Pagila Data Challenge

The Pagila Data Challenge is a coding test designed to evaluate a candidate's data engineering and SQL proficiency.

What is the Pagila Dataset?

The Pagila dataset is a sample database that models a DVD rental store, similar to real-world systems like Blockbuster or Netflix (during its DVD rental era). It is an enhanced version of the Sakila database and is commonly used for SQL and data engineering practice.

Prerequisites

To complete this challenge, you need **Docker Engine** installed on your machine.

If you don't have Docker, the easiest way to install it is via **Rancher Desktop** (an open-source tool).
📌 Installation instructions for Rancher Desktop

Setup PostgreSQL Database

The database will run inside a Docker container. Follow these steps:

1. Clone the repository and start the database using Docker:

```
git clone https://github.com/devrimgunduz/pagila.git
cd pagila
docker-compose up
```

2. Expected Result

Once the container starts, the **PostgreSQL database** will be **exposed on the following ports**:

- **PostgreSQL** → `5432`
- **pgAdmin** → `5050`

Database Credentials

Service	Username	Password
PostgreSQL	`postgres`	`123456`
pgAdmin4	`admin@admin.com`	`root`

After the container is created, you can access **pgAdmin** at:
👉 **http://localhost:5050/**

Log in with:

- **Email**: `admin@admin.com`
- **Password**: `root`

Challenge Task

You need to **set up a Spark project** and perform **data aggregations** using the PostgreSQL database running inside your Docker container.

Steps to Complete the Challenge

1. Setup Spark Project

- Create a virtual environment if necessary.
- Install the required dependencies (e.g., `pyspark`) using `requirements.txt`.

2. Connect to PostgreSQL

- Update `main.py` with the correct **JDBC URL** and connection properties.
- Ensure the PostgreSQL **JDBC driver** is properly linked.

3. Implement Data Aggregations

- The project structure is **preconfigured** (check `main.py`).
- Implement the **data loading and aggregation functions**.
- Use `.show()` to display the results.

4. Write Unit Tests

- Implement tests in `test_main.py`.
- Follow best practices such as **mocking** database connections where needed.

5. Submit Your Solution

- **Do NOT include the Pagila data** in your submission.
- Zip **only the project files** and send them to the **same email** from which you received the task.

Best Practices

✔ Follow **clean code principles** and **modularize your functions** properly. ✔ Make sure your code is **efficient and scalable**.
✔ Add proper **error handling** where necessary.

🚀 **Good luck!**