To create a fully functional replication process that runs automatically and updates data in real time, we need to ensure that all the components are properly configured and tested. Here are the key steps:

Set up the Oracle, PostgreSQL, and Cassandra databases with the necessary tables and data.

Install Rubyrep on the machine where the replication process will run. This can be done using the RubyGems package manager.

Configure Rubyrep with the necessary database connection details and table mappings. This will involve creating a YAML file that specifies the source and target databases, tables to be replicated, and any filters or transformations to be applied.

Test the replication process manually to ensure that it works as expected. This can be done by running Rubyrep from the command line and monitoring the logs for any errors or issues.

Schedule the replication process to run automatically at a specified interval. This can be done using a tool like Cron on Linux or Task Scheduler on Windows.

Monitor the replication process to ensure that it is running smoothly and addressing any issues that arise.

By following these steps and conducting thorough testing and monitoring, we can create a reliable and efficient replication process that updates data in real time and runs automatically.

Sure! Here are the detailed steps for setting up the fully functional replication process using Rubyrep:

Ensure that both source databases (PostgreSQL and Cassandra) are configured correctly and the data to be replicated is available.

Install Rubyrep by running the command

***gem install rubyrep.***

Create a new configuration file for Rubyrep by running the command ***rubyrep newconfig***.

Provide the necessary configuration details for both source databases and the target Oracle database in the newly created configuration file. This includes the database type, hostname, port, username, password, and database name. Also, specify the tables to be replicated and any necessary filters to be applied.

Verify that the configuration file is correctly set up by running the command ***rubyrep checkconfig.***

Start the replication process by running the command ***rubyrep replicate.***

Schedule the replication process to run automatically every day at midnight using a scheduler like cron or Windows Task Scheduler.

To ensure that the replication process runs smoothly and efficiently, it is important to regularly monitor the replication status and troubleshoot any errors that may arise. This includes checking for data consistency, monitoring system performance, and updating the configuration file as necessary.

Overall, with the above steps, you should have a fully functional replication process that runs automatically and updates data in real time.