GW2-SRS LOAD

powered by LaTeX

Daniel Lopez: **Load algorithm**

November 10, 2022

3.0 LOAD

3.1 Introduction

The last part concerning the ETL process is loading the results into a database. In this case I decided going for MongoDB and SQLite. The reason behind this decision was storing big JSON files in a suitable database like MongoDB, while maintaining an space for structured tables on SQLite that I built with queries.

Now the key part here, is that at first I was saving entire JSONs on MongoDB, and while this is the purpose of a No-SQL database, I preferred just uploading the dictionaries I created manually within the ETL code. As for SQLite, there are two main queries, one for dps data and one for users.

3.2 SQLite Queries

In order to have every boss classified, I created a table for each boss, so the data loading was a matter of inserting the data within the ETL, therefore I needed a connection with the database.

I used Python and sqlite3[1] to set up the connection and execute the queries. The main connection can be set up using the following:

```
        conn = sqlite3.connect("Your_database_path")
        cur = conn.cursor()

```

Listing 1: SQLite Connection

From this line, we can easily execute any query inside the database by calling the cursor:

```
        cur.execute(
            f"INSERT INTO vg_dps(phase1_dps,phase2_dps,
    phase3_dps,FK_player_id) \
            VALUES({dps1},{dps2},{dps3},'{acc}')"
        )

```

Listing 2: Query example

[1]sqlite3 is a Python library used to work with SQLite database

3.3 MongoDB Connection

As for MongoDB, the connection is quite simple as well. I chose to use PyMongo library and a MongoDB Atlas Cluster to help me out. I used the local cluster, but this process could also be made on the cloud cluster as well. The connection would look like the following:

```
        client = pymongo.MongoClient('mongodb://localhost
    :27017/')

```

Listing 3: MongoDB Connection

```
        db = client['GW2_SRS']
        collection = db['players_info']

        collection.insert_one(json_data)
        print('MongoDB load done!')

```

Listing 4: MongoDB data load