3NF Normalization Report for Netflix & Hulu Datasets

1. Overview

This report details the normalization process of `netflix_titles.csv` and `hulu_titles.csv` to meet the 3rd Normal Form (3NF), resolving issues like multi-valued attributes, partial dependencies, and transitive dependencies.

2. Step 1: Achieve 1NF (Atomic Values)

Violations in Raw Data

- `listed_in`: Multi-valued attribute (e.g., "Comedy, Drama" is not atomic).
- `cast`: Multi-valued attribute (e.g., "Leonardo DiCaprio, Kate Winslet" is not atomic).

Solutions

- Split `listed_in` into a separate `genre` table (store unique genre names: "Comedy", "Drama").
- Split `cast` into a separate `actor` table (store unique actor names: "Leonardo DiCaprio").
- Use junction tables (`content_genre`, `content_actor`) to link `content` with `genre`/`actor` (many-to-many relationships).

3. Step 2: Achieve 2NF (Eliminate Partial Dependencies)

Violations

- Raw table contains `director_name` and `director_country`, which depend on `director_id` (not the primary key `content_id`) → partial dependency.

Solutions

- Split into a `director` table (store `director_id`, `director_name`, `director_country`).
- Retain only `director_id` (foreign key) in the `content` table to link with `director`.

4. Step 3: Achieve 3NF (Eliminate Transitive Dependencies)

Violations

- `actor_country` in the raw table depends on `actor_id` (not `content_id`) → transitive dependency.

Solutions

- Store `actor_country` only in the `actor` table; link `content` and `actor` via `content_actor` junction table.

5. Final 3NF Tables (Total 5)

1. `content` (core content information)
2. `director` (director information)

3. `genre` (genre information)
4. `actor` (actor information)
5. `content_genre` (association class for content-genre relationship)