

এখানে JSON এবং SQL নিয়ে আলোচনা করা মূল বিষয়গুলি বাংলায় তুলে ধরা হলো:

১. JSON (JavaScript Object Notation)

JSON একটি সর্বজনীন ডেটা ফরম্যাট (Universal Format)। এর অর্থ হলো, এটি প্রায় সমস্ত প্রোগ্রামিং ভাষা (যেমন জাভা, পাইথন ইত্যাদি) দ্বারা সহজেই বোঝা যায়।

- গুরুত্ব: API (Application Programming Interface)-এর মাধ্যমে যখন কোনো সার্ভার থেকে ডেটা চাওয়া হয়, তখন প্রাপ্ত ডেটা সাধারণত JSON ফরম্যাটেই থাকে।
- ডেটা লোড করা: JSON ফাইল লোড করার জন্য Pandas লাইব্রেরির `pd.read_json()` ফাংশনটি ব্যবহৃত হয়। এটি `read_csv` ফাংশনের মতোই সহজ।
- উদাহরণ ডেটা: একটি রেসিপি ডেটা সেট, যেখানে প্রতিটি রো-তে খাবারের কুইজিন (**Cuisine**) এবং সেটির উপাদানগুলির (**Ingredients**) একটি তালিকা থাকে। এই ফাংশনটি সহজেই JSON ডেটাকে একটি ডেটাফ্রেমে রূপান্তরিত করে।
- প্যারামিটার: `read_json()` ফাংশনেও `read_csv`-এর মতো কিছু গুরুত্বপূর্ণ অপশন রয়েছে, যেমন:
 - encoding**: ডেটা এনকোডিং সমস্যা সমাধানের জন্য।
 - chunksize**: যদি JSON ফাইলটি খুব বড় হয়, তবে মেমরির উপর চাপ কমাতে ডেটা ছোট ছোট অংশে লোড করার জন্য।
 - nrows**: কতগুলি রো লোড করতে চান তা সীমিত করার জন্য।
- URL** থেকে লোড: শুধু লোকাল ফাইল নয়, আপনি একটি **URL** ব্যবহার করেও সরাসরি ওয়েব থেকে JSON ফরম্যাটে থাকা ডেটা লোড করতে পারেন। উদাহরণস্বরূপ, একটি এক্সচেঞ্জ রেট ডেটা সেট এভাবে লোড করে দেখানো হয়েছে।

২. SQL (Structured Query Language)

SQL হলো ডেটাবেস থেকে সরাসরি ডেটা তুলে আনার জন্য ব্যবহৃত ভাষা। এটি একটি খুব জনপ্রিয় ডেটা ফরম্যাট।

SQL ডেটাবেস থেকে ডেটা Pandas ডেটাফ্রেমে আনার প্রক্রিয়াটি কয়েকটি ধাপে বিভক্ত:

A. ডেটাবেস সেটআপ

- সার্ভার ইনস্টল করা: প্রথমে আপনাকে একটি ডেটাবেস সার্ভার (যেমন MySQL) সেটআপ করতে হবে। **XAMPP** নামক একটি সফটওয়্যার ব্যবহার করা হয়েছে।
- সার্ভার চালু করা: XAMPP কন্ট্রোল প্যানেল থেকে **Apache** (ওয়েব সার্ভার) এবং **MySQL** (ডেটাবেস সার্ভার) চালু করতে হয়।
- ডেটাবেস তৈরি ও ইম্পোর্ট: এরপর `localhost/phpmyadmin` ইউআরএল-এ গিয়ে একটি নতুন ডেটাবেস (উদাহরণস্বরূপ 'World' নামে) তৈরি করতে হয় এবং সেই ডেটাবেসে SQL ফাইলটি ইম্পোর্ট করতে হয়।
- ডেটাবেস স্ট্রাকচার: ইম্পোর্ট করার পর ডেটাবেসে সিটি (City), কান্ট্রি (Country) এবং কান্ট্রি ল্যাঙ্গুয়েজ (Country Language) এর মতো টেবিল তৈরি হয়।

B. পাইথনের সাথে সংযোগ স্থাপন

1. সংযোগকারী লাইব্রেরি: ডেটাবেস (MySQL) এবং পাইথন সফটওয়্যারের মধ্যে সংযোগ স্থাপন করার জন্য **mysql.connector** নামের একটি লাইব্রেরি ইনস্টল করতে হয় (**pip install mysql.connector**)। এই লাইব্রেরিটি একটি সেতুর (Bridge) মতো কাজ করে।
2. কানেকশন তৈরি: লাইব্রেরিটি ইম্পোর্ট করার পর, **mysql.connector.connect()** ফাংশন ব্যবহার করে ডেটাবেসের সাথে কানেকশন অবজেক্ট তৈরি করা হয়। এর জন্য কিছু প্যারামিটার দরকার:
 - **host**: সার্ভারের ঠিকানা (যেমন নিজের মেশিনের জন্য **localhost**)।
 - **user**: ব্যবহারকারীর নাম (যেমন ডিফল্টভাবে **root**)।
 - **password**: পাসওয়ার্ড (সাধারণত ফাঁকা থাকে)।
 - **database**: যে ডেটাবেস থেকে ডেটা নিতে চান তার নাম (যেমন 'World')।

C. ডেটাক্রমে রূপান্তর

1. ফাংশন ব্যবহার: ডেটাবেস থেকে ডেটাক্রমে ডেটা আনার জন্য Pandas-এর **pd.read_sql_query()** ফাংশনটি ব্যবহার করা হয়।
2. আগ্রুমেণ্ট: এই ফাংশনটিতে দুটি জিনিস পাস করতে হয়:
 - **SQL** কোয়ারী: আপনি কোন ডেটা চান, তা SQL ভাষায় লিখতে হবে (যেমন: **SELECT * FROM City**)।
 - কানেকশন অবজেক্ট: আগের ধাপে তৈরি করা সংযোগকারী অবজেক্ট।
3. ডেটা ফিল্টারিং: SQL কোয়ারী ব্যবহার করে আপনি ডেটা নিজের প্রয়োজন অনুযায়ী পরিবর্তন এবং ফিল্টার করতে পারেন। উদাহরণস্বরূপ, **WHERE** ক্লজ ব্যবহার করে কেবল একটি নির্দিষ্ট দেশের (যেমন ভারত বা আমেরিকা) সিটিগুলির ডেটা আনা যায়।
4. প্যারামিটার: **read_sql_query()** তেও **index_col** সেট করা যায় এবং বড় ডেটা সেটের জন্য **chunksize** বিকল্পটি ব্যবহার করা যেতে পারে, যাতে ডেটা রামের উপর চাপ না দিয়ে লোড করা যায়।

৩. পরবর্তী ধাপ

একটি **API** থেকে ডেটা ফেচ করে ডেটা সেট তৈরি করা হবে।

source:

<https://www.apachefriends.org/index.html>

https://pandas.pydata.org/docs/reference/api/pandas.read_sql_query.html#pandas.read_sql_query

https://pandas.pydata.org/docs/reference/api/pandas.read_json.html

<https://www.kaggle.com/datasets/busiellmorley/worldcities-pop-lang-rank-sql-create-tbls>