Canonical Account Quote Model

Version 1.0.0 (meta rev 5)

Abstract

This document defines canonical JSON entity models for Account, Census, and Quote, along with references between models using JSON Schema 2020-12. It also provides a FHIR-aligned representation for Quote using standard resources for insurance quoting workflows.

1.1 Introduction

The canonical model organizes business data into three entity models: Account (the customer organization), Census (population snapshot used for underwriting), and Quote (pricing proposal derived from the Account and Census).

Each model is published as an independent JSON Schema with its own semantic version and metaVersion. Quote references Account and Census by model-level $ref, enabling consistent validation of embedded sub-objects across services and databases.

Figure 1 shows the high-level UML structure and relationships.

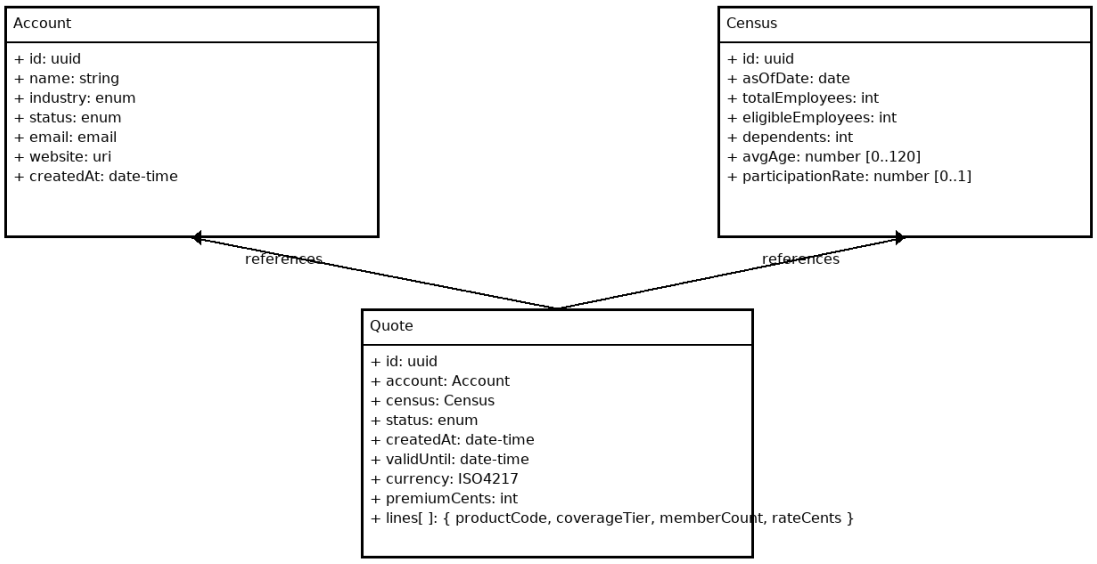

Figure 1 — UML-style structure: Quote references Account and Census.

1.2 Key Concepts

1. **Model-level references:** Use $ref to import the structure of another entity model at design time (not instance IDs).
2. **Semantic version:** SemVer string that signals meaning/shape changes (e.g., 1.2.0).
3. **Meta version:** Document revision counter for republishing without meaning change.
4. **Closed content:** Use additionalProperties:false (and when composed, unevaluatedProperties:false) to prevent typos.
5. **Reference options at instance level:** foreign keys, link objects, or embedded snapshots. (Out of scope here; this doc focuses on model-level typing.)
6. **Validation:** All schemas target JSON Schema Draft 2020-12 and use standard format, enum, and numeric constraints.

1.3 JSON Entity Modeling Best Practices

Naming: lowerCamelCase for properties; avoid abbreviations; positive booleans (isActive).

Identity: Use opaque IDs (UUID/ULID). Don’t encode semantics in identifiers.

Dates & money: ISO-8601 UTC date-time; currency in minor units (integer) or decimal string; avoid binary float for currency.

Nullability: Document semantics: missing = unknown; null = intentionally empty; model nullable with type: [T, "null"].

Versioning: Pin $id with semantic version; bump *metaVersion* on non-functional republishes; keep old IDs resolvable.

Documentation: Use description annotations; ship minimal and maximal examples; maintain a changelog.

1.4 Quote Model Structure — JSON

The Quote schema references Account and Census models by their versioned $id and $anchor. Business rules and field constraints are captured with standard JSON Schema keywords.

```
{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "$id": "https://schemas.example.com/models/Quote/1.0.0",
  "$anchor": "Quote",
  "title": "Quote",
  "type": "object",
  "additionalProperties": false,
  "model": {
    "canonical": "https://schemas.example.com/models/Quote",
    "version": "1.0.0",
    "metaVersion": 5,
    "status": "active",
    "dependsOn": {
      "https://schemas.example.com/models/Account": "^1.0.0",
      "https://schemas.example.com/models/Census": "^1.0.0"
    }
  },
  "properties": {
    "id": {
      "type": "string",
      "format": "uuid",
      "description": "Quote identifier (opaque UUID)."
    },
    "account": {
      "$ref": "https://schemas.example.com/models/Account/1.0.0#Account",
      "description": "The Account this quote is for (typed by the Account model)."
    },
    "census": {
      "$ref": "https://schemas.example.com/models/Census/1.0.0#Census",
      "description": "Census data used to produce this quote (typed by the Census model)."
    },
    "status": {
      "type": "string",
      "enum": [
        "draft",
        "proposed",
        "accepted",
        "rejected",
        "expired"
      ],
      "description": "Quote lifecycle status."
    },
    "createdAt": {
      "type": "string",
      "format": "date-time",
      "description": "UTC creation timestamp."
    },
    "validUntil": {
      "type": "string",
      "format": "date-time",
      "description": "UTC expiration timestamp for this quote."
    },
    "currency": {
      "type": "string",
      "pattern": "^[A-Z]{3}$",
      "description": "ISO 4217 currency (e.g., 'USD')."
    },
    "premiumCents": {
      "type": "integer",
      "minimum": 0,
      "description": "Total premium in minor units (e.g., cents)."
    },
    "lines": {
      "type": "array",
      "minItems": 1,
      "description": "Line items comprising the quote.",
      "items": {
        "type": "object",
        "additionalProperties": false,
        "properties": {
```

```
            "productCode": {
              "type": "string",
              "maxLength": 50,
              "pattern": "^[A-Z0-9_.-]{1,50}$",
              "description": "Product/plan identifier (system-specific)."
            },
            "coverageTier": {
              "type": "string",
              "enum": [
                "EE",
                "ES",
                "EC",
                "EF"
              ],
              "description": "Coverage tier: Employee, Employee+Spouse, Employee+Children, Employee+Family."
            },
            "memberCount": {
              "type": "integer",
              "minimum": 1,
              "description": "Members covered by this line."
            },
            "rateCents": {
              "type": "integer",
              "minimum": 0,
              "description": "Rate per member or unit, in minor units."
            }
          },
          "required": [
            "productCode",
            "coverageTier",
            "memberCount",
            "rateCents"
          ]
        }
      },
      "notes": {
        "type": [
          "string",
          "null"
        ],
        "maxLength": 2000,
        "description": "Underwriting comments or caveats."
      }
    },
    "required": [
      "id",
      "account",
      "census",
      "status",
      "createdAt",
      "currency",
      "premiumCents",
      "lines"
    ]
  }
```

Business Rules (informative):

- validUntil SHALL be >= createdAt (enforced in service logic).
- status transitions: draft → proposed → accepted | rejected; expired is terminal.
- currency SHALL be a valid ISO 4217 code (pattern enforces 3 letters; service validates membership).
- premiumCents equals the sum of lines.memberCount * lines.rateCents (computed rule).

1.5 Quote Model Structure — FHIR-aligned

A price quote can be represented in FHIR using the Claim resource with use='predetermination'. The adjudicated totals may be returned as a ClaimResponse. The customer account maps to Organization, and the census (population snapshot) maps to Group (type=person, actual=true).

Alternative/adjacent flow: CoverageEligibilityRequest/Response can be used to check eligibility and plan-specific benefits; however, for full pricing with line items, Claim/ClaimResponse is the canonical fit.

FHIR Bundle (Quote as Predetermination)

```
Bundle.type = 'collection'
  - Organization (Account)
  - Group (Census)
  - Claim (use='predetermination')
      .provider -> Organization
      .item[n].productOrService -> CodeableConcept (plan/benefit code)
      .item[n].quantity.value   -> memberCount
      .item[n].unitPrice        -> rate (Money)
  - ClaimResponse (optional result, totals)
```

Notes:

- Group captures the employee population; characteristics or contained members may be used depending on privacy and size.
- Organization identifies the quoting customer (your Account).
- Claim.item lines map naturally from Quote.lines; totals belong in ClaimResponse.
- Use standard terminologies where available; internal product codes may be profiled as CodeSystems.

1.5 Glossary

Semantic version	SemVer string indicating meaning/shape changes of the model (e.g., 1.2.0).
Meta version	Document revision counter for republishing without meaning change.
$id	Canonical, version-pinned URI of a schema, used as the base for $ref resolution.
$anchor	Local anchor name to reference a particular schema location (e.g., the root).
$ref	Reference to another schema location (can be absolute URL + #anchor).
additionalProperties:false	Disallow unspecified properties to catch typos and ensure closed content.
FHIR Claim (predetermination)	A 'what-if' claim used for quoting or cost estimation.
FHIR Group	Collection of persons used to represent a census/population.
FHIR Organization	Entity representing an account/customer organization.

1.6 References

- JSON Schema Draft 2020-12 — Validation: https://json-schema.org/draft/2020-12/json-schema-validation
- JSON (RFC 8259): https://www.rfc-editor.org/info/rfc8259
- JSON Pointer (RFC 6901): https://www.rfc-editor.org/info/rfc6901
- FHIR Claim (use: predetermination): https://build.fhir.org/claim.html
- FHIR ValueSet: claim-use: https://build.fhir.org/valueset-claim-use.html
- FHIR Group: https://build.fhir.org/group.html
- FHIR Organization: https://build.fhir.org/organization.html
- CoverageEligibilityRequest: https://build.fhir.org/coverageeligibilityrequest.html