Wikipedia assistant

1. Create a database based on the Simple English Wikipedia content. There are periodic dumps with the data that you can find here. And here, you can find a list of the tables and their schema. The tables that are relevant for this assignment are: page, pagelinks and categorylinks. The database should contain the following:
   - Basic metadata for every wiki page:
     - Page title
     - Categories of the page
     - Date of last modification
   - The links between the wiki pages, defined by the following:
     - The page which refers to another page
     - The referred page

   You're allowed to make simplifications (in case something is not trivial to implement) but you have to document your decisions about these simplifications.
   You can download and preprocess the data dumps as you wish, but we recommend using the SQL data dumps instead of the XML ones, since they are easier to work with.

2. Create an API to query the contents of the database. The API has to expose two endpoints:
   a. An endpoint that receives an arbitrary SQL query and returns the result of executing the query on the database.
   b. An endpoint that receives a category and returns the *most outdated* page for that category.

      A page is called outdated if at least one of the pages it refers to was modified later than the page itself. The measure of this outdatedness is the biggest difference between the last modification of a referred page and the last modification of the page.

      This query can be a bit slow, so you should precompute the results for the top 10 categories with more pages. You can assume that this endpoint will only be called with one of the top 10 categories.

3. Automate step 1 and the computation of the *most outdated* page for the top 10 categories:
   a. The pipeline has to run every month (according to the wikimedia dump frequency)
   b. Downtime while updating the database is accepted

c. You can use whatever automation tool you see fit (Ex. Airflow, cron, etc.), as long as you explain your decision.

4. Almost as important as developing software, is to provide ways in which that software can be run with ease in different systems (both for sharing with other developers and for deployment). You can use whatever tools you think are better for that purpose (Ex. Docker, Makefiles, bash scripts, etc.), as long as you explain your decision.

5. You have to submit all of your code and a short documentation about your application and thought process.