



# AI ADVANTAGE AGENCY

AI INSIGHTS FOR BUSINESS FORESIGHT

## How to Add Product Schema for Agentic Commerce

Complete Implementation Guide — All Six Schema Types, Attribute Fill Rate Auditing, and Platform-Specific Deployment

|                            |                            |                             |
|----------------------------|----------------------------|-----------------------------|
| <b>6 SCHEMA TYPES</b>      | <b>JSON-LD CODE BLOCKS</b> | <b>95% FILL RATE GUIDE</b>  |
| <b>WordPress + Shopify</b> | <b>Validation Workflow</b> | <b>Maintenance Calendar</b> |

Published by **AI Advantage Agency** — aiadvantageagency.com

**Product schema for agentic commerce requires six schema types working together: Product, Offer, AggregateRating, Review, FAQPage, and ReturnPolicy.** Standard implementations include only the first two. The missing four are what separate products that get recommended by AI agents from products that get ignored. This guide gives you the complete JSON-LD code for all six types, an attribute fill rate audit framework, and a validation workflow that confirms your implementation is citation-ready, not just technically valid.

## The Quick Take

| Standard Product Schema                            | Agentic-Ready Product Schema   |
|--|--|
| <b>2 schema types: Product + Offer</b>             | 6 schema types: Product, Offer, AggregateRating, Review, FAQPage, ReturnPolicy       |
| <b>Basic attributes: name, price, availability</b> | Full attribute set: GTIN, MPN, brand, category, dimensions, warranty, return policy  |
| <b>Static implementation: set and forget</b>       | Maintained schema: price, availability, and ratings updated continuously             |
| <b>Goal: Google Rich Results eligibility</b>       | Goal: AI agent recommendation eligibility across ChatGPT, Perplexity, and Google AIO |
| <b>Attribute fill rate: typically 30-50%</b>       | Attribute fill rate: 95%+ required to avoid citation penalty                         |

**Bottom line: AI agents need structured, machine-readable answers across six decision criteria before they recommend a product. If your schema only covers price and availability, you are handing the recommendation to a competitor who filled in the rest.**

**Pro Tip:** The 95% attribute fill rate threshold is not arbitrary. AI agents use data completeness as a trust signal. Products falling below 80% fill rate see a measurable penalty in recommendation frequency regardless of product quality. You are not just filling in fields for Google. You are building the data layer that AI agents read to decide what to suggest to buyers.

## Standard vs. Agentic-Ready Schema: What Most Implementations Miss

**Standard product schema passes Google validation but fails the agentic commerce test.** Most implementations include a Product type with name, description, image, and an Offer with price and availability. That satisfies Rich

Results requirements. It does not satisfy what AI agents need to make a confident purchase recommendation on behalf of a buyer.

AI agents operating in an agentic commerce context do not just check whether a product is available. They compare return policies before recommending to risk-averse buyers. They verify social proof through review counts before recommending to new-to-brand buyers. They cross-reference unique identifiers (GTIN, MPN) against multiple product databases to confirm the listing is legitimate. Every missing field is a gap in the agent's confidence model.

| Schema Field           | Why AI Agents Need It   |
|------------------------|---|
| <b>GTIN / MPN</b>      | Cross-references listing against product databases for authenticity and pricing comparison  |
| <b>AggregateRating</b> | Confirms social proof before recommending to buyers who ask for "well-reviewed" options     |
| <b>ReturnPolicy</b>    | Resolves buyer risk objections without requiring the agent to visit a separate policy page  |
| <b>FAQPage</b>         | Answers product questions without forcing a follow-up crawl, keeping the agent on your data |
| <b>priceValidUntil</b> | Signals price freshness so agents do not flag your listing as potentially stale             |

**Pro Tip:** The gap between attribute-rich schema and generic product schema is not a matter of degree. It is a matter of eligibility. Products with comprehensive schema appear in AI recommendations 3-5x more frequently. The citation rate difference between attribute-rich and generic schema runs to more than 20 percentage points.

## The Complete Product Schema JSON-LD Code Block

The Product schema block carries the core identity layer for your product. Fill every field. Fields left empty or with placeholder values actively degrade your attribute fill rate score. The code below includes every recommended attribute for agentic commerce eligibility.

```

{
  "@context": "https://schema.org",
  "@type": "Product",
  "name": "Your Product Name",
  "description": "A complete, keyword-rich description. 2-3 sentences.",
  "image": [
    "https://yourdomain.com/images/product-main.jpg",
    "https://yourdomain.com/images/product-angle2.jpg"
  ],
  "sku": "YOUR-SKU-12345",
  "mpn": "MANUFACTURER-PART-NUMBER",
  "gtin12": "012345678901",
  "gtin13": "0012345678901",
  "brand": { "@type": "Brand", "name": "Your Brand Name" },
  "category": "Primary Category > Subcategory",
  "color": "Midnight Blue",
  "material": "Premium Nylon",
  "weight": { "@type": "QuantitativeValue", "value": "2.5", "unitCode": "LBR" },
  "height": { "@type": "QuantitativeValue", "value": "12", "unitCode": "INH" },
  "width": { "@type": "QuantitativeValue", "value": "8", "unitCode": "INH" },
  "depth": { "@type": "QuantitativeValue", "value": "4", "unitCode": "INH" },
  "manufacturer": { "@type": "Organization", "name": "Manufacturer Name" },
  "countryOfOrigin": { "@type": "Country", "name": "United States" },
  "additionalProperty": [
    { "@type": "PropertyValue", "name": "Warranty", "value": "2-year limited" },
    { "@type": "PropertyValue", "name": "Certification", "value": "CE Certified" }
  ]
}

```

## Product Schema Field Priorities

| Tier                               | Fields  |
|------------------------------------|---|
| <b>Tier 1 — Required</b>           | name, description, image, sku, brand, category, Offer (nested)                            |
| <b>Tier 2 — High Impact</b>        | gtin12 or gtin13, mpn, AggregateRating, color, material, weight, dimensions               |
| <b>Tier 3 — Authority Builders</b> | manufacturer, countryOfOrigin, additionalProperty (warranty, certifications), isVariantOf |

**Pro Tip:** If you only have one GTIN type, use gtin14 as the universal container field. It accepts any GTIN length and avoids validation errors. Always validate identifiers against the official GS1 registry before publishing.

## Want Your Products Recommended by AI Agents?

AI Advantage Agency builds and maintains full agentic commerce schema stacks for e-commerce brands. We handle implementation, validation, and ongoing maintenance so your product listings stay citation-ready as AI agents evolve.

[aiadvantageagency.com/services](https://aiadvantageagency.com/services)

## Offer Schema: Price, Availability, and Seller Attributes

**Offer schema is where AI agents confirm a product is purchasable right now, at a known price, from a trusted seller.** An Offer block with a stale price or missing priceValidUntil field signals to AI agents that your data may not be current. That kills recommendation eligibility faster than almost any other error.

```

{
  "@type": "Offer",
  "url": "https://yourdomain.com/products/your-product-slug/",
  "priceCurrency": "USD",
  "price": "149.00",
  "priceValidUntil": "2025-12-31",
  "availability": "https://schema.org/InStock",
  "itemCondition": "https://schema.org/NewCondition",
  "seller": {
    "@type": "Organization",
    "name": "Your Store Name",
    "url": "https://yourdomain.com"
  },
  "shippingDetails": {
    "@type": "OfferShippingDetails",
    "shippingRate": { "@type": "MonetaryAmount", "value": "0", "currency": "USD" },
    "deliveryTime": {
      "@type": "ShippingDeliveryTime",
      "handlingTime": { "@type": "QuantitativeValue",
        "minValue": "0", "maxValue": "1", "unitCode": "DAY" },
      "transitTime": { "@type": "QuantitativeValue",
        "minValue": "2", "maxValue": "5", "unitCode": "DAY" }
    },
    "shippingDestination": { "@type": "DefinedRegion", "addressCountry": "US" }
  },
  "hasMerchantReturnPolicy": { "@id": "#return-policy" }
}

```

**Update priceValidUntil at minimum quarterly, and immediately after any pricing change.** The availability field accepts only schema.org-defined URL values: <https://schema.org/InStock>, <https://schema.org/OutOfStock>, or <https://schema.org/PreOrder>. String values like "in stock" are not valid and generate Search Console warnings.

## AggregateRating and Review Schema Code Blocks

**AggregateRating** is the social proof layer that AI agents check before recommending a product to a buyer who asks for "the best" or "most popular" option in a category. Without it, your product competes for those queries with one hand tied behind its back.

### AggregateRating Schema

```
{
  "@type": "AggregateRating",
  "ratingValue": "4.7",
  "reviewCount": "284",
  "bestRating": "5",
  "worstRating": "1"
}
```

## Individual Review Schema

```
{
  "@type": "Review",
  "reviewRating": { "@type": "Rating", "ratingValue": "5", "bestRating": "5" },
  "name": "Review headline text goes here",
  "reviewBody": "Full text of the review. Write at minimum 2-3 sentences
    describing the reviewer's experience with the product.",
  "datePublished": "2024-11-15",
  "author": { "@type": "Person", "name": "Reviewer Name" },
  "publisher": { "@type": "Organization", "name": "Your Store Name" }
}
```

**Pro Tip:** Include 3-5 individual Review blocks, not just one. AI agents that process review sentiment look at the spread of individual reviews, not only the aggregate number. Prioritize recent reviews (within 12 months) with detailed reviewBody text over older brief reviews.

## FAQPage and ReturnPolicy Schema Code Blocks

**FAQPage and ReturnPolicy are the two schema types most consistently missing from e-commerce implementations.** They handle two of the most common AI agent decision criteria: "Does this answer my buyer's question?" and "What happens if the buyer needs to return it?"

### FAQPage Schema for Product Pages

Write at least 8 Q&A pairs focused on purchase decision criteria: sizing, compatibility, warranty, shipping, and use-case fit. Each answer should run 2-4 sentences and start with the direct answer.

```
{
  "@context": "https://schema.org",
  "@type": "FAQPage",
  "mainEntity": [
    {
      "@type": "Question",
      "name": "What sizes does this product come in?",
      "acceptedAnswer": {
        "@type": "Answer",
        "text": "This product comes in Small, Medium, Large, and XL.
                Size runs true to standard US sizing."
      }
    },
    {
      "@type": "Question",
      "name": "What warranty does this product include?",
      "acceptedAnswer": {
        "@type": "Answer",
        "text": "This product includes a 2-year limited manufacturer warranty
                covering defects in materials and workmanship."
      }
    }
  ]
  // Add minimum 6 more Q&A pairs ...
}
```

## ReturnPolicy Schema

**ReturnPolicy** schema resolves the single most common AI agent purchase blocker: **buyer uncertainty about what happens if the product does not meet expectations**. An absent ReturnPolicy block forces the agent to navigate to your policy page separately, which often breaks the recommendation flow.

```
{
  "@context": "https://schema.org",
  "@type": "MerchantReturnPolicy",
  "@id": "#return-policy",
  "applicableCountry": "US",
  "returnPolicyCategory":
    "https://schema.org/MerchantReturnFiniteReturnWindow",
  "merchantReturnDays": 30,
  "returnMethod": "https://schema.org/ReturnByMail",
  "returnFees": "https://schema.org/FreeReturn",
  "refundType": "https://schema.org/FullRefund",
  "inStoreReturnsOffered": false,
  "restockingFee": { "@type": "MonetaryAmount", "value": "0", "currency": "USD" }
}
```

## The Complete Stacked JSON-LD Block

Stacking all six schema types into a single JSON-LD script block using the `@graph` pattern is the correct implementation for agentic commerce. A single block is cleaner, reduces validation surface area, and is easier to maintain. Place it in the `<head>` of your product page.

Part 1 of 2 — Product, Offer, AggregateRating, and Review nodes:

```
<script type="application/ld+json">
{
  "@context": "https://schema.org",
  "@graph": [
    {
      "@type": "Product",
      "@id": "https://yourdomain.com/products/slug/#product",
      "name": "Your Product Name",
      "sku": "YOUR-SKU-12345",
      "gtin12": "012345678901",
      "brand": { "@type": "Brand", "name": "Your Brand Name" },
      "offers": {
        "@type": "Offer",
        "price": "149.00", "priceCurrency": "USD",
        "priceValidUntil": "2025-12-31",
        "availability": "https://schema.org/InStock",
        "hasMerchantReturnPolicy": { "@id": "#return-policy" }
      },
      "aggregateRating": {
        "@type": "AggregateRating",
        "ratingValue": "4.7", "reviewCount": "284",
        "bestRating": "5", "worstRating": "1"
      },
      "review": [{ "@type": "Review",
        "reviewRating": { "@type": "Rating", "ratingValue": "5" },
        "reviewBody": "Full review text. Minimum 2 sentences.",
        "datePublished": "2024-11-15",
        "author": { "@type": "Person", "name": "Reviewer Name" }
      }]
    }
  ],
}
```

Part 2 of 2 — MerchantReturnPolicy and FAQPage nodes (same @graph block):

```

{
  "@type": "MerchantReturnPolicy",
  "@id": "#return-policy",
  "returnPolicyCategory":
    "https://schema.org/MerchantReturnFiniteReturnWindow",
  "merchantReturnDays": 30,
  "returnFees": "https://schema.org/FreeReturn"
},
{
  "@type": "FAQPage",
  "mainEntity": [
    {
      "@type": "Question",
      "name": "Your FAQ question here?",
      "acceptedAnswer": {
        "@type": "Answer",
        "text": "Full answer text matching visible FAQ on page."
      }
    }
    // Repeat for all 8+ Q&A pairs
  ]
}
]
}
</script>

```

**Pro Tip:** Use @id references throughout the graph rather than nesting full objects wherever a reference will do. This reduces file size and makes schema easier to validate and maintain.

## Attribute Fill Rate: Audit It, Fix It, Maintain It

**Attribute fill rate** measures the percentage of available schema fields you actually populate across your **product catalog**. A fill rate below 95% exposes you to citation penalties. A fill rate below 80% actively reduces how often your products appear in AI-assisted purchase journeys.

| Fill Rate Range | Agentic Commerce Impact  |
|-----------------|--|
| 95%+            | Full recommendation eligibility. Products compete at the top of AI recommendation queues.                                      |
| 80-94%          | Partial eligibility. Products appear in some recommendation contexts but miss query types that require the missing attributes. |

| Fill Rate Range  | Agentic Commerce Impact   |
|------------------|---|
| <b>Below 80%</b> | Active citation penalty applies. AI recommendation frequency drops measurably vs. competing products above the threshold.                         |
| <b>Below 50%</b> | Effectively invisible to agentic commerce. Schema exists but does not meet the minimum trust threshold for AI agents processing purchase queries. |

## The Most Common Fill Rate Gaps

**Four fields account for the majority of fill rate deficits across e-commerce catalogs.** Fix these four first before auditing less common fields:

- AggregateRating with both ratingValue and reviewCount is empty on over 60% of product pages that have Product schema at all
- GTIN (gtin12, gtin13, or gtin14) requires a GS1-registered barcode number and many teams skip it because they lack the data
- priceValidUntil in Offer is present on fewer than 40% of product pages
- ReturnPolicy schema is absent on the vast majority of product pages outside enterprise retail

## Validation Workflow: From Valid to Citation-Ready

**Technical validity and citation-readiness are not the same thing.** A schema block can pass Google's Rich Results Test with zero errors and still fail to generate AI citations because the content within the fields is thin, stale, or mismatched with visible page content.

### Step 1: Google Rich Results Test

Run your product URL through [search.google.com/test/rich-results](https://search.google.com/test/rich-results). This confirms whether your Product schema qualifies for rich result display. Fix any errors before proceeding. A clean Rich Results Test is the baseline, not the finish line.

### Step 2: Schema Markup Validator

Run the same URL through [validator.schema.org](https://validator.schema.org). This tool checks against the full schema.org specification, including fields that Google does not validate but AI engines do read. Pay close attention to "Missing expected property" warnings in ReturnPolicy and FAQPage blocks.

### Step 3: Content Consistency Check

**Compare schema field values against visible page content.** The price in your Offer schema must match the displayed price. ReviewCount must match the visible review total. FAQ answers must match the visible FAQ text exactly. Mismatches flag the entire listing as a potential manipulation signal.

## Step 4: Freshness Check

Verify priceValidUntil is a future date. Confirm AggregateRating values match current review data. Check availability against actual inventory. Set a monthly calendar reminder and repeat immediately following any price change or inventory update.

| Common Error                               | How to Fix It   |
|--|---|
| <b>Missing required field: offers</b>      | Nest the Offer block inside the Product block, not as a separate sibling type                                 |
| <b>Invalid value for availability</b>      | Use full URL format: <a href="https://schema.org/InStock">https://schema.org/InStock</a> — not just "InStock" |
| <b>priceValidUntil is in the past</b>      | Update to a future date and add to your quarterly maintenance calendar  |
| <b>AggregateRating missing reviewCount</b> | Always include ratingValue and reviewCount as a pair. Neither works without the other.                        |
| <b>JSON syntax error (trailing comma)</b>  | Run the raw JSON through <a href="https://jsonlint.com">jsonlint.com</a> before publishing                    |

## Platform-Specific Implementation: WordPress, Shopify, Custom

Where and how you inject the JSON-LD block depends on your platform. The schema itself does not change. The delivery mechanism does.

### WordPress: Manual JSON-LD Wins for Agentic Commerce

RankMath Pro and Yoast both generate Product schema. Neither generates the full six-type agentic commerce stack without significant customization. RankMath Pro gets closer with custom schema blocks. For agentic commerce, inject the stacked JSON-LD block via the Insert Headers and Footers plugin or a conditional hook in your child theme's functions.php.

**Important WPBakery note:** Raw HTML blocks in WPBakery do not reliably render script tags. Do not inject schema through Raw HTML blocks. Always use the theme functions.php hook or the Insert Headers and Footers plugin.

### Shopify: Liquid Template Injection

Inject your JSON-LD into main-product.liquid or sections/product-template.liquid. Use Liquid object variables for dynamic fields so schema updates automatically when product data changes. Key Liquid variables:

```

"name": "{{ product.title | escape }}",
"description": "{{ product.description | strip_html | escape }}",
"sku": "{{ product.selected_or_first_available_variant.sku }}",
"price": "{{ product.price | money_without_currency }}",
"availability": "{% if product.available %}
  https://schema.org/InStock
{% else %}
  https://schema.org/OutOfStock
{% endif %}"

```

### Custom: Head Injection

On custom-built sites, inject the JSON-LD block in the head section of your product page template, after title and meta tags but before JavaScript files. Use server-side rendering to populate dynamic fields (price, availability, ratings) from the product database.

## Schema Maintenance: What to Update and When

**Schema is not a set-and-forget implementation.** Stale schema actively damages citation eligibility because AI engines that detect outdated field values flag the entire product listing as unreliable. Build a maintenance calendar at the same time you build the schema.

| Schema Block              | Update Trigger   |
|---------------------------|--|
| Offer (price)             | Immediately on any price change  |
| Offer (availability)      | Immediately on any inventory status change   |
| Offer (priceValidUntil)   | Quarterly minimum. Set calendar alert 30 days before expiration.                       |
| AggregateRating           | Monthly, or automated sync with review platform  |
| ReturnPolicy              | Only when your actual return policy changes  |
| Full validation (4 steps) | Monthly for priority products, quarterly for full catalog, after every platform update |

## The Bottom Line on Product Schema for Agentic Commerce

**Most e-commerce sites implement 30-40% of the schema they need to compete in agentic commerce.** The gap is not a technical problem. The code in this guide covers everything you need. The gap is a prioritization problem:

teams implement basic Product and Offer schema for Google Rich Results eligibility and stop there, not realizing that AggregateRating, ReturnPolicy, FAQPage, and Review schema are exactly what AI agents look for when deciding whose product to recommend.

The implementation pattern here, six schema types in a single @graph block, full attribute fill across every tier, and a maintained calendar that keeps data current, is what puts your products in front of buyers who arrive through AI recommendation channels. Agentic commerce conversion rates run 2-4x higher than traditional organic traffic because buyers who arrive through AI recommendations already trust the recommendation. The schema is what earns that recommendation in the first place.

**Start with the stacked JSON-LD block, validate it through both tools, audit your fill rate against the 95% threshold, and build the maintenance workflow before you move on.** Schema that you implement today and maintain consistently will compound in citation value as agentic commerce volume continues to grow.

## Ready to Make Your Products Agentic-Commerce Ready?

AI Advantage Agency builds, validates, and maintains full six-type schema stacks for e-commerce brands that want their products recommended by AI agents at scale. Implementation, QA, and ongoing maintenance included.

[aiadvantageagency.com/services](https://aiadvantageagency.com/services)

## Frequently Asked Questions

### Q: What is product schema for agentic commerce?

Product schema for agentic commerce is a complete set of six structured data types: Product, Offer, AggregateRating, Review, FAQPage, and ReturnPolicy, implemented as JSON-LD on product pages. This gives AI agents the data they need to evaluate, compare, and recommend products to buyers in autonomous purchase workflows. Standard product schema covers only Product and Offer. The additional four types are what qualify a product for AI recommendation eligibility.

### Q: How is agentic commerce schema different from standard product schema?

Standard product schema targets Google Rich Results eligibility with a Product type and nested Offer. Agentic commerce schema adds AggregateRating for social proof, Review blocks for sentiment data, FAQPage to answer buyer questions without additional page crawls, and ReturnPolicy to resolve purchase risk objections. It also enforces a 95% attribute fill rate threshold and requires ongoing maintenance to keep price, availability, and rating data current.

**Q: What is attribute fill rate and why does it matter?**

Attribute fill rate is the percentage of available schema fields that contain accurate, non-placeholder values. A fill rate below 95% reduces AI recommendation eligibility. A fill rate below 80% triggers an active citation penalty where products appear measurably less often in AI recommendation outputs.

**Q: Can I add all six schema types to one JSON-LD block?**

Yes. Use the @graph pattern to combine all six types into a single script block. Nest Offer, AggregateRating, and Review inside the Product node. Define MerchantReturnPolicy as a separate graph node with an @id value, then reference it from the Offer block using hasMerchantReturnPolicy. Add FAQPage as a third graph node.

**Q: How often should I update my product schema?**

Update Offer schema immediately on any price or availability change. Update AggregateRating monthly or automate it. Update priceValidUntil quarterly. Update ReturnPolicy only when your actual return policy changes. Run full four-step validation monthly for priority products and quarterly for the full catalog.

**Q: Does product schema affect how ChatGPT and Perplexity recommend products?**

Yes. ChatGPT's GPTBot and Perplexity both crawl and process structured data from product pages. Products with complete schema provide machine-readable data that feeds directly into product comparison and recommendation responses. Products without structured data require AI engines to parse unstructured page content, which is less reliable and less likely to produce accurate recommendations.

**Q: Should I use RankMath or manual JSON-LD for WordPress?**

Manual JSON-LD gives you full control and is the recommended approach for complete agentic commerce schema. RankMath Pro supports custom schema blocks but neither it nor Yoast generates the complete six-type stack natively. Inject the custom block through Insert Headers and Footers plugin or a conditional hook in your child theme's functions.php. Do not inject schema via WPBakery Raw HTML blocks.

**Q: What tools should I use to validate product schema?**

Use two tools in sequence. Run Google Rich Results Test at [search.google.com/test/rich-results](https://search.google.com/test/rich-results) to confirm Google eligibility and catch critical errors. Then run Schema Markup Validator at [validator.schema.org](https://validator.schema.org) for a full schema.org specification check that surfaces missing recommended fields the Google tool does not flag. Follow both with a manual content consistency check.

---

Published by AI Advantage Agency — AEO, Paid Media, and AI-Powered Marketing  
[aiadvantageagency.com](https://aiadvantageagency.com)