Objective: To transform a set of 8 relational database tables into a single, wide, and feature-rich CSV file suitable for advanced machine learning tasks.

Executive Summary: The project began with a normalized dataset spread across 8 separate CSV files, representing users, restaurants, visit histories, and medical data. This format is efficient for data storage but unsuitable for training machine learning models which require a single, "flattened" table. The executed Python script successfully implemented a multi-stage data engineering pipeline to address this. It intelligently selected the most relevant user-restaurant pairs, aggregated true historical transaction data, and enriched it by simulating a novel "craving" feature. The final output is a 200,000-row file, `flattened_hybrid_craving_final.csv`, where each row represents a comprehensive profile of a user's interaction with a specific restaurant, making it an ideal input for our subsequent modeling efforts.

Methodology: A Step-by-Step Breakdown

The flattening process was handled in three major logical stages:

Stage 1: Pre-computation and Data Preparation

Before the main flattening operation could begin, the script first prepared all the necessary data components. This stage was about creating the foundational building blocks.

- **1a. Feature Standardization (Identifying Top Dishes):**
  - **What:** The script analyzed the `VisitHistory.csv` and `VisitMenuItem.csv` files to determine the 15 most frequently ordered dishes across the entire dataset.
  - **Why:** This was a critical first step. To create a "wide" table, we need a consistent number of columns. By identifying a "master list" of the top 15 dishes, we ensured that every single user-restaurant row in our final file would have the same set of 180 transaction columns (`12 months x 15 dishes`), making the data uniform and model-ready.
- **1b. Data Transformation (Pivoting Medical Data):**
  - **What:** The `UserMedicalCondition.csv` file was in a "long" format (multiple rows per user). The script used the `pivot_table` function to transform this into a "wide" table.
  - **Why:** This conversion created a single row for each `user_id`, with each medical condition as its own column containing a `1` (if the user has the condition) or `0` (if they do not). This format is essential for using the conditions as features in a machine learning model.
- **1c. Intelligent Entity Pairing (Restaurant Selection):**
  - **What:** Instead of creating an inefficient row for every possible user-restaurant combination, the script intelligently selected the two most relevant restaurants for each of the 100,000 users.
  - **Why:** This logic ensures the final dataset is focused on meaningful interactions. It prioritizes restaurants a user has actually visited, and for new users, it smartly selects from their local city. This created the 200,000-row `base_df` which served as the structural backbone of our final file.

Stage 2: The Core Flattening Operation

This stage is where the primary transformation from a "long" to a "wide" data format occurred.

- **2a. Aggregating Historical Transactions:**
  - **What:** The script first joined the `VisitHistory.csv`, `VisitMenuItem.csv`, and `RestaurantMenu.csv` tables. It then performed a `groupby` operation to count the *actual number of times* each user ordered each specific dish at each restaurant, for each month.
  - **Why:** This step aggregated the raw event data into a summarized format, creating the ground truth for our historical features.
- **2b. Pivoting the Transaction Data:**
  - **What:** The script used a `pivot_table` on the aggregated transaction counts. This is the central "flattening" action.
  - **Why:** This operation took the `month` and `dish_name` from the rows and converted them into columns. The result was a new table where each row was a unique `(user_id, restaurant_id)` pair, and the columns were named `1_Pav Bhaji`, `2_Sushi Rolls`, etc. The value in each cell was the true historical count of orders. This single operation created the 180 `month_X_menu_item_Y` features that form the core of our behavioral data.

Stage 3: Final Assembly and Feature Enrichment

In the final stage, the script combined all the prepared pieces and generated the new, simulated `craving` feature.

- **3a. Assembling the Final DataFrame:**
  - **What:** The script performed a series of `merge` operations to combine the `base_df` (user-restaurant pairs), the `historical_data_pivot` (flattened transaction counts), the pivoted medical data, and the static user/restaurant information (age, location, etc.).
  - **Why:** This brought all our prepared data into a single, cohesive, and wide DataFrame.
- **3b. Generating the Hybrid "Craving" Feature:**
  - **What:** The script iterated through the assembled DataFrame in memory-safe chunks. For each of the 180 real transaction columns, it generated a new, corresponding `craving` column.
  - **Why:** This enriched our dataset with a powerful, predictive feature. The generation logic was hybrid: 75% of the time, the craving value was set to be identical to the real transaction (ensuring high correlation), while the other 25% of the time, it was a new, simulated value based on seasonal probabilities. This created a realistic feature that simulates a user's interest, which is related to but not identical to their actual behavior.

File Output

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
user_id	user_lat	user_long	age	restaurant_id	rest_name	rest_lat	rest_long	month_1_	month_1_	month_1_	month_1_	month_1_	month_1_	month_1_	month_1_	month_1_
1	12.998	77.5867	27	62	The Mint Mansion Kitchen	12.9807	77.6049	0	0	0	0	0	0	0	0	0
1	12.998	77.5867	27	284	The Rasgulla Restaurant Kitchen	12.9232	77.5546	0	0	0	0	0	0	0	0	0
2	17.3989	78.4527	42	227	Kolkata Kitchen Grill	17.4217	78.4499	0	0	0	0	0	0	0	0	0
2	17.3989	78.4527	42	308	The Biryani Bowl Cafe	17.3676	78.5348	0	0	0	0	0	0	0	0	0
3	12.2951	76.6738	28	72	The Papadum Paradise House	11.301	75.7474	0	0	0	0	0	0	0	0	0
3	12.2951	76.6738	28	221	The Roti Room Kitchen	12.3283	76.6191	0	0	0	0	0	0	0	0	0
4	13.3895	74.7516	36	625	The Curry Pot Cafe	9.89313	76.2278	0	0	0	0	0	0	0	0	0
4	13.3895	74.7516	36	238	The Royal Thali House	13.3641	74.6947	0	0	0	0	0	0	0	0	0
5	12.8905	74.8726	62	687	Butter Chicken Bungalow Grill	12.9324	74.8534	0	0	0	0	0	0	0	0	0
5	12.8905	74.8726	62	509	The Sesame Street Kitchen House	12.8741	74.8709	0	0	0	0	0	0	0	0	0
6	12.9975	77.6075	38	153	The Kulfi Corner Kitchen	12.2898	76.6568	0	0	0	0	0	0	0	0	0
6	12.9975	77.6075	38	399	The Barfi Bazaar Cafe	12.9677	77.5642	0	0	0	0	0	0	0	0	0
7	21.1814	72.8434	73	435	The Tandoor Oven Bistro	21.1335	72.7852	0	0	0	0	0	0	0	0	0
7	21.1814	72.8434	73	866	The Peacock Pavilion Cafe	21.182	72.8114	0	0	0	0	0	0	0	0	0
8	31.6372	74.875	33	318	The Rasgulla Restaurant Grill	19.1149	72.8454	0	0	0	0	0	0	0	0	0
8	31.6372	74.875	33	821	Tikka Trails Bistro	31.6644	74.8661	0	0	0	0	0	0	0	0	0
9	22.6215	88.3745	46	182	The Bay Leaf Bistro Cafe	25.2715	82.9558	0	0	0	0	0	0	0	0	0
9	22.6215	88.3745	46	24	The Ginger Garden Grill	22.5758	88.3687	0	0	0	0	0	0	0	0	0

Conclusion

The flattening script successfully served as a robust data engineering pipeline. It transformed 8 separate, normalized tables into a single, de-normalized, and feature-rich file. By aggregating true historical data and intelligently simulating a new predictive feature, it created the ideal foundation (`flattened_hybrid_craving_final.csv`) for the subsequent deep learning and clustering tasks.