Python for Everybody

Final Project - ETL Data from API to Database

Hùng Phạm

July 26, 2023

Got It
Expert in seconds

Python for Everybody - Covered Contents

Course contents:

Apart from Course 5 - Capstone (mostly optional Honors Content), each course of the program covers the following aspects of Python & programming as a whole:

Course 1: Getting Started

- Hardware guide, Python installation, ‘hello world’
- Expressions
- Conditional code: if, else, elif
- Function syntax
- Loops & iterations: for, while, nested

Course 2: Data Structures

- Strings
- Files
- Lists
- Dictionaries
- Tuples

...and how to process data of each structure

Course 3: Accessing Web Data

- RegEx
- Network, socket, HTTP requests
- Unicode, ASCII, UTF-8
- Using socket & urllib to open and read URLs
- Parsing HTML with BeautifulSoup, XML with ElementTree, JSON with json

Course 4: Using Databases

- OOP: classes & objects
- SQL overview
- Designing data models, types of relationships: 1-to-1, 1-to-Many, Many-to-Many
- Visualization (introducing an geocoding app written in Python, JavaScript, and HTML)

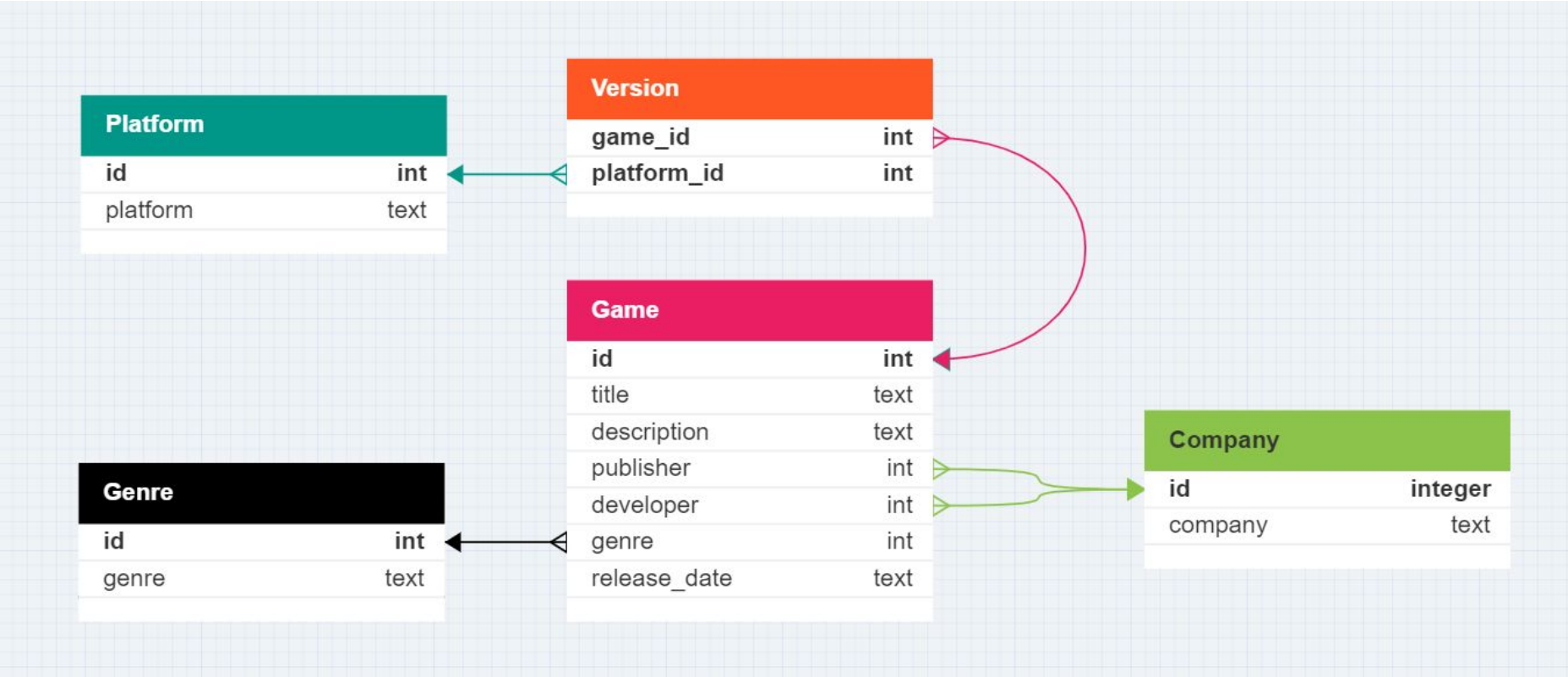
Python for Everybody - Final Project Overview

- **Features:**
  - Send GET requests to API endpoint, extract JSON data, parse data into a Python dictionary
  - Create a relational database, insert retrieved data into its corresponding tables
  - Display users' SQL query results on an HTML webpage
- **API chosen:** freetogame.com

Reasons: 100% free access to all endpoints, no authorization required, almost no rate limit (< 4 requests PER SECOND), small-size dataset (~400 games), a widely popular API for coding projects.

- **Technologies used:**
  - Python & its libraries (i.e. sqlite3, urllib, json)
  - SQL (i.e. SQLite and its GUI - DB Browser for SQLite)
  - JSON
  - HTML, CSS, Markdown (outside of Py4E's scope)

Python for Everybody - Database Design



Python for Everybody - How To Use

Follow these steps to go through the app's workflow:

1. Open *API_call.py*, edit parameters (if needed), run file to check if data retrieval & JSON parsing works.
2. Run *DB_fill.py* to insert retrieved data into newly created/existing database (up to you).
3. Run *display_results.py*, enter valid SQL query & HTML filename when asked -> result set is written to said HTML file.
4. Open HTML file and see the table of query results.

****NOTE****: Non-executable files:**

- *SQL_script.py:* stores the script that creates all tables and defines their relationships.
- *games.json*: pre-retrieved data (in case of API connection problems).
- *styles.css*: apply styles to generated HTML file.

Thank you for your time!

Q&A