This document summarizes the Sequelize models and relationships for the **Product**, **Release**, **Artist**, **Label**, and **ReleaseArtist** tables, and maps each model's fields to the example JSON output.

Product

- **id** (PK): auto-increment INTEGER primary key.
- **releaseId** (FK): INTEGER, NOT NULL, references `Release.id`. Maps to the JSON field `release_id` (e.g. `168350`raw.githubusercontent.com).
- **title**: STRING, NOT NULL. Source: JSON `title` (e.g. `"Evil Spirit"`raw.githubusercontent.com).
- **year**: INTEGER, NOT NULL. Source: JSON `year` (e.g. `1998`raw.githubusercontent.com).
- **catalogNumber**: STRING. Source: JSON `catno` (e.g. `"Evil 001"`raw.githubusercontent.com).
- **country**: STRING. Source: JSON `country` (e.g. `"UK"`raw.githubusercontent.com).
- **artwork**: STRING. Source: JSON `artwork` (e.g. `"1-productartwork.webp"`raw.githubusercontent.com).
- (Sequelize will also add `createdAt`/`updatedAt` TIMESTAMP fields.)

Associations: Product **belongsTo** Release (via `releaseId`), meaning each Product record links to one Release.

Release

- **id** (PK): auto-increment INTEGER primary key.
- **discogsId** (UNIQUE): INTEGER, NOT NULL. Stores the external Discogs release ID. Source: JSON `release_id`raw.githubusercontent.com (same value used for `Product.releaseId`).
- **title**: STRING, NOT NULL. Source: JSON `title`raw.githubusercontent.com.
- **year**: INTEGER. Source: JSON `year`raw.githubusercontent.com.
- **country**: STRING. Source: JSON `country`raw.githubusercontent.com.
- **catalogNumber**: STRING. Source: JSON `catno`raw.githubusercontent.com.
- **labelId** (FK): INTEGER, references `Label.id`. The label's name comes from JSON `label` (e.g. `"Evil Spirit"`raw.githubusercontent.com); the corresponding `Label.id` is looked up/inserted.
- (Also has `createdAt`/`updatedAt`.)

Associations: Release **belongsTo** Label (via `labelId`), and **belongsToMany** Artist through the `ReleaseArtist` join table. A Release **hasMany** Products (inverse of Product→Release).

Artist

- **id** (PK): auto-increment INTEGER primary key.
- **name**: STRING, NOT NULL, UNIQUE. Source: JSON `artists` string (e.g. `"Pressurehead / Heretik (2)"`raw.githubusercontent.com). When importing, split this string by `" / "` to create one **Artist** record per name (e.g. `"Pressurehead"` and `"Heretik (2)"`).
- (Has `createdAt`/`updatedAt`.)

Associations: Artist **belongsToMany** Release through `ReleaseArtist`. Each Artist can be linked to many Releases via the join table.

Label

- **id** (PK): auto-increment INTEGER primary key.
- **name**: STRING, NOT NULL, UNIQUE. Source: JSON `label` (e.g. `"Evil Spirit"`raw.githubusercontent.com).
- (Has `createdAt`/`updatedAt`.)

Associations: Label **hasMany** Release (one label can appear on many releases).

ReleaseArtist

- **id** (PK): auto-increment INTEGER primary key.
- **releaseId** (FK): INTEGER, NOT NULL, references `Release.id`.
- **artistId** (FK): INTEGER, NOT NULL, references `Artist.id`.

This is a pure join table (no JSON field) linking Releases and Artists. For each artist name extracted from JSON, a **ReleaseArtist** record is created associating the corresponding `Release.id` and `Artist.id`.

Associations: ReleaseArtist bridges Release ↔ Artist (many-to-many).

Summary of Relationships: A Product links to one Release (via `releaseId`). Each Release belongs to one Label (`labelId`) and to many Artists (through ReleaseArtist). Labels and Artists are standalone entities. ReleaseArtist records join specific Release and Artist records according to the JSON data. All fields above should match the example JSON structure: e.g. `title`, `year`, `country`, `catno`, `label`, `artists`, and `artwork` map to the fields shown, and foreign-key lookups link by label name and artist namesraw.githubusercontent.com.