Includes and Structures:

The code includes standard libraries and defines structures for tables, columns, functional dependencies, and multi-valued dependencies.

Column represents a column in a table.

Table contains information about the table's name, columns, primary key, and candidate keys.

FunctionalDependency and MultiValuedDependency represent dependencies for normalization.

Helper Functions:

generateSQLQuery generates a SQL CREATE TABLE query string from a Table structure.

printSQLQueries prints SQL queries for a list of tables.

parseTableFromCSV reads a CSV file and parses it into a Table structure. It assumes the first column as the primary key.

Normalization Functions: Placeholder functions for normalization (normalizeTo1NF, normalizeTo2NF, normalizeTo3NF, normalizeToBCNF, normalizeTo4NF, normalizeTo5NF) that currently return dummy data or the original tables.

Main Function: Reads a CSV file into a Table structure.

Calls normalization functions in sequence.

Prints SQL queries for each normalization step.

Expected Output

Given that the normalization functions are placeholders, the actual output will depend on the specific implementations of these functions.

normalizeTo1NF returns the original table.

normalizeTo2NF, normalizeTo3NF, normalizeToBCNF, normalizeTo4NF, and normalizeTo5NF return predefined tables or the original tables.

Execution Flow:

CSV Parsing: Parses "dataset.csv" (if the file exists and is correctly formatted).

Creates a Table object with the parsed data.

Normalization:

The table is normalized step by step:

1NF: Returns the original table.

2NF: Returns the original table and a predefined "CustomerOrders" table.

3NF: Returns a predefined set of tables ("Orders", "Customers", "CustomerOrders").

BCNF: Returns the predefined set of tables from 3NF (no change).

4NF: Returns the predefined set of tables from BCNF (no change).

5NF: Returns the predefined set of tables from 4NF (no change).

SQL Query Generation:

Generates and prints SQL CREATE TABLE statements for each normalization step.  
  
sql query output  
1NF (First Normal Form):

CREATE TABLE ParsedTable (Column1 VARCHAR(255), Column2 VARCHAR(255), PRIMARY KEY (Column1));

2NF (Second Normal Form):

CREATE TABLE ParsedTable (Column1 VARCHAR(255), Column2 VARCHAR(255), PRIMARY KEY (Column1));

CREATE TABLE CustomerOrders (CustomerID INT, OrderID INT, PRIMARY KEY (CustomerID, OrderID));

3NF (Third Normal Form):

CREATE TABLE Orders (OrderID INT, OrderDate DATE, PRIMARY KEY (OrderID));

CREATE TABLE Customers (CustomerID INT, PRIMARY KEY (CustomerID));

CREATE TABLE CustomerOrders (CustomerID INT, OrderID INT, PRIMARY KEY (CustomerID, OrderID));

BCNF (Boyce-Codd Normal Form):

CREATE TABLE Orders (OrderID INT, OrderDate DATE, PRIMARY KEY (OrderID));

CREATE TABLE Customers (CustomerID INT, PRIMARY KEY (CustomerID));

CREATE TABLE CustomerOrders (CustomerID INT, OrderID INT, PRIMARY KEY (CustomerID, OrderID));

4NF (Fourth Normal Form):

CREATE TABLE Orders (OrderID INT, OrderDate DATE, PRIMARY KEY (OrderID));

CREATE TABLE Customers (CustomerID INT, PRIMARY KEY (CustomerID));

CREATE TABLE CustomerOrders (CustomerID INT, OrderID INT, PRIMARY KEY (CustomerID, OrderID));

5NF (Fifth Normal Form):

CREATE TABLE Orders (OrderID INT, OrderDate DATE, PRIMARY KEY (OrderID));

CREATE TABLE Customers (CustomerID INT, PRIMARY KEY (CustomerID));

CREATE TABLE CustomerOrders (CustomerID INT, OrderID INT, PRIMARY KEY (CustomerID, OrderID));

The SQL queries' placeholder column names and types are based on the CSV data.

In general, the output for each normalization form will be SQL CREATE TABLE commands, which will be based on the table that has been parsed from the CSV and the given normalizing procedures.