Palantir Data Engineering Certification

Volume 3 — Ontology, Semantic Modeling & Business Actions

How to Use This Volume

This volume explains **how Foundry turns data into operational decision systems**.

You should read this volume if you want to: - Understand how Palantir expects data to be consumed - Reason about exam questions involving business users, workflows, or applications - Avoid common semantic modeling mistakes

Ontology is one of the **most differentiating and heavily tested** areas of the certification.

Chapter 12 — Why the Ontology Exists

12.1 The Problem Ontology Solves

Traditional data platforms stop at datasets and dashboards. They assume that: - Users understand schemas - Users know how to join data - Users will not misuse fields

In reality: - Business users think in **entities and actions**, not tables - Dashboards do not support operational workflows - Raw datasets are easy to misinterpret

The Ontology exists to close this gap.

It transforms:

> “Rows and columns” into “Real-world objects and decisions”

12.2 Ontology as a Contract

The Ontology is a **semantic contract** between: - Data engineers - Domain experts - Application builders

Once defined, it guarantees: - Stable meaning - Controlled access - Predictable behavior

Changing ontology semantics is a **breaking change**, similar to changing an API.

Exam Insight: If a question involves stability, safety, or cross-team usage, ontology-based answers are often correct.

Chapter 13 — Core Ontology Concepts (Deep Definitions)

13.1 Objects

An Object represents a **real-world entity** that: - Has identity - Persists over time - Accumulates state

Examples: - Customer - Order - Asset - Case

An object must: - Have a stable primary key - Map cleanly to curated datasets

Objects are **not**: - Temporary aggregates - One-off metrics - Derived summaries

---

13.2 Properties

Properties are attributes of an object.

Examples: - Customer status - Order amount - Asset location

Properties: - Are typed - Can change over time - May come from multiple datasets

Important distinction: - Properties describe state - Metrics describe aggregates

Do not confuse the two.

13.3 Relationships

Relationships define how objects connect.

Examples: - Customer → places → Order - Asset → located at → Site

Good relationships: - Reflect real-world meaning - Are navigable - Preserve referential integrity

Bad relationships: - Encode temporary logic - Exist only for convenience

---

13.4 Actions

Actions represent **business operations**, not data transformations.

Examples: - Approve - Flag - Escalate - Assign

Actions: - Change object state - Trigger workflows - Are auditable

Actions are not: - Batch transformations - Data cleanup jobs

Chapter 14 — Mapping Datasets to Ontology

14.1 Curated Data as the Source of Truth

Ontology should be backed by: - Curated datasets only - Never raw datasets

Why: - Raw data may be inconsistent - Business meaning is unresolved - Quality checks are incomplete

The ontology assumes **trustworthy inputs**.

14.2 One Object, Multiple Datasets

An object may draw properties from: - A base entity table - Status change logs - Derived enrichment datasets

This is acceptable if: - Identity remains stable - Semantics are clear

Do not fragment object identity.

14.3 Handling Slowly Changing Attributes

Some properties: - Change rarely - Must be historically correct

Correct handling: - Preserve change history - Expose current state clearly

Never overwrite historical meaning silently.

Chapter 15 — Ontology Design Principles (Exam-Critical)

15.1 Domain-Driven Modeling

Ontology should reflect: - Business language - Domain concepts - User mental models

Avoid: - Technical naming - Source-system leakage

Ontology is for **humans**, not pipelines.

15.2 Minimal but Complete

Good ontology design: - Exposes what users need - Hides what they don’t

Overly rich ontology: - Confuses users - Increases maintenance risk

Under-modeled ontology: - Forces users back to datasets

15.3 Stability Over Optimization

Ontology should change: - Slowly - Deliberately

Frequent changes: - Break applications - Undermine trust

Exam Insight: If a solution requires frequent ontology changes, it is likely wrong.

Chapter 16 — Ontology and Security

16.1 Ontology as an Access Boundary

Ontology enforces: - Object-level permissions - Property-level visibility - Action-level authorization

This allows: - Safe self-service - Controlled exposure

Users interact with objects, not tables.

16.2 Why Ontology Is Safer Than Direct Dataset Access

Direct dataset access: - Requires schema knowledge - Risks misuse - Exposes sensitive fields

Ontology: - Abstracts complexity - Enforces policy - Limits blast radius

This is why Palantir strongly favors ontology-driven access.

Chapter 17 — Common Ontology Anti-Patterns

17.1 Treating Ontology as Metadata Only

This leads to: - Underpowered applications - Duplicate logic elsewhere

Ontology is executable, not descriptive.

17.2 Encoding Business Logic in Actions Only

Actions should: - Trigger workflows - Change state

They should not: - Replace curated transformations

Business logic belongs primarily in data pipelines.

17.3 Modeling Metrics as Objects

Metrics: - Are aggregates - Change with context

Objects: - Represent entities

Confusing the two causes semantic errors.

Chapter 18 — How Ontology Appears in the Exam

Typical scenarios: - “Enable business users to interact safely with data” - “Expose curated data without schema knowledge” - “Build operational workflows”

Correct answers: - Introduce ontology objects - Use actions for decisions - Restrict raw access

---

End of Volume 3

Next volumes: - Volume 4: Data quality, governance, security, lineage - Volume 5: Debugging, operations, and exam reasoning