ETL Project

Extract:

(a) We are using Barcelona data sets (Administration, Urban environment, Population, Territory, Economy and Business) from KAGGLE:
https://www.kaggle.com/xvivancos/barcelona-data-sets/version/2

(b) Download csv data files related to Barcelona demography:

- ***births.csv:*** Births by districts and by neighborhoods of the city (2013-2017).
- ***deaths.csv****:* Deaths by quinquennial ages and by neighborhoods of the city (2015-2017).
- ***population.csv:*** Population by neighborhood, by quinquennial ages and by genre of the city (2013-2017).
- ***unemployment.csv***: Registered unemployment by neighborhoods and genre in the city of Barcelona (2013-2017).

(c) Extract CSVs into DataFrames using Python Pandas library:

- ***births.csv*** => ***births_file_df***
- ***deaths.csv*** => ***deaths_file_df***
- ***population.csv => population_file_df***
- ***unemployment.csv => unemployment_file_df***

Transform:

(a) Data Frame ***births_file_df*** and ***population_file_df*** have data from the year **2013-2017** and data frame ***deaths_file_df*** has data from the year **2015 to 2017**.
Action: Remove all the rows for Years **2013** and **2014** from ***births_file_df*** and ***population_file_df***

(b) Data Frame ***population_file_df*** and ***deaths_file_df*** has **Age** Column with category **0-4**. Data Frame ***births_file_df*** does not have **Age** Column. Without Age column in ***births_file_df***, new births cannot be counted as 0-4 category.
Action: Add a new column **Age** in data frame ***births_file_df***. Fill all rows with **0-4** category.

(c) Rename ***births_file_df*** columns:

Year => year
District Code => district_code
District Name => district_name
Neighborhood Code => neighbor_code
Neighborhood Name => neighbor_name
Gender => gender
Number => number

(d) Rename ***deaths_file_df*** columns:
Year => year
District Code => district_code

District Name => district_name
Neighborhood Code => neighbor_code
Neighborhood Name => neighbor_name
Age => age
Number => number

(e) Rename ***population_file_df*** columns:
District Code => district_code
District Name => district_name
Neighborhood Code => neighbor_code
Neighborhood Name => neighbor_name
Gender => gender
Age => age
Number => number

(f) Rename ***unemployment_file_df*** columns:
Year => Year
District Code => district_code
District Name => district_name
Neighborhood Code => neighbor_code
Neighborhood Name => neighbor_name
Number => number

Load:

* All the queries can be found in the Github repository

(a) Using MySQL Workbench create database: **barcelona_demogaphy_db**

(b) Using MySQL Workbench put **barcelona_demogaphy_db** to use.

(c) Using MySQL Workbench create table **birth** as follows:

```
CREATE TABLE birth (
   year INT,
   district_code INT,
   district_name TEXT,
   neigbhor_code INT,
   neigbhor_name TEXT,
   gender TEXT,
   number INT,
  age TEXT
);
```

(d) Using MySQL Workbench create table **death** as follows:

```
CREATE TABLE death (
    year INT,
    district_code INT,
    district_name TEXT,
    neigbhor_code INT,
    neigbhor_name TEXT,
    age TEXT,
    number INT
);
```

(e) Using MySQL Workbench create table **population** as follows:

```
CREATE TABLE population (
    year INT,
    district_code INT,
    district_name TEXT,
    neigbhor_code INT,
    neigbhor_name TEXT,
    gender TEXT,
    age TEXT,
    number INT
);
```

(f) Using MySQL Workbench create table **unemployment** as follows:

```
CREATE TABLE unemployment (
    year INT,
    district_code INT,
    district_name TEXT,
    neigbhor_code INT,
    neigbhor_name TEXT,
    number INT
);
```

(g) Using Python create connection engine to MySQL to connect to **barcelona_demogaphy_db.** Use **sqlalchemy. create_engine.table_names** method to check for tables names**.**

(h) Verify the existing tables: **birth**, **death**, **population** and **unemployment**

(i) Truncate all the four tables before uploading data.

(j) Use pandas to load data frame into MySQL **barcelona_demogaphy_db** database:
- **birth_file_df => birth**
- **death_file_df => death**
- **population_file_df => population**
- **unemployment_file_df => unemployment**

(k) Confirm data has been added by querying **birth**, **death**, **population** and **unemployment** table.

(l) Summarize data from the four tables, and create view **pop_summary** and **umployment_data.**

SQL QUERIES:

Perform SQL Queries:

1)

```
/*Create the summary view to include the results for all the tables*/
create view pop_summary as (
SELECT
   birth_group.year,
   birth_group.district_code,
   birth_group.district_name,
   birth_group.neigbhor_code,
   birth_group.neigbhor_name,
   birth_group.number AS birth_number,
   death_group.number AS death_number,
   population_group.number AS total_num
FROM
          /*Group the birth table by year, district and neigbhor*/
   /*Take grouped data as the new birth_group table*/
   (SELECT
      birth.year,
         birth.district_code,
         birth.district_name,
         birth.neigbhor_code,
         birth.neigbhor_name,
         SUM(birth.number) AS number
   FROM
      birth
   GROUP BY birth.year , birth.district_code , birth.district_name , birth.neigbhor_code ,
birth.neigbhor_name) birth_group,
          /*Group the death table by year, district and neigbhor*/
   /*Take the grouped data as the new death table*/
   (SELECT
      death.year,
         death.district_code,
         death.district_name,
         death.neigbhor_code,
         death.neigbhor_name,
         SUM(death.number) AS number
   FROM
      death
```

```
    GROUP BY death.year , death.district_code , death.district_name , death.neigbhor_code ,
  death.neigbhor_name) death_group,
          /*Group the population table by year, district and neigbhor*/
    /*Take the grouped data as the new population table*/
    (SELECT
      population.year,
        population.district_code,
        population.district_name,
        population.neigbhor_code,
        population.neigbhor_name,
        SUM(population.number) AS number
    FROM
      population
    GROUP BY population.year , population.district_code , population.district_name ,
  population.neigbhor_code , population.neigbhor_name) population_group
  WHERE
    birth_group.district_code = death_group.district_code
      AND birth_group.neigbhor_code = death_group.neigbhor_code
      AND birth_group.year = death_group.year
      AND population_group.district_code = birth_group.district_code
      AND population_group.neigbhor_code = birth_group.neigbhor_code
      AND population_group.year = birth_group.year
  );
```

2)

```
create view umployment_data as (
SELECT
  u.year,
  u.district_code,
  u.district_name,
  u.neigbhor_code,
  u.neigbhor_name,
  SUM(u.number)
FROM
  unemployment u
GROUP BY u.year , u.district_code , u.district_name , u.neigbhor_code , u.neigbhor_name
order by u.year , u.district_code , u.neigbhor_code
);
```