🏗️ Big Appetite OS — Build Plan (Non-Technical)

What We're Building

Think of Big Appetite OS as a **living business brain** with three main parts:

1. **Memory (Supabase)** — Where everything is stored and remembered
2. **Mind (Cursor + Logic)** — Where thinking and decisions happen
3. **Eyes & Hands (Dashboards)** — Where you see insights and take action

---

The Big Picture: How It All Works

The Core Loop:

```
WORLD → SIGNALS → UNDERSTANDING → DECISIONS → ACTIONS → OUTCOMES → LEARNING →
(repeat)
```

In plain English:

1. **Signals come in** from the world (customer messages, reviews, orders, social comments)
2. **System understands them** by figuring out who they're from and what they mean
3. **Knowledge grows** about customers (their preferences, behaviors, contradictions)
4. **Patterns emerge** as similar customers naturally group together (cohorts)
5. **Decisions are made** about what to say/offer based on these patterns
6. **Actions happen** (campaigns, offers, responses go out)
7. **Results come back** (did it work? who responded? what happened?)
8. **System learns** and gets smarter for next time

Why This Is Different

Traditional System:

- You tell it: "Send this offer to customers who like spicy food"
- It does exactly that, no learning
- You manually update rules when things change

Big Appetite OS:

- You feed it signals about customers
- **It discovers** that there's a group who likes spicy + values deals + orders late night
- **It learns** which offers work for which groups
- **It adapts** as customers change and new patterns emerge
- **It explains** why it made each decision (traceable reasoning)

The Five Core Layers

1. SENSING (Signals Schema)

What it does: Captures raw input from the world

Sources:

- WhatsApp messages from customers
- Reviews from Google/Yelp
- Orders from Shopify/POS
- Social media comments
- Website behavior
- Email interactions

Key principle: Store everything raw first, process later. Never lose the original.

---

2. UNDERSTANDING (Actors Schema)

What it does: Builds knowledge about people

For each person, it tracks:

- Who they are (demographics, identities)
- What they believe vs. what they actually do (contradictions)
- How certain we are about each belief (confidence scores)
- How their beliefs change over time (evolution)

Key principle: Everything is probabilistic. "This person probably likes spicy food (78% confident)" not "This person likes spicy food."

---

3. PATTERNS (Cohorts Schema)

What it does: Discovers natural groupings

No predefined categories — patterns emerge from data:

- "Group A: High spice + low price sensitivity + Instagram-driven"
- "Group B: Health-conscious + willing to pay premium + influenced by friends"

Cohorts evolve:

- New patterns form as data arrives
- Groups split when they become too different internally
- Groups merge when they become too similar
- Groups dissolve when members shift away

Key principle: Clusters are discovered, not designed. The system finds the patterns.

---

4. DECIDING (Stimuli Schema)

What it does: Creates responses and campaigns

Based on understanding + patterns:

- Which cohort should we target?
- What offer/message will resonate?
- How confident are we this will work?
- Why did we choose this? (reasoning recorded)

Key principle: Every decision is traceable back to the signals and beliefs that caused it.

---

5. LEARNING (Outcomes + AI Schema)

What it does: Measures results and improves

Tracks:

- What happened after each campaign/offer?
- Which cohorts responded? Which didn't?
- Which patterns predict success?
- Which logic functions are accurate?

Feeds back:

- Update beliefs based on behavior
- Refine cohort patterns
- Improve decision-making functions
- Trigger re-clustering when beliefs shift

Key principle: The loop closes. Outcomes become new signals, system continuously learns.

The Build: 5 Phases

Phase 1: Foundation (The Memory)

Goal: Set up Supabase with core schemas ready to receive data

What gets built:

- Database tables for all 5 layers
- Connections and relationships between tables
- Security rules (who can see what)
- Basic helper functions

End state: Empty database structure, ready to fill

Time estimate: 1-2 sessions

You'll know it's done when: You can manually insert a test signal and see it stored correctly

---

Phase 2: Signal Intake (The Senses)

Goal: Get real data flowing into the system

What gets built:

- Pick 1-2 signal sources to start (e.g., WhatsApp + Reviews)
- Build connectors to pull data in
- Actor matching logic (is this a new person or existing?)
- Basic belief extraction (what does this signal tell us?)

End state: Signals arriving automatically, actors being created/updated

Time estimate: 2-3 sessions

You'll know it's done when: A customer WhatsApp message automatically creates/updates their actor profile

---

Phase 3: Intelligence (The Mind)

Goal: System starts understanding and reasoning

What gets built:

- Bayesian belief update functions (how confidence changes with evidence)
- Contradiction detection (when behavior doesn't match words)
- Basic clustering algorithm (discover initial cohorts)
- Reasoning logs (why did the system believe X?)

End state: System making inferences, detecting patterns, explaining itself

Time estimate: 3-4 sessions

You'll know it's done when: You can ask "Why does the system think this person likes spicy food?" and get a traceable answer

---

Phase 4: Action (The Hands)

Goal: System generates and deploys stimuli

What gets built:

- Stimulus creation logic (generate offers/messages based on cohorts)
- Target selection (which cohort should receive this?)
- Deployment tracking (when did it go out? to whom?)
- Outcome recording (what happened?)

End state: System can recommend or deploy campaigns, measure results

Time estimate: 2-3 sessions

You'll know it's done when: System suggests "Target cohort A with spicy offer because..." and you can deploy it

---

Phase 5: Learning (The Loop)

Goal: Close the loop, system improves itself

What gets built:

- Outcome analysis (which stimuli worked? why?)
- Cohort evolution triggers (when to re-cluster)
- Performance tracking (are our functions accurate?)
- Feedback incorporation (outcomes update beliefs)

End state: Fully autonomous learning loop

Time estimate: 2-3 sessions

You'll know it's done when: After a campaign, system automatically updates beliefs and refines cohorts based on results

---

Phase 1 Detailed: Foundation

What Cursor Will Build:

1. **Database Schema Files**
   - SQL migration files for all tables
   - Organized by schema (core, signals, actors, cohorts, stimuli, ops, outcomes, ai)
2. **Table Structures**
   - All fields with correct types
   - Relationships (foreign keys)
   - Indexes for performance
   - Default values and constraints
3. **Row Level Security (RLS)**
   - Service role can do everything (for Cursor)
   - Brand-scoped access (each brand sees only their data)
   - Audit trails (who changed what)
4. **Helper Functions (Stubs)**
   - Basic functions created but not fully implemented yet
   - `match_or_create_actor()`

- `update_actor_belief()`
- `trigger_clustering_run()`
- etc.

5. **Seed Data**
   - Sample brand (Wing Shack)
   - Sample actors with beliefs
   - Sample signals
   - Demonstrates how data flows
6. **Documentation**
   - Comments explaining each table's purpose
   - ER diagram showing relationships
   - Setup instructions

What You'll Do:

1. **Give Cursor access** to your Supabase project (connection details)
2. **Review the schema** Cursor creates
3. **Run migrations** to create tables
4. **Validate** with seed data

Success Criteria:

✅ All tables exist in Supabase
✅ You can view them in Supabase dashboard
✅ Seed data loads successfully
✅ Relationships work (foreign keys connect correctly)
✅ You understand what each schema does

What Happens After Phase 1

Once foundation is built:

Phase 2: We pick your first signal source (probably WhatsApp) and build the intake pipeline

Phase 3: We add the intelligence layer (belief updates, clustering)

Phase 4: We build stimulus generation and deployment

Phase 5: We close the loop with outcome tracking and learning

Each phase builds on the previous, and you can use the system at increasing levels of sophistication.

Key Decisions for Phase 1

Before Cursor starts building, clarify:

1. Supabase Connection

- You have project ID
- You have service role key (for Cursor to access)
- You have anon key (for future frontend)

2. GitHub Setup

- Repo: `big-appetite-os` (ready ✓)
- Cursor will commit schema files there
- You'll review PRs (or Cursor commits directly if you prefer)

3. Starting Scope

For Phase 1, focus on:

- **1 brand** (Wing Shack)
- **Signal sources** (we'll wire them in Phase 2, but schema ready for all)
- **Actor structure** (full Bayesian schema)
- **Basic cohorts** (structure ready, clustering logic in Phase 3)

4. Development Approach

- Cursor builds locally, you review, then deploy to Supabase
- OR Cursor applies migrations directly to your Supabase project
- (Recommend: local first, review, then apply)

Non-Technical Summary

What you're building:

A system that **listens to customers, understands them, discovers patterns, makes smart decisions, and learns from results**.

Why it's special:

- **Self-aware:** It knows what it knows and doesn't know (confidence scores)
- **Emergent:** Patterns form naturally from data, not forced categories
- **Traceable:** Every decision can be explained back to the signals that caused it
- **Adaptive:** Gets smarter with every interaction

Your role:

- **Phase 1:** Review and validate the database structure
- **Phase 2+:** Guide which features to build next, provide domain knowledge, test the system

Cursor's role:

- Write the code
- Create the database structure
- Build the logic functions
- Connect the pieces

The outcome:

A living, learning business operating system that improves itself over time.

Ready for Phase 1 starter prompt? 🚀