Language: **Java**

Frameworks/technologies that **must** be used: **Spring**, **JPA**, **Postgresql**.

The applicant is permitted to use any 3rd-party libraries of their choice.

Task.

Imagine a grid which stores bit strings. Each string consists of *1*s and/or *0*s and cannot contain any other symbols. One string occupies exactly one grid cell. A cell can store **exactly one or zero** strings. All the strings are of **the same** length. Consider the grid shown below.

|  |  |  |
| --- | --- | --- |
| 10001100 |  | 11110000 |
|  |  | 11110010 |
| 10101100 |  |  |
| 10100100 |  | 11111111 |

This grid has four rows and three columns. Some of the cells contain strings. Each string contains exactly 8 symbols.

Now, one needs to answer the following question. For a given column what is **the longest common prefix** of the strings stored in that column? For the grid above the answer for the first column would be “10”. The answer for the second column would be the empty string. The answer for the third column would be “1111”.

Your task is to create an API that performs the following operations.

1. Create a grid with the given name and the given *cell size* (length of strings it stores). The name must not be blank and must consists of between *1* and *200* symbols. *cell size* must be a positive integer not greater than *100000*. The newly created grid must be of size 1x1, i.e. it must contain exactly one row and one column.
2. Update the name of the given grid.
3. Delete the given grid. All the data the grid stores must also be deleted.
4. List all the grids.
5. Insert a new column next to the given column. The column must be assigned the correct number. For example, if a column is inserted after the column number 5 then its number must be 6.
6. Delete the given column.
7. Set the given (nullable) value to the given cell. The value must be a valid bit string of length *cell size*.
8. For the given column get the longest common prefix.

It is excepted that operation 8 will be executed very often and an answer is expected to be given **as quickly as possible**. The other operations are not expected to be executed often and are not required to return immediately. A grid is expected to have no more than *100000* rows and no more than *10* columns. A grid is expected to store strings of length up to *100000.*

1. All the *ToDo*s in the source code need to be resolved.
2. You only need to create an API described above. No additional endpoints needed.
3. All the tests must pass.
4. You do *not* *have to* write your own tests however you are permitted to do so. Also, keep in mind that additional tests - not included in the source code - will be run against your application.
5. For the task jdk 15+ needs to be installed on the machine. Postgresql must be accessible either locally or remotely. docker-compose may be used to get postgresql up and running.

Good luck!