```

Execute procedure:



- As we can see, now our ‘child’ and ‘family’ tables contain the data that was stored in our main table ‘people_table’

Child table results

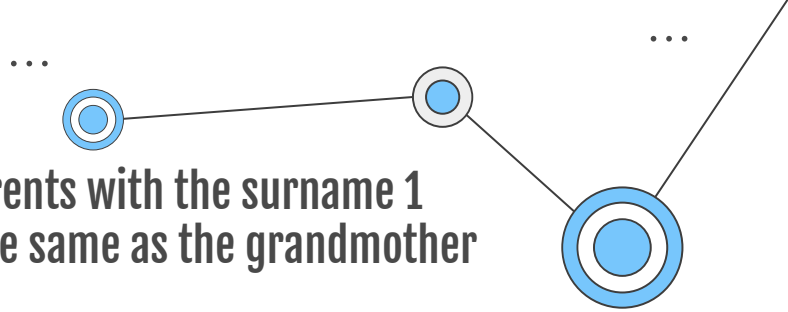


- We can see the relationship between child and parents with the surname 1 being the same as the father and surname 2 being the same as the mother

id	name	surname1	surname2	father_name	father_surname1	mother_name	mother_surname1
1	Francisca	Figuerola	Llompard	Ramon	Figuerola	Cathalina	Llompard
2	Esperanza	Carbonell	Coch	Miguel	Carbonell	Francisca	Coch
3	Margarita	Carbonell	Coch	Miguel	Carbonell	Francisca	Coch
4	Rosa	Barcelo	Gual	Vicente	Barcelo	Cathalina	Gual
5	Bartholome	Marti	Pascual	Sebastian	Marti	Antonia	Pascual
6	Francisca	Pujades	Llompard	Juan	Pujades	Cathalina	Llompard
7	Cathalina	Rexach	Bertran	Juan	Rexach	Cathalina	Bertran

- It also shows childs that are brothers or sisters of the same parents
- It solves the problem of parents being also childs of their parents

Family table results



- We can see the relationship between parents and grandparents with the surname 1 being the same as the grandfather and surname 2 being the same as the grandmother

id	father_name	father_surname1	father_surname2	father_grandfather_name	father_grandfather_surname1	father_grandmother_name	father_grandmother_surname1
1	Ramon	Figuerola	Alberti	Juan	Figuerola	Theresa	Alberti
2	Miguel	Carbonell	Ramis	Miguel	Carbonell	Cathalina	Ramis
3	Miguel	Carbonell	Ramis	Miguel	Carbonell	Cathalina	Ramis
4	Vicente	Barcelo	Bernat	Andres	Barcelo	Margarita	Bernat
5	Sebastian	Marti	Pol	Juan	Marti	Antonia	Pol

mother_name	mother_surname1	mother_surname2	mother_grandfather_name	mother_grandfather_surname1	mother_grandmother_name	mother_grandmother_surname1
Cathalina	Llompard	Rayo	Francisco	Llompard	Francisca	Rayo
Francisca	Coch	Bassa	Pedro	Coch	Francisca	Bassa
Francisca	Coch	Bassa	Pedro	Coch	Francisca	Bassa
Cathalina	Gual	Ferrer	Miguel	Gual	Rosa	Ferrer
Antonia	Pascual	Alemañy	Juan	Pascual	Margarita	Alemañy