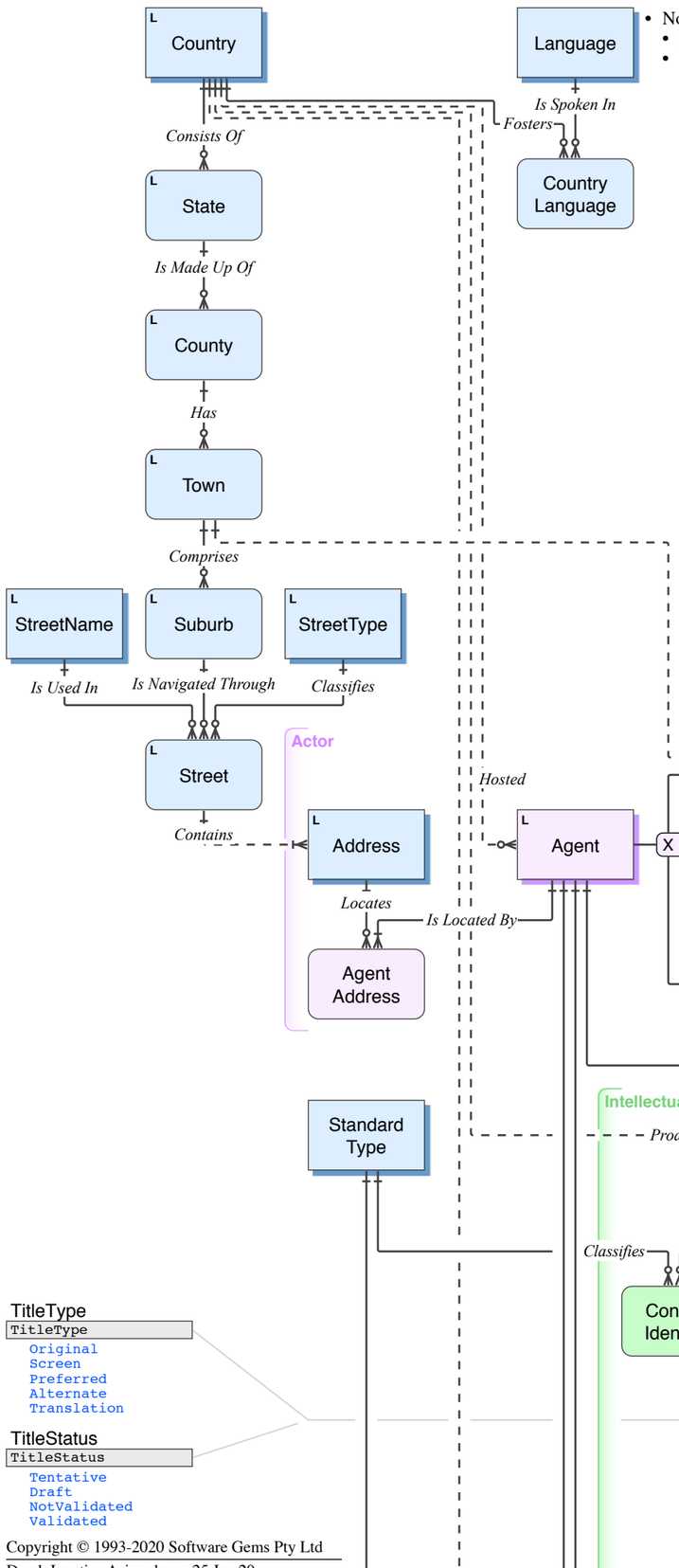
Table Relation



- Non-Identifying Relations (numerous)
  - not shown
  - instead it is signified by L.

Logical stage, showing tables and relations, equivalent to the theoretical "conceptual model" minus the isolation and limitations.

Copyright © 1993-2020 Software Gems Pty Ltd

Movie Title TR V0_8

Table Relation

SOFTWARE GEMS

Purpose
Purpose
- Industrial
- Training
- Scientific
- Amateur
- Ethno-anthropological
- Advertising
- Experimental

Genre
Genre

Duration
Duration

Length
Length

Audience
Audience

CrewRole
CrewRole
- Cinematograhper
- Director
- Presenter
- Producer
- ProductionCompany

TitleType
TitleType
- Original
- Screen
- Preferred
- Alternate
- Translation

TitleStatus
TitleStatus
- Tentative
- Draft
- NotValidated
- Validated

MediumType (Format)
MediumType
- 8mm
- Super8
- Video ?
- Half inch open reel
- U-Matic
- VHS
- DVD
- FileMP3...

D ArticleType

D AgentType

D CompositionType

Classifies

Project, Production

Produced

Played

Realisation

Is Rendered As

Is Varied As

Constructed

Is Registered As

Is Constructed By

Is Played By

Portrays

Recalls

Is Recalled By

Features

Is Featured In

Collection

Realisation Identifier

Realisation Crew

Realisation Cast

Realisation Character

Realisation Keyword

Realisation Keyword Direct

Component

Component Concept

Component Realisation

Component Edition

Comprises

Is Member Of

Collection Realisation

Is Known As

Realisation Title

Is Recalled By

Realisation Keyword Title

Manifests As

Post-Production

Package

Edition

Comprises

Dubbed

Is Member Of

Is Dubbed By

Is Registered As

Package Edition

Edition DubCrew

Edition Identifier

Is Distributed As

Merchandise Intake

Instance

Estate

Article

Article Undetermined

Comprises

Is Component Of

Estate Article

SG Relational Notation

Once Codd's 3NF ("full" Functional Dependency) is understood, the notation is simple to understand. The Functional Dependency is declared thus:

```
Table[ Key ] → ( Descriptor[ Value ], ... )
```

The notation, therefore, is:

```
Table { + | - } [ Key ] = ( Descriptor[ Value ], ... )
```

Where Key is `{ PK_Value | AK_Name[ Value ] }`

Where the operator is:

```
+ Insert
- Delete
= Update
```

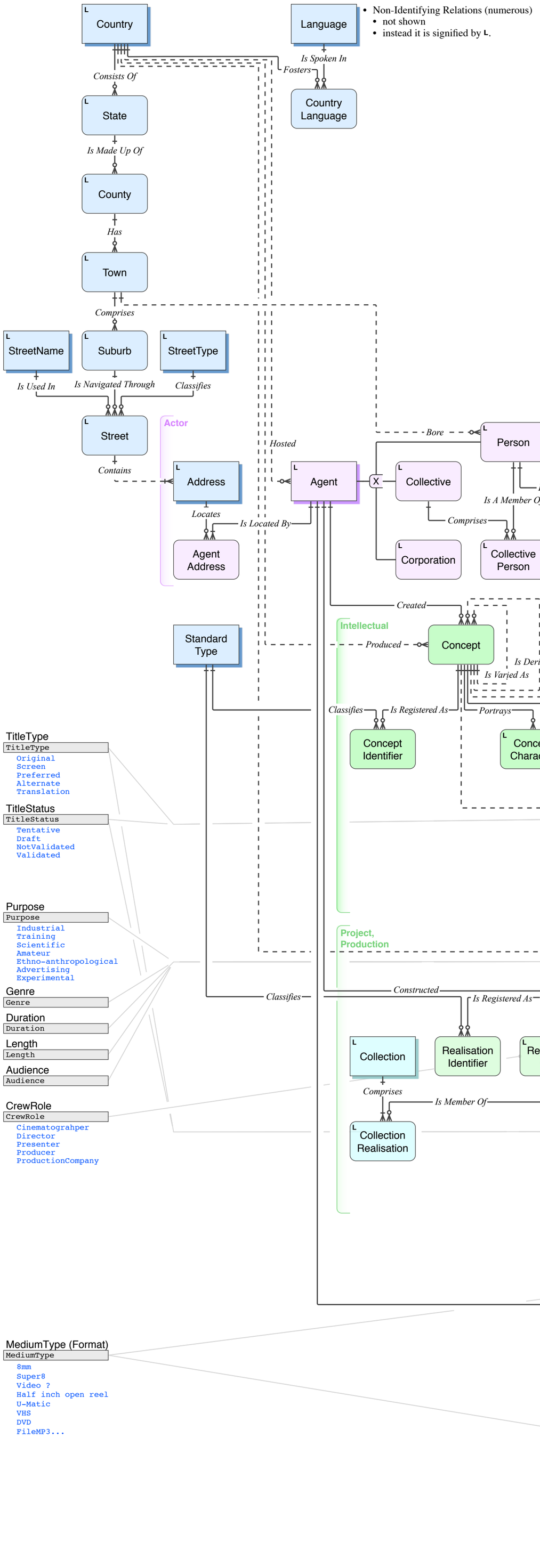
Example:

```
Person[ 123456 ] = Height[ 183 ]
Person[ Name[ Asirvadem, Derek ] ] = Weight[ 90 ]
Language[ LanguageCode ] → CharSetCode
```


Copyright © 1993-2020 Software Gems Pty Ltd

Table Relation • A2

Logical stage, showing tables and relations, equivalent to the theoretical "conceptual model" minus the isolation and limitations.



D ArticleType

D AgentType

D CompositionType

SG Relational Notation

Once Codd's 3NF ("full" Functional Dependency) is understood, the notation is simple to understand. The Functional Dependency is declared thus:

```
Table[ Key ] → ( Descriptor[ Value ], ... )
```

The notation, therefore, is:

```
Table { + | - } [ Key ] = ( Descriptor[ Value ], ... )
```

Where `Key` is `{ PK_Value | AK_Name[ Value ] }`

Where the operator is:

```
+ Insert
- Delete
= Update
```

Example:

```
Person[ 123456 ] = Height[ 183 ]
Person[ Name[ Asirvadem, Derek ] ] = Weight[ 90 ]
Language[ LanguageCode ] → CharSetCode
```


Copyright © 1993-2020 Software Gems Pty Ltd