# multiagent\_system.py

import sqlite3

from datetime import datetime

# -------- SQLite Setup & Utilities --------

def get\_connection():

return sqlite3.connect("multiagent.db")

def execute\_query(query, params=()):

conn = get\_connection()

cursor = conn.cursor()

cursor.execute(query, params)

conn.commit()

conn.close()

def fetch\_query(query, params=()):

conn = get\_connection()

cursor = conn.cursor()

cursor.execute(query, params)

result = cursor.fetchall()

conn.close()

return result

def initialize\_db():

schema = [

"""CREATE TABLE IF NOT EXISTS Users (

user\_id INTEGER PRIMARY KEY,

name TEXT,

age INTEGER,

gender TEXT,

location TEXT

);""",

"""CREATE TABLE IF NOT EXISTS Products (

product\_id INTEGER PRIMARY KEY,

name TEXT,

category TEXT,

price REAL,

popularity\_score REAL

);""",

"""CREATE TABLE IF NOT EXISTS UserActivity (

activity\_id INTEGER PRIMARY KEY AUTOINCREMENT,

user\_id INTEGER,

product\_id INTEGER,

activity\_type TEXT,

timestamp DATETIME DEFAULT CURRENT\_TIMESTAMP

);""",

"""CREATE TABLE IF NOT EXISTS Recommendations (

rec\_id INTEGER PRIMARY KEY AUTOINCREMENT,

user\_id INTEGER,

product\_id INTEGER,

reason TEXT,

timestamp DATETIME DEFAULT CURRENT\_TIMESTAMP

);"""

]

for query in schema:

execute\_query(query)

# -------- Customer Agent --------

def log\_activity(user\_id, product\_id, activity\_type):

query = """

INSERT INTO UserActivity (user\_id, product\_id, activity\_type)

VALUES (?, ?, ?)

"""

execute\_query(query, (user\_id, product\_id, activity\_type))

# -------- Product Agent --------

def get\_product\_info(product\_id):

query = "SELECT \* FROM Products WHERE product\_id = ?"

result = fetch\_query(query, (product\_id,))

return result[0] if result else None

# -------- Recommender Agent --------

def recommend\_for\_user(user\_id):

query = """

SELECT product\_id, COUNT(\*) as freq

FROM UserActivity

WHERE user\_id != ?

GROUP BY product\_id

ORDER BY freq DESC

LIMIT 5

"""

top\_products = fetch\_query(query, (user\_id,))

for product in top\_products:

insert\_recommendation(user\_id, product[0], "collaborative")

def insert\_recommendation(user\_id, product\_id, reason):

query = """

INSERT INTO Recommendations (user\_id, product\_id, reason)

VALUES (?, ?, ?)

"""

execute\_query(query, (user\_id, product\_id, reason))

# -------- Data Coordinator Agent --------

def process\_user\_interaction(user\_id, product\_id, activity\_type):

log\_activity(user\_id, product\_id, activity\_type)

recommend\_for\_user(user\_id)

# -------- Helper: Insert Sample Data --------

def insert\_sample\_data():

users = [

(1, "Alice", 28, "Female", "NY"),

(2, "Bob", 34, "Male", "CA"),

(3, "Charlie", 25, "Male", "TX")

]

products = [

(101, "Wireless Mouse", "Electronics", 25.99, 8.5),

(102, "Gaming Keyboard", "Electronics", 49.99, 9.2),

(103, "Running Shoes", "Sportswear", 79.99, 7.8),

(104, "Smart Watch", "Electronics", 99.99, 9.5)

]

for u in users:

execute\_query("INSERT OR IGNORE INTO Users VALUES (?, ?, ?, ?, ?)", u)

for p in products:

execute\_query("INSERT OR IGNORE INTO Products VALUES (?, ?, ?, ?, ?)", p)

# -------- View Recommendations --------

def get\_user\_recommendations(user\_id):

results = fetch\_query("""

SELECT Products.name, Recommendations.reason

FROM Recommendations

JOIN Products ON Recommendations.product\_id = Products.product\_id

WHERE Recommendations.user\_id = ?

""", (user\_id,))

print(f"\nRecommendations for User {user\_id}:")

for name, reason in results:

print(f"- {name} (Reason: {reason})")

# -------- Main Simulation --------

def simulate():

initialize\_db()

insert\_sample\_data()

interactions = [

(1, 101, "viewed"),

(2, 101, "purchased"),

(3, 102, "viewed"),

(1, 103, "added\_to\_cart"),

(2, 104, "purchased")

]

for user\_id, product\_id, activity\_type in interactions:

process\_user\_interaction(user\_id, product\_id, activity\_type)

get\_user\_recommendations(1)

if \_\_name\_\_ == "\_\_main\_\_":

simulate()