```
/**
 * PROJECT: BuildWhileBleeding.com
 * DB: tactical-curriculum-db (25f74a5f-6030-41a9-ae14-061a2e594e13)
 * STACK: Cloudflare Worker + D1
 */

export interface Env {
  DB: D1Database;
}

const CORS_HEADERS = {
  "Access-Control-Allow-Origin": "*",
  "Access-Control-Allow-Methods": "GET, OPTIONS",
  "Content-Type": "application/json",
};

export default {
  async fetch(request: Request, env: Env): Promise<Response> {
    if (request.method === "OPTIONS") return new Response(null, {
headers: CORS_HEADERS });

    const { pathname } = new URL(request.url);

    try {
      // 1. GET ALL (Overview Mode)
      if (pathname === "/codex") {
        const { results } = await env.DB.prepare(
          "SELECT term, category FROM concepts ORDER BY category ASC,
term ASC"
        ).all();
        return new Response(JSON.stringify(results), { headers:
CORS_HEADERS });
      }

      // 2. GET SINGLE (Deep Dive + Relationships)
      // Path: /codex/term-name
      const term = decodeURIComponent(pathname.split("/")[2]);
      if (!term) throw new Error("Term Required");

      // Execution: Atomic Parallel Fetch
      const [concept, relations] = await Promise.all([
        env.DB.prepare("SELECT * FROM concepts WHERE term =
?").bind(term).first(),
        env.DB.prepare(`
          SELECT target_term as related, relationship_type as type,
rationale
          FROM concept_relationships WHERE source_term = ?
          UNION
          SELECT source_term as related, relationship_type as type,
rationale
          FROM concept_relationships WHERE target_term = ?
        `).bind(term, term).all()
      ]);

      if (!concept) return new Response("Not Found", { status: 404 });

      // FORMATED CONTENT RESPONSE
      return new Response(JSON.stringify({
        ...concept,
        graph: relations.results
      }), { headers: CORS_HEADERS });

    } catch (e: any) {
      return new Response(JSON.stringify({ error: e.message }), {
status: 500, headers: CORS_HEADERS });
    }
  }
};
```