

Writeup

on

How I translated the provided data to the final schema

- I have made a file **json_parser.php**, that translates provided JSON data to MySQL Schema.
- I've copied the content of JSON file provided by you and put it a file **db.json**.
- I made a **\$file** variable and passed **“./db.json”** as a value in it. **“./”** indicates that **db.json** file is in the same directory where the **json_parser.php** is.
- Next we have to read the JSON file and store its contents to a PHP variable. For that, PHP supports the function **file_get_contents()** which will read an entire file and returns it as a string. So I passed **\$file** variable in this function and stored it in **\$json** variable.
- The next step for us is to convert json to array. Which is likely we have to convert the JSON string we got from the above step to PHP associative array. We will use PHP **json_decode()** function which decodes JSON string into PHP array.

```
$content = json_decode($json,true);
```

The first parameter **\$json** contains the JSON file contents.

The second parameter **true** will convert the string into php associative array.

- I've created **movies** table that contains all the fields specified in JSON file except **directors** field and **genre** field. Those two fields are in separate tables. The reason is explained in the next point.
- There are many fields which are **multi-valued**. But in a JSON format, it takes as one string. We will need each and every values of directors and genre for this project. So we have to break those strings and will have to make it multi-valued. For that, we have to **normalize** the tables. So I've made **separate tables for directors and genre**. **directors** table contains fields like **director_id** (which is AUTO_INCREMENT) and **director_name**. Same as directors table, I've made **genre** table containing fields like **genre_id** (which is AUTO_INCREMENT) and **genre**. To connect this tables with movies table, we have to create bridge tables. We have a primary key named **imdb_id** in movies table. We can connect directors and genre with movies table using this **imdb_id** field. To connect directors table with movies table, I've created **movies_directors bridge table** which contains **imdb_id** field as a foreign key of movies table and another field is **director_id** as a foreign key of directors table. Same thing I've done to connect genre table with movies table. I've created **movies_genre bridge table** which contains **imdb_id** field as a foreign key of movies table and another field is **genre_id** as a foreign key of genre table.
- To insert the records, I've used **mysqli** extension. For this. in **bind_param**, **“s”** stands for string and **“i”** for integer.
- Now we have to extract the array values. For that I've checked all the contents of a file using for loop. If there is comma (,) in the string of director or genre, it will be multi-valued. So I exploded such type of strings using PHP **explode** function. Name of directors and genres are

separated by comma(,) that is why I've passed “,” as a first parameter of explode function. The second parameter is **\$movie->director** and **\$movie->genre**. Where \$movie contains content of each and every record, and director or genre are name of that particular record.

- ➔ Now value for director and genre are exploded. So one movie can have more than one director or genre. Now we'll insert all the directors and genre in their respective tables. To prevent the table from duplicate names, we'll check if director or genre is already there in table or not. If it isn't there insert it using insert query. I've done it using for loops, **SELECT** query an **INSERT** query. I had to use trim function to prevent duplicate records. If there are more than 1 value it will be separated by comma and after each comma there will a space before next value. That's why “Drama” and “(space)Drama” will be different records. To prevent this I've used **trim** function.
- ➔ Then I inserted records in movies table using **INSERT INTO** query.
- ➔ Likewise we'll have to insert records into bridge tables. It will insert imdb_id of each records and it's director_id or genre_id. There can be more than one director_id or genre_id for one imdb_id. That's why I have used **foreach** loop to insert each and every record.
- ➔ This way I've translated JSON data into MySQL database and inserted records in the tables.