ECS 116 Databases for Non-Majors / Data Management for Data Science Programming Assignment 1

A. Prelude

- 1. The assignment is of 25 points.
- 2. Last date of submission is April 26, 2024, Friday @ 11:59 pm.
- 3. Late submissions will be graded according to the late policy. Specifically, 10% of grade is deducted if you are up to 24 hours late, 20% is deducted if you are 24 to 48 hours late, and no credit if turned in after 48 hours.
- 4. This assignment will be solo.
- 5. Create a new sql file for each step namely (Step_2, Step_3, Step_4) if you have to use sql commands through DBeaver.
- 6. Your assignment will be graded based on correctness (passing all tests), ingenuity and originality.
- 7. All the required files (csv) can be found under Files in Canvas.
- 8. Plagiarism is strictly prohibited. You're free to discuss high-level concepts amongst your peers. However, cheating will result in no points on the assignment and reporting to OSSJA.

B. Step 1: Uploading $africa_fs_after_cleaning_db.csv$ into PostgreSQL

- 1. In DBeaver create a new database faostat. Set that as the default database
- 2. Create a schema food_sec (or "food_sec_v01") in your database faostat. Set that as default schema.
- 3. Do set search_path to food_sec;
- 4. Load the file africa_fs_after_cleaning_db.csv into the schema food_sec to make table africa_fs_ac.
- 5. Modify the data types of some of the columns of africa_fs_ac as follows:
 - area_code_m49: varchar(3)
 - element_code: varchar(4)
 - year_code: varchar(8)
 - value: numeric
 - After making these changes, click on "Save" at bottom of pane.
- 6. Check whether the values for value column have been imported correctly.
 - Do a selection query to get distinct values that are ≤ 2 .
 - Using Excel see what are the values ≤ 2 .
 - Do these match?
- 7. Do an SQL query to DELETE all tuples from **africa_fs_ac** (it will ask you to confirm that you want to do this delete).
- 8. Use DBeaver to *import* the file **africa_fs_after_cleaning_db.csv** (don't use the SQL "COPY" command because it complains about a data type encoding issue).
 - Do a sanity check that the number of tuples in your table is same as in csv file.
 - Again check on the values in column value.

C. Step 2: Build Table $gdp_stunting_overweight_anemia$

- 1. Similar to the construction of gdp_stunting_overweight shown in the 2024-04-09 lecture and the SQL script faostat-part_02-transforming_africa_fs.sql, use DBeaver and SQL commands to build a table gdp_stunting_overweight_anemia which has, for each country-year pair the following associated values for:
 - GDP per capita Purchasing Power Parity (22013): use column name **gdp_p_ppp**.
 - Percentage of children over 5 years of age who are stunted (21025): . childhood_stunting
 - Percentage of children over 5 years of age who are overweight (21043): childhood_overweight.
 - Prevalence of anemia among women of reproductive age: anemia
- 2. Add this table into your schema **food_sec**.



Figure 1: Almost correct example of the table $gdp_stunting_overweight_anemia$. Your table should have 3 characters for area_code_m49 column, and may have some decimal values for the last 4 columns.

D. Step 3: Build table energy_undernourished

- 1. Note that many records in africa_fs_ac have year and year_code values based on 3-year intervals rather than single years. We will use some of this data to gain more insight about countries. In particular, we will interpret a 3-year interval as applying to the year in the middle, e.g., we will interpret 2000-2002 as applying to the year 2001.
- 2. First, build a table **energy_undernourished** which has, for each *country-year_code* pair the associated values for:
 - Average dietary energy supply adequacy (21010): use column name dietary_energy.
 - Prevalence of undernourishment (210041): use column name undernourished.
 - Note: this table should have 1040 rows in it.
- 3. Now add a column *derived_year* to the table **energy_undernourished**, where for each tuple, the derived year value is computed by using the year in the middle of the first and third years in the *year_code* of the tuple.
- 4. The column you added probably has data type integer. Convert this to varchar(4).

	area_code_m49	ABC area ▼	RR year_code ▼	123 dietary_energy 🔻	123 undernourishec 🔻	RBC derived_year ▼
1	12	Algeria	2000	127	8	2001
2	12	Algeria	2001	129	7	2002
3	12	Algeria	2002	130	7	2003
4	12	Algeria	2003	130	7	2004
5	12	Algeria	2004	131	6	2005
6	12	Algeria	2005	132	6	2006
7	12	Algeria	2006	133	5	2007
8	12	Algeria	2007	135	5	2008
9	12	Algeria	2008	136	5	2009
10	12	Algeria	2009	139	4	2010

Figure 2: Almost correct example of the table *energy_undernourished*. As with Figure 1, the area_code_m49 column should have 3 characters, and the values for last 3 columns may have decimal values.

E. Step 4: Joining the $gdp_stunting_overweight_anemia$ and $energy_undernourished$ tables to create new table $gdp_energy_with_fs_indicators$

- 1. Create a selection query that combines the table **gdp_stunting_overweight_anemia** and **energy_undernourished** to form a new table **gdp_energy_with_fs_indicators**
 - The columns should include $area_code_m49$, area, $year_code$, gdp_pc_ppp , $dietary_energy$, $childhood_stunting$, $childhood_overweight$, anemia and undernourished.
 - Tuples in this table should be formed by combining tuples from **gdp_stunting_overweight_anemia** and **energy_undernourished** where *year_code* from the first table equals *derived_year* of the second table.
 - Note: your table should have 895 tuples in it.
- 2. Export the table gdp_energy_with_fs_indicators as a csv file gdp_energy_with_fs_indicators.csv.
- 3. Sort this csv file by area (country name) and then year_code.
- 4. CONGRATULATIONS: you have created a table that we can use later to determine whether there are statistical correlations between gdp per capita and/or stunting, childhood overweight, anemia in women and/or undernourishment.

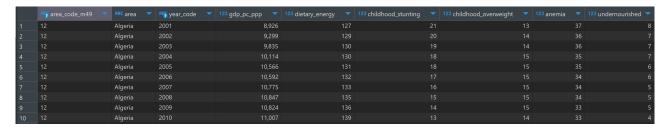


Figure 3: Almost correct example of the table $gdp_energy_with_fs_indicators$

- 5. Create a new table **gdp_energy_fs_aggs**.
 - Which has columns as:
 - area_code_m49
 - area
 - $avg_gdp_pc_ppp$
 - $avg_dietary_energy$

- $-\ avg_childhood_stunting$
- $avg_childhood_overweight$
- avg_anemia
- avg_undernourished
- The "avg" columns should hold the averages of the corresponding items for each country, over all of the years of available data.
- Use the round operator on the "avg" value, so that they have type numeric and are rounded to 2 decimal points. Use the following kind of expression: round(< expression for average >::numeric, 2).
- 6. Export the table gdp_energy_fs_aggs as a csv file gdp_energy_fs_aggs.csv.
- 7. The table should be sorted by area (i.e., country name). (You can use ORDER BY in the query or rort the csv file once you have created it.)

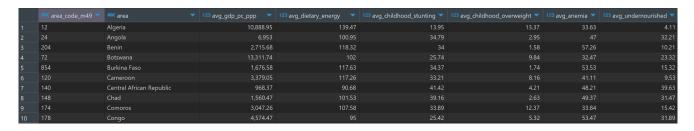


Figure 4: Almost correct example of the table $gdp_energy_fs_aggs$. You will obtain slightly different values. This table was computed with rounded values for various columns, rather than with values having decimals.

F. Submission

- 1. Please make a single zip file that includes
 - $gdp_energy_with_fs_indicators.csv$
 - $gdp_energy_fs_aggs.csv$
 - The DBeaver sql scripts that you used to create these 2 csv files, specifically, Step_2.sql, Step_3.sql, Step_4.sql.
 - Name the zip file as FirstName_LastName_LastFourDigitsOfStudentID_ECS116_A1.
- 2. Upload it on Canvas for Assignment 1 (This is a solo assignment so don't add your peers to your submission).