

Globant's Data Engineering Coding Challenge

Welcome to Globant's Data Engineering coding challenge!

You will find several different sections in here. Mind that:

- You can **choose which sections** to solve based on your experience and available time
- if you don't know how to solve a section, you can proceed with the following one
- You can **use whichever language, libraries, and frameworks** that you want.
- The usage of cloud services is allowed, you can choose whichever cloud provider that you want
- Try to always apply best practices and develop a scalable solution.
- We recommend you to solve everything
- If you don't have time to solve any sections, try to think the toolstack you would like to use and the resulting architecture, and why.
- Every complement you might want to add is highly welcome!
- In case you have a personal github repository to share with the interviewer, please do!

Section 1: API

In the context of a DB migration with 3 different tables (departments, jobs, employees) , create a local REST API that must:

1. Receive historical data from CSV files
2. Upload these files to the new DB
3. Be able to insert batch transactions (1 up to 1000 rows) with one request

You need to publish your code in GitHub. It will be taken into account if frequent updates are made to the repository that allow analyzing the development process. Ideally, create a markdown file for the Readme.md

Clarifications

- You decide the origin where the CSV files are located.
- You decide the destination database type, but it must be a SQL database.
- The CSV file is comma separated.

Section 2: SQL

You need to explore the data that was inserted in the previous section. The stakeholders ask for some specific metrics they need. You should create an end-point for each requirement.

Requirements

- Number of employees hired for each job and department in 2021 divided by quarter. The table must be ordered alphabetically by department and job.

Output example:

department		job	Q1	Q2	Q3	Q4
Staff		Recruiter	3	0	7	11
Staff		Manager	2	1	0	2
Supply Chain		Manager	0	1	3	0

- List of ids, name and number of employees hired of each department that hired more employees than the mean of employees hired in 2021 for all the departments, ordered by the number of employees hired (descending).

Output example:

id	department	hired
7	Staff	45
9	Supply Chain	12

Bonus Track! Cloud, Testing & Containers

Add the following to your solution to make it more robust:

- Host your architecture in any public cloud (using the services you consider more adequate)
- Add automated tests to the API
 - You can use whichever library that you want

- Different tests types, if necessary, are welcome
- Containerize your application
 - Create a Dockerfile to deploy the package

Csv files structures

hired_employees.csv:

id INTEGER Id of the employee
name STRING Name and surname of the employee
datetime STRING Hire datetime in ISO format
department_id INTEGER Id of the department which the employee was hired for
job_id INTEGER Id of the job which the employee was hired for

4535,Marcelo Gonzalez,2021-07-27T16:02:08Z,1,2

4572,Lidia Mendez,2021-07-27T19:04:09Z,1,2

[Link to the file](#)

departments.csv

id INTEGER Id of the department
department STRING Name of the department

1, Supply Chain

2, Maintenance

3, Staff

[Link to the file](#)

jobs.csv:

id INTEGER Id of the job
job STRING Name of the job

1, Recruiter

2, Manager

3, Analyst

[Link to the file](#)