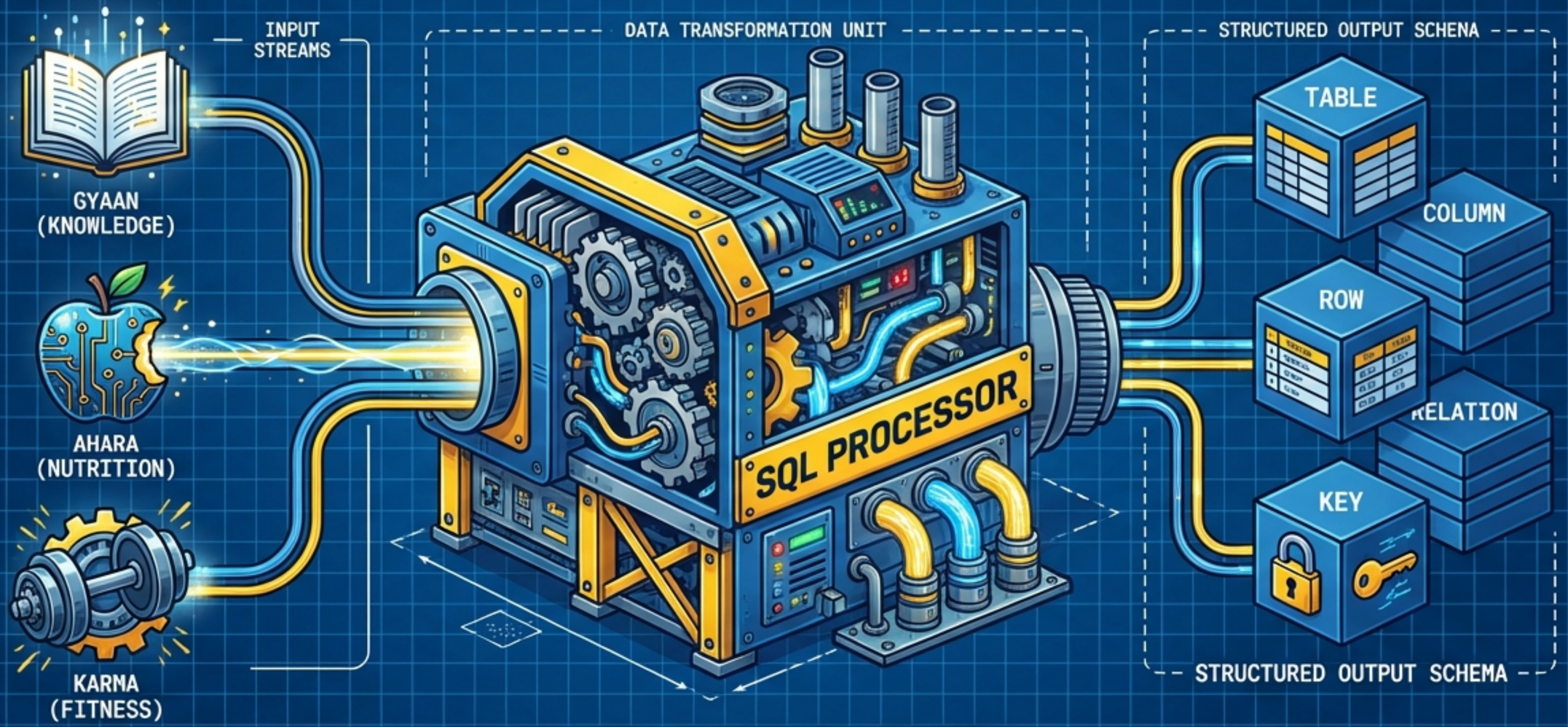
FROM CONCEPT TO CODE

THE GAK DATABASE PLAYBOOK: A STRATEGY GUIDE



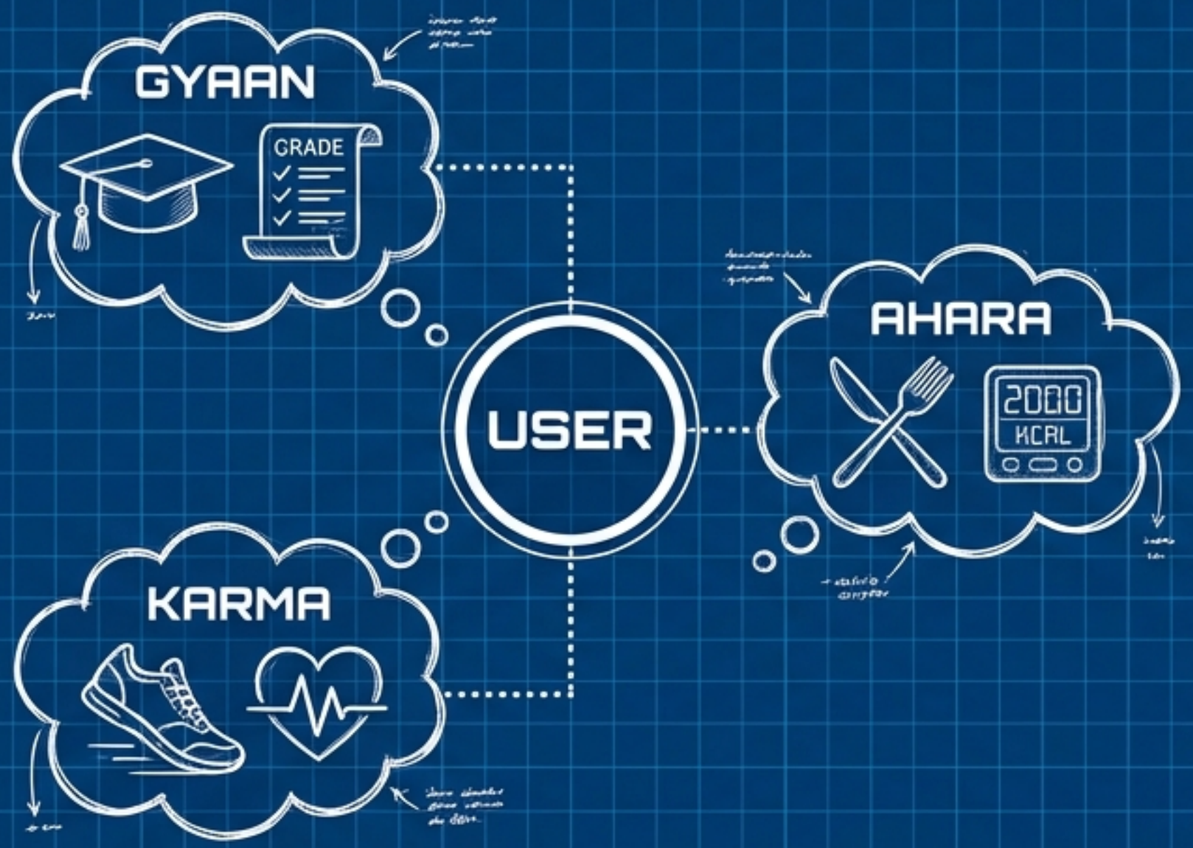
PLAYER 1: READY. SYSTEM STATUS: NORMALIZING...

OPTIMIZING...

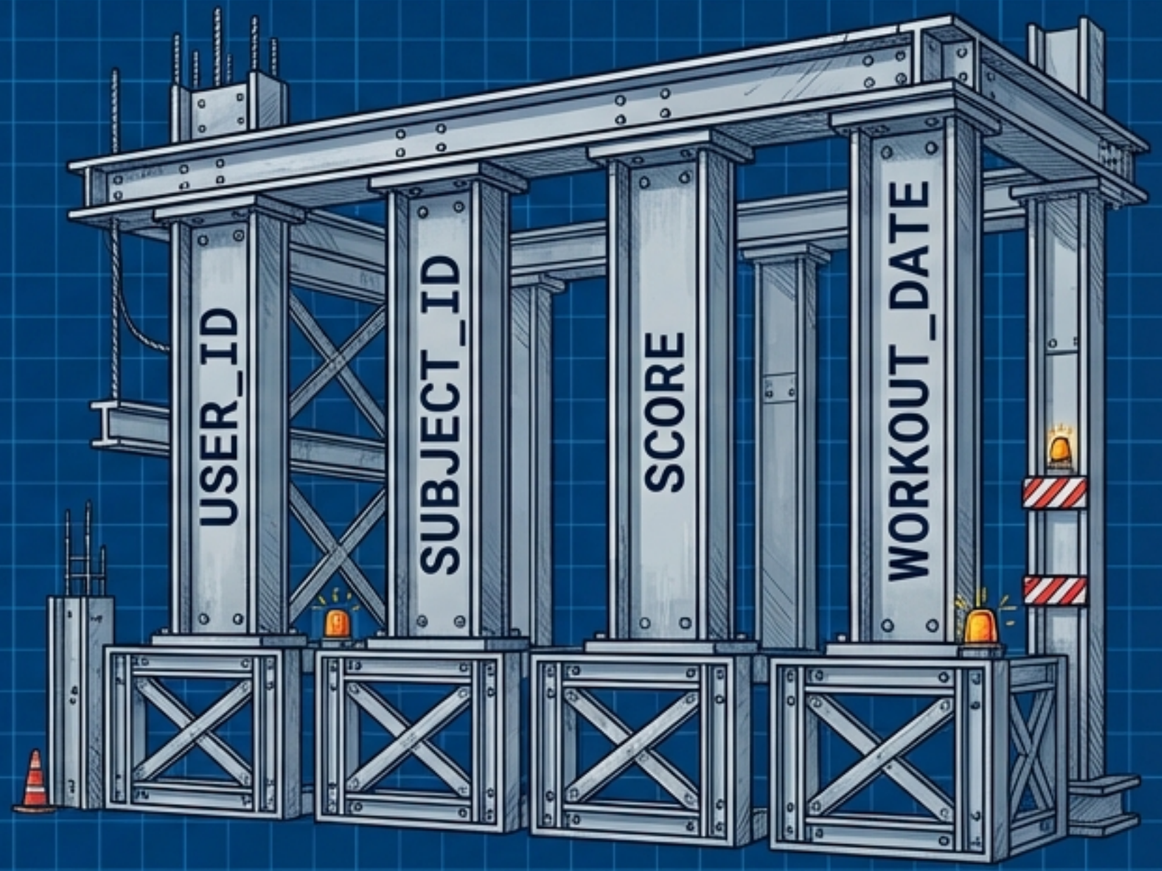
NotebookLM

THE MISSION

THE BLUEPRINT



THE REALITY



Computers don't speak 'Wellness' or 'Academics'. They speak ROWS and COLUMNS.
GOAL: Translate the GAK Blueprint into a functional SQL Reality.

NotebookLM

LEVEL 1: THE CORE UNITS

USER

- user_id (PK)
- full_name
- email

SUBJECT

- subject_id (PK)
- subject_name
- credits

WORKOUT_PLAN

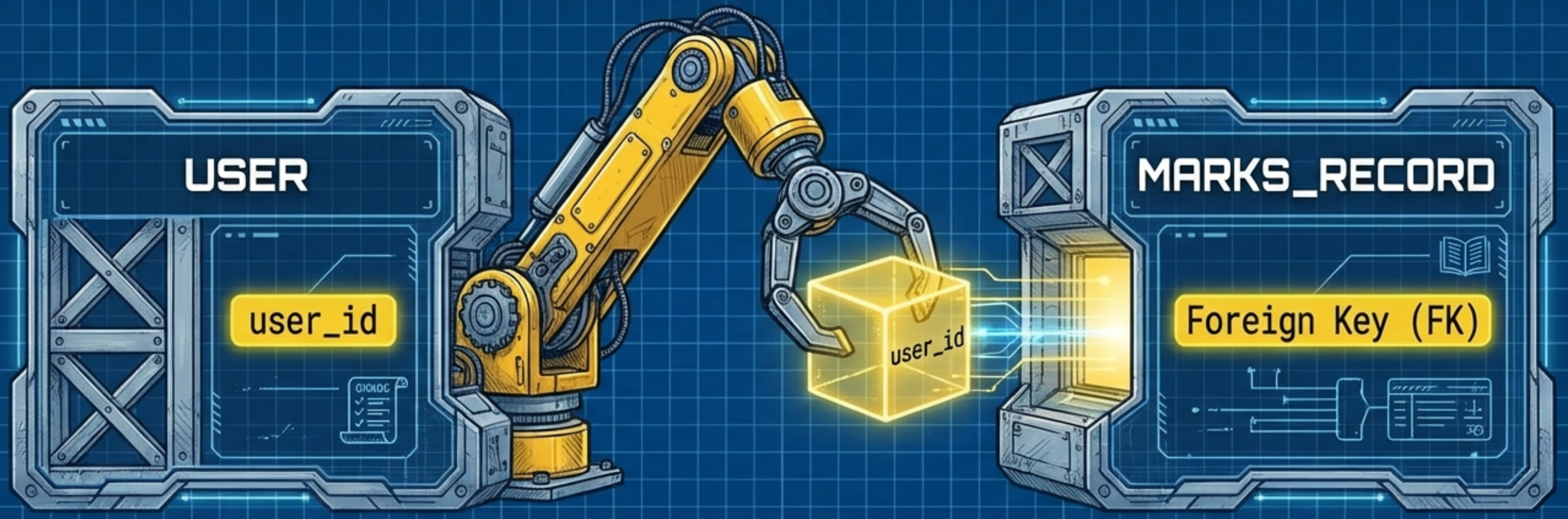
- plan_id (PK)
- user_id
- created_at

THE RULE: 1 ENTITY = 1 TABLE

Every core concept in GAK gets its own base table.
Identify the Primary Key (PK). This is the Unit's unique Tag.

NotebookLM

LEVEL 2: THE LINK (One-to-Many)

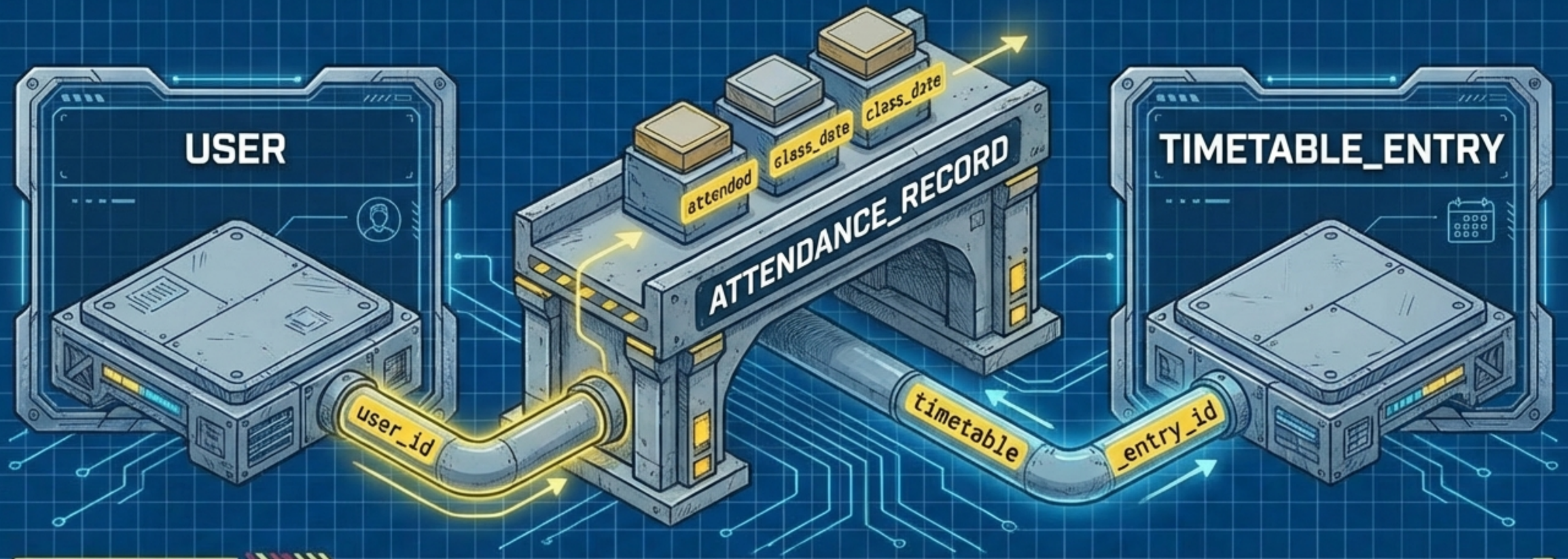


ONE-TO-MANY MECHANIC:

The 'Many' side (MARKS_RECORD) needs a reference to the 'One' side (USER).
Action: Add the 'One' side's Key (user_id) as a new column. This is the FOREIGN KEY.

NotebookLM

LEVEL 3: THE BRIDGE (Many-to-Many)



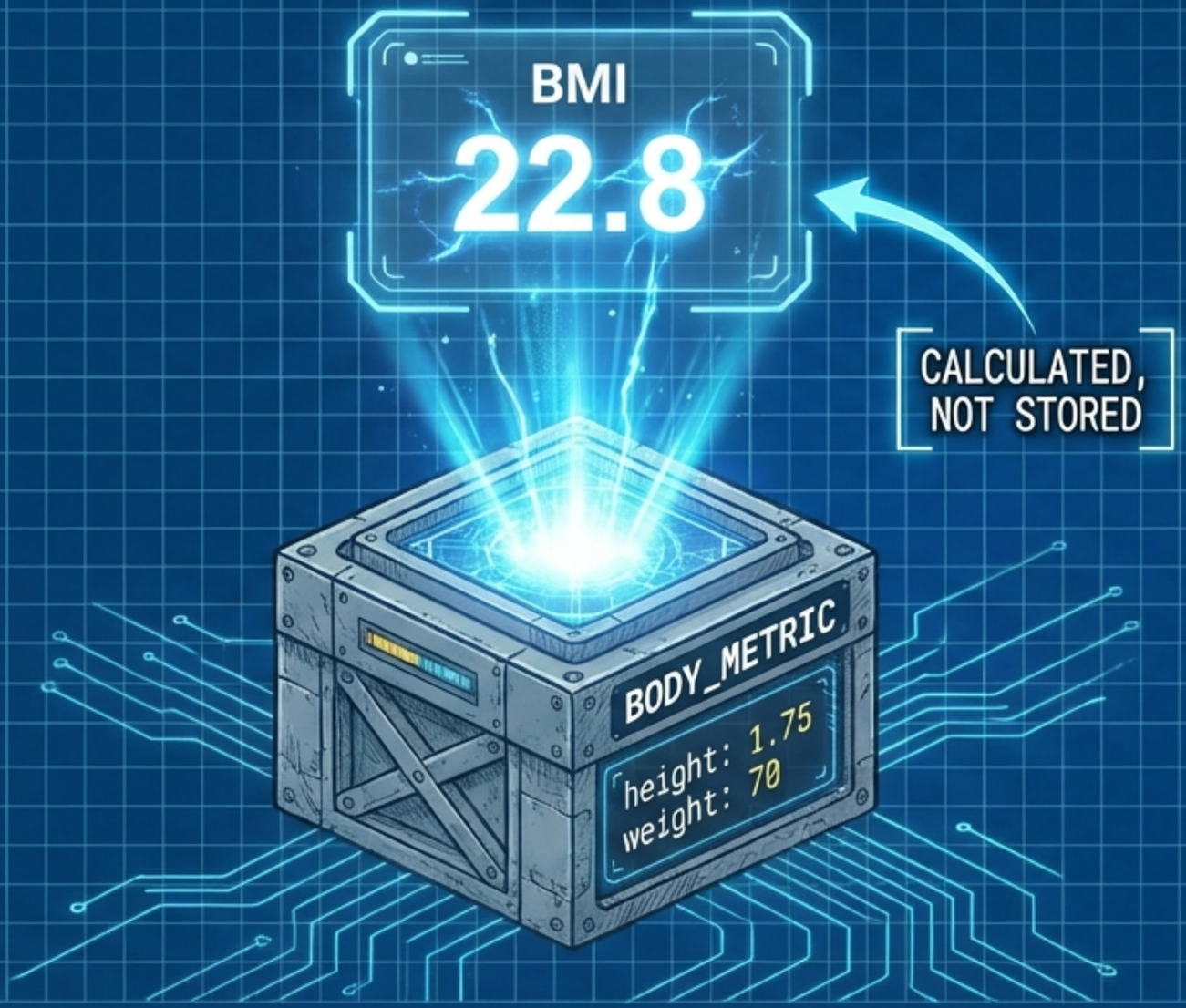
MANY-TO-MANY SOLUTION:

Problem: A student attends many classes; a class has many students. You can't fit a list in a cell!
Solution: Build a Bridge Table (ATTENDANCE_RECORD).
Ingredients: FK from Table A + FK from Table B + Extra Data.

NotebookLM

POWER-UP: THE INTELLIGENCE LAYER

Derived Attributes



INTELLIGENCE MECHANIC:

RULE: Don't store what you can calculate!

Example 1: BMI is computed from weight and height on the fly.

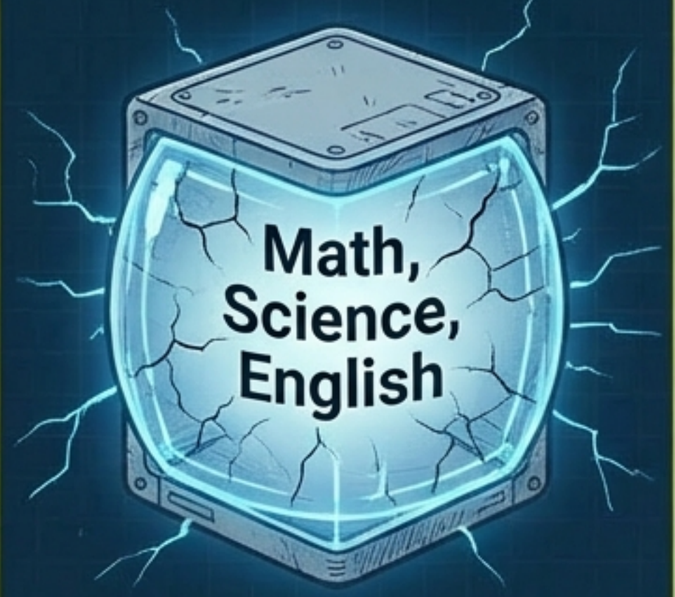
Example 2: attendance_percentage is counted from ATTENDANCE_RECORD rows.

Benefit: Keeps the database fast, smart, and free of redundant data.

NotebookLM

AVOID GAME OVER ERRORS

THE STUFFED CELL

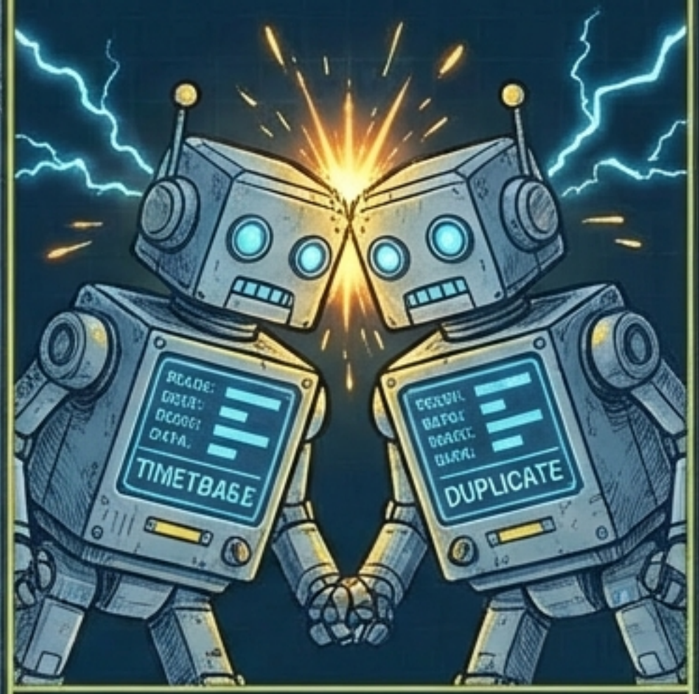


Don't put lists in one column! (1NF Violation)

THE ORPHAN RECORD

Foreign Keys must point to existing Primary Keys.

THE DUPLICATE

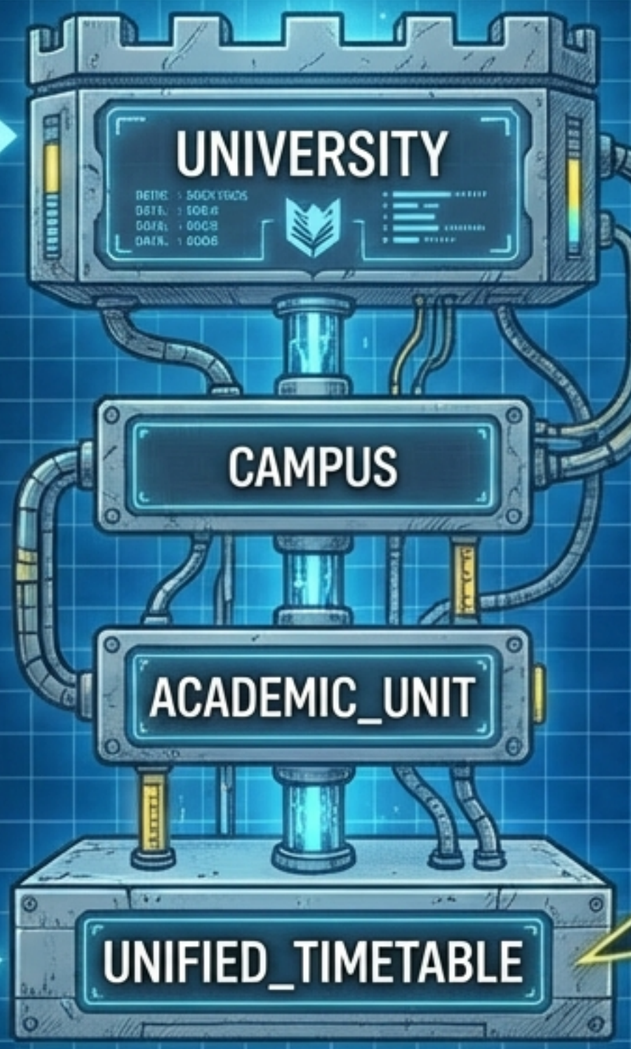


Don't repeat student names in the Marks table; use ID references.

NotebookLM

ZONE MAP A: THE ACADEMIC FORTRESS

INSTITUTIONAL BACKBONE



Shared Asset.
Not duplicated
per student.

Hierarchy:
University -> Campus -> Unit -> Timetable.

Key Insight:
The UNIFIED_TIMETABLE is the master schedule. Students 'subscribe' to it via their ACADEMIC_PROFILE.

NotebookLM

ZONE MAP B: THE LIFESTYLE CIRCUITS

KARMA (Fitness)

WORKOUT_PLAN → WORKOUT_SESSION → WORKOUT_ACTION

Status: Completed/Skipped

AHARA (Nutrition)

FOOD_IMAGE → DETECTED_FOOD_ITEM → CONFIRMED_FOOD_ITEM → FOOD_LOG

Standard Pattern: Plan/Input -> Event -> Logged Result.

NotebookLM

SPECIAL MECHANIC: THE VERIFICATION LOOP

DETECTED_FOOD_ITEM

food_name: Rice
confidence_score: 0.95

Human Review

CONFIRMED_FOOD_ITEM

calories: 200
protein: 4g
fats: 1g

AI Guesses, Player Confirms.
The database stores the AI's *confidence* separate from the User's *confirmation*. This ensures the FOOD_LOG is accurate even if image recognition fails.

NotebookLM

CONSOLE COMMAND: DEFINE STRUCTURE (DDL)

```
CREATE TABLE timetable_entry (
  timetable_entry_id varchar(255) PRIMARY KEY,
  unified_timetable_id varchar(255),
  start_time time,
  end_time time,
  day_order int
);
```

Text Strings (IDs)

Pure time (09:00:00)

Unique Handle

NotebookLM

CONSOLE COMMAND: ENFORCE GRAVITY (Constraints)

```
ALTER TABLE timetable_entry
ADD FOREIGN KEY (faculty_id) REFERENCES
faculty (faculty_id);
```

Referential Integrity:
Prevents "Floating" Data.
You cannot schedule a class in a room that doesn't exist.

NotebookLM

PRO STRAT: THE CALCULATED HUD

ATTENDANCE STATUS

(attended / total) * 100

CALORIE BURN RATE

10 cal/min

calories_burned / duration

GOAL PROGRESS

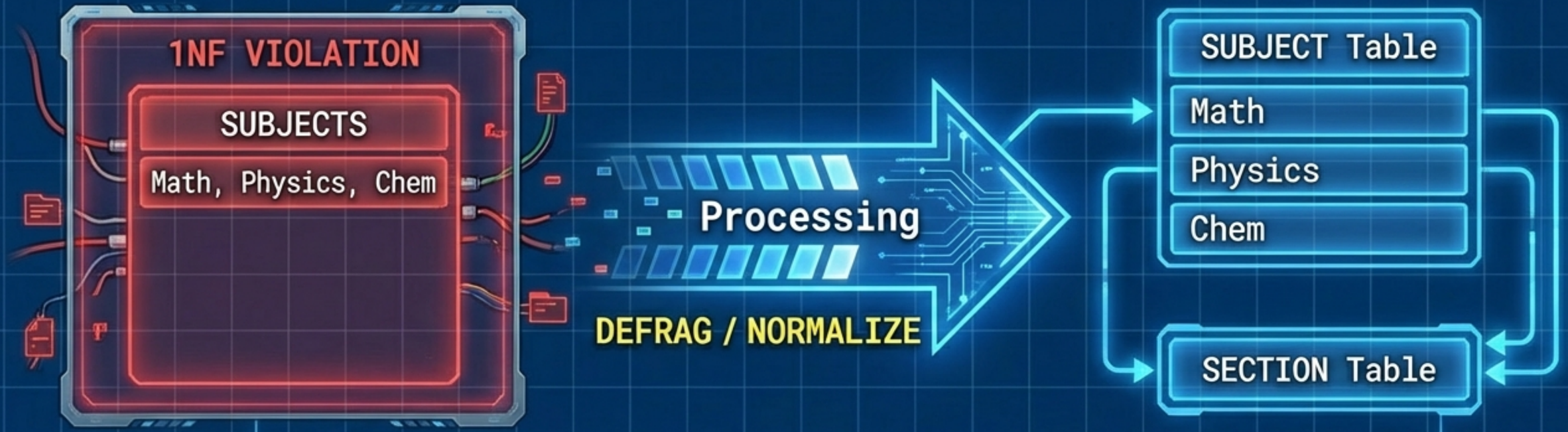
75%

current vs target

These metrics live in the Query, not the Table. Always calculate fresh.

NotebookLM

OPTIMIZATION MODE: NORMALIZATION



The 3NF Standard:

1. Atomic Values (No lists in cells).
2. No Partial Dependencies.
3. No Transitive Dependencies.

Result: Zero Data Redundancy.

NotebookLM

MISSION ACCOMPLISHED: SYSTEM ARCHITECTURE

SUMMARY

UNIFIED: Academic + Fitness + Nutrition in one schema.

SCALABLE: Supports multiple Universities (university_id).

INTELLIGENT: Derived attributes drive Decision Support.

THE BLUEPRINT IS COMPLETE. THE LOGIC IS SOUND. GAME ON.

NotebookLM